Non-Enumerative Path Delay Fault Diagnosis

Saravanan Padmanaban Spyros Tragoudas
Department of Electrical and Computer Engineering
Southern Illinois University
Carbondale, IL 62901

Abstract

The first non-enumerative framework for diagnosing path delay faults using zero suppressed binary decision diagrams is introduced. We show that fault free path delay faults with a validated non-robust test may together with fault free robustly tested faults be used to eliminate faults from the set of suspected faults. All operations are implemented by an implicit diagnosis tool based on the zero suppressed binary decision diagram. The proposed method is space and time non-enumerative as opposed to existing methods which are space and time enumerative. Experimental results on the ISCAS'85 benchmarks show that the proposed technique is on an average least three times more efficient to improve the diagnostic resolution than existing techniques.

1 Introduction

With the advent of deep-submicron technology, testing for the performance of an integrated circuit (IC) has become a difficult task. Even small process variations can cause a fault in the circuit. Therefore before mass production of the IC, a small number (first silicon) is produced to perform the various tests and check for the performance of the IC. The check is performed by applying test vectors and comparing the expected output to the sampled output. We assume a slow-fast test application methodology on the combinational component of the digital synchronous circuit and that the circuit under diagnosis is functionally correct. The process of locating the region in the chip that caused the delay fault is termed as *delay fault diagnosis*.

We consider the path delay fault (PDF) model for the purpose of diagnosis. The number of PDFs in a circuit can be exponential. This necessitates the need for a data-structure that can efficiently represent and manipulate the path delay faults. [8] introduced a methodology to represent path delay faults as zero-suppressed binary decision diagrams (ZBDD). The ZBDD is a canonical data structure and is a variant of the binary decision diagram and is very effective in representing and manipulating single PDF and multiple PDF. We use the ZBDD as the underlying data structure for the proposed diagnosis methodology.

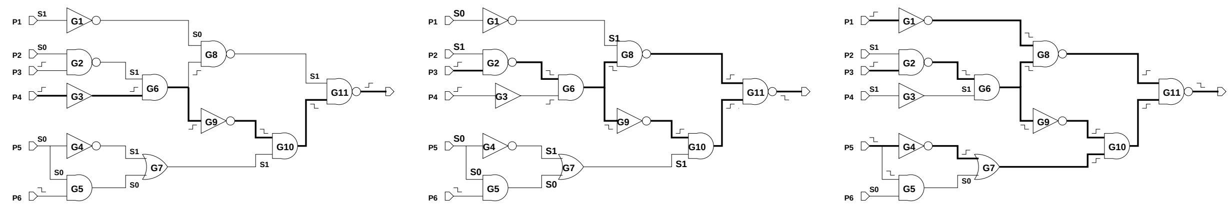
Different methods have been proposed for diagnosing GDFs and PDFs [9]. We propose an enhanced ZBDD based framework for path delay fault diagnosis based on effect-cause analysis. In effect-cause analysis methods, a set of input vector pairs are applied to the circuit under test, the sampled output of the applied test is compared against the expected logical output. This information is used to prune the set of possible faults, in an attempt to locate the fault. Each of the input vector pairs for which the expected and observed output is the same is termed as a *passing test*. Each of the input vector pairs for which the expected and observed output differ is termed as a *failing test*. The set of all passing tests (respectively failing tests) is termed as a *passing set*(respectively *failing set*). The set of all PDFs sensitized by the passing set and is guaranteed to be fault free is termed as a *fault free set*. The set of all PDFs sensitized by the failing set that could explain for the error observed is termed as a *suspect set*.

Some PDFs sensitized by a passing test vector may not be indeed fault free. This happens if the PDFs are sensitized non-robustly, in which case the transition may be masked during the actual test application process. The PDF diagnosis methodology proposed in [9], defines the term *fault free PDF* as a PDF that is sensitized by a given passing test and is guaranteed to be fault free. These are precisely the single and multiple PDFs that are robustly tested by a passing test as a fault free PDF. We show here that an additional class of PDFs, those with a validatable non-robust(*VNR*) test in the passing set can also be be classified as fault free PDFs.

A drawback of [9] is that it uses a graph (cyclic) based data structure to represent multiple path delay faults (MPDF). Each node in the graph is represents a SPDF (which can be exponential to the number of lines sensitized in the circuit). Thus the method is space enumerative to the number of single path delay faults (SPDF) since we have to explicitly store each SPDF as a node. However a MPDF is stored as a cycle in the graph. If a MPDF is to be removed from the set, it is done by removing the corresponding cycle from the graph. This removal process is enumerative, since each MPDF is removed one at a time. To overcome the usage of these complex data structures and graph operations, we introduce a ZBBD based method to non-enumeratively store and manipulate PDFs (SPDFs and MPDFs).

Several PDFs in the suspect set may be fault-free. Based on the PDFs in the fault free and suspect set, many such PDFs in the suspect are shown to be fault free, thus reducing the search space of the potential fault. We define the *resolution* of the diagnosis process (diagnostic resolution) to be the ratio of the cardinality of the suspect set after all applied reductions to the cardinality of the original suspect

1530-1591/03 $17.00 © 2003 IEEE



(a)Passed Test - Pattern I (b)Passed Test - Pattern II (c)Failed Test - Pattern III

Figure 1: Path Delay Fault Diagnosis - A Review

Passing Test - T_1 and T_2				**Failing Test - T_3**		
PDF ID	**PDF**	**Sensitization**	**Type**	**PDF ID**	**Suspect Set**	**Type**
PD_1	$\uparrow \{P4.G3.G6.G9.G10.G11\}$	Robust	SPDF	FD_1	$\uparrow \{P3.G2.G6.G9.G10.G11\}$	SPDF
PD_2	$\uparrow \{P3.G2.G6.G8.G11\}$	Non-Robust	SPDF	FD_2	$\downarrow \{P5.G4.G7.G10.G11\}$	SPDF
PD_3	$\uparrow \{P3.G2.G6.G9.G10.G11\}$	Non-Robust	SPDF	FD_3	$\{P1.G1.G8.G11,$ $P3.G2.G6.G8.G11\} \downarrow$	MPDF

Table 1: Sensitized Path Delay Faults

set. This is the reduction in the cardinality of the suspect set expressed as a ratio.

Section 2 illustrates improvement in resolution when using PDFs with VNR tests. It also summarizes the rules for improving the diagnostic resolution, which is a clear super set of those introduced in [9]. Section 3 shows the data structure and algorithms that non-enumeratively improve the diagnostic resolution. It also presents the first non-enumerative algorithm to find the exact set of PDFs with a VNR test for a given passing set. Section 4 presents our diagnosis methodology using the procedures and operators introduced in Sections 2 and 3. Experimental results and conclusion are provided in Sections 5 and 6 respectively.

2 Diagnosis using PDFs With a VNR Test

When a PDF P is shown to be fault free, then any PDF of higher cardinality which is a superset of P cannot have a delay fault [4]. The conventional effect-cause analysis based method prunes the suspect set by eliminating PDFs in the suspect set whose subfault is a robustly tested PDF in the fault free set. By the definition of a robustly tested path delay fault, some PDFs in the suspect set could not have caused the fault at the output. A class of PDFs termed as *Path Delay Faults with VNR test* has been defined in [10] and has been shown to have the quality of a robust fault.

Definition 1 (Validatable Non-Robust Test) *A set of two pattern tests S is termed as a validatable non-robust test for a path P if and only if no element of S is a robust test for P and if the circuit under test passes all tests in S, it can be concluded that the desired transition propagates along the path P in the time allowed. (Restated from [10])*

A non-robust test for a PDF becomes invalid if the signal arriving at any of the non-robust off-inputs has a delay fault. If the PDF through the non-robust off-inputs can be tested robustly and is fault free, the non-robust test for the target PDF, together with the robust test for the PDFs through the non-robust off-inputs, form a validatable non-robust (VNR) test for the target PDF. A VNR test is guaranteed to detect a delay fault on the target PDF independent of the delays in the rest of the circuit if the circuit passes all the tests.

For a given instance of a circuit shown in Figure 1, let the diagnostic test set T consist of three test vectors $T_1 = \{10001, 10100\}$, $T_2 = \{01001, 10100\}$ and $T_3 = \{01010, 11100\}$. Assume that the vectors T_1 and T_2 are passing tests and T_3 be a failing test. Figures 1a, 1b and 1c show the PDFs sensitized by the test vectors T_1, T_2 and T_3 respectively. Table 1 shows the type of each PDF (single or multiple) and its corresponding sensitization type[1]. Based on the definition of a PDF with VNR test, tests T_1 and T_2 form a VNR test for the PDF PD_3. (The non-robust off-input of $G6$ is the line on PD_1). The test guarantees that the PDF PD_3 is fault free. Our method prunes the suspect set using the set of PDFs tested by robust and VNR tests from the passing set. Namely, the suspect set $\{FD_1, FD_2, FD_3\}$ can be pruned using the set $\{PD_1, PD_3\}$ instead of just using the fault free PDF $\{PD_1\}$ that has been tested robustly by passing test T_1. The PDF FD_1 can be eliminated from the suspect set, pruning the suspect set to $\{FD_2, FD_3\}$. Without using the PDFs with a VNR test no pruning of the suspect set is possible.

The definition of a PDF with a VNR test is recursive and this causes computation difficulties. However Section 3.1 presents a method where PDFs with VNR tests of any cardinality can be identified non-enumeratively by only three traversals on the passing test set. We emphasize that VNR tests may sometimes be invalid for PDF testing. However they can be used for in diagnosis without any skepticism. For more detailed discussions see also [5].

The remaining of the section gives the set of rules for performing diagnosis. Let the test set T consists of X

[1]Table 1 represents the SPDFs as its constituent path together with the transition on its primary input. The MPDF is represented by its constituent subpaths together with the transition on its primary output

vectors - $\{T_1, T_2, \ldots, T_X\}$, with m passing tests and n failing tests. Let $P = \{P_1 \cup P_2 \ldots P_m\}$ be the set of fault free PDFs tested by each of the m passing tests. Let $F = \{F_1 \cup F_2 \ldots F_n\}$ be the set of PDFs tested by each of the n failing test vectors in T.

Once the fault free set and the suspect set are derived, the process of diagnostic resolution can be performed using the following rules:

1. Let Q_i be a SPDF in the fault free set P tested by a robust test or a VNR test. Any superset of Q_i in the suspect set F (such as MPDF - Q_iQ_j) can be eliminated. We call such MPDFs as *redundant PDFs*. This elimination is valid because, a MPDF can have a delay fault only if all subfaults of the MPDF have a delay fault.

2. Let Q_iQ_j be a MPDF in the fault free set P tested by a robust test or a VNR test. Any superset of Q_iQ_j in the suspect set F (such as MPDF - $Q_iQ_jQ_k$) can be eliminated from the suspect set.

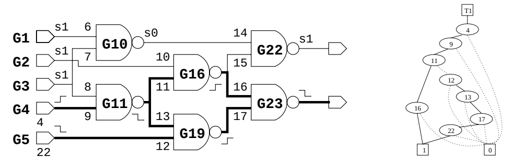
We note that [9] only uses a part of the rules shown, taking advantage of the fault free PDFs with robust tests alone. In addition to the above said rules, the set of fault free PDFs can be optimized for computational purposes. Let Q_i and Q_iQ_j be a SPDF and MPDF in the fault free set P, respectively. Then Q_iQ_j can be eliminated from the set P because, if the SPDF Q_i is fault free, then Q_iQ_j or any other MPDF of which Q_i is a subfault, is also guaranteed to be fault free. Although such PDF elimination does not improve the resolution, it is very important for computational purposes.

3 Data Structures and Operations

The process of extracting a fault free set is performed in two phases. The first phase involves the identification of the robustly tested PDFs and the second phase involves the identification of PDFs with VNR tests. For a given passing set, the set of tested PDFs needs to be extracted non-enumeratively and stored in a compact format. It has been shown in [8] that the problem of representing PDFs can be reduced to representing them as combinational sets. This task can be performed by just using standard ZBDD operators introduced in [7].

Extraction of the fault free set is performed by *three passes on the passing test set*. During the first pass, we extract all PDFs (single and multiple) that are robustly tested by the passing set. While performing the first pass, we collect all the information regarding fault free PDFs that are robustly tested by the passing set. The second pass identifies, the set of PDFs that are non-robustly tested by the passing set. During the third pass, we identify the set of PDFs with VNR which is the subset of the non-robustly tested PDFs (identified during the second pass) for which the off-inputs have a passing robust test.

The example below briefly illustrates the methodology to extract the set of robustly tested PDFs for a given test vector.



(a) Simulation of T (b) PDFs Sensitized

Figure 2: Extraction of the Sensitized PDFs

Let the test vector $T = \{11101, 11110\}$ be a passing test applied to the circuit shown in Figure 2a. Each line and gate of a circuit be assigned a unique variable and each of the primary input is assigned with two variables (one for rising and one for falling transitions). The presence of a variable v in a combinational element representing a PDF implies that $v = 1$ and the absence implies $v = 0$. The numbers marked on the lines are the variables used to denote that line. For the primary inputs, variables 1-5 are used to represent the rising transitions on each input and variables 18-22 are used to represent the falling transitions. The bold lines in the circuit in Figure 2a indicate the sensitized lines.

Procedure: Extract_RPDF

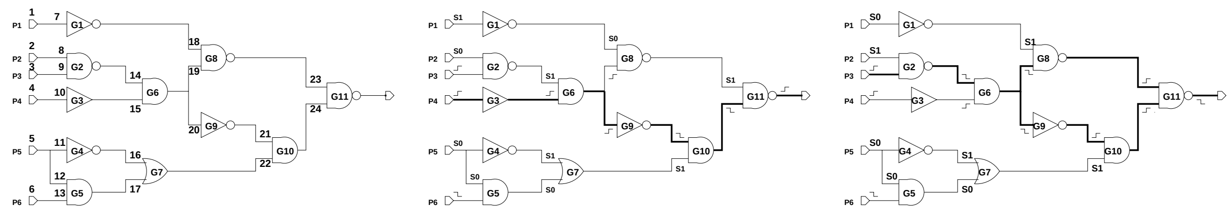
for each test t in test set T **do**
  for each gate g in topological order **do**
    if g is a primary input **then**
      assign g a variable depending on the transition
    else if any line l of g is robustly sensitized **then**
      store partial PDFs in set R_l
    else if g is robustly co-sensitized **then**
      store partial PDFs in set R_l
  for each primary output line l **do**
    $R_t \leftarrow R_t \cup R_l$
  Eliminate *redundant PDFs* from R_T and R_t
  $R_T \leftarrow R_T \cup R_t$

The PDFs tested robustly by a single passing test can be identified by a single topological traversal. At each gate, all possible partial PDFs from the primary input to that gate are stored as a set. In the circuit shown in Figure 2a, the partial PDFs from primary inputs to line 12 and 13 are $R_{12} = \{4.9.13\}$ and $R_{13} = \{22.12\}$ respectively. Gate G19 is co-sensitized, so a *product operation* is performed between the two partial products to represent the co-sensitization between them. At the end of the topological traversal each primary output contains the set of robustly tested PDFs from all the primary inputs to that particular primary output. The PDFs tested by the passing test t is R_t=$\{4.9.11.16, 4.9.12.13.17.22\}$, is represented as a ZBDD as shown in Figure 2b. The set of PDFs tested robustly by the passing set T, R_T is obtained by performing a union of the sets of PDFs tested robustly by each test t in the set T. This method is formally outlined by **Procedure Extract_RPDF**.

Procedure: Eliminate(P, Q)

Ensure: $Q \neq \emptyset$
Result $\leftarrow P - (P \cap (Q * (P \supset Q)))$
return (**Result**)

We now turn our attention to implement a procedure to eliminate redundant PDFs, which in fact can also be used to improve the diagnostic resolution. In [8] a new ZBDD operator, the *containment operator* ($\supset$) was introduced. The



(a) Circuit Under Diagnosis (b)Passed Test - Pattern I (c)Passed Test - Pattern II

Figure 3: Identification of PDFs with a VNR Test

R_T={↑4·10·15·20·21·24}
N_1 = ∅ ; N_2 = {↑3·9·14·19·23,↑3·9·14·20·21·24}

Line	R^l_T	N^l_2 Before VNR Check	N^l_2 After VNR Check	P^l_2	Line	R^l_T	N^l_2 Before VNR Check	N^l_2 After VNR Check	P^l_2
3	-	↑3·9·14·19·23, ↑3·9·14·20·21·24	↑3·9·14·19·23, ↑3·9·14·20·21·24	-	19	-	-	-	-
4	↑4·10·15·20·21·24	-	-	-	20	20·21·24	20·21·24	20·21·24	↑3·9·14
9	-	9·14·19·23, 9·14·20·21·24	9·14·19·23, 9·14·20·21·24	↑3	21	21·24	21·24	21·24	↑3·9·14·20
10	10·15·20·21·24	-	-	↑4	23	-	-	-	-
14	-	14·19·23, 14·20·21·24	14·20·21·24	↑3·9	24	24	24	24	↑3·9·14·20·21
15	15·20·21·24	-	-	↑4·10					

Table 2: Procedure to Extract PDFs tested by VNR tests

operator can be used together with other ZBDD operators to identify if a minterm is contained in another minterm.

Definition 2 (Containment) Containment *(P ⊃ Q), is the union of all the quotients of dividing P by the cubes of Q.*

Example: Let P = {abd, abe, abg, cde, ceg, egh} and Q = {ab, ce}. Then $(P \supset Q)$ is obtained as:

$$\begin{aligned}(P \supset Q) &= (P/\{\text{ab}\}) \cup (P/\{\text{ce}\}) \\ &= \{\text{d, e, g}\} \cup \{\text{d, g}\} \quad = \quad \{\text{d, e, g}\}\end{aligned}$$

More details about the implementation of the containment operator can be found in [8]. A new procedure *Eliminate* (described in **Procedure Eliminate()**) is introduced, that uses the containment operator and other ZBDD based set operators to determine whether minterm P is contained in minterm Q. Such a procedure is central for the diagnosis problem studied here. **Procedure Eliminate()** shows that the Eliminate procedure uses the standard ZBDD operators, which clearly is a measure for the time efficiency of the operator. In the remaining, we illustrate the use of the Eliminate procedure in the diagnosis problem studied in this paper.

Example: Assume X_1 = {abd, abe, abg, cde, ceg, egh} be the set of MPDFs and X_2 = {ab, ce} be the set of SPDFs. Then a call to the procedure Eliminate(X_1, X_2) results in the set $\{egh\}$ which contains MPDFs that does not contain any of the SPDFs of X_2. If X_1 is described as a set of PDFs and X_2 as a set of subfaults, then $Eliminate(X_1, X_2)$ identifies the set of PDFs in X_1 which does not contain any subfaults of X_2. This procedure can be used to eliminate the fault free MPDFs from the suspect set (similar to X_1), using the fault free set (similar to X_2).

3.1 Identifying PDFs with a VNR Test

In this paper we present the first non-enumerative method to identify the exact set of PDFs with VNR test, of any cardinality. Let R_T represent the set of all PDFs robustly tested by the passing set T. For a given line l, let R^l_T represent the set of all partial PDFs tested robustly by the passing set T, originating from the line l and terminating at any primary output. This is a subset of set R_T. For a given passing test $t \in T$, let N_t represent the set of all PDFs non-robustly tested by a passing test t. For a given line l, let N^l_t represent the set of all partial PDFs tested non-robustly by a passing test t, originating from the line l and terminating at any primary output. For a given line l, let P^l_t represent the set of all partial PDFs tested robustly by a passing test t, originating from the line l and terminating at any primary output.

Procedure: Extract_VNRPDF

$R_T \leftarrow$ Set of all PDFs tested by robust tests from passing set T
for each line l in topological order **do**
  $R^l_T \leftarrow$ partial PDFs tested robustly from l to Primary Outputs
for all t such that $t \in$ passing set T **do**
  $N_t \leftarrow$ Set of all PDFs tested non-robustly by test t
  for all l such that $l \in$ Primary Input **do**
    $N^l_t \leftarrow$ partial PDFs tested non-robustly from l to Primary Outputs
  for each line l in topological order; $l \notin$ Primary Input **do**
    $P^l_t \leftarrow$ partial PDFs tested robustly from Primary Inputs to l
    $N^l_t \leftarrow$ Subset of $N^{l_p}_t$ from l to Primary Outputs
    (l_p is the predecessor of l)
    if l is non-robustly sensitized **then**
      for each off-input l_o of l **do**
        /**Check for Robustly Tested Off-Input**/
        $N^l_t \leftarrow (N^l_t \cap ((R_T \cap (R^{l_o}_T * P^{l_o}_t)) \supset P^{l_o}_t))$
        if $N^l_t \neq \emptyset$ **then**
          $P^l_t \leftarrow$ Extend $P^{l_p}_t$ from l_p to l
  for all l such that $l \in$ Primary Output **do**
    $VNR \leftarrow VNR \cup P^l_t$

The set R_T, is computed by Procedure Extract_RPDF. Then a topological traversal and for each line l in the circuit, we compute R^l_T. Subsequently, for each passing test $t \in T$, the set of all PDFs non-robustly tested by the passing test

t is done by a modified Procedure Extract_RPDF where we only change the sensitization criteria. Next each line l in the circuit is processed in topological order. If a line l is robustly sensitized, P_t^l is propagated to a line which is the successor of l. If line l is non-robustly tested, then a check is performed at each off-input l_o of the gate whose on-input is l, to identify if there exists a VNR test for the partial PDFs terminating at line l. If there exists a VNR test at the off-inputs, then there exists no delay fault at any off-input l_o and we propagate the set P_t^l forward to a successor. For each primary output po, the set P_t^{po} represents the set of fault free PDFs with VNR tests that terminate at the output po.

Let us now describe the method. Consider the CUD shown in Figure 3a (with line numbers indicated). Let the passing set be $T = \{T_1, T_2\}$ where T_1 = {100001, 101100} and T_2 = {010001, 011100} shown in Figures 3b and 3c respectively. The set of PDFs with robust tests from T is R_T={↑4·10·15·20·21·24}. The set of PDFs with VNR tests are identified using the *P*rocedure Extract_VNRPDF. The Table 2 illustrates the procedure to extract PDFs tested by VNR tests. Test T_1 does not test any PDF non-robustly. The set of PDFs non-robustly tested by T_2 is {↑3·9·14·19·23, ↑3·9·14·20·21·24}. N_2^9, the set of partial PDFs tested by T_2 non-robustly from line 9 to the primary output, is initialized to {9·14·19·23,9·14·20·21·24}. R_T^{15}, the set of partial PDFs tested by robust tests of T, from line 15 to the primary output, is initialized to {20·21·24}. We observe that line 14 is non-robustly tested. A check for a VNR test is performed using the relation,

$$N_2^{14} \leftarrow N_2^{14} \cap ((R_T \cap (R_T^{15} * P_2^{15})) \supset P_2^{15})$$

where N_2^{14} = {14·19·23, 14·20·21·24} before VNR check, P_2^{15} = {4·10} and R_T^{15} = {15·20·21·24}. So N_2^{14} becomes {14·20·21·24} after the VNR check. The partial path {14·20·21·24} allows the PDFs of P_2^{14} to propagate without a delay fault despite line 14 being non-robustly tested. The off-input (line 15) is robustly tested. At the end of topological traversal, P_2^{24} = {↑3·9·14·20·21}. It can be concluded that the PDF {↑3·9·14·20·21·24} is a PDF with VNR test and is fault free.

4 Path Delay Fault Diagnosis

For a given passing set T, procedures Extract_RPDFs and Extract_VNRPDF extracts the fault free set. For the failing set F, a procedure (a modified version of the procedure Extract_RPDF) is used to identify all sensitized PDFs. We first extract the set of sensitized PDFs for the passing set and then the failing set. Using the Eliminate operator and other standard ZBDD operators, we propose a method to perform the diagnosis using the rules provided in Section 2. Let P_m represent the fault free MPDFs (robustly or VNR tested). Let P_s represent the fault free SPDFs (robustly or VNR tested). Let S be the suspect set representing the set of suspicious PDFs. The whole process of diagnosis is described as shown below:

1. Phase I - Extract the sets P_s,P_m and S.
2. Phase II - Optimize the fault free set by eliminating the redundant MPDFs in the set. For example, the PDF P_iP_j is eliminated from the fault free set if PDF P_i is also present.
3. Phase III - Eliminate the PDFs from the suspect set, that are present in both the suspect set and the fault free PDFs. Subsequently, eliminate the fault free MPDFs from the suspect set using the optimized fault free PDFs.

```
Procedure: Diagnosis
    S = (S − P_s)
    S = (S − P_m)
    S = Eliminate(S, P_s)
    S = Eliminate(S, P_m)
```

Note that the Procedure Eliminate does not eliminate the PDFs that are common to both the suspect set and the fault free set. PDFs common to the suspect set and the fault free set are eliminated using a set difference operator. Then we eliminate the fault free MPDFs from the S using the set of optimized fault free PDFs. More formally the Phase III is described as the Procedure Diagnosis.

5 Experimental Results

The experiments were run on a 750MHz SUN Blade-1000 workstation with 1GB RAM. The ISCAS'85 benchmarks were used as the circuits under diagnosis. For the purpose of experimentation, tests were generated using the method proposed in [6]. The method of [6] generates robust and non-robust tests. However it does not generate pseudo-VNR tests. Hence the test sets generated for the ISCAS'85 benchmarks consist only of robust and non-robust tests. Out of the tests generated, 75 tests were assumed to form the failing set and the rest be the passing set.

Table 3 reports the result of extracting the fault free PDFs and the optimization of the set of fault free PDFs. Column 2 shows the cardinality of the passing set. Columns 3 and 4 show the number of fault free MPDFs and SPDFs tested by the given test set, respectively. Column 5 reports the number of fault free MPDFs after optimization (i.e., elimination) using the set of fault free PDFs with robust tests (SPDFs and MPDFs). After the fault free PDFs with robust tests are extracted and optimized, the set of PDFs with VNR tests is extracted. The number of PDFs with VNR tests is shown in Column 6. Certain MPDFs in the fault free set can be still optimized using the identified fault free PDFs with VNR tests (reported in Column 7). The cardinality of the fault free set of PDFs is reported in Column 8 (Sum of Columns 4,6 and 7). Column 9 shows the total processing time. The performance of a diagnosis methodology should not be judged by the total processing time taken. Time is not a crucial factor in fault diagnosis. For a given passing set, the more the number of faults identified as fault free, the better the chance for locating the fault. The proposed method is guaranteed to identify at least as many fault free PDFs as the methodology of [9]. The increase in the number of the fault free PDFs is attributed to the PDFs with the VNR tests. More details about the fault free set is shown in Table 4. Column 2 shows the number of PDFs identified

Benchmark	Passing Test Vectors	Fault Free MPDFs	Fault Free SPDFs	MPDFs (Optm.)	PDFs with VNR Test	MPDFs (Optm.)	Fault Free PDFs	Time (sec)
C880	2,896	922	8,681	902	1,129	902	10,712	21.63
C1355	16,697	1,209	16,846	852	23,735	796	41,377	256.11
C1908	11,404	2,693	23,325	2,693	4,648	2,352	30,325	290.81
C2670	5,353	3,634	9,449	3,152	1,809	3,089	14,347	424.38
C3540	15,730	5,472	22,332	4,861	6,912	4,814	34,058	1986.09
C5315	11,373	4,206	33,936	2,961	4,613	2,961	41,510	1090.81
C6288	3,892	985	1,983	985	693	985	3,661	942.69
C7552	27,902	6,977	52,577	6,872	4,931	6,863	64,371	3749.02

Table 3: Identification of Fault Free PDFs

Benchmark	Fault Free PDFs [11]	Fault Free PDFs (proposed)	Increase in Fault Free PDFs
C880	9,583	10,712	1,129
C1355	17,698	41,377	23,679
C1908	26,018	30,325	4,307
C2670	12,601	14,347	1,746
C3540	27,193	34,058	6,865
C5315	36,897	41,510	4,613
C6288	2,968	3,661	693
C7552	59,449	64,371	4,922

Table 4: Improvement in Diagnosis

	Suspect Set			Diagnosis [11]			Diagnosis (proposed)			Resolution		
Benchmark	MPDFs	SPDFs	Cardinality	MPDFs	SPDFs	Cardinality	MPDFs	SPDFs	Cardinality	[11]	(proposed)	Improvement
C880	42	126	168	42	114	156	28	91	119	7.1	29.2	310%
C1355	98	269	367	87	249	336	62	158	220	8.4	40.1	410%
C1908	81	214	295	73	197	270	42	151	193	8.5	34.6	480%
C2670	115	232	347	109	219	328	74	168	242	5.4	30.3	560%
C3540	92	178	270	78	158	236	55	117	172	12.6	36.3	189%
C5315	154	226	380	139	194	333	82	159	241	12.4	36.5	195%
C6288	97	188	285	93	179	272	62	147	209	4.6	26.7	480%
C7552	132	172	304	119	156	275	82	121	203	9.5	33.2	250%

Table 5: Result of Diagnosis

as fault free using the method in [9]. (Sum of Columns 4 and 5). Column 3 reports the number of PDFs identified as fault free by the proposed method and Column 4 reports the increase in the number of fault free PDFs.

The results of the diagnostic resolution process is presented in Table 5. Columns 2 and 3 represent the number of suspicious MPDFs and SPDFs in the suspect set. Column 4 represents the cardinality of the suspect set, which is the sum of Columns 2 and 3. Columns 5 and 6 show the size of the MPDFs and SPDFs in the suspect set after performing diagnosis using the set of fault free PDFs with robust tests as proposed in [9] (The set shown in Column 2 of Table 4). The cardinality of the resulting suspect set is shown in Column 7. Columns 8 and 9 show the size of the MPDFs and SPDFs in the suspect set after performing diagnosis using the set of fault free PDFs with robust and VNR tests, as proposed in this paper. The cardinality of the resulting suspect set using the proposed method is shown in Column 10. It can be seen from Column 10 that the size for the suspect set is greatly reduced when compared to [9] which is attributed to the identification of more fault free PDFs. Columns 11 and 12 shows the resolution of the diagnosis process using the method of [9] and the proposed technique. It is observed from Column 13 that there is an average increase of 360% in the resolution using the proposed method over [9].

The method [9] presented results on some of the ISCAS'89 benchmarks and showed that the method could result in a reasonable resolution. However Table 5 (Column 11) shows that the method results in a poor resolution of 10%. This is attributed to the fact that the ISCAS'89 benchmarks that were examined in [9] have more than 90% of the PDFs to be robustly testable (shown by [1]). However the ISCAS'85 benchmarks for which results are reported in Table 5 have less than 15% of the PDFs in the circuit to be robustly testable (shown by [3]). This impacts the performance of [9]. The proposed method is expected to perform better if the test set generated for performing diagnosis, explicitly targets the generation of pseudo-VNR tests, like [2].

6 Conclusion

A new non-enumerative framework has been introduced for performing diagnosis using polynomial number of ZBDD operations. The method also takes advantage of the PDFs with VNR tests to improve diagnostic resolution by three folds than existing work. The first method to identify PDFs with VNR test for a given test set is also introduced. The proposed algorithms use the ZBDD as the underlying data structure which has been shown to be effective in storing and manipulating PDFs, non-enumeratively.

References

[1] Bhattacharya D., Agrawal P. and Agrawal V.D., *Test Generation for Path Delay Faults using Binary Decision Diagrams*, IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems, vol. 44, no. 3, Mar. 1995, pp. 434-447.

[2] Cheng K.T., Krstic A. and Chen H.C., *Generation of High Quality Tests for Robustly Untestable Path Delay Faults*, IEEE Trans. on Computers, vol.45, no. 12, Dec. 1996, pp. 1379-1392.

[3] Cheng K.T and Chen H.C.,*Classification and Identification of Nonrobust Untestable Path Delay Faults*, IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems, vol. 15, Aug. 1996, pp. 845-853.

[4] Ke W. and Menon P.R., *Synthesis of Delay-Verifiable Combinational Circuits*, IEEE Trans. on Computers, vol. 44, Feb. 1995, pp.213-222 .

[5] Konuk H., *On Invalidation Mechanisms for Non-Robust Delay Tests*, Proc. of International Test Conference, 2000.

[6] Michael M. and Tragoudas S., *ATPG for Path Delay Faults without Path Enumeration*, Proc. of International Symposium on Quality of Electronic Design, 2001, pp. 384-389.

[7] Minato S-I., *Zero-Suppressed BDDs for Set Manipulation in Combinatorial Problems*, Proc. of Design Automation Conference, 1993, pp. 272-277.

[8] Padmanaban S. and Tragoudas S., *Exact Grading of Multiple Path Delay Faults*, Proc. Design Automation and Test in Europe, 2002, pp. 84-88.

[9] Pant P., Hsu Y.C., Gupta S.K and Chatterjee A., *Path Delay Fault Diagnosis in Combinational Circuits With Implicit Fault Enumeration*, IEEE Trans. Computer Aided Design, vol.20, no. 10, Oct. 2001, pp. 1226-1235.

[10] Reddy S.M., Lin C. and Patil S., *An Automatic Test Pattern Generator for the Detection of Path Delay Faults*, Proc. Intl. Conf. Computer Aided Design, 1987, pp. 284-287.