A Diagnosis Algorithm for Constant Degree Structures and Its Application to VLSI Circuit Testing

Kaiyuan Huang, Vinod K. Agarwal, *Fellow, IEEE*, Laurence LaForge, and K. Thulasiraman, *Fellow, IEEE*

***Abstract*— A simple diagnosis algorithm is presented for constant degree systems such as rectangular grids connected as tori. The algorithm determines the status of a unit according to the size of its *faction*, a cluster of units that call each other fault-free but outsiders faulty. Almost all units are correctly identified with this algorithm under a binomial failure distribution even when the probability of failure is rather high. The complexity of the algorithm is $O(n)$, where n is the number of units in a constant degree system. The application of the algorithm to production testing of VLSI chips is also considered. With a test board that houses a large number of chips to be tested, all the chips can be tested in parallel in a way that they test each other and the test outcomes, not necessarily correct, are reported to a host system for analysis. The actual status of each chip is determined by using this new diagnosis algorithm. The above chip screening process can be repeated for higher accuracy. It is shown that no more than two steps are needed in most real situations. Compared with testing by test equipment that usually tests only one chip at a time, the saving of test time and the test equipment cost could be significant with our approach.**

***Index Terms*— Probabilistic diagnosis, system level diagnosis, diagnosis algorithm, production testing, VLSI testing.**

I. Introduction

The CLASSICAL system level diagnosis was originated by Preparata, Metze, and Chien [1]. They suggested that a multiple processor system be diagnosed by letting the processors test each other and then by analyzing the outcomes. The outcomes of these interunit tests are classified as "fault-free" or "faulty." It is assumed that a fault-free testing unit always gives the correct outcome while a faulty testing unit can give an arbitrary outcome. It is also assumed that the number of faulty units cannot exceed a predetermined upper bound t. A system is said to be *t-diagnosable* if all faulty units can be identified as long as the number of faulty units does not exceed t. This model is called the PMC model. In order for a system to be t-diagnosable every unit must be tested by at least t other units.

However, for practical structures such as rectangular and octagonal grids, a unit is connected only to a small constant number of other units and therefore the t-diagnosability of these systems is very small although they may contain large numbers of units. Somani, Agarwal, and Avis [2] considered diagnosis of sparsely connected systems, and determined that a large number of fault sets of sizes larger than t (the degree of diagnosability) can still be identified on grids. Probabilistic diagnosis has also gained attention recently [3]–[12]. Scheinerman [3] presented a probabilistic diagnosis algorithm with the probability of correctly identifying every unit approaching one as $n \to \infty$, where n is the number of units in the system. It requires that a unit be connected to slightly more than $\log n$ other units, in contrast to t, $t \leq (n/2)$, other units for a system to be t-diagnosable. Another probabilistic algorithm that imposes the same structural requirement was proposed by Blough [4]. Blough also showed that probabilistic diagnosis with a high probability is impossible if each unit is connected only to $\log n$ other units. Fussell and Rangarajan [8] proposed a different testing strategy for diagnosing constant degree structures. Instead of a single test per test link, multiple tests are performed corresponding to each test link. More specifically, slightly more than $\log n$ tests are performed with respect to each test link. They showed that the probability of correctly identifying every unit approaches one as $n \to \infty$. More recently [9] they further showed that the number of test links per unit and the number of tests per test link can be traded off as long as the product of these two parameters grows as $\Omega(\log n)$ as $n \to \infty$. It is assumed (in the analysis) that the coverage of each test is uniform and independent. The test outcomes must be analyzed in some way to locate faulty units. In the case when the test outcomes are analyzed by a host system, the reporting of test outcomes demands a higher communication bandwidth as $\log n$ bits of data per test link need to be reported to the host instead of just one bit per test link. Together with Malek [13], they discussed the application of their diagnosis algorithm to testing of integrated circuit wafers. In [13], the authors presented good performance numbers for practical sized systems rather than discussing asymptotic behaviors. LaForge *et al.* [14] presented a new approach to diagnosing constant degree systems. Instead of correctly identifying every unit with high probability, it correctly identifies every faulty unit and a specified fraction of good units with high probability while the degree of connection is upper bounded by a constant. Basically, the

Manuscript received December 15, 1991; revised October 26, 1992 and June 7, 1994.

K. Huang was with the Microelectronics and Computer Systems Laboratory, McGill University, Montreal, P. Q., Canada. He is now with the Bell-Northern Research, Ottawa, Ont., Canada.

V. K. Agarwal is with the Microelectronics and Computer Systems Laboratory, McGill University, Montreal, P. Q., Canada.

L. LaForge was with the Microelectronics and Computer Systems Laboratory, McGill University, Montreal, P. Q., Canada. He is now with the Department of Electrical Engineering, University of Nevada, Reno, NV 89557 USA.

K. Thulasiraman is on leave of absence from the Department of Electrical and Computer Engineering, Concordia University, Montreal, P. Q., Canada. He is now with the School of Computer Science, University of Oklahoma, Norman, OK 73019 USA.

IEEE Log Number 9409330.

1045-9219/95$04.00 © 1995 IEEE

algorithm looks for the largest cluster of units that call each other fault-free and declares it to be the set of fault-free units.

In this paper, we will present a new diagnosis algorithm for constant degree systems. As in the traditional system level diagnosis, only one test is performed with respect to each test link, and this eases test generation as well as the reporting of test outcomes. However, unlike probabilistic diagnosis algorithms wherein the probability of correctly identifying every unit approaches one when $n \rightarrow \infty$, our algorithm correctly identifies a constant fraction (very close to 1) of fault-free units and a constant fraction (very close to 1) of faulty units irrespective of the size (except, of course, when n is trivially small, such as $n = 2$) of the system. The objective of this approach differs from that of [14] in that a small fraction of faulty units are allowed to be mis-identified. The benefit is that a larger fraction of fault-free units can be correctly identified.

Finally, we will discuss the application of the algorithm to VLSI circuit testing. Modern VLSI testers usually test one chip at a time. A very high speed tester may be multiplexed to test more chips but this can only achieve a moderate improvement. Our proposed alternative is to have a test board housing a large number of chips to be tested and through connections between the chip sockets to perform interchip tests. Further we add comparators to the test board; the chips to be tested are not necessarily "processors," which are able to perform tests on one another. Basically, they can contain any digital circuit designed with full scan [15]. A set of pseudo random or deterministic test vectors can be broadcast to all chips under test and their results be compared bit by bit by the comparators [16]. The results of the comparisons are equivalent to interprocessor tests. A host system is employed to analyze the test outcomes using our diagnosis algorithm so as to identify the status of the chips in the same way as system level diagnosis of multiple processor systems. As the chips are tested in parallel, the saving of test time could be significant. The actual test time improvement depends on the size of the test board.

II. Preliminaries

The test relationship can be modeled by a digraph $D(V, A)$, called the *test digraph,* wherein V is the set of vertices, one corresponding to each unit, and A is the set of arcs. We assume that if (u, v) belongs to A, then so does (v, u).

As in [4], we assume that the probability of failure of each unit is independent and identical to some predetermined value $p, 0 \leq p \leq 1$. In other words, the distribution of failure is binomial. Therefore, if there are a total of n units in the system and $g(f)$ is the total number of good (bad) units, then $g+f = n$ and the expected number of good units is $E(g) = n(1-p)$ and the expected number of bad units is $E(f) = np$.

The set of test outcomes is called the *syndrome* of the system and can be seen as a function w of arcs. As mentioned earlier, we assume that, for a fault-free vertex $u, w(u,v) = 0$ if v is fault-free, and $w(u,v) = 1$ if v is faulty. On the other hand, if u is faulty, then $w(u,v)$ can take on any value independent of the status of v. In other words, the test outcomes conform to the PMC model. However, we further assume that the probability $w(u,v) = 0$ when u and v are both faulty is some small value represented by $\gamma, 0 \leq \gamma \leq 1$. If u is faulty and v is fault-free, no assumption is made on the probability of $w(u,v) = 0$. This test outcome interpretation can be considered to be a generalization of the BGM model [17]. In fact, the model defaults into the BGM model when $\gamma = 0$.

III. Diagnosis Algorithm

In this section, we will present our diagnosis algorithm. Unlike other probabilistic diagnosis algorithms that use various forms of voting for deciding the status of a unit, our algorithm considers the size of a cluster of units that claim each other to be fault-free but outsiders to be faulty, and then determine the status of the whole cluster of units. Such a cluster of units is called a *faction.* A faction may take various geometric shapes. In the case that a faction of fault-free units forms a narrow string, any of them will likely be misidentified to be faulty with a majority voting based diagnosis algorithm because a unit in such a faction has very few fault-free neighbors to vote for it. The geometric shape of a faction is not important in our algorithm, only is the number of units in the whole faction.

Before presenting our algorithm, we need to define some terms and notation. The *neighbor set* $N(v)$ of a vertex $v \in V$ is the set of vertices adjacent to v in $D(V, A)$. In other words, a vertex u is in $N(v)$ if (v,u) is in A or (u,v) is in A. Recall that in our test digraph, $(u,v) \in A$ implies $(v,u) \in A$. The neighbor set $N(V')$ of a subset $V' \subseteq V$ is the set of vertices adjacent to some vertices in V' but not in V' themselves. The number of neighbors of vertex v in a subset $V' \subset V$ is denoted by $\delta(v, V')$. For a subset $V' \subseteq V$, the V'*-induced subgraph* of $D(V, A)$, denoted by $D[V']$, is the subgraph of $D(V, A)$ whose vertex set is V' and in which there is an arc from u to v if and only if u and v are both in V' and (u, v) is in A. The *out-going arc set* $A^+[V']$ of $V' \subseteq V$ is a subset of the arc set A whose heads are in $V - V'$ and whose tails are in V'. The *internal arc set* $A[V']$ of V' is a subset of A whose heads and tails are both in V'.

With the above preparation, a faction can now be formally defined as follows:

Definition 3.1: A subset of units $V' \subseteq V$ forms a faction of V if the following holds:

1) $D[V']$ is strongly connected, where $D[V']$ is a subgraph of $D(V, A)$ induced by V'.
2) Every arc in $A[V']$ has the weight of 0.
3) Every arc in $A^+[V']$ has the weight of 1.

By the above definition of a faction, we have the following observation:

Every fault-free vertex is always in a faction. This fact can be explained as follows: If a fault-free vertex v is completely isolated by faulty vertices, by our test model, every arc in $A^+[v]$ must assume the weight of 1 and the other two conditions of Definition 3.1 are trivially satisified. This means that v forms a faction by itself. Assume that v is adjacent to some other fault-free vertices. Let V' be the vertex set containing v and the fault-free vertices to which v is adjacent.

If any other fault-free vertices are adjacent to any of the members of V', add them to V'. Repeat this process until all vertices adjacent to vertices in V' that are outside V' are faulty. By our test model, every arc in $A^+[V']$ must assume the weight of 1 and every arc in $A[V']$ must assume the weight of 0. Furthermore, we can see, from the construction of V', that $D[V']$ is strongly connected. This means that V' forms a faction and therefore v is in a faction. Note that faulty vertices may or may not be in a faction and therefore each faction is not necessarily a cluster of fault-free vertices.

A threshold, k, is used to determine the status of each unit. A unit is considered to be fault-free if it is in a faction of size larger than a predetermined k, or faulty otherwise. This simple algorithm can be formalized as follows:

```
Algorithm 3.1 (Fault Identification)
Input: Test digraph $D(V, A)$, syndrome $w$ and threshold value $k$.
Output: Set of vertices declared fault-free: $R$.
Step 1: Set $R = \emptyset$.
Step 2: For every vertex $v \in V$, set $R = R \cup \{v\}$ if $v$ is in a faction of size larger than $k$.
End
```

It is easy to see that the performance of the algorithm depends on the selection of the threshold k. We will address this issue in the next section.

From Algorithm 3.1 we can see that the main task in determining the status of a unit is to determine the size of its faction. A straightforward procedure to determine the size is as follows.

1) Form an *agreement graph* $G(V, E)$, which is an undirected graph, such that an edge (u, v) is in E if and only if $w(u, v) = w(v, u) = 0$.
2) Find the connected components of $G(V, E)$ and determine the sizes of the connected components. It is easy to see that if $G[V']$ is a connected component of $G(V, E)$, then $D[V']$ is strongly connected; i.e., the first condition of Definition 3.1 is satisfied. Furthermore, in the test digraph $D(V, A)$, either (u, v), or (v, u), or both have the weight of 1 for every vertex $u \in V'$ and every vertex $v \in N(u) - V'$.
3) For each connected component, determine if there are two vertices in it such that one calls the other "faulty" and if there is a unit in this component that calls a unit outside the component "fault-free." If both answers are a "no," the last two conditions of Definition 3.1 are satisfied and the number of units in the component is the size of the faction; otherwise, none of the units in the connected component is in any faction. The two types of violations are illustrated in Fig. 1, where both the test digraph and the corresponding agreement graph are shown. In (a) vertices v_0, v_1, v_2 and v_3 form a connected component of the agreement graph. However, because v_3 calls v_0 "faulty," none of these four vertices is in any faction. In (b) vertices v_0 and v_1 form a connected component of the agreement graph. Because v_1 calls v_2, a vertex outside the connected component, "fault-free," neither v_0 nor v_1 is in any faction.

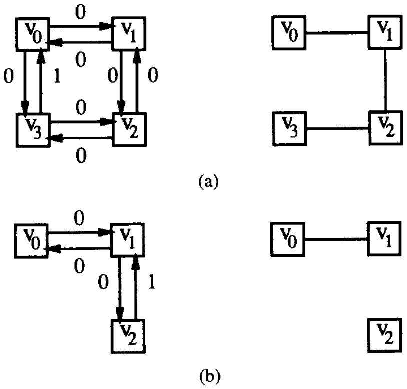


Fig. 1. Examples of violation.

However, the above procedure can be carried out in a more efficient way as shown below:

```
Algorithm 3.2
Input: Test digraph $D(V, A)$, syndrome $w$ and threshold value $k$.
Output: Set $R$ of vertices declared fault-free.
Step 1: Set $R = \emptyset$ and set status $(v) = 2$ for every vertex $v \in V$.
Step 2: If there is a vertex $v_i \in V$ with status $(v_i) = 2$, then do the following:
    1) Set $T = 0$, $count = 1$, $status(v_i) = 3$ and $LIST = \{v_i\}$.
    2) Call the procedure $FACTION(v_i)$, which is defined below.
    3) Set $R = R \cup List$ and $status\ v = 0$ for every vertex $v$ on $LIST$ if $T = 0$ and $count > k$; otherwise, set status $(v) = 1$ for every vertex $v$ on $LIST$.
Procedure FACTION $(v_i)$
begin
For every vertex $v_j$ adjacent to $v_i$ with respect to $D(V, A)$, do the following:
    1) If $status\ (v_j) = 2, w(v_i, v_j) = 0$ and $w(v_j, v_i) = 0$, set $LIST = LIST \cup \{v_j\}$, $status\ (v_j) = 3$ and $count = count + 1$ and call $FACTION\ (v_j)$.
    2) If $w(v_i, v_j) = 0$ and $w(v_j, v_i) = 1$, set $T = 1$.
    3) If $status\ (v_j) = 3$ and either or both of $w(v_i, v_j)$ and $w(v_j, v_i)$ are equal to 1, set $T = 1$.
end
```

Theorem 3.1: A vertex v is placed in R by Algorithm 3.2 if and only if v is in a faction of size larger than k. The complexity of the algorithm is $O(\max(|V|, |A|))$ for general digraphs and $O(|V|)$ for constant degree digraphs.

Proof: It is easy to see from the algorithm that a set of vertices is placed on the same *LIST* if and only if they are in the same connected component of the agreement graph. Steps 2 and 3 of the procedure *FACTION* eliminate the two possible violations mentioned earlier. Regarding the complexity of the algorithm, note that each unit is visited exactly once and each arc is visited twice, once from each direction. The complexity is therefore $O(\max(|V|, |A|))$. For constant degree digraphs, since $O(|V|) = O(|A|)$, the complexity is $O(|V|)$. Q.E.D.

IV. Performance Analysis

In this section the performance of Algorithm 3.1 on constant degree structures will be analyzed. In particular, the rectangular and octagonal grids will be considered. We will show that the ratio of the expected number of correctly identified fault-free (faulty) vertices to the expected number of fault-free (faulty) vertices is very close to 1 for reasonable values of p, γ and a chosen threshold value k. We will also show that these ratios increase monotonically with the decrease of p and the decrease of γ. As p and γ cannot be accurately predicted, this feature allows one to get lower bounds on these ratios as long as one can get upper bounds on p and γ. The variance of the number of fault-free (faulty) units correctly identified will also be discussed.

For an easier and uniform analysis, we assume that the test digraph $D(V, A)$ is a torus connected grid. One of the nice features of the torus connected grids is that every vertex is in an identical position as far as the connection topology is concerned. Any contiguous portion of a torus connected grid has a natural planar embedding as a nontorus connected grid. A rectangular grid is shown in Fig. 2. For any subset of vertices $V' \subseteq V$, if the V'-induced subgraph $D[V']$ is strongly connected, then $D[V']$ is called a *patch* and V' is said to form the patch. Visually, a patch is a region of the plane in which the grid is embedded and therefore has a geometric shape. For the convenience of discussion, if V' forms a patch of shape S, then V' is also said to have shape S. The orientation of a patch is also important. Rotating the template of a patch by $90°, 180°$ or $270°$ produces a shape *symmetric* to the old one. For a rectangular grid, there are six different patch shapes for patches containing three vertices and they are listed in Fig. 3. Shapes (a) and (d) are symmetric. The other four shapes are also symmetric to each other. Moving the template of a patch horizontally and/or vertically to cover some other vertices produces a different patch of the same shape. For instance, in Fig. 2, the subgraphs induced by vertex sets $\{v_8, v_{12}, v_{13}\}$ and $\{v_{13}, v_{17}, v_{18}\}$ are two different patches. There are s patches of size s that a vertex can be in for any specific patch shape (this is true only when the grid is big enough). For example, in Fig. 2, vertex v_{13} can be in any of the three patches of shape (b) (Fig. 3), whose size is three, formed by $\{v_8, v_{12}, v_{13}\}, \{v_{13}, v_{17}, v_{18}\}$ and $\{v_9, v_{13}, v_{14}\}$. If there are m different patch shapes of size s, then the number of different patches of size s that a vertex can be in is m times s (this is true only when the grid is big enough). For example, in Fig. 2, the number of patches of size three that vertex v_{13} can be in is 18, since there are six patch shapes of size three.

We next introduce the concept of an *isolated set,* which is useful in the performance analysis of our diagnosis algorithm. An isolated set is a maximal size cluster of vertices that evaluate each other to be fault-free. This concept is formally defined as follows.

Definition 4.1: A subset of vertices $V' \subseteq V$ forms an isolated set if the following holds:

1) $D[V']$ is strongly connected, where $D[V']$ is a subgraph of $D(V, A)$ induced by V'.
2) Every arc in $A[V']$ has the weight of 0;

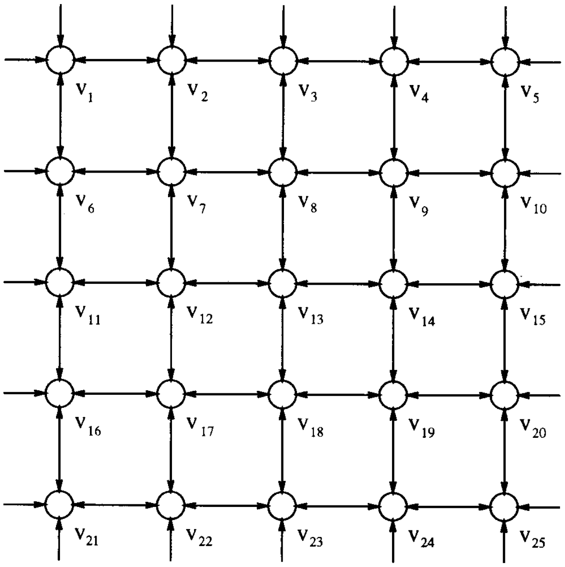


Fig. 2. A rectangular grid.

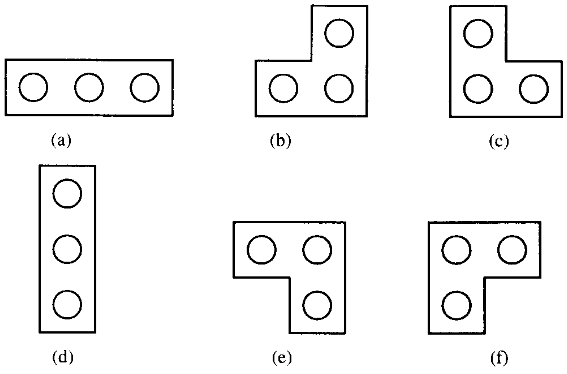


Fig. 3. Patch shapes of size three.

3) For every vertex $u \in V'$ and every vertex $v \in N(u) - V'$, either (u, v), or (v, u), or both have the weight of 1.

An isolated set may possibly be a faction but is not necessarily one while a faction is always an isolated set. The difference is illustrated in Fig. 4. Vertices u and v call each other fault-free and for each of the other vertices either it calls one of u and v faulty or it is called faulty by one of u and v. Therefore, u and v form an isolated set. However, they do not form a faction because u calls w, an outsider, fault-free.

The probability that a fault-free vertex v is in a faction of patch shape S is denoted by $q_f(v, S)$. The probability that a fault-free vertex v is in a faction of size no larger than k is denoted by $q_F(v, k)$. The probability that a faulty vertex v is in an isolated set of patch shape S is denoted by $q_i(v, S)$ and the probability that v is in an isolated set of size no larger than k is denoted by $q_I(v, k)$. The probability that a fault-free vertex v is correctly identified is denoted by $P_G(v)$ and the probability that a faulty vertex v is correctly identified is denoted by $P_B(v)$. In the following, $\mathcal{P}_S$ represents one of the patches of shape S that contain v; s is the number of vertices in any patch of shape S; and $V(\mathcal{P}_S)$ represents its vertex set.

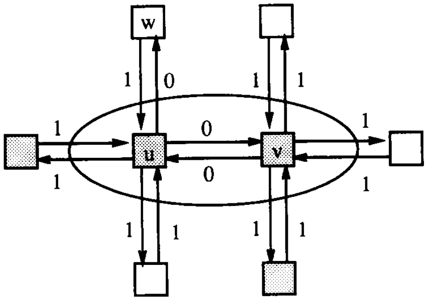


Fig. 4. An example of a nonfaction isolated set.

Lemma 4.1: For a fault-free vertex v and a patch shape S, the following holds:

$$q_f(v,S) = s(1-p)^{s-1}p^{|N(V(\mathcal{P}_S))|}. \tag{1}$$

For a fault-free vertex v and threshold value k, the following holds

$$q_F(v,k) = \sum_{S,s\leq k} q_f(v,S). \tag{2}$$

Proof: As $V(\mathcal{P}_S)$ contains v and v is fault-free, all members of $V(\mathcal{P}_S)$ must be fault-free for $V(\mathcal{P}_S)$ to form a faction. Furthermore, all vertices in $N(V(\mathcal{P}_S))$ must be faulty. The event that $V(\mathcal{P}_S)$ forms a faction implies that $s-1$ more specific vertices are fault-free and that $|N(V(\mathcal{P}_S))|$ specific vertices are faulty. As every vertex happens to be faulty with the probability of p, the probability that the above event happens is $(1-p)^{s-1}p^{|N(V(\mathcal{P}_S))|}$. Because there are s patches of shape S that v can be in and because the events that these patches form factions are pairwise disjoint (otherwise we can arrive at a contradiction that a vertex is both faulty and fault-free), (1) follows. Equation 2 follows from the fact that the events that v is in factions of different patch shapes are exclusive. Q.E.D.

Lemma 4.2: For a faulty vertex v and a patch shape S, the following holds

$$q_i(v,S) = sp^{s-1}\gamma^{|A[V(\mathcal{P}_S)]|} \prod_{u\in N(V(\mathcal{P}_S))} (1-p+p(1-\gamma^2)^{\delta(u,V(\mathcal{P}_S))}). \tag{3}$$

For a faulty vertex v and threshold value k, the following holds

$$q_I(v,k) = \sum_{S,s\leq k} q_i(v,S). \tag{4}$$

Proof: $V(\mathcal{P}_S)$ forms an isolated set only if all arcs in $A[V(\mathcal{P}_S)]$ assume the weight of 0 and at least one of the arcs (u,w) and (w,u) assumes the weight of 1 for every vertex $u \in N(V(\mathcal{P}_S))$ and each of its neighbors $w \in V(\mathcal{P}_S)$. As v is faulty, every vertex in $V(\mathcal{P}_S)$ must be faulty for $V(\mathcal{P}_S)$ to be an isolated set. This means that $s-1$ other vertices are faulty and it happens with the probability of p^{s-1}. By our model, the probability that all arcs in $A[V(\mathcal{P}_S)]$ assume the weight of 0 is $\gamma^{|A[V(\mathcal{P}_S)]|}$. If u is fault-free, u will always evaluate its neighbors in $V(\mathcal{P}_S)$ to be faulty. If u is faulty, the probability that at least one of u and w evaluates the other to be faulty is $1-\gamma^2$. The probability that at least one arc between u and each of its neighbors in $V(\mathcal{P}_S)$ assumes the weight of 1 is therefore $1-p+p(1-\gamma^2)^{\delta(u,V(\mathcal{P}_S))}$. Note that there are s patches of shape S that v can be in and that the events that these s patches form isolated sets are pairwise disjoint. Combine all these conditions together, we have (3). Equation 4 follows from the fact that isolated sets of different patch shapes are exclusive. This completes the proof. Q.E.D.

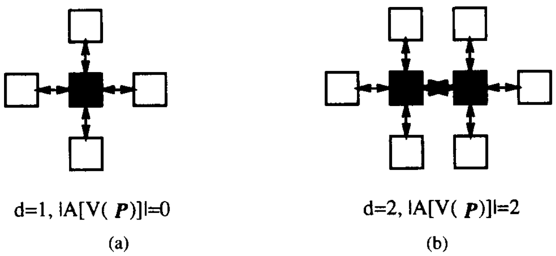
Theorem 4.1: Consider all vertices in R produced by Algorithm 3.1 to be fault-free and all vertices in $V-R$ to be faulty. The probability of correct identification $P_G(v)$ for every fault-free vertex v is $1-q_F(v,k)$ and the probability of correct identification $P_B(v)$ for every faulty vertex v is greater than $q_I(v,k)$.

Proof: As was observed earlier, a fault-free vertex is always in a faction, no matter what its size may be. If a fault-free vertex is not in a faction of size no larger than k, then it is in a faction of size larger than k. Since a fault-free vertex v is identified as fault-free by Algorithm 3.1 if and only if v is in a faction of size larger than k and since the probability that v is in a faction of size no larger than k is $q_F(v,k)$, the probability that v is correctly identified is $1-q_F(v,k)$. If a faulty vertex v is in an isolated set of size no larger than k, then it cannot be in a faction of size larger than k and therefore it is correctly identified. Note that even if v is not in an isolated set of size no larger than k, it is still possible that v is not in a faction of size larger than k and therefore it will not be identified to be fault-free. The probability that v is not in an isolated set of size no larger than k nor in a faction of size larger than k is obviously greater than 0. Hence the probability of correct identification for a faulty vertex v is larger than $q_I(v,k)$. Q.E.D.

From Theorem 4.1, the probability of correct identification of a vertex v depends on the value of k. There is a conflict on the selection of the value of k. In order to get a higher probability of correct identification of a fault-free vertex, a smaller threshold value k is needed while a larger threshold value k is needed for a higher probability of correct identification of a faulty vertex. This means a good balance is needed in choosing the threshold value k. Other factors that should be considered in choosing the value of k are the probability of failure p and the probability γ of a faulty processor claiming another faulty processor to be fault-free. Fortunately, we will see that with $k=2$ Algorithm 3.1 is suited to the rectangular and octagonal grids for wide ranges of values of p and γ. With such a value of k, both fault-free and faulty vertices can be correctly identified with high probabilities.

To calculate $q_F(v,k)$ and $q_I(v,k)$ for rectangular and octagonal grids, we need to enumerate all patch shapes of size no larger than k. For the convenience of enumerating, we consider all patch shapes that are symmetric to be an equivalence class and take any one representative from this class and call it a *basic patch shape.* In calculating $q_F(v,k)$ and $q_I(v,k)$, we take into account the number of occurrences of a basic patch shape—the number of patch shapes in the corresponding equivalence class.

Figs. 5 and 6 are complete lists of basic patch shapes of sizes no larger than 2 on a rectangular grid and on an octagonal grid, respectively, with the number of symmetric patch shapes



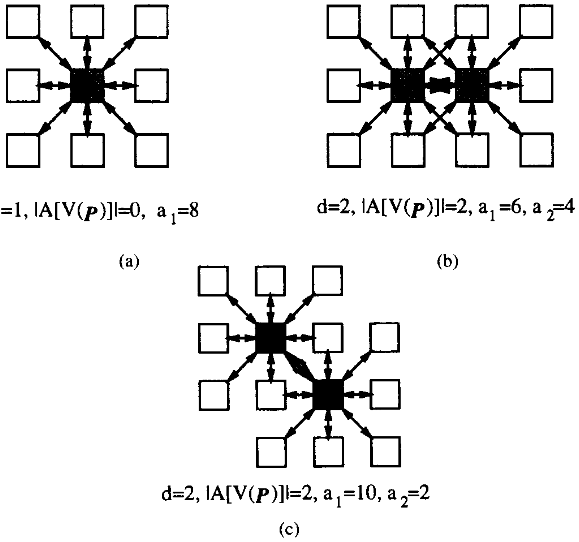
$d=1, |A[V(\mathcal{P})]|=0$

(a)

$d=2, |A[V(\mathcal{P})]|=2$

(b)

Fig. 5. Patch shapes of sizes no larger than 2 on a rectangular grid.



$d=1, |A[V(\mathcal{P})]|=0, a_1=8$

(a)

$d=2, |A[V(\mathcal{P})]|=2, a_1=6, a_2=4$

(b)

$d=2, |A[V(\mathcal{P})]|=2, a_1=10, a_2=2$

(c)

Fig. 6. Patch shapes of sizes no larger than 2 on an octagonal grid.

d and the number of internal arcs $|A[V(\mathcal{P})]|$ indicated. For the octagonal grid, another set of parameters a_1 and a_2 is also given in the figure, where a_1 is the number of vertices in $N(V(\mathcal{P}))$ adjacent to exactly one vertex in $V(\mathcal{P})$ and a_2 is the number of vertices in $N(V(\mathcal{P}))$ adjacent to exactly two vertices in $V(\mathcal{P})$. The vertices in a patch $\mathcal{P}$ are shaded, and internal arcs are represented by thick lines. From these figures, we can deduce analytic expressions for $q_F(v,2)$ and $q_I(v,2)$. In what follows, we assume that a grid has at least four rows and at least four columns.

Lemma 4.3: For a vertex v on a torus connected rectangular grid, the following holds:

1) $q_F(v,2) = p^4 + 4(1-p)p^6$.
2) $q_I(v,2) = (1-p\gamma^2)^4 + 4(1-p\gamma^2)^6 p\gamma^2$.

Proof: Consider Fig. 5. For patch shape (a) we have $s = 1$ and $|N(V(\mathcal{P}))| = 4$. The number of symmetric shapes is 1. By Lemma 4.1, we have a corresponding term of p^4 in $q_F(v,2)$. For patch shape (b), we have $s = 2$ and $|N(V(\mathcal{P}))| = 6$. The number of symmetric shapes is 2. This brings about a term of $2 \cdot 2(1-p)p^6$. This completes the proof of condition 1.

Note that in both patch shapes, $\delta(u, V(\mathcal{P})) = 1$ for every vertex $u \in N(V(\mathcal{P}))$. Also note that $|N(V(\mathcal{P}))| = 4$ and $|A[V(\mathcal{P})]| = 0$ for shape (a) and $|N(V(\mathcal{P}))| = 6$ and $|A[V(\mathcal{P})]| = 2$ for shape (b). Applying Lemma 4.2, we have condition 2. Q.E.D.

Lemma 4.4: For a vertex v on a torus connected octagonal grid, the following holds:

1) $q_F(v,2) = p^8 + 4(1-p)p^{10} + 4(1-p)p^{12}$.
2) $q_I(v,2) = (1-p\gamma^2)^8 + 4(1-p\gamma^2)^6(1-p+p(1-\gamma^2)^2)^4 p\gamma^2 + 4(1-p\gamma^2)^{10}(1-p+p(1-\gamma^2)^2)^2 p\gamma^2$.

Proof: It can be proved in a way similar to the proof of Lemma 4.3. One only needs to note that here $\delta(u, V(\mathcal{P}))$ varies from vertex to vertex. There are a_1 vertices u in $N(V(\mathcal{P}))$ with $\delta(u, V(\mathcal{P})) = 1$ and a_2 vertices u in $N(V(\mathcal{P}))$ with $\delta(u, V(\mathcal{P})) = 2$. Q.E.D.

So far, we have discussed the probability of correctly identifying each individual vertex. A more useful criterion for assessing an algorithm is the percentage of the vertices that are correctly identified by the algorithm. This translates to finding the ratio of the expected number of correctly identified fault-free (faulty) vertices to the expected number of fault-free (faulty) vertices in the whole system. An interesting result is, as one will see in the following, that these ratios are identical to the probability of correctly identifying a fault-free vertex and the probability of correctly identifying a faulty vertex, respectively. Let g and f be the number of fault-free vertices and the number of faulty vertices, respectively. Let g_c and f_c be the number of correctly identified faulty-free vertices and the number of correctly identified faulty vertices. These numbers are random variables. We use $E[h]$ to represent the expectation of a random variable h.

Lemma 4.5: For a torus connected grid, the following holds:

1)

$$\frac{E[g_c]}{E[g]} = P_G(v).$$

2)

$$\frac{E[f_c]}{E[f]} = P_B(v).$$

where v is a vertex on the grid.

Proof: Let X_i be a random variable such that $X_i = 1$ if vertex v_i is fault-free and correctly identified, and $X_i = 0$ otherwise. Obviously, $g_c = \Sigma_i X_i$ and therefore $E[g_c] = \Sigma_i E[X_i]$. Note that $E[X_i] = (1-p)P_G(v_i)$. As $P_G(v)$ assumes the same value for every vertex, $E[g_c] = n(1-p)P_G(v)$, where n is the number of vertices in the system. Since $E[g] = n(1-p)$, we have the first equation of Lemma 4.5. The proof of the second equation is similar and hence omitted. Q.E.D.

Using the equations of Lemma 4.3 and Lemma 4.4, we calculated the probabilities of correct identification $P_G(v)$ and $P_B(v)$ (using $q_I(v,2)$ as an approximation) for a wide variety of values of p and γ. The results are listed in Tables I and II for the rectangular grid and octagonal grid, respectively. We can see from these tables that both fault-free and faulty vertices can be correctly identified with high probability in a wide variety of situations. By Lemma 4.5, Tables I and II also represent the ratios $E[g_c]/E[g]$ and $E[f_c]/E[f]$.

Lemma 4.5 shows the fraction of fault-free (faulty) vertices that are correctly identified on the average. It is not clear yet

TABLE I
PERFORMANCE OF THE ALGORITHM ON A RECTANGULAR GRID WITH $k = 2$

p	$q_I(v,2) < P_B(v)$					$P_G(v)$
	γ=0.05	γ=0.1	γ=0.2	γ=0.3	γ=0.4	
0.1	0.999999	0.99998	0.9997	0.999	0.996	0.9999
0.2	0.999996	0.99993	0.999	0.994	0.983	0.998
0.3	0.99999	0.9998	0.998	0.988	0.964	0.990
0.4	0.99998	0.9997	0.996	0.979	0.940	0.965
0.5	0.99997	0.9995	0.993	0.968	0.910	0.906
0.6	0.99996	0.9994	0.990	0.956	0.877	0.796

TABLE II
PERFORMANCE OF THE ALGORITHM ON AN OCTAGONAL GRID WITH $k = 2$

p	$q_I(v,2) < P_B(v)$					$P_G(v)$
	γ=0.05	γ=0.1	γ=0.2	γ=0.3	γ=0.4	
0.1	0.99999	0.9999	0.999	0.994	0.982	0.99999999
0.2	0.99998	0.9997	0.995	0.977	0.935	0.999997
0.3	0.99995	0.9993	0.989	0.951	0.871	0.9999
0.4	0.99992	0.999	0.981	0.919	0.796	0.999
0.5	0.9999	0.998	0.971	0.882	0.717	0.993
0.6	0.9998	0.997	0.960	0.841	0.637	0,970

how far the performance of each instance of diagnosis may deviate from the expectation. The following theorem gives lower bounds on the probability that the performance is within a range of the expectation. In the following, $P[\langle predicate \rangle]$ represents the probability that $\langle predicate \rangle$ holds, $\mathrm{Var}\,[X]$ the variance of a random variable X and $\mathrm{Cov}\,[X_i, X_j]$ the covariance of random variables X_i and X_j.

Theorem 4.2: For a torus connected rectangular grid, the threshold $k = 2$ and a constant $t, 0 \leq t < 1$, the following holds:

1)

$$P\left[\frac{g_c}{E[g]} > tP_G]\right] \geq 1 - \frac{41(1-(1-p)P_G)}{41(1-(1-p)P_G) + n(1-p)P_G(1-t)^2}.$$

2)

$$P\left[\frac{f_c}{E[f]} > tP_B\right] \geq 1 - \frac{41(1-pP_B)}{41(1-pP_B) + npP_B(1-t)^2}.$$

Proof: It is easy to see that

$$P\left[\frac{g_c}{E[g]} > tP_G\right] = P[g_c > tP_G E[g]] = P[g_c > tE[g_c]].$$

By the *one-sided inequality* [18], we have

$$P[g_c > tE[g_c]] \geq 1 - \frac{\mathrm{Var}\,[g_c]}{\mathrm{Var}\,[g_c] + (E[g_c])^2(1-t)^2}. \quad (5)$$

Let X_i be a random variable such that $X_i = 1$ if vertex v_i is fault-free and correctly identified and $X_i = 0$, otherwise.

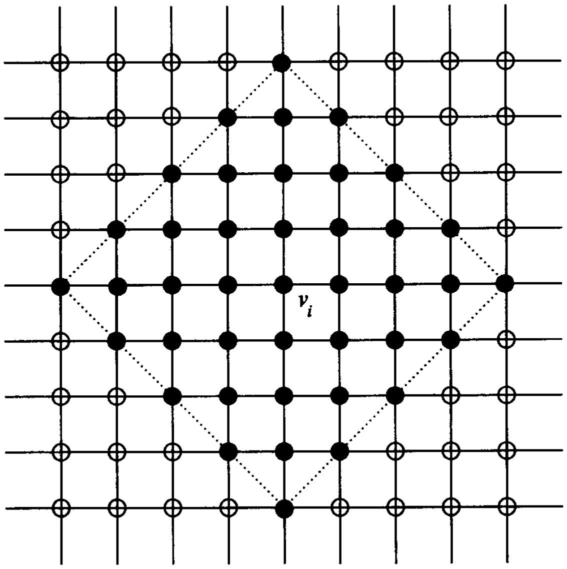


Fig. 7. Correlated region.

It is clear that

$$\mathrm{Var}\,[g_c] = \sum_{i=1}^{n} \mathrm{Var}[X_i] + \sum_{i,j=1,i\neq j}^{n} \mathrm{Cov}\,[X_i, X_j].$$

The number of edges on a shortest path from vertex v_i to vertex v_j is called the *distance* between v_i and v_j. As can be seen from the proof of Lemma 4.3, the event that a fault-free vertex v_i is correctly identified is independent of the event that a fault-free vertex v_j is correctly identified if the distance between v_i and v_j is greater than 4. By hypothesis, v_i and v_j occur to be fault-free independently. Therefore, $\mathrm{Cov}\,[X_i, X_j] = 0$ for any two vertices v_i and v_j whose distance is greater than 4. We only need to concentrate on those vertices v_j's whose distances from v_i are less than or equal to 4 in counting $\mathrm{Cov}\,[v_i, v_j]$ for vertex v_i. These v_j's are on a diamond of diameter 8 centered at v_i, as shown in Fig. 7. As we can see, there are altogether 40 of them. Note that $(\mathrm{Cov}\,[X_i, X_j])^2 \leq \mathrm{Var}\,[X_i]\,\mathrm{Var}\,[X_j]$ and that $\mathrm{Var}\,[X_i] = \mathrm{Var}\,[X_j]$. We have $\mathrm{Var}[g_c] \leq 41n\,\mathrm{Var}\,[X_i]$. It is obvious that $\mathrm{Var}\,[X_i] = (1-p)P_G(1-(1-p)P_G)$. Therefore

$$\mathrm{Var}\,[g_c] \leq 41n(1-p)P_G(1-(1-p)P_G).$$

Apply the above inequality to (5), condition 1 of the theorem follows. The proof of condition 2 is similar and hence omitted. Q.E.D.

Similar results can also be deduced for the octagonal grids. As an immediate implication of Theorem 4.2, we have the following corollary:

Corollary 4.1: For a torus connected grid, $k = 2, 0 \leq t < 1$ and $0 < p < 1$, the following holds

1) For any $\epsilon, 0 \leq \epsilon < 1$, there is an N such that when $n \geq N$

$$P\left[\frac{g_c}{E[g]} > tP_G\right] \geq \epsilon.$$

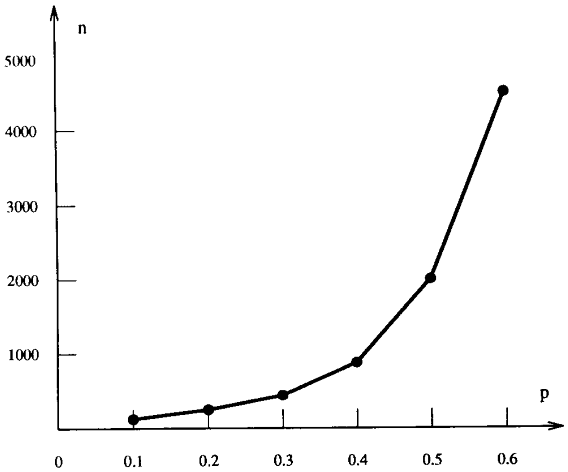


Fig. 8. Number of units n such that $P[g_c/E[g] > 0.9P_G] \geq 0.9$.

2) For any $\epsilon, 0 \leq \epsilon < 1$, there is an N such that when $n \geq N$

$$P\left[\frac{f_c}{E[f]} > tP_B\right] \geq \epsilon.$$

Fig. 8 shows how the number of vertices n varies with the probability of failure p such that $P[(g_c/E[g]) > 0.9P_G] \geq 0.9$. The results were obtained by computing the exact value of $\mathrm{Var}\,[g_c]$. It is done by exhausting the 40 correlation cases mentioned in the proof of Theorem 4.2. As can be seen from the figure, for a moderately large size, $g_c/E[g]$ falls into a small range of its expectation with high probability.

In practice, the parameters p and γ cannot be accurately predicted. Now let us consider how the deviations of the values of these parameters affect the probabilities P_G and P_B predetermined with predicted p and γ values. As stated in the following theorem, one can get lower bounds on the actual values of P_G and P_B as long as one can get upper bounds on p and γ.

Theorem 4.3: For a torus connected rectangular or octagonal grid and the threshold $k = 2, P'_G \geq 1 - q''_F$ as long as $p' \leq p''$ and $P'_B > q''_I$ as long as $p' \leq p''$ and $\gamma' \leq \gamma''$, where P'_G and P'_B are the probabilities of correct identification determined with $p = p'$ and $\gamma = \gamma'$ while q''_F and q''_I are determined with $p = p''$ and $\gamma = \gamma''$.

Proof: It suffices to show that q_F increases monotonically with the increase of p and that q_I decreases monotonically with the increase of p and the increase of γ. We only consider the rectangular grids to save space. From Lemma 4.3, we know that $q_F = p^4 + 4(1-p)p^6$. Let us compute the derivative

$$\begin{aligned}\frac{d(q_F)}{d(p)} &= 4p^3 + 24p^5 - 28p^6\\ &= 4p^3(1 + 6p^2 - 7p^3)\\ &= 4p^3\{(1-p^2) + (7p^2 - 7p^3)\}.\end{aligned}$$

Since $0 \leq p \leq 1, (d(q_F)/d(p)) \geq 0$, which implies that q_F increases monotonically with the increase of p. Also from Lemma 4.3 we have that $q_I = (1-p\gamma^2)^4 + 4(1-p\gamma^2)^6 p\gamma^2$. Substitute β for $p\gamma^2$. We have $q_I = (1-\beta)^4 + 4(1-\beta)^6\beta$. Let us take the derivative

$$\begin{aligned}\frac{d(q_I)}{d(\beta)} &= -4(1-\beta)^3 + (4(1-\beta)^6 - 24\beta(1-\beta)^5) &(6)\\ &= -4\beta(1-\beta)^3\{(3-\tfrac{5}{2}\beta)^2 + \tfrac{3}{4}\beta^2\}. &(7)\end{aligned}$$

As p and γ both range from 0 to 1, β also ranges from 0 to 1. We can see from the above that $d(q_I)/d(\beta) \leq 0$. Now let us take the original partial derivatives

$$\begin{aligned}\frac{\partial(q_I)}{\partial(p)} &= \frac{d(q_I)}{d(\beta)} \cdot \frac{\partial(\beta)}{\partial(p)} = \gamma^2\frac{d(q_I)}{d(\beta)} \leq 0\\ \frac{\partial(q_I)}{\partial(\gamma)} &= \frac{d(q_I)}{d(\beta)} \cdot \frac{\partial(\beta)}{\partial(\gamma)} = 2p\gamma\frac{d(q_I)}{d(\beta)} \leq 0.\end{aligned}$$

This completes the proof. Q.E.D.

V. Application to VLSI Circuit Testing

In this section we will show how our diagnosis algorithm can be applied in production testing of VLSI chips. The chips to be tested can be arbitrary digital circuits, not necessarily "processors." We assume that the chips to be tested are identical. We are not concerned with the generation of test data. We assume that for at least one test vector a bad chip will produce a response different from that produced by a good chip. If some faults are not detectable, then they cannot be detected by any other method with the same set of test data. With this assumption, the PMC model is justified.

To perform interunit tests we have a test board as shown in Fig. 9. There are four comparators for each chip socket, each corresponding to one of its neighbors. The connection topology is a torus-connected rectangular grid. Test data is broadcast to all chips and each chip (its comparators) compares its results with those from its neighbors. A chip (its comparator) evaluates a neighbor fault-free if and only if the responses are identical for all test vectors.

The outcomes of the comparisons can be reported to a host system through the output buses as shown in Fig. 10. Each column of chips shares a common output bus. The latches in the four comparators for each chip, where the comparison outcomes are held, are connected into a shift register for outcome reporting. Each register has an output strobe pin. The strobe pins are connected together for each row of the grid. At any one time during outcome reporting exactly one row of registers is selected. The selection can be done through the shift register as shown at the left edge of the figure. If the number of output buses is more than that the host system can accommodate, they can be grouped and each group be equipped with a shift register.

With the above arrangement test data can be broadcast to the chips to be tested and the outcomes of the comparison tests be reported to the host system. Our diagnosis algorithm can then be run on the host system to identify the status of the

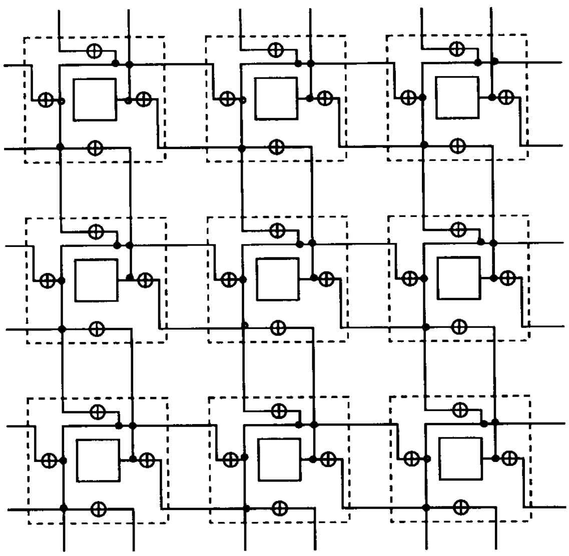


Fig. 9. Test board schematic.

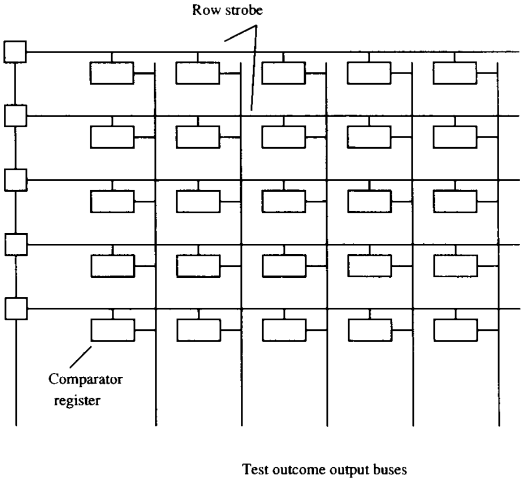


Fig. 10. Test outcome reporting circuitry.

chips. The chips that are declared fault-free are retained and those that are declared faulty are discarded.

The chip testing scheme described above has the following characteristics.

1) The chips to be tested can be arbitrary digital circuits, not necessarily "processors".
2) There is no need for storing the correct test responses anywhere nor do we even need to know what the correct responses are.
3) The massive transmission of test responses (chip responses to test vectors) is limited to local neighborhoods and is carried out in parallel. As only the comparison outcomes (four bits of data per unit) need to be transmitted to the host, the demand on communication bandwidth is low.

Now let us take an example to see the performance of this scheme. Assume that the failure rate of the given batch of chips to be tested is $p = 0.1$ (we also consider this to be the probability that a chip is faulty). We also assume that the probability that a faulty chip calls another faulty chip "fault-free" is 0.1; i.e., $\gamma = 0.1$. After one application of the procedure the number of fault-free chips correctly identified (in the sense of expectation, the same for the following) is $n(1-p)P_G$ and the number of faulty units wrongly identified as fault-free is $np(1 - P_B)$, where n is the total number of chips in the batch. The failure rate of the set of chips that passed the screening procedure is therefore

$$p' = \frac{np(1-P_B)}{np(1-P_B)+n(1-p)P_G} = \frac{p(1-P_B)}{p(1-P_B)+(1-p)P_G}.$$

Using the formulae of last section to calculate P_G and P_B we get from the above equation that $p' = 1.99 \times 10^{-6}$. If this failure rate is not low enough, the test-diagnosis screening procedure can be repeated once more and we can get a new failure rate of passed chips of $p'' = 1 \times 10^{-20}$. This is obviously an extremely low failure rate and acceptable in almost all real situations. Assume that the total number of chips in the batch is 10^6. The number of faulty chips that passed the screening is virtually zero. The expense is throwing away only about 0.01% of good chips. The saving of test time is obviously significant, and the extent of the saving depends on the size of the test board.

VI. Conclusion

We have presented a simple and efficient diagnosis algorithm for constant degree structures. The algorithm is shown to be able to correctly identify almost all units even when the probability of failure is rather high. The application of the algorithm to the production testing of VLSI chips is also presented. With a very simple test board, a large number of chips can be tested in parallel and the actual status of the chips can be determined using our diagnosis algorithm. This screening procedure can be repeated for a higher accuracy of diagnosis. It is shown that an extremely high accuracy can be achieved with only two repetitions. The saving of test time and the test equipment cost could be significant with this approach.

References

[1] F. P. Preparata, G. Metze, and R. T. Chien, "On the connection assignment problem of diagnosable systems," *IEEE Trans. Electron. Comput.*, vol. EC-16, pp. 848–854, Dec. 1967.

[2] A. K. Somani, V. K. Agarwal, and D. Avis, "A generalized theory for system-level diagnosis," *IEEE Trans. Comput.*, vol. C-36, pp. 538–546, May 1987.

[3] E. R. Scheinerman, "Almost sure fault tolerance in random graphs," *SIAM J. Comput.*, vol. 16, pp. 1124–1134, 1987.

[4] D. M. Blough, *Fault Detection and Diagnosis in Multiprocessor Systems*, Ph.D. dissertation, The Johns Hopkins Univ., Baltimore, MD, 1988.

[5] D. M. Blough, G. F. Sullivan, and G. M. Masson, "Almost certain diagnosis for sparsely interconnected faulty systems," in *Digest FTCS-18*, pp. 260–271, June 1988.

[6] D. M. Blough, G. F. Sullivan, and G. M. Masson, "Fault diagnosis for sparsely interconnected multiprocessor systems," in *Digest FTCS-19*, pp. 62–69, 1989.

[7] D. M. Blough and A. Pelc, "Reliable diagnosis and repair in constant-degree multiprocessor systems," in *Digest FTCS-20*, pp. 316–323, June 1990.

[8] D. Fussell and S. Rangarajan, "Probabilistic diagnosis of multiprocessor systems with arbitrary connectivity," in *Digest FTCS-19,* pp. 560–565, 1989.

[9] S. Rangarajan and D. Fussell, "Probabilistic diagnosis algorithm tailored to system topology," in *Digest FTCS-21,* pp. 230–237, 1991.

[10] A. T. Dahbura, K. K. Sabnani, and L. L. King, "The comparison approach to multiprocessor fault diagnosis," *IEEE Trans. Comput.*, vol. C-36, pp. 373–378, Mar. 1987.

[11] S. Lee and K. G. Shin, "Optimal multiple syndrome probabilistic diagnosis," in *Digest FTCS-20,* June 1990, pp. 324–330.

[12] P. Berman and A. Pelc, "Distributed probabilistic fault diagnosis for multiprocessor systems," in *Digest FTCS-20,* June 1990, pp. 340–346.

[13] S. Rangarajan, D. Fussell, and M. Malek, "Built-in testing of integrated circuit wafers," *IEEE Trans. Comput.*, vol. 39, pp. 195–205, Feb. 1990.

[14] L. LaForge, K. Huang, and V. K. Agarwal, "Almost sure diagnosis of almost every good element," *IEEE Trans. Comput.*, vol. C-43, pp. 295–305, Mar. 1994.

[15] E. Eichelberger and T. Williams, "A logic structure for LSI testability," in *Proc. Design Automat. Conf.,* 1977, pp. 462–468.

[16] B. Nadeau-Dostie, P. S. Wilcox, and V. K. Agarwal, "A scan-based BIST technique using pair-wise compare of identical components," in *Proc. 4th CSI/IEEE Int. Symp. VLSI Design,* Jan. 1991.

[17] F. Barsi, F. Grandoni, and P. Maestrini, "A theory of diagnosability of digital systems," *IEEE Trans. Comput.*, vol. C-25, pp. 585–593, June 1976.

[18] A. O. Allen, *Probability, Statistics, and Queueing Theory with Computer Science Applications.* New York: Academic, 1978.

Kaiyuan Huang received the Bachelor's degree in computer engineering and the Master's degree in computer science from Chongqing University, Chongqing, China. He received the Ph.D. degree in electrical engineering from McGill University, Montreal, P. Q., Canada.

He was on the faculty of Chongqing University, then a Visiting Scientist with the VLSI Design Lab, McGill University, and was a Postdoctoral Fellow in the Department of Electrical and Computer Engineering, Concordia University, Montreal, P. Q., Canada. He is now with the Scientific Staff of Bell-Northern Research, Ottawa, Ont., Canada.

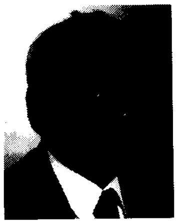
Vinod K. Agarwal (S'74–M'77–SM'84–F'92) received the Ph.D. degree from Johns Hopkins University, Baltimore, MD.

He is President of LVS Software, Inc. and on a leave of absence from the NSERC—NT/BNR Industrial Research Chair at McGill University. He has authored or coauthored over 100 papers in international journals and conferences and presented numerous invited seminars in the United States, Germany, Japan, France, and The Netherlands. He is a coinventor with BNR researchers of several U.S. patents on embedded memory BIST and board-level BIST.

Dr. Agarwal was General Chair of the IEEE International Symposium on Fault Tolerant Computing (Montreal, P. Q. , Canada, 1991). He has served on the technical committees of many IEEE conferences, including FTCS, ICCAD, and ICCD. He is also a founding member of IFIP Working Group 10.4 on Dependable Computing.

Laurence LaForge received the B.S. degree in mathematics from the Massachusetts Institute of Technology, Cambridge. In 1991 he received the Ph.D. degree in computer science from McGill University, Montreal, P. Q., Canada, specializing in architectures for diagnosis and configuration.

He is one of two International Game Technology Professors of Electrical and Computer Engineering at the University of Nevada, Reno.



K. Thulasiraman (M'72–SM'84–F'90) received the Bachelor's and Master's degrees from the University of Madras, Madras, India, in 1963 and 1965, respectively, and the Ph.D. degree from the Indian Institute of Technology, Madras, in 1968, all in electrical engineering.

He joined the Indian Institute of Technology in 1965, where he was with the Department of Electrical Engineering (1965–1973) and the Department of Computer Science (1973–1981). He was promoted to the rank of Professor in January 1977. After serving for a year (1981–1982) in the Department of Electrical Engineering, Technical University of Nova Scotia, Halifax, Canada, he joined· Concordia University, Montreal, P. Q., Canada, as Professor in the Department of Mechanical Engineering, where he was involved in the development of programs in industrial engineering at the undergraduate and graduate levels. In 1984 he transferred to the Department of Electrical and Computer Engineering and from 1993–1994, he served as Chair. In August 1994 he moved to the University of Oklahoma at Norman as Hitachi Chair in Computer Science. He visited Concordia University during the periods of 1970–1972, 1975–1976, and 1979. He also visited the Tokyo Institute of Technology, Tokyo, Japan (March–July 1988), supported by a senior fellowship awarded by the Japan Society for Promotion of Science, and the University of Karlsruhe, Germany (September–December 1990), supported by a visiting professorship awarded by the German Science Foundation. He has published extensively on different aspects of electrical network theory, graph theory, and design and analysis of algorithms. He has coauthored two graduate level text books: *Graphs, Networks and Algorithms* (New York: Wiley-Interscience, 1981) and *Graphs: Theory and Algorithms* (New York: Wiley-Interscience, 1992). He has also contributed two chapters to a forthcoming book, *Handbook of Circuits and Filters*, to be published by the CRC Press. His current research interests are in graph theory, parallel and distributed computations, operations research, and computational graph theory with applications in VLSI design automation, computer networks and communication network planning

Dr. Thulasiraman was an Associate Editor of the IEEE Transactions on Circuits and Systems and is currently a Regional Editor of the *Journal of Circuits, Systems and Computers.* He was technical program chair of the IEEE International Symposium on Circuits and Systems in 1993.