New Results and Algorithms for MCM Substrate Testing

A. B. Kahng, G. Robins, and E. A. Walkup†

Dept. of Computer Science, UCLA, Los Angeles, CA 90024-1596
† Dept. of Computer Science, Univ. of Washington, Seattle, WA 98195

Abstract

Multi-chip module packaging techniques present several new technical challenges, notably substrate testing. We formulate substrate testing as a problem of connectivity verification in trees via k-probes. *We then present a linear-time algorithm which computes a set of probes achieving complete open fault coverage, and formulate efficient probe scheduling as a special type of metric traveling salesman optimization. Simulations using industry benchmarks show reductions in testing costs of up to 21% over the best previous methods.*[1]

1 Preliminaries

Multi-chip module (MCM) technology has recently emerged as an economically viable means for packaging complex, high-performance systems [2] [7] [14] [15]. 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, allowing high circuit density and yield while decreasing interconnect delay.

MCMs eliminate individual integrated circuit (IC) packages, allowing die to be situated closer together. This 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 using high-density printed circuit boards [4]. A typical MCM (Figure 1) consists of a multi-layer substrate containing inter-chip wiring, upon which are mounted a number of bare die.

The increased use of multi-chip module packaging has focused attention on several new and challenging CAD problems, especially those related to layout, thermal reliability, and testing [6] [14]. Testing in particular presents one of the most persistent challenges of the MCM approach [1] [7] [15]. It is desirable to discover defects in the MCM substrate early in the manufacturing process. Certainly, the fully-assembled MCM package can be tested using combinatorial IC testing techniques. However, the pre-assembly MCM substrate contains only a set of disjoint wiring connections with no active devices; it cannot be tested using conventional techniques.

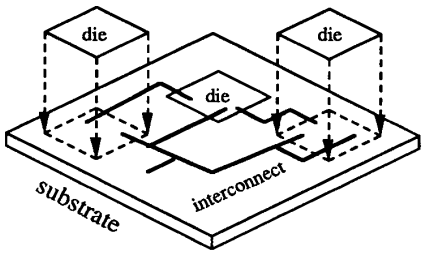

Figure 1: A multi-chip module.

In an MCM substrate, 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, wherein 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.

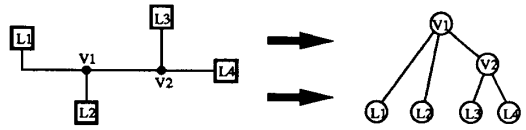

Figure 2: A sample net (left) and its corresponding tree representation (right); pins and vias become leaves and internal nodes, respectively.

An open fault is an electrical disconnection between two points that are to be connected. 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. Traditional methods for connectivity checking involve either parallel probing of the circuit under test (e.g. bed-of-nails), or combinatorial exercising of the logic

[1] This research was supported in part by NSF grant MIP-9110696, by an IBM Graduate Fellowship (Robins) and by an NSF Graduate Fellowship (Walkup).

0-7803-0593-0/92 $3.00 1992 IEEE

(e.g., the boundary-scan), neither of which is appropriate to MCM substrate testing [2]. Several groups have recently proposed new methods for verifying substrate connectivity, relying on *sequential probing* of the MCM substrate in contrast to the standard approaches above.

[6] proposes an electron-beam method to test MCM substrates for faults by injecting charge into individual nets and then scanning them for faults. Unfortunately, electron-beam testers can be prohibitively slow [13]. 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 (usually $k = 2$). A single k-probe simultaneously verifies all paths between pairs of terminals in the probe set by measuring resistance and capacitance values.

A 2-probe sequential testing approach developed in [3] 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. 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. As noted above, the incomplete fault coverage thus afforded is economically unacceptable.

Thus, MCM manufacturers are now adopting substrate test methodologies which provide complete open fault coverage for all nets [12]. Yao et al. [16] recently proposed a quadratic-time algorithm that determines a set of 2-probes which will check for all possible open faults. Because 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) [3] [16], our discussion is confined to the issue of complete open fault coverage. We give a linear-time algorithm which for any $k \geq 2$ determines a near-optimal k-probe set which accomplishes complete open fault coverage of each net.

Once probes are found which adequately test for open faults, one must efficiently schedule the probes for execution by a mechanical tester. Previous groups [3] [16] have used generic greedy or iterative traveling salesman heuristics to attack this problem. We propose an effective heuristic for probe scheduling based on new observations concerning the metricity and allowable structure of the probe set. Theorem statements and proofs have been omitted for brevity; see [9] for a more complete treatment.

2 Open Fault Detection

In this section we address the following problem:

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

We will 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. Current probing technology uses $k = 2$, but our work extends to arbtrary k.

2.1 Detection of Wire Open Faults

In order to test the integrity of all wire segments, certainly 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. A simple 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 $\{p_i, p_{i+\lfloor \frac{l}{2} \rfloor}\}$, $1 \leq i \leq \lfloor \frac{l}{2} \rfloor$, will cover all edges of the tree (Figure 3); if l is odd, an additional probe $\{p_1, p_l\}$ is generated.

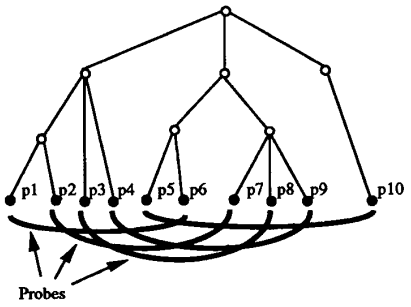

Figure 3: Optimal wire open fault detection.

2.2 Detection of Cracked Via Faults

In manufacturing the MCM substrate, a via can physically "crack" due to misalignment in lithography or thermal stress. Thus, subtrees rooted at this internal node of the net can become electrically separated (Figure 4) [16], so that certain sets of probes will fail to detect the cracked via. We now give a linear-time algorithm that efficiently tests for *both* open wire faults and cracked vias.

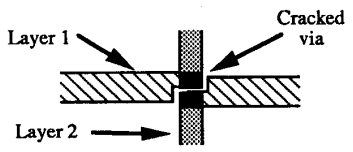

Figure 4: A cracked via in a routing.

We first root the tree at an internal node R of maximum degree d and then orient all edges towards R. Each leaf node sends a message list containing its label to its parent . When a given node has received

message lists from all of its children, it iteratively generates probes by pairing labels from distinct incoming lists, at least one of which must contain more than one node label; when the total size of incoming lists at that node has been reduced to less than $d+1$, the lists are concatenated and sent to the node's parent. This process is repeated at each node until only the root remains unprocessed; a simple cleanup step is then performed. Figure 5 traces a sample execution, and a formal statement is given in Figure 6.

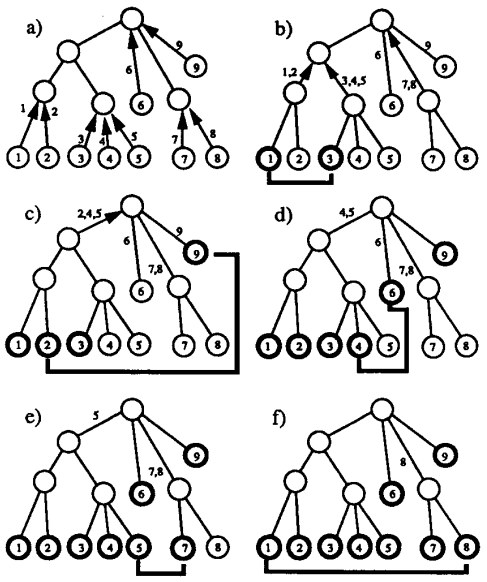

Figure 5: A sample run on a net topology, generating five probes (thick arcs).

Except at the root, each probe generated will remove two distinct leaf node names from the messages being passed. At most d leaf names will remain to be processed at the root, requiring at most $d-1$ additional probes. Therefore, to test an l-pin net our algorithm generates at most $\frac{l-d}{2}+(d-1)=\frac{l}{2}+\frac{d}{2}-1$ probes, and this bound is the best possible (tight for a star topology). Assuming that d is bounded by a constant dependent on technology, each node v passes no more than d leaf names up to its parent, and thus each node will certainly receive fewer than d^2 leaf names from its children. The overall time complexity is linear in the size of the routing topology.

3 Efficient Probe Scheduling

Efficient probe scheduling algorithms are necessary because testing cost is dependent on the total travel time of the probe heads. In mechanical probing, individual stepper motors will 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 $\max(|A_{i_x}-B_{i_x}|, |A_{i_y}-B_{i_y}|)$. This distance function

Probe set generator
Input: A routing tree topology
Output: A probeset for complete fault coverage
Let V = all nodes in routing tree
Root tree at arbitrary node $r \in V$
Direct all edges towards r
Each leaf v sends $\{v\}$ to $parent(v)$
While $\exists v \in V$ that received msgs $M_1, \ldots, M_{deg(v)}$
  While $\exists i \ni |M_i| \geq 2$
    Let $x \in M_i$
    Let $y \in M_j$ where $|M_j| \geq 1$
    Print (x, y)
    $M_i = M_i - \{x\}$
    $M_j = M_j - \{y\}$
  Let $L = M_1 \cup \ldots \cup M_{deg(v)}$
  If $v = r$ **Then**
    If $|M_i| \leq 1\ \forall i$ **Then**
      Print $(L_1, L_i)\ \forall\ 2 \leq i \leq |L|$; **Stop**
    Else $\forall L_i \in M_i \ni |M_i| \geq 2$ **Print** (L_i, L_k),
      L_k a leaf descendant of M_k, $k \neq i$; **Stop**
  Else send L to $parent(v)$
  $V = V - \{v\}$

Figure 6: Optimal detection of both wire open and cracked via faults.

reflects the fact that the maximum time interval for which any motor is engaged will determine the delay between consecutive probes. For k-probes, the *cost* of moving the probe heads between two sets of pin locations $A = \{A_1, \ldots, A_k\}$ and $B = \{B_1, \ldots, B_k\}$ is given by $\min_{\{\sigma\}} \{\max[dist(A_1, B_{\sigma(1)}), \ldots, dist(A_k, B_{\sigma(k)})]\}$ where $\{\sigma\}$ denotes the set of all permutations of the probe indices $\{1, \ldots, k\}$.

In some technologies, each probe head may be carried by its own moving horizontal bar. If the probe carriers lie in the same plane, then no two bars may cross each other's path, i.e., the y coordinates of the k probe heads must satisfy $y_1 \leq \ldots \leq y_k$ at all times. Thus, the probe head coordinates are always sorted lexicographically [16]. This constraint yields what we call the *collision-free* distance in contrast to the *general* distance function dicussed above. In particular, for $k=2$ the collision-free distance 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 thus given by the expression $\max\{|x_1 - x_3|, |y_1 - y_3|, |x_2 - x_4|, |y_2 - y_4|\}$.

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 [5] shows that the k-MPS problem is NP-hard. Thus, 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. Previous work relies on generic traveling salesman heuristics to optimize the probe schedule [3], and indeed the resulting schedule is often exceedingly inefficient. [16] uses

simulated annealing and the Kernighan-Lin 2-*opt* criterion [10] as the basis of an iterative interchange approach; their schedules save up to 83% of travel costs over [3], yet all of these heuristics [3] [16] have unbounded error.

We can show that the k-probe travel costs are metric [9]. Hence traveling salesman heuristics with constant-factor error bound may be applied [11]. In particular, Christofides' TSP heuristic [11] yields an $O(n^3)$ time heuristic which for any fixed probe set guarantees a schedule with cost at most $\frac{3}{2}$ times optimal. We may also exploit the synergy between the choice of probes and the optimal tour cost to find probe sets which can co-exist in an efficient probe schedule, and apply heuristics which achieve bounded error for *metric* TSP instances [9].

4 Experimental Results

We tested our algorithms on the Hughes MCM benchmark used in [16], containing 44 components and 199 nets. A minimal probe set was generated, covering all possible wire open and cracked via faults for all nets. The schedule for these probe sets was 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 that takes advantage of the structure of special nets, as discussed 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. 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. In each experiment, 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, using 2-opt improvement after every 10 probes added, and 3-opt improvement after every 50 probes added. All of the above benchmarks were run using both the generalized and the collision-free distance functions.

As expected, the variants which were able to carefully choose probes while constructing the heuristic tour, outperformed the basic algorithm by a considerable margin, with results somewhat better when 3-opt is incorporated. Our method obtained up to 21% reduction in travel cost over the results of [16], i.e., 118,497,000 units as opposed to 150,525,000 units. Since simulated annealing usually gives solutions quite close to optimal [8], our results indeed confirm that careful choice of compatible probes is an important issue.

5 Conclusions

Substrate testing is critical to cost-effective MCM production. We have formulated this problem as the verification of a set of trees using k-probes, and presented linear-time algorithms for optimal probe generation. Our algorithms yield near-optimal probe sets for complete fault coverage. We showed that the associated probe scheduling problem is metric, and obtained a bounded-error scheduling heuristic. Furthermore, we presented an insertion-based heuristic which exploits the special structure of 3-pin and the power/ground nets in the MCM substrate. This heuristic significantly improves probing costs over previous methods. In general, the concept of verifying connectivity by checking paths, rather than edges, is quite novel and can be applied in a number of other fields.

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. Ho, *Multi chip modules*, in Proc. Design Automation Conf., Santa Cruz, June 1989, pp. 389–393.

[3] J. C. Crowell, R. Keogh, and J. Conti, *Moving probe bare board tester offers unlimited testing flexibility*, Industrial Electronics Equipment Design, (1984).

[4] 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.

[5] M. R. Garey and D. S. Johnson, *Computers and Intractability: a Guide to the Theory of NP Completeness*, W. H. Freeman, San Francisco, 1979.

[6] 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.

[7] D. Herrell, *Multichip module technology at mcc*, in Proc. IEEE Intl. Symp. on Circuits and Systems, June 1990, pp. 2099–2103.

[8] 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.

[9] A. B. Kahng, G. Robins, and E. A. Walkup, *On connectivity verification in multi-chip module substrates*, Tech. Rep. CSD-TR-910074, University of California, Los Angeles, Computer Science Department, 1991.

[10] S. Lin, *Computer solutions of the traveling salesman problem*, Bell System Technical Journal, 44 (1965), pp. 2245–2269.

[11] C. H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization*, Prentice-Hall, 1982.

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

[13] R. G. Sartore, N. Shastry, U. Brahme, K. Jefferson, and R. Halaviati, *Tutorial for computer aided diagnostic e-beam testing of asics*, in Proc. 4th IEEE Intl. ASIC Conf., Rochester, NY, September 1991, pp. T8:1.1 – T8:1.7.

[14] K. P. Shambrook, *An overview of multichip module technologies*, in Proc. IEEE Workshop on Multichip Modules, Santa Cruz, CA, March 1991, pp. 1–6.

[15] S. Weber, *For vlsi, multichip modules may become the packages of choice*, Electronics, (1989), pp. 106–112.

[16] S.-Z. Yao, N.-C. Chou, C.-K. Cheng, and T. C. Hu, *A multichip module substrate testing algorithm*, in Proc. 4th IEEE Intl. ASIC Conf., Rochester, NY, September 1991, pp. P9:4.1 – P9:4.4.