Test Vector Chains for Increased Targeted and Untargeted Fault Coverage

Irith Pomeranz[1] and Sudhakar M. Reddy[2]
School of Electrical & Computer Eng. Purdue University W. Lafayette, IN 47907, U.S.A
Electrical & Computer Eng. Dept. University of Iowa Iowa City, IA 52242, U.S.A

Abstract - We introduce the concept of test vector chains, which allows us to obtain new test vectors from existing ones through single-bit changes without any test generation effort. We demonstrate that a test set T_0 has a significant number of test vector chains that are effective in increasing the numbers of detections of target faults, i.e., faults targeted during the generation of T_0, as well as untargeted faults, i.e., faults that were not targeted during the generation of T_0.

I. Introduction

Two concepts that were shown to be effective in producing high-quality test sets, which have high coverages of targeted and untargeted faults, are: (1) the generation of new test vectors from existing ones through single-bit changes [1]-[4], and (2) n-detection test generation where each target fault is detected by several different test vectors [5]-[10].

Considering a test vector t, new test vectors may be obtained from t by complementing single input values. For example, for t = 0000, a single-bit complementation results in 0001, 0010, 0100 or 1000. When t detects target faults, it sets up activation and propagation conditions for certain fault sites. Test vectors obtained from t by complementing single input values are likely to retain most of the conditions set up by t and thus detect additional target faults, or untargeted faults associated with the same fault sites. Single-bit complementation was used as part of a simulation-based test generation process in [1] and [4], and for built-in test generation in [2] and [3].

N-detection test sets, which contain multiple different test vectors for each target fault, were shown to be effective in detecting untargeted faults and in identifying defective chips [5]-[10]. Different test vectors for a target fault set up different activation and propagation conditions for the fault site. They are thus likely to detect untargeted faults or defects associated with the site of the target fault.

In this work we explore a new concept for defining sequences of single-bit changes in order to obtain new test vectors from existing ones. Given a test set T_0 for a target fault model, new test vectors are generated based on the test vectors in T_0 through sequences of single-bit changes that yield what we call *test vector chain*s. We demonstrate that, for a test set T_0, there are significant numbers of test vector chains that are effective in increasing the numbers of detections of target faults, i.e., faults targeted during the generation of T_0. These test vector chains are also effective in detecting untargeted faults, i.e., faults that were not targeted during the generation of T_0. Compared to n-detection test sets of the same size, and using a specific set of untargeted faults, we demonstrate that test sets generated based on test vector chains detect approximately the same numbers of untargeted faults as n-detection test sets generated by an n-detection test generation procedure. The advantage of test vector chains is that they do not require any test generation effort.

In the proposed approach, a sequence of single-bit changes is defined based on a *pair* of test vectors $t_1,t_2 \in T_0$. The single-bit changes are made such that they gradually modify t_1 into t_2. For example, let t_1 = 00001111 and t_2 = 10101001. A sequence of single-bit changes that modifies t_1 into t_2 may go through the following test vectors: t_1 = 00001111, 10001111, 10101111, 10101011, t_2 = 10101001. Each additional test vector is one bit further from t_1 and one bit closer to t_2. These test vectors constitute a test vector chain.

In Section II we provide a formal definition of a test vector chain. We also explain the reasons due to which test vector chains based on a test set T_0 are likely to be effective in increasing the numbers of detections of target faults and in detecting untargeted faults.

In Section III we describe a procedure for selecting effective test vector chains from a given test set T_0, and constructing a test set based on the selected test vector chains. Effectiveness is measured by the detection of target faults, or a limited number of untargeted faults.

Experimental results are given in Section IV. The results demonstrate the advantages and the limitations of test vector chains.

II. Test Vector Chains

Given a test set T_0, a test vector chain $C(t_1,t_2)$ is defined by two test vectors $t_1,t_2 \in T_0$. Let $t_i(j)$ be the value of input j under t_i. Let $diff(t_1,t_2)$ be the set that includes -1 and in addition every input j such that $t_1(j) \neq t_2(j)$. For example, with t_1 = 00001111 and t_2 = 10101001 (considered above) we obtain $diff(t_1,t_2)$ = {-1, 0, 2, 5, 6}. For every $j \in diff(t_1,t_2)$, we define a test vector $t^j_{1,2}$ and include it in $C(t_1,t_2)$. The vector $t^j_{1,2}$ is defined as follows: (1) For $k \leq j$, $t^j_{1,2}(k) = t_2(k)$. (2) For $k > j$, $t^j_{1,2}(k) = t_1(k)$.

1. Research supported in part by SRC Grant No. 2004-TJ-1244.
2. Research supported in part by SRC Grant No. 2004-TJ-1243.

978-1-4244-1922-7/08/$25.00 ©2008 IEEE

In the example above, we obtain $t_{1,2}^{-1} = 00001111 = t_1$, $t_{1,2}^{0} = 10001111$, $t_{1,2}^{2} = 10101111$, $t_{1,2}^{5} = 10101011$ and $t_{1,2}^{6} = 10101001 = t_2$. Starting from t_1, t_1 is modified into t_2 bit-by-bit until t_2 is obtained.

It is possible to change the order of the inputs and obtain different chains for the same pair of test vectors. Throughout this work we keep the inputs at a fixed order and define a single chain for each pair of test vectors. Under this definition, $C(t_1,t_2) \neq C(t_2,t_1)$.

Let $C(t_1,t_2) = \{t_{1,2}^{j_0}, t_{1,2}^{j_1}, \cdots, t_{1,2}^{j_{m-1}}\}$. We have that $t_{1,2}^{j_0} = t_1$, $t_{1,2}^{j_{m-1}} = t_2$, and $t_{1,2}^{j_k}$ differs from $t_{1,2}^{j_{k-1}}$ by one bit, for $1 \le k < m$. Thus, considering the chain of test vectors $C(t_1,t_2)$ starting from t_1 and ending at t_2, each additional test vector is one bit further from t_1 and one bit closer to t_2. This is likely to have several effects on the subsets of faults detected by the test vectors in $C(t_1,t_2)$.

(1) Suppose that t_1 detects a set of faults F_1. The test vectors $t_{1,2}^{j_k}$, for the lower values of k, are likely to detect all or most of the faults in F_1. Consequently, using $C(t_1,t_2)$ increases the numbers of detections of faults in F_1, and it is likely to improve the fault coverage of untargeted faults associated with the same fault sites.

(2) Suppose that t_2 detects a set of faults F_2. The test vectors $t_{1,2}^{j_k}$, for the higher values of k, are likely to detect all or most of the faults in F_2. Consequently, using $C(t_1,t_2)$ increases the numbers of detections of faults in F_2, and it is likely to improve the fault coverage of untargeted faults associated with the same fault sites.

(3) During the transition from detecting the faults in F_1 to detecting the faults in F_2, additional faults may be detected, that are not included in F_1 or F_2. This will increase the numbers of detections of such faults and the likelihood of detecting untargeted faults associated with the same fault sites.

The number of test vectors in a chain is at most N_{PI}+1, where N_{PI} is the number of primary inputs. This upper bound is reached when t_2 is the complement of t_1.

III. Selecting Test Vector Chains

For a test set T_0 that consists of M_0 test vectors, it is possible to define $M_0(M_0-1)$ test vector chains, one for each ordered pair of test vectors from T_0. Certain test vector chains may be more effective than others in increasing the numbers of detections of target faults, or in detecting untargeted faults (untargeted faults are ones that are not targeted during the generation of T_0). In this section we describe a procedure for selecting the most effective test vector chains for a given test set T_0. Of the test vectors included in these test vector chains we then select test vectors to be included in a test set denoted by T_{chains}.

To measure the quality of test vector chains, we use a set of faults G. When test vector chains are evaluated based on the detection of untargeted faults, G may be a set of faults that were not targeted during the generation of T_0. When test vector chains are evaluated based on their ability to detect target faults multiple times, we assume for simplicity of presentation that G will include n copies of each target fault, for a constant n, and that at most one copy will be considered detected by each test vector (in practice it is sufficient to maintain a count of the number of detections for each fault).

We select test vector chains $C(t_1,t_2)$, where $t_1,t_2 \in T_0$, by using a fault simulation process where the faults in G are simulated with fault dropping under test vector chains from T_0. The fault simulation process is given by Procedure 1 below. During Procedure 1, we identify an initial set of test vector pairs P whose chains are effective in detecting the faults in G. After P is computed, it is reduced by a reverse order simulation process called forward-looking reverse order fault simulation [11]. Finally, using the test vectors included in the chains defined by P, we define a test set T_{chains} that consists of test vectors that detect the faults of G. Additional details are explained following Procedure 1.

Procedure 1: Identifying effective test vector chains

(1) Let T_0 be a given test set. Let G be a set of faults. Set $ind(g) = -1$ for every $g \in G$.

[Initial set P]

(2) Set $\hat{G} = G$. Simulate $\hat{G}$ under T_0 and remove detected faults from $\hat{G}$. Set $P = \phi$. Set $i = 0$.

(3) For every pair of test vectors $t_1,t_2 \in T_0$, considering the test vector pairs in a random order:
   (a) Find the test vector chain $C(t_1,t_2)$.
   (b) Simulate $\hat{G}$ under $C(t_1,t_2)-\{t_1,t_2\}$. For every detected fault $g \in \hat{G}$, set $ind(g) = i$ and remove g from $\hat{G}$.
   (c) If any fault in $\hat{G}$ was detected due to $C(t_1,t_2)$, add t_1,t_2 to P and set $i = i+1$.

[Forward-looking reverse order simulation]

(4) Set $\hat{G} = G$. Simulate $\hat{G}$ under T_0 and remove detected faults from $\hat{G}$.

(5) Let $P = \{p_0,p_1, \cdots ,p_{m-1}\}$ where p_i is the ith vector pair added to P. Set $i = m-1$.

(6) Let p_i consist of the test vectors t_1 and t_2. If there is no fault $g \in \hat{G}$ with $ind(g) = i$, remove p_i from P. Otherwise, simulate $\hat{G}$ under the test vectors in $C(t_1,t_2)-\{t_1,t_2\}$, and remove detected faults from $\hat{G}$.

(7) Set $i = i-1$. If $i \ge 0$ go to Step 6.

[Test set T_{chains}]

(8) Set $\hat{G} = G$. Simulate $\hat{G}$ under T_0 and remove detected faults from $\hat{G}$. Set $T_{chains} = T_0$.

(9) For every $p_i \in P$:
   Let p_i consist of the test vectors t_1 and t_2. For every test vector $t \in C(t_1,t_2)-\{t_1,t_2\}$:
      Simulate $\hat{G}$ under t. Remove detected faults from $\hat{G}$. If t detected any fault, add t to T_{chains}.

During Steps 2-3 of Procedure 1, an initial set P is constructed for detecting as many faults in G as possible. We consider the test vector pairs in a random order in these steps to avoid selecting large numbers of pairs with the same first or second component.

It is possible that Steps 2-3 of Procedure 1 will include redundant test vector pairs in P. A test vector pair t_1,t_2 is redundant if the faults detected by $C(t_1,t_2)-\{t_1,t_2\}$ are also detected by test vector chains of other vector pairs in P. We perform forward-looking reverse order fault simulation [11] in Steps 4-7 of Procedure 1. This guarantees that no redundant test vector pairs will be left in P.

After P is reduced by forward-looking reverse order fault simulation, we simulate all the test vector chains based on P, and store in a test set T_{chains} the test vectors that are effective in detecting faults from G. This is done in Steps 8-9 of Procedure 1.

IV. Experimental Results

Experimental results of Procedure 1 are reported in this section to demonstrate the effectiveness of the notion of test vector chains. The circuits considered are full-scan versions of ISCAS-89 and ITC-99 benchmarks.

A. Experiments

We applied Procedure 1 using a one-detection test set for single stuck-at faults [14] as the test set T_0. For the set G we considered two options in two experiments.

(1) A set that consists of 10 copies of each single stuck-at fault detected by T_0. In this case, our goal is to obtain T_{chains} that is a 10-detection test set. In the actual implementation, instead of duplicating each fault, we associate a number of detections with each fault.

(2) A set of four-way bridging faults [12]-[13]. For a circuit with L lines, we included in G $c \cdot L$ randomly selected four-way bridging faults, for an integer $1 \leq c \leq 10$. We use $c = 10$ if the size of T_{chains} does not exceed 10 times the size of T_0 for this value of c, or a smaller value of c such that the size of T_{chains} is approximately 10 times the size of T_0.

For comparison, we performed n-detection test generation to obtain a test set T_{ndet} of the same size as T_{chains}. The n-detection test generation procedure we used [10] starts from $n = 1$, and it increases n until the size of the test set reaches the size of T_{chains}.

For further comparison, we considered a set of $100L$ randomly selected four-way bridging faults. We denote this set of faults by $BR100$. $BR100$ represents a large set of untargeted faults (faults that were not targeted during the generation of T_0, T_{chains} or T_{ndet}). Through the fault coverage obtained with respect to $BR100$ we will measure the ability of a test set to detect untargeted faults, and indirectly, defects. We simulated $BR100$ under T_0, and under T_{chains} and T_{ndet}, which have the same size. In addition, we performed n-detection fault simulation of T_0, T_{chains} and T_{ndet} to determine the minimum number of times any single stuck-at fault is detected by these test sets. We are interested in the minimum, and not in the average number of detections, since the average is typically very high for all the test sets.

All the test sets T_0, T_{chains} and T_{ndet} detect all the detectable single stuck-at faults in the circuit.

B. Results for 10-detections

The results using T_0 which is a deterministic compact one-detection stuck-at test set [14], and G which consists of 10 copies of each stuck-at fault detected by T_0, are shown in Table 1.

Under column *initial* we show the number of test vectors in T_0, the minimum number of detections of a single stuck-at fault detected by T_0, and the fault coverage of T_0 with respect to $BR100$.

TABLE 1
Deterministic Stuck-at Test Set and 10-detections

	initial			chains				n-detect	
circuit	vect	nmin	br100	pairs	vect	nmin	br100	nmin	br100
s208	27	1	72.88	65	315	9	81.09	11	81.00
s298	24	1	84.48	62	297	10	92.75	12	92.68
s344	15	1	78.05	39	217	10	90.62	15	90.66
s400	24	1	84.53	43	327	10	92.70	13	93.25
s420	43	1	72.68	101	516	2	80.39	11	82.66
s510	54	1	72.47	154	617	5	77.79	11	79.06
s526	50	1	86.10	102	630	10	92.73	12	93.02
s641	22	1	85.84	46	405	10	95.02	17	97.44
s820	94	1	69.57	245	1294	9	75.10	13	75.01
s953	76	1	76.72	91	860	10	80.39	11	81.04
s1196	138	1	78.88	199	1616	10	84.55	13	83.86
s1423	26	1	83.15	48	810	10	93.45	30	95.57
s5378	100	1	89.71	94	2391	10	94.17	24	97.36
s9234	111	1	81.16	144	3645	10	87.77	32	92.13
s15850	97	1	86.38	115	4416	10	93.30	45	97.60
average	60	1	80.17	103	1224	9	87.45	18	88.82

Under column *chains* we show the results of Procedure 1. We show the number of pairs in P, the number of test vectors in T_{chains}, the minimum number of detections of a stuck-at fault detected by T_{chains}, and the fault coverage of T_{chains} with respect to $BR100$.

Under column $n-detect$ we show information about the n-detection test set T_{ndet} that has the same size as T_{chains}. We show the minimum number of detections for a stuck-at fault detected by T_{ndet}, and the fault coverage of T_{ndet} with respect to $BR100$.

The following points can be seen from Table 1.

(1) Several test vector chains based on T_0 are effective for every circuit in increasing the numbers of detections of target stuck-at faults.

(2) In most of the circuits considered, test vector chains based on T_0 increase the numbers of detections of stuck-at faults to the target of 10. However, there are also circuits for which the test vector chains are unable to reach 10-detections for all the faults. This is due to the fact that test vector chains produce a limited set of test vectors. Considering the average numbers of detections (not reported here) we note that even when the minimum number of detections is smaller than 10, the average is close to 10, indicating that most of the faults are detected 10 times.

(3) The size of T_{chains} is somewhat larger than 10 times the size of T_0. For n-detection test sets generated by n-detection test generation procedures, the size of an n-detection test set is approximately n times the size of a one-detection test set. As a result, for T_{ndet} that has the same size as T_{chains}, the minimum number of detections of a stuck-at fault is always somewhat above 10.

(4) The fault coverage of T_{chains} with respect to $BR100$ is very close to that of T_{ndet}, and sometimes higher. It is significantly higher than that of T_0. This is important since $BR100$ represents a large set of untargeted faults.

In the comparison of T_{chains} with T_{ndet} it should be noted that the generation of T_{ndet} requires a test generation process, while generation of T_{chains} involves no test generation effort. Nevertheless, T_{chains} is similar to T_{ndet} in its ability to detect faults n times and in its ability to detect untargeted faults.

C. Results for Untargeted Faults

The results using T_0 which is a deterministic compact stuck-at test set [14], and G which is a set of $c \cdot L$ four-way bridging faults, are shown in Table 2. After the circuit name we show the value of c. Under column *initial* we show the number of test vectors in T_0, the fault coverage of T_0 with respect to G, and the fault coverage of T_0 with respect to $BR100$.

TABLE 2
Deterministic Stuck-at Test Set and Untargeted Faults

		initial			chains					n-detect	
circuit	c	vect	G	br100	pairs	vect	G	nmin	br100	nmin	br100
s208	10	27	72.26	72.88	33	111	81.84	1	81.10	4	80.12
s298	10	24	84.14	84.48	30	112	93.62	1	92.67	4	91.90
s344	10	15	78.61	78.05	26	128	92.77	1	91.62	9	89.82
s400	10	24	84.41	84.53	34	145	94.33	1	93.61	5	92.94
s420	10	43	72.60	72.68	84	261	83.98	1	81.21	5	81.72
s510	10	54	71.93	72.47	86	221	78.78	1	77.89	4	78.49
s526	10	50	87.18	86.10	51	210	94.30	1	92.74	4	92.21
s641	8	22	85.78	85.84	53	220	97.78	1	96.84	9	97.16
s820	10	94	69.12	69.57	93	358	76.03	1	74.97	3	73.71
s953	10	76	76.69	76.72	94	314	82.35	1	81.19	4	80.55
s1196	10	138	79.09	78.88	149	529	86.78	1	85.31	4	82.84
s1423	2	26	83.07	83.15	48	255	96.19	1	93.83	9	95.19
s5378	3	100	89.19	89.71	184	981	97.41	1	95.66	9	97.03
s9234	1	111	80.52	81.16	212	979	92.64	1	89.43	8	91.45
s15850	1	97	86.51	86.38	234	1328	97.53	1	95.72	13	97.42
average	8	60	80.07	80.17	94	410	89.76	1	88.25	6	88.17

Under column *chains* we show the results of Procedure 1. We show the number of pairs in P, the number of test vectors in T_{chains}, and the fault coverage of T_{chains} with respect to G. Next, we show the minimum number of detections of a stuck-at fault detected by T_{chains}, and the fault coverage of T_{chains} with respect to $BR100$.

Under column $n-detect$ we show information about the n-detection test set T_{ndet}. We show the minimum number of detections of a stuck-at fault detected by T_{ndet}, and the fault coverage of T_{ndet} with respect to $BR100$.

Overall, the results of Table 2 are consistent with the results of Table 1. Specifically, the following points can be seen from Table 2.

(1) Several test vector chains are effective for every circuit in increasing the fault coverage with respect to G.

(2) For the circuits considered, the fault coverage with respect to G increases by approximately 10%. This increase is achieved without any test generation effort.

(3) The size of T_{chains} is approximately 5-10 times the size of T_0. For T_{ndet}, this is also the minimum number of detections of stuck-at faults. For T_{chains}, the minimum number of detections of stuck-at faults is significantly smaller. This indicates that T_{chains} does not increase the number of detections uniformly for all the faults. Nevertheless, it is able to detect untargeted faults by increasing the numbers of detections for certain faults.

(4) The fault coverage of T_{chains} with respect to $BR100$ is very close to that of T_{ndet}, and sometimes higher. It is always significantly higher than that of T_0 ($BR100$ was not considered during Procedure 1).

V. Concluding Remarks

We defined a test vector chain as a sequence of test vectors that starts from a test vector t_1, and gradually modifies t_1 into another test vector t_2 by performing single-bit changes. We demonstrated that a test set T_0 has a significant number of test vector chains that are effective in increasing the numbers of detections of target faults, i.e., faults targeted during the generation of T_0, as well as untargeted faults, i.e., faults that were not targeted during the generation of T_0. Compared to n-detection test sets of the same size, and using a specific set of untargeted faults, we demonstrated that test sets based on test vector chains detect approximately the same numbers of untargeted faults as n-detection test sets. The main advantage of test sets based on test vector chains is that they do not require any test generation effort. The main limitation of test vector chains is that in some cases, they may not provide a sufficient number of different tests for a target fault. However, this occurred only rarely in benchmark circuits.

References

[1] J. Snethen, "Simulation-Oriented Fault Test Generator", in Proc. Design Autom. Conf., 1977, pp. 88-93.

[2] P. Girard, C. Landrault, V. Moreda and S. Pravossoudovitch, "An Optimized BIST Test Pattern Generator for Delay Testing", in Proc. VLSI Test Symp., 1997, pp. 94-100.

[3] K.-H. Tsai, J. Rajski and M. Marek-Sadowska, "Scan Encoded Test Pattern Generation for BIST", in Proc. Intl. Test Conf., 1997, pp. 548-556.

[4] I. Pomeranz and S. M. Reddy, "Sensitivity Levels of Test Patterns and Their Usefulness in Simulation-Based Test Generation", in Proc. Intl. Conf. on Computer Design, 2000, pp. 389-394.

[5] S. C. Ma, P. Franco and E. J. McCluskey, "An Experimental Chip to Evaluate Test Techniques Experiment Results", in Proc. Intl. Test Conf., 1995, pp. 663-672.

[6] S. M. Reddy, I. Pomeranz and S. Kajihara, "Compact Test Sets for High Defect Coverage", IEEE Trans. on Computer-Aided Design, Aug. 1997, pp. 923-930.

[7] M. R. Grimaila, S. Lee, J. Dworak, K. M. Butler, B. Stewart, H. Balachandran, B. Houchins, V. Mathur, J. Park, L.-C. Wang and M. R. Mercer, "REDO - Random Excitation and Deterministic Observation - First Commercial Experiment", in Proc. VLSI Test Symp., 1999, pp. 268-274.

[8] B. Benware, C. Schuermyer, N. Tamarapalli, K.-H. Tsai, S. Ranganathan, R. Madge, J. Rajski and P. Krishnamurthy, "Impact of multiple-detect test patterns on product quality", in Proc. Intl. Test Conf., 2003, pp. 1031-1040.

[9] S. Venkataraman, S. Sivaraj, E. Amyeen, S. Lee, A. Ojha and R. Guo, "An Experimental Study of n-Detect Scan ATPG Patterns on a Processor", in Proc. VLSI Test Symp., 2004, pp. 23-28.

[10] I. Pomeranz and S. M. Reddy, "Forming *N*-Detection Test Sets from One-Detection Test Sets Without Test Generation", in Proc. Intl. Test Conf., 2005, pp. 527-535.

[11] I. Pomeranz and S. M. Reddy, "Forward-Looking Fault Simulation for Improved Static Compaction", IEEE Trans. on Computer-Aided Design, Oct. 2001, pp. 1262-1265.

[12] S. Sengupta et. al., "Defect-Based Tests: A Key Enabler for Successful Migration to Structural Test", Intel Technology Journal, Q.1, 1999.

[13] V. Krishnaswamy, A. B. Ma, P. Vishakantaiah, "A Study of Bridging Defect Probabilities on a Pentium (TM) 4 CPU", Intl. Test Conf., 2001, pp. 688-695.

[14] S. Kajihara, I. Pomeranz, K. Kinoshita and S. M. Reddy, "Cost-Effective Generation of Minimal Test Sets for Stuck-at Faults in Combinational Logic Circuits", IEEE Trans. on Computer-Aided Design, Dec. 1995, pp. 1496-1504.