Planar-DME: Improved Planar Zero-Skew Clock Routing With Minimum Pathlength Delay*

Andrew B. Kahng and Chung-Wen Albert Tsao

UCLA Computer Science Dept., Los Angeles, CA 90024-1596 USA

Abstract

Clock routing has become a critical issue in the layout design of high-performance systems. We show that the two passes (bottom-up and top-down) of the DME algorithm [2, 3, 4, 8] can be replaced by a single top-down pass, which yields exactly the same (optimal) solution. From this, we develop a top-down algorithm which dynamically determines and embeds the clock tree topology, such that (i) the embedding is guaranteed to be planar, and (ii) the result has provably minimum total wirelength and minimum pathlength delay for that topology. A simple version of our method produces planar exact zero-skew solutions with total wirelengths that are competitive with the best non-planar exact zero-skew results in the literature.

1 Introduction

The placement phase of physical layout determines positions for the synchronizing elements of a circuit, which are the *sinks* of the clock net. Large cell-based designs can have thousands of sinks in a clock net, located arbitrarily throughout the layout region. Following [2, 4], we denote the set of sink locations in a clock routing instance as $S = \{s_1, s_2, \ldots, s_n\} \subset \Re^2$. A *connection topology* is a rooted binary tree, G, which has n leaves corresponding to the sinks in S. A *clock tree* $T(S)$ is an embedding of the connection topology in the Manhattan plane, i.e., a placement in $\Re^2$ which assigns each internal node $v \in G$ to a location $pl(T, v)$ (alternatively, we denote the placement of v simply as $pl(v)$). Clock trees with internal nodes of degree greater than three arise when edges of the binary tree have zero length. The clock tree is rooted at the *source*, s_0, and hence any edge between a parent node v and its child w may be identified with the child node; we denote this edge as e_w. The *cost* of the edge e_w is simply its wirelength, denoted $|e_w|$; this is always at least as large as the Manhattan distance between the endpoints of the edge. The cost of $T(S)$ is the total wirelength of the edges in $T(S)$.

For a given clock tree $T(S)$, let $t(s_0, s_i)$ denote the signal propagation time from source s_0 to sink s_i. The *skew* of $T(S)$ is the maximum value of $|t(s_0, s_i) - t(s_0, s_j)|$ over all $s_i, s_j \in S$. If the skew of $T(S)$ is zero then $T(S)$ is a *zero skew tree* (ZST). As in related work on planar clock routing [15], we address this problem in the context of the linear delay approximation: linear delay is simply the edge length, allowing geometric intuitions to be applied. In Section 5, we note extensions of our method to Elmore delay[1]. For the linear delay model, [5] observed that the general minimum-cost bounded-skew routing problem is NP-hard. On the other hand, [2, 3, 4, 8] show that the optimal ZST $T(S)$ having a *prescribed* connection topology G can be found in $O(|S|)$ time.

*Partial support for this work was provided by NSF MIP-9223740 and NSF MIP-9257982.

Overview of Contribution

Our work lies at the juncture of two recent directions in clock routing: (i) the DME (Deferred-Merge Embedding) algorithm of [2, 3, 4, 8] which embeds a given topology with guaranteed exact zero pathlength skew and minimum cost; and (ii) the method of [15], which is the first to guarantee a *planar-embeddable* clock routing solution. We show that under the linear delay model, the two passes (bottom-up and top-down) of the DME algorithm can be replaced by a single top-down pass. While DME nominally requires a prescribed topology as input, our result allows the clock tree topology to be determined *dynamically* and *flexibly* in top-down fashion, even while it is being optimally embedded.

We thus develop a top-down algorithm which (i) determines a topology that is guaranteed to be *planar-embeddable*, and (ii) simultaneously embeds this topology in the Manhattan plane. The resulting tree has provably minimum total wirelength and minimum pathlength delay for its topology. A very simple version of our method produces planar solutions with total wirelengths that are competitive with the best *non-planar* exact zero-skew results in the literature [9, 4]. We also obtain an average of 15.5% wirelength savings over the previous planar routing algorithm of [15].

2 Previous Work

2.1 Minimum-Cost Zero-Skew Trees

The first clock tree construction for cell-based layouts with arbitrary sink locations was that of Jackson et al. [11]; their MMM algorithm does recursive top-down partitioning of the set of sinks into two

[1]Our work does not address such issues as buffer insertion and wire-sizing; the most relevant design style is in some sense the "monolithic" single-buffer clocking approach [1, 7].

equal-sized subsets, always connecting the centroid of a set to the centroids of its subsets. Cong et al. [5] constructed clock tree topologies using a bottom-up matching approach. Their "KCR" algorithm obtains zero pathlength skew in practice (i.e., zero skew under the linear delay model) but has no theoretical guarantee. Tsay [14] was the first to guarantee exact zero skew for any input; this was accomplished with respect to the Elmore delay model.

The above methods all concentrate on generation of the clock tree *topology*: the topology is then embedded in the plane more or less arbitrarily as it is generated. The DME method, discovered independently by three groups [2, 3, 8], is the first algorithm to consider the generation *and the embedding* of the topology together: this linear-time algorithm optimally embeds any given topology in the Manhattan plane, using minimum total wirelength. We briefly review the DME method, following the development of [2].

Given a set of sinks S and a topology G, DME embeds internal nodes of G via: (i) a bottom-up phase that constructs a tree of *merging segments* which represent loci of possible placements of internal nodes in the ZST T; and (ii) a top-down phase that determines exact locations for the internal nodes in T (see Fig. 1, reproduced from [2]). In our discussion, the *distance* between two points s and t is the Manhattan distance $d(s,t)$, and the distance between two sets of points P and Q, $d(P,Q)$, is $max\{d(p,q) \mid p \in P \text{ and } q \in Q\}$.

In the **bottom-up phase** (Fig. 1a), each node $v \in G$ is associated with a merging segment, denoted $ms(v)$, which represents a set of possible placements of v. Let a and b be the children of node v, and let TS_a and TS_b denote the subtrees of merging segments rooted at a and b. The construction of $ms(v)$ depends on $ms(a)$ and $ms(b)$, hence the bottom-up processing order. We seek placements of v which allow TS_a and TS_b to be merged with *minimum* added wire while preserving zero skew. This means that we want to minimize $|e_a|+|e_b|$ in the output tree $T(S)$, while balancing sink delays from $pl(v)$.

We use the following terminology. A *Manhattan arc* is a line segment, possibly of zero length, with slope +1 or -1. The collection of points within a fixed distance of a Manhattan arc is called a *tilted rectangular region*, or *TRR*; the boundary of a TRR consists of Manhattan arcs. The Manhattan arc at the center of a TRR is its *core*; the *radius* of a TRR is the distance between its core and its boundary.

We can now recursively define $ms(v)$. If v is a sink s_i, then $ms(v) = \{s_i\}$. If v is an internal node, then $ms(v)$ is the set of all placements $pl(v)$ which merge TS_a and TS_b with minimum wire cost, i.e., all points within distance $|e_a|$ of $ms(a)$ and within distance $|e_b|$ of $ms(b)$. More precisely, $ms(v)$ is obtained by intersecting two TRRs, trr_a with core $ms(a)$ and radius $|e_a|$, and trr_b with core $ms(b)$ and radius $|e_b|$; i.e., $ms(v) = trr_a \cap trr_b$. A lemma in [2] shows that all merging segments so constructed are Manhattan arcs.

Given the tree of merging segments, the **top-down phase** (Fig. 1b) embeds each internal node v of G as follows: (i) if v is the root node, then DME selects any point in $ms(v)$ to be $pl(v)$[2]; or (ii) if v is an internal node other than the root, DME chooses $pl(v)$ to be any point on $ms(v)$ that is at distance $|e_v|$ or less from the embedding $pl(p)$ of v's parent[3]. With (ii), DME creates a square TRR trr_p with radius $|e_v|$ and core $\{pl(p)\}$; $pl(v)$ can be any point in $ms(v) \cap trr_p$.

Note that DME requires an input topology. Several works have thus proposed topologies that yield low-cost routing solutions when DME is applied. Below, we compare against the non-planar KCR+DME [2] and Greedy-DME [9] methods.

```
Procedure Build_Tree_of_Segments (G,S)
Input: Topology G; set of sink locations S
Output: Tree of merging segments TS containing
  ms(v) for each node v in G, and edge length |e_v|
  for each v ≠ s_0

for each node v in G (bottom-up order)
  if v is a sink node,
    ms(v) ← {pl(v)}
  else
    Let a and b be the children of v
    Calculate_Edge_Lengths(|e_a|,|e_b|)
    Create TRRs trr_a and trr_b as follows:
      core(trr_a) ← ms(a)
      radius(trr_a) ← |e_a|
      core(trr_b) ← ms(b)
      radius(trr_b) ← |e_b|
    ms(v) ← trr_a ∩ trr_b
  endif
```

(a) Bottom-up Phase: Construction of the tree of merging segments TS.

```
Procedure Find_Exact_Placements(TS)
Input: Tree of segments TS containing ms(v),
  and value of |e_v| for each node v in G
Output: ZST T(S)

for each internal node v in G (top-down order)
  if v is the root
    Choose any pl(v) ∈ ms(v)
  else
    Let p be the parent node of v
    Construct trr_p as follows:
      core(trr_p) ← {pl(p)}
      radius(trr_p) ← |e_v|
    Choose any pl(v) ∈ ms(v) ∩ trr_p
  endif
```

(b) Top-down Phase: Construction of the ZST by embedding internal nodes of G within TS.

Figure 1: The DME algorithm.

2.2 Planar-Embeddable Trees

It is not easy to realize the above "exact zero skew" solutions through actual placement of the wires into the layout plane. Often, many vias must be introduced. This difficulty was first noted by Zhu and Dai [15], who gave several reasons to seek a *planar-embeddable* clock routing solution. For instance: (i)

[2] If a clock source location clk has been specified, DME simply chooses $pl(s_0) \in ms(s_0)$ with minimum distance from clk and connects a wire directly from clk to $pl(s_0)$.

[3] Because $ms(p)$ was constructed such that $d(ms(v), ms(p)) \leq |e_v|$, there must exist feasible embedding points for node v.

the clock net may be best routed on a prescribed layer; (ii) routing on fewer layers makes the layout more process-variation independent and simplifies buffering optimizations; and (iii) single-layer routing eliminates delay and attenuation of the clock signal through vias. Thus, our goal is to solve the **Planar** Zero Skew Clock Routing Problem, i.e., given sink set S, find a *planar-embeddable* ZST $T(S)$ with minimum cost.

Notice that "planar-embeddable" intuitively means that the tree can be drawn in the plane without edges crossing. However, this concept is not easily characterized in the Manhattan plane, and the previous work of [15] implicitly relies on Euclidean planar-embeddability being sufficient for Manhattan planar-embeddability. Thus, we define two tree edges as crossing each other precisely when the corresponding *open* line segments in the Euclidean plane properly intersect (i.e., share exactly one point). This definition is necessitated by possible *degenerate* optimal planar clock routing solutions, where the embeddings of edges in the min-cost zero-skew tree are superposed. Fig. 2 shows four sinks having an optimal "planar" clock tree whose edges pass over each other. Since this overlapping can be made planar with minimum increase in wirelength, we accept such a degenerate solution as planar. This is also the convention of [15].

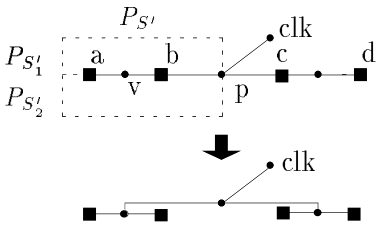


Figure 2: For certain inputs, overlapping of edges in an "optimal, planar" ZST is inevitable; however, the ZST can be easily made into a non-overlapping solution. Convex polygons $P_{S'}$, $P_{S'_1}$ and $P_{S'_2}$, and points p and v pertain to the discussion below concerning partitioning rules.

The planar clock routing method of [15] is as follows. Start with a tree containing only a connection from the source node to the furthest sink. At each iteration, a sink outside the current tree is connected to a "balance point" in the tree, i.e., via a connection to an existing edge such that zero pathlength skew is maintained and no tree edges are crossed (the Euclidean embedding is assumed). Two rules are applied: (i) Min-Rule: a new sink is always connected to the balance point which requires the least wirelength added to the tree; and (ii) Max-Rule: the new sink added to the tree is the one which has the greatest distance to its closest balance point. An elegant analysis shows that this method always yields a planar-embeddable zero-skew solution with minimum possible source-sink pathlength. However, if we consider the simple case of four sinks at the corners of the unit square (with the source at the center), the method of [15] will create an "X" clock tree with cost = 4, while the optimal "H" solution has cost = 3.

3 The Planar-DME Algorithm

Before describing our new planar clock routing algorithm, we need a little more terminology. For any sink subset $S' \subseteq S$, we define the *diameter* of S' as $diameter(S') = \max\{d(s_i, s_j) | s_i, s_j \in S'\}$. The *radius* of S' is given by $radius(S') = diameter(S')/2$. We also define $center(S')$ to be the merging segment of v, where v is the root of the tree of merging segments constructed by DME over S'. We use $c(S')$ to denote the midpoint of $center(S')$. (Fact 1 below shows that the distance from $center(S')$ to any sink in S' is at most $radius(S')$, hence the term "$center(S')$".) A *Manhattan disk* is a TRR with a core consisting of a single point; we use $MD(s_i, r)$ to denote the Manhattan disk with core $\{s_i\}$ and radius $r \geq 0$. In other words, a Manhattan disk is the "diamond-shaped" set of all points within a prescribed radius of a central point.

Finally, we use two terms that are defined in the Euclidean plane: (i) $P_{S'}$ denotes any *convex polygon* containing S', and (ii) *convex-hull*(S') is the $P_{S'}$ with minimum area. We say that a point p lies *inside* $P_{S'}$ if p is on the boundary of $P_{S'}$ or is strictly interior to $P_{S'}$. These terms will be used in proving that the Planar-DME algorithm yields a planar solution: wires which are embedded along the boundary between two disjoint convex polygons cannot intersect subsequent wires that are embedded internally to these polygons.

3.1 Single-Pass DME

Our first theoretical result is that under the linear delay model, a single top-down phase can yield the same output as the two-phase DME algorithm. Essentially, we prove that the tree of merging segments constructed in the bottom-up DME phase can be generated *during* the top-down phase. This result follows from properties of the minimum-pathlength zero-skew subtree over any sink set S', notably that the root of the subtree over S' must be located at $center(S')$.

The following Facts and Theorem are crucial to the development of the Single-Pass DME and then Planar-DME algorithms. Proofs for the three Facts can be found in [12].

Fact 1: For any sink set S and topology G, let S_v be the set of sinks in the subtree rooted at v in the DME solution. Let $t_{LD}(u)$ be the linear delay (i.e., pathlength) from point $u \in ms(v)$ to each sink in S_v. Then $t_{LD}(u) = radius(S_v)$. □

Fact 2: Let G be the connection topology of the ZST $T(S)$ that is produced by DME. Let $d = diameter(S)$ and let $TRR(v)$ denote the special tilted rectangular region that corresponds to either $TRR(v) = MD(pl(v), d/2)$ if v is a sink node, or $TRR(v) = TRR(a) \cap TRR(b)$ if v is an internal node of G with children a and b. Then for each node $v \in G$, $ms(v) = core(TRR(v))$. □

Fact 3: For any sink subset $S' \subseteq S$, $diameter(S')$ can be computed in linear time. □

Theorem 1: Given a set of n sinks $S \in \Re^2$ and a connection topology G, we can produce the same output ZST $T(S)$ that the DME algorithm will produce under the linear delay model, using only a single top-down phase with time complexity between $\Omega(n \log n)$ and $O(n^2)$. $\square$

Proof: We determine the merging segments and incident edge lengths for all nodes in top-down order as follows. Let v be a node in G with parent p (if v is not the root). Also, let S_v and S_p respectively denote the sets of sinks in the subtrees rooted at v and p in the DME solution. From Fact 2, we have that $ms(v) = core(TRR(v)) = core(\bigcap_{u \in S_v} TRR(u))$. By Fact 1, the length of the edge incident to node v in G, $|e_v|$, is equal to $t_{LD}(p) - t_{LD}(v) = radius(S_p) - radius(S_v)$. Now we have all the information that would have been provided by the bottom-up phase of DME; this immediately yields Single-Pass DME. With regard to time complexity, by Fact 3, we can compute $ms(v)$ and $|e_v|$ for node v in time $\Theta(|S_v|)$. If L_i denotes the set of the nodes at level i of G, and l is the height of G, then $\sum_{v \in L_i} S_v \leq n$, and $\log n \leq l \leq n$, implying overall time complexity is $l \times \sum_{v \in L_i} S_v \leq ln = O(ln)$. This is between $\Omega(n \log n)$ and $O(n^2)$. $\square$

Because Single-Pass DME results in the same optimum ZST that the original DME would achieve, established properties of the output tree (minimum source-sink pathlength, and minimum total tree cost with respect to the generated connection topology) are maintained. The worst-case and best-case time bounds are the same as those for the method of [15].

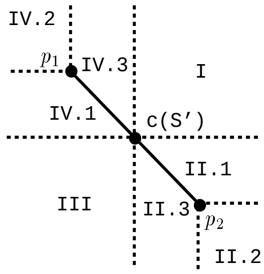
3.2 Planar-DME

The impact of Theorem 1 may not be immediately apparent, since DME can already accomplish the same construction in linear time. However, the proof showed that as soon as Single-Pass DME has been given a partitioning of S_v into S_a and S_b, it can immediately find the $ms(a)$ and $ms(b)$ that are compatible with an optimal ZST having this "top part" of the clock topology. Thus, Single-Pass DME allows the connection topology to be determined dynamically in a top-down fashion, yet still finds a minimum-pathlength, minimum-cost embedding of whatever topology is eventually determined.

The Planar-DME algorithm is essentially a version of Single-Pass DME wherein the connection topology is determined based on the existing routing, such that future routing cannot interfere with this existing routing. We use the (Euclidean) *convex polygon* concept to guide the top-down partitioning of both the routing area and the set of sinks: given $S' \subseteq S$ and a convex polygon $P_{S'}$ (i.e., containing S'), we recursively divide $P_{S'}$ into two smaller convex polygons such that routing inside each convex polygon cannot interfere with routing that is either inside the other convex polygon or on the boundary between the polygons.

Embedding and Partitioning Rules

The Planar-DME algorithm is derived from Single-Pass DME by adopting the following rules for embedding the internal nodes of the ZST, and for top-down bipartitioning of the sinks in each subtree.



The Embedding Rules

	Location of p	Embedding point for node v
A1	Regions I, III	$c(S')$
A2	Regions II.1, IV.1	intersection of $\overline{p_1p_2}$ with horizontal line through p
A3	Regions II.3, IV.3	intersection of $\overline{p_1p_2}$ with vertical line through p
A4	Region II.2 (IV.2)	p_2 (p_1)

The Partitioning Rules

	Locations of p, v	Splitting line
B1	$p \neq v$	$\overleftrightarrow{pv}$
B2	$p = v$; $p \neq c(S')$	$\overleftrightarrow{p_1p_2}$
B3	$p = v$; $p = c(S')$	Vertical line through p

Figure 3: Rules to choose the embedding point of v (the root of the subtree over sink set $S' \subseteq S$ in *any* minimum-radius ZST), and to choose the splitting line to partition the sink set S' based on the relative positions of v's parent p and $center(S') = \overline{p_1p_2}$. Let the coordinates of $c(S')$, p_1 and p_2 be (x_c, y_c), (x_1, y_1), and (x_2, y_2), respectively. We define: Region I: $x \geq x_c$, $y \geq y_c$; Region $II.1$: $y \geq -x + y_1 + x_1$, $y \leq y_c$, $y \geq y_2$; Region $II.2$: $x \geq x_2$, $y \leq y_2$; and Region $II.3$: $y \leq -x + y_1 + x_1$, $x \geq x_c$, $x \leq x_1$. Regions III, $IV.1$, $IV.2$, and $IV.3$ are defined similarly.

- **Embedding Rules**: In each recursive call, Planar-DME accepts a subset of sinks $S' \subseteq S$, some convex polygon $P_{S'}$ containing S', and some point p inside $P_{S'}$ which is to connect to a point v on $ms(v) = center(S')$. The point p is the embedding of the parent of the root of the subtree over S'; this point has already been determined earlier in the top-down pass[4]. The existing routing is outside $P_{S'}$, hence if we can select a feasible embedding point v inside $P_{S'}$, the routing from p to v will not interfere with the routing external to $P_{S'}$. Thus, the resulting routing will be planar. The embedding rules in Fig. 3 ensure that such an embedding point will be selected in $O(1)$ time.

- **Partitioning Rules**: Our goal is to find a *splitting line* which (i) divides $P_{S'}$ into two convex polygons $P_{S'_1}$ and $P_{S'_2}$ and thus also partitions the sink set between the two subtrees of v, and (ii) allows the routing from p to v to be on the boundary between $P_{S'_1}$ and $P_{S'_2}$. The rules to determine the splitting line for S' are shown in

[4]Note that when the meaning is clear, our discussion will use, say, v to denote either a node in the tree topology, or the point at which that node has been embedded in the Manhattan plane (that is to say, $pl(v)$).

Fig. 3. Sinks lying inside one of the convex polygons are assigned to that polygon to determine sets S'_1 and S'_2; a sink on $\overleftrightarrow{pv}$ can be assigned to either polygon, so long as neither S'_1 or S'_2 is empty. In the example of Fig. 2, $S' = \{a, b\}$ is divided into $S'_1 = \{a\}$ and $S'_2 = \{b\}$, and $P_{S'}$ is divided into $P_{S'_1}$ and $P_{S'_2}$ accordingly. A total of $\Theta(|S'|)$ time is needed to partition the sinks in set S'.

[12] shows that these embedding and partitioning rules satisfy the following:

Theorem 2: Given a subset $S' \subseteq S$, a convex polygon $P_{S'}$ containing S', and a point p inside $P_{S'}$, (i) the embedding rules will select a feasible embedding point v inside $P_{S'}$ (thus the routing from p to v will lie within $P_{S'}$), and (ii) the partitioning rules will divide $P_{S'}$ into two smaller convex polygons $P_{S'_1}$ and $P_{S'_2}$ that contain nonempty sink subsets S'_1 and S'_2, respectively, such that the routing from p to v is on the boundary between $P_{S'_1}$ and $P_{S'_2}$. $\square$

The overall Planar-DME algorithm is given in Fig. 4. Steps 4 and 6 in Planar-DME-Sub are the key difference between Planar-DME and generic Single-Pass DME. If the clock source location is not specified, then Planar-DME will set the root of the clock tree to be the clock location. Fig. 5 illustrates how the planar clock routing is achieved by Planar-DME. Finally, inductive application of Theorem 2 easily yields:

Theorem 3: The zero-skew clock routing tree constructed by Planar-DME is planar. $\square$

For any given sink set, the partitioning and embedding rules take the same (linear) time as is required to compute merging segments and edge lengths, hence Planar-DME has the same time complexity as Single-Pass DME.

4 Experimental Results

We implemented the Planar-DME algorithm on Sun SPARC IPC workstations in the C/UNIX environment. The same seven examples as in [9, 4, 15] were studied.

For the linear delay model, we compared Planar-DME against three other methods: the method of Zhu and Dai which guarantees a planar ZST; the KCR+DME method which gave the best results in [4]; and the Greedy-DME method of Edahiro [9], which gives the best-known wirelength results for zero-skew trees in the linear model.

Table 1 shows an average of 15.5% reduction in wirelength versus the previous method of [15]. Surprisingly, our planar solutions have lower cost than the *non-planar* solutions of KCR+DME for Primary1 and r5, and wirelengths are comparable to those of KCR+DME for the other test cases.

Finally, Fig. 6 shows the planar clock routing solutions constructed by Planar-DME and the algorithm of [15] for the Primary 1 benchmark.

Algorithm Planar-DME (S,clk)
Input: Set of sinks S; clock location clk in P_S.
Output: Planar ZST $T(S)$ with root s_0; $cost(T)$.

1. $r = radius(S)$
2. Build $TRR(u) = MD(u, r)$ for all sinks $u \in S$
3. $center(S) = core(\bigcap_{u \in S} TRR(u))$
4. **if** clk not specified
5. Embed s_0 at $c(S)$ (i.e., $pl(s_0) = c(S)$)
   else
6. Embed s_0 at clk (i.e., $pl(s_0) = clk$)
   endif
7. $|e_{s_0}| = d(pl(s_0), center(S))$
8. $t_{LD}(s_0) = |e_{s_0}| + r$
9. P_S = a rectangle containing S and clk
10. Planar-DME-Sub(S,P_S,s_0)
11. $cost(T) = \sum_{v \in T} |e_v|$

Procedure Planar-DME-Sub (S',$P_{S'}$,p)
Input: Set of sinks $S' \subseteq S$; convex polygon $P_{S'}$ containing S'; parent node p lying inside $P_{S'}$.
Output: Planar ZST $T(S')$ with root v .

1. $t_{LD}(v) = radius(S')$
2. $ms(v) = center(S') = core(\bigcap_{u \in S'} TRR(u))$
3. $|e_v| = t_{LD}(p) - t_{LD}(v)$
4. Embed node v at $pl(v) \in ms(v)$ by embedding rules in Fig. 3
5. Connect a wire from $pl(p)$ to $pl(v)$
6. Divide S' and $P_{S'}$ into S'_1, S'_2 and $P_{S'_1}$, $P_{S'_2}$ by partitioning rules in Fig. 3
7. $parent(v) = p$
8. **if** $|S'| = 1$ RETURN **endif**
9. Planar-DME-Sub(S'_1,$P_{S'_1}$,v)
10. Planar-DME-Sub(S'_2,$P_{S'_2}$,v)

Figure 4: The Planar-DME Algorithm.

5 Conclusions

We have extended the DME algorithm of [2, 3, 4, 8] to construct a guaranteed-planar exact zero-skew clock routing tree. In the linear delay model, the source-sink delay is the minimum possible, i.e., it is equal to the radius of the sink set, $radius(S)$. Moreover, the wirelength of our solution is the minimum possible with respect to the generated topology. Our planar solutions are competitive with the best previous *non-planar* solutions, and we obtain substantial cost reductions over the previous planar routing method of [15].

Current work is investigating several areas of improvement for Planar-DME. First, note that our method generates a routing tree which is Euclidean planar-embeddable; as noted above, this is not required for the routing to be Manhattan planar-embeddable. As described, Planar-DME uses very simple embedding and partitioning rules to guarantee planarity. Better partitioning strategies are possible, e.g., the splitting line can be changed to a splitting *path* of two or more line segments. Preliminary trials of such an approach are very promising [13]; improving the embedding rules seems harder and less promising than improving the partitioning rules.

A second major extension of Planar-DME involves

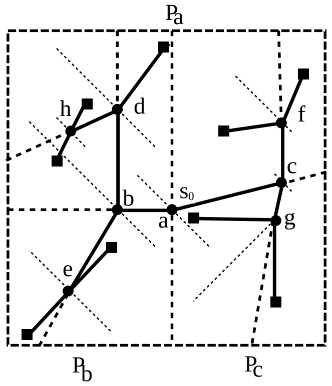


Figure 5: Example with 9 sinks (squares at leaf nodes in tree), illustrating the execution of Planar-DME. The entire routing region (P_a) is recursively divided into convex polygons (boundaries of polygons indicated by thick dashed lines and tree edges) until only one sink lies within each convex polygon. The tree of merging segments is given by thin dotted lines. The root of the clock tree, s_0, is embedded as specified by Planar-DME in Fig. 4. The internal nodes $a, b, \cdots, h$ are embedded by Rules $A1$, $A2$, $A1$, $A3$, $A1$, $A3$, $A4$, and $A1$. Note that $s_0 = a$. Polygon P_a is divided by the vertical line through a (Rule $B3$) into polygons P_b and P_c which contain sink sets rooted at nodes b and c. All other partitioning steps invoke Rule $B1$ only.

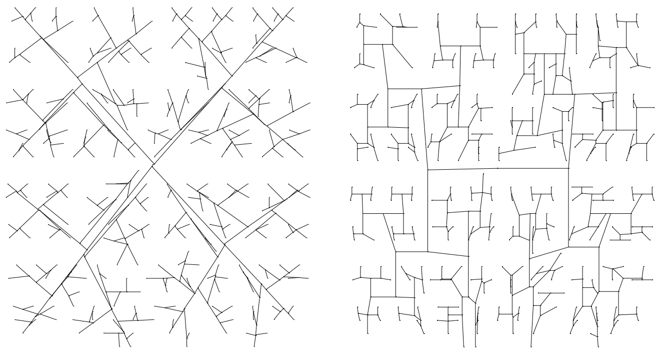
Benchmark (#sinks)	Greedy -DME	KCR +DME	Planar -DME	Zhu-Dai
P1(269)	131.1	140.3	136.0	167.9
P2(603)	309.4	350.4	353.7	422.5
r1(267)	1,288.4	1,497	1,511.8	1,778.3
r2(598)	2,554.5	3,013	3,363.5	3,580.1
r3(862)	3,277.4	3,902	3,943.9	4,635.9
r4(1,903)	6,588.5	7,782	7,835.7	9,577.1
r5(3,101)	9,806.8	11,665	11,491.1	14,119.4
Time	$O(n \log n)$	$O(n \log n)$	$O(n^2)$	$O(n^2)$
Planarity	NO	NO	YES	YES

Table 1: Comparison of Planar-DME with other algorithms for the linear delay model, using MCNC benchmarks Primary1 (P1) and Primary2 (P2), and benchmarks r1 through r5 from Tsay [14]. The clock source locations are not given. Average cost savings over [15] are 15.5%.

its applicability to more sophisticated delay models such as Elmore delay [13]. Our intuition is that top-down partitioning is more "global" than bottom-up methods: a top-down strategy can easily enforce planarity of the resulting solution, and is the logical candidate for extension into more sophisticated clock tree constructions. Finally, a more practical clock tree synthesis approach will use a bounded-skew, rather than an exact zero-skew, objective. The DME approach can be modified to use the concept of a merging area in the bounded-skew tree construction [10].

Acknowledgments

We thank Mr. Qing Zhu and Professor Wayne W.-M. Dai of UC Santa Cruz for providing preprints of their work. Charles J. Alpert, Lars Hagen, Kenneth D. Boese and Jenhsin Huang provided helpful comments on early drafts.



(a) Zhu-Dai [15] (cost=167.9) (b) Planar-DME (cost=136.0)

Figure 6: Planar zero-skew clock trees with minimum source-sink pathlength delay, for the Primary1 benchmark.

References

[1] H. Bakoglu, *Circuits, Interconnections and Packaging for VLSI* , Addison-Wesley, 1990.

[2] K. D. Boese and A. B. Kahng, "Zero-Skew Clock Routing Trees With Minimum Wirelength," *Proc. IEEE Intl. Conf. on ASIC*, 1992, pp. 1.1.1 - 1.1.5.

[3] T.-H. Chao, Y.-C. Hsu and J.-M. Ho, "Zero Skew Clock Net Routing," in *Proc. ACM/IEEE Design Automation Conf.*, 1992, pp. 518-523.

[4] T.-H. Chao, Y. C. Hsu, J. M. Ho, K. D. Boese and A. B. Kahng, "Zero Skew Clock Routing With Minimum Wirelength", *IEEE Trans. on Circuits and Systems* 39(11), November 1992, pp. 799-814.

[5] J. Cong, A. B. Kahng and G. Robins, "Matching-Based Methods for High-Performance Clock Routing", *IEEE Trans. on CAD* 12(8), August 1993, pp. 1157-1169.

[6] S. Dhar, M. A. Franklin and D. F. Wann, "Reduction of Clock Delays in VLSI Structures," *Proc. IEEE Intl. Conf. on Computer Design*, 1984, pp. 778-783.

[7] D. Dobberpuhl et al., "A 200MHz 64b Dual-Issue CMOS Microprocessor", *Proc. IEEE Intl. Solid State Circuits Conf.*, Feb. 1992, pp. 106-107.

[8] M. Edahiro, "Minimum Skew and Minimum Path Length Routing in VLSI Layout Design", *NEC Research and Development* 32(4), October 1991, pp. 569-575.

[9] M. Edahiro, "Minimum Path-Length Equi-Distant Routing", *IEEE Asia-Pacific Conference on Circuits and Systems*, December, 1992, pp. 41-46.

[10] J. H. Huang, A. B. Kahng and C.-W. A. Tsao, "On the Bounded-Skew Clock and Steiner Routing Problems", UCLA CS Dept. *technical report* TR-940026, 1994.

[11] M. A. B. Jackson, A. Srinivasan and E. S. Kuh, "Clock Routing for High Performance ICs," *Proc. ACM/IEEE Design Automation Conf.*, 1990, pp. 573-579.

[12] A. B. Kahng and C.-W. Albert Tsao. "Planar-DME: Improved Planar Zero-Skew Clock Routing With Minimum Pathlength Delay," *technical report UCLA CSD-940006*, February 1994.

[13] A. B. Kahng and C.-W. Albert Tsao. "Low-Cost Single-Layer Clock Trees With Exact Zero Elmore Delay Skew," *Proc. IEEE Intl. Conf. on Computer-Aided Design* (to appear), 1994.

[14] R. S. Tsay, "Exact Zero Skew", *Proc. IEEE Intl. Conference on Computer-Aided Design*, 1991, pp. 336-339.

[15] Q. Zhu and W. W.-M. Dai, "Perfect-Balance Planar Clock Routing With Minimal Path-Length", *Proc. IEEE Intl. Conf. on Computer-Aided Design*, Nov. 1992, pp. 473-476.