Test Vector Chains for Increased Targeted and Untargeted Fault Coverage

Irit Pomeranz
School of Electrical & Computer Eng.
Purdue University
W. Lafayette, IN 47907, U.S.A

and

Sudhakar M. Reddy
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 \( T_0 \). 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: \( t_1 = 00001111, 10001111, 10101111, 10101101, t_2 = 10101001 \). 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.

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 chains. 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: \( t_1 = 00001111, 10001111, 10101111, 10101101, t_2 = 10101001 \). 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 \( \text{diff}(t_1,t_2) \) 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 \( \text{diff}(t_1,t_2) = \{-1, 0, 2, 5, 6\} \). For every \( j \in \text{diff}(t_1,t_2) \), we define a test vector \( t_{i,2} \) and include it in \( C(t_1,t_2) \). The vector \( t_{i,2} \) is defined as follows: (1) For \( k \leq j \), \( t_{i,2}(k) = t_2(k) \). (2) For \( k > j \), \( t_{i,2}(k) = t_1(k) \).
In the example above, we obtain \( t_{1,2}^{1} = 00001111 = t_{1} \), \( t_{1,2}^{0} = 10011111, t_{1,2}^{1} = 10101111, t_{1,2}^{1} = 10101001 \) and \( t_{1,2}^{2} = 10101001 = t_{2} \). Starting from \( t_{1} \), \( t_{1} \) is modified into \( t_{2} \) bit-by-bit until \( t_{1} \) 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 \( C(t_{1}, t_{2}) = \{ t_{1,2}^{0}, t_{1,2}^{1}, \ldots, t_{1,2}^{m-1} \} \). We have that \( t_{1,2}^{i} = t_{1} \), \( t_{1,2}^{i} = t_{2} \), and \( t_{1,2}^{i} \) differs from \( t_{1}^{i} \) by one bit, for \( 1 \leq k < m \). 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 \( t_{1,2}^{j} \), 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 \( t_{1,2}^{j} \), 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 \( N_{p} m+1 \), where \( N_{p} \) 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 \( G = G \). Simulate \( G \) under \( T_{0} \) and remove detected faults from \( G \). Set \( P = \phi \). Set \( i = 0 \).

(3) For every pair of test vectors \( t_{1}, t_{2} \in T_{0} \), consider the test vector pairs in a random order:

   (a) Find the test vector chain \( C(t_{1}, t_{2}) \).

   (b) Simulate \( G \) under \( C(t_{1}, t_{2}) \). For every detected fault \( g \in G \), set \( ind(g) = i \) and remove \( g \) from \( G \).

   (c) If any fault in \( G \) 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 \( G = G \). Simulate \( G \) under \( T_{0} \) and remove detected faults from \( G \).

(5) Let \( P = \{ P_{1}, P_{2}, \ldots, P_{m-1} \} \) where \( P_{i} \) is the \( i \)th 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 \( g \in G \) with \( ind(g) = i \), remove \( p_{i} \) from \( P \). Otherwise, simulate \( G \) under the test vectors in \( C(t_{1}, t_{2}) \), remove detected faults from \( G \).

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

**[Test set \( T_{chains} \)]**

(8) Set \( G = G \). Simulate \( G \) under \( T_{0} \) and remove detected faults from \( G \). 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 \( t \in C(t_{1}, t_{2}) \) add \( t \) to \( T_{chains} \):

     - Simulate \( G \) under \( t \). Remove detected faults from \( G \). 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 \( t_{1,2} \) is redundant if the faults detected by \( C(t_{1}, t_{2}) \) 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 \( BR \). \( BR \) 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 \( BR \) we will measure the ability of a test set to detect untargeted faults, and indirectly, defects. We simulated \( BR \) 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 \( BR \).

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 \( BR \).

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 \( BR \).

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 \( BR \) is very close to that of \( T_{ndet} \), and sometimes higher. It is significantly higher than that of \( T_0 \). This is important since \( BR \) 100 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.

<table>
<thead>
<tr>
<th>Circuit</th>
<th>Initial</th>
<th>Minimum</th>
<th>BR100</th>
<th>Pairs</th>
<th>Chain</th>
<th>Minimum</th>
<th>BR100</th>
<th>N-Detect</th>
<th>Minimum</th>
<th>BR100</th>
</tr>
</thead>
<tbody>
<tr>
<td>s298</td>
<td>24</td>
<td>1</td>
<td>84.84</td>
<td>62</td>
<td>297</td>
<td>10</td>
<td>92.75</td>
<td>12</td>
<td>92.68</td>
<td></td>
</tr>
<tr>
<td>s344</td>
<td>15</td>
<td>1</td>
<td>78.05</td>
<td>39</td>
<td>217</td>
<td>10</td>
<td>90.62</td>
<td>15</td>
<td>90.66</td>
<td></td>
</tr>
<tr>
<td>s400</td>
<td>24</td>
<td>1</td>
<td>84.53</td>
<td>43</td>
<td>327</td>
<td>10</td>
<td>92.70</td>
<td>13</td>
<td>93.25</td>
<td></td>
</tr>
<tr>
<td>s420</td>
<td>43</td>
<td>1</td>
<td>72.68</td>
<td>101</td>
<td>516</td>
<td>2</td>
<td>80.39</td>
<td>11</td>
<td>82.66</td>
<td></td>
</tr>
<tr>
<td>s510</td>
<td>54</td>
<td>1</td>
<td>72.47</td>
<td>154</td>
<td>617</td>
<td>5</td>
<td>77.79</td>
<td>11</td>
<td>79.06</td>
<td></td>
</tr>
<tr>
<td>s526</td>
<td>50</td>
<td>1</td>
<td>86.10</td>
<td>102</td>
<td>630</td>
<td>10</td>
<td>92.73</td>
<td>12</td>
<td>93.02</td>
<td></td>
</tr>
<tr>
<td>s641</td>
<td>22</td>
<td>1</td>
<td>85.84</td>
<td>46</td>
<td>405</td>
<td>10</td>
<td>90.39</td>
<td>13</td>
<td>95.57</td>
<td></td>
</tr>
<tr>
<td>s820</td>
<td>94</td>
<td>1</td>
<td>69.57</td>
<td>245</td>
<td>1294</td>
<td>9</td>
<td>94.17</td>
<td>24</td>
<td>97.36</td>
<td></td>
</tr>
<tr>
<td>s953</td>
<td>76</td>
<td>1</td>
<td>76.72</td>
<td>91</td>
<td>860</td>
<td>10</td>
<td>80.39</td>
<td>11</td>
<td>81.04</td>
<td></td>
</tr>
<tr>
<td>s1196</td>
<td>138</td>
<td>1</td>
<td>78.88</td>
<td>199</td>
<td>1616</td>
<td>10</td>
<td>84.55</td>
<td>13</td>
<td>83.86</td>
<td></td>
</tr>
<tr>
<td>s1423</td>
<td>26</td>
<td>1</td>
<td>83.15</td>
<td>48</td>
<td>810</td>
<td>10</td>
<td>93.45</td>
<td>30</td>
<td>95.57</td>
<td></td>
</tr>
<tr>
<td>s5378</td>
<td>100</td>
<td>1</td>
<td>89.71</td>
<td>94</td>
<td>2391</td>
<td>10</td>
<td>94.17</td>
<td>24</td>
<td>97.36</td>
<td></td>
</tr>
<tr>
<td>s9234</td>
<td>111</td>
<td>1</td>
<td>81.16</td>
<td>144</td>
<td>3645</td>
<td>10</td>
<td>87.77</td>
<td>32</td>
<td>92.13</td>
<td></td>
</tr>
<tr>
<td>s58580</td>
<td>97</td>
<td>1</td>
<td>86.38</td>
<td>115</td>
<td>4416</td>
<td>10</td>
<td>93.30</td>
<td>45</td>
<td>97.60</td>
<td></td>
</tr>
<tr>
<td>Average</td>
<td>60</td>
<td>1</td>
<td>80.17</td>
<td>103</td>
<td>1224</td>
<td>9</td>
<td>87.48</td>
<td>16</td>
<td>88.82</td>
<td></td>
</tr>
</tbody>
</table>
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 BR 100.

**Table 2**

<table>
<thead>
<tr>
<th>circuit</th>
<th>$c$</th>
<th>test vectors</th>
<th>$T_0$ br100</th>
<th>faults in $G$</th>
<th>$T_0$ nmin</th>
<th>$T_0$ br100 nmin</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s_{208}$</td>
<td>10</td>
<td>27</td>
<td>72.26</td>
<td>72.88</td>
<td>90</td>
<td>112</td>
</tr>
<tr>
<td>$s_{344}$</td>
<td>10</td>
<td>24</td>
<td>84.14</td>
<td>84.48</td>
<td>30</td>
<td>112</td>
</tr>
<tr>
<td>$s_{400}$</td>
<td>10</td>
<td>24</td>
<td>84.41</td>
<td>84.53</td>
<td>34</td>
<td>145</td>
</tr>
<tr>
<td>$s_{420}$</td>
<td>10</td>
<td>43</td>
<td>72.60</td>
<td>72.68</td>
<td>84</td>
<td>261</td>
</tr>
<tr>
<td>$s_{510}$</td>
<td>10</td>
<td>54</td>
<td>71.93</td>
<td>72.47</td>
<td>86</td>
<td>221</td>
</tr>
<tr>
<td>$s_{526}$</td>
<td>10</td>
<td>50</td>
<td>87.18</td>
<td>86.10</td>
<td>51</td>
<td>210</td>
</tr>
<tr>
<td>$s_{641}$</td>
<td>8</td>
<td>22</td>
<td>85.78</td>
<td>85.84</td>
<td>53</td>
<td>220</td>
</tr>
<tr>
<td>$s_{820}$</td>
<td>10</td>
<td>94</td>
<td>69.12</td>
<td>69.57</td>
<td>93</td>
<td>358</td>
</tr>
<tr>
<td>$s_{953}$</td>
<td>10</td>
<td>76</td>
<td>76.60</td>
<td>76.72</td>
<td>94</td>
<td>314</td>
</tr>
<tr>
<td>$s_{1196}$</td>
<td>10</td>
<td>138</td>
<td>79.09</td>
<td>78.88</td>
<td>149</td>
<td>529</td>
</tr>
<tr>
<td>$s_{1423}$</td>
<td>2</td>
<td>26</td>
<td>83.07</td>
<td>83.15</td>
<td>48</td>
<td>255</td>
</tr>
<tr>
<td>$s_{1458}$</td>
<td>10</td>
<td>97</td>
<td>86.51</td>
<td>86.38</td>
<td>234</td>
<td>1328</td>
</tr>
<tr>
<td>$s_{15850}$</td>
<td>8</td>
<td>60</td>
<td>80.07</td>
<td>80.17</td>
<td>94</td>
<td>419</td>
</tr>
</tbody>
</table>

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 BR 100.

Under column n-detect we show information about the n-detection test set $T_{n-detect}$. We show the minimum number of detections of a stuck-at fault detected by $T_{n-detect}$, and the fault coverage of $T_{n-detect}$ with respect to BR 100.

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 $T_{n-detect}$ 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 BR 100 is very close to that of $T_{n-detect}$, and sometimes higher. It is always significantly higher than that of $T_0$ (BR 100 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


