

Test Set Identification for Improved Delay Defect Coverage in the Presence of Statistical Delays

Pavan Kumar Javvaji, Basim Shanyour, Spyros Tragoudas
Department of Electrical & Computer Engineering
Southern Illinois University Carbondale, USA.
pavan.javvaji@siu.edu, basim.shanyour@siu.edu, spyros@siu.edu

Abstract

Due to statistical gate delays, the critical probability of a testable path varies among its test patterns. Thus, the delay defect coverage of a selected set of critical paths depends on the selected test set. A new framework to select a test set for improved delay defect coverage is presented. It uses an algorithm that computes the critical probability of a path by a test pattern and machine learning to identify a small test set that maximizes the combined delay defect coverage. Experimental results show a significant improvement in delay defect coverage over existing static critical path approach.

Keywords

Delay Defects, Critical Paths, Delay defect coverage

1. Introduction

With advances in technology, modern circuits have higher complexities and reduced transistor sizing. The parameter variation-control is difficult and component delays vary from one manufactured chip to another. Therefore, the delays are not discrete values but are a statistical quantity. The inability of any such manufactured chip to execute intended design requirement is called as a defect. Defects affecting the timing behavior of the design are called delay defects. Path delay fault (PDF) model is commonly used for delay defects [14]. It takes small delay defects across the whole path into consideration and determines if it exceeds the intended clock period. Since it can take accumulative delays spread across the paths, it has drawn much attention in modern technology nodes.

Though this is useful in capturing small delay defects, there are exponential number of paths in a circuit and testing all of them is impractical. In literature, several non-enumerative test generation techniques have been proposed for robust and non-robust path delay faults [8]. The number of paths in the circuit limits these algorithms and ATPG for delay defects across all these paths is not scalable. Critical paths are the paths of the circuit whose delay exceeds the reference test clock (T_{clk}).

Existing critical path selection approaches can be classified into statistical static critical path (SCP)-based, as in [1], [3], [10], [15], [17], [18], and delay defect coverage (DC)-based, as in [11], [13], [16]. The SCP-based approaches attempt to quickly select critical paths without considering a test set. DC-based approaches select critical paths in presence of a test set. Delay defect coverage (DC) of the path by a pattern is defined as the number of circuit instances detected by the pattern where the path delay is maximum and exceeds

T_{clk} . The DC of the pattern is defined as the number of instances in which the pattern sensitizes a path whose delay exceeds T_{clk} .

In this paper, we present a scalable DC-based technique that identifies a test set for a given test cost to achieve maximum joint delay defect coverage of the circuit. The goal is to identify a test set T of a predetermined cardinality K for which the delay defect coverage (DC) is maximized. First, we outline recent SCP-based approaches.

The techniques in [18], [3], select a set of paths based on branch and bound technique. In [18] paths are selected to represent the timing behavior of the circuit by a max operator. They perform an exhaustive search over all the paths. The approach in [3] only considers testable paths and selects critical paths by statistical max operator to cover the process parameter space. Both approaches suffer from the error in the max operation [12].

The approach in [1] considers spot delay defect with delay thresholds at each gate. Subsequently, they select paths at each gate with non-zero criticality w.r.t given gate threshold such that their joint probability is maximum. Techniques in [15], [10] consider delays due to process variations and determine k paths at each gate for testing. Since these approaches consider static paths, the derived criticality or path probability may not be sensitized by a test set completely. Consequently, defective circuit instances are not detected as expected. Authors in [17] determine the criticality of a path based on its probability to exceed a reference clock in each Monte Carlo (MC) instance without considering a test set. It is not scalable and the computed static criticality may not be sensitized by a test pattern.

The following review DC-based approaches that consider delay defect coverage by a test set. The authors in [13] observe that the estimated DC by considering the critical paths is not the same as the DC by a robust test set of these paths for an IC. These results are drawn by MC analysis and wide discrepancies are reported. Authors in [11] consider a special problem of small delay faults at a gate that are expected to show at a given rate. Then they try to generate a test set based on confidence based metric. Large number of MC instances must be evaluated to have higher confidence making the approaches non-scalable.

In [16] it was observed that when a statistical maximum operator is applied on critical paths the resulting distribution is skewed normal. Subsequently, a statistical maximum operator on skewed normal distributions is proposed to derive the efficiency of a test set T . The reported test efficiency is sometimes overestimate which is not an effective indicator. The recent work in [6] determines the accurate lower bound

This research has been supported in part by grant NSF IIP 1361847 from the NSF I/UCRC for Embedded Systems at SIUC. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

of the sensitization probability of a test t for a critical path p without using MC analysis. It is shown that this is very close to the actual sensitization probability by MC analysis. Path correlations are taken into consideration and the sensitized delays of the paths by test t are disjoint. It is shown that the sensitized delay distribution of path p by the test t is not guaranteed to be normal unless it is a strong robust path.

The work in this paper considers a different problem than the DC-based approaches [6], [11], [13], [16]. To our knowledge, given a test pattern t , any existing work capable of determining precisely the sensitized path delay distributions or the DC of a path by t relies on Monte Carlo. We have implemented such an approach and we found out that with increasing processing variations it is not scalable for large circuits where a pattern t sensitizes many paths. However, we also implemented the recent approach in [6] and we confirmed that it guarantees very accurate lower bound estimates of the path sensitization in a scalable manner.

The presented work is based on [6] instead of [16], due to the confirmed accuracy by that method. As in [6], gate delays are represented by probability mass functions (pmfs). Therefore, delays due to process variations are not restricted to normal distributions. For each pattern t , sensitized delay distribution of each critical path are disjoint and a pmf for the total DC of test pattern t is generated. This observation led to an initial approach for generating a test set T of cardinality K where each test t has a high lower bound of DC.

The sensitized delay pmfs corresponding to various patterns may be correlated. Therefore, the calculation of the joint DC poses challenges. We use novel machine learning techniques to identify a test set of cardinality K that enhances the lower bound on the joint DC. Monte Carlo analysis shows that the approach in Section 4 outperforms the initial approach in Section 3.

The rest of the paper is organized as follows. Section 2 presents the preliminaries. Section 3 describes an initial approach. The proposed method is described in Section 4. Section 5 provides the experimental evidence and Section 6 concludes.

2. Preliminaries

Static delay distribution of a path differs from the sensitized delay distribution when a test pattern is applied [13]. Authors in [6] present a scalable technique that determines the path sensitization probability (PSP) along all the sensitized paths by a given test pattern. They calculate the sensitized delay pmf ($d_{p,t}$) of the path p under a pattern t with a topological circuit traversal. Various pmf operations at a gate are described in detail in [6] so that the $d_{p,t}$ considers path correlations due to reconvergences. Fig. 1 illustrates how to generate a $d_{p,t}$ in presence of reconvergences as in [6] and shows respective DC.

The circuit in Fig. 1 consists of four gates A, C, D and E where d_A and d_C are the delay pmfs of gates A and C respectively and gates D and E have a unit delay. All the primary inputs are considered to have zero delay. The sensitized critical path A-B-C-E is marked in red. At each gate, the approach PSP propagates only the portion of on-input delays that cannot be masked. At gate A, the $0 \rightarrow 1$

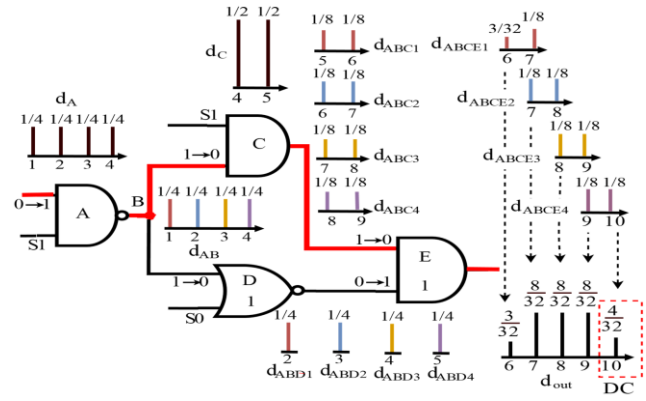


Figure 1. PSP operation in presence of reconvergent section transition propagates to its output B without any masking. This transition at B has a delay d_{AB} which is addition of input delay of A and the gate delay d_A . Since the input delay of A is zero, d_{AB} is equal to d_A .

The delays of inputs at the merging gate E are correlated. Hence, each sample of pmf d_{AB} at stem B are propagated individually through the reconvergent section to avoid correlation. As shown in Fig.1, four groups of delays d_{ABC_i} and d_{ABD_i} are formed at the output of gates C and D respectively, $1 \leq i \leq 4$. These groups are not merged together but are processed pair wise at the merging gate E. Hence, at output of E each pairwise operation between d_{ABC_i} and d_{ABD_i} yields a respective output delay pmf d_{ABCE_i} , $1 \leq i \leq 4$. Since d_{ABCE_i} is at the output of reconvergent section, they are grouped to obtain the sensitized output delay d_{out} of the path.

Since the maximum delay (T_{max}) of the circuit in Fig.1 is 10 units, let T_{clk} be 90% of T_{max} , then the instances exceeding 9 units of delay are critical. Such instances represent the DC of the test pattern which is highlighted in the d_{out} in Fig.1. The DC of the test pattern along various sensitized critical paths are disjoint. The joint probability of disjoint pmfs is additive [5], by which total DC of a pattern is determined. The proposed work considers the DC by a collection of test patterns T . We select T using machine learning to provide maximum DC.

Machine learning (ML) helps to identify the redundancies in the DCs when considering a set of test patterns. Principle component analysis (PCA) is an unsupervised machine learning technique that is used for dimensionality reduction to remove redundant data [7]. Singular value decomposition (SVD) is one of the most popular algorithms used with PCA whose objective is to remove data redundancies by finding a new set of orthogonal principle components (PCs) [2], [4].

In our case, SVD is used to determine the minimum number of PCs, i.e. test patterns, that can represent the DC of a large set of test patterns. The input for SVD is the DC matrix $D = [DC_1, DC_2, \dots, DC_N]$ of size $W \times N$. Here N is number of DCs which is equal to the size of test set and each DC consists of W delay instances, $DC_i = [d_1, d_2, \dots, d_w]$. SVD decomposes the input matrix D to three separate matrices U , S and V where U and V are real orthogonal matrices and S is a diagonal matrix that contains the singular values. The values of S represent the energy of the PCs in the decreasing order of their importance. To obtain the minimum number of

components for a given threshold E_{th} , we use the percentage of energy criterion as in [2]:

$$E_{th} = \frac{\sum_{i=1}^{i=R} s^2}{\sum_{i=1}^{i=N} s^2} \times 100 \quad (1)$$

Parallely, QR Decomposition with column pivoting (QRcp) is used to select and rank DCs according to their contribution as in [2]. It decomposes the input matrix to three separate matrices Q, R and Π where Q is a unitary matrix, R is an upper triangular matrix and Π is the permutation matrix. Each column in Π contains only one element with value "1". The position of this element in Π determines the importance of the corresponding DC in D. For illustration, suppose that D consists of DC_1, DC_2, DC_3 and R is equal to 2, then the DCs are selected as follows:

$$D = \begin{bmatrix} DC_{11} & DC_{21} & DC_{31} \\ DC_{12} & DC_{22} & DC_{32} \\ DC_{13} & DC_{23} & DC_{33} \end{bmatrix}, \Pi = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

$$D^{\wedge} = D\Pi = \begin{bmatrix} DC_{31} & DC_{11} & DC_{21} \\ DC_{32} & DC_{12} & DC_{22} \\ DC_{33} & DC_{13} & DC_{23} \end{bmatrix}; D^R = [DC_3, DC_1].$$

3. Initial Approach: 1 Test per Path (1TP)

In this section, we present a technique to determine a test set with high DC. We modify the fast and accurate path sensitization technique in [6] to determine the DC of the test pattern. The path sensitization probability of path p under a test t is denoted as $d_{p,t}$. The $DC_{p,t}$ is defined as the delay defect coverage of path p w.r.t the test pattern t . $DC_{p,t}$ is the critical portion of $d_{p,t}$ that exceeds T_{clk} . All the reported delay defect coverages in this paper are lower bounds.

The approach considers the set of testable critical paths (\mathbb{P}) with an objective to maximize the DC of the IC. The DC of a test pattern t is expressed as

$$DC_t = \bigcup_{\forall p \in P_t} DC_{p,t} \quad (2)$$

where P_t is the set of critical paths sensitized by the test pattern t . Initially, we identify the best test pattern for each critical path p . The best pattern t_{best} belongs to the test function f_p of path p and has maximum DC. A test function f_p consists of all the patterns that can sensitize the path p . Hence, the DC of the pattern t_{best} has the following property

$$DC_{t_{best}} = \max \{DC_t\} \forall t \in f_p \quad (3)$$

Initially, the $DC_{p,t}$ of the critical path p by a test pattern t in f_p is calculated. Since, this pattern can sensitize other critical paths in the circuit, the sensitization probability and DC of t along each path in its set of sensitized critical paths (P_t) is calculated. The joint probability of disjoint pmfs, i.e. their union, is the summation of all the pmfs [5]. Since, all such DCs are disjoint, their joint DC is determined by (2). Hence, the pattern that has the maximum DC in each test function of a critical path is selected. Finally, the best test pattern from each f_p are sorted in the order of their DCs and top K for the required cardinality are selected to form the best test set (T_{best}).

The number of critical paths sensitized with non-zero DC is scalable in nature and their delay distributions are non-normal. Hence, all critically sensitized paths are considered to find accurate DC.

Algorithm 1: 1TP

```

1: Inputs:  $\mathbb{P}, K, N$ 
2: Output:  $T_{best}$ 
3: for each  $p \in \mathbb{P}$  do
4:   while  $r < N$  do
5:      $t \leftarrow \text{TestGen}(f_p)$ 
6:      $DC_t \leftarrow \text{GenerateDC}(t)$ 
7:     if ( $|DC_t| > |DC_{best}|$ ) then
8:        $t_{best} \leftarrow t$ 
9:        $|DC_{best}| \leftarrow |DC_t|$ 
10:    end if
11:     $r \leftarrow r + 1$ 
12:  end while
13:   $T_{temp} \leftarrow T_{temp} \cup t_{best}$ 
14: end for
15:  $T_{temp} \leftarrow \text{Sort}(T_{temp})$ 
16: for  $i < K$  do
17:    $T_{best} \leftarrow T_{temp}[i]$ 
18:    $i \leftarrow i + 1$ 
19: end for
20: return  $T_{best}$ 

```

1TP is described in detail in Algorithm 1. The input to the approach is the set of testable critical paths (\mathbb{P}) through a gate, the test budget (K) and the number of test patterns per path (N). The output of the approach is the set of best K test patterns (T_{best}). For each path p in the critical set \mathbb{P} , a test pattern t from the test function f_p of the target path is picked using the function TestGen as in line 5. TestGen is an ATPG which generates test patterns for target path to test delay faults along the path as in [9] that considers non-robust tests.

Subsequently, in line 6, procedure GenerateDC is invoked that returns the total delay defect coverage (DC_t) of the generated test pattern t . It is observed that the DC of a pattern t in f_p is dominant along the respective critical path p . Hence, the heuristic of selecting one test pattern per path is justified as the best pattern covers the maximum defects along path p .

Next in line 7, if the absolute value $|DC_t|$ is greater than existing best delay defect coverage value $|DC_{best}|$ by a previous test pattern, then current test pattern t is considered as the best pattern (t_{best}). The DC of this pattern is assigned as the best DC as shown in line 9. The approach is continued until the N patterns from f_p are evaluated to identify its best test pattern. The iterator r is incremented in line 11 until N.

Once t_{best} for a path p is identified, it is stored in the set of patterns (T_{temp}) as in line 13. This is repeated until the best pattern for each path in \mathbb{P} is identified. Subsequently, the patterns in T_{temp} are sorted in the decreasing order of their DCs as shown in line 15.

Finally, we pick the first K test patterns from the sorted T_{temp} in the loop 16-19 and determine T_{best} . The test set T_{best} aims to provide high DC.

Procedure 1 describes procedure GenerateDC in detail. The input to this procedure is the test pattern t . The output of the procedure is the total delay defect coverage (DC_t) of pattern t . Initially, in line 3, set of sensitized critical paths by pattern t are determined and stored in P_t . Then for each critical path p in P_t , we determine the respective sensitized path delay ($d_{p,t}$) using PSP as shown in line 5. In line 6, the instances of

Procedure 1: GenerateDC

```
1: Input:  $t$ 
2: Output:  $DC_t$ 
3:  $P_t \leftarrow \text{SensitizedPaths}(t)$ 
4: for each  $p \in P_t$  do
5:    $d_{p,t} \leftarrow \text{PSP}(p,t)$ 
6:    $DC_{p,t} \leftarrow \text{Critical}(d_{p,t})$ 
7:    $DC_t \leftarrow DC_t \cup DC_{p,t}$ 
8: end for
9: return  $DC_t$ 
```

$d_{p,t}$ that exceed the T_{clk} are identified which is the delay defect coverage ($DC_{p,t}$) of path p by pattern t . In line 7, it is added to the DC_t using a union operation. The process is repeated until all the paths in P_t are evaluated. Finally, the DC_t of the test pattern is returned.

4. Proposed Approach: K Tests for critical set (KT)

Though there is significant improvement in the DC by the initial approach, its DC is not optimal because it does not consider correlation between DCs of various patterns. In this section, we describe the proposed approach to achieve maximum joint DC using machine learning.

Let there be \mathbb{P} sensitizable critical paths in the circuit. Like initial approach, we determine the DCs of N patterns from the test function f_p of each path p in \mathbb{P} . However, instead of selecting just the pattern with highest DC_t per path, the DCs of all the N test patterns for all paths in \mathbb{P} are stored in the set D and is fed to ML engine to identify the optimum subset of DCs.

KT is described in Algorithm 2. The inputs to the algorithm are the set of sensitizable critical paths \mathbb{P} , the test budget K and the number of test patterns per path N . The output of the procedure is the best test (T_{best}) of size K . The outer loop works on each critical path p in \mathbb{P} to find the DC_t of N test patterns in f_p . A test pattern t is selected from the f_p of p using TestGen as shown in line 5. Subsequently, the DC_t of the generated test is determined by using the procedure GenerateDC in line 6. In line 7, the generated DC_t is included in the set D by performing union operation. This is continued until N such DCs are generated for the path. This is repeated until all the paths in \mathbb{P} are processed.

The DCs in D are correlated because different patterns can sensitize same circuit instances. Hence, redundancies

Algorithm 2: KT

```
1: Inputs:  $\mathbb{P}$ ,  $K$ ,  $N$ 
2: Output:  $T_{\text{best}}$ 
3: for each  $p \in \mathbb{P}$  do
4:   while  $r < N$  do
5:      $t \leftarrow \text{TestGen}(f_p)$ 
6:      $DC_t \leftarrow \text{GenerateDC}(t)$ 
7:      $D \leftarrow D \cup DC_t$ 
8:      $r \leftarrow r + 1$ 
9:   end while
10: end for
11:  $D^R \leftarrow \text{ML}(D)$ 
12:  $T_{\text{best}} \leftarrow \text{ExtractTP}(D^R, K)$ 
13: return  $T_{\text{best}}$ 
```

Procedure 2: ML

```
1: Inputs:  $D$ 
2: Output:  $D^R$ 
3:  $S \leftarrow \text{SVD}(D)$ 
4: while  $E < E_{\text{th}}$  do
5:    $E \leftarrow \frac{\sum_{j=1}^{j=R} s^2}{\sum_{j=1}^{j=N} s^2}$ 
6:    $R \leftarrow R + 1$ 
7: end while
8:  $\Pi \leftarrow \text{QRcp}(D)$ 
9:  $D^{\wedge} \leftarrow T\Pi$ 
10:  $D^R \leftarrow D^{\wedge}[:,R]$ 
11: return  $D^R$ 
```

between the DCs in D must be eliminated which is achieved by using ML as explained in Section 2. In line 11, the set D is fed to the procedure ML. It returns a subset of DCs denoted as D^R that can represent the set D . The DCs in D^R are arranged in the decreasing order of their contribution in D . The elements in D^R only contain the DC_t information from which the respective test patterns need to be extracted. Hence, in line 12, first K test patterns associated with the DCs in D^R are selected using the function ExtractTP. This returns the required test set T_{best} of cardinality K . Experimental results show significant improvement in DC over initial approach with minor time overhead for large benchmarks.

ML is described in procedure 2. The input to the procedure is a set D where each column represents a DC_t and each row corresponds to the delay instance of the DC_t . The output is the subset D^R . In line 3, ML engine starts by running SVD algorithm for D to find the number of principal components (PCs) that represent the set D . It returns matrix S . Based on S , from lines 4-7, ML engine determines the required number of DCs that represent the initial set D . The loop in lines 4-7 continues until the combined energy (E) of the selected components exceeds a given energy threshold E_{th} , i.e. 99.9%. In line 8, we execute the QRcp algorithm on D that returns the permutation matrix Π as explained in Section 2. Subsequently, a ranked matrix D^{\wedge} of all DCs is calculated by multiplying Π with D as shown in line 9. Finally, we select the first R columns from the matrix D^{\wedge} to produce the required ranked matrix D^R .

5. Experimental Results

The focus of this section is to derive the quality of the delay defect coverage (DC) by the test set obtained from the initial approach ITP as described in Section 3 and subsequently the proposed approach KT as given in Section 4. The DCs of the two approaches are compared to a recent method in literature [15] which selects critical paths statically. It is referred to as static critical path (SCP) method which is traditional path selection technique.

Traditionally test cost is limited. Therefore, one pattern per critical path is selected. Hence, in this experimental evaluation we assume that number of patterns applied (test cost) is equal to the number of critical paths. Hence, in case of SCP, a random pattern from the test function of each critical path is selected. TestGen considers non-robust test functions for each critical path in selecting test set of approach SCP. The set of testable statistical critical paths \mathbb{P} for ITP and KT are

TABLE 1: EXECUTION TIME VS DEFECT COVERAGE WHEN NUMBER OF TEST PATTERNS (K) = 5

Circuit	# gates	1TP		KT		DC _{max}	DC Improvement by KT over 1TP	Speedup by 1TP over KT
		Execution Time (in sec)	DC (# defects/30000 MC samples)	Execution Time (in sec)	DC (# defects/30000 MC samples)			
c5315	2307	1164.98	213	1487.45	284	365	19.45%	1.28x
c7552	3512	1378.14	233	1740.11	309	397	19.14%	1.26x
s35932	16065	5503.02	208	5647.99	325	483	24.22%	1.03x
s38417	22197	6090.74	225	6420.31	344	471	25.27%	1.05x
s38584	19253	6516.24	206	6882.31	321	459	25.05%	1.06x
b18	114621	9824.66	276	10206.73	378	521	19.58%	1.04x
b19	231320	10073.32	266	10412.19	386	509	23.58%	1.03x
Average							22.33%	1.11x

generated using [15].

The approaches SCP, 1TP and KT have been implemented in the C++ language and the ML engine is implemented in R language on a 1.6 GHz Sun Sparc workstation with 8 GB RAM. The delay distributions at each gate was obtained from a 45nm technology library with interconnection delays accumulated into the gate delays. Experimentation was performed on the largest ISCAS' 85, ISCAS'89 and ITC'99 benchmarks. We consider 30000 MC instances in the experiment and T_{clk} is set at 90% of the maximum delay obtained from static timing analysis.

Once the test set of given cardinality K is determined by either (SCP, 1TP, KT) we perform MC analysis as a common evaluation platform to derive and compare the DCs by the test set of each approach. Fig. 2 shows the DCs by the test set of the three approaches on circuit c5315 for various cardinality K. When K is 1, approach 1TP provides the pattern with highest DC among the test functions of all the critical paths. This is the same pattern returned by KT approach using ML since it returns the pattern with highest DC contribution. Therefore, both approaches have same DC. However, for K equal to 1, SCP is limited to selecting one critical path and a random pattern from its test function. It is observed that such pattern yields low DC as shown in Fig. 2.

As the value of K increases, we observe that the test set by the approach KT has higher DC compared to 1TP and reaches a max DC (DC_{max}) with a less number of patterns. We observe that the approach 1TP has significant improvement in the DC compared to SCP but is lower than KT until the DC_{max} is reached. Approach 1TP reaches DC_{max} when K is 130 whereas KT reaches it when K is 59. The approach SCP is the worst performer with lower DC at any value of K. It requires a large test set to reach the DC_{max} which is not scalable to perform MC analysis. Hence, KT is best suitable approach to achieve high

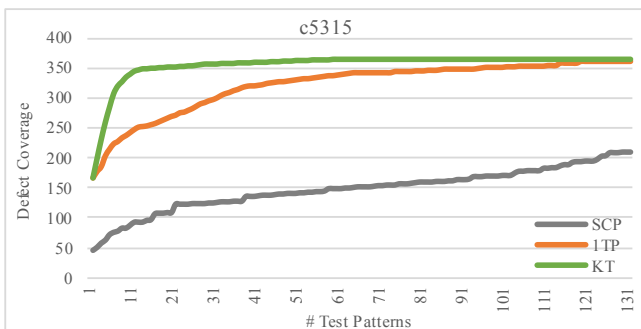


Figure 2. Defect coverage vs number of test patterns (K) by MC analysis.

DC with low test cost. Similar trends have been observed for various benchmarks of ISCAS'85, ISCAS'89 and ITC'99. Fig. 3 shows the plots of DC vs the number of test patterns applied in the MC analysis for various benchmarks with similar trend.

Table 1 summarizes the executions time and the DC by approaches 1TP and KT when number of test patterns is 5. The execution time includes the time taken to generate the test set. Column 1 shows the circuit names followed by the number of gates in the circuit in column 2. The circuits b18 and b19 are extremely large compared to other circuits. Column 3 shows the execution time by approach 1TP to generate its T_{best} of size 5 and column 4 shows its respective DC by MC analysis. Column 5 shows the execution time taken to generate the T_{best} by approach KT followed by its respective DC in column 6. The DC_{max} that can be obtained is shown in column 7.

Column 8 shows the improvement in DC by approach KT over 1TP. The speedup by approach 1TP over KT is shown in column 9. It is observed that approach KT shows a significant improvement in the DC with minor overhead in time. On average, approach KT provides 22.33% better DC compared to 1TP whereas 1TP is 1.11x faster than KT on average. Therefore, the two approaches have tradeoff between DC and execution time. It is observed that the speed of KT almost reaches the speed of 1TP for large circuit. For e.g. KT has 23.6% improvement in DC compared to mere 1.03x speedup due to 1TP in b19.

Table 2 shows the number of test patterns required to obtain the DC_{max} approaches 1TP and KT. It is compared to the DC obtained by traditional SCP by applying same number of patterns as in 1TP and KT respectively. Column 1 lists the names of the circuit. Column 2 shows the number of patterns applied to reach the DC_{max} by approach 1TP. Column 3 shows

TABLE 2: # TEST PATTERNS FOR MAX DEFECT COVERAGE BY 1TP AND KT VS DEFECT COVERAGE BY SCP

Circuit	# Test Patterns for DC _{max}		Comparison to existing approach (SCP)	
	1TP	KT	% of DC _{max} with # test patterns as in 1TP	% of DC _{max} with # test patterns as in KT
c5315	120	59	54.44%	41.09%
c7552	162	66	64.03%	39.80%
s35932	98	36	47.5%	27.74%
s38417	152	79	65.44%	44.68%
s38584	191	94	47.15%	37.48%
b18	198	82	58.11%	34.55%
b19	230	73	57.40%	38.51%
Average			56.3%	37.7%

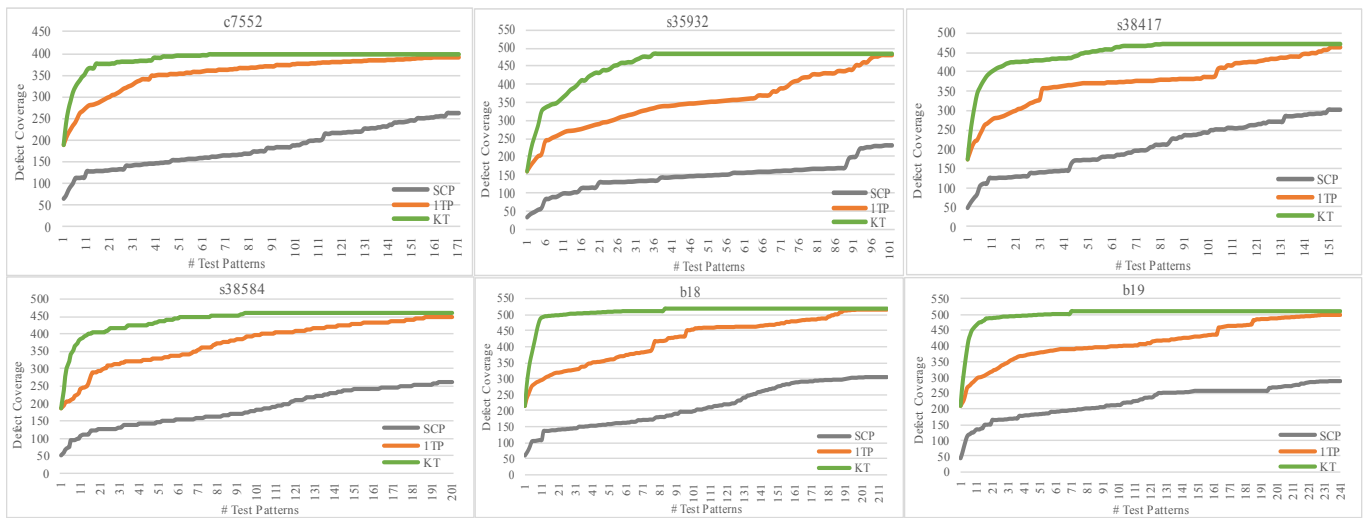


Figure 3. Defect coverage vs number of test patterns (K) by MC analysis for various benchmarks.

number of patterns applied to reach the DC_{max} by approach KT. Column 4 shows the percentage of DC_{max} by SCP method when the number of test patterns applied is same as that in approach ITP. The percentage of DC_{max} by SCP when number of test patterns applied is same as that of approach KT is given in column 5. It is observed from columns 2 and 3 that KT requires significantly less number of patterns than ITP to obtain DC_{max} .

From column 4, we observe that the traditional SCP approach achieves on average only 56.3% of the DC by the ITP approach. Furthermore, column 5 shows that SCP can achieve only 37.7% of the DC of the KT approach on an average. This shows that both the approaches have significant improvement in delay defect coverage over traditional SCP method.

6. Conclusion

Efficient test set identification method to achieve high delay defect coverage for a given test set size have been presented. There is a tradeoff in obtained delay defect coverage versus execution time. Experimental results showed that both approaches are scalable and outperform the state-of-the-art approach.

7. References

- [1] P. Alladi, S. Tragoudas, "Efficient selection of critical paths for delay defects in the presence of process variations," in *Proc. IEEE DTIS*, pp. 1-6, April 2016.
- [2] S. Chakroborty and G. Saha, "Feature selection using singular value decomposition and QR factorization with column pivoting for text independent speaker identification," *Elsevier Speech Communication*, vol. 25, no. 9, pp. 693–709, 2010.
- [3] J. Chung, J. Xiong, V. Zolotov and J. A. Abraham, "Testability Driven Statistical Path Selection," in *IEEE Trans. Comput.- Aided Des. Integr. Circuits Syst.*, pp. 1275–1287, August 2012.
- [4] F. Firouzi, F. Ye, K. Chakrabarty, M. B. Tahoori, "Aging- and Variation-Aware Delay Monitoring Using Representative Critical Path Selection," *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, vol. 20, no. 3, pp. 1-23, 2015.
- [5] C. M. Grinstead and J. L. Snell, "Introduction to probability," *American Mathematical Society*, RI, 1997.
- [6] P. K. Javvaji and S. Tragoudas, "Efficient Computation of the Sensitization Probability of a Critical Path

Considering Process Variations and Path Correlation," in *Proc. of ISCAS*, May 28-31, 2017.

- [7] I. T. Jolliffe, "Principle Component Analysis," *Springer-Verlag, New York*, pp. 1-10, 2002,.
- [8] A. Krstic and K. -T. Cheng, "Delay fault testing for VLSI circuits," *Kulwer Academic Publishers*, 1998.
- [9] I. Pomeranz and S. M. Reddy, "Size of test sets for path delay faults using strong and weak non-robust tests," *IEE Computers & Digital techniques*, vol. 5, no. 5, pp. 405-414, 2011.
- [10] W. Qiu and D. M. H. Walker "An Efficient Algorithm for Finding the K Longest Testable Paths Through Each Gate in a Combinational Circuit," in *Proc. of IEEE Int. Test Conf.*, pp. 592–601, 2003.
- [11] M. Sauer, I. Polian, M. Imhof, A. Mumtaz, E. Schneider, A. Czutro, H.-J. Wunderlich, and B. Becker, "Variation-aware deterministic ATPG," in *Proc. IEEE Eur. Test Symp.*, pp. 1–6, 2014.
- [12] D. Sinha, H. Zhou, and N. Shenoy, "Advances in computation of the maximum of a set of Gaussian random variables," *IEEE Trans. Comput.- Aided Des. Integr. Circuits Syst.*, vol. 26, no. 8, pp. 1522–1533, Aug. 2007.
- [13] M. Sivaraman and A. J. Strojwas, "Path delay fault diagnosis and coverage - a metric and an estimation technique," *IEEE Trans. Comput.- Aided Des. Integr. Circuits Syst.*, pp. 448–457, 2001.
- [14] G. L. Smith, "Model for delay faults based upon paths," in *Proc. Int. Test Conf.*, 1985, pp. 342–349.
- [15] A. M. Somashekar, S. Tragoudas, "Efficient Critical Path Selection under a Probabilistic Delay," in *Proc. ACM Great Lakes Symposium on VLSI*, pp. 185-190, May 2017.
- [16] M. Wagner, "Efficient Algorithms for Fundamental Statistical Timing Analysis Problems in Delay Test Applications of VLSI Circuits," *Doctoral Dissertation, Institut für Technische Informatik der Universität Stuttgart*, November 2016.
- [17] Li. -C Wang, J. -J Liou, and K. -T Cheng "Critical path selection for delay fault testing based upon a statistical timing model," in *IEEE Trans. Comput.- Aided Des. Integr. Circuits Syst.*, pp. 1550–1565, 2004.
- [18] V. Zolotov, J. Xiong, H. Fatemi, and C. Visweswariah, "Statistical Path Selection for At-Speed Test," in *IEEE Trans. Comput.- Aided Des. Integr. Circuits Syst.*, pp. 749–759, May 2010.