Short Papers

Interconnect Layout Optimization Under Higher Order *RLC* Model for MCM Designs

Jason Cong, Cheng-Kok Koh, and Patrick H. Madden

***Abstract*—In this paper, we study the interconnect layout optimization problem under a higher order resistance–inductance–capacitance model to optimize not only delay, but also waveform for interconnects with nonmonotone signal response in the context of multipchip-module global routing. We propose a unified approach that considers topology optimization and waveform optimization simultaneously. Using a new incremental moment-computation algorithm, we interleave topology construction with moment computation to facilitate accurate delay calculation and evaluation of waveform quality. Our algorithm considers a large class of routing topologies, ranging from shortest path Steiner trees to bounded-radius Steiner trees and Steiner routings. We construct a set of required arrival-time Steiner (RATS) trees, providing smooth tradeoffs among signal delay, waveform, and routing area. When combined with the MINOTAUR MCM global router (Cong and Madden, 1998), (Madden, 1998) that we have developed, the RATS-tree solutions prove to be effective in reducing overall routing congestion.**

***Index Terms*—Incremental moment computation, required arrival time, RLC interconnect model, routing congestion, signal integrity, topology optimization, topology selection.**

I. Introduction

As very large scale integration (VLSI) circuitry reaches deep-submicrometer device dimension, operates at gigahertz clock frequencies, and is packaged in highly integrated multichip modules, the performance of interconnect structures becomes a dominant factor in determining system performance. Recent studies showed that interconnect performance could be optimized with several techniques, which included topology optimization, wire-sizing optimization, and/or repeater optimization; [9] gave a comprehensive survey of these techniques. However, most of these techniques were applied independently and were based on resistance–capacitance (*RC*) models that might not be appropriate for multichip-module (MCM) designs in which the inductance effect dominated. Furthermore, because they assumed a *single* "optimal" layout structure for each timing-critical net, these layout techniques stringently restricted the global routing solution space. Such a restriction significantly reduced the flexibility that a global router had in minimizing global congestion. In this paper, we study the problem of interconnect layout optimization for performance and signal integrity under a higher order resistance–inductance–capacitance (*RLC*) model and its application to global routing for MCM designs. Our approach has the following advantages.

1) It considers a higher order moment-based *RLC* model, which is more suitable for MCM designs. Most of the previous results on interconnect optimization were achieved under *RC* interconnect models for integrated circuit designs only. Studies on topology optimization such as A trees [7], low-delay trees [1], iterative Dreyfus–Wagner and constructive force-directed Steiner trees [16], non-Hanan routing [17], wire-sizing optimization [3], [6], [7], [11], [12], [27], [31], as well as the more recent studies that combined topology construction with repeater insertion and/or wire sizing [18], [28], [29], [34] assumed *RC* models for the interconnects. With the exceptions of [12], [18], and [29], which considered higher order *RC* models, all other studies used the Elmore delay model [10], which accounted for the first order *RC* effect only. These studies did not consider the inductance effect, a dominant factor in high-speed MCM designs. Our proposed paper overcomes these shortcomings by considering an *RLC* interconnect model during the optimization process. By employing a higher order *RLC* model in our study, not only can we improve the accuracy of delay estimation, we can also better approximate the time domain waveform.
2) Our method is capable of constructing a large class of routing topologies, ranging from shortest path Steiner trees to bounded-radius Steiner trees and Steiner routings. All of the previous topology construction algorithms were limited to a single class of routing topology. For example, the A-tree algorithm considered only the shortest path Steiner trees [7] and the set of topologies generated by the P-tree algorithm [28] was restricted to a permutation-induced abstract topology predetermined by a traveling salesman heuristic. By considering a large class of routing topologies, our algorithm is able to produce *a set of topologies*, providing smooth tradeoffs among signal delay, waveform, and routing area. It generates, for each timing-critical net, several timing-correct topologies from which a global router can select one that is compatible with the routing topologies of other nets. Such flexibility results in significant improvements in the overall congestion levels of global routing solutions.

Section II of this paper formulates the interconnect layout optimization problem. In Section III, we present the proposed interconnect layout optimization algorithm. In Section IV, we present an incremental approach for computing moment in a bottom-up fashion. We present the experimental results in Section V and conclude the paper in Section VI.

II. Problem Formulation

Given a net of pins or *terminals* $\{s_0, s_1, s_2, \ldots, s_n\}$ to be electrically connected, we assume that s_0 denotes the source (or driver) of the net and that the rest of the pins are sinks (or receivers). We use $\mathrm{P_T}(u, v)$ to denote the unique path from u to v in an interconnect tree $\mathrm{T}, d_{\mathrm{T}}(u, v)$ the path-length of $\mathrm{P_T}(u, v)$, and $d(u, v)$ the Manhattan distance between u and v. We denote the signal delay from u to v by $t_{\mathrm{T}}(u, v)$. The source node s_0 will generally be referred to as the root of an interconnect tree and each node v in the tree is connected to its

Manuscript received February 16, 2001; revised April 20, 2001. This work was supported in part by the Defense Advanced Research Projects Agency/Electronics Technology Office under Contract DAAL01-96-K-3600, in part by the National Science Foundation under Grant CCR-9984553, and in part by the Intel Corporation under the California MICRO Program. This paper was presented in part at the IEEE/ACM International Conference on Computer-Aided Design, San Jose, CA, November 1997. This paper was recommended by Associate Editor R. Saleh.

J. Cong is with the Computer Science Department, University of California, Los Angeles, CA 90095 USA.

C.-K. Koh is with the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907-1285 USA.

P. H. Madden is with the Computer Science Department, the State University of New York at Binghamton, Binghamton, NY 13902-6000 USA.

Publisher Item Identifier S 0278-0070(01)10634-2.

0278–0070/01$10.00 © 2001 IEEE

parent by edge e_v. We use T_v to denote the subtree that is rooted at v. We use T_e to denote the subtree with e being its root edge. $|e|$ denotes the length of e.

Let q_i be the required arrival time of sink s_i. A minimal required arrival-time Steiner (RATS) tree is a minimum-cost Steiner tree such that $q_i \geq t_{\mathrm{T}}(s_0, s_i)$ for all sinks. RATS trees include several commonly used topologies when we use the path-length delay formulation with $t_{\mathrm{T}}(u,v) = d_{\mathrm{T}}(u,v)$. For example, by setting $q_i = d(s_0, s_i)$ for all sinks, an optimal RATS tree is an optimal Steiner arborescence [26]. If we relax the requirement such that all sinks have the same required arrival time, then an optimal RATS tree is an optimal bounded-radius Steiner tree [4]. Lastly, an optimal RATS tree with unbounded q_i's is an optimal Steiner tree [22].

The path-length formulation captures the delay for unloaded lossless transmission lines in MCM printed circuit board designs perfectly. In this formulation, the output response at the end of an interconnect e is a replica of the input signal delayed by the time-of-flight (or propagation delay) $t_f = \sqrt{\mathrm{LC}}|e|$ where L and C are the unit-length interconnect inductance and capacitance, respectively, and $|e|$ is the length of the interconnect. In general, it is assumed that $\sqrt{\mathrm{LC}}$ is a constant. Therefore, the time-of-flight (or propagation delay) on the path from the source to sink s_i is $t_f(s_0, s_i) = \sum_{e_v \in \mathrm{P}_{\mathrm{T}}(s_0,s_i)} \sqrt{\mathrm{LC}}|e_v|$, which is proportional to the path length $d_{\mathrm{T}}(s_0, s_i)$.

A more general formulation is to model MCM interconnect structures as lossy transmission lines. Under this formulation, the delay at each sink is the sum of the propagation delay t_f and the rising/falling (or transition) delay τ_t of the signal response waveform [32]

$$t_{\mathrm{T}}(s_0, s_i) = t_f(s_0, s_i) + \tau_t(s_0, s_i). \tag{1}$$

In this paper, we model the transition delay with a higher order moment-based delay model. More specifically, we use the two-pole-based analytical delay model proposed in [21] to approximate the time taken for a rising signal to reach 90% of V_{dd} for the first time at node i as follows:

$$\tau_t(s_0, s_i) \approx \begin{cases} 2.36 \cdot \frac{m_i^1 + \sqrt{4m_i^2 - 3\left(m_i^1\right)^2}}{2}, & \text{if } 4m_i^2 - 3\left(m_i^1\right)^2 > 0 \\ 1.66 \cdot \frac{2\left(\left(m_i^1\right)^2 - m_i^2\right)}{\sqrt{3\left(m_i^1\right)^2 - 4m_i^2}}, & \text{if } 4m_i^2 - 3\left(m_i^1\right)^2 < 0 \\ 3.90 \cdot \frac{m_i^1}{2}, & \text{if } 4m_i^2 - 3\left(m_i^1\right)^2 = 0 \end{cases} \tag{2}$$

where m_i^j is the jth order moment of the voltage transfer function of node i. Moments of an *RLC* interconnect can be computed by the methods proposed in [20] and [33]. In this paper, we present a new approach to moment computation in Section IV.

Signal response waveform is another important factor in interconnect design. Under ideal situations, one would prefer the transmission of the input signal to the output not to be distorted. However, due to impedance mismatch and reflection, ringing may occur at the output node, resulting in excessive settling time and voltage overshoot or undershoot, which may adversely affect the circuit performance. In this paper, we define the signal settling time to be the time taken for the signal to settle above 90% of V_{dd}. Voltage overshoot (undershoot) is the maximum deviation over (under) the final voltage.

Similar to [13], [14], and [25], we use moments as an indirect metric to measure signal quality. For example, if we use the two-pole model in (2) to model the interconnect, then ringing can be attributed to the existence of complex poles in the transfer function. The condition for the poles to be complex (i.e., for the output response to be nonmonotonic, or *underdamped*) is for $\lambda_i = 4m_i^2 - 3(m_i^1)^2$ to be negative. When λ_i is strictly positive, we have an *overdamped* and monotone response. When λ_i is exactly zero, the signal is said to be *critically damped*. An underdamped signal generally has a faster transition delay when compared to a damped signal. With other attributes (such as the signal delay and the wiring length) being equal, we prefer a RATS tree with λ_i's as close to zero as possible.[1]

To summarize, we propose to solve the following problem.

1) RATS Tree Routing: Given a set of terminals s_i's and required arrival times q_i's, construct a RATS T such that $q_i \geq t_{\mathrm{T}}(s_0, s_i)$, critical damping is achieved for good signal quality (i.e., λ_i's are as close to zero as possible) and the total wire-length or wiring area is minimized.

III. RATS-Tree Algorithm

Given a set of terminals, the RATS-tree algorithm operates on a Hanan grid [15] induced by the terminals. Let (x_m, y_m) denote the coordinates of grid point m and $|m| = d(s_0, m)$ denote the Manhattan distance between s_0 and m. All Hanan grid points are *ordered* according to their Manhattan (rectilinear) distance from the source in a descending fashion, with arbitrary tie breaking. Let $\mathrm{order}(m)$ denote the order of m. Given two distinct points p and q, we define the Steiner merging point $\langle p, q \rangle$ to be $(\mathrm{med}(x_{s_0}, x_p, x_q), \mathrm{med}(y_{s_0}, y_p, y_q))$, where $\mathrm{med}()$ returns the median of three numbers. We say that q is dominated by p or $q \preceq p$ if and only if $\langle p, q \rangle = q$ and $p \neq q$. In other words, q lies along a shortest path from s_0 to p.

In general, the RATS-tree algorithm follows a branch-and-bound (B&B) paradigm by considering the *merging* and the *skipping* (of merging) of subtrees at a Steiner merging point as in [26]. There are three significant differences. First, we consider the *rerooting* of a subtree (a concept introduced in [19]) at Hanan grid points. Second, [26] was purely based on path-length delay model; we consider a higher order *RLC* model. Third, we generate a set of topologies.

Each node in the B&B search tree is associated with a *peer topology set* $\mathcal{T}$ that contains a forest of subtrees constructed so far and a scan level $K = \mathrm{order}(m)$, where m is the Steiner merging point last considered in the subtree merging process. The B&B search starts with a peer topology set with all single-terminal trees and a scan level $K = \infty$. Not to be confused with a node in a constructed topology, we refer to a node in the B&B search tree as a B&B node.

We expand a B&B node characterized by $(\mathcal{T}, K)$ by considering a new Steiner merging point m among all Steiner merging points of tree roots in $\mathcal{T}$ such that m is the highest ordered node with $\mathrm{order}(m) < K$. Let $\mathrm{P}_{\succeq}(m) = \{p \mid m \preceq p, \mathrm{T}_p \in \mathcal{T}\}$ denote the set of tree roots in $\mathcal{T}$ that dominate m. For each node $p \in \mathrm{P}_{\succeq}(m)$, a merging operation connects m and p by a shortest path and eliminates T_p from $\mathcal{T}$. We make a shortest path connection between m and p by *growing* T_p along the Hanan grid points from p to m in a bottom-up fashion. A new subtree T_m is added to $\mathcal{T}$ and K is updated to $\mathrm{order}(m)$. If the new Steiner merging point is a terminal, then such a merging is called a *terminal merging*. Otherwise, it is called a *Steiner merging*. Note that when $x_m \neq x_p$ and $y_m \neq y_p$, there are several shortest Manhattan paths from m to p. We consider the two single-bend routings between m and p; in contrast, [26] considered only a single routing. As in the B&B-based minimum rectilinear Steiner arborescence algorithms in [26], we consider the *skipping* of a Steiner merging operation. In this case, while K is updated to $\mathrm{order}(m)$, we keep $\mathcal{T}$ unchanged. Therefore, skipping a Steiner merging operation generates an additional child B&B node in the B&B search tree.

Both terminal merging and Steiner merging operations consider merging at only the roots of subtrees, thereby producing only shortest path trees. In order to consider a large class of routing solutions, we

[1] We have also tried an alternative metric, the third central moment $\mu_i = m_i^3 - 3m_i^1 m_i^2 + 2(m_i^1)^3$ proposed by [25]. We do not observe significant differences between the results obtained under λ_i and those obtained under μ_i.

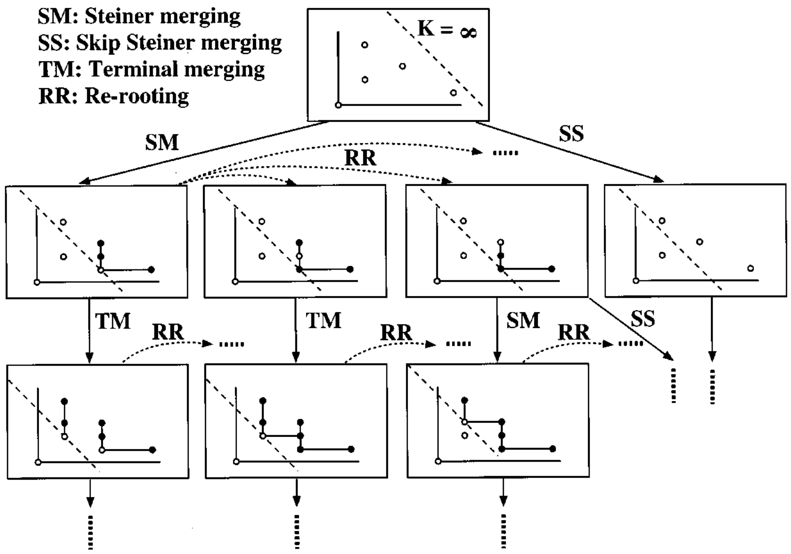

Fig. 1. Snapshot of the B&B search tree illustrating the application of the RATS-tree algorithm to a five-terminal net. Root nodes of peer topology sets are depicted by empty circles and nonroot nodes by filled circles. Scan levels K are shown as dashed lines of gradient -1. Four operations expand the B&B search tree: Steiner merging (SM), skip Steiner merging (SS), terminal merging (TM), and rerooting (RR).

allow merging at nonroot nodes of the subtrees. We achieve nonroot merging by *rerooting*. After each terminal merging or Steiner merging, the resultant topology T is rerooted at various Hanan grid points in T, creating several topologies that consist of the same nodes and edges in T, but rooted at different nodes. These rerooted topologies have identical routing structure, but different sink delays and signal waveforms. Therefore, rerooting creates several sibling B&B nodes for the B&B node newly generated by the terminal merging or Steiner merging operations.

Fig. 1 shows a partial B&B search tree corresponding to the application of the RATS-tree algorithm to a five-terminal net. In this example, we assume that the source is at the origin and the sinks are in the first quadrant. In the B&B search tree, root nodes of peer topology sets are depicted by empty circles and nonroot nodes by filled circles. For illustrative purpose, the scan level $K = \mathrm{order}(m)$ is shown as a dashed line of distance $|m|$ from the origin. The topmost starting B&B node is expanded into the center row of B&B nodes. The leftmost B&B node in the center row is the result of Steiner merging and the rightmost node corresponds to the skipping of the Steiner merging operation. The leftmost B&B node is further expanded by rerooting its newly created subtree at other Hanan grid points. The middle two B&B nodes in the center row show two out of the three rerooted topologies. The leftmost B&B node in the bottom row is due to the application of terminal merging to the leftmost B&B node in the center row. While we do not skip terminal merging, rerooting still applies.

A brief outline of our RATS-tree algorithm is given in Fig. 2. We use a queue to implement a breadth-first traversal of the B&B search tree. At the beginning of the algorithm, the queue contains a B&B node with a peer topology set $\mathcal{T}$ containing all single-terminal subtrees and a scan level $K = \infty$. We use the pair $(\mathcal{T}, K)$ to denote a B&B node. Until the queue is empty, the algorithm iterates the expansion of a B&B node removed from the head of the queue by either performing a terminal/Steiner merging operation or skipping a Steiner merging operation. In the case of subtree merging, the resultant topology is rerooted to generate sibling B&B nodes. All newly generated B&B nodes are appended to the end of the queue.

```
RATS-tree Algorithm($\mathcal{N}$)
Input: A set of pin locations $\mathcal{N}$ with timing requirement
Output: A set of topologies with smooth trade-off among routing
        cost, signal delay, and signal integrity
B&B-Queue ← $\{(\{\{s_0\},\{s_1\},\{s_2\},\cdots,\{s_n\}\},\infty)\}$
while B&B-Queue not empty
    $(\mathcal{T},K)$ ← Remove-Head-of-Queue(B&B-Queue)
    if all RATS-trees in $\mathcal{T}$ are irredundant then
        $(m,m_N,m_E,$Merger-Type$)$ ← Find-Merging-Point$((\mathcal{T},K))$
        if Merger-Type = TERMINAL-MERGING then
            $(\mathcal{T}',K')$ ←(Terminal-Merge$(m,\mathcal{T})$,$|m|$)
            if $m$ is root node
                store the topology of $\mathcal{T}'$ if irredundant
            end if
        else
            /* STEINER-MERGING */
            $(\mathcal{T}',K')$ ←(Steiner-Merge$(m,m_N,m_E,\mathcal{T})$,$|m|$)
            /* SKIP STEINER-MERGING */
            $K$ ← $|m|$
            Append $(\mathcal{T},K)$ to B&B-Queue
        end if
        New-B&B-Set ← Re-root$((\mathcal{T}',K'))$ + $(\mathcal{T}',K')$
        Append all B&B nodes in New-B&B-Set to B&B-Queue
    end if
end while
return all irredundant complete topologies
```

Fig. 2. Outline of the RATS-tree algorithm.

We explore the B&B search tree in a breadth-first traversal order. The reason for the breadth-first expansion order is to facilitate as much pruning as possible. We prune the solution space as follows: each topology T is associated with a triple $(\mathrm{Cap}(\mathrm{T}), \mathrm{Slack}(\mathrm{T}), \mathrm{SQ}(\mathrm{T}))$, where $\mathrm{Cap}(\mathrm{T})$ is the total capacitance, $\mathrm{Slack}(\mathrm{T}) = \min_{s_i \in \mathrm{T}}(q_i - t_{\mathrm{T}}(s_i))$ with $t_{\mathrm{T}}(s_i)$ being the two-pole sink delay of s_i in T (Section II), and $\mathrm{SQ}(\mathrm{T})$ is the signal quality of the topology. Recall that we use the metrics $\lambda_i = 4m_i^2 - 3(m_i^1)^2$ as a measure of the degree of damping for sink s_i. If $\lambda_i > 0$, s_i is overdamped. If $\lambda_i < 0$, s_i is underdamped. Otherwise, it is critically damped. Therefore, we measure the signal quality of a tree T by $\mathrm{SQ}(\mathrm{T}) = \min_{s_i \in \mathrm{T}} \lambda_i$; the signal quality is measured by the worst signal response waveform among all sinks. If two RATS trees T and T' are rooted at the same node and cover the same set of sinks, then they share the same *alias*. Considering two topologies

T and T' that have the same alias, we say that T' is *redundant* if $\mathrm{Cap}(\mathrm{T}) \leq \mathrm{Cap}(\mathrm{T}'), \mathrm{Slack}(\mathrm{T}) \geq \mathrm{Slack}(\mathrm{T}'), \mathrm{SQ}(\mathrm{T}') \geq \mathrm{SQ}(\mathrm{T}')$, *and* at least one of the three inequalities is a strict inequality. The RATS-tree algorithm given in Fig. 2 assumes no pruning of the B&B search tree. To consider pruning, a B&B node with $(\mathcal{T}, K)$ is expanded only if *all* partial RATS trees in $\mathcal{T}$ are irredundant. After a new subtopology is constructed, the RATS-tree algorithm prunes the set of topologies that share the same alias as the newly created topology.

Although the pruning technique is very effective in keeping the solution space small, it should be noted that it is a heuristic, as the signal delays in different edges under a higher order *RLC* model are not independent as in the case for the Elmore delay model—the Elmore delay contributed by each edge, which is due to the edge resistance and the total downstream capacitance, can be computed independent of topology construction at the upstream. This is due to the additive property of the Elmore delay. Under the higher order delay model, however, adding a new edge at the upstream affects signal delays contributed by the downstream edges, since the $(p+1)$th order moments depend on the pth order moments (see Section IV). The fact that the addition of an upstream edge also has different levels of effect on the signal integrity of downstream sinks further complicates the matter. In Section IV, we present a bottom-up algorithm that computes the moments of sinks in a topology in an incremental fashion, exploiting the fact that the interconnect structure changes only slightly from an iteration to the next in the RATS-tree algorithm.

IV. Incremental Bottom-Up Moment Computation

Moments can be computed by the polynomial-time algorithms in [20] and [33]. However, these works compute moments by traversing the *entire tree* iteratively and do not allow incremental computation of moments as the tree topology changes. As a result, when the topology changes during routing tree construction, another round of iterative tree traversals is needed to recompute the moments. Even when we restrict the topology change to a simple addition of an *RLC* segment to the root of the original tree, which is the basis of our bottom-up topology construction algorithm, moments cannot be incrementally updated easily using the previous methods. Therefore, these previous approaches are not suitable for our RATS-tree algorithm. An independent study by [29] recently presented a method to incrementally compute moment in a bottom-up fashion for *RC* interconnects. However, it did not consider the inductance effect. In the following, we present our algorithm to handle moment computation of an *RLC* tree.

Consider an *RLC* tree T_v rooted by node v. For any node w in T_v, let m_w^p be the pth moment of node w and $\mathrm{C}^p_{\mathrm{T}_w} = \sum_{j \in \mathrm{T}_w} m_j^p \cdot \mathrm{C}_j$ be the total pth order moment weighted capacitance of T_w [33], where C_j is the capacitance connected to node j. The moment-weighted capacitance is also called the *subtree admittance* in [20]. Now, we add a new edge uv at the root of T_v to obtain a new tree T_u rooted at u (see Fig. 3). Let $\bar{\mathrm{C}}^p_{\mathrm{T}_w}$ be the new total pth order moment weighted capacitances of T_w for w in T_u. Similarly, let $\bar{m}_w^p$ be the new pth moment of node w in T_u. Let $\mathrm{R}_v, \mathrm{L}_v$, and C_v be the total resistance, inductance, and capacitance of the edge uv, respectively. In [33], moments were derived recursively as follows:

$$m_z^p = \begin{cases} 0, & \text{if } z \text{ is the root} \\ m_{\bar{z}}^p + \mathrm{R}_z \cdot \mathrm{C}^{p-1}_{\mathrm{T}_z} - \mathrm{L}_z \cdot \mathrm{C}^{p-2}_{\mathrm{T}_z}, & \text{if } z \neq \text{ root} \end{cases} \tag{3}$$

where $\bar{z}$ is the parent node of z. The following theorem illustrates how we can express $\bar{m}_w^p$ for $p \geq 0$ in terms of $\bar{m}_v^q$ and m_w^q for $q = 0 \cdots p$.

Theorem 1: For root node u

$$\bar{m}_u^p = \begin{cases} 0, & \text{if } p = -1 \quad \text{or} \quad p > 0 \\ 1, & \text{if } p = 0. \end{cases} \tag{4}$$

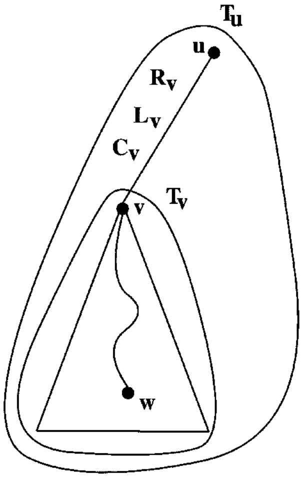

Fig. 3. Incremental moment computation after growing tree T_v with a new edge uv to form a new tree T_u.

For node v and any node w in T_v

$$\bar{m}_v^p = m_v^p, \quad \text{if } p = 0 \quad \text{or} \quad p = -1 \tag{5}$$

$$\bar{m}_w^p = m_w^p, \quad \text{if } p = 0 \quad \text{or} \quad p = -1. \tag{6}$$

For $p \geq 0$

$$\bar{\mathrm{C}}^{p-1}_{\mathrm{T}_v} = \bar{m}_v^{p-1}\mathrm{C}_v + \sum_{q=0}^{p-1} \bar{m}_v^{p-1-q}\mathrm{C}^q_{\mathrm{T}_v} \tag{7}$$

For $p \geq 1$

$$\bar{m}_v^p = \mathrm{R}_v\bar{\mathrm{C}}^{p-1}_{\mathrm{T}_v} - \mathrm{L}_v\bar{\mathrm{C}}^{p-2}_{\mathrm{T}_v} \tag{8}$$

$$\bar{m}_w^p = \sum_{q=0}^{p} \bar{m}_v^{p-q} m_w^q \quad \forall w \in \mathrm{T}_v. \tag{9}$$

Proof: Equations (4)–(6) follow from (3) directly. For $p = 0$, we can verify that (7) is trivially true. For $p = 1$, (7)–(9) are trivially true since the first moment is equivalent to the Elmore delay and one can easily verify that (7)–(9) compute the Elmore delay. We will prove that (7)–(9) hold for $p > 1$ by induction. □

Assuming that (7)–(9) are valid for $p = k > 1$, we consider $p = k + 1$. By the definition of $\bar{\mathrm{C}}^{p-1}_{\mathrm{T}_v}$ in [33]

$$\begin{aligned} \bar{\mathrm{C}}^{p-1}_{\mathrm{T}_v} &= \bar{m}_v^{p-1}\mathrm{C}_v + \sum_{w \in \mathrm{T}_v, w \neq v} \mathrm{C}_w \bar{m}_w^{p-1} \\ &= \bar{m}_v^{p-1}\mathrm{C}_v + \sum_{w \in \mathrm{T}_v, w \neq v} \mathrm{C}_w \left\{ \sum_{q=0}^{p-1} \bar{m}_v^{p-1-q} m_w^q \right\} \\ &= \bar{m}_v^{p-1}\mathrm{C}_v + \sum_{q=0}^{p-1} \bar{m}_v^{p-1-q} \sum_{w \in \mathrm{T}_v, w \neq v} \mathrm{C}_w m_w^q \\ &= \bar{m}_v^{p-1}\mathrm{C}_v + \sum_{q=0}^{p-1} \bar{m}_v^{p-1-q}\mathrm{C}^q_{\mathrm{T}_v}. \end{aligned}$$

Therefore, (7) is true.

According to (3), (8) is trivially true, as $\bar{m}_u^p = 0$. Now, let w be a child node of v. We first prove that (9) holds for w and then prove that (9) holds for all nodes in T_v by induction. From (3)

$$\begin{aligned}
\bar{m}_w^p &= \bar{m}_v^p + \mathrm{R}_w\bar{\mathrm{C}}_{\mathrm{T}_w}^{p-1} - \mathrm{L}_w\bar{\mathrm{C}}_{\mathrm{T}_w}^{p-2} \\
&= \bar{m}_v^p m_w^0 + \mathrm{R}_w \sum_{z\in\mathrm{T}_w} \bar{m}_z^{p-1}\mathrm{C}_z - \mathrm{L}_w \sum_{z\in\mathrm{T}_w} \bar{m}_z^{p-2}\mathrm{C}_z \\
&= \bar{m}_v^p m_w^0 + \mathrm{R}_w \sum_{z\in\mathrm{T}_w} \mathrm{C}_z \left\{\sum_{q=0}^{p-1} \bar{m}_v^{p-1-q} m_z^q\right\} \\
&\quad - \mathrm{L}_w \sum_{z\in\mathrm{T}_w} \mathrm{C}_z \left\{\sum_{q=0}^{p-2} \bar{m}_v^{p-2-q} m_z^q\right\} \\
&= \bar{m}_v^p m_w^0 + \sum_{q=0}^{p-1} \bar{m}_v^{p-1-q} \left\{\mathrm{R}_w \sum_{z\in\mathrm{T}_w} \mathrm{C}_z m_z^q\right\} \\
&\quad - \sum_{q=0}^{p-2} \bar{m}_v^{p-2-q} \left\{\mathrm{L}_w \sum_{z\in\mathrm{T}_w} \mathrm{C}_z m_z^q\right\} \\
&= \bar{m}_v^p m_w^0 + \bar{m}_v^{p-1}\mathrm{R}_w \sum_{z\in\mathrm{T}_w} \mathrm{C}_z \\
&\quad + \sum_{q=1}^{p-1} \bar{m}_v^{p-1-q} \left\{\mathrm{R}_w \sum_{z\in\mathrm{T}_w} \mathrm{C}_z m_z^q\right\} \\
&\quad - \sum_{q=0}^{p-2} \bar{m}_v^{p-2-q} \left\{\mathrm{L}_w \sum_{z\in\mathrm{T}_w} \mathrm{C}_z m_z^q\right\} \\
&= \bar{m}_v^p m_w^0 + \bar{m}_v^{p-1} m_w^1 \\
&\quad + \sum_{q=1}^{p-1} \bar{m}_v^{p-1-q} \left\{\mathrm{R}_w C_{\mathrm{T}_w}^q - \mathrm{L}_w C_{\mathrm{T}_w}^{q-1}\right\} \\
&= \bar{m}_v^p m_w^0 + \bar{m}_v^{p-1} m_w^1 + \sum_{q=1}^{p-1} \bar{m}_v^{p-1-q} m_w^{q+1} \\
&= \sum_{q=0}^{p} \bar{m}_v^{p-q} m_w^q.
\end{aligned}$$

Therefore, (9) holds for all child nodes of v, i.e., it is true for all descendant nodes one hop from v. Assuming that (9) holds for all descendant nodes h hops from v, we shall prove that it also holds for all descendant nodes $h+1$ hops from v. Let w be a descendant node h hops from v and z be a child node of w. From (3)

$$\begin{aligned}
\bar{m}_z^p &= \bar{m}_w^p + \mathrm{R}_z\bar{\mathrm{C}}_{\mathrm{T}_z}^{p-1} - \mathrm{L}_z\bar{\mathrm{C}}_{\mathrm{T}_z}^{p-2} \\
&= \sum_{q=0}^{p} \bar{m}_v^{p-q} m_w^q + \mathrm{R}_z \sum_{z'\in\mathrm{T}_z} \bar{m}_{z'}^{p-1}\mathrm{C}_{z'} - \mathrm{L}_z \sum_{z'\in\mathrm{T}_z} \bar{m}_{z'}^{p-2}\mathrm{C}_{z'} \\
&= \bar{m}_v^p + \sum_{q=1}^{p} \bar{m}_v^{p-q} m_w^q + \mathrm{R}_z \sum_{z'\in\mathrm{T}_z} \mathrm{C}_{z'} \left\{\sum_{q=0}^{p-1} \bar{m}_v^{p-1-q} m_{z'}^q\right\} \\
&\quad - \mathrm{L}_z \sum_{z'\in\mathrm{T}_z} \mathrm{C}_{z'} \left\{\sum_{q=0}^{p-2} \bar{m}_v^{p-2-q} m_{z'}^q\right\} \\
&= \bar{m}_v^p m_z^0 + \sum_{q=1}^{p} \bar{m}_v^{p-q} m_w^q \\
&\quad + \sum_{q=0}^{p-1} \bar{m}_v^{p-1-q} \left\{\mathrm{R}_z \sum_{z'\in\mathrm{T}_z} \mathrm{C}_{z'} m_{z'}^q\right\} \\
&\quad - \sum_{q=0}^{p-2} \bar{m}_v^{p-2-q} \left\{\mathrm{L}_z \sum_{z\in\mathrm{T}_z} \mathrm{C}_{z'} m_{z'}^q\right\}
\end{aligned}$$

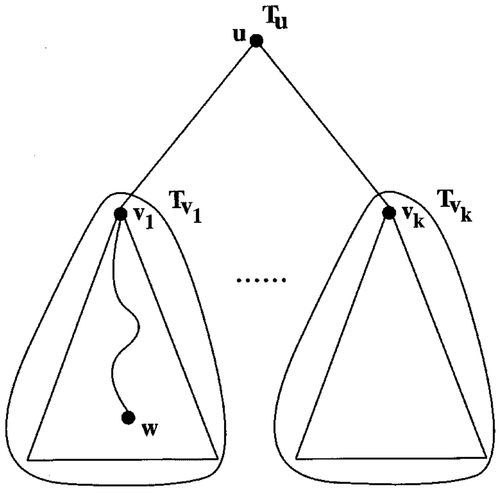

Fig. 4. Merging of k topologies T_{v_i} for $i = 1\cdots k$ at u.

$$\begin{aligned}
&= \bar{m}_v^p m_z^0 + \bar{m}_v^{p-1} \left\{m_w^1 + \mathrm{R}_z \sum_{z'\in\mathrm{T}_z} \mathrm{C}_{z'}\right\} \\
&\quad + \sum_{q=2}^{p} \bar{m}_v^{p-q} m_w^q + \sum_{q=1}^{p-1} \bar{m}_v^{p-1-q} \left\{\mathrm{R}_z \sum_{z'\in\mathrm{T}_z} \mathrm{C}_{z'} m_{z'}^q\right\} \\
&\quad - \sum_{q=0}^{p-2} \bar{m}_v^{p-2-q} \left\{\mathrm{L}_z \sum_{z'\in\mathrm{T}_z} \mathrm{C}_{z'} m_{z'}^q\right\} \\
&= \bar{m}_v^p m_z^0 + \bar{m}_v^{p-1} m_z^1 \\
&\quad + \sum_{q=2}^{p} \bar{m}_v^{p-q} \left\{m_w^q + \mathrm{R}_z C_{\mathrm{T}_z}^{q-1} - \mathrm{L}_z C_{\mathrm{T}_z}^{q-2}\right\} \\
&= \bar{m}_v^p m_z^0 + \bar{m}_v^{p-1} m_z^1 + \sum_{q=2}^{p} \bar{m}_v^{p-q} m_z^q \\
&= \sum_{q=0}^{p} \bar{m}_v^{p-q} m_z^q.
\end{aligned}$$

Therefore, by induction on the number of hops a descendant node is from v, (9) holds for all nodes in T_v. Then, by induction on the order p, (7)–(9) hold. □

The theorem implies that if we keep $\mathrm{C}_{\mathrm{T}_v}^i$ for $i = 0\cdots p-1$, we can compute up to the pth order the new moments for v by computing $\bar{\mathrm{C}}_{\mathrm{T}_v}^{i-1}$ followed by $\bar{m}_v^i$ for $i = 0\cdots p$ by (7) and (8). Then, we can update the moments of all the sinks in the topology by (9). From the above theorem, we can state the following corollary, which allows us to incrementally update sink moments during the merging operation and compute the total pth order moment weighted capacitance at the new root u.

Corollary 1: Consider k topologies, denoted T_{v_i} for $i = 1\cdots k$. If the k topologies are merged at a new node u as in Fig. 4, then for $p \geq 0$ and $i = 1\cdots k, \bar{\mathrm{C}}_{\mathrm{T}_{v_i}}^{p-1}, \bar{m}_{v_i}^p$ and $\bar{m}_w^p$ $\forall w \in \mathrm{T}_{v_i}$ can be computed by (7)–(9), respectively. Let c_u^s be the sink capacitance at the root node u, i.e., $c_u^s = 0$ if u is not a sink. For $p \geq 0$

$$\bar{\mathrm{C}}_{\mathrm{T}_u}^{p-1} = \bar{m}_u^{p-1} c_u^s + \sum_{i=0}^{k} \bar{\mathrm{C}}_{\mathrm{T}_{v_i}}^{p-1}. \tag{10}$$

From Theorem 1 and Corollary 1, the time complexity to update the moments of n sinks in a tree is $O(n\cdot p^2)$. The auxiliary space requirement is $O(p)$. On the other hand, the time complexity of the method proposed by [33] is $O(g \cdot p)$, where g is the total number of grid nodes in the tree and the auxiliary space requirement is $O(g)$. Since our RATS-tree algorithm is based on the Hanan-grid, g could be in the order of $O(n^2)$.

We can integrate the incremental bottom-up moment computation algorithm with our RATS-tree algorithm easily. For each irredundant topology constructed by our algorithm, we keep their corresponding sink moments (up to a prespecified pth order) for the topology. As we grow a topology along the path of Hanan grid points toward the new root in the merge operation, we compute the length of each new edge and derive the interconnect resistance, inductance and capacitance. These *RLC* parasitics are used to update the moments of the sinks using Theorem 1. We then use Corollary 1 to compute the weighted capacitances at the new root. The RATS-tree algorithm then prunes the set of topologies that share the same alias as the newly created topology.

V. Experimental Results

We implement the RATS-tree algorithm in C++ language and evaluate the algorithm for MCM designs in two groups of experiments. The first experiment demonstrates the ability of the RATS-tree algorithm to construct a set of topologies that provide a smooth tradeoff among signal delay, waveform, and routing area. The second experiment demonstrates how RATS-tree solutions could provide a performance-driven global router new opportunities in reducing overall routing congestion level.

A. Results on Random Nets

In the first experiment, we use randomly generated netlists with six to 12 terminals on a 10 × 10-cm MCM substrate. In the first set of experiments, we run the RATS-tree algorithm under the path-length formulation to investigate the tradeoff between the path-length and the routing cost of the topologies generated. Ten random n-pin nets are generated for each n ranging from six to 12. We apply the RATS-tree algorithm under the two-pole model. The purpose of this experiment is to investigate the impact of routing topology on signal delay and integrity. The interconnect parameters that are used by our algorithms for moment, delay, and signal quality computations are obtained from the Micro Module System (MMS) D500 process on aluminum offered through MIDAS. Assuming a nominal width of 19 μm, the interconnect resistance, inductance, and capacitance are 236.84 Ω/m, 301.49 nH/m, and 128.99 pF/m, respectively. The load capacitance of each sink is assumed to be 1 pF, and the driver resistances range from 10 to 30 Ω, depending on the size of the net and the proximity of the terminals.

For each net, we set the required arrival time of sink s_i to be $k \times \sqrt{\mathrm{LC}}d(s_0, s_i)$, where $k > 1$; k is larger than one in order to account for the transition delay. For each net, our algorithm constructs a large class of topologies satisfying the delay requirements. We then run SPICE simulations using the transmission line model to evaluate the sink delay and measure signal integrity in terms of the signal settling time and voltage overshoot of these constructed topologies. The signal delay is the time taken for the output response waveform to reach 90% of V_{dd} (assuming a rising signal) for the first time. The signal settling time is the time taken for the signal to settle above 90% of V_{dd}. In general, a sufficiently large set of topologies is generated by our algorithm, which provides tradeoffs among maximum sink delay, signal settling time, voltage overshoot, and routing cost.

Table I shows the maximum delays, signal settling times, voltage overshoots, and routing costs for the topologies generated by our RATS-tree algorithm for one of the randomly generated 9-pin nets for $k = 2, 3$, and 6. We also include the topology generated by the Borah–Owens–Irwin (BOI) Steiner algorithm proposed by [2]. Both the delay and settling time of each topology are in nanoseconds and the total wire capacitance is in picofarads. The voltage overshoot is normalized with respect to V_{dd}.

TABLE I
Tradeoff Among Maximum Sink Delay, Signal Settling Time, Voltage Overshoot, and Routing Cost for RATS Trees Generated Under $k = 2, 3$, and 6

k	Topology	Max-delay (ns)	Settling Time (ns)	Overshoot	Wire Cap. (pF)
2	*RATS*1 ([26])	2.15	0.83	0.10	33.4
	*RATS*2	2.28	0.76	0.07	38.3
	*RATS*3	2.26	0.00	0.04	37.7
	*RATS*4	2.24	0.71	0.06	35.7
	*RATS*5	2.18	0.82	0.10	31.9
3	*RATS*6	2.65	0.00	0.09	29.9
	*RATS*7	2.39	0.58	0.15	30.5
6	*RATS*8	2.84	0.00	0.12	26.7
	*RATS*9	2.81	0.00	0.11	27.4
-	BOI [2]	2.96	0.00	0.11	26.7

Also included are the maximum sink delay, signal settling time, voltage overshoot, and routing cost for the BOI Steiner topology [2].

When $k = 2$, most of the RATS trees generated are shortest path Steiner tree. As we can see from Fig. 5(a)–(e), except for RATS5, the topologies RATS1-4 are all shortest path Steiner tree. In fact, RATS1 is an optimal Steiner arborescence [26]. While it is the best among RATS1-5 in terms of maximum delay, RATS1's total wire capacitance, settling time, and voltage overshoot are not necessary the smallest. The topology generated by [2] shown in Fig. 5(j) has the longest signal delay, but the least total wire capacitance, which is 20% smaller than that of RATS1. Fig. 5(f)–(g) and (h)–(i) shows topologies generated for $k = 3$ and $k = 6$, respectively. Note that the total wire capacitance of RATS8 is 20% smaller than that of RATS1. In fact, the total wire capacitance of RATS8 is identical to that of the BOI Steiner topology generated by [2]. However, RATS8 has a better signal delay because the RATS-tree algorithm keeps the solution with a larger delay slack during the pruning. In total, 16 and 20 topologies are generated for $k = 3$ and 6, respectively. Note that if we do not consider pruning, the numbers could be much larger. There are also overlaps among the three sets of topologies for $k = 2, 3$, and 6.

The runtime required by the RATS-tree algorithm increases as the net cardinality increases. For example, the worst case CPU time for 12-pin nets is 152 s, whereas the worst case CPU time for 6-pin nets is 0.3 s. In general, the average CPU time is much lower; the average CPU times for 12-pin nets and 6-pin nets are 22 and 0.2 s, respectively. It is interesting to note that for the larger examples ($n = 10, 11$, and 12), the runtimes increase when we increase k to a certain extent. Beyond that, the runtimes decrease. We believe that the runtime is reflective of the number of (sub)topologies generated by the RATS-tree algorithm. When k is either very small or very large, we observe that the pruning is effective in reducing the number of (sub)topologies generated, thereby cutting down the runtimes.

B. Integration With MINOTAUR Global Router

The ability of the RATS-tree algorithm to generate a set of routing topologies for a timing-critical net complements a performance-driven MCM global router called MINOTAUR [8], [30]. Presented with a set of candidate routing topologies for each timing-critical net, the MINOTAUR global router has the flexibility to choose a topology from the set of candidate topologies of each net (not necessary of the least cost) to minimize the *overall* congestion and possibly optimize other objective functions of the global routing solution. Essentially, the global router considers routing of multiple nets simultaneously; it performs a

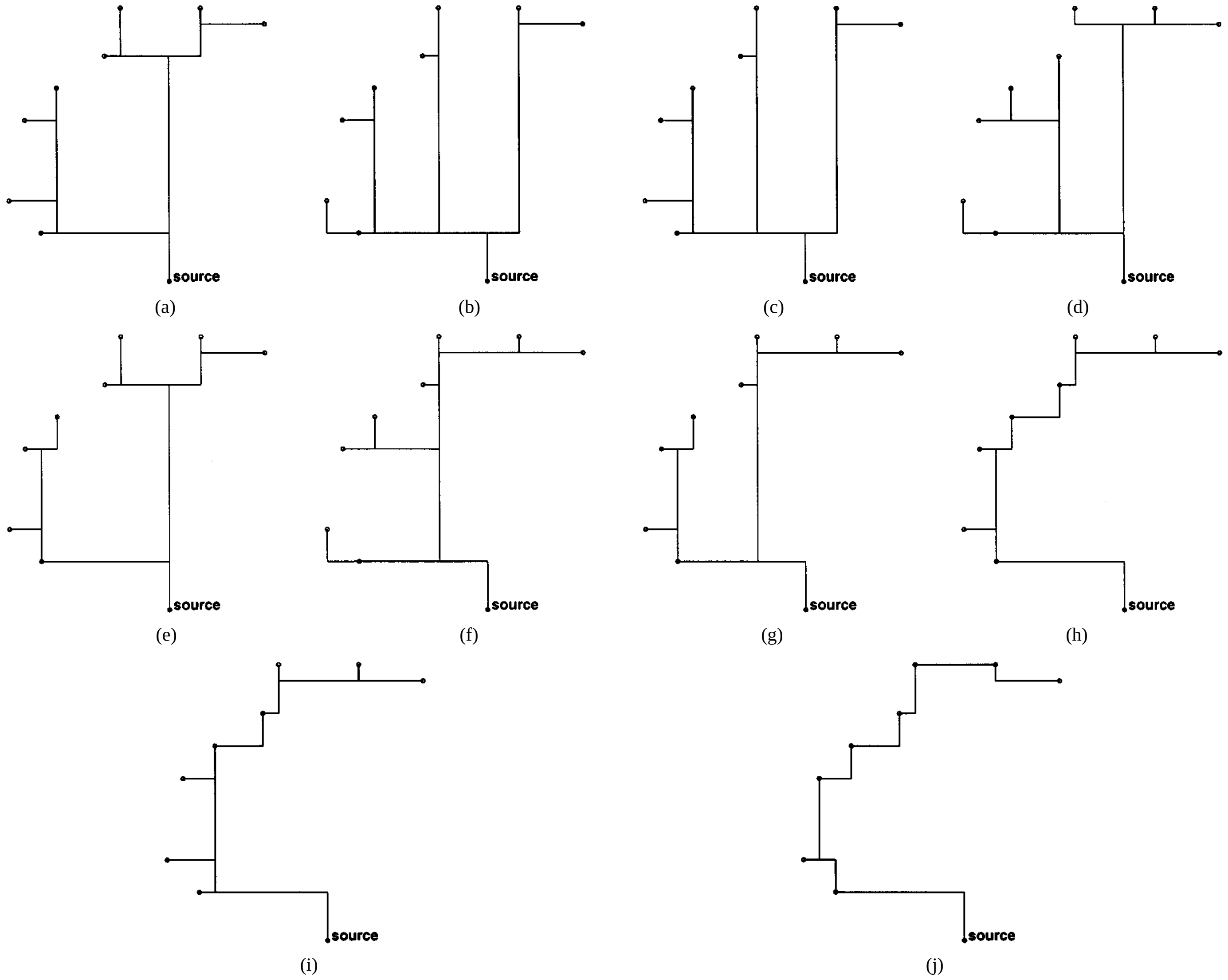

Fig. 5. Topologies generated by the RATS-tree algorithm for a 9-pin net. (a) RATS1, $k = 2$. (b) RATS2, $k = 2$. (c) RATS3, $k = 2$. (d) RATS4, $k = 2$. (e) RATS5, $k = 2$. (f) RATS6, $k = 3$. (g) RATS7, $k = 3$. (h) RATS8, $k = 6$. (i) RATS9, $k = 6$. (j) Topology generated by the BOI Steiner heuristic [2].

topology selection heuristic called *iterative deletion* to remove candidate topologies of all nets one at a time until each net is left with only one candidate topology. The objective is to select a set of compatible topologies for congestion reduction.

To evaluate the effectiveness of the RATS-tree algorithm in providing a set of compatible topologies for different nets, we perform experiments on two Microelectronics and Computer Technology Corporation (MCC) multichip module benchmark circuits *mcc1* and *mcc2* that have been used in [23]. The distributions of net sizes in *mcc1* and *mcc2* are shown in Table II [30].

TABLE II
DISTRIBUTION OF NET CARDINALITIES FOR THE BENCHMARKS *mcc1* AND *mcc2*

	Net cardinality			
Benchmark	2	3	4	5+
mcc1	608	87	50	57
mcc2	6699	415	4	0

Only a small portion of the nets are large enough to obtain benefit from interconnect topology optimization.

We first construct approximate minimum Steiner tree topologies [2] for nets in *mcc1* and *mcc2*. We sort the pins from these nets in descending order of their delays and designate the first T% of the sorted pins as critical pins for optimization. For each of these critical pins, we set its required arrival time to be X% smaller than its delay in the corresponding approximate minimum Steiner tree topology. For each of the remaining $(100 - T)$% noncritical pins, we use its delay in the corresponding approximate minimum Steiner tree topology as its required arrival time. Subsequently, for every net that has critical pins with tighter timing constraints, we apply the RATS-tree algorithm to generate candidate topologies that meet the specified timing constraints for *all* pins in the net. In our experiments, we evaluate the maximum border congestion, which measures the number of crossings from one routing region to another, for the benchmarks using $T = \{5\%, 10\%\}$ and $X = \{5\%, 10\%, 15\%\}$. The number of nets optimized and the minimum, maximum, and average number of candidate RATS trees for the nets are shown in Table III [30]. These candidate RATS trees achieve the specified X% delay reduction for all the T% critical pins without compromising the performance of noncritical pins. Note that most timing-critical nets in *mcc2* do not have multiple candidate topologies because most of them are 2-pin nets and the RATS-tree algorithm considers routing on a Hanan grid only.

We perform experiments using two general approaches in the topology selection problem to illustrate the benefits of considering

TABLE III
NUMBER OF NETS OPTIMIZED AND THE MINIMUM, MAXIMUM, AND AVERAGE NUMBER OF RATS TREES FOR EACH NET

Bench		Total	$X=5\%$			$X=10\%$			$X=15\%$		
mark	T	Nets	min	max	avg	min	max	avg	min	max	avg
mcc1	5%	38	4	12	8.16	4	12	8.13	4	12	8.13
mcc1	10%	51	3	14	8.67	3	17	8.71	3	14	8.55
mcc2	5%	614	1	6	1.02	1	4	1.02	1	4	1.02
mcc2	10%	1278	1	6	1.01	1	4	1.01	1	4	1.01

We seek to obtain $X\%$ delay reduction for the top $T\%$ high-delay pins.

TABLE IV
IMPACT OF PERFORMANCE OPTIMIZATION ON *mcc1* AND *mcc2* IF THE GLOBAL ROUTER CONSIDERS (a) A SINGLE-CANDIDATE HIGH-PERFORMANCE INTERCONNECT STRUCTURE OR (b) MULTIPLE HIGH-PERFORMANCE TOPOLOGIES

Benchmark	T	Single High-Performance Topology					
		$X=5\%$		$X=10\%$		$X=15\%$	
		C_{max}	C_{avg}	C_{max}	C_{avg}	C_{max}	C_{avg}
mcc1	0%	33	23.0	33	23.0	33	23.0
mcc1	5%	34	22.6	34	22.6	34	22.6
mcc1	10%	41	22.3	41	22.3	41	22.3
mcc2	0%	167	93.1	167	93.1	167	93.1
mcc2	5%	184	93.6	184	93.6	184	93.6
mcc2	10%	223	93.3	223	93.3	223	93.3

(a)

Benchmark	T	Multiple Topologies with Iterative Deletion					
		$X=5\%$		$X=10\%$		$X=15\%$	
		C_{max}	C_{avg}	C_{max}	C_{avg}	C_{max}	C_{avg}
mcc1	5%	32	22.6	33	22.7	33	22.7
mcc1	10%	33	22.8	33	22.8	33	22.8
mcc2	5%	184	93.6	184	93.6	185	93.6
mcc2	10%	222	93.3	222	93.3	222	93.3

(b)

In these experiments, we select $T\%$ of the high-delay pins and attempt to obtain at least an $X\%$ delay reduction for each. If no performance optimization $(T = 0\%)$ is required, the maximum $(\mathrm{C}_{\max})$ and average $(\mathrm{C}_{\mathrm{avg}})$ congestion levels obtained by the MINOTAUR global router for *mcc1* are 33 and 23, respectively. The maximum and average congestion levels for *mcc2* are 167 and 93.1, respectively.

multiple candidate topologies. In the first approach, we select a single high-performance RATS tree (the one with minimum area) for each performance critical net; this is the "traditional" approach taken by a performance-driven global router. In the second approach, the RATS-tree algorithm generate (possibly) many candidate topologies for each timing-critical net and the MINOTAUR global router applies the iterative deletion heuristic to select a set of compatible topologies for several timing-critical nets. Results of these experiments are shown in Table IV, [30]. Note that without considering performance optimization $(T = 0\%)$, the maximum congestion level, denoted $\mathrm{C}_{\max}$, and the average congestion level, denoted $\mathrm{C}_{\mathrm{avg}}$, for *mcc1* are 33 and 23, respectively. The maximum and average congestion levels for *mcc2* are 167 and 93.1, respectively.

The results in Table IV(a) show that when the global router is restricted to a single high-performance topology for each timing-critical net, the performance-driven global routing will result in a considerably higher congestion level. Note that in this experiment, the RATS tree returns only a single high-performance (minimum area) RATS tree, possibly among many high-performance RATS trees constructed, to the MINOTAUR global router. We believe that similar results will hold for any topology optimization algorithm that returns a single topology for each net to the MINOTAUR global router. In fact, most of the existing topology optimization algorithms construct only a single high-performance topology.

Table IV(b) shows the results when the MINOTAUR global router considers several candidate topologies for each timing-critical net, select one that is compatible with those of other timing-critical nets. It is interesting to observe that for *mcc1*, the maximum and average global congestion can be maintained at levels similar to those when no performance optimization is required. On the other hand, the congestion levels increase significantly for *mcc2* whether we consider a single high-performance interconnect structure or multiple candidate topologies. A possible explanation for the above behaviors is that there are several candidate topologies available for each timing-critical net in *mcc1* as shown in Table III. However, most timing-critical nets in *mcc2* do not have multiple candidate topologies due to the fact that most of them are 2-pin nets and the RATS-tree algorithm considers routing on a Hanan grid only. As the routings for these nets are restricted, the overall congestion levels increase as in Table IV(a).[2]

VI. CONCLUSION

In this paper, we describe a RATS-tree construction algorithm under a higher order *RLC* interconnect model. The RATS-tree algorithm optimizes signal waveform, not just delay; it considers the impact of routing on signal delays and response waveforms by incrementally computing sink moments in a bottom-up manner during topology construction. Due to its capability to handle a large class of topologies, the RATS-tree algorithm returns not one, but a set of routing topologies that provide tradeoffs among routing cost, signal delay, and signal integrity. We also show that performance optimization needs to be carried out in a more global context; optimization that produces a single solution for each net is unlikely to work well. Coupling the RATS-tree algorithm with the MINOTAUR global router using the iterative deletion heuristic illustrates the potential of routing of multiple nets simultaneously.

ACKNOWLEDGMENT

The authors would like to thank Dr. D. Zhou of the University of Texas at Dallas, Dr. S. Muddu of SGI, Dr. Y.-Y. Hwang of Altera, Mr. K.-Y. Khoo of Verplex, and Mr. H. Leung of UCLA for their helpful discussions and critical comments. The authors would also like to thank the Avant! Corporation, whose generous donation of software is greatly appreciated.

REFERENCES

[1] K. D. Boese, A. B. Kahng, B. A. McCoy, and G. Robins, "Near-optimal critical sink routing tree constructions," *IEEE Trans. Computer-Aided*, vol. 14, pp. 1417–1436, Dec. 1995.

[2] M. Borah, R. M. Owens, and M. J. Irwin, "An edge-based heuristic for Steiner routing," *IEEE Trans. Computer-Aided Design*, vol. 13, pp. 1563–1568, Dec. 1994.

[3] C. P. Chen, Y. P. Chen, and D. F. Wong, "Optimal wire-sizing formula under the Elmore delay model," in *Proc. Design Automation Conf.*, 1996, pp. 487–490.

[4] J. Cong, A. B. Kahng, G. Robins, M. Sarrafzadeh, and C. K. Wong, "Provably good performance-driven global routing," *IEEE Trans. Computer-Aided Design*, vol. 11, pp. 739–752, June 1992.

[5] J. Cong and C.-K. Koh, "Interconnect layout optimization under higher order *RLC* model," in *Proc. Int. Conf. Computer-Aided Design*, 1997, pp. 713–720.

[2] Our preliminary study shows that graph-based RATS-tree construction [24] may increase the number of candidate topologies for timing-critical nets in *mcc2*; the minimum, maximum, and average number of candidates graph-based RATS trees per net are 1, 7, and 4.76, respectively for $T = 5\%$ and $X = 15\%$. The availability of options greatly improves the global routing solutions of *mcc2*. When graph-based RATS trees are used, the maximum congestion is maintained at 167 (as for $T = 0\%$) and the average congestion decreases slightly to 92.90.

[6] J. Cong and K. S. Leung, "Optimal wiresizing under the distributed Elmore delay model," *IEEE Trans. Computer-Aided Design*, vol. 14, pp. 321–336, Mar. 1995.

[7] J. Cong, K. S. Leung, and D. Zhou, "Performance-driven interconnect design based on distributed *RC* delay model," in *Proc. Design Automation Conf.*, 1993, pp. 606–611.

[8] J. Cong and P. H. Madden, "Performance driven multi-layer general area routing for PCB/MCM designs," in *Proc. Design Automation Conf.*, June 1998, pp. 356–361.

[9] J. Cong, L. He, C.-K. Koh, and P. H. Madden, "Performance optimization of VLSI interconnect layout," *Integr. VLSI J.*, vol. 21, no. 1–2, pp. 1–94, Nov. 1996.

[10] W. C. Elmore, "The transient response of damped linear networks with particular regard to wide-band amplifiers," *J. Appl. Phys.*, vol. 19, no. 1, pp. 55–63, Jan. 1948.

[11] J. P. Fishburn, "Shaping a VLSI wire to minimize Elmore delay," in *Proc. Eur. Design and Test Conf.*, 1997, pp. 244–251.

[12] Y. Gao and D. F. Wong, "Shaping a VLSI wire to minimize delay using transmission line model," in *Proc. Int. Conf. Computer-Aided Design*, 1998, pp. 611–616.

[13] R. Gupta and L. T. Pileggi, "Constrained multivariable optimization of transmission lines with general topologies," in *Proc. Int. Conf. Computer-Aided Design*, 1995, pp. 130–137.

[14] R. Gupta and L. T. Pillage, "OTTER: Optimal termination of transmission lines excluding radiation," in *Proc. Design Automation Conf.*, 1994, pp. 640–645.

[15] M. Hanan, "On Steiner's problem with rectilinear distance," *SIAM J. Appl. Math.*, vol. 14, pp. 255–265, 1966.

[16] X. Hong, T. Xue, E. S. Kuh, C. K. Cheng, and J. Huang, "Performance-driven Steiner tree algorithms for global routing," in *Proc. Design Automation Conf.*, 1993, pp. 177–181.

[17] H. Hou, J. Hu, and S. S. Sapatnekar, "Non-Hanan routing," *IEEE Trans. Computer-Aided Design*, vol. 18, pp. 436–444, Apr. 1999.

[18] J. Hu and S. S. Sapatnekar, "Simultaneous buffer insertion and nonhanan optimization for vlsi interconnect under a higher order AWE model," in *Proc. Int. Symp. Physical Design*, 1999, pp. 133–138.

[19] J. H. Huang, A. B. Kahng, and C.-W. A. Tsao, "On the bounded-skew routing tree problem," in *Proc. Design Automation Conf.*, June 1995, pp. 508–513.

[20] A. B. Kahng and S. Muddu, "Two-pole analysis of interconnection trees," in *Proc. IEEE Multi-Chip Module Conf.*, Jan. 1995, pp. 105–110.

[21] ——, "An analytical delay model for *RLC* interconnects," in *Proc. IEEE Int. Symp. Circuits and Systems*, May 1996, pp. 4.237–4.240.

[22] A. B. Kahng and G. Robins, "A new class of iterative Steiner tree heuristics with good performance," *IEEE Trans. Computer-Aided Design*, vol. 11, pp. 893–902, July 1992.

[23] K. Y. Khoo and J. Cong, "An efficient multilayer MCM router based on four-via routing," *IEEE Trans. Computer-Aided Design*, vol. 14, pp. 1277–1290, Oct. 1995.

[24] C.-K. Koh and P. H. Madden, "Manhattan or non-Manhattan? A study of alternative VLSI routing architectures," in *10th Great Lakes Symp. VLSI*, Mar. 2000, pp. 47–52.

[25] B. Krauter, R. Gupta, J. Willis, and L. T. Pileggi, "Transmission line synthesis," in *Proc. Design Automation Conf.*, 1995, pp. 358–363.

[26] K.-S. Leung and J. Cong, "Fast optimal algorithms for the minimum rectilinear Steiner arborescence problem," in *Proc. IEEE Int. Symp. Circuits and Systems*, vol. 3, 1997, pp. 1568–1571.

[27] J. Lillis, C. K. Cheng, and T. T. Y. Lin, "Optimal wire sizing and buffer insertion for low power and a generalized delay model," in *Proc. Int. Conf. Computer-Aided Design*, Nov. 1995, pp. 138–143.

[28] J. Lillis, C. K. Cheng, T. T. Y. Lin, and C. Y. Ho, "New performance driven routing techniques with explicit area/delay tradeoff and simultaneous wire sizing," in *Proc. Design Automation Conf.*, June 1996, pp. 395–400.

[29] F.-J. Liu, J. Lillis, and C.-K. Cheng, "A new layout-driven timing model for incremental layout optimization," in *Proc. Asia South Pacific Design Automation Conf.*, 1997, pp. 127–131.

[30] P. H. Madden, "High performance VLSI global routing," Ph.D. dissertation, Univ. California, Los Angeles, CA, 1998.

[31] N. Menezes, S. Pullela, F. Dartu, and L. T. Pillage, "*RC* interconnect synthesis—A moment fitting appraoch," in *Proc. Int. Conf. Computer-Aided Design*, 1994, pp. 418–425.

[32] T. Xue, E. S. Kuh, and Q. Yu, "A sensitivity-based wiresizing approach to interconnect optimization of lossy transmission line topologies," in *Proc. IEEE Multi-Chip Module Conf.*, 1996, pp. 117–121.

[33] Q. Yu and E. S. Kuh, "Exact moment matching model of transmission lines and application to interconnect delay estimation," *IEEE Trans. VLSI Syst.*, vol. 3, pp. 311–322, June 1995.

[34] H. Zhou, D. F. Wong, I.-M. Liu, and A. Aziz, "Simultaneous routing and buffer insertion with restrictions on buffer locations," in *Proc. Design Automation Conf.*, 1999, pp. 96–99.

Scheduling of Microfluidic Operations for Reconfigurable Two-Dimensional Electrowetting Arrays

Jie Ding, Krishnendu Chakrabarty, and Richard B. Fair

***Abstract*—We present an architectural design and optimization methodology for performing biochemical reactions using two-dimensional (2-D) electrowetting arrays. We define a set of basic microfluidic operations and leverage electronic design automation principles for system partitioning, resource allocation, and operation scheduling. Fluidic operations are carried out through the electrostatic configuration of a set of grid points. While concurrency is desirable to minimize processing time, the size of the 2-D array limits the number of concurrent operations of any type. Furthermore, functional dependencies between the operations also limit concurrency. We use integer linear programming to minimize the processing time by automatically extracting parallelism from a biochemical assay. As a case study, we apply our optimization method to the polymerase chain reaction, which is an important step in many lab-on-a-chip biochemical applications.**

***Index Terms*—Architectural optimization, integer linear programming, microelectrofluidics, partition map, reconfigurable architecture, scheduling.**

I. Introduction

Composite microsystems that incorporate microelectromechanical systems (MEMS) and microelectrofluidic systems (MEFS) are emerging as the next generation of system-on-a-chip (SoC) designs. These systems combine microstructures with solid-state electronics to integrate multiple energy domains, such as electrical, mechanical, and fluidic. The combination of microelectronics and microstructures is enabling a new class of integrated systems targeted at environmental sensing, actuation and control, biomedical analyses, agent detection, and precision fluid dispensing.

Microfluidics not only offers size reduction, e.g., in small medical implants and minimal-invasive surgery, but it also reduces power dissipation and increases system reliability. Microfluidics allows us to control small amounts of fluids for precision dispensing (microdosing) and reduce reagent consumption for online chemical analysis and real-time process monitoring. However, current-generation of MEFS are application-specific and they are incapable of performing a collection of differing analyses or procedures. We now need design methodologies and tools that allow microfluidic devices to be assembled into an SoC that can perform a variety of tasks supporting a diverse set of applications. This SoC can then be readily reconfigured and reused for chemical detection, analysis, diagnostics, and dispensing. While a number of com-

Manuscript received December 11, 2000; revised July 29, 2001. This work was supported in part by the Defense Advanced Research Projects Agency under Contract F30602-98-2-0140. This paper was recommended by Associate Editor R. Gupta.

The authors are with the Department of Electrical and Computer Engineering, Duke University, Durham, NC 27708 USA.

Publisher Item Identifier S 0278-0070(01)10640-8.

0278–0070/01$10.00 © 2001 IEEE