

Diagnosis of Clustered Faults and Wafer Testing

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

Abstract—A probabilistic diagnosis algorithm is presented for constant degree structures. The performance of the algorithm is analyzed under a negative binomial failure distribution to account for fault clustering. It is shown that the algorithm can correctly identify almost all units even when the yield is low (much lower than 50%) and when faults are clustered. A wafer test structure is proposed, which utilizes the test access port of each die to perform comparison tests on its neighbors and incorporates a localized version of the diagnosis algorithm to determine the status of each die. Both the test time and the diagnosis time are invariant with respect to the number of dies on the wafer. The saving of test costs could be significant as compared with probe testing, because with probe testing dies are probed one at a time while they are tested in parallel with this scheme. The scheme is unique in that it is shown to work well when faults are clustered and when the yield is low.

Index Terms—Boundary scan, diagnosis, fault clustering, probabilistic diagnosis, probeless testing, VLSI testing, wafer testing.

I. INTRODUCTION

CONTINUING advances in the semiconductor technology have made possible the development of very large digital systems comprising hundreds of thousands of components or units. Yet it is impossible to build such systems without defects. As the size of a system grows, it is more likely to develop faults both in the manufacturing process and during the operation period. Testing of such systems becomes extremely difficult due to their large sizes. First, the complexity of test generation for such large systems is overwhelming. Second, the application of test data and observation and analysis of test responses are extremely difficult and costly, even if test data could be generated. This problem may be further aggravated by possible geographical distribution of units. Testing of such systems with the traditional stimuli-supplying and responses-observing philosophy has become virtually impossible.

System level diagnosis, originated by Preparata *et al.* [1], offers a viable alternative for such large systems. Instead of having a tester to test the whole system, the units are made to test each other through the interconnects. The result of such an inter-unit test may be unreliable since the testing unit may be faulty itself. Therefore, the whole set of test outcomes must be analyzed to locate the real faulty units. No postulate is to be made in the course of test outcome analysis either on the status

(fault-free or faulty) of any of the units or on the correctness of any of the test outcomes produced by the testing units.

Fig. 1 shows an example of inter-unit testing, where each unit is represented by a vertex and each test by an arc. An arc from vertex u to vertex v means that u tests v . Test outcomes are classified as fault-free or faulty. The set of test outcomes is called the *syndrome* of the system. Units can test others or be tested by others. It is assumed that test outcomes produced by fault-free testing units are always correct while those produced by faulty testing units can be anything (fault-free or faulty), irrespective of the status of the tested units. This kind of test outcome interpretation has since been known as the Preparata, Metzger, and Chien (PMC) model. A system is said to be *one-step t -diagnosable* if all faulty units can be identified from any syndrome produced by the system as long as the number of faulty units present does not exceed t .

Hakimi and Amin [2] presented the first full characterization of one-step t -diagnosable systems. Dahbura and Masson [3] presented an $O(n^{2.5})$ diagnosis algorithm for one-step t -diagnosable systems. In addition, several variations of the PMC model have been proposed in the literature [4]–[6] arising from different considerations on fault types, ways of testing, test invalidation, etc. While the PMC model assumes that the test outcomes produced by a faulty testing unit are unpredictable, the model proposed by Barsi *et al.* [4], known as the Barsi, Grandoni, and Maestrini (BGM) model, assumes that faulty units are always evaluated to be faulty even if the testing units are faulty. Chwa and Hakimi [5] suggested that the stimuli-supplying/response observing-like testing scheme be replaced by comparison of computed results. Maeng and Malek [6] suggested that the computing units themselves serve as comparators.

One-step t -fault diagnosis requires a large number of tests between units. In many existing systems, however, each unit is usually connected to a very small number of other units. For instance, in a rectangular grid connection structure each unit is connected only to four neighboring units, irrespective of its size (the number of units in the system). Even in hypercubes, a unit is connected only to $\log_2 n$ other units, where n is the number of units in the system. The degrees of one-step t -diagnosability, the largest values of t for which the systems are t -diagnosable, of these systems are very small compared with their sizes. Somani *et al.* [7], however, found that many fault sets of large sizes were still diagnosable in these sparsely connected systems, though their degrees of t -diagnosability were small. Further work in this direction stems from a probabilistic approach used first by Scheinerman [8] and then by Blough [9], which firmly establishes that as n tends to infinity, each unit can be correctly identified when each unit

Manuscript received September 20, 1993; revised August 1, 1997. This paper was recommended by Associate Editor W. Maly.

K. Huang is with Nortel, Ottawa, Canada.

V. K. Agarwal is with LV Software, San Jose, CA USA.

K. Thulasiraman is with the School of Computer Science, University of Oklahoma, Norman, OK 73019 USA (e-mail: thulasi@cs.ou.edu).

Publisher Item Identifier S 0278-0070(98)02547-0.

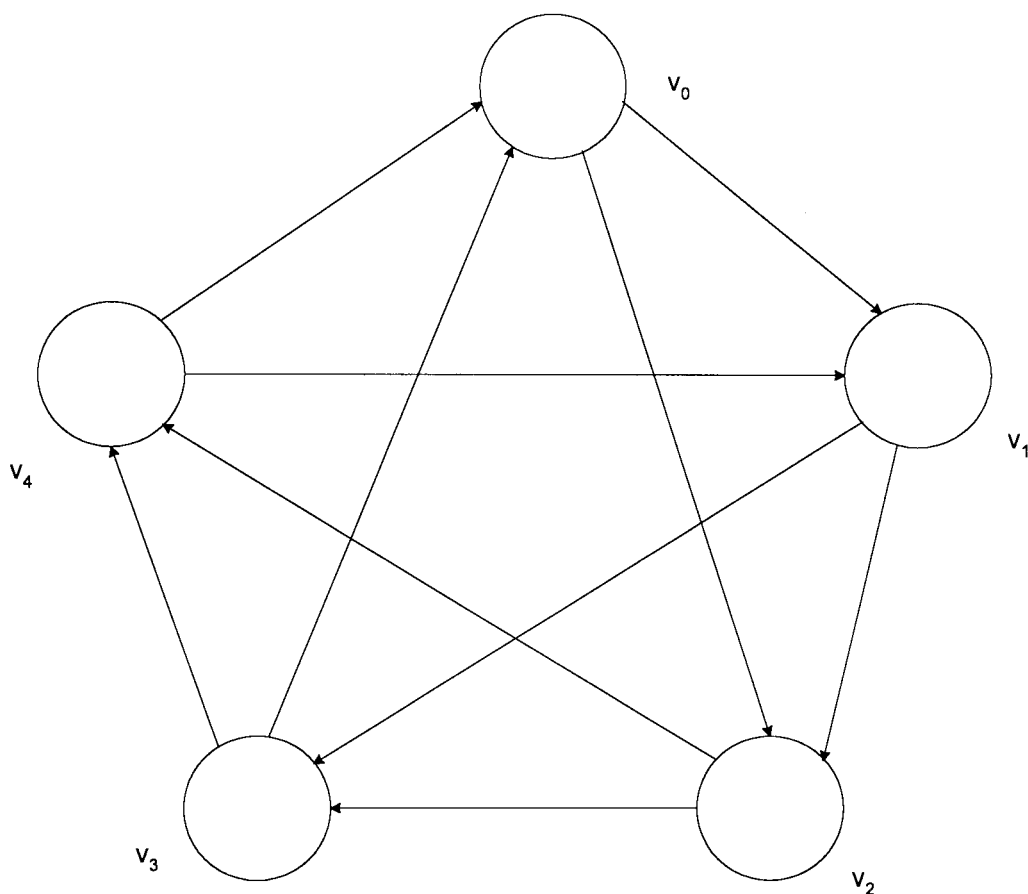


Fig. 1. An example of inter-unit testing.

is tested by slightly more than $\log n$ other units. Blough also showed that correct diagnosis with high probability was impossible if each unit was tested by only $O(\log n)$ other units. Fussell and Rangarajan [10] considered performing multiple tests to achieve correct diagnosis of constant degree connection structures. 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 $n \rightarrow \infty$. As in [8] and [9], a binomial failure distribution was assumed. More recently [11] 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 $O(\log n)$ as $n \rightarrow \infty$.

System level diagnosis was originally oriented to determining the status of units in an interconnected setting, such as in a distributed system. As suggested by Rangarajan *et al.* [12], the concepts of system level diagnosis may also be applied in testing dies on a wafer if temporary connections can be laid down on the wafer to connect the dies. Interunit tests can then be performed through the temporary connections and the status of the dies can be determined by a system level diagnosis procedure. An immediate advantage of this approach is the saving of test time as the tests can be performed in parallel. In [12], the authors showed that the status of each die could be effectively determined with their diagnosis algorithm proposed in [10]. Again, a binomial failure distribution was assumed; i.e., every die was assumed to have the same probability of

failure and faults were assumed to occur independently. No specific test structure was provided.

In this paper, we will present a simple diagnosis algorithm which takes into account fault clustering. A probabilistic analysis of this algorithm under the negative binomial failure distribution will also be presented. The application of this algorithm to the production testing of wafers will also be discussed. This algorithm and its application to wafer testing are significant because in previous works fault clustering has not been considered and it has been observed that faults are actually clustered on the wafer.

Our wafer testing scheme is a combination of the built-in self test and system level diagnosis techniques. It uses scan paths to facilitate comparison testing on neighboring dies and determines the status of each die with the help of a system diagnosis procedure.

In an earlier paper [13], we proposed a system level diagnosis algorithm for constant degree structures. The units were assumed to test each other rather than compared by comparators. Neither fault clustering nor the application of this diagnosis algorithm to wafer testing was considered. The basic idea behind this diagnosis algorithm will be used again in the algorithm to be used in our wafer testing scheme. However, comparison testing will be employed rather than letting the units test each other. As we will see later, a localized version of our diagnosis algorithm is incorporated in the wafer test structure. No host system is needed to analyze

the comparison outcomes. The test structure only needs a very small amount of circuitry and is easily implementable. The performance of the scheme is analyzed under a negative binomial failure distribution to account for fault clustering. The yield, the percentage of units which are fault-free, is allowed to be fairly low (much lower than 50%) while almost all units are guaranteed to be correctly identified. In fact, the performance of the algorithm is insensitive to yield variations. Some implementation related issues, such as clock skew and power dissipation have also been addressed. The reader is referred to [14] for more details on these issues.

The remainder of the paper is organized as follows: some basic concepts are introduced in Section II; the diagnosis algorithm is described in Section III; the performance of the diagnosis algorithm is analyzed in Section IV; the wafer test structure, as well as the wafer testing procedure, is described in Section V; finally, Section VI concludes the paper.

II. PRELIMINARIES

As in [5], diagnosis is performed on the basis of comparison testing of neighboring units. A pair of units are considered to be neighbors if there is a comparator between them. Comparison tests can be performed by assigning identical jobs to neighboring units and then by comparing their computed results. The jobs to be executed on the units can be real jobs which have useful results or jobs which are designed exclusively for testing. The comparison tests can be modeled by an undirected graph $G(V, E)$ called the *comparison graph*, where V is the set of vertices representing the units and E the set of edges representing comparison tests. There is an edge between vertices u and v if and only if there is a comparator between u and v . The outcome of each comparison test is either "0" or "1," representing "match" or "mismatch." The set of comparison outcomes is called the *syndrome of the system* and can be seen as a function w of edges. $w(u, v)$ will be called the *weight* of edge (u, v) . As in [15], we will allow the comparison tests to be imperfect, i.e., a faulty unit may produce correct responses for all the test patterns applied. To be specific, we assume that the probability that $w(u, v) = 1$ for a fault-free vertex u and faulty vertex v takes on some value c , $0 \leq c \leq 1$ called the *coverage of the test*. For an easier analysis, we also assume that this probability is independent for the comparison tests. For any two fault-free vertices u and v , the comparison always leads to a match; in other words, we always have $w(u, v) = 0$. Furthermore, the probability that two faulty vertices produce all identical responses, or $w(u, v) = 0$ is assumed to take on some small value β , $0 \leq \beta \leq 1$. The *agreement graph*, denoted by $G_A(V, E_A)$ deduced from the comparison graph $G(V, E)$ and the syndrome is a subgraph of $G(V, E)$ with the same vertex set such that an edge (u, v) is in E_A if and only if $(u, v) \in E$ and $w(u, v) = 0$.

III. DIAGNOSIS ALGORITHM

Unlike other probabilistic diagnosis algorithms which use various forms of voting for deciding the status of a unit, our algorithm considers the size of a cluster of units which claim each other to be fault-free but outsiders to be faulty and

then determines the status of the whole cluster of units. Such a cluster of units will be called a *faction*. More precisely, a faction is a set of vertices V' such that $G_A[V']$ is a connected component of $G_A(V, E_A)$. 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. Nonetheless, the geometric shape of a faction is not important in our algorithm; only the number of units in the whole faction matters.

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 assignment, syndrome, and threshold value k .

Output: Set of vertices declared fault-free: R .

Step 1: Set $R = \Phi$.

Step 2: For every unit v , set $R = R \cup \{v\}$ if v is in a faction of size larger than k .

End

Note that the set of units V is partitioned into (disjoint) subsets, each of which forms a faction. A unit, whether fault-free or faulty, is always in a faction. If a unit is not in a faction of size no larger than k , then it is in a faction of size larger than k . Therefore, it suffices to check if a unit is in a faction of size no larger than k in determining the status of the unit. This has two advantages. First, performance analysis is simplified. Second, the diagnosis algorithm can be localized, as only local information is needed in determining the status of each unit.

Algorithm 3.1 can be executed in the following steps.

- 1) Construct the agreement graph from the comparison graph and the syndrome of the system.
- 2) Find the connected components of the agreement graph and the sizes of the components.
- 3) For each unit, declare it faulty if it is in a connected component of size no larger than k , otherwise declare this unit fault-free.

It is obvious that Step 1 can be done in $O(|E|)$ time. Step 2 can also be done in $O(|E|)$ time with a simple depth-first search of the agreement graph. Step 3 needs only $O(|V|)$ time. The time complexity of the algorithm is therefore $O(|E|)$. As we only consider constant degree structures, the time complexity is identical to $O(|V|)$.

As it will be seen later, we can have very good diagnosis resolution with a small threshold value k . Therefore, the determination of the status of a vertex can be done in a small neighborhood of the vertex under consideration. This allows for a distributed implementation of the algorithm. For the threshold $k = 2$, syndrome decoding can be done locally as follows.

Algorithm 3.2 (Local Syndrome Decoding)

Step 1: For each vertex v , label v fault-free if there are two edges incident on v with weights equal to 0.

Step 2: For each vertex v not labeled in Step 1, label it fault-free if it has a neighbor labeled fault-free in Step 1 and the weight of the edge joining them is 0.

Step 3: For each vertex v not labeled, label it faulty.

End

Theorem 3.1: A vertex is identified to be faulty by Algorithm 3.2 if and only if it is in a faction of size no larger than two.

Proof: If a vertex is in a faction of size larger than two, then either it has two neighbors in the faction or it has a neighbor which has two neighbors in the faction. This means that it will be identified to be fault-free by Algorithm 3.2 in either Steps 1 or 2. It implies that a vertex is identified to be faulty only if it is in a faction of size no larger than two. On the other hand, if a vertex v is in a faction of size no larger than two, then it has at most one neighbor, say u , in the same faction and u and v are the only vertices in that faction. This means that neither of them can be identified to be fault-free in Step 1. It is clear that they will not be identified to be fault-free in Step 2 either. \square

The above theorem implies that any faction of size three or more where faulty units agree on the result of a test will be declared good, irrespective of additional tests with units which are not members of the faction.

IV. PERFORMANCE ANALYSIS

It has been observed in very large scale integration (VLSI) manufacturing that faults are spatially clustered on the wafer. This phenomenon was studied by Stapper *et al.* [16] and a negative binomial failure distribution was used in modeling the yield of VLSI circuits. This yield model has been widely used in the VLSI industry. Some variations of this model have also been proposed. In [17], the authors have presented a unified approach to yield analysis using the negative binomial distribution. For a review of yield modeling, the reader is referred to [18].

In the following, we will examine the performance of our diagnosis algorithm under a negative binomial failure distribution to account for fault clustering. It is worth noting that this failure distribution model has not been used heretofore in performance analysis of diagnosis algorithms. As will be seen later in this section, a larger fraction of units can be correctly identified with our diagnosis algorithm when faults are clustered. This phenomenon is similar to what has been observed in yield modeling, where the yield improves when faults are clustered. It can be intuitively explained as follows. When faulty units cluster together, as a dual effect, fault-free units also clustered together. Because of the reliable nature of fault-free units, a larger cluster of fault-free units necessarily results in a larger faction and therefore they are more likely to be correctly identified. On the other hand, a larger cluster of faulty units is unlikely to result in a larger faction because two faulty units are unlikely to produce identical responses.

Due to space limitation, in our analysis we will consider rectangular grids only.

A. Negative Binomial Failure Distribution

Poisson statistics are commonly used to model the distribution of the number of faults per unit (die) in yield modeling. If faults are evenly distributed on the wafer with defect density D and unit area A then the yield Y can be expressed as

$$Y = e^{-\lambda}$$

where $\lambda = DA$ is the average number of faults per unit.

Fault clustering can be taken into account by assuming λ to be a random variable rather than a constant. The modified yield formula is then obtained by averaging the above formula with respect to λ

$$Y = \int_0^{\infty} e^{-\lambda} f(\lambda) d\lambda$$

where $f(\lambda)$ is the probability density function of λ .

A commonly used probability density function $f(\lambda)$ is the Gamma distribution with two parameters α and δ

$$f(\lambda) = \frac{\lambda^{\alpha-1} e^{-\lambda/\delta}}{\delta^{\alpha} \Gamma(\alpha)}.$$

Using the above expression for $f(\lambda)$, we can obtain the following well-known integrated circuit yield formula

$$Y(1 + \bar{\lambda}/\alpha)^{-\alpha}$$

where α is a clustering parameter and $\bar{\lambda} = \delta\alpha$ is the average number of faults per unit over the whole wafer. With the Gamma distribution, λ can be interpreted as the average number of faults per unit in areas where the fault density is $D = \lambda/A$. $\bar{\lambda}$ is therefore the grand average number of faults per unit over the whole wafer. It can be shown that $\bar{\lambda}$ is in effect the expected value of λ when the probability density function assumes the Gamma distribution.

A larger value of α implies decreased clustering. In the limit when $\alpha \rightarrow \infty$, there is absence of clustering and the yield formula defaults into a yield formula for an even failure distribution. Actual values of α are typically around one.

Later on we will use the Laplace–Stieltjes transform of $F(x)$

$$X^*(\theta) = \int_0^{\infty} e^{-\theta x} dF(x)$$

where $F(x)$ is the cumulative distribution function of the average number of faults per unit and $dF(x) = f(x) dx$. If the probability density function assumes the Gamma distribution, then

$$X^*(\theta) = \left(1 + \theta \frac{\bar{\lambda}}{\alpha}\right)^{-\alpha}.$$

For more details of the Laplace–Stieltjes transform, the reader is referred to [19].

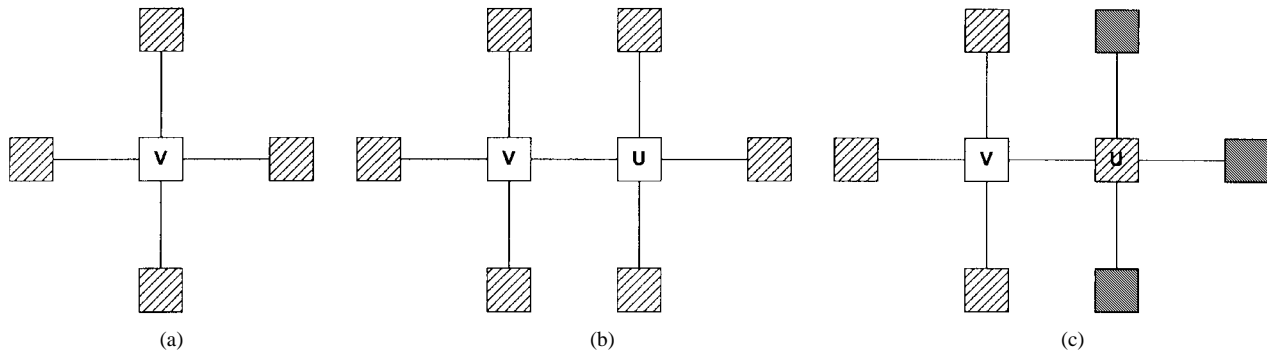


Fig. 2. Status-syndrome patterns for a fault-free vertex.

B. Local Performance

In the following, we will analyze the local performance of our diagnosis algorithm, i.e., the probability that an individual unit is correctly identified. The global performance, by which we mean the fraction of units which are correctly identified, will be discussed in the next section.

Let $P_G(v)$ be the probability that given v fault-free, v is correctly identified, and let $P_B(v)$ be the probability that given v faulty, v is correctly identified. In the following, we will deduce analytical expressions for $P_G(v)$ and $P_B(v)$. For simplicity, we will only consider the case of $k = 2$. In addition, our discussion will be limited to the rectangular grids. Similar analysis can also be done for the octagonal grids.

Assume that the average number of faults per unit in areas with fault density D is A . The yield for such areas is therefore

$$Y_\lambda = e^{-\lambda}.$$

We consider Y_λ to be the probability that a unit in such an area to be fault-free and denote it as q . In other words, we assume that

$$q = e^{-\lambda}.$$

The probability of failure p is therefore $1 - q = 1 - e^{-\lambda}$. It is clear that q is a function of λ , or, in turn, of D . There is a one-to-one correspondence among these four parameters p, q, λ , and D . Therefore, whenever one parameter is given, the other three are also given. With a negative binomial failure distribution, the probability of failure p for a unit varies from unit to unit and this results in difficulty in computing $P_G(v)$ and $P_B(v)$. In order to get analytical expressions for $P_G(v)$ and $P_B(v)$, we will make some approximations. As we will see in the following analysis, only the information about units in a small neighborhood of the unit under consideration is needed to deduce the expressions for P_G and P_B . For purposes of tractability in analysis, we will make the assumption that all units in a small neighborhood of the unit under consideration have the same fault density and therefore the same q and p values. Similar assumptions were also made in [16]. In the following discussions, we assume that the failure distribution in a neighborhood of a radius of two centered at the unit under consideration is binomial with a probability of failure p .

If vertex v is fault-free, there are three classes of local status-syndrome patterns corresponding to the event that v is in a faction of size no larger than two. There is a single pattern

in the first class in which v disagrees with all of its four neighbors. As v is fault-free, the four neighbors must be all faulty. This status-syndrome pattern is depicted in Fig. 2(a). In this figure, a dashed block represents a faulty vertex, a dark block, a vertex possibly faulty or fault-free, and an unfilled block, a fault-free vertex. An internal edge of a faction is represented by a thick line. In the second class of status-syndrome patterns, v has a single fault-free neighbor u and they are surrounded by faulty vertices. Note that the fault-free neighbor can be anyone of the four neighbors of v . The pattern in which u is to the east of v is depicted in Fig. 2(b). Rotating the picture around v by 90° at a time produces the other three patterns in this class. In the third class of status-syndrome patterns, v is completely surrounded by faulty vertices but exactly one of them, say u , agrees with v and all the other three neighbors of u disagree with u . As u is faulty, these three neighbors may be either fault-free or faulty. The pattern in which u is to the east of v is depicted in Fig. 2(c). Rotating the picture around v by 90° at a time produces the other three patterns. It is easy to see that these three classes of status-syndrome patterns exhaust all the possibilities that v is in a faction of size no larger than two. Note that all these patterns are pair-wise disjoint. This can be shown as follows. Assume, to the contrary, that there are two patterns which appear simultaneously. Overlay the two patterns and align them according to the positions of v . We can find either a vertex which is faulty in one pattern and fault-free in the other or an edge whose weight is 1 in one pattern and 0 in the other. This will contradict the assumption that the two patterns can appear simultaneously. Let $q_1^0(v)$, $q_2^0(v)$, and $q_3^0(v)$ be the probabilities that a fault-free vertex v is in a faction corresponding to a first class status-syndrome pattern, to a second class status-syndrome pattern, and to a third class status-syndrome pattern, respectively. From the above discussions we have the following.

Lemma 4.1: $q_1^0(v) = p^4 c^4$.

Proof: Consider Fig. 2(a). Vertex v is given to be fault-free. There are four vertices which are faulty. This can happen with the probability p^4 . Four edges each joining a fault-free vertex and a faulty vertex have the weights of 1. The probability that this event happens is c^4 according to our model. The probability of occurrence for the whole event is therefore $p^4 c^4$. \square

Lemma 4.2: $q_2^0(v) = 4(1-p)p^6 c^6$.

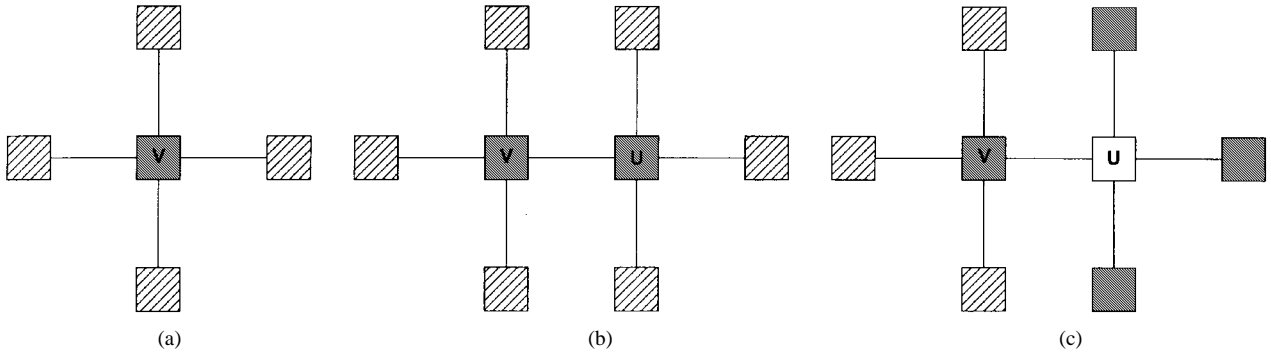


Fig. 3. Status-syndrome patterns for a faulty vertex.

Proof: Consider Fig. 2(b). There is another vertex u which is fault-free. The probability that this vertex is fault-free is $1 - p$. Six vertices are faulty and the probability of its occurrence is p^6 . We can also note that six edges each joining a fault-free vertex and a faulty vertex have the weights of 1 and its probability of occurrence is c^6 . Note that the weight on the edge joining v and u is definitely 0 given that they are both fault-free. The probability of occurrence for the status-syndrome pattern shown in Fig. 2(b) is therefore $(1 - p)p^6c^6$. As there are four symmetric patterns in the same class, we have the factor of 4. \square

In the following, recall that β refers to the probability that two faulty units produce all identical responses.

Lemma 4.3: $q_3^0(v) = 4(1 - c)c^3p^4[(1 - p)c + p(1 - \beta)]^3$.

Proof: The proof is similar to the proof of Lemma 4.2. Note that there are three edges each joining the faulty vertex v and a vertex which may be faulty or fault-free. The probability that such an edge has the weight of 1 is $(1 - p)c + p(1 - \beta)$. The first term corresponds to the case of the other end of the edge being fault-free and the second term to the case of the other end of the edge being faulty. \square

Theorem 4.1: $P_G(v) = 1 - [q_1^0(v) + q_2^0(v) + q_3^0(v)]$

Proof: It follows from the above discussions that the three classes of status-syndrome patterns are pairwise disjoint and exhaust all the patterns corresponding to the event that v is fault-free and is in a faction of size no larger than two. Hence the theorem. \square

Now let us assume that vertex v is faulty. Similar to the case that v is fault-free, there are three classes of local status-syndrome patterns corresponding to the event that v is in a faction of size no larger than two. The first class contains a single pattern and is shown in Fig. 3(a). In this figure, a dark block represents a faulty vertex, a dashed block, a vertex possibly faulty or fault-free, and an unfilled block, a fault-free vertex. All the four edges incident on v have the weights of 1. The vertices adjacent to v may be either fault-free or faulty. In the second class of patterns v has a unique faulty neighbor u which agrees with v . The pattern with u to the east of v is depicted in Fig. 3(b). Rotating the picture by 90° at a time produces the other three symmetric patterns. In the third class of patterns v has a unique fault-free neighbor u which agrees with v . The pattern with u to the east of v is depicted in Fig. 3(c). Rotating the picture by 90° at a time produces the other three symmetric patterns in the class. Similar to the

case that v is fault-free which we discussed above, we can show that these three classes of patterns are pairwise disjoint and they exhaust all the patterns corresponding to the event that v is in a faction of size no larger than two. Let $q_1^1(v)$ be the probability that, given that v is faulty, v is in a faction of size no larger than two corresponding to a status-syndrome pattern in the first class, $q_2^1(v)$ the probability corresponding to a status-syndrome pattern in the second class, and $q_3^1(v)$ the probability corresponding to a status-syndrome pattern in the third class. The following conclusions follow from these definitions and discussions and their proofs are omitted to save space.

Lemma 4.4: $q_1^1(v) = [(1 - p)c + p(1 - \beta)]^4$.

Lemma 4.5: $q_2^1(v) = 4p\beta[(1 - p)c + p(1 - \beta)]^6$.

Lemma 4.6: $q_3^1(v) = 4(1 - p)(1 - c)p^3c^3[(1 - p)c + p(1 - \beta)]^3$

Theorem 4.2: $P_B(v) = q_1^1(v) + q_2^1(v) + q_3^1(v)$.

The following result shows how $P_G(v)$ and $P_B(v)$ would change with the change of β .

Theorem 4.3: $P_G(v)$ monotonically increases with the increase of β ; $P_B(v)$ monotonically decreases with the increase of β .

Proof: Let us compute the partial derivatives $\partial[P_G(v)]/\partial(\beta)$ and $\partial[P_B(v)]/\partial(\beta)$

$$\frac{\partial[P_G(v)]}{\partial(\beta)} = 12(1 - c)c^3p^5[(1 - p)c + p(1 - \beta)]^2 \geq 0.$$

This means that $P_G(v)$ monotonically increases with the increase of β .

$$\begin{aligned} \frac{\partial[P_B(v)]}{\partial(\beta)} &= -4p[(1 - p)c + p(1 - \beta)]^3 \\ &\quad - 24p^3\beta[(1 - p)c + p(1 - \beta)]^5 \\ &\quad + 4p[(1 - p)c + p(1 - \beta)]^6 \\ &\quad - 12(1 - p)(1 - c)p^4c^3[(1 - p)c + p(1 - \beta)]^2. \end{aligned}$$

All the terms are less than or equal to zero except the third term. Note that the sum of the first term and the third term is no larger than zero because $(1 - p)c + p(1 - \beta) \leq 1$. The partial derivative is therefore less than or equal to zero. This completes the proof. \square

The changes of $P_G(v)$ and $P_B(v)$ with the change of β are very small within a reasonable range of β (e.g., $[0, 0.01]$). For

example, with $p = 0.5$ and $c = 0.99$, $P_G(v)$ is 0.908 19 when $\beta = 0.01$ and 0.908 15 when $\beta = 0$. The difference in $P_G(v)$ is only 0.000 04. According to the above theorem, $P_G(v)$ will settle between 0.908 15 and 0.908 19 for any value of β within the range of $[0, 0.01]$. Similarly, with $p = 0.5$ and $c = 0.99$, $P_B(v)$ is 0.9818 when $\beta = 0.01$ and 0.9825 when $\beta = 0$. The difference in $P_B(v)$ is only 0.0007. As $P_B(v)$ monotonically decreases with the increase of β , it will settle between 0.9818 and 0.9825 for any value of β within the range of $[0, 0.01]$. As we have seen, $P_G(v)$ and $P_B(v)$ are not sensitive to the change of β as long as β is moderately small. Since a nonzero value of β is allowed, our scheme is able to tolerate symmetric failures.

C. Global Performance

In Section IV-B, we deduced analytical expressions for P_G and P_B for a unit in an area with a probability of unit failure p . However, it is still not clear what percentage of units we are able to correctly identify. In this section, we will analyze the performance of our diagnosis algorithm from a global perspective.

Assume that the probability of correctly identifying a good unit P_G in an area of fault density D is some polynomial in q (as stated earlier, q is a function of D). Later on, we will show that it is indeed a polynomial in q and give the actual polynomial but let it be a polynomial in q for now. Denote this probability to be

$$P_{G,q} = \sum_i a_i q^i$$

where a_i 's are constants.

Similarly, assume that the probability of correctly identifying a bad unit in an area of fault density D is

$$P_{B,q} = \sum_i b_i q^i$$

where b_i 's are constants.

For the convenience of discussion, we have indexed these parameters by q instead of D . In the following, we will discuss and give expressions for the fractions of good (bad) units correctly identified with a negative binomial failure distribution.

Let Y be the expected ideal yield for the whole wafer (the fraction of units which are good but not necessarily correctly identified). As we have shown

$$Y = \int_0^\infty e^{-\lambda} f(\lambda) d\lambda = \left(1 + \frac{\bar{\lambda}}{\alpha}\right)^{-\alpha}.$$

Let F be the expected fraction of units which are bad. It is clear that $F = 1 - Y$. Let Y_c be the expected fraction of units which are good and correctly identified. Let F_c be the expected fraction of units which are bad and correctly identified. We have the following theorem.

Theorem 4.4: Given that $P_{G,q} = \sum_i a_i q^i$ and $P_{B,q} = \sum_i b_i q^i$, the following holds:

$$\begin{aligned} 1) \quad \frac{Y_c}{Y} &= \frac{\sum_i a_i \left(1 + (i+1) \frac{\bar{\lambda}}{\alpha}\right)^{-\alpha}}{\left(1 + \frac{\bar{\lambda}}{\alpha}\right)^{-\alpha}} \\ 2) \quad \frac{F_c}{F} &= \frac{\sum_i b_i \left(\left(1 + i \frac{\bar{\lambda}}{\alpha}\right)^{-\alpha} - \left(1 + (i+1) \frac{\bar{\lambda}}{\alpha}\right)^{-\alpha} \right)}{1 - \left(1 + \frac{\bar{\lambda}}{\alpha}\right)^{-\alpha}}. \end{aligned}$$

Proof: It is easy to see that the probability that a unit in an area of fault density D is good and $qP_{G,q}$ is correctly identified. The expected fraction of units which are good and correctly identified is therefore

$$\begin{aligned} Y_c &= \int_0^\infty q P_{G,q} f(\lambda) d\lambda \\ &= \int_0^\infty q \sum_i a_i q^i f(\lambda) d\lambda \\ &= \sum_i a_i \int_0^\infty (e^{-\lambda})^{i+1} f(\lambda) d\lambda \\ &= \sum_i a_i \int_0^\infty e^{-(i+1)\lambda} f(\lambda) d\lambda. \end{aligned}$$

Note that $\int_0^\infty e^{-(i+1)\lambda} f(\lambda) d\lambda$ is the Laplace–Stieltjes transform $X^*(i+1)$ of $F(\lambda)$, and therefore

$$Y_c = \sum_i a_i \left(1 + (i+1) \frac{\bar{\lambda}}{\alpha}\right)^{-\alpha}.$$

As $Y = \left(1 + \frac{\bar{\lambda}}{\alpha}\right)^{-\alpha}$, we have the first equation of the theorem. Similarly, we have the following:

$$\begin{aligned} Y_c &= \int_0^\infty (1-q) P_{B,q} f(\lambda) d\lambda \\ &= \int_0^\infty (1-q) \sum_i b_i q^i f(\lambda) d\lambda \\ &= \int_0^\infty \sum_i b_i q^i f(\lambda) d\lambda - \int_0^\infty \sum_i b_i q^{i+1} f(\lambda) d\lambda \\ &= \sum_i b_i \left(\int_0^\infty e^{-i\lambda} f(\lambda) d\lambda - \int_0^\infty e^{-(i+1)\lambda} f(\lambda) d\lambda \right) \\ &= \sum_i b_i \left[\left(1 + (i+1) \frac{\bar{\lambda}}{\alpha}\right)^{-\alpha} - \left(1 + (i+1) \frac{\bar{\lambda}}{\alpha}\right)^{-\alpha} \right]. \end{aligned}$$

It is obvious that $F = 1 - Y = 1 - \left[1 + \frac{\bar{\lambda}}{\alpha}\right]^{-\alpha}$ and hence we have the second equation of the theorem. \square

In Section IV-B we gave analytical expressions for P_G and P_B , the probability of correctly identifying a good unit and the

TABLE I
PERFORMANCE WITH $c = 0.99$ AND $\beta = 0.01$

Y	$\frac{Y_C}{Y} / \frac{F_C}{F}$				
	$\alpha = 0.3$	$\alpha = 0.6$	$\alpha = 1$	$\alpha = 2$	$\alpha = \infty$
0.9	0.9949/0.9741	0.9976/0.9698	0.9987/0.9678	0.9995/0.9661	0.9999/0.9644
0.8	0.9803/0.9824	0.9844/0.9773	0.9887/0.9743	0.9933/0.9716	0.9982/0.9684
0.7	0.9661/0.9875	0.9637/0.9829	0.9672/0.9799	0.9748/0.9768	0.9900/0.9726
0.6	0.9551/0.9908	0.9409/0.9871	0.9376/0.9845	0.9414/0.9816	0.9652/0.9771
0.5	0.9476/0.9931	0.9191/0.9903	0.9036/0.9882	0.8939/0.9858	0.9082/0.9818
0.4	0.9429/0.9947	0.8998/0.9927	0.8679/0.9912	0.8346/0.9894	0.8003/0.9864
0.3	0.9402/0.9960	0.8835/0.9946	0.8322/0.9935	0.7655/0.9924	0.6274/0.9908
0.2	0.9390/0.9969	0.8707/0.9961	0.7976/0.9955	0.6871/0.9949	0.3937/0.9946
0.1	0.9386/0.9977	0.8617/0.9973	0.7645/0.9970	0.5958/0.9968	0.1449/0.9972

TABLE II
PERFORMANCE WITH $c = 0.95$ AND $\beta = 0.01$

Y	$\frac{Y_C}{Y} / \frac{F_C}{F}$				
	$\alpha = 0.3$	$\alpha = 0.6$	$\alpha = 1$	$\alpha = 2$	$\alpha = \infty$
0.9	0.9953/0.8764	0.9978/0.8563	0.9988/0.8469	0.9995/0.8394	0.9999/0.8315
0.8	0.9820/0.9164	0.9857/0.8915	0.9896/0.8774	0.9938/0.8645	0.9983/0.8494
0.7	0.9689/0.9417	0.9668/0.9188	0.9701/0.9040	0.9770/0.8890	0.9907/0.8688
0.6	0.9587/0.9587	0.9459/0.9396	0.9430/0.9263	0.9466/0.9118	0.9680/0.8897
0.5	0.9517/0.9706	0.9257/0.9557	0.9116/0.9447	0.9032/0.9323	0.9163/0.9117
0.4	0.9473/0.9793	0.9077/0.9682	0.8786/0.9599	0.8487/0.9503	0.8190/0.9342
0.3	0.9448/0.9859	0.8925/0.9782	0.8454/0.9723	0.7846/0.9657	0.6621/0.9559
0.2	0.9437/0.9910	0.8805/0.9862	0.8129/0.9826	0.7112/0.9787	0.4457/0.9752
0.1	0.9433/0.9950	0.8721/0.9928	0.7817/0.9911	0.6246/0.9896	0.2015/0.9902

probability of correctly identifying a bad unit, for a unit with a probability of failure p . These expressions can be used for $P_{G,q}$ and $P_{B,q}$ except that the p variables need to be substituted by $1 - q$.

Theorem 4.5: Assume that all units in a neighborhood of a radius of two have the same probability q of being fault-free and that the comparison structure is a rectangular grid. Let $x = 1 - \beta$ and $y = \beta + c - 1$. The following holds for the threshold $k = 2$:

$$1) \quad P_{G,q} = \sum_{i=0}^7 a_i q^i$$

where

$$\begin{aligned} a_0 &= 1 - (c^4 + 4(1-c)c^3x^3) \\ a_1 &= 4c^4 - 4c^6 - 4(1-c)c^3(3x^2y - 4x^3) \\ a_2 &= -6c^4 + 24c^6 - 4(1-c)c^3(3xy^2 - 12x^2y + 6x^3) \\ a_3 &= 4c^4 - 60c^6 - 4(1-c)c^3(y^3 - 12xy^2 + 18x^2y - 4x^3) \\ a_4 &= -c^4 + 80c^6 - 4(1-c)c^3 \\ &\quad \times (-4y^3 + 18xy^2 - 12x^2y + x^3) \\ a_5 &= -60c^6 - 4(1-c)c^3(6y^3 - 12xy^2 + 3x^2y) \\ a_6 &= 24c^6 - 4(1-c)c^3(-4y^3 + 3xy^2) \\ a_7 &= -4c^6 - 4(1-c)c^3y^3 \end{aligned}$$

$$2) \quad P_{B,q} = \sum_{i=0}^7 b_i q^i$$

where

$$\begin{aligned} b_0 &= x^4 + 4\beta x^6 \\ b_1 &= 4x^3y + 4\beta x^5(6y - x) + 4c^3(1-c)x^3 \\ b_2 &= 6x^2y^2 + 12\beta x^4y(5y - 2x) + 12c^3(1-c)x^2(y - x) \\ b_3 &= 4xy^3 + 20\beta x^3y^2(4y - 3x) + 12c^3(1-c) \\ &\quad \times x(y^2 - 3xy + x^2) \\ b_4 &= y^4 + 20\beta x^2y^3(3y - 4x) + 4c^3(1-c) \\ &\quad \times (y^3 - 9xy^2 + 9x^2y - x^3) \\ b_5 &= 12\beta xy^4(2y - 5x) - 12c^3(1-c)y(x^2 - 3xy + y^2) \\ b_6 &= 4\beta y^5(y - 6x) - 12c^3(1-c)y^2(x - y) \\ b_7 &= -4\beta y^6 - 4c^3(1-c)y^3. \end{aligned}$$

Proof: The proof involves some meticulous manipulation of the analytical expressions of Section IV-B and hence omitted. \square

Using the above formulae, we have calculated Y_C/Y and F_C/F : values for some typical values of c and β . The results are listed in Tables I–III. As can be seen from the tables, both Y_C/Y and F_C/F increase with the decrease of α except F_C/F in Table III. This means that the diagnosis performance improves with increased fault clustering and matches our intuitive expectation. The improvement is very significant when the yield is very low. Note that there is an exception when the yield is very high, in which case, Y_C/Y may decrease a little bit with the decrease of α . This can be explained as follows: when the yield is very high, most good

TABLE III
PERFORMANCE WITH $c = 1$ AND $\beta = 0.01$

Y	$\frac{Y_C / F_C}{Y / F}$				
	$\alpha = 0.3$	$\alpha = 0.6$	$\alpha = 1$	$\alpha = 2$	$\alpha = \infty$
0.9	0.9948/0.9997	0.9975/0.9999	0.9987/0.9999	0.9994/0.99995	0.9999/0.99995
0.8	0.9799/0.9994	0.9841/0.9996	0.9885/0.9997	0.9932/0.9998	0.9982/0.9999
0.7	0.9654/0.9991	0.9629/0.9994	0.9665/0.9995	0.9743/0.9997	0.9899/0.9998
0.6	0.9543/0.9989	0.9397/0.9992	0.9363/0.9993	0.9401/0.9995	0.9646/0.9997
0.5	0.9466/0.9987	0.9176/0.9990	0.9016/0.9991	0.8916/0.9993	0.9062/0.9996
0.4	0.9418/0.9986	0.8979/0.9988	0.8653/0.9989	0.8312/0.9991	0.7958/0.9994
0.3	0.9391/0.9985	0.8813/0.9986	0.8290/0.9987	0.7609/0.9989	0.6187/0.9991
0.2	0.9379/0.9984	0.8683/0.9985	0.7939/0.9986	0.6813/0.9987	0.3807/0.9989
0.1	0.9375/0.9983	0.8592/0.9984	0.7603/0.9984	0.5888/0.9985	0.1313/0.9986

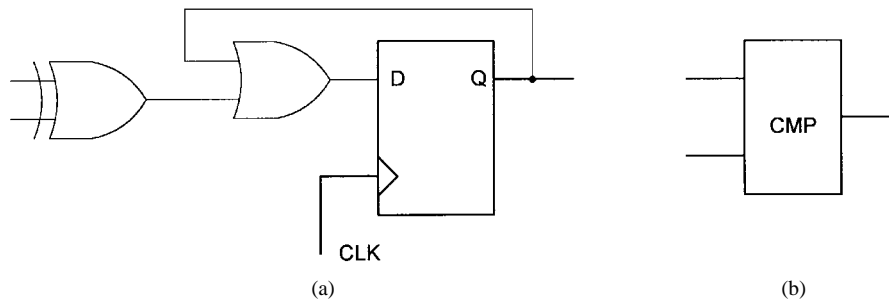


Fig. 4. Comparator circuit.

units are already in factions of sizes larger than two with a binomial failure distribution.

Clustering of good units means moving good units to form larger clusters and hence larger factions. If a good unit is already in a faction of size larger than two, then moving it to a larger faction does not improve the performance of diagnosis. Rather, some good units originally in factions of sizes larger than two may be left over by such moving and end up in factions of sizes no larger than two. This adverse effect is very small and negligible as can be seen from the tables. We can also see from the tables that the diagnosis performance is insensitive to yield variations when faults are heavily clustered and this shows that our scheme is particularly suitable for diagnosis of clustered faults. This characteristic can be attributed to the nature of our diagnosis algorithm.

V. APPLICATION TO WAFER TESTING

In this section, we will apply Algorithm 3.1 to wafer testing. The dies on a wafer are assumed to be identical. The basic idea is to add a small amount of additional circuitry to the wafer such that comparison tests can be performed on the dies and to identify the status of each die with the diagnosis algorithm. A test structure will be presented which utilizes the test access port of each die to facilitate comparison testing. The localized version (Algorithm 3.2) of Algorithm 3.1 is incorporated in the test structure which determines the status of each die right on the wafer.

We assume that each die is designed with internal scan [20] philosophy which allows it to be tested by pseudo-random or weighted pseudo-random patterns [21], [22]. We also assume that each die conforms with the IEEE 1149.1

testability standard with the test access port (TAP) and the associated test bus signals test data in (TDI), test data out (TDO), test mode select (TMS), and test clock (TCLK). For further details on this standard, the reader is referred to [23].

The basic idea is to make comparison tests on neighboring dies with the help of comparators. As in [24], the test structure has a comparator corresponding to each pair of neighboring dies, as shown in Fig. 4(a). The comparator symbol is shown in Fig. 4(b). The comparator consists primarily of an exclusive-OR gate and a latch. Test responses from two adjacent dies are fed into the comparator bit by bit and the comparison result is latched. In the end, the latch has a "1" value if and only if there is a mismatch for at least one test vector supplied or "0" value otherwise.

In real implementations the comparator can be placed in either of these two dies. A diagram of the die test structure is given in Fig. 5 for the rectangular connection topology. Each die has two comparators, one for the north neighbor and one for the west neighbor. The circuit under test is, however, omitted to signify the test structure. The overall connection topology is shown in Fig. 6.

With this arrangement, test vectors can be broadcast to all dies through the TDI pins and the following comparison testing procedure can be carried out in parallel by each die for each of the test vectors supplied:

Procedure 5.1 (Comparison Testing):

Step 1: Receive a pseudo-random vector through the TDI pin and store it in the internal scan chain.

Step 2: Capture the resulting test response in the internal scan chain.

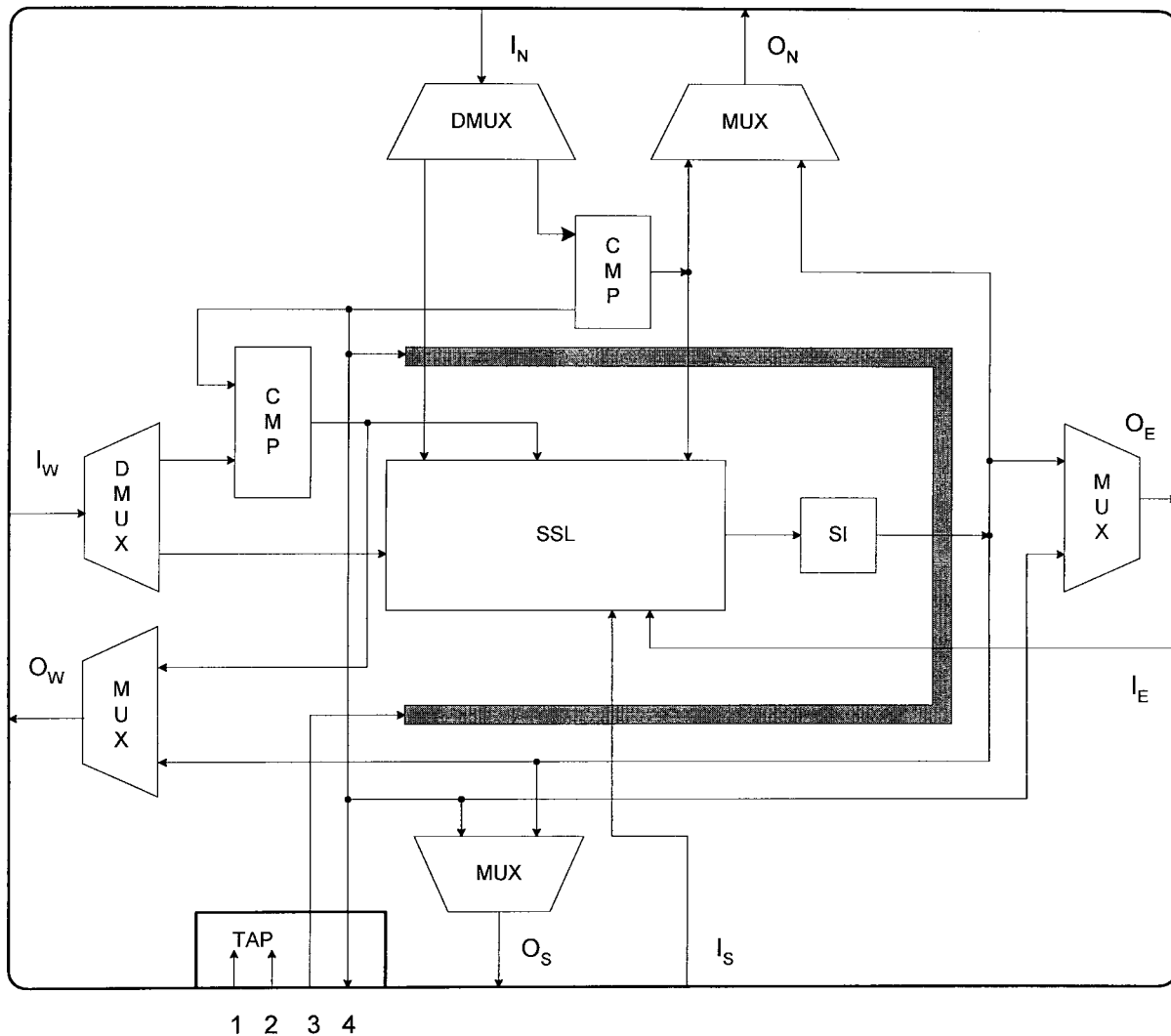


Fig. 5. Die test structure (A).

Step 3: Scan out the test response from the TDO pin and send it to its two internal comparators and to the corresponding comparators in the east and south neighbors through the O_E and O_S pins. In the meantime, each comparator compares the responses from its host die and the corresponding neighboring die bit by bit and latches the resulting agreement/disagreement bit.

The above comparison testing procedure can be executed as many times as the number of test vectors need to be applied. Note that the next test vector is scanned in at the same time while the test response is scanned out.

At the end of the comparison testing procedure, each of the comparator latches contains a 0 (1) if the dies compared agree (disagree) in all (at least one) of the test vectors.

As we can see from Fig. 5, for each die there is also some additional circuitry, the *status setting logic (SSL)* and *status indicator (SI)*. These are designed for the implementation of Algorithm 3.2 on the wafer. The status setting logic sets the status indicator according to the comparison outcomes regarding the neighbors and the contents of the SI's of the neighbors. With the above circuitry, the whole wafer testing procedure can be described as follows.

Procedure 5.2 (Wafer Testing): Each die performs the following steps in parallel.

Step 1: Initialize its SI to 1.

Step 2: Perform the comparison testing procedure.

Step 3: The SSL sets SI to 0 if two or more of the four corresponding comparator latches (the two internal ones and one from each of its east and south neighbors) have the 0 value.

Step 4: Send its SI value through $O_N, O_E, O_S,$ and O_W to and receive the SI values through $I_N, I_E, I_S,$ and I_W from the four neighbors.

Step 5: The SSL sets SI to 0 if it received a 0 SI value from any of its neighbors and if the corresponding comparator latch contains the 0 value.

Step 6: All dies with SI equal to 0 are declared fault-free and the remainder are declared faulty.

It is easy to see that the wafer testing procedure described above is equivalent to Algorithm 3.2. Therefore, the results on the performance of Algorithm 3.2 shown in Section IV apply. As we have seen in Section IV, we can achieve a very high diagnostic resolution even when the yield is low, especially when faults are clustered. As nonzero values of β are allowed, the scheme can tolerate symmetric failures caused

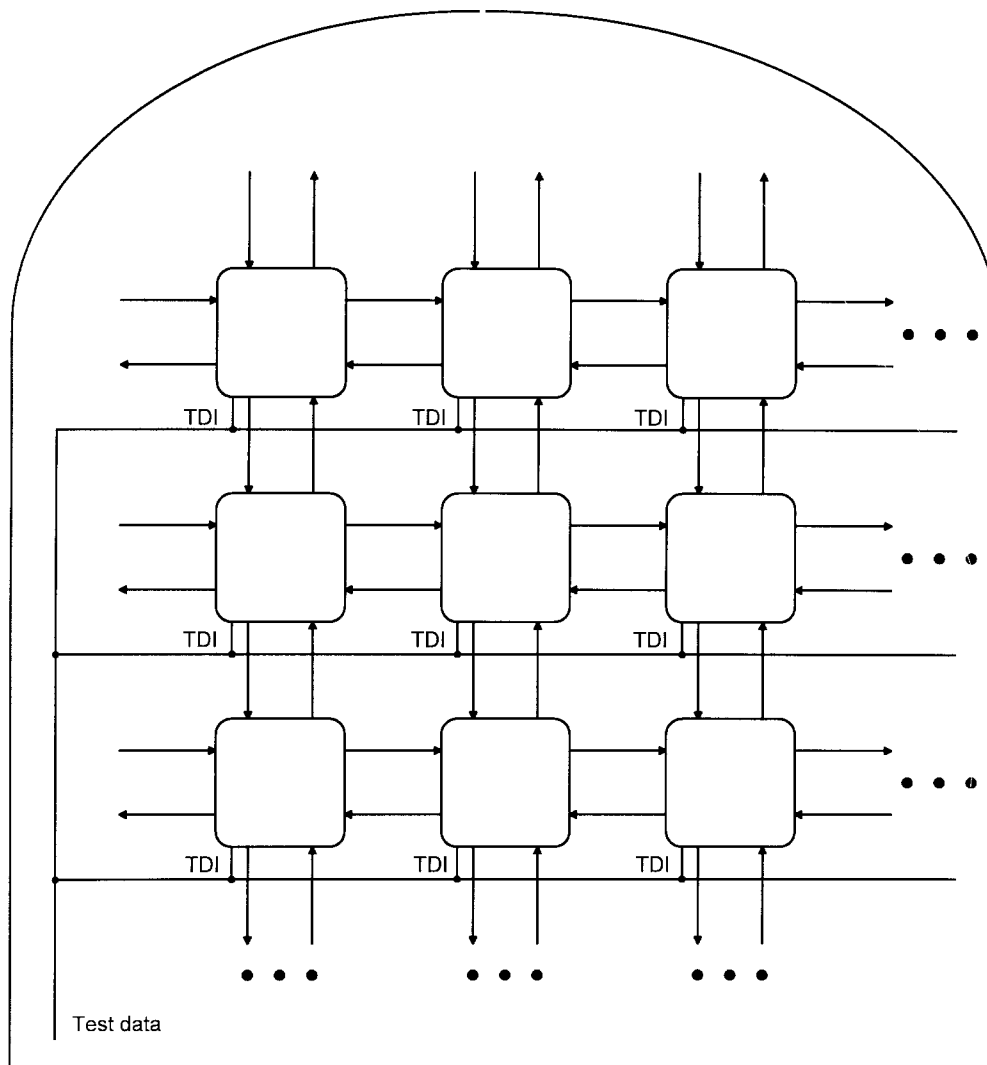


Fig. 6. Test connection schematic.

by systematic fabrication errors to some extent. Note that since test data is broadcast to all dies simultaneously, all comparison tests are performed at the same time and therefore take a constant time proportional to the number of test vectors times the length of the internal scan chain. The diagnosis stage only takes several steps and is negligible as compared to the time spent in comparison testing. This brings about a significant saving of test time. Another advantage of this wafer testing scheme is that it does not require the storage of any fault-free data for comparison.

We note that in our proposed implementation (Fig. 6), since TDI is distributed through a single wire, a fault on that wire could make all chips on a row receive the same faulty patterns thus potentially creating a fraction as large as one row. In the limit the chips could have received no patterns at all. We would like to emphasize that our analysis does not take into account the impact of single point failures causing common misdiagnosis that could fault many good chips or declare good several faulty chips.

We conclude this section with a discussion of certain issues pertaining to the implementation of our wafer testing

scheme, specifically, power dissipation, clock skews, and test time.

Power dissipation during wafer test can be managed by running the test at clock rates well below the operational clock rates for the chip. For CMOS circuits, this results in a proportional reduction in power dissipation. This reduced test vector execution rate is also required by the proposed approach to allow the serial loading of test vectors through the TDI port. Note however from Fig. 6 that while test vectors are loaded into each chip in bit serial fashion, they are broadcast to all chips in parallel. Thus the number of shift clocks needed to load the vectors (and scan out the results for comparison) approximately equals the pin count for a chip. Thus test vector application rates can be expected to be one to two orders of magnitude below operational clock rates. Power dissipation per chip will be scaled down by the same factor.

This reduced test application rate is not a major disadvantage since wafer probe testing is usually DC functional and rarely runs "at speed." Performance data at wafer probe is mostly extracted from parametric measurements from test chips on the wafer.

We also observe that test application rates are limited by the time needed to shift in the test vectors and scan out the results for comparison. These can be sped up somewhat by running a fast shift clock. This is possible because the shift operation is simple and can support high clock rates. Even so, test application rates, an order of magnitude below “at speed” clock rates, can be expected.

Since all comparisons are done among neighboring chips, clock skews will not be a problem if a linear array is tested. However, on a two-dimensional array, the skew grows arbitrarily large with the size of the array. In such cases some special techniques need to be applied to alleviate the problem.

As we mentioned before, the test time saving comes from testing several hundred chips on the wafer in parallel. There is additional saving in that the probe head does not need to be positioned above each chip in turn. This mechanical operation in traditional wafer probe operations can add significantly to total wafer test time. Thus even with reduced test application rates, a factor of ten or better test time saving can be realistically expected.

VI. CONCLUSIONS

We have presented a probabilistic diagnosis algorithm for constant degree structures. The performance of the algorithm is analyzed under a negative binomial failure distribution to account for fault clustering. Closed form solutions are given. The application of this diagnosis algorithm to the production testing of wafers is explored. A simple test structure is given which incorporates a localized version of the diagnosis algorithm for determining the status of each die. With this test structure, all dies can be tested in parallel and therefore a considerable saving in test time is achieved as compared with probe testing. Our diagnosis algorithm is unique in that it is shown to work well in the presence of fault clustering. As faults have been observed to cluster on the wafer, this characteristic is of great practical importance. The scheme works well when the yield is low. The performance of the scheme is insensitive to yield variations when faults are clustered. We have also discussed certain issues and difficulties one may encounter in the implementation of our wafer testing scheme.

ACKNOWLEDGMENT

The authors would like to thank Dr. A. Singh of Auburn University for his help in clarifying implementational issues relating to their wafer testing scheme. They would also like to thank the anonymous referees for their careful reading of the paper and suggestions for improving its quality.

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] S. L. Hakimi and A. T. Amin, “Characterization of connection assignment of diagnosable systems,” *IEEE Trans. Comput.*, vol. C-23, pp. 86–88, Jan. 1974.
- [3] A. T. Dahbura and G. M. Masson, “An $O(n^{2.5})$ fault identification algorithm for diagnosable systems,” *IEEE Trans. Comput.*, vol. C-33, pp. 486–492, June 1984.

- [4] 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.
- [5] K. Chwa and S. L. Hakimi, “Schemes for fault-tolerant computing: A comparison of modularly redundant and t -diagnosable systems,” *Inform. Control*, vol. 49, pp. 212–239, June 1981.
- [6] J. Maeng and M. Malek, “A comparison connection assignment for self-diagnosis of multicomputer systems,” *Dig. Fault Tolerant Comput. Symp.*, vol. 11, pp. 173–175, June 1981.
- [7] 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.
- [8] E. R. Scheinerman, “Almost sure fault tolerance in random graphs,” *SIAM J. Comput.*, vol. 16, pp. 1124–1134, 1987.
- [9] D. M. Blough, “Fault Detection and Diagnosis in Multiprocessor systems,” Ph.D. dissertation, Johns Hopkins Univ., Baltimore, MD, 1988.
- [10] D. Fussell and S. Rangarajan, “Probabilistic diagnosis of multiprocessor systems with arbitrary connectivity,” *Dig. Fault Tolerant Comput. Symp.*, vol. 19, pp. 560–565, 1989.
- [11] S. Rangarajan and D. Fussell, “Probabilistic diagnosis algorithm tailored to system topology,” in *Dig. Fault Tolerant Comput. Symp.*, vol. 21, pp. 230–237, 1991.
- [12] S. Rangarajan, D. Fussell, and M. Malek, “Built-in testing of integrated circuit wafers,” *IEEE Trans. Comput.*, vol. C-39, pp. 195–205, Feb. 1990.
- [13] K. Huang, V. K. Agarwal, L. LaForge, and K. Thulasiraman, “A diagnosis algorithm for constant degree structures and its application to VLSI testing,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 6, pp. 363–372, Apr. 1995.
- [14] K. Huang, “System level diagnosis and wafer testing,” Ph.D. dissertation, McGill Univ., Montreal, Can., Nov. 1992.
- [15] 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.
- [16] C. H. Stapper, F. M. Armstrong, and K. Saji, “Integrated circuit yield statistics,” *Proc. IEEE*, vol. 71, pp. 453–470, Apr. 1983.
- [17] I. Koren, Z. Koren and C. H. Stapper, “A unified negative-binomial distribution for yield analysis of defect-tolerant circuits,” *IEEE Trans. Comput.*, vol. C-42, pp. 724–733, June 1993.
- [18] I. Koren and C. H. Stapper, “Yield models for defect-tolerant vlsi circuits: A review,” in *Defect and Fault Tolerance in VLSI Systems*, vol. 1, I. Koren, Ed. New York: Plenum, 1989, pp. 1–21.
- [19] A. O. Allen, *Probability, Statistics, and Queueing Theory with Computer Science Applications*. New York: Academic, 1978.
- [20] Williams and J. M. Angell, “Enhancing testability of large-scale integrated circuits via test points and additional logic,” *IEEE Trans. Comput.*, vol. 22, pp. 46–59, Jan. 1973.
- [21] P. Agrawal and V. D. Agrawal, “Probabilistic analysis of random test generation method for irredundant combinational logic circuits,” *IEEE Trans. Comput.*, pp. 691–695, July 1975.
- [22] H. D. Schnurmann, E. Lindbloom, and R. G. Carpenter, “The weighted random testpattern generator,” *IEEE Trans. Comput.*, pp. 695–700, July 1975.
- [23] *IEEE Standard Test Access Port and Boundary-Scan Architecture*, 1149.1, May 1990.
- [24] 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.



Kaiyuan Huang received the B.E. degree in computer engineering and the M.E. degree in computer science from Chongqing University, Chongqing, China. He received the Ph.D. degree in electrical engineering from McGill University, Montreal, Canada.

He has worked at Chongqing University, McGill University, Concordia University, Montreal, Canada. Since 1994, he has been with Nortel, Ottawa, Canada. He is the Architect of the SWACT system of the Nortel Spectrum project, which provides WARM standby and hitless switching of activity between spared nodes. He is also the prime designer of the Data Synchronization and Clock Synchronization software for Nortel Spectrum at Nortel. His interests include fault tolerance, system robustness, communication protocols, real-time systems, algorithms and software engineering.

Dr. Huang has received several awards while working at Nortel, such as the Award of Excellence.



Vinod K. Agarwal (F'92) received the Ph.D. degree from Johns Hopkins University, Baltimore, MD.

He is Founder, President, and CEO of LogicVision, Inc. He has over 20 years experience in information technology, microelectronics design, and CAD software. An industry expert in digital testing, he has been active in design-for-test (DFT) and built-in-self-test technologies for over a decade. He is co-inventor of several U.S. patents on BIST technology. He has published over 100 technical papers in international journals and conference proceedings.

He has held the NSERC-NT/BNR Industrial Research chair at McGill University, Montreal, Canada.



K. Thulasiraman (F'90) received the Bachelors and Masters 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, 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 where he served as Chair from 1993-1994. In August 1994 he moved to the University of Oklahoma at Norman as the 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, design and analysis of algorithms, and applications. He has co-authored 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 in *Handbook of Circuits and Filters* (CRC Press, 1995). 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 from 1989 to 1991 and has been a Regional Editor of the *Journal of Circuits, Systems, and Computers* since 1990. He was Technical Program Chair of the IEEE International Symposium on Circuits and Systems in 1993. Recently he was elected Administrative Vice-President of the IEEE CAS Society.