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

[17] 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.

References

[1] H. Bakoglu, J. T. Walker and J. D. Meindl, "A Symmetric Clock-Distribution Tree and Optimized High-Speed Interconnections for Reduced Clock Skew in ULSI and WSI Circuits", *Proc. IEEE Intl. Conf. on Computer Design*, 1986, pp. 118-122.

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

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

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

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

[10] M. Edahiro, "Clustering-Based Optimization Algorithm in Zero-Skew Routings", *Proc. ACM/IEEE Design Automation Conf.*, June 1993, pp. 612-616.

[11] W. C. Elmore, "The Transient Response of Damped Linear Networks With Particular Regard to Wide-Band Amplifiers," *Journal of Applied Physics* 19(1), Jan. 1948, pp. 55-63.

[12] J. P. Fishburn, "Clock Skew Optimization," *IEEE Transactions on Computers* 39(7), July 1990, pp. 945-951.

[13] A. L. Fisher and H. T. Kung, "Synchronizing Large Systolic Arrays", *Proceedings of SPIE* 341, May 1982, pp. 44-52.

[14] 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.

[15] A. B. Kahng, J. Cong, and G. Robins, "High-Performance Clock Routing Based on Recursive Geometric Matching," *Proc. ACM/IEEE Design Automation Conf.*, 1991, pp. 322-327.

Appendix: The Top-Down and Bottom-Up Phases in DME

The following two subroutine templates are from [5]; they reproduce the bottom-up and top-down phases of the DME algorithm as presented in [3, 5].

Procedure Build_Tree_of_Segments
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 $v \neq s_0$

```
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
```

Figure 9: Construction of the tree of segments. Details of the Calculate_Edge_Lengths subroutine depend on the specific delay model used (DME can be applied with any monotone delay function).

Procedure Find_Exact_Placements
Input: Tree of segments TS containing $ms(v)$ and $|e_v|$ 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
```

Figure 10: Construction of the ZST by embedding internal nodes of the topology.

better embedding points for nodes v in the topology; for instance, it seems that a good embedding point for v should not cause many sinks to be close to the $\overline{pv}$ line segment. We have implemented one version of the method suggested in the Footnote; it yields wirelength strictly better than KCR+DME for all cases, and achieves an average of 21.2% wirelength reduction over [17].

The second major extension of Planar-DME involves its applicability to more sophisticated delay models. While the previous works of [3, 5, 9, 17] also rely heavily on the linear delay model (indeed, Boese et al. [3, 5] show that DME does not produce optimal delay and wirelength according to the Elmore model), we nevertheless hope to find guaranteed-planar methods under the Elmore model. To this end, heuristics which partition the current sink set according to estimates of total sink and wiring capacitance may be effective (e.g., the "Balanced Bipartition" heuristic proposed in [4, 5]). Our intuition is that top-down partitioning is more "global" than bottom-up methods in its outlook: the top-down strategy can easily enforce planarity of the resulting solution, and is the logical candidate for extension into more sophisticated clock tree constructions.

Acknowledgments

We are grateful to Mr. Qing Zhu and Professor Wayne W.-M. Dai for providing preprints of their work. Charles J. Alpert, Lars Hagen, Kenneth D. Boese and Jenhsin Huang provided helpful comments on early drafts of this report.

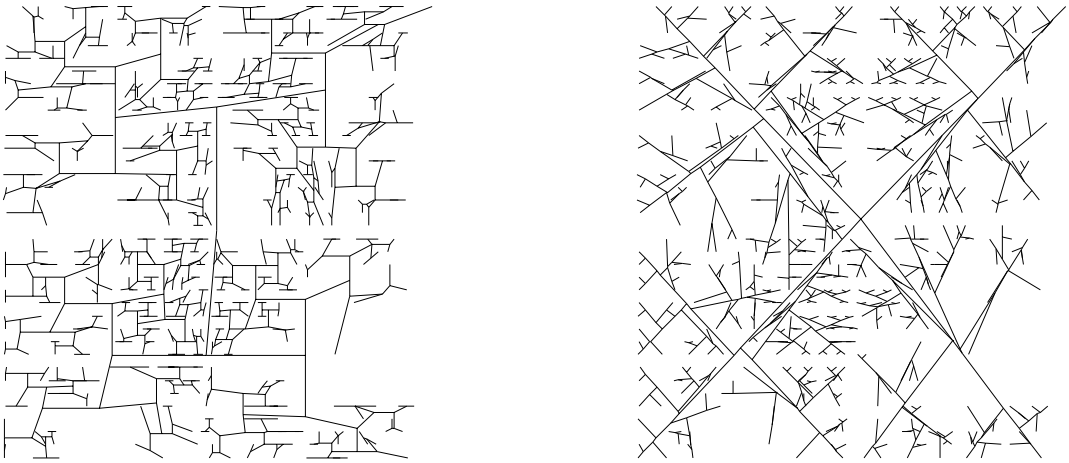

Figure 8: Planar zero-skew clock trees with minimum source-sink pathlength delay, for the Primary2 benchmark. Left: the solution produced by Planar-DME. Right: the solution produced by the algorithm of Zhu and Dai.

on the fact that at any level of the ZST topology, Single-Pass DME determines optimal merging segments based only on the *partition* of the sinks. Instead of executing Single-Pass DME on a prescribed topology, we therefore can *generate the topology dynamically* in top-down order, while still preserving all the useful properties of the DME solution. In particular, the Planar-DME algorithm invokes embedding and partitioning rules which guarantee the planarity of the resulting clock tree. We find that our planar solutions are competitive with the best previous *non-planar* solutions, and we obtain an average of 15.5% tree cost reduction over the previous planar routing method of [17] (note that all these previous works use the same linear delay model as we do).

Our current work is investigating several areas of improvement for Planar-DME. First, note that our method generates a routing tree which is planar in the Euclidean embedding; as noted above, this is not required for the routing to be Manhattan planar-embeddable. Currently, we use very simple embedding and partitioning rules (e.g., choosing vertical lines or extensions of $\overline{pv}$ as splitting lines) to guarantee planarity. Thus, several extensions explore better partitioning strategies, e.g., the splitting line can be changed to a splitting *path* of two line segments.[11] Alternatively, we may also search for

[11] In other words: Instead of drawing a straight line through points p and v, we can draw a line segment plus a ray to separate the polygons P_{S_1} and P_{S_2}. The line segment is $\overline{pv}$, and the ray emanates from v. Since we no longer have a straight splitting line, one of the new smaller regions, say S_1, will not be a convex polygon, so we would have to ensure that S_1 will be partitioned into convex polygons at the next step. We can exploit the added flexibility provided by this strategy. For instance, one criterion for routing this splitting path could be to choose the ray emanating from v so as to minimize the convex hull diameters of the resulting sink sets S_1 and S_2, since presumably this affects the wiring needed to construct ZST's over each of these sets.

method; we suspect that this is due to the very regular arrangement of sinks in the Primary1 layout. On the other hand, our results are worst for the r2 example, and we suspect that this is due to the highly irregular distribution of sinks in this test case. We emphasize that our current implementation uses a very simple polygon partitioning scheme, and that this can be easily modified to yield improved results for highly non-uniform sink locations (results for one such simple improvement are described in Section 5 below).

	#sinks	Greedy-DME	KCR+DME	Planar-DME	Zhu-Dai
P1	269	137.0	140.3	136.0	167.9
P2	594	311.4	350.4	353.7	422.5
r1	267	1,331.9	1,497	1,511.8	1,778.3
r2	598	2,590.8	3,013	3,363.5	3,580.1
r3	862	3,317.8	3,902	3,943.9	4,635.9
r4	1,903	6,780.2	7,782	7,835.7	9,577.1
r5	3,101	9,890.5	11,665	11,491.1	14,119.4
Complexity		$O(n \log n)$	$O(n \log n)$	$O(n^2)$	$O(n^2)$
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. Clock tree costs for KCR+DME are quoted from [5]. We execute Planar-DME without any prescribed clock source locations. Average cost savings over Zhu-Dai are 15.5%.

Finally, Figure 8 shows the planar clock routing solutions constructed by Planar-DME and the algorithm of [17].

5 Discussion and Conclusions

In this work, we have extended the DME algorithm of [3, 4, 5, 9] 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-DME method hinged on two key observations. First, in the linear delay model, it is possible to eliminate the bottom-up phase of the DME algorithm: for any sink set $S' \subseteq S$, the merging segment for the root of *any* minimum-delay ZST $T(S')$ is always equal to $center(S')$. Thus, we obtain the Single-Pass DME algorithm, which requires at most $O(|S|^2)$ runtime (note that if the topology is balanced, Single-Pass DME requires only $O(|S| \log |S|)$ time). The second key observation hinges

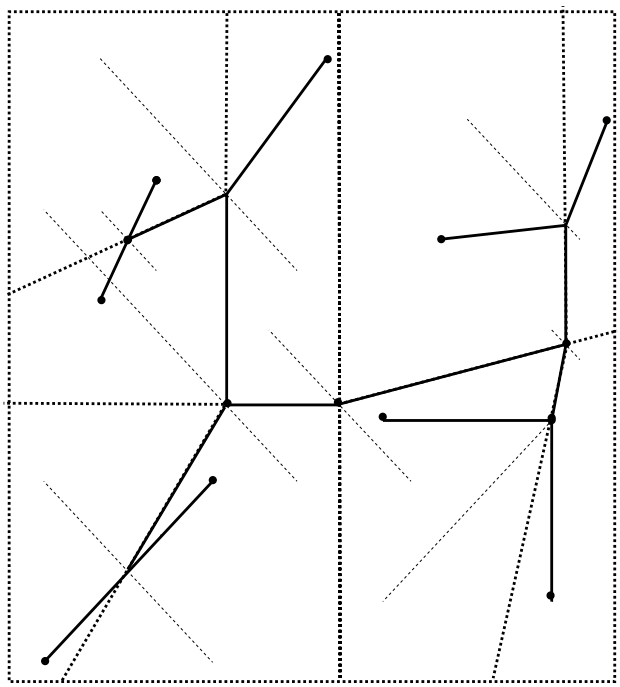

Figure 7: Example with 9 sinks (circular dots at leaf nodes in tree), illustrating the execution of Planar-DME. The overall routing region is recursively divided into convex polygons (boundaries of polygons indicated by thick dotted lines) until only one sink lies within each convex polygon. Also shown is the tree of merging segments (thin dashed lines), by which one can trace how the embedding rules determined the embedding points on the merging segments. The application of the partitioning rules described above is also apparent from the diagram.

For the linear delay model, we compared Planar-DME against three other methods: the method of Zhu and Dai; the KCR+DME method which gave the best results in [5]; and the Greedy-DME method of Edahiro [10], which gives the best-known wirelength results for zero-skew trees. The method of [17] is planar; the KCR+DME method yields a height-balanced tree topology; and Greedy-DME can yield a height-unbalanced solution.

Our results are shown in Table 1. As we expect, our method provides significant improvements for guaranteed-planar clock routing: we obtain an average of 15.5% reduction in wirelength versus the previous method of [17]. Surprisingly, our planar solutions have lower cost than the *non-planar* solutions of KCR+DME for Primary1 and r5, and indeed Planar-DME has wirelength very comparable to that of KCR+DME for the other test cases. For the Primary1 test case, our Planar-DME method even surpasses the best (non-planar) zero-skew clock routing method known, i.e., the Greedy-DME

Algorithm Planar-DME (S,P_S,clk)
Input: Set of sinks S; convex polygon P_S containing S; clock location clk in P_S.
Output: Planar ZST $T(S)$ with root s_0; $Cost(T(S))$.
Step 1. $r = radius(S)$;
Step 2. Build $TRR(u) = MD(u, r)$ for all sinks $u \in S$;
Step 3. $center(S) = core(\bigcap_{u \in S} TRR(u))$;
Step 4. **if** clk not specified
Step 5. Embed s_0 at the mid-point of $center(S)$ ($pl(s_0) = c(S)$);
else
Step 6. Embed s_0 at clk ($pl(s_0) = clk$);
endif
Step 7. $\|e_{s_0}\| = d(pl(s_0), center(S))$;
Step 8. $t_{LD}(s_0) = \|e_{s_0}\| + r$;
Step 9. Planar-DME-Sub(S,P_S,s_0) ;
Step 10. $cost(T) = \sum_{v \in T} \|e_v\|$;

Procedure Planar-DME-Sub (S',$P_{S'}$,p)
Input: Set of sinks $S' \subseteq S$; convex polygon $P_{S'}$ containing S'; parent node p in $P_{S'}$.
Output: Planar ZST $T(S')$ with root v .
Step 1. $t_{LD}(v) = radius(S')$;
Step 2. $ms(v) = center(S') = core(\bigcap_{u \in S'} TRR(u))$;
Step 3. $\|e_v\| = t_{LD}(p) - t_{LD}(v)$;
Step 4. Use the embedding rules to embed node v at $pl(v) \in ms(v)$;
Step 5. Connect a wire from $pl(p)$ to $pl(v)$;
Step 6. Use the partitioning rules to divide S' and $P_{S'}$ into S'_1 and S'_2, and $P_{S'_1}$ and $P_{S'_2}$;
Step 7. $parent(v) = p$;
Step 8. **if** $\|S'\| == 1$ RETURN; **endif**
Step 9. Planar-DME-Sub(S'_1,$P_{S'_1}$,v) ;
Step 10. Planar-DME-Sub(S'_2,$P_{S'_2}$,v) ;

Figure 6: The Planar-DME Algorithm.

4 Experimental Results

We implemented the Planar-DME algorithm, using the simple polygon partitioning scheme described above, on Sun SPARC IPC workstations in the C/UNIX environment. Two sets of benchmarks were used: (i) the sink placements for the MCNC Primary1 and Primary2 benchmarks used in [14] and [15], and originally provided by the authors of [14]; and (ii) the sink placements for the five benchmarks r1 - r5 used in [16]. These seven examples have also been studied by Edahiro [10] and by Zhu and Dai [17].

- $v = p = c(S')$: The vertical line through v separates Regions I and III and hence separates $\overline{AB}$ and $\overline{CD}$.
- $v = p$, $p \neq c(S')$: The extension of $center(S)$ again separates $\overline{AB}$ and $\overline{CD}$.

□

Details of the Planar-DME algorithm are given in Figure 6. If the location of clocking source is not prescribed, then Planar-DME will set the root of the clock tree as the clock location.

We point out that Steps 4 and 6 in Planar-DME-Sub are the crux of the difference between Planar-DME and the generic Single-Pass DME. (Single-Pass DME would more or less arbitrarily choose a feasible embedding point at Step 4, and partition the sinks in the subtree according to the prescribed connection topology at Step 6. In contrast, Planar-DME chooses both the embedding and the partition of the sinks (thus dynamically determining the topology) so that planarity is maintained.) The example in Fig. 7 illustrates how the planar clock routing is achieved by the Planar-DME algorithm.[10]

Theorem 3: The clock routing tree constructed by Planar-DME is planar.

Proof: Initially, there is a convex polygon P_S which contains the set of sinks S and a clock location clk; the clock location clk is the embeddding point of the parent node p of v, where v is the root of the minimum-pathlength delay ZST $T(S)$. The embedding rules guarantee that we can find embedding point $pl(v)$ within P_S so that the routing from $pl(p)$ to $pl(v)$ lies within P_S. The partition rules guarantee that we can partition P into two smaller convex polygons P_{S_1} and P_{S_2} that contain non-empty sink subsets S_1 and S_2, respectively, such that the routing from $pl(p)$ to $pl(v)$ is on the boundary between P_{S_1} and P_{S_2}. Node v will become the parent node p for the ZSTs $T(S_1)$ and $T(S_2)$, and is contained in both P_{S_1} and P_{S_2}. Inductively, all future routing over S_1 and S_2 will be confined within P_{S_1} and P_{S_2}, respectively. We conclude that no routing crosses any other. □

Note that Planar-DME terminates with between $\log|S|$ and $|S|$ levels of recursive calls to Planar-DME-Sub, since the maximum size of any sink subset decreases by at least one between consecutive levels in the recursion.

[10] An interesting sidelight is that the early MMM algorithm of Jackson et al. [14] seems to be another candidate for planar routing, although some modification is necessary. Recall that the MMM algorithm connects *centroids* (i.e., centers of mass) of sink sets in top-down fashion; in contrast, Planar-DME connects *geometric centers* of sink sets. However, the MMM algorithm does not guarantee any bounds on skew or source-sink pathlength delay.

and $P_{S'_2}$ accordingly. We still need to consider the case where $p = v$: if $p \neq c(S')$ then the line segment *center*(S') is extended to form the splitting line, otherwise we arbitrarily choose the vertical line through p as the splitting line. We emphasize that this is only one extremely simple choice of embedding and partitioning rules which guarantee a planar result for Single-Phase DME. As noted in the Conclusions, even slightly more sophisticated rules are likely to improve the total tree cost of the planar clock trees that we produce.

We now prove that these embedding and partitioning rules guarantee that the routing for a subset of sinks $S' \subseteq S$ lies within any given $P_{S'}$. Our discussion again refers to Figure 5.

Theorem 2: Given a subset $S' \subseteq S$, a convex polygon $P_{S'}$, and a point p inside $P_{S'}$, the embedding rules will select a feasible embedding point v inside $P_{S'}$ and the partitioning rules will divide S' into two nonempty subsets.

Proof: First, we can see that for all the embedding cases v lies on the portion of *center*(S') that is closest to p; thus, v is a feasible embedding point. Second, we show that v always lies inside $P_{S'}$ as follows. Recall that $\overline{AB}$ and $\overline{CD}$ must each contain a point of S'. Let X and Y be these two points on $\overline{AB}$ and $\overline{CD}$ respectively. If point p lies inside Region I or Region III, then by Fact 4 $v = c(S')$ is inside *convex-hull*$(S') \subseteq P_{S'}$, whence v lies inside $P_{S'}$. If point p is in Region II or Region IV, it is easy to see from the Figure that v lies inside *convex-hull*$(p, c(S'), X, Y) \subseteq P_{S'}$. Third, we assume that $p \neq v$ and then show that for each case for the location of p, $\overleftrightarrow{pv}$ divides S' into two non-empty subsets.

- Regions I, III: Since $v = c(S')$, by Fact 4 we have that v is inside *convex-hull*(S'), whence $\overleftrightarrow{pv}$ divides *convex-hull*(S') into two semi-planes each containing at least one point from S'.
- Regions $II.1, IV.1$: The horizontal line $\overleftrightarrow{pv}$ separates $\overline{AB}$ from $\overline{CD}$.
- Regions $II.3, IV.3$: The vertical line $\overleftrightarrow{pv}$ separates $\overline{AB}$ from $\overline{CD}$.
- Regions $II.2, IV.2$: The line $\overleftrightarrow{pv}$ separates $\overline{AB}$ from $\overline{CD}$.

Last, we show that the splitting line selected by the partitioning rules also works for the case $p = v$.

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.

First, let us look at the **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.[9] The existing routing is outside $P_{S'}$, so if we can select a feasible embedding point v inside $P_{S'}$, then the routing from p to v will not interfere with the routing external to $P_{S'}$. Thus, the resulting routing will be planar.

We use the same notation and assumptions from the proof of Fact 4, along with Figure 5, in the following. The location of point p inside a given region of $P_{S'}$ completely determines the embedding of v, according to the following cases:

- Region I, III: $v = c(S')$.
- Region $II.1, IV.1$: v is the point of intersection of $center(S')$ with the horizontal line through point p.
- Region $II.3, IV.3$: v is the point of intersection of $center(S')$ with the vertical line through point p.
- Region $II.2$: $v = p_2$.
- Region $IV.2$: $v = p_1$.

The **partitioning rules** for S' are also straightforward, except that we must consider several degenerate cases. Our goal is to find a *splitting line* which divides the $P_{S'}$ into two convex polygons and thus also partitions the sink set between the two subtrees of v. If $p \neq v$ we extend the line segment $\overline{pv}$ to be a splitting line $\overleftrightarrow{pv}$, which divides $P_{S'}$ into two convex polygons $P_{S'_1}$ and $P_{S'_2}$. 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 of the polygons, so long as neither S'_1 or S'_2 is empty. For example, in Fig. 3, $S' = \{a, b\}$, is divided into $S'_1 = \{a\}$ and $S'_2 = \{b\}$, and $P_{S'}$ is divided into $P_{S'_1}$

[9] 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, $pl(v)$).

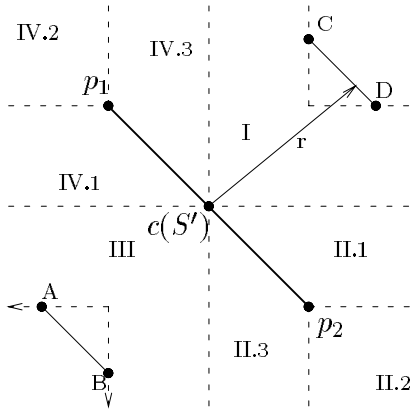

Figure 5: Rules to choose the embedding point of v (the root of the subtree over sink set $S' \subseteq S$ in *any* minimum-radius ZST), and to choose the the splitting line to partition the sink set S' based on the relative positions of v's parent p and $center(S')$. If we denote the coordinates of $c(S')$ by (x_c, y_c), then the regions are defined by the following inequalities: 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.

complexity as Single-Pass DME, i.e., $O(|S|^2)$. In order to prove that our algorithm divides any $P_{S'}$ appropriately, we require the following Fact 4, which states that for any sink set $S' \subseteq S$, the midpoint of $center(S')$, $c(S')$, must lie inside *convex-hull*(S'). Note that $center(S')$ does not necessarily lie entirely in *convex-hull*(S') – consider the case of S' containing two points along a diagonal line.

Fact 4: For any $S' \subseteq S$, $c(S')$ lies inside *convex-hull*(S').

Proof: Without loss of generality, assume that the Manhattan Arc $center(S')$ has slope -1. Let $p_1 = (x_1, y_1)$ and $p_2 = (x_2, y_2)$ be the two endpoints of $center(S')$, with $x_1 < x_2$ and $y_2 < y_1$ (see Figure 5). Let $radius(S') = r$ and let $F = \{s \in S' \mid d(s, c(S')) = r\}$ denote the set of sinks that are furthest from $c(S')$. Since $center(S')$ lies in the intersection of all Manhattan disks of radius r centered at sinks in S, we know that $F \subset \overline{AB} \cup \overline{CD}$, where $A = (x_1 - d, y_2), B = (x_1, y_2 - d), C = (x_2, y_1 + d)$, and $D = (x_2 + d, y_1)$ where $d = r - (x_2 - x_1) = r - (y_1 - y_2) \geq 0$. In other words, all points of F lie on the segments $\overline{AB}$ and $\overline{CD}$. Either $B \in F$ or $D \in F$ (or both), otherwise $center(S)$ could be extended into $IV.2$ in the Figure, and then p_1 would not be as specified. Similarly, either $A \in F$ or $C \in F$. We see that:

(i) If $A, D \in F$ ($B, C \in F$), then $\overline{AD}$ ($\overline{CD}$) contains $c(S')$ and the fact is proven.

(ii) If $A, B \in F$ ($C, D \in F$), some point $E \in F$ must exist on segment $\overline{CD}$ ($\overline{AB}$) and $c(S)$ lies within the convex hull of points A, B (C, D) and E. □

ZST over S_v that has minimum source-sink pathlength delay. This immediately yields the "Single-Pass DME" method. Because Single-Pass DME results in the same optimum ZST that the original DME would achieve, established properties of the output tree (i.e., minimum source-sink pathlength and minimum total tree cost with respect to the generated connection topology) are maintained. We note that the same worst-case and best-case complexity bounds in Theorem 1 are the same as those for the method of Zhu and Dai [17].

3.2 The Planar-DME Algorithm

The impact of Theorem 1 above may not be immediately apparent, since DME can already accomplish the same construction in linear time. However, the reader should note that the proof showed how to determine $ms(v)$ without any *explicit* knowledge of the subtree topology rooted at v: i.e., we required only the partitioning of sinks between the two subtrees rooted at v and at v's sibling in G. The linear delay model then allowed us to find $ms(v)$ solely from these *lists* of sinks in S_v. It is this observation which is exploited to yield our Planar-DME algorithm.

The crucial observation is 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.

Our work hinges on the fact that if Single-Pass DME chooses the connection topology and embeds it carefully, then a *planar* routing can be achieved. Our Planar-DME algorithm is essentially a version of Single-Pass DME, with the top-down selection of node embeddings and connection topology guided by rules that guarantee the planarity of the routing. In Planar-DME, the connection topology is determined based on the existing routing, such that the 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. This method achieves a planar routing solution with the same worst-case time

Fact 3: For a sink subset $S' \subseteq S$ in the Manhattan plane, $diameter(S')$ can be computed in linear time.

Proof: [3] remarks that it is well known that under a 45 degree rotation of the coordinate axes (and scaling by a $\sqrt{2}$ factor), Manhattan distance is transformed into L_∞ distance (i.e., $d((x,y),(x',y')) = \max(|x-x'|,|y-y'|)$. Such a rotation can be accomplished in constant time per point in S'. Let $x_{max}(y_{max})$ and $x_{min}(y_{min})$ denote respectively the largest and smallest x-coordinates (y-coordinates) among all the points in the rotated S'. Then $diameter(S') = \max(x_{max} - x_{min}, y_{max} - y_{min})$ and is easily computed in $O(|S'|)$ time. □

Theorem 1: Given a set of 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 $O(|S|^2)$.

Proof: We will show that for any node v in G, $ms(v)$ can be found in time linear in the total number of descendants of v. Let the terms d and $TRR(v)$ be defined as in the statement of Fact 2. The value of d can be found in $O(|S|)$ time (Fact 3), and in $O(|S|)$ time we can build $TRR(s)$ for all sinks s (leaf nodes in G). According to Fact 2, we can infer that $ms(v) = core(TRR(v)) = core(\bigcap_{u \in S_v} TRR(u))$, where S_v is the set of descendants of node v in G. Since the intersection of any two TRR's can be found in constant time and is also a TRR, we can compute $TRR(v)$ (and its core) in time proportional to the number of v's descendants.[8]

If v is not the root of G, let p be its parent. By Fact 1, the length of the edge incident to node v in G, $|e_v|$, is equal to $t_{LD}(p) - t_{LD}(v) = radius(S_p) - radius(S_v)$, where S_v and S_p are the sets of descendants of node v and p, respectively. Thus, the overall effort to compute $ms(v)$ and $|e_v|$ is bounded by $O(|S|)$, and we now have the information that would have been provided by the bottom-up phase of DME. In the best case, the height of the tree of merging segments will be $O(\log|S|)$, so that the overall time complexity will be $O(|S|\log|S|)$. In the worst case, the height of the tree of merging segments will be $\Theta(|S|)$, implying overall time complexity of $\Theta(|S|^2)$. □

Theorem 1 shows that under the linear delay model, $ms(v)$ is *independent of the connection topology* over the sinks in S_v. Furthermore, $t_{LD}(v) = radius(S_v)$ implies that all the sinks in S_v are within distance $radius(S_v)$ of $center(S_v)$, i.e., $center(S_v)$ is the merging segment of the root of *any*

[8] Boese and Kahng [3] state this as "Lemma 2" in their paper.

use $c(S')$ to denote the midpoint of $center(S')$. (Note that Fact 1 below shows that the distance from $center(S')$ to any sink in S' is at most $radius(S')$, hence the name "$center(S')$".) 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 set of all points within a prescribed radius of a central point. (In the Manhattan plane, such a "disk" is diamond-shaped.)

Finally, we establish 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 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 to prove that top-down Planar-DME 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' (in particular, that the root of the subtree over S' must be located at the "center" of S').

Two useful facts are due to [3]. Fact 1 is a straightforward extension of Theorem 2 in [3], and Fact 2 is proved in the analysis of the same Theorem 2.

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 $t_{LD}(v)$ be the linear delay (i.e., pathlength) from v to each sink in S_v. Then $t_{LD}(v) = radius(S_v)$. □

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 $v \in G$, $core(TRR(v)) = ms(v)$ and $radius(TRR(v)) = d/2 - t_{LD}(v)$. □

shows that this method always (i) yields a planar-embeddable zero-skew solution, (ii) with minimum possible source-sink pathlength, (iii) with between $\Omega(n \log n)$ and $O(n^2)$ time complexity. However, if we consider the simple case of four sinks at the corners of the unit square (with the source at the center), we see that the method of [17] will adopt an "X" clock tree with cost = 4, while the optimal "H" solution has cost = 3. A larger 400-point example is shown in Figure 4: the solution of Zhu and Dai creates the "X-based" configuration which is clearly less desirable than the "H-based" solution. For 1600 sinks arranged in a 40-by-40 square array, an H-based solution will use 48% less wire than the solution of [17].

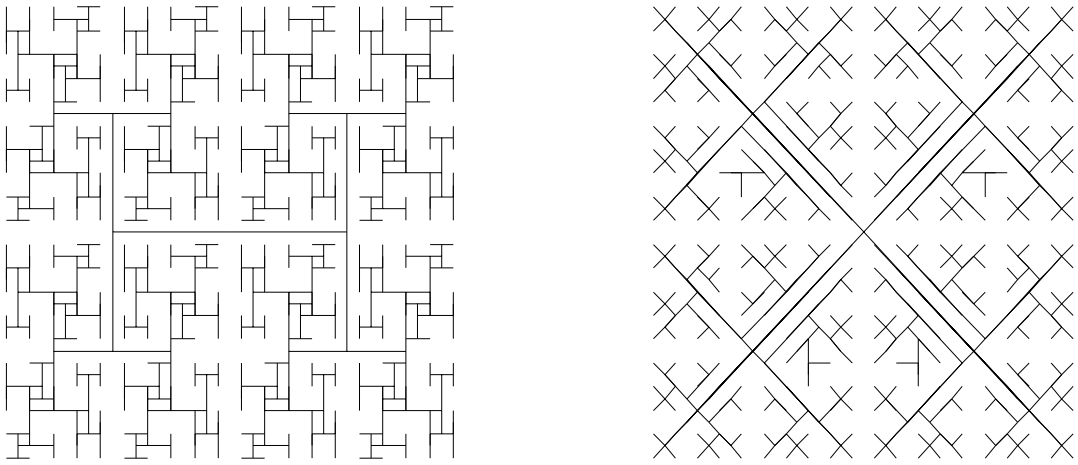

Figure 4: Example showing the difference between an H-tree like solution (left) and the solution of Zhu and Dai (right). Note that there are much overlapping routing in the right clock tree, so the actual wirelength of the right clock tree is much longer than it appears to. The solution at left is actually the output of the Planar-DME method that we describe below (on a larger array of 1600 sinks, Planar-DME saves 48% of the wirelength used by the Zhu-Dai algorithm). Note that for such regular array instances, both Greedy-DME (Edahiro) and the H-tree construction of Bakoglu will also return essentially the optimum solution.

3 Our New Algorithm: Planar-DME

Before describing our new planar clock routing algorithm, we establish some 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

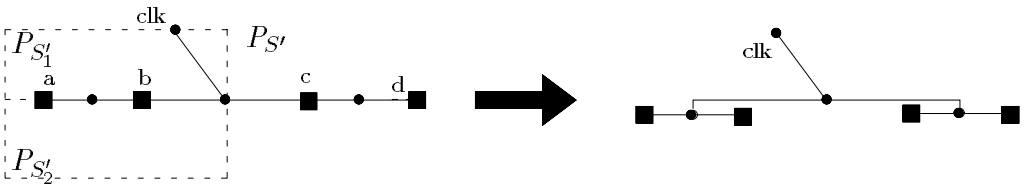

Figure 3: For certain inputs such as these four sinks on a line, overlapping of edges in an "optimal, planar" ZST is inevitable. We accept this since the ZST can be made into a non-overlapping solution with minimal increase in the wirelength. The convex polygon $P_{S'}$ and the labels $P_{S'_1}, P_{S'_2}, a$ and b pertain to the discussion of Section 3.2 below.

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; existing planar clock routing work [17] implicitly use the fact that planar-embeddability in the Euclidean plane is a sufficient condition for planar-embeddability in the Manhattan plane (since any line segment in the Euclidean plane can clearly be approximated to any desired accuracy by a monotone "staircase" in the Manhattan plane). Our development will adopt this perspective since it helps to motivate the Planar-DME algorithm below.

Second, we must 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 the possibility of a *degenerate* optimal planar clock routing solution where the embeddings of edges in the min-cost zero-skew tree are superposed. The example of Figure 3 shows this phenomenon: four sinks that are collinear will have an optimal "planar" clock tree whose edges pass over each other. Since this sort of overlapping can be made planar with minimum increase in wirelength, our convention is to accept such a degenerate solution as planar. This is also the convention of [17], whose algorithm will generate the solution shown in Figure 3.

The planar clock routing method of [17] is as follows. The method starts 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

Because DME requires an input topology, several works have focused on the choice of topology that yields a good routing solution when DME is applied. Two of the most successful DME-based methods are KCR+DME [3] and Greedy-DME [10]. Although these methods are non-planar, we use their results for comparison in Section 4 below. Kahng et al. [15] constructed a clock tree with near-zero average pathlength skew based on bottom-up recursive geometric matching; Boese et al. [3] then applied DME to this topology, with the combined KCR+DME algorithm producing average total wirelength reduction of 9% over the original KCR results. The best DME-based algorithm so far is the Greedy-DME approach of Edahiro [10], which determines the connection topology greedily in bottom-up order, such that each merging segment entails minimum increase in total wirelength.[7] Greedy-DME achieves nearly 17% wire length reduction over KCR+DME.

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, it is impossible to perform the actual layout without having to introduce many vias. This difficulty was first noted by Zhu and Dai, whose paper [17] stated several compelling reasons to seek a *planar-embeddable* clock routing solution. For instance:

- We may want to route the clock net on a prescribed layer, or more generally on the metal layer which has smallest RC delay.
- If one can avoid having different electrical parameters for different parts of the clock tree (e.g., the parts on distinct routing layers), the layout becomes more process variation-independent. Furthermore, uniform electrical parameters make it easier to apply buffering optimizations to reduce phase delay.
- The single-layer routing eliminates the delay and attenuation of clock signal through vias, so that both performance and signal integrity are improved.

Given these observations, our goal is now 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. Before stating the planar clock routing approach of Zhu and Dai, we make two observations. First, "planar-embeddable"

[7] This is the "CL" algorithm in [10].

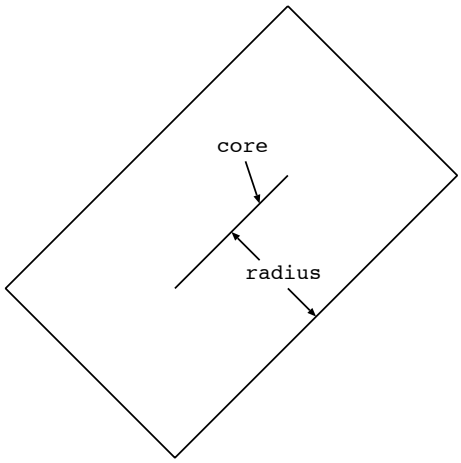

Figure 1: An example of a TRR.

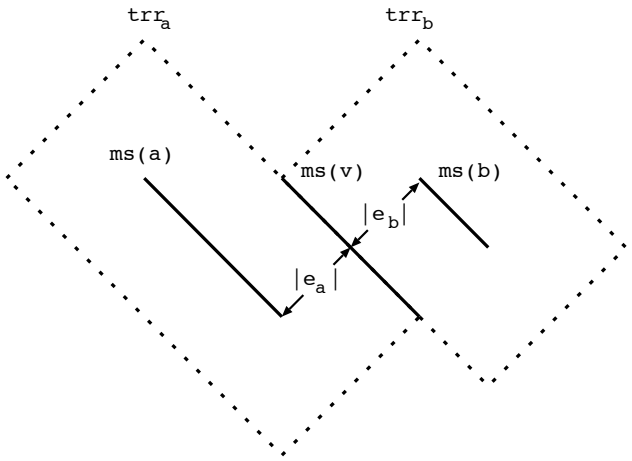

Figure 2: Construction of merging segment $ms(v)$.

cost, i.e., all points within distance $|e_a|$ of $ms(a)$ and within distance $|e_b|$ of $ms(b)$. If $ms(a)$ and $ms(b)$ are both Manhattan arcs, then $ms(v)$ 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., $ms(v) = trr_a \cap trr_b$. Figure 2 (reproduced from [3]) depicts an example of the construction of $ms(v)$. A lemma in [3] shows that if $ms(a)$ and $ms(b)$ are both Manhattan arcs, then $ms(v)$ is also a Manhattan arc. Since the merging segment $ms(s_i)$ for each sink s_i is a single point and thus a Manhattan arc, by induction all merging segments are Manhattan arcs. The pseudo-code for the bottom-up phase of DME is reproduced in the Appendix below.

Given the tree of merging segments, the **top-down phase** of DME chooses exact embeddings of internal nodes in the ZST, as follows. For node v in topology G, (i) if v is the root node, then DME selects any point in $ms(v)$ to be $pl(v)$;[5] or else (ii) if v is an internal node other than the root, DME chooses $pl(v)$ to be any point in $ms(v)$ that is at distance $|e_v|$ or less[6] from the placement of v's parent p (because the merging segment $ms(p)$ was constructed such that $d(ms(v), ms(p)) \leq |e_v|$, there must exist some choice of $pl(v)$ satisfying this condition). In case (ii), the algorithm first creates a square TRR trr_p with radius $|e_v|$ and core equal to $\{pl(p)\}$. Then, $pl(v)$ can be any point from $ms(v) \cap trr_p$. The pseudo-code for this top-down phase is also reproduced in the Appendix below; additional terminology and properties of DME will be developed in Section 3 as needed.

[5] If a fixed 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 $pl(s_0)$.

[6] If $d(pl(p), pl(v)) \leq |e_v|$, then detour wiring is used.

optimally embeds any given topology in the Manhattan plane, using minimum total wirelength. Because the properties of the DME construction are central to our present work, the remainder of this subsection and the Appendix provide a detailed review of the method, following the development in [3].

The DME Algorithm

Given a set of sinks S and a topology G, the DME algorithm 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. In our discussion, the *distance* between two points p and q is the Manhattan distance $d(p,q)$, and the distance between two sets of points P and Q, $d(P,Q)$, is $min\{d(p,q) \mid p \in P \text{ and } q \in Q\}$.

In the **bottom-up phase**, each node $v \in G$ is associated with a merging segment which represents a set of possible placements of v. The merging segment of a node depends on the merging segments of its two children, hence the bottom-up processing order. More precisely, let a and b be the children of node v in G, and let TS_a and TS_b denote the subtrees of merging segments rooted at a and b, respectively. 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 delays from $pl(v)$. The values of $|e_a|$ and $|e_b|$ which achieve this are unique; they are computed and retained for use in the top-down embedding phase of DME.

In the following, we require the following terminology. A *Manhattan arc* is defined to be a line segment, possibly of zero length, with slope +1 or -1; in other words, a Manhattan arc is a line segment tilted at 45 degrees from the wiring directions. The collection of points within a fixed distance of a Manhattan arc is called a *tilted rectangular region*, or *TRR*, whose boundary is composed of Manhattan arcs. The Manhattan arc at the center of the TRR is called its *core*. Finally, the *radius* of a TRR is the distance between its core and its boundary.

We can now give a formal recursive definition of $ms(v)$, the merging segment of node $v \in G$. If v is a sink s_i, then $ms(v) = \{s_i\}$ (note that this single point is a Manhattan arc). 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

its output topology. Section 3 also establishes the correctness and worst-case time complexity of Planar-DME. Section 4 gives experimental results comparing Planar-DME with the algorithm of Zhu and Dai, as well as with the best *non-planar* methods known, i.e., the Greedy-DME method of [10] and the KCR+DME method of [3, 5]. We conclude in Section 5 by listing several extensions and directions for future work.

2 Previous Work

2.1 Minimum-Cost Zero-Skew Trees

Minimum-cost, exact zero-skew clock trees for large cell-based designs have been addressed in a series of works by Kuh et al., Tsay, Boese et al., and others – primarily via geometric optimizations applied to the set of sink locations. The first clock tree construction for cell-based layouts with arbitrary sink locations was that of Jackson, Srinivasan and Kuh [14]; their "method of means and medians" (MMM) algorithm uses recursive top-down partitioning of the set of sinks into two equal-sized subsets, always connecting the centroid of a set to the centroids of its subsets.[3] Kahng et al. [15] [6] constructed clock tree topologies using a bottom-up matching approach, along with "H-flipping" and other heuristics. In practice, their "KCR" algorithm obtains zero pathlength skew (i.e., zero skew under the linear delay model) but has no theoretical guarantee. The work of Tsay [16] was the first to guarantee exact zero skew for any input; this was accomplished with respect to the Elmore delay model. In the same spirit as [15], Tsay recursively combines pairs of zero skew trees at "tapping points" to yield larger zero skew trees. To maintain the exact zero skew, wires are elongated as necessary.[4]

The methods of Jackson et al. [14], Kahng et al. [15], and Tsay [16] 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 Deferred-Merge Embedding (DME) method, which was discovered independently by three groups [3, 4, 9] of whom the earliest was M. Edahiro, is the first algorithm to consider the generation *and the embedding* of the topology together: the result is a linear-time algorithm which

[3]The pathlength skew of the MMM tree can be quite large even for a four-point example [15]; Boese et al. [3] ascribe this to "an inherent weakness" in the top-down approach, namely, that it can commit to a bad topology early in the construction.

[4]In parallel with this history, it is important to realize that for *regularly-spaced arrays* of clock sinks (e.g., in systolic array layouts), H-tree constructions are known to be optimal and have been extensively used throughout the literature [1, 7]. For regular arrangements of the clock sinks, the H-tree is often significantly superior to the other constructions. Zhu and Dai [17] have noted that the MMM and KCR methods, because they yield height-balanced binary tree topologies, can be viewed as "generalized H-trees".

Overview of Contribution

This report presents a new method which falls naturally at the juncture of two leading directions in the recent research on clock routing: (i) the DME (Deferred-Merge Embedding) algorithm of [3, 4, 5, 9] which embeds a given topology with guaranteed exact zero pathlength skew and minimum cost; and (ii) the method of [17], which is the first to guarantee a *planar-embeddable* clock routing solution. Our work exploits geometric observations to 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 which yields exactly the same solution. In other words, we show that for any given topology, we can find the DME solution using a single top-down pass. Further, we show that while the DME algorithm 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 (i.e., with minimum cost and minimum source-sink delay) at the same time.

Based on these observations, we 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. Beyond being planar, the resulting tree has provably minimum total wirelength and minimum pathlength delay for its topology (this is a consequence of the single top-down pass achieving the same effect as DME). We have tested a very simple version of our method, and found that it produces planar solutions with total wirelengths that are competitive with the best *non-planar* exact zero-skew results in the literature [10, 5]. We also obtain an average of 15.5% wirelength savings over the previous planar routing algorithm of [17]. A simple improvement of our scheme, outlined in Section 5 below, achieves an average of 21.2% wirelength savings over [17] for the same suite of test cases.

The remainder of this report is organized as follows. Section 2 reviews previous approaches to minimum-skew clock routing, with particular attention to the details of the DME algorithm [3, 4, 5] and the planar zero-skew algorithm of Zhu and Dai [17]. A simple example illustrates that the algorithm of [17], which was the first to guarantee planar-embeddability, leaves "room for improvement" in terms of total clock tree cost. Section 3 develops our main theoretical results. We first prove that our Single-Pass DME algorithm is equivalent to the original (Two-Pass) DME algorithm; we then develop our Planar-DME algorithm, which yields planar output and has optimal wirelength for

internal nodes of degree greater than three if the embedding is such that some edges of the binary tree have zero length.) The root of the clock tree is the *source*, denoted by s_0. When the clock tree is rooted at the source, any edge between a parent node v and its child w may be identified with the child node, i.e., 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, i.e., $|e_w| \geq d(pl(v), pl(w))$. 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 on the unique path from the source s_0 to the 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 sink pairs $s_i, s_j \in S$. If the skew of $T(S)$ is zero then $T(S)$ is a *zero skew tree* (ZST). In what follows, we address the

Zero Skew Clock Routing Problem (S): *Given a set S of sink locations, construct a ZST $T(S)$ with minimum cost.*

It should be noted that "zero skew" is well-defined only in the context of a method for evaluating signal delays. Due to the large size of typical clock routing instances, simple approximations such as linear delay or Elmore delay are used. As in related work on planar clock routing [17] and on minimum-cost zero-skew routing [3, 4, 5, 9], we develop our method in the context of the linear delay approximation: linear delay is simply the edge length, so intuitive geometric ideas can be applied to optimize this model. (In Section 5, we will discuss ways in which our method may be extended to other models such as Elmore delay.[1]) For the linear delay model, [15] observed that the general minimum-cost bounded-skew routing problem is NP-hard.[2] On the other hand, [3, 4, 5, 9] show that the variant formulation where the ZST $T(S)$ has a *prescribed* connection topology G can be solved in $O(|S|)$ time.

[1]The scope of our work does not address such issues as buffer insertion and wire-sizing. Thus, the most relevant design style is in some sense the "monolithic" single-buffer clocking approach for high-performance chips; cf. the discussion in Bakoglu's book [2] and the actual design of the DEC Alpha microprocessor, which uses the single-clock-buffer strategy [8]. We recognize the many artificial attributes of our formulation, even up to the objective of "exact zero skew", which is actually inappropriate from a power-management perspective.

[2]However, the NP-hardness of the minimum-cost *exact zero* skew clock routing problem has not been established.

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

Andrew B. Kahng and C.-W. Albert Tsao

UCLA Computer Science Department, Los Angeles, CA 90024-1596

Abstract

Clock routing has become a critical issue in the layout design of high-performance systems. Two important recent advances in this area are: (i) the DME (Deferred-Merge Embedding) algorithm of [3, 4, 5, 9] which embeds a given topology with exact zero pathlength skew and minimum total wirelength, and (ii) the method of [17], which is the first algorithm that guarantees a planar-embeddable clock routing solution. In this report, we show that the two passes (bottom-up and top-down) of the DME algorithm 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 very 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. We also obtain an average of 15.5% wirelength savings over the previous planar routing algorithm of [17].

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, with these sinks located quite arbitrarily throughout the layout region. Following [3, 5], 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 which we denote as $pl(T, v)$ (alternatively, we denote the placement of v simply as $pl(v)$). (The clock routing solution can have

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