A Coloring Approach to the Structural Diagnosis of Interconnects *

X. T. Chen and F. Lombardi
Department of Computer Science
Texas A&M University
College Station TX 77843

Abstract

This paper presents a new approach for diagnosing stuck-at and short faults in interconnects whose layouts are known. This structural approach exploits different graph coloring and coding techniques to generate a test set with no aliasing and confounding. The conditions for aliasing and confounding are analyzed with respect to the size and number of the shorts in the fault set. The characteristics of unbalanced/balanced codes for encoding the colors in the vector generation process for interconnect diagnosis are discussed and proved using a novel algebra. An algorithm for diagnosis is then presented.

1 Introduction

The high density of today's digital circuits has made possible the manufacturing of complex systems with a substantial reliance on sophisticated interconnect resources. Diagnostic costs have increased considerably for either manufacturing and yield enhancement, or for a customized architecture. Diagnosis consists of fault detection and location and is important because at completion of testing, it permits an efficient repair of chips and boards, thus increasing the yield and decreasing the overall costs.

A characteristic that has been largely ignored in previous papers, is the process of vector generation and in particular, the encoding of the vectors and the overall execution of the diagnosis process. While this is not as important for a behavioral approach [7], it is of primary relevance in a structural technique due to the knowledge of the layout. Also, there is surprising little algebra available for establishing the correct criteria for test vector generation (albeit balanced codes are introduced in [6] for this purpose).

The objective of this paper is to analyze and propose a new structural approach for the diagnosis of shorts in interconnects based on graph coloring techniques (note that the proposed approach is also valid for stuck-at faults). Different coloring techniques such as color mixing, are utilized to generate a test set for diagnosis of the interconnect. An algebra for establishing appropriate codes of the colors for test vector generation is proposed.

***This research supported by A grant from TARP.**

2 Review

There are three types of faults commonly associated with nets: stuck-at faults, open faults and bridge (short) faults. These faults can be tested by using either a *behavioral* or a *structural testing* strategy. Several papers [6] have discussed behavioral testing. Recently in [7] the conditions for optimal diagnosis for a behavioral approach have been given such that the test length is between a lower bound of $\log_2 N$ and an upper bound of $N + 1$, where N is the number of nets depending on the resolution (either full or partial diagnosis), the nature of the testing process (either adaptive or off-line) and the assumed fault model. In behavioral testing, it is commonly postulated that every net on the board can be shorted to any other net. The *Counting Sequence Algorithm* of [2] can be used to detect all bridge faults with a test length of $\lceil \log_2 N \rceil$. The *Sequential Response Vector (SRV)* of a net to a *Sequential Test Vector (STV)* is then used to detect and/or diagnosis shorts between nets. If the SRV of a net differs from its STV, then this vector is referred to as a *fault syndrome.* If a syndrome in the presence of a fault is the same as the fault-free response of a net, then it is impossible to determine whether or not this net is also part of the short. The response in this case is referred to as an *aliasing syndrome.* Similarly, the bridge fault between a pair of nets may produce the same syndrome as between another net pair; so, it is impossible to determine whether or not there is a short between which pair of nets. The response is called a *confounding syndrome.*

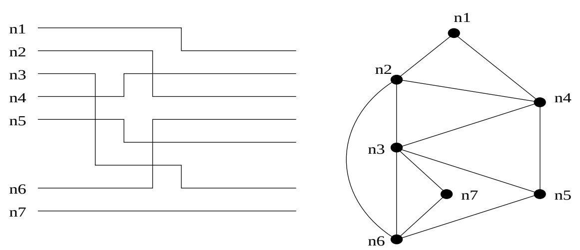
The *walking-1* test set proposed by [1] can avoid aliasing and confounding, i.e. it can be used to diagnose all bridge faults in the nets. The drawback to this method however, is that large test sets are generated as a sequence of length n is required. Test generation is very simple as each parallel test vector (PTV) has only a walking-1.

[4] has considered the restricted, yet realistic scenario in which two nets can be shorted only if they terminate at adjacent pins, or their tracks are adjacent within a predetermined tolerance. This is referred to as the *adjacency fault model.* Diagnosis is shown to be equivalent to the *coloring problem* and a possible solution is proposed. The approach of [4] draws from a previous paper [3] in which structural testing of interconnects is accomplished using coloring.

ICCAD '96

3 Preliminaries

The interconnect consists of N nets arranged in a known layout (given by the arbitrary graph G). Figure (1a) gives an example of an interconnect with $N = 7$. The structure of the interconnect can be modeled by using a so-called adjacency graph $G_{ad} = (V, E)$ in which a node v_i in V represents a net n_i in G and an edge e_{ij} (connecting v_i to v_j) exists in E if and only if n_i is adjacent to n_j in G. Let D denote the maximum degree of the nodes in G_{ad}. Figure (1b) shows the adjacency graph G_{ad} of G of Figure (1a). The following definitions also apply. 1. *Sequential Test Vector (STV):* the binary string, as test data applied to a specific net of the interconnect in the diagnosis process. STV_i denotes the STV for net n_i. 2. *Parallel Test Vector (PTV):* the test data applied to all nets in parallel in one round of the diagnosis process, where the number of rounds is given by the cardinality of the STV. PTV_i denotes the PTV for the ith round in the diagnostic process. 3. PSN *(primary shorting net):* the net which is likely to be shorted with a given net, i.e. for a node n_i in G_{ad} all the nodes v_j such that $e_{ij} \in E$. 4. SSN *(secondary shorting net):* the primary shorting net of a primary shorting net.



(a). Interconnect Graph (b). Adjacency Graph

Figure 1: Example

As fault model, a strictly physical characterization consisting of short faults, is used. The model used in this paper is based on extending the adjacency fault model and adding a further restriction [8]. This restriction is as follows: since most short faults are caused by the excessive metal between the nets or by the failure to remove metal between tracks, then it is much unlikely that a fault will affect two non adjacent nets, without affecting the nets in between. Hence, in the paper it is assumed that: 1. *Adjacency assumption:* a short fault may happen between two nets only if these two nets are physically adjacent. 2. *Continuous assumption:* given two non-adjacent nets n_i, n_j and a subset of nets (denoted by B) between n_i and n_j, if n_i and n_j are shorted, then all nets in B are also shorted together.

Hereafter, the following assumptions are valid in the analysis. 1. The adjacency and continuous assumptions are applicable to shorts as modeled by G_{ad}. The OR short is assumed for simplicity. Note that even though stuck-at faults are not dealt explicitly in this paper, the proposed approaches fully diagnose them. 2. Probing is only allowed at the input and output pins of each net. 3. Every fault must be located at relative ease such that repair or rework can take place, i.e. no aliasing or confounding must exist. 4. $D \ll N$, i.e. G_{ad} is relatively sparse (as commonly found in practice). 5. Diagnosis is executed off-line.

In this paper, graph coloring is used; this problem can be formulated as follows [3]: a k-coloring of an arbitrary graph $G_{ad} = (V, E)$ is a mapping C:V->$\{C_1, C_2, \ldots, C_k\}$ which assigns a color C_i to each node in such a way that no two adjacent nodes receive the same color, i.e. $e_{ij} \in E$ implies that $C(n_i) \neq C(n_j)$. A minimum or optimal coloring of G_{ad} is a $\xi(G_{ad})$-coloring of G_{ad} where $\xi(G_{ad})$ called the chromatic number of G_{ad}, is the least k such that there exists a k-coloring of G_{ad}. No efficient algorithm is known for optimally coloring an arbitrary graph.

4 Fault Detection

Initially, an algorithm of a rather intuitive nature (hence, referred to as simple coloring) is presented for testing an interconnect on a go/no-go basis (fault detection only). The algorithm for fault detection using simple coloring is given as follows.

Algorithm 1: Fault Detection by Simple Coloring.

Step 1: Generate the adjacency list from G.
Step 2: Sort the adjacency list ($D \times N$).
Step 3: From the highest D down to 1 do :
Select the current node;
Check all neighbors of the current node;
Assign the lowest color number as possible;
Flag the current node in the table.

The time complexity of the above algorithm is $O(N^2 + D \times N + D \times N) = O(N^2)$.

Having generated the colors, the tests for fault detection can be generated immediately using a suitable coding arrangement; this means that if for example, a counting sequence is used [2], then the number of tests is given by $T = \lceil \log_2(C + 2) \rceil$, where C is the number of colors found by Algorithm (1).

Note that this coding arrangements permit the detection of stuck-at faults with 100% fault coverage as each net receives at least a 0 or a 1. Figure 2 shows C versus N for randomly generated interconnects (for different values of D). However, Algorithm (1) does not guarantee full diagnosis; so this test set represents only the initial vectors, not the final tests of the interconnect [4].

5 Color Mixing

A further feature of using coloring is the capability of utilizing the colors found by Algorithm (1) for generating new colors (within the assumed fault model). This process is referred in this paper as *color mixing.* Color mixing has been implicitly used in [4] by triangularization; in [4], it was shown that no two SSNs can have the same color for full diagnosis; in [8] it has been proved that this condition is neither sufficient, nor necessary. Color mixing also consists of establishing the conditions by which the syndromes are still distinguishable in their codes. The basic principles of the proposed coding technique are simple: give to most nets the color corresponding to a 0 in the STVs,

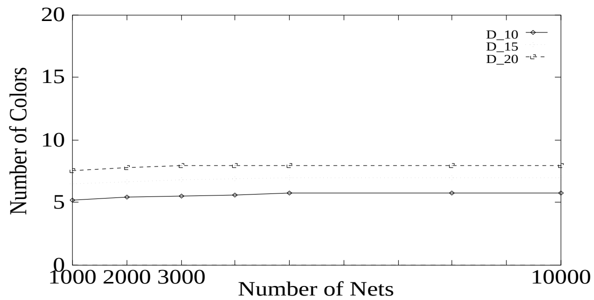


Figure 2: Number of Colors (Detection Only)

while utilizing as few 1's as possible to avoid aliasing and confounding (for AND shorts, the reverse is applicable). These explicitly correspond to the rules by which a STV is constructed in the walking-1 technique by using an unbalanced code [1]. The following Theorem is applicable (the proof follows from the above discussion).

Theorem 1. There is no aliasing in the diagnosis of an interconnect if the test set is generated from a balanced code of the colors found using Algorithm (1).

Note that Theorem (1) is stronger than the results of [6] as it fully characterizes aliasing independent of the assumed fault model as well as encompassing the conditions of [5] for the Counting sequence with complemented tests (as an example of balanced code). Let the number of bits in the balanced code for a STV be denoted by B_{bc}; then, $B_{bc} = 2 \times \lceil \log_2 C \rceil$ where $C \leq D$; Figure 3 has shown that for random interconnects, the cardinality of the test set remains almost constant for $10 \leq D \leq 20$. Therefore, a balanced code is a practical solution for interconnect testing with no aliasing.

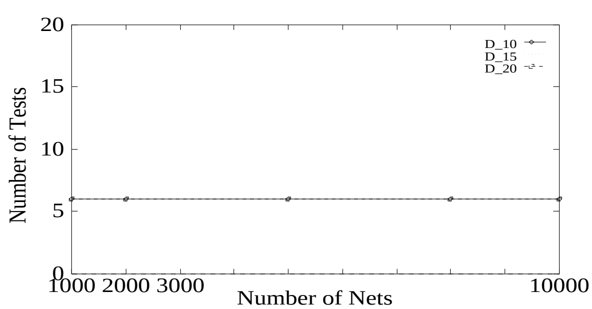


Figure 3: Number of Tests (No Aliasing)

While aliasing is relatively simple to take care, confounding requires a different approach. This process is based on finding a cut of G_{ad} along every edge joining a pair of nodes such that it is possible to distinguish the syndrome (caused by a short involving these two nodes only) and the syndromes caused by the involvement in a short of further nodes on each side of the cut. This yields the following Theorem.

Theorem 2. Confounding syndromes may occur in the diagnosis of an interconnect provided that there exist at least two subsets of nets (which can be shorted with the PSNs and SSNs too) with an equal syndrome prior to the occurrence of the faults (if any).

Let F^i be the number of faulty nets in the ith (disjoint) short, where $F = \sum_{i=0}^{N} F^i$ for $2 \leq F^i \leq N$ and $0 \leq i \leq N$, and $F^{i_{max}} = max\{F^i\}$. Then, the following four cases can be distinguished to establish the proper conditions for no aliasing and confounding in the syndromes of the nets.

Case 1: $F^{i_{max}} = 2$; then a simple coloring technique is sufficient for full diagnosis. This is shown in the example of Figure 4, in which the dotted vertical line identifies the chosen cut across the edge joining the nets with signatures a and b in the adjacency graph.

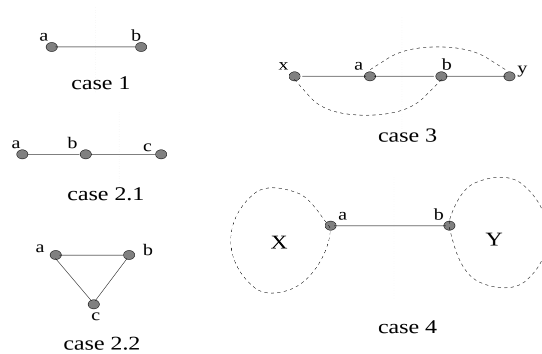


Figure 4: Cases for Theorem

Case 2: $F^{i_{max}} = 3$; then two further subcases (Cases 2.1 and 2.2 in Figure 4 can be identified. Only Case 2.1 must be considered, as Case 2.2 is undiagnosable by Theorem (2). For Case 2.1, diagnosis corresponds to check the condition aORb=c for the syndromes of these three nets. By Theorem (1), if a balanced code is used, no aliasing will be encountered. This also corresponds to no confounding, because $F^{i_{max}} = 3$.

Case 3: $F^{i_{max}} = 4$; confounding can only happen between the nets with signatures a and b for the case shown in Figure 4. This corresponds to the condition by comparing aORx with bORy in the syndromes. Using coloring, $a \neq b$ for the signatures; so, confounding will be present provided the syndromes yield $x = b$ AND $y = a$.

Case 4: $F^{i_{max}} > 4$; then, this corresponds to checking the condition by comparing XORa with YORb for the syndromes, where X and Y are syndromes for two subsets of nets (i.e. they can be themselves generated by a mixture of colors). Let *cover* ($\overline{cover}$) denote the bit-wise covering (not covering) operation by 1's for two strings (as corresponding to the signatures of the colors), i.e. (101010) *cover* (001010) and (100010)$\overline{cover}$(001010). Three further subcases are possible: a. $X\overline{cover}a$ AND $Y\overline{cover}b$; there will be no confounding if and only if $X \neq b$OR$Y \neq a$ (same as Case 3 above). b. $x cover a$ AND $Y cover b$; there will be no confounding if and only if $X \neq Y$. c. $X cover a$ AND $Y\overline{cover}b$; there will be no confounding if and only if $X \not\equiv b$ORY, i.e. $X\overline{coverb}$.

According to the above algebra, it is possible to establish the rules for diagnosis by coloring with relation to the value of $F^{i_{max}}$; these are given as follows: 1. If $F^{i_{max}} \leq 3$, use simple coloring and balanced codes for test vector generation. 2. If $F^{i_{max}} \geq 4$, use coloring in each possible disjoint faulty set and then use walking-1 for color coding.

6 Diagnosis

Hereafter, in the analysis, $F^{i_{max}}$ will be restricted to shorts involving the PSNs and SSNs, i.e. for a value up to $O(D^2)$ if all PSNs and SSN are shorted together. The basic principles of the proposed approach to full diagnosis are as follows: use simple coloring (and Algorithm (1)) with a balanced code first. This test set is used for fault detection using G_{ad}; the codes for the tests are then checked for aliasing and confounding. Then, color mixing and a walking-1 code are used to avoid aliasing/confounding (if required) by modifying the codes for the tests involved. The algorithm for full diagnosis is given as follows.

Algorithm 2: Diagnosis by Coloring.

Step 1: Generate the adjacency list from G.

Step 2: Sort the adjacency list of $D \times N$.

Step 3: From the highest D down to 1, choose the current node.

Step 4: Execute Algorithm (1) for simple coloring (i.e. to check all the PSNs for the current node). Find the lowest color number as the candidate color.

Step 5: Use walking-1 coding for the candidate color and recursively check the SSNs for confounding according to the four cases described in Section (5).

Step 6: If there is confounding for the assumed value of $F^{i_{max}}$, go to Step (4).

Step 7: Assign the color and flag the current node in the table.

Step 8: If the PSNs have the same (highest) degree and are not colored, denote the PSNs as current nodes and go to Step (3).

Step 9: If there is any uncolored node, go to Step (3); else, exit.

The time complexity of Algorithm (2) is $O(N^2) + O(N \times D \times D! \times D!)$. This time complexity is however not acceptable for implementation, because the second term (and in particular the factorization of D) will make the whole process impractical (even for moderate values of D). So, a novel approach has been used in the implementation of Algorithm (2); this has reduced the time complexity to $O(N^2) + O(N \times D^3)$. For a walking-1 code, all the different colors of the PSNs can be represented by submatrices of size $D \times D'$, where D' is the number of colors used and is proportional to D. Assume without loss of generality that this matrix is an identical matrix (denoted as I_D), as $D' = O(D)$. Therefore, the operation for a possible color mixing is given by an OR of unit vectors in the permutation matrix of I_D. The worst case of a syndrome for color mixing involves all the D unit vectors of size D, resulting in a new vector (denoted as V_{new}) of size D with more than one 1's due to the OR operation. For example (100 ... 000)OR(010 ... 000)OR ... OR(000 ... 010) results in V_{new}=(110 ... 010).

As each node has this type of syndrome and it is required to check at most D times for confounding in all PSNs, then if only some of the PSNs are shorted, there are $D!$ cases for each node and $D!$ checks for verifying all possible mixing cases for D unit vectors. However, it is possible to check the $D!$ cases for the PSNs through the worst case vector V_{new}.

If all possible subsets of V_{new} of two nodes are different, there no confounding is possible between them (by definition). So instead of considering $D!$ combinations of D unit vectors in the permutation matrix of I_D, all the cases with at most D 1's in a vector are checked. This needs only one (bit-wise) OR operation provided the word size of the computer is larger than D'. To differentiate each of these D cases, the concept of *mailbox* is introduced. A mailbox stores the value of the color between the node and a PSN. Different mailboxes are introduced in every node for each PSN to record the color mixed through that PSN. The mailbox between a node v_i and a PSN v_j is denoted as $v_i\_v_j$. Due to the coding employed in the proposed approach, its value is given by the color of v_j. As there are only $O(D)$ PSNs and $O(D)$ SSNs for each PSN, then only $O(D) \times O(D)$ cases must be checked for each node (instead of $O(D!) \times O(D!)$). Hence, the overall time complexity in the implementation of Algorithm (2) is now $O(max\{N^2, N \times D^3\})$

A further issue analyzed in this Section is the exact bounds for D' when the faults are restricted to PSNs and SSNs. The worst case of D' using simple coloring is given by $D + 1$. If the shorts with the PSNs are considered, then $2 \times (D + 1)$ colors are sufficient for differentiating all cases, i.e. at most $D + 1$ new colors must be added to differentiate PSNs and SSNs by a similar argument.

7 An Example

Consider the interconnect shown in Figure (1a); its adjacency graph is shown in Figure (1b). In Step (1), the degree of each node is given as follows: $\{n_1, n_7\}$=>2, $\{n_2, n_4, n_6\}$=>4, $\{n_3\}$=>5, $\{n_5\}$=>3. In Step (2), the nodes are sorted in ascending order of degree, as follows: $\{n_1, n_7\}$, $\{n_5\}$, $\{n_2, n_4, n_6\}$, $\{n_3\}$.

So, by executing simple coloring in Algorithm (1), a walking-1 code for each color is applicable as follows: n_3=00000001, n_2=00000010, n_4=00000100. Hence, the mailboxes are as follows: $n_2\_n_3$=1, $n_2\_n_4$=100, $n_3\_n_2$=10, $n_3\_n_4$=100, $n_4\_n_2$=10, $n_4\_n_3$=1.

If color mixing is restricted to PSNs, then Case 2.2 is avoided by assumption; as, there is no confounding under Case 2.1, then n_6=00000100 by simple coloring. The mailboxes are $n_2\_n_6$=$n_3\_n_6$=100. So by using simple coloring, there will no confounding among n_3, n_2, n_4 and n_6 (as by Case 3).

Again, using simple coloring n_5=00000010, n_1=00000001, n_7=00000010. Therefore, the number of colors is 3, i.e. 1 (01), 2 (10) and 3 (11); only two PTVs are required. The test set for fault detection (T_{fd}) using binary coding is given in Table (1). The test set with no aliasing (T_{noa}) is also given in Table (2) using four PTVs with balanced coding (the code

Table 1:

n_i	Color	$T_{fd}(STV_i(\text{->}))$	$T_{noa}(STV_i(\text{->}))$	$T_{noac}(STV_i(\text{->}))$
1	1	01	0110	10000
2	2	10	1001	01000
3	1	01	0110	10000
4	3	11	1010	00100
5	2	10	1001	00010
6	3	11	1010	00100
7	2	10	1001	00001

01 is used for color 0 and 10 for color 1 in Table (2), or vice versa).

For no confounding, all PSNs and SSNs of each node must be checked according to Case 4. If n_5=00000010, then for n_2 (corresponding to node a in Case 4) and n_4 (corresponding to b), a=00000010 and b=00000100. So, $X = n_2\_n_3ORn_2\_n_6 = 101$ and $Y = n_4\_n_3ORn_4\_n_5 = 011$ In the worst case, this corresponds to $aORX = bORY = 111$ while for all other cases, $aORn_2\_n_6 = bORn_4\_n_5 = 110$

Therefore, confounding is possible. Using Algorithm (2), n_5=00001000, n_1=00000001, n_7=00010000. So, 5 colors are required for no confounding among PSNs and SSNs. Five tests are required using a walking-1 coding as shown in Table (1).

8 Simulation Results

The proposed approach to diagnosis has been evaluated by simulation on different size random interconnects by varying D. Figure 5 shows the number of tests for full diagnosis (no aliasing/confounding) under the worst case, i.e. all neighbors (PSNs and SSNs) of each node are shorted.

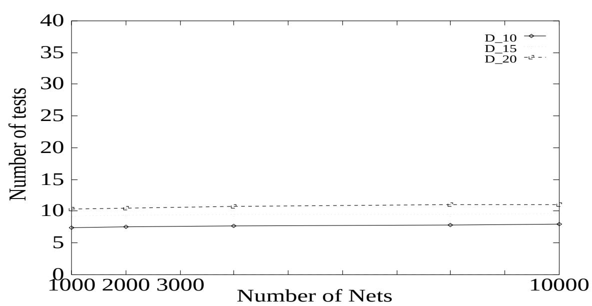


Figure 5: Number of Colors versus Number of Nets (Full Diagnosis under Worst Case Condition)

The simulation results for all cases are shown in Figure 6. Even though the complexity of the algorithm is the same, Section (7) has proved that the number of tests is upper bounded by $2D$+2 (using a test per color as in a walking-1 code). These results show that the average number of tests generated by the proposed algorithm, is 70% of the theoretical upper bound.

9 Conclusions

This paper has proposed a new structural approach to full diagnosis (detection and location with no aliasing and confounding) of shorts in interconnects. The proposed approach utilizes graph coloring techniques and appropriate codes to generate a test set based on the adjacency and continuous assumption of [4]. By simulation, it has been shown that for benchmarks and random interconnects this approach requires a significant smaller number of tests than previous approaches.

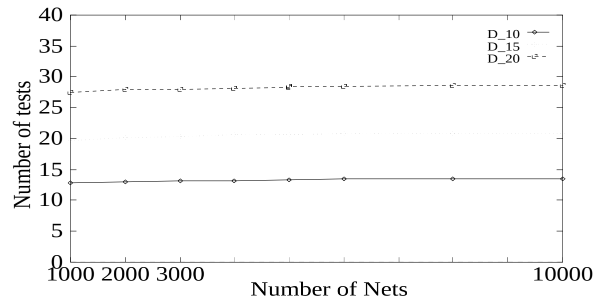


Figure 6: Number of Colors versus Number of Nets (Full Diagnosis under All Cases)

References

[1] Hassan, A., J. Rajski and V.K. Agrawal, "Testing and Diagnosis of Interconnects using Boundary-scan", *International Test Conference,* pp.126-137, 1985.

[2] Kautz, W. H., "Testing for Fault in Wiring Networks", *IEEE Transactions on Computers,* Vol. C-23, No. 4, pp. 358 - 363, 1974.

[3] Garey, M., Johnson, D. and H. So, "An Application of Graph Coloring to Printed Circuit Testing," *IEEE Trans. on Circuits and Systems,* Vol. CAS23, No. 10, pp. 591-599, 1976.

[4] McBean, D. and W. R. Moore, "Testing Interconnects: A Pin Adjacency Approach", *IEEE European Test Conference,* pp. 484 - 490, 1993.

[5] Wagner, P.T., "Interconnect Testing with Boundary Scan", *International Test Conference,* pp.52-57, 1987.

[6] Yau, C. W. and N. Jarwala, "A Unified Theory for Designing Optimal Test Generation and Diagnosis Algorithms for Board Interconnects", *International Test Conference,* pp. 71 - 77, 1989.

[7] Shi, W. and K. Fuchs, "Optimal Interconnect Diagnosis of Wiring Networks," *IEEE Trans. on VLSI Systems,* to appear.

[8] Feng, C., W.K. Huang and F. Lombardi, "A New Diagnosis Approach for Short Faults in Interconnects," *Proc. IEEE FTCS,* pp. 331-339, Pasadena, June 1995.

[9] Cheng, W.T., Lewandowski, J.L. and E. Wu, "Optimal Diagnostic Methods for Wiring Interconnects," *IEEE Trans. on CAD of ICAS,* Vol. CAD11, No. 9, pp. 1161-1166, 1992.