Optimal Algorithms for Substrate Testing in Multi-Chip Modules*

Andrew B. Kahng, Gabriel Robins† and Elizabeth A. Walkup‡

Department of Computer Science, UCLA, Los Angeles, CA 90024-1596
†Department of Computer Science, University of Virginia, Charlottesville, VA, 22903-2442
‡Department of Computer Science, University of Washington, Seattle, WA 98195

Abstract

Multi-chip module (MCM) packaging techniques present several new technical challenges, notably substrate testing. We formulate MCM substrate testing as a problem of connectivity verification in trees via *k-probes*, and present a linear-time algorithm which computes a minimum set of probes achieving complete open fault coverage. Since actual substrate testing also involves scheduling probe operations, we formulate efficient probe scheduling as a special type of metric traveling salesman optimization and give a provably-good heuristic. Empirical results using both random and industry benchmarks demonstrate reductions in testing costs of up to 21% over previous methods. We conclude with generalizations to alternate probe technologies and several open problems.

Key words: MCMs, testing, graph algorithms, fault detection, circuit probing, TSP.

1 Introduction

Multi-chip module (MCM) technology has recently emerged as an economically viable means for packaging complex, high-performance systems [2] [11] [21] [24]. Traditionally, system performance is limited by interconnection delays at the upper levels of the hierarchy (e.g., printed circuit board or backplane), and may be improved by increasing circuit density and die size. However, as we approach wafer-scale integration, poor manufacturing yield and incompatibility with mixed technologies make such a monolithic system implementation unattractive. The MCM approach resolves this dilemma, improving circuit density and yield while decreasing interconnect delay.

MCMs eliminate individual integrated circuit (IC) packages, allowing dies to be situated closer together. This shortens interconnect length and enables up to a three-fold increase in clock frequency, a seven-fold decrease in area, and a 30% decrease in power consumption over the best values achievable

*Andrew B. Kahng is supported by NSF Young Investigator Award MIP-9257982. Gabriel Robins is supported by NSF Young Investigator Award MIP-9457412. Elizabeth A. Walkup is supported by an NSF Graduate Fellowship. Additional related papers may be found at http://www.cs.virginia.edu/~robins/ and http://ballade.cs.ucla.edu:8080/~abk/.

using high-density printed circuit boards (PCBs) [8]. A typical MCM (see Figure 1(a)) consists of a substrate containing inter-chip wiring, upon which are mounted a number of bare die. The MCM substrate is made of silicon, alumina, or cofired ceramic, and usually consists of multiple layers (up to thirty or more wiring layers). The bare die are bonded to pads on the upper-most "chip layer" of the substrate using solder bumps or tape-automated bonding (TAB) technology [22].

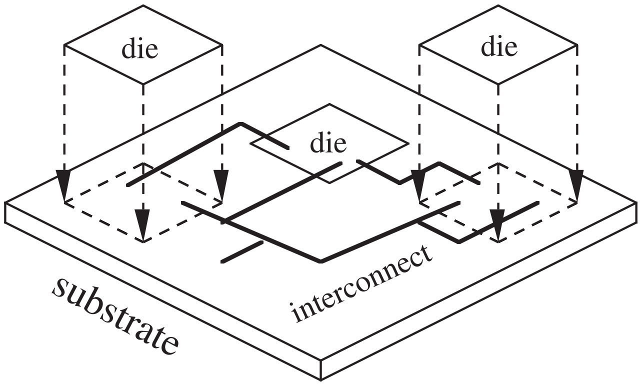

Figure 1: An example of a multi-chip module, showing the underlying substrate containing interconnect, as well as several mounted die.

The increased use of multi-chip module packaging for large, high-performance systems has focused attention on several new and challenging CAD problems, especially those related to layout, thermal reliability, and testing [9] [21]. Testing in particular presents one of the most persistent challenges of the MCM approach [1] [11] [24]. It is desirable to discover defects in the MCM substrate as early as possible, since the cost of locating and fixing a system fault increases geometrically with each successive stage of the system manufacturing and marketing process. Certainly, the fully-assembled MCM package can be tested using combinatorial IC testing techniques. However, the pre-assembly MCM substrate contains a set of disjoint wiring connections with no active devices; thus, the substrate cannot be tested using conventional techniques. With this in mind, we address verification of electrical connectivity in MCM substrates.

We model interconnect in the MCM substrate as follows. A *net* is a set of pins p_i that are to be electrically connected. Each signal net is routed on multiple routing layers using a tree topology, where we assume without loss of generality that each leaf is a net terminal, each edge is a wire segment on a single wiring layer, and each internal node is a *via* between two or more routing layers (Figure 2). We wish to verify that the routing topology of each net is properly implemented, with no faults.

Two fault classes are of interest in MCM substrate testing: open faults, and short faults. An open fault is an electrical disconnection between two points that are to be connected. As will be discussed in

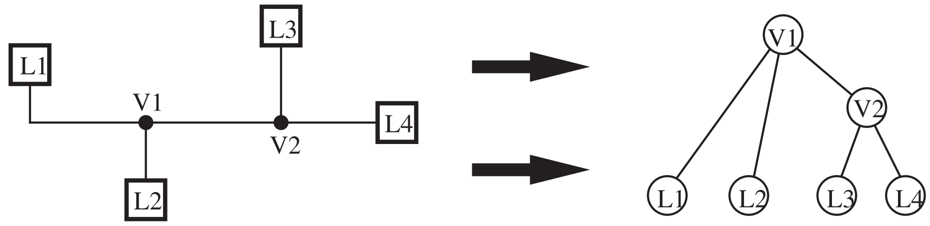

Figure 2: A sample net (left) and its corresponding tree representation (right); pins become leaf nodes while vias become internal nodes.

Section 2 below, there are two types of open faults: wire opens, which correspond to edge failures in the tree topology, and cracked vias, which correspond to a physical form of node failure which arbitrarily disconnects subtrees and is not necessarily detected by tests designed to cover wire opens. A short fault is defined to be an electrical connection between two nets that are *not* intended to be connected.

Traditional methods for connectivity checking involve either parallel probing of the circuit, or combinatorial exercising of the logic, neither of which apply to MCM substrate testing [2]. In verifying connectivity for PCBs, a bed-of-nails tester will simultaneously access every grid point, yielding an efficient, parallel checking procedure. However, this idea cannot be applied to MCMs as feature sizes are too small to allow such a grid-based methodology. A combinatorial approach, e.g., the boundary-scan method for hierarchical design, requires system-specific, built-in test circuitry [10] [23]. In general, this method will apply only to a completely assembled MCM, but not to a substrate which contains isolated interconnect with no active circuit elements.

Several groups have recently proposed new methods for verifying circuit connectivity during MCM manufacturing. Each of these new methodologies relies on *sequential probing* of the MCM substrate, in contrast to the standard approaches above which use parallel probing or combinatorial testing.

Golladay et al. [9] propose an electron-beam method to test MCM substrates for short/open faults by injecting charge into individual nets and then scanning them for faults. Unfortunately, electron-beam testers typically have a relatively small working window of access to the chip/substrate, so that probing a location outside that window requires physical motion of an apparatus. Also, an electron-beam may require a long time to charge up large nets, so that a testing methodology based on this process can be prohibitively slow [20].

All other sequential probing approaches involve variants of k-probe testing, where k "flying" probe heads simultaneously move around the circuit, measuring resistance and capacitance values to determine the existence of shorts between pairs of nets and opens between two pins of a single net. Formally, we define a *k-probe* to be a set of k distinct net terminals which are visited simultaneously by k movable probe heads (typical probe technology uses $k = 2$, but probe machines with higher values of k are under development [19]). A single k-probe simultaneously verifies all $\binom{k}{2}$ paths between pairs of terminals in the probe set by measuring resistance and capacitance values. For example, when $k = 2$ the unique path between the two terminals is checked (Figure 3).

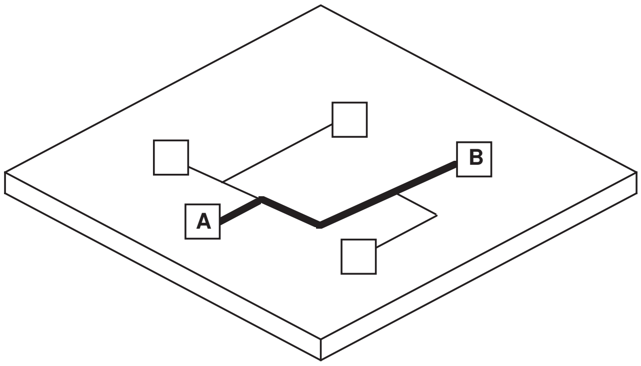

Figure 3: Testing the interconnect on the subsrate using probes: for example, the (A,B) probe tests the shown A-B path for open faults.

A 2-probe sequential testing approach developed by Crowell et al. [7] for bare-board testing has been adopted by some MCM manufacturers [17]. The method of Crowell et al. uses only one probe for each net in the layout, placing the probe heads on the two pins of the net which are physically farthest apart. Unless the measured resistance deviates significantly from the value predicted for the correct circuit, one assumes that no open fault exists. Similarly, only when capacitance is far from the predicted value will a possible short fault between two nets be investigated more carefully.

The algorithm of Crowell et al. [7] is efficient in that it uses just one probing operation per net. However, an unfortunate choice of probe locations may yield measured capacitance and resistance very similar to the predicted values, even in the presence of a fault. For example, an open fault caused by a disconnected pad will be detected only by directly probing a path through the pad itself; probing any other path will fail to notice the small deviation in capacitance and resistance values. Indeed, the number of pads in the net induces a lower bound on the number of probe operations needed for fault coverage.

The incomplete fault coverage afforded by such methods as [7] is economically unacceptable. Thus, MCM manufacturers are now adopting substrate test methodologies which provide complete open fault coverage for all nets [19] [25]. With this in mind, Yao et al. [25] have proposed a quadratic-time algorithm that determines a set of 2-probes which will check for all possible open faults (this work was later further developed in [3] [4] [5] [6] [26] [27]). In this method, sufficient capacitance measurements are taken during the open fault checking process to determine whether two nets have been shorted together (i.e., we will encounter a capacitance value that is too high) [7] [25]. Thus, the remainder of this discussion is confined to the issue of complete open fault coverage. In this paper, we give a linear-time algorithm which for any $k \geq 2$ determines a k-probe set which accomplishes complete open fault coverage of each net, using the minimum possible number of probes.

Once probes are found which adequately test the required classes of open faults, we must schedule the probes for execution by a mechanical tester. Obtaining a good schedule is critical, especially with large production runs. Previous groups [7] [25] have used generic greedy or iterative traveling salesman heuristics to attack this problem. We propose two effective heuristics for probe scheduling based on new observations concerning the metricity and allowable structure of the probe set.

The remainder of this paper is organized as follows. Section 2 formulates optimal open fault detection as a *tree testing* problem and present linear-time algorithms which find an optimal number of probes to cover all possible open faults. Section 3 shows that probe scheduling to minimize total travel time is a form of metric traveling salesman problem (TSP); we present two effective heuristics, one of which has small constant-factor error bound for scheduling any given set of probes. Section 4 gives experimental results on random and industry benchmark layouts, and Section 5 concludes with directions for future research. Preliminary versions of this work have appeared in [13] [14] [18].

2 Open Fault Detection

In this section we address the following problem:

Minimal Probe Generation (MPG) Problem: Given a routing topology for a signal net with l leaves (i.e., pins), determine a minimum set of k-probes needed to verify the net routing.

We consider two levels of open fault coverage: (i) coverage of all open faults on wire segments, and (ii) coverage of all open faults on wire segments and "cracked" vias (see Subsection 2.2). This section presents optimal solutions for the two corresponding versions of the MPG problem. Due to the nature of current probing technology, the discussion assumes $k = 2$; extensions to arbitrary k are straightforward.

2.1 Optimal Detection of Wire Open Faults

In order to test the integrity of all wire segments, every segment which is incident to a pin must be tested. Thus, the number of pins l (leaves in the routing topology) induces a lower bound of $\lceil \frac{l}{2} \rceil$ probes when $k = 2$. Our first probe generation algorithm orders the pins of a net as $p_1, \ldots, p_l$ via an arbitrary in-order traversal of the routing tree. Choosing the $\lfloor \frac{l}{2} \rfloor$ probes $\{p_i, p_{i+\lfloor \frac{l}{2} \rfloor}\}$, $1 \le i \le \lfloor \frac{l}{2} \rfloor$, will cover all edges of the tree, as illustrated in Figure 4; if l is odd, an additional probe $\{p_1, p_l\}$ is generated. Figure 5 gives a formal description of the algorithm, which we call PROBE1.

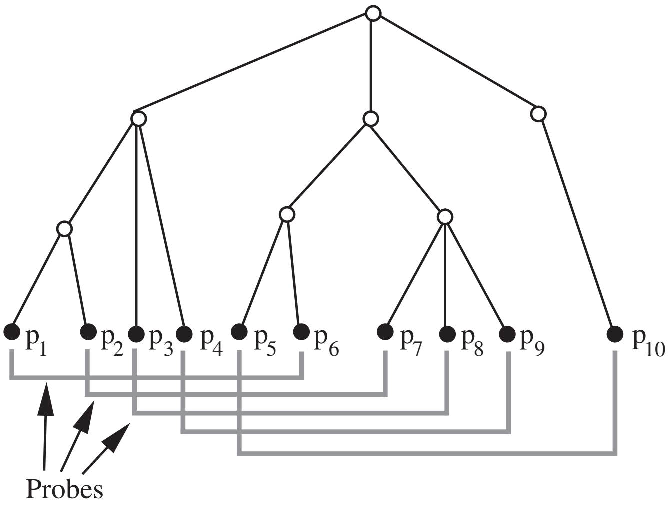

Figure 4: Selecting a minimal set of probes to detect the existence of any wire open faults. The probes $\{p_i, p_{i+\lfloor \frac{l}{2} \rfloor}\}$, $1 \le i \le \lfloor \frac{l}{2} \rfloor$, provide complete wire open fault coverage.

PROBE1: Computing a minimum probe set for edge fault detection	
Input: Tree $T = (V, E)$ with l leaves $p_1, p_2, \ldots, p_l \in V$ **Output:** Minimum probe set which detects all possible edge faults	
1:	Root the tree arbitrarily at an internal node
2:	Induce an in-order labeling $p_1, \ldots, p_l$ of the leaves
3:	Output the set of probes $\{(p_i, p_{i+\lfloor \frac{l}{2} \rfloor}) \mid 1 \le i \le \lfloor \frac{l}{2} \rfloor\}$
4:	**If** l is odd **Then** output the probe $\{p_1, p_l\}$

Figure 5: **PROBE1:** Generation of minimum probe set for edge fault detection.

PROBE1 is time-optimal because it requires time linear in the size of the input tree T. Optimality of PROBE1 in terms of its probe set size follows from two simple observations.

Lemma 2.1 *Every edge which is incident to a leaf node must be tested, implying a lower bound of* $\lceil \frac{l}{2} \rceil$ *for the size of a probe set which detects all possible edge faults.* $\square$

Lemma 2.2 *For any edge* $e = \{v_i, v_{i'}\}$ *in* T *where* v_i *is the father of* $v_{i'}$*, let* T' *denote the proper* *subtree of T rooted at $v_{i'}$. Then algorithm* PROBE1 *outputs some probe which connects a leaf in T' to a leaf not in T'.*

Proof: Assume toward a contradiction that every leaf $v_{j'}$ in T' is connected by a probe edge to $v_{k'}$ which is also in T'. Because the set of leaves in T' is nonempty, the in-order labeling implies that leaves $p_{\lfloor \frac{l}{2} \rfloor}$ and $p_{\lfloor \frac{l}{2} \rfloor+1}$ are also in T'. Our assumption then implies that p_1 is also in T', along with p_l when l is even or p_{l-1} when l is odd. For l even, T' thus contains all leaves of T, contradicting the fact that T' is a proper subtree of T. For l odd, any probe which tests the sole leaf p_l not in T' must connect p_l to some leaf in T', contradicting the assumption that probes involving leaves of T' are internal to T'. □

Theorem 2.3 *Given an interconnection tree T with l leaves, $\lceil \frac{l}{2} \rceil$ probes are necessary and sufficient for detection of all possible edge faults in T.*

Proof: Necessity follows from Lemma 2.1. To see sufficiency, consider the graph G formed by adding "probe edges" to T, i.e., add an edge to T between each pair of leaves that corresponds to a probe output by PROBE1. Note that a set of probes will detect all possible edge faults if every edge $e \in T$ lies on some simple cycle in G that contains only one probe edge. As in the statement of Lemma 2.2, for any arbitrary edge $e = \{v_i, v_{i'}\}$ in T with v_i the father of $v_{i'}$, let T' denote the proper subtree of T rooted at $v_{i'}$. By Lemma 2.2, PROBE1 outputs some probe (p_j, p_k) with $p_j \in T'$ and $p_k \notin T'$. Then, the cycle $v_{i'}, \ldots, p_j, p_k, \ldots, v_m, \ldots, v_i, v_{i'}$ in G (v_m is the lowest common ancestor of v_i and p_k in T) contains edge e (Figure 6). Because PROBE1 outputs only $\lceil \frac{l}{2} \rceil$ probes, this number of probes is sufficient (and algorithm PROBE1 is optimal). □

2.2 Detection of Cracked Via Faults

In manufacturing the MCM substrate, a via can physically "crack" due to factors such as misalignment in lithography and thermal stress. In other words, subtrees rooted at this internal node of the net can become electrically separated (see Figure 7) [25], so that certain sets of probes will detect this open fault, while other sets will fail to find the cracked via. This section gives a linear-time algorithm, called PROBE2, that tests for *both* wire faults and cracked vias using the minimum possible number of probes.

PROBE2 (Figure 8) first chooses an internal node R of maximum degree d and then roots the tree at R by orienting all edges towards R. Each leaf p_i is given the *label* i. The algorithm propagates *message lists* containing labels, starting from the leaves in bottom-up order. Initially, each leaf p_i sends

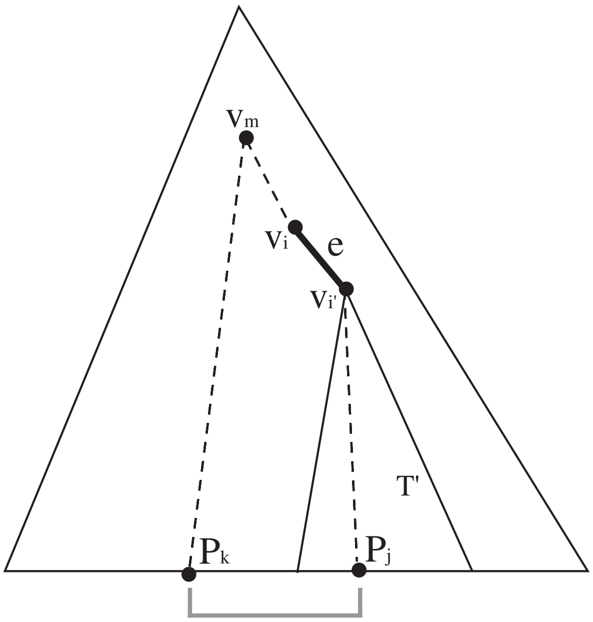

Figure 6: PROBE1 checks each edge $e = \{v_i, v_{i'}\}$ for a fault using the probe (p_j, p_k), whose probe edge completes a cycle containing e.

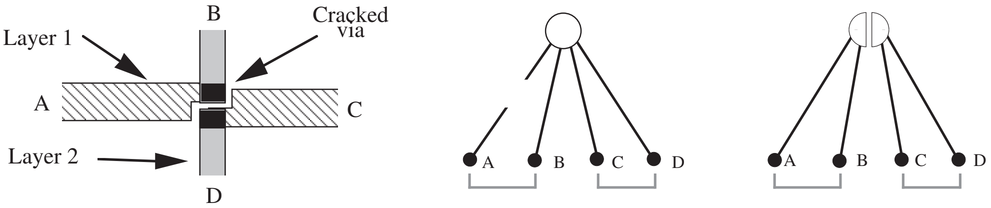

Figure 7: A cracked via in a routing: the two routing layers are depicted using different shadings, while the cracked via (depicted in black) disconnects the circuit as shown (left). The pair of probes $\{(A, B), (C, D)\}$ will detect all edge faults (middle), but may fail to detect some node faults arising from cracked vias (right).

to its parent a message list of size one, i.e., containing only is own label i. Phase I of the algorithm (lines 4-13) pertains to internal nodes $v \neq R$: when such a node v has received message lists from all of its children, it iteratively generates a probe by pairing two labels from distinct incoming message lists as long as one of these lists is of size > 1; the two labels are then deleted from these lists. After the total number of labels in the incoming message lists has been reduced to d or less, all remaining labels are concatenated into a single message list that is passed up to v's parent. When only the root R remains unprocessed, Phase II (lines 14-23) performs a simple cleanup step. Figure 9 traces the execution of PROBE2 on a small example. The algorithm statement in Figure 8 allows non-deterministic choice of label pairings, e.g., at lines 7-9. This can easily be made deterministic; however, our optimality result is stronger since it holds even for the given (non-deterministic) PROBE2 statement.

PROBE2: Computing a minimum probe set for edge and node fault detection

Input: Tree $T = (V, E)$ with l leaves $p_1, p_2, \ldots, p_l \in V$
Output: Minimum probe set which detects all possible edge and node faults

1: Let $W = V$
2: Find internal node $R \in W$ with maximum degree d
3: Root T by directing all edges towards R

/* Phase I: processing internal nodes other than R */

4: **For** $i = 1$ to l, send message list $\{i\}$ from p_i to $parent(p_i)$
5: **While** $\exists v \in W$, $v \neq R$ having received message lists $M_1, \ldots, M_{deg(v)-1}$
6:     **While** $\Sigma_{k=1}^{deg(v)-1} |M_k| > d$
    /* note that this implies $\exists i \ni |M_i| \geq 2$ */
7:         Choose arbitrary $x \in M_i$ for some M_i with $|M_i| \geq 2$
8:         Choose arbitrary $y \in M_j$ for some $j \neq i$, $|M_j| \geq 1$
9:         Output probe (p_x, p_y)
10:         $M_i = M_i - \{x\}$; $M_j = M_j - \{y\}$
11:     Concatenate message lists: $L = M_1 \cup \ldots \cup M_{deg(v)-1}$
12:     Send message list L to $parent(v)$
13:     $W = W - \{v\}$

/* Phase II: $W = \{R\}$; processing R which has received message lists $M_1, \ldots, M_d$ from its children */

14: **While** there are at least 2 nonempty message lists with one having size ≥ 2
15:     Reorder $M_1, \ldots, M_d$ such that $|M_i| \geq |M_{i+1}|$ for all $1 \leq i < d$
16:     Find maximum index $k \leq d$ such that $|M_k| > 0$ /* M_k is smallest non-empty list */
17:     Choose arbitrary $x \in M_1$, $y \in M_k$
18:     Output probe (p_x, p_y)
19:     $M_1 = M_1 - \{x\}$; $M_k = M_k - \{y\}$
20: Concatenate message lists: $L = M_1 \cup \ldots \cup M_d$
21: **If** $|L| > 1$ **Then** output probes (p_{L_1}, p_{L_i}) $\forall\ 2 \leq i \leq |L|$ and terminate
    /* note that L_k denotes the label at position k in the concatenated list L */
22:         **Else** choose leaf node p_i such that i, L_1 were not both passed by the same child of R
23:             Output probe (p_i, p_{L_1}) and terminate

Figure 8: **PROBE2:** Optimal detection of all edge and node faults.

Except at the root, each probe generated by PROBE2 will remove two distinct labels from the message lists being passed. At most d labels will remain to be processed at the root R, requiring at most $d-1$ additional probes. Therefore, to test an interconnection tree with l leaves, PROBE2 uses at most $\frac{l-d}{2} + (d-1) = \frac{l}{2} + \frac{d}{2} - 1$ probes; this bound is tight for, e.g., a star topology. Using a sequence of technical lemmas [13] [18] we can prove that (i) that PROBE2 outputs a probe set which detects all possible edge and node faults, and (ii) that PROBE2 uses the minimum possible number of probes for

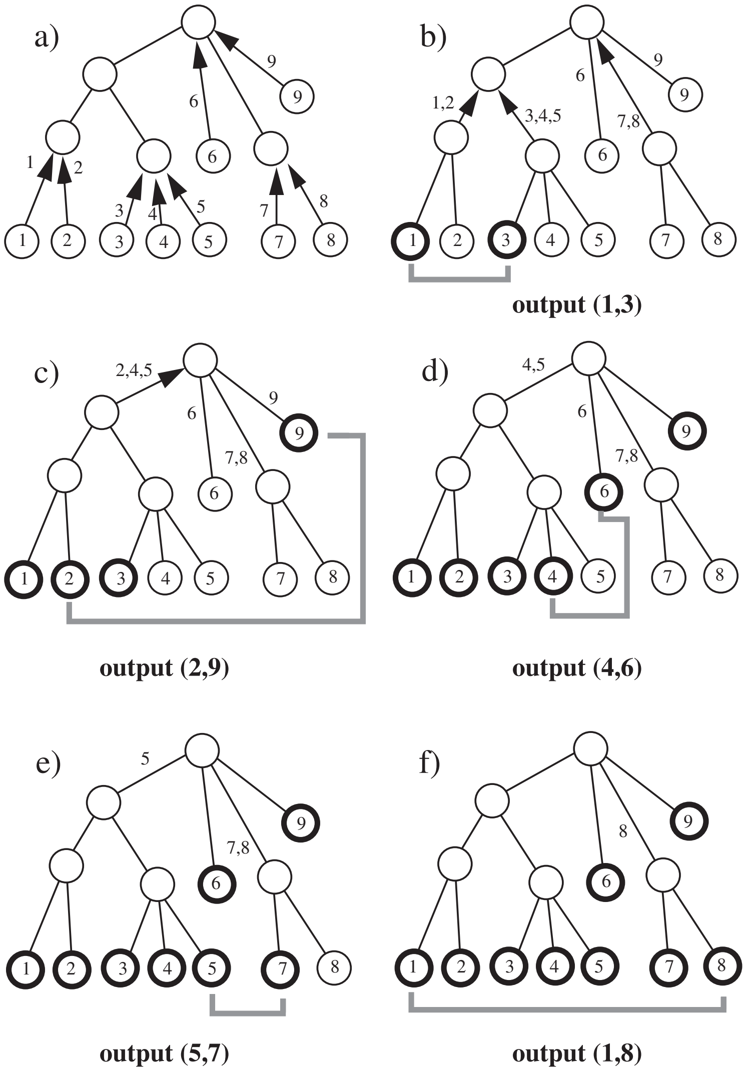

Figure 9: Execution of PROBE2 on a tree containing 9 leaves and 5 internal nodes; a total of 5 probes are generated (shaded lines). Message lists are shown on the edges of T.

complete fault detection.

The time complexity of PROBE2 is optimal as well: each node v passes no more than d labels to its parent, and thus each node will receive fewer than d^2 labels from its children. Assuming that d is a technology-dependent constant, the amount of processing at each node is a constant, and since each node is processed only once, the overall time complexity of PROBE2 is linear in the size of the input tree.

3 Efficient Probe Scheduling

Efficient probe scheduling algorithms are essential because testing cost depends largely on the total travel time of the probe heads. In mechanical probing, individual stepper motors control the $x-$ and $y-$coordinates of each moving head. The distance $dist(A_i, B_i)$ traveled by the i^{th} probe head is given by

$$dist(A_i, B_i) = max[\ |A_{i_x} - B_{i_x}|\ ,\ |A_{i_y} - B_{i_y}|\ ].$$

This distance function (also known as the Chebyshev or L_∞ norm) reflects the fact that the maximum time interval for which any motor is engaged will determine the delay between consecutive probes; such a metric is typical in manufacturing applications and is quite accurate despite second-order effects such as acceleration and deceleration of the moving heads. For k-probes, when $k = 2$, the *cost* of moving the probe heads from a set of pin locations $A = \{A_1, A_2\}$ to another set of locations $B = \{B_1, B_2\}$ is given by

$$c(A, B) = \min\ \{\ \max[\ dist(A_1, B_1)\ ,\ dist(A_2, B_2)\ ]\quad ,\quad \max[\ dist(A_1, B_2)\ ,\ dist(A_2, B_1)\ ]\ \}.$$

For $k > 2$, the cost of moving the probe heads from $A = \{A_1, \ldots, A_k\}$ to $B = \{B_1, \ldots, B_k\}$ is given by

$$c(A, B) = \min_{\{\sigma\}}\ \ max[dist(A_1, B_{\sigma(1)})\ ,\ dist(A_2, B_{\sigma(2)})\ ,\ \ldots\ ,\ dist(A_k, B_{\sigma(k)})]$$

where $\{\sigma\}$ denotes the set of all permutations of the probe indices $\{1, \ldots, k\}$. In other words, we choose the mapping of A onto B in such a way that the maximum travel time of any probe head is minimized (see Figure 10).

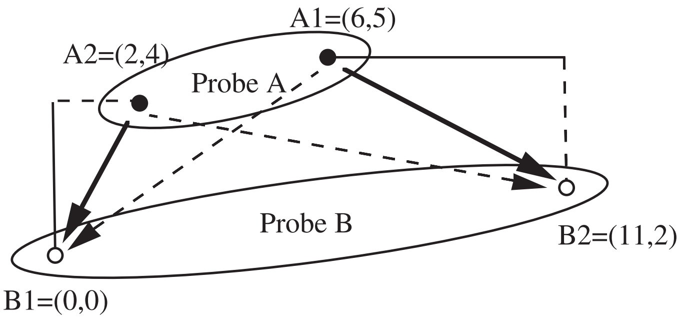

Figure 10: An example showing the distance between two probes $A = \{A1, A2\}$, $B = \{B1, B2\}$. We have $dist(A1, B1) = 6$, $dist(A2, B2) = 9$, $dist(A2, B1) = 4$, and $dist(A1, B2) = 5$; thus, the distance between the two probes A and B is $\min(\max(6, 9), \max(4, 5)) = \min(9, 5) = 5$ (i.e., the best strategy will move one probe head from $A2$ to $B1$ while the other probe head moves from $A1$ to $B2$).

In some technologies, each probe head may be carried by its own moving horizontal bar, and collisions between probes become a concern (i.e., no two such parallel bars are allowed to cross each other's path, so the y coordinates of the k probe heads must satisfy $y_1 \leq y_2 \leq \ldots \leq y_k$ at all times). Thus, the probe head coordinates are always sorted lexicographically [25]. This constraint clearly yields a metric, which we call the *collision-free* metric; in contrast, the metric discussed above will be referred to as the *generalized* metric. The collision-free metric is more restrictive, since there is always a unique feasible permutation of the probe heads in traveling from one set of locations to another. In particular, for $k = 2$ the cost under the collision-free metric of moving the probe heads from $A = \{(x_1, y_1), (x_2, y_2)\}$ to $B = \{(x_3, y_3), (x_4, y_4)\}$, $y_1 < y_2$, $y_3 < y_4$ is given by $\max\{\ |x_1 - x_3|\ ,\ |y_1 - y_3|\ ,\ |x_2 - x_4|\ ,\ |y_2 - y_4|\}$.

The Minimal k-Probe Scheduling (k-MPS) Problem: Given a set of k-probes, minimize the total probe moving cost required in executing all probes.

A straightforward reduction from the geometric traveling salesman problem [15] yields:

Theorem 3.1 *The k-MPS problem is NP-hard.*

Proof: We can transform a geometric instance of TSP into an instance of MPS by introducing k copies of each site, then considering each set of k identical copies of a site as a single k-probe. Distances between probes will correspond to the original distances between the corresponding sites in the TSP instance. □

The probe scheduling problem seems quite unapproachable, both due to its theoretical intractability and because the distance and travel cost functions are not easily intuited. Thus, previous work relies on generic traveling salesman heuristics to optimize the probe schedule. For example, when $k = 2$ probe heads are available, Crowell et al. [7] use a bandsort algorithm to optimize the movement of *one* of the probe heads. Unfortunately, the other probe head may be forced to travel very large distances between probes, and indeed the resulting schedule is often exceedingly inefficient. Yao et al. [25] use simulated annealing and the Kernighan-Lin 2-*opt* criterion [16] as the basis of an iterative interchange approach; their schedules save up to 83% of travel costs over the method of [7]. All of the heuristics proposed in [7] and [25], however, have unbounded error.

In this section, we first show that the k-probe travel costs are actually metric (although clearly not geometric), i.e., distances between k-probes satisfy the triangle inequality for all values of $k \geq 1$. As a consequence, traveling salesman heuristics with constant-factor error bound apply [15]. Second, we exploit flexibility in the choice of probes to find probe sets which can co-exist in an efficient probe schedule.

3.1 Metricity of the k-MPS Problem

For the collision-free metric, the travel costs of the probe heads satisfy the triangle inequality, since the probe head coordinates are always in lexicographic order. Thus, moving the probe heads from A to C via an intermediary B yields the same final probe permutation as would result by moving directly from A to C. Metricity follows from the metricity of the Chebyshev norm.

For arbitrary k-probes A, B and C, we may view the travel costs $c(A,B)$, $c(B,C)$ and $c(A,C)$ in the generalized metric as being respectively determined by the optimal permutations $\sigma_1 : A \rightarrow B$, $\sigma_2 : B \rightarrow C$ and $\sigma_3 : A \rightarrow C$. Comparing the composed permutation $\sigma_1 \circ \sigma_2 : A \rightarrow C$ with the permutation $\sigma_3 : A \rightarrow C$ yields the following:

Theorem 3.2 *For any three k-probes A, B and C, the travel costs $c(A,B)$, $c(B,C)$ and $c(A,C)$ in the generalized metric satisfy the triangle inequality, i.e., $c(A,B) + c(B,C) \geq c(A,C)$.*

Proof: Compare the set of edges of permutation $\sigma_3 : A \rightarrow C$ that defines $c(A,C)$, with the induced permutation $\sigma_1 \circ \sigma_2 : A \rightarrow C$ (see Figure 11). Define $\max(\sigma)$ to be the maximum distance traveled by any probe head according to the permutation σ. Clearly $c(A,C) \leq \max(\sigma_1 \circ \sigma_2)$, since $\sigma_1 \circ \sigma_2$ is not necessarily the minimum-cost permutation between A and C. On the other hand, $\max(\sigma_1 \circ \sigma_2) \leq c(A,B) + c(B,C)$ by the triangle inequality and the metricity of the Chebyshev norm. It follows that $c(A,C) \leq c(A,B) + c(B,C)$. $\square$

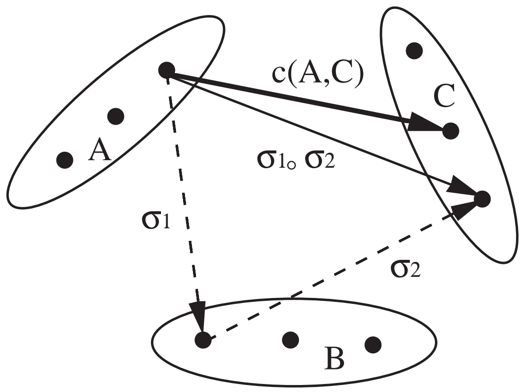

Figure 11: Metricity of the probe travel cost function.

Theorem 3.2 allows us to apply heuristics which achieve bounded error for *metric* TSP instances. In particular, Christofides' combination of a minimum spanning tree construction and minimum matching [15] yields:

Corollary 3.3 *Given a set of n k-probes, for any fixed $k \geq 1$, a heuristic probe schedule with cost at most $\frac{3}{2}$ times optimal can be found within $O(n^3)$ time.* □

3.2 Varying the Probe Set

A further optimization of the tour schedule is possible because the set of probes is itself variable. Figure 12 depicts an instance where a "smarter" choice of probes reduces the optimal tour cost by one-quarter. Most tree topologies can be tested with the minimum number of probes in many distinct ways. For example, each three-pin net in Figure 12 can be tested by a minimal set of 2-probes in three distinct ways (i.e., any two of the three possible probes can be used); in fact, the 2-pin net is the only connection topology with a unique minimum probe set.

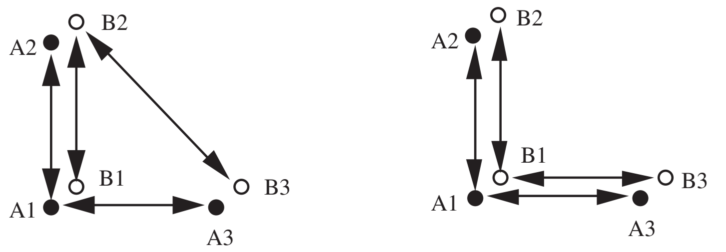

Figure 12: An example of how judicious probe selection can reduce the total tour length by as much as one-quarter: four probes are required for complete wire open fault coverage over the two 3-pin nets $\{A_1 = (0,0),\ A_2 = (0,1),\ A_3 = (1,0)\}$ and $\{B_1 = (\epsilon,\epsilon),\ B_2 = (\epsilon, 1+\epsilon),\ B_3 = (1+\epsilon,\epsilon)\}$. Assuming that the probe tour must start and end at the origin, the probe set on the left will be optimally ordered as $((A_1,A_2)\ ,\ (B_1,B_2)\ ,\ (B_2,B_3)\ ,\ (A_1,A_3))$, requiring about four units of travel time. The probe set on the right may be ordered as $((A_1,A_2)\ ,\ (B_1,B_2)\ ,\ (B_1,B_3)\ ,\ (A_1,A_3))$, requiring only about three units of travel time.

We thus obtain a new type of *compatibility TSP* problem, where sets of k-probes are selected to cover every net such that the optimal tour cost for the *union* of all probe sets is minimized. In other words, we wish to exploit the synergy between the choice of probes and the optimal tour cost:

The Minimal Probe Generation/Scheduling (MPG/S) Problem: Given a routing topology for a signal net, determine and schedule a set of probes so that the total probe moving cost is minimized.

In order to hybridize the probe-generation phase with the tour-scheduling phase, and to take advantage of the non-determinism inherent in probe selection, we propose the heuristic PROBE3 (Figure 13). PROBE3 is based on a minimum-cost insertion strategy, i.e., it schedules all probes for a small subset S of nets, then iteratively adds the probe which has lowest insertion cost in the tour while still allowing a minimum probe set. Note that a probe set which allows us to minimize travel cost may have more than the minimum possible number of probes. However, the heuristics discussed in this section require that the number of probes is minimum. As seen in the following section, PROBE3 yields significantly shorter schedules than other methods.

PROBE3: Insertion-based method for probe selection
Input: A collection N of nets and their routing tree topologies **Output:** An efficient heuristic probe schedule

Compute a minimal set of probes P which verifies a subset $S \subset N$ of the nets
Compute a heuristic schedule (tour) $P_1, \ldots, P_m, P_1$ of P
While $\exists$ a net not having complete fault coverage
    Find a probe P^* for any net N_i such that
      (i) N_i is still coverable by a minimal number of probes after P^* is added, and
      (ii) the probe's minimum insertion cost between consecutive probes is
      minimized, i.e., $\min_{\text{feasible } P^*} \min_i \{c(P_i, P^*) + c(P^*, P_{i+1}) - c(P_i, P_{i+1})\}$
    Insert P^* into the tour between probes P_i and P_{i+1},
      where i was the tour index where P^* had minimum insertion cost

Figure 13: **PROBE3:** An insertion-based heuristic for probe selection.

4 Experimental Results

We tested our algorithms on an MCM benchmark design obtained from Hughes Aircraft Co., containing 44 components and 199 nets (this is the same benchmark used by Yao et al. in [25]). We also used two randomized versions of the Hughes benchmark, where the same net topologies were retained, but with pin coordinates reassigned randomly from a uniform distribution in the layout region. Algorithm PROBE2 was used to generate minimal probe sets which cover all possible wire open and cracked via faults. The schedules for these probe sets were optimized using the 2-opt TSP heuristic, as well as by 2-opt followed by 3-opt (in a separate run).

We also tested a variant of PROBE3 on the same benchmark, as described in Section 3.2 above. We first generated a minimal set of probes for all nets other than the power, ground, and nets with 3 or less pins, then computed a heuristic tour for these probes, using the 2-opt TSP heuristic (again, in a separate run, we used 2-opt followed by 3-opt). Finally, we iteratively added additional probes for the remaining nets which (i) could be inserted into the current tour with minimum cost, and (ii) were compatible with previously chosen probes in some minimum probe set. In all cases, a total of 634 probes were generated by our algorithm, the same number as that generated by the algorithm of [25]. With each of the PROBE3 experiments, 226 probes were initially chosen to cover the nets which had

> 3 pins and which were neither power nor ground; the remaining 408 probes were added incrementally. In the PROBE3 experiments, we optionally ran 2-opt improvement after every 10 probes added, and optionally ran 3-opt improvement after every 50 probes added. All of the above benchmarks were run with the collision-free distance function, as well as with the generalized distance function. These results are summarized in Table 1.

MCM	metric	PROBE2 + 2-Opt	PROBE2 + 2-Opt + 3-Opt	PROBE3 + 2-Opt	PROBE3 + 2-Opt + 3-Opt
Hughes	generalized	160,435,000	153,185,000	126,210,000	118,497,000
	collision-free	163,202,000	157,600,000	131,010,000	126,637,500
Random1	generalized	294,164,000	286,679,000	265,276,000	257,838,000
	collision-free	302,684,000	289,843,000	269,346,000	260,897,000
Random2	generalized	295,956,000	285,379,000	271,869,000	260,150,000
	collision-free	304,885,000	294,421,000	270,767,000	263,113,000

Table 1: Performance of PROBE2 and PROBE3 variants on the industry benchmark and on random examples. Note that the best probe schedule cost obtained by Yao et al. for the industry benchmark, using simulated annealing, was 150,525,000 units. The tour obtained by PROBE3 + 2-opt + 3-opt gives savings of up to 21% over this value. Each benchmark was run with the collision-free distance function, as well as with the generalized distance function.

As expected, the PROBE3 variants, being able to carefully choose probes while constructing the heuristic tour, outperformed PROBE2 by a considerable margin. Results are somewhat better when 3-opt is incorporated, also as expected. For the benchmark design, the best tour obtained in [25] using simulated annealing had cost 150,525,000; in comparison, our PROBE3 variants obtain up to 21% improvement over the results of [25]. Since simulated annealing usually gives solutions quite close to optimal [12], our results confirm that careful choice of compatible probes is an important issue.

5 Future Work

Substrate testing for open faults is a critical phase in the production of multi-chip module packages. We have formulated MCM substrate testing as a problem of connectivity verification for trees using k-probes, and presented linear-time algorithms for optimal probe generation. Our algorithms yield minimum probe sets for covering all possible wire open and cracked via faults. Since the associated probe scheduling problem is metric, a bounded-error scheduling heuristic can be obtained. We presented an effective insertion-based heuristic which exploits the synergy between choice of probes and the resulting optimal schedule cost.

There remain a number of interesting open problems. The fact that many different probe sets can cover a given net yields an interesting TSP variant, as noted above. It is possible that a "prize-collecting salesman" formulation (e.g., at least two of the three possible probes must be "collected" for each three-pin net) can be solved with constant-factor error via an LP-relaxation scheme. This would be quite useful, as the bounded error heuristic of Corollary 3.3 in Section 3 applies only when all of the probes have been fixed. Analyzing the maximum error inherent in arbitrarily fixing the probes is also of interest. Advances in probe technology allow $k > 2$ probe heads to move simultaneously, which affords even greater freedom in choosing the probe sets. Thus, the interplay between probe choice and tour cost will continue to be important. More sophisticated strategies for efficiently inserting probes into a partial tour are possible (e.g., we may iteratively look for the best improving combination of added and deleted probes). Finally, the concept of verifying connectivity by checking paths, rather than edges, as well as the "physical" node failure mode (via cracking), can be applied to both trees and arbitrary graphs arising in other fields of study.

6 Acknowledgements

We thank Brian Tien and Ed Shi for access to the benchmark. We are also grateful to Professor C. K. Cheng, Nan-Chi Chou, David Yao, and Tom Russell, for many interesting discussions.

References

[1] R. W. Bassett, P. S. Gillis, and J. J. Shushereba, *Testing and Diagnosis of High-Density CMOS Multichip Modules*, in Proc. IEEE Workshop on Multichip Modules, Santa Cruz, March 1991, pp. 108–113.

[2] R. H. Bruce, W. P. Meuli, and J. M. Ho, *Multi Chip Modules*, in Proc. ACM/IEEE Design Automation Conf., Las Vegas, June 1989, pp. 389–393.

[3] R. Carragher, N. C. Chou, C. K. Cheng, and T. Russell, *Distortion Mapping for Cofired Ceramic Substrate Testing*, in Proc. Intl. Symp. on Microelectronics, November 1993, pp. 295–300.

[4] N. C. Chou and C. K. Cheng, *Optimal Test Size and Efficient Probe Scheduling for Substrate Verification Using Two-Probe Testers*, in Proc. Intl. Symp. on Microelectronics, November 1993, pp. 276–281.

[5] N. C. Chou, C. K. Cheng, and T. Russell, *Dynamic Probe Scheduling Optimization for MCM Substrate Test*, IEEE Trans. on Components, Hybrids, and Manufacturing Tech., (1994), pp. 182–189.

[6] N. C. Chou, C. K. Cheng, and T. C. Russell, *High-Performance Microelectronic Substrate Verification Using Probe Testers*, in Proc. IEEE Intl. ASIC Conf., Rochester, NY, September 1992, pp. 230–233.

[7] J. C. Crowell, R. Keogh, and J. Conti, *Moving Probe Bare Board Tester Offers Unlimited Testing Flexibility*, Industrial Electronics Equipment Design, (1984).

[8] W. W. Dai, *Performance Driven Layout of Thin-film Substrates for Multichip Modules*, in Proc. IEEE Workshop on Multichip Modules, Santa Cruz, March 1991, pp. 114–121.

[9] S. Golladay, N. Wagner, J. Rudert, and R. Schmidt, *Electron-Beam Technology for Open/Short Testing of Multi-Chip Substrates*, IBM J. Res. Develop., 34 (1990), pp. 250–259.

[10] J. T. A. Group, *JTAG Boundary-Scan Architecture Standard Proposal*, version 2.0 ed., March 1988.

[11] D. Herrell, *Multichip Module Technology at MCC*, in Proc. IEEE Intl. Symp. Circuits and Systems, New Orleans, LA, June 1990, pp. 2099–2103.

[12] D. S. Johnson, C. R. Aragon, L. A. McGeogh, and C. Schevon, *Optimization by Simulated Annealing: An Experimental Evaluation (part 1)*, Operations Research, 37 (1989), pp. 865–892.

[13] A. B. Kahng, G. Robins, and E. A. Walkup, *On Connectivity Verification in Multi-Chip Module Substrates*, Tech. Rep. CSD-TR-910074, Computer Science Department, UCLA, 1991.

[14] A. B. Kahng, G. Robins, and E. A. Walkup, *New Results and Algorithms for MCM Substrate Testing*, in Proc. IEEE Intl. Symp. Circuits and Systems, San Diego, CA, May 1992, pp. 1113–1116.

[15] E. L. Lawler, *The Traveling Salesman Problem: a Guided Tour of Combinatorial Optimization*, Wiley, Chichester, New York, 1985.

[16] S. Lin, *Computer Solutions of the Traveling Salesman Problem*, Bell System Technical Journal, 44 (1965), pp. 2245–2269.

[17] B. McWilliams, *private communication.* (invited talk at CANDE meeting), San Marcos, CA, April 1991.

[18] G. Robins, *On Optimal Interconnections*, PhD thesis, Department of Computer Science, UCLA, CSD-TR-920024, 1992.

[19] T. Russell, *private communication*, August 1991. ALCOA Corp.

[20] R. G. Sartore, N. Shastry, U. Brahme, K. Jefferson, and R. Halaviati, *Tutorial for Computer Aided Diagnostic E-Beam Testing of ASICs*, in Proc. IEEE Intl. ASIC Conf., Rochester, NY, September 1991, pp. T8:1.1 – T8:1.7.

[21] K. P. Shambrook, *An Overview of Multichip Module Technologies*, in Proc. IEEE Workshop on Multichip Modules, Santa Cruz, CA, March 1991, pp. 1–6.

[22] M. Taylor and W. W. Dai, *TinyMCM*, in Proc. IEEE Workshop on Multichip Modules, Santa Cruz, CA, March 1991, pp. 143–147.

[23] L. Wang, M. Marhoefer, and E. McCluskey, *A Self-Test and Self-Diagnosis Architecture for Boards Using Boundary Scans*, in Proc. European Test Conf., Paris, April 1989, pp. 119–126.

[24] S. Weber, *For VLSI, Multichip Modules May Become the Packages of Choice*, Electronics, (1989), pp. 106–112.

[25] S. Z. Yao, N. C. Chou, C. K. Cheng, and T. C. Hu, *A Multi-Chip Module Substrate Testing Algorithm*, in Proc. IEEE Intl. ASIC Conf., Rochester, NY, September 1991, pp. P9:4.1 – P9:4.4.

[26] S. Z. Yao, N. C. Chou, C. K. Cheng, and T. C. Hu, *An Optimal Probe Testing Algorithm for The Connectivity Verification of MCM Substrates*, in Proc. IEEE Intl. Conf. Computer-Aided Design, Santa Clara, CA, November 1992, pp. 264–267.

[27] S. Z. Yao, N. C. Chou, C. K. Cheng, and T. C. Hu, *A Multi-Probe Approach for MCM Substrate Testing*, IEEE Trans. Computer-Aided Design, 13 (1994), pp. 110–121.

Biographies

Andrew B. Kahng (b. Oct. 1963, San Diego, CA) holds the A.B. degree in applied mathematics and physics from Harvard College, and the M.S. and Ph.D. degrees in computer science from the University of California at San Diego. Since July 1989, he has been on the faculty of the computer science department at UCLA, where he has been associate professor since 1994. He has received NSF Research Initiation and Young Investigator Awards, and co-directs both the VLSI CAD and Commotion (cooperative motion) Laboratories. His research interests include computer-aided design of high-performance VLSI circuits, practical global optimization, and the theory of cooperative task-solving.

Gabriel Robins received his Ph.D. in 1992 from UCLA, where he won a Distinguished Teaching Award and held an IBM Fellowship. Dr. Robins is now Assistant Professor of Computer Science at the University of Virginia, where he won an NSF Young Investigator Award, a Lilly Foundation Teaching Fellowship, and an All-University Outstanding Teaching Award. Dr. Robins is a member of an advisory board to the U.S. Department of Defense. He is General Chair of the 1996 ACM/SIGDA Physical Design Workshop, and he also serves on the program committees of several other leading conferences. He is a member of ACM, IEEE, SIAM, and MAA.

Elizabeth A. Walkup received bachelor's degrees in Computer Science and Theatre at the University of California, San Diego in 1989, and in 1992 a master of science degree in Computer Science at the University of Washington, where she is currently finishing her Ph.D. thesis, on "Optimization of Linear Max-Plus Systems with Application to Timing Analysis".