Z-Sets and Z-Detections: Circuit Characteristics that Simplify Fault Diagnosis

Irith Pomeranz[1], School of ECE, Purdue University, W. Lafayette, IN 47907
Srikanth Venkataraman, Intel Corporation, Hillsboro, OR 97124
Sudhakar M. Reddy[2], ECE Dept., University of Iowa, Iowa City, IA 52242
Bharath Seshadri, School of ECE, Purdue University, W. Lafayette, IN 47907

Abstract

We define the concepts of z-sets and z-detections for combinational circuits (or the combinational logic of scan circuits). Based on these concepts we define structural characteristics and characteristics based on fault simulation. We show that these characteristics determine the numbers of fault pairs that are *guaranteed* to be distinguished by a given fault detection test set. These fault pairs do not need to be considered during diagnostic fault simulation or test generation. We demonstrate that benchmark circuits as well as industrial circuits have these characteristics to a larger extent than may be expected. As a result, only small percentages of fault pairs need to be considered during diagnostic fault simulation or test generation once a fault detection test set is available. In addition, these fault pairs can be identified efficiently.

1. Introduction

Fault diagnosis [1] is a process that requires direct or indirect consideration of pairs of faults. For example, the goal of diagnostic test generation is to ensure that for every pair of faults f_i and f_j there is at least one test t such that the circuit-under-test produces a different output response depending on whether f_i or f_j is present in the circuit. Efficient procedures for diagnostic fault simulation and test generation rely on implicit consideration of fault pairs [1]-[10]. For example, in a diagnostic tree based on a test set T [1], a test $t \in T$ partitions a subset of faults F into two subsets. The subset $F_P \subseteq F$ contains the faults that pass the test t, while the subset $F_F \subseteq F$ contains the faults that fail the test t. After considering all the tests in T, the subsets of faults obtained in the leaves of the diagnostic tree contain indistinguished faults, while two faults in two different subsets are distinguished by T.

The methods of [1]-[10] are based on the use of data structures or test generation techniques that will result in efficient manipulation of fault pairs. A different approach is to use structural circuit information to speed up the diagnostic process. In [11], if a circuit-under-test exhibits a faulty value on output z, it is concluded (under the single fault assumption) that the circuit contains a fault which is located in the cone of logic driving output z. All other faults can be excluded from consideration.

In this work, we show new circuit properties based on structural information and information available from fault simulation for combinational circuits (or the combinational logic of scan circuits). The properties are based on the concept of z-sets, which are related to structural information, and the concept of z-detections, which are derived from fault simulation. We show that fault pairs possessing certain properties based on z-sets and z-detections are guaranteed to be distinguished by a given fault detection test set. Such fault pairs do not need to be considered during diagnostic fault simulation or test generation. All the benchmark circuits we analyzed as well as industrial circuits possess these properties to a larger extent than may be expected. As a result, only small percentages of fault pairs need to be considered for diagnostic fault simulation or test generation once a fault detection test set is available. This result is consistent with the results of earlier works; however, this is the first time this phenomenon is explained, and it is done through the concepts of z-sets and z-detections. Furthermore, these concepts provide a way to efficiently identify fault pairs that are not guaranteed to be distinguished by a fault detection test set.

It is important to note that the concept of z-set is independent of any test set. The concept of z-detection is test set dependent; however, a fault is either z-detected by a test set or not, and thus it is not related to any individual test. As a result, fault analysis based on these concepts is done independent of any individual test.

In the following sections we define the concepts of z-sets and z-detections and the related circuit properties. We demonstrate the extent to which benchmark circuits and industrial circuits possess these properties, and the effects of these properties on the numbers of fault pairs that are distinguished by a fault detection test set. For simplicity of presentation we consider only single stuck-at faults. We use the collapsed set of single stuck-at faults in all the examples we consider.

1. Research supported in part by NSF Grant No. CCR-0098091 and in part by SRC Grant No. 2001-TJ-950.
2. Research supported in part by NSF Grant No. CCR-0097905 and in part by SRC Grant No. 2001-TJ-949.

2. z-sets and related properties

To present the first property we use the following notation. We consider a circuit with n outputs denoted $z_0, z_1, \cdots, z_{n-1}$. For a line g in the circuit, we denote by $Z(g)$ the set that contains every output z_i such that there is a directed path in the circuit from g to z_i. We refer to $Z(g)$ as the z*–set* of g.

We represent a z-set Z (or $Z(g)$) using a vector $\zeta_0\zeta_1 \cdots \zeta_{n-1}$ where $\zeta_i = 1$ if $z_i \in Z$ and $\zeta_i = 0$ if $z_i \notin Z$.

For a fault f, which is associated with line g, we define the z-set $Z(f) = Z(g)$. The fault f can be the fault g stuck-at 0 or the fault g stuck-at 1. We denote by $F(Z)$ the set of faults that contains every fault f such that $Z(f) = Z$.

For illustration, we show in Figure 1 a circuit with two outputs z_0 and z_1. For line 1 we have the z-set $Z(1) = \{z_0\}$, for line 3 we have the z-set $Z(3) = \{z_0, z_1\}$, and for line 9 we have the z-set $Z(9) = \{z_1\}$. The lines in the circuit are partitioned into three subsets depending on the outputs they drive. Each (non-empty) subset of lines corresponds to a different z-set. The three z-sets in this example are $Z_0 = \{z_0, z_1\}$ (represented as $\zeta_0\zeta_1 = 11$), $Z_1 = \{z_0\}$ (represented as $\zeta_0\zeta_1 = 10$) and $Z_2 = \{z_1\}$ (represented as $\zeta_0\zeta_1 = 01$). The faults are partitioned into subsets $F(Z_i)$ according to the same z-sets. Denoting the fault g stuck-at a by g/a we have $F(11) = \{3/1, 4/1, 8/0, 8/1\}$ (these are the faults whose sites have paths to both outputs), $F(10) = \{1/1, 2/1, 7/0, 10/0, 12/0, 12/1\}$ (these are the faults whose sites have paths to z_0) and $F(01) = \{5/1, 6/1, 9/0, 11/0, 13/0, 13/1\}$ (these are the faults whose sites have paths to z_1).

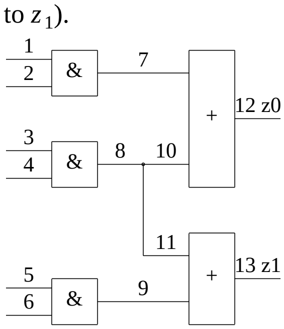

Figure 1: Example circuit

For further illustration, we consider the combinational logic of MCNC finite-state machine benchmark *train*4. The circuit has three outputs, z_0, z_1, z_2 and 34 faults. The lines in the circuit are partitioned into five subsets depending on the outputs they drive. Each subset of lines corresponds to a different z-set. The five z-sets are $Z_0 = \{z_0, z_1, z_2\}$ (or 111), $Z_1 = \{z_0, z_1\}$ (or 110), $Z_2 = \{z_1\}$ (or 010), $Z_3 = \{z_0\}$ (or 100) and $Z_4 = \{z_2\}$ (or 001). The faults in $F(Z_i)$ are the ones whose fault sites drive the outputs in the z-set Z_i. We show the numbers of faults for each z-set in Table 1. Under column z*–set* we show the z-set and under column *flts* we show the number of faults in the z-set. The numbers under column z*–det* will be explained later.

Table 1: z-sets for *train*4

	z-set	flts	z-det
0	111	9	3
1	110	15	6
2	010	6	6
3	100	2	2
4	001	2	2

It is important to note that the number of z-sets that can be defined for a circuit with n outputs is $2^n - 1$. However, only some of these z-sets correspond to any fault. For example, in the case of *train*4 that has three outputs, we can define seven z-sets; however, only five of them (shown in Table 1) have any faults associated with them. In general, the number of relevant z-sets is not larger than the number of faults.

Next, we focus on faults whose z-sets contain a single output. Such a fault can only be propagated to the single output contained in its z-set. Two faults f_i and f_j such that $|Z(f_i)| = |Z(f_j)| = 1$ and $Z(f_i) \neq Z(f_j)$ are guaranteed to be distinguished as long as at least one of them is detected. Since detection is a necessary condition for diagnosis, diagnosis is achieved without any additional effort for such faults. This is the motivation for our interest in these faults. In this section we concentrate on the number of faults with z-sets of size one. We will return to the effect on diagnosis later.

For the circuit of Figure 1, we have two z-sets of size one, 10 and 01. The sets of faults with these z-sets are $F(10) = \{1/1, 2/1, 7/0, 10/0, 12/0, 12/1\}$ and $F(01) = \{5/1, 6/1, 9/0, 11/0, 13/0, 13/1\}$. Thus, we have 12 faults in z-sets of size one in this example. The total number of faults for this circuit is 16. For *train*4, we have three z-sets of size one, 010, 100 and 001. The number of faults included in these z-sets is 10 out of a total of 34 faults.

One may expect that in large circuits, most of the logic would be shared among multiple outputs. As a result, there would be very few faults with z-sets of size one. In reality, benchmark circuits as well as industrial circuits have very large percentages of the faults in logic that only drives a single output. In Table 2 we show information about the faults with z-sets of size one in the combinational logic of ISCAS-89 and ITC-99 benchmark circuits that have more than 1500 faults (more than 1M fault pairs). We also show the same information for industrial circuits (named $a1$, $a2$, $\cdots$, $a8$). After the circuit name we show the total number of faults in the circuit and the number of different z-sets that these faults define. We then show the number of faults that have z-sets of size one, and the percentage of such faults out of the total number of faults.

Table 2: Faults in z-sets of size one
(a) Benchmark circuits

circuit	flts	zsets	z=1	%z=1
s1423	1515	181	809	53.40
s5378	4603	662	1279	27.79
s9234	6927	567	4066	58.70
s13207	9815	1424	5778	58.87
s15850	11725	1477	6453	55.04
s35932	39094	4112	16516	42.25
s38417	31180	3505	14861	47.66
b14	9981	738	5595	56.06
b20	22579	1508	10486	46.44

(b) Industrial circuits

circuit	flts	zsets	z=1	%z=1
a1	37209	7568	20411	54.85
a2	46111	9701	31323	67.92
a3	13687	2084	7874	57.52
a4	30887	6403	19046	61.66
a5	293816	58659	214464	72.99
a6	236106	32564	109329	46.31
a7	144490	27148	84947	58.79
a8	171034	37511	121224	70.87

Table 2 demonstrates that large percentages of the faults in benchmark circuits and in industrial circuits drive single outputs (or have z-sets of size one). We continue to discuss the importance of this property for diagnosis after introducing another property in the next section.

3. z-detection and related properties

To introduce the second property, consider a fault f_i with z-set $Z(f_i)$. Suppose that there is a test t for f_i, which propagates the effects of f_i to *all* the outputs in $Z(f_i)$. If $Z(f_i)$ is of size one, then every test that detects f_i must satisfy this property. In general, this property is important for diagnosis even if $|Z(f_i)| > 1$ due to the following reason.

Let f_i and f_j be two faults such that $Z(f_i) \not\subseteq Z(f_j)$. Suppose that a test t detects f_i by propagating the effects of f_i to all the outputs in $Z(f_i)$. Since $Z(f_i) \not\subseteq Z(f_j)$, even if t detects f_j, it cannot propagate the effects of f_j to all the outputs in $Z(f_i)$. Consequently, t distinguishes f_i from f_j at least on one of the outputs in $Z(f_i)$–$Z(f_j)$.

To demonstrate this point, we consider two faults of MCNC finite-state machine benchmark $dk27$. The circuit has four inputs and five outputs. We consider fault f_8 with $Z(f_8) = 01100$ and fault f_{11} with $Z(f_{11}) = 00101$. Under the test 0001, the fault free output vector is 00000. For f_8 we obtain the output vector 01100. Regardless of the output vector of f_{11}, this test is guaranteed to distinguish f_8 from f_{11} since f_8 is propagated to output z_1, which f_{11} does not drive.

Due to this property, we are interested in the number of faults with tests that propagate fault effects to all the outputs in their z-sets. If a fault has more than one test that detects it, we are interested in any one of these tests propagating fault effects to all the outputs in the z-set of the fault. We refer to the detection of a fault on all the outputs in its z-set as z*–detection*. We will generalize this property later.

To find the faults that are z-detected by a given test set T, we simulate T, dropping a fault f from further simulation only if it is z-detected by a test in T. As long as the fault is not z-detected, we continue to simulate it even if it is already detected.

We report on the numbers of z-detected faults in the combinational logic of the same ISCAS-89 and ITC-99 benchmark circuits considered in Table 2 under three types of test sets (when they are available): (1) compacted conventional fault detection test sets; (2) compacted 10-detection test sets; and (3) uncompacted conventional fault detection test sets.

The results for conventional test sets are shown in Table 3 (we comment on the results for 10-detection test sets later but we do not present them for space considerations). In each part of Table 3, after the circuit name we show the number of faults, and the number of faults detected by each one of the test sets considered. We then show the number of z-detected faults (i.e., the number of faults detected on all the outputs in their z-sets by at least one test), and the percentage of z-detected faults out of all the detected faults. Under columns *ndet* of Table 3 we show the following measure of fault simulation effort for finding z-detected faults.

Table 3: z-detections
(a) Compacted test sets for benchmark circuits

circuit	flts	det	zdet	%zdet	ndet
s1423	1515	1501	888	59.16	1.16
s5378	4603	4563	2811	61.60	2.00
s9234	6927	6475	4288	66.22	2.29
s13207	9815	9664	6599	68.28	2.83
s15850	11725	11336	7802	68.82	1.71
s35932	39094	35110	18557	52.85	1.20
s38417	31180	31015	19798	63.83	1.96
b14	9981	8213	4284	52.16	1.15
b20	22579	19721	8354	42.36	1.37

(b) Uncompacted test sets for benchmark circuits

circuit	flts	det	zdet	%zdet	ndet
s1423	1515	1501	916	61.03	1.72
s5378	4603	4563	2780	60.92	4.70
s9234	6927	6475	4337	66.98	6.16
s13207	9815	9664	6546	67.74	5.32
s15850	11725	11336	7840	69.16	4.34
s35932	-	-	-	-	-
s38417	31180	31015	21493	69.30	7.06
b14	9981	8213	4287	52.20	1.19
b20	22579	19721	8365	42.42	1.59

During the fault simulation process, we count for every fault f the number of times it is detected by the test set T until it is dropped, or until the fault simulation process ends. We denote this number by $n_{det}(f)$. For example, suppose that a fault f is simulated under a test set $T = \{t_0, t_1, \cdots, t_5\}$. Suppose that f is not detected by t_0, f is detected by t_1, f is detected by t_2, f is not detected by t_3, and f is z-detected by t_4. Since the last detection is a z-detection it causes f to be dropped. In this case, $n_{det}(f) = 3$ due to the detection of f by t_1, t_2 and t_4 before it is dropped.

Considering all the z-detected faults, we show in columns *ndet* of Table 3 the average value of $n_{det}(f)$. This number has the following interpretation. The fault simulation process to determine z-detections is similar to an n-detection fault simulation process, where a fault is dropped only after it is detected n times. In our case, n varies from one fault to the next, with an average value as reported in columns *ndet* of Table 3.

It can be seen that fault simulation to determine z-detections drops a fault after it is detected an average of approximately two to three times for a conventional compacted test set. Larger numbers are obtained for uncompacted and for 10-detection test sets, since the faults are detected larger numbers of times by these test sets.

More important, it can be seen from Table 3 that there are large percentages of faults in benchmark circuits that are z-detected. This is in spite of the fact that the test sets we consider are not designed to result in z-detections. Some of these z-detections are due to faults with z-sets of size one, where every detection is a z-detection; however, other faults are z-detected as well. Considering, for example, $s1423$, Table 2 shows that 53.40% of the faults have z-sets of size one. Considering only detectable faults, the percentage of faults with z-sets of size one is similar, 53.30%. Table 3 shows that a larger percentage of the faults, 59.16%, are z-detected under a compacted conventional test set. When a compacted 10-detection test set is used, the percentage of z-detections is even higher, 63.56% for $s1423$. Uncompacted test sets result in numbers of z-detected faults that are very close to those of the compacted test sets.

We used specific test sets to obtain the data in Table 3. Therefore, if a fault is not z-detected, we have no information as to whether the fault can be z-detected by any test, or it is z*–undetectable*. To collect information about the percentages of z-detectable faults, we considered the combinational logic of MCNC finite-state machine benchmarks, that have small numbers of inputs, under exhaustive test sets. We found percentages of z-detected faults that are similar to the ones in Table 3. We conclude that z-detection occurs accidentally for large percentages of the z-detectable faults.

It is possible to generalize the concept of z-detection as follows. We say that a fault f is z-detected by a test t that propagates the effects of f to all the outputs in its z-set. Suppose that instead of t, we have tests $t_1, t_2, \cdots, t_m$, such that t_i propagates the effects of f to a set of outputs $Z(f, t_i)$. Suppose in addition that $\cup_{i=1}^{m} Z(f, t_i) = Z(f)$. We say that f is $\cup z$*–detected* by the test set in this case. The notion of $\cup z$-detection can replace the notion of z-detection in diagnosis as follows.

Let f_i and f_j be two faults such that $Z(f_i) \not\subseteq Z(f_j)$. Suppose that f is $\cup z$-detected by the test set T. Since $Z(f_i) \not\subseteq Z(f_j)$, we have at least one test $t \in T$ that detects f_i on an output which is not in $Z(f_j)$. This test cannot detect f_j on the same output. Consequently, t distinguishes f_i from f_j on an output in $Z(f_i) - Z(f_j)$.

Fault simulation to collect information about $\cup z$-detected faults keeps track of the z-set for every fault f. Every time f is detected, we mark the outputs in its z-set on which f is detected. A fault can be dropped once all the outputs in its z-set are marked. In this case, the fault is $\cup z$-detected.

We show information on the numbers of $\cup z$-detected faults in benchmark circuits in Table 4. We only consider compact conventional test sets in this case.

Table 4: $\cup z$-detections

circuit	flts	det	zdet	%zdet	ndet
s1423	1515	1501	939	62.56	1.30
s5378	4603	4563	3180	69.69	3.09
s9234	6927	6475	4786	73.92	3.39
s13207	9815	9664	6984	72.27	4.85
s15850	11725	11336	8567	75.57	2.76
s35932	39094	35110	25348	72.20	1.51
s38417	31180	31015	22140	71.38	2.42
b14	9981	8213	4715	57.41	2.40
b20	22579	19721	9914	50.27	3.29

Table 4 demonstrates that the numbers of $\cup z$-detected faults in benchmark circuits are larger than the numbers of z-detected faults. This will translate into increases in the numbers of fault pairs that are guaranteed to be distinguished by a fault detection test set.

4. Using z-sets and z-detections

In this section we show how the properties related to z-sets and z-detections can be used to identify fault pairs that are guaranteed to be distinguished by a given fault detection test set. Such pairs do not need to be considered during diagnostic fault simulation or test generation once a fault detection test set is available. We describe efficient procedures for counting (or enumerating) the fault pairs that remain. We demonstrate that relatively small percentages of fault pairs remain after z-sets and z-detections are taken into account. The remaining fault pairs can be considered under implicit methods [1]-[10], or explicitly.

We first use only z-sets, which can be derived based on structural information. We then use z-detections, which require additional fault simulation. In all the cases, we consider only pairs of detectable faults.

When considering only z-sets, we distinguish between pairs of z-sets Z_i and Z_j that have at least one output in common, and pairs of z-sets that are disjoint ($Z_i \cap Z_j = \phi$). If $Z_i \cap Z_j = \phi$, then for every pair of faults $f_i \in F(Z_i)$ and $f_j \in F(Z_j)$, a test that detects any one of the faults will distinguish them. For example, we consider faults $f_{17} \in F(110)$ and $f_{33} \in F(001)$ of *train* 4. If a test t detects f_{17}, then either z_0 or z_1 assume faulty values in the presence of f_{17}. Since f_{33} cannot affect these outputs, the faults are distinguished by the test. Similarly, if a test t detects f_{33}, then z_2 assumes a faulty value in the presence of f_{33}. Since f_{17} cannot affect z_2, the faults are distinguished by the test.

Thus, when $Z_i \cap Z_j = \phi$, fault pairs $f_i \in F(Z_i)$ and $f_j \in F(Z_j)$ are guaranteed to be distinguished by a fault detection test set. In contrast, if $Z_i \cap Z_j \neq \phi$, the faults may be detected on the same outputs, and it is necessary to consider the faults further. As a special case, if $Z_i \cap Z_j = \phi$ and $|Z_i| = |Z_j| = 1$, the fault pairs over $F(Z_i)$ and $F(Z_j)$ are guaranteed to be distinguished by tests that detect them. We discussed this special case earlier and demonstrated that large numbers of faults have z-sets of size one in benchmark circuits as well as industrial circuits.

Based on the discussion above, we use the following procedure to count the number of fault pairs that remain to be considered after z-sets are taken into account (i.e., the number of fault pairs that are not guaranteed to be distinguished by a fault detection test set). We denote this number by N_{P1}. A similar procedure can be used to enumerate the fault pairs if needed.

(1) Set $N_{P1} = 0$.

(2) For every pair of z-sets Z_i and Z_j (including the case where $Z_i = Z_j$), if $Z_i \cap Z_j \neq \phi$:
If $i = j$, set $N_{P1} = N_{P1} + |F(Z_i)| \cdot [|F(Z_i)| - 1]/2$.
Else, set $N_{P1} = N_{P1} + |F(Z_i)| \cdot |F(Z_j)|$.

Considering *train* 4 with the data shown in Table 1, for $i = j = 0$, Z_0 contributes 9(9-1)/2 = 36 fault pairs to N_{P1}; for $i = 0$ and $j = 1$, Z_0 and Z_1 contribute 9·15 = 135 fault pairs to N_{P1}; and so on. The total number of fault pairs obtained is $N_{P1} = 503$, compared to a total of 561 fault pairs.

To use z-detections, we consider a given test set T. Based on T we partition the set of faults $F(Z_i)$ associated with z-set Z_i into two subsets. The subset $A(Z_i) \subseteq F(Z_i)$ contains the faults that are z-detected by T. The subset $B(Z_i) \subseteq F(Z_i)$ contains the faults that are not z-detected by T (however, they are detected by T). Considering two z-sets Z_i and Z_j, we have the following cases.

Case 1: $Z_i = Z_j$. In this case, the faults in $A(Z_i)$ are guaranteed to be distinguished from the faults in $B(Z_i)$ by the tests that z-detect the faults in $A(Z_i)$. For example, consider faults $f_a \in A(Z_i)$ and $f_b \in B(Z_i)$ under the test t that z-detects f_a. Since f_b is not z-detected, there is an output $z \in Z_i$ such that f_a is detected on this output but f_b is not. Thus, t distinguishes f_a from f_b. This leaves all the fault pairs over $A(Z_i)$ and all the fault pairs over $B(Z_i)$ that need to be considered.

Case 2: $Z_i \not\subset Z_j$ but $Z_j \subset Z_i$. In this case, the faults in $A(Z_i)$ are guaranteed to be distinguished from the faults in $F(Z_j)$ by the tests that z-detect the faults in $A(Z_i)$. This leaves the fault pairs where $f_i \in B(Z_i)$ and $f_j \in F(Z_j)$ to be considered.

Case 3: $Z_i \subset Z_j$ but $Z_j \not\subset Z_i$. In this case, the faults in $A(Z_j)$ are guaranteed to be distinguished from the faults in $F(Z_i)$ by the tests that detect the faults in $A(Z_j)$. This leaves the fault pairs where $f_i \in F(Z_i)$ and $f_j \in B(Z_j)$ to be considered.

Case 4: $Z_i \not\subset Z_j$ and $Z_j \not\subset Z_i$ but $Z_i \cap Z_j \neq \phi$. In this case, the faults in $A(Z_i)$ are guaranteed to be distinguished from the faults in $F(Z_j)$ by the tests that detect the faults in $A(Z_i)$. In addition, the faults in $A(Z_j)$ are guaranteed to be distinguished from the faults in $F(Z_i)$ by the tests that detect the faults in $A(Z_j)$. This leaves the fault pairs where $f_i \in B(Z_i)$ and $f_j \in B(Z_j)$ to be considered.

Case 5: $Z_i \cap Z_j = \phi$. In this case, the faults in $F(Z_i)$ are guaranteed to be distinguished from the faults in $F(Z_j)$ by the tests that detect either one of the faults.

Based on the discussion above, we use the following procedure to count the number of fault pairs that remain to be considered after z-sets and z-detections are taken into account. We denote this number by N_{P2}. A similar procedure can be used to enumerate the fault pairs.

(1) Set $N_{P2} = 0$.

(2) For every pair of z-sets Z_i and Z_j (including the case where $Z_i = Z_j$):
   (a) If $Z_i = Z_j$, set
   $N_{P2} = N_{P2} + |A(Z_i)| \cdot [|A(Z_i)| - 1]/2 + |B(Z_i)| \cdot [|B(Z_i)| - 1]/2$.
   (b) If $Z_i \not\subset Z_j$ but $Z_j \subset Z_i$, set
   $N_{P2} = N_{P2} + |B(Z_i)| \cdot |F(Z_j)|$.
   (c) If $Z_i \subset Z_j$ but $Z_j \not\subset Z_i$, set
   $N_{P2} = N_{P2} + |F(Z_i)| \cdot |B(Z_j)|$.
   (d) $Z_i \not\subset Z_j$ and $Z_j \not\subset Z_i$ but $Z_i \cap Z_j \neq \phi$, set
   $N_{P2} = N_{P2} + |B(Z_i)| \cdot |B(Z_j)|$.

Considering *train* 4 with the data shown in Table 1, the last column of Table 1 shows the number of z-detected faults for every z-set. For $i = j = 0$, Z_0 contributes 3(3-1)/2+6(6-1)/2 = 18 fault pairs to N_{P2}; for $i = 0$ and $j = 1$, Z_0 and Z_1 contribute (9-3)15 = 90 fault pairs to N_{P2}; and so on. The total number of fault pairs obtained is $N_{P2} = 308$, compared to a total of 561 fault pairs for this circuit, and compared to $N_{P1} = 503$ obtained earlier.

We applied the procedures above to count the numbers of fault pairs that remain to be considered after using z-sets, and after using z-detections. The results are reported in Table 5 for the same circuits and test sets considered in Tables 2 and 3. After the circuit name we show the total number of fault pairs defined over the detectable circuit faults. Under column *zsets* we show the value of N_{P1} computed using only z-sets, and under column *%zsets* we show the percentage of N_{P1} out of the total number of fault pairs. Under column *zdet* we show the value of N_{P2} computed using both z-sets and z-detections, and under column *%zdet* we show the percentage of N_{P2} out of the total number of fault pairs. For industrial circuits we only report the results related to N_{P1}, which is based on z-sets.

Table 5: Indistinguished fault pairs

(a) Compacted test sets for benchmark circuits

circuit	total	zsets	%zsets	zdet	%zdet
s1423	1125750	369266	32.80	364347	32.36
s5378	10408203	1147716	11.03	779089	7.49
s9234	20959575	3000343	14.31	2662558	12.70
s13207	46691616	3446376	7.38	3195550	6.84
s15850	64246780	6437785	10.02	6095945	9.49
s35932	616338495	2996733	0.49	2519720	0.41
s38417	480949605	9709275	2.02	9098292	1.89
b14	33722578	11490378	34.07	11329572	33.60
b20	194449060	49865503	25.64	49575658	25.50

(b) Uncompacted test sets for benchmark circuits

circuit	total	zdet	%zdet
s1423	1125750	360898	32.06
s5378	10408203	787849	7.57
s9234	20959575	2700232	12.88
s13207	46691616	3154719	6.76
s15850	64246780	6127417	9.54
s35932	-	-	-
s38417	480949605	8909970	1.85
b14	33722578	11328839	33.59
b20	194449060	49573887	25.49

(c) Industrial circuits

circuit	total	zsets	%zsets
a1	46200078	5751480	12.45
a2	65671530	2842470	4.33
a3	15210370	2596090	17.07
a4	135984786	2441340	1.80
a5	19556980878	36326700	0.19
a6	4111023150	48578100	1.18
a7	8620173253	19116100	0.22
a8	10763892726	11070100	0.10

From Table 5, only small percentages of the fault pairs are not guaranteed to be distinguished by fault detection test sets. For example, for *s*38417, only 1.89% of the fault pairs are left to be distinguished when a compacted fault detection test set is available. Even smaller percentages are obtained for industrial circuits, even though only z-set information is used.

5. Concluding remarks

We defined the concept of a z-set, which is the set of outputs driven by a line or a fault site in a combinational circuit (or the combinational logic of a scan circuit). We defined the concept of z-detection, where a fault is detected on all the outputs in its z-set by at least one test. Based on these concepts we defined circuit characteristics, which ensure that a fault pair will be distinguished by a fault detection test set. For example, if two faults have disjoint z-sets, they are guaranteed to be distinguished by a test set that detects them. In addition, if the z-set of a z-detected fault is not contained in the z-set of another fault, then the two faults are guaranteed to be distinguished by a fault detection test set. We demonstrated that benchmark circuits as well as industrial circuits have these characteristics to a larger extent than may be expected. As a result, the percentages of fault pairs that are not distinguished by a given fault detection test set in benchmark circuits were shown to be small. These fault pairs can be computed efficiently by using z-sets and z-detections.

References

[1] M. Abramovici, M. A. Breuer and A. D. Friedman, *Digital Systems Testing and Testable Design*, Computer Science Press, 1990.

[2] P. Camurati, D. Medina, P. Prinetto and M. Sonza Reorda, "A Diagnostic Test Pattern Generation Algorithm", in Proc. Intl. Test Conf., 1990, pp. 52-58.

[3] T. Gruning, U. Mahlstedt and H. Koopmeiners, "DIATEST: A Fast Diagnostic Test Pattern Generator for Combinational Circuits", in Proc. Intl. Conf. on Computer-Aided Design, Nov. 1991, pp. 194-197.

[4] K. Kubiak, S. Parkes, W. K. Fuchs and R. Saleh, "Exact Evaluation of Diagnostic Test Resolution", 29th Design Autom. Conf, June 1992, pp. 347-352.

[5] E. M. Rudnick, W. K. Fuchs and J. H. Patel, "Diagnostic Fault Simulation of Sequential Circuits", Intl. Test Conf., 1992, pp. 178-186.

[6] F. Corno, P. Prinetto, M. Rebaudengo and M. Sonza Reorda, "GARDA: A Diagnostic ATPG for Large Synchronous Sequential Circuits", in Proc. Europ. Design&Test Conf., March 1995, pp. 267-271.

[7] S. Venkataraman, I. Hartanto, W. K. Fuchs, E. M. Rudnick, S. Chakravarty and J. H. Patel, "Rapid Diagnostic Fault Simulation of Stuck-at Faults in Sequential Circuits using Compact Lists", in Proc. Design Autom. Conf., June 1995, pp. 133-138.

[8] V. Boppana, I. Hartanto and W. K. Fuchs, "Full Fault Dictionary Storage Based on Labeled Tree Encoding", in Proc. VLSI Test Symp., April 1996, pp. 174-179.

[9] S. Venkataraman and W. K. Fuchs, "Distributed Diagnostic Simulation of Stuck-at Faults in Sequential Circuits", in Proc. VLSI Design, Jan. 1997, pp. 381-385.

[10] I. Pomeranz and S. M. Reddy, "A Diagnostic Test Generation Procedure for Synchronous Sequential Circuits Based on Test Elimination", in Proc. Intl. Test Conf., Oct. 1998, pp. 1074-1083.

[11] J. A. Waicukauski and E. Lindbloom, "Failure Diagnosis of Structured VLSI", IEEE Design and Test of Computers, Aug. 1989, pp. 49-60.