RC Interconnect Optimization under the Elmore Delay Model

Sachin S. Sapatnekar
Department of Electrical Engineering and Computer Engineering
Iowa State University
Ames, IA 50011

ABSTRACT

An efficient solution to the wire sizing problem (WSP) using the Elmore delay model is proposed. Two formulations of the problem are put forth: in the first, the minimum interconnect delay is sought, while in the second, we minimize the net delay under delay constraints at the leaf nodes; previous approaches solve only the former problem. Theoretical results on these problems are proved, and a sensitivity-based algorithm is devised. It is shown experimentally that the second formulation provides significantly better engineering solutions.

1 Introduction

It is rapidly becoming obvious that with the current trends in technology, interconnect delays have become an increasingly dominant factor in determining circuit speed. Until recently, interconnect resistance was insignificant, while its capacitance was not, and hence optimal interconnect design frequently involved ensuring that all wire sizes were minimal. However, with advancement in technology, reduction in circuit geometries, increases in circuit speeds, and the advent of MCM's, the *wire sizing* problem (WSP) has become significant.

The problem of wire sizing has not received very much attention until recently. Cong *et al.* presented some work in the area in [1, 2]. The approach in [1] used a delay model based on an upper bound [3] on the Elmore delay, and minimized the delay of the interconnect under minimum and maximum wire width constraints. This was extended in [2], where the Elmore delay was directly used to perform the timing optimization. The form of the Elmore delay model in this work makes the assumption that the critical leaf nodes of the interconnect tree are provided by the user. This information, however, may not be available in all design situations, particularly in iterative optimization where the critical sinks may change between iterations. A weighted sum of the Elmore delays to these leaf nodes is minimized, where the weights are apparently user-defined.

In this work, we first use a form of the Elmore delay that does not require the critical leaf nodes to be specified. Like [1, 2], this work assumes that the interconnect network to be optimized is a tree structure. The objective here is to minimize the maximum of all Elmore delays at leaf nodes of the interconnect tree. Under this model, the separability property of the models in [1, 2] does not hold, and hence those algorithms will not provide the solution to this problem. Under this different delay model, we prove some properties of the WSP.

The route to formulating the problem is described in Section 2, and the two meaningful formal statements of the problem are suggested in Section 3. One formulation minimizes the overall delay of the tree, while the other minimizes the wiring area under delay constraints at leaf nodes of the tree. Properties of the two suggested formulations are described in Section 4, to lay the basis for an efficient algorithm to solve the problems, presented in Section 5. Finally, we present experimental results in Section 6, and conclude the paper in Section 7.

The results of this work also show that a goal of finding the absolute minimum delay of an interconnect tree does not provide good engineering solutions. Instead, a delay target of even 10-15% over the minimum delay can lead to a substantial savings in wiring area. The formulation that performs wire sizing under delay constraints serves to illustrate the area-delay tradeoff.

2 Formulation of the Problem

A. Modeling Interconnect and Interconnect Delay

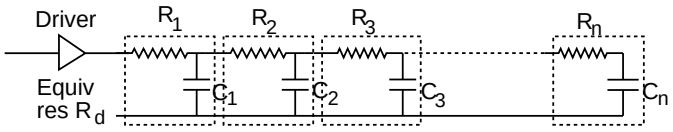


Figure 1: RC model of interconnect

This work models a wire as a succession of RC segments, shown in Figure 1, connected in series. The resistance, R_i, and capacitance, C_i, of the i^{th} segment are given by the formulæ

$$R_i = \rho l_i / w_i \qquad C_i = \beta l_i \cdot w_i \tag{1}$$

where w_i and l_i are, respectively, the width and length of the i^{th} segment. Under the above model, any interconnect tree can be modeled using an equivalent RC tree. In this work, we will use the words *width* and *size* interchangeably.

In an actual circuit, the root node is connected to a driver with equivalent resistance R_d. Moreover, in addition to wire capacitances, there may be several loading capacitances along the length of the wire. The Elmore delay to any node of the corresponding RC tree may easily be calculated using Eq. (2).

The delay $T_{d,i}$ of an RC tree is given by the well-known Elmore delay formula [3]. If P_i is the unique

path from the root of the RC tree to node i, and $desc(j)$ represents all nodes that are descendants of node j in the tree, then according to this formula, the delay to node i is given by

$$T_{d,i} = \sum_{j \in P_i} R_j \sum_{k \in desc(j)} C_k \qquad (2)$$

We take the Elmore delay of a tree as the maximum of the Elmore delays to any leaf node. An advantage of this definition is that the delay value for the tree is a physical quantity that a circuit designer can relate to immediately. Moreover, as will be shown later, this provides a natural extension into the problem of wire sizing under delay constraints. Note that our definition of the Elmore delay of a tree differs from the model in [2], where the user is required to identify the critical leaf nodes (we require no such user input), and a weighted sum of the Elmore delays to these leaf nodes is minimized.

B. Properties of the General WSP

We begin by proving a few results on the optimal wire sizes. Some of these results have been proved in [1, 2] for their delay model. We show here that some of those results are also valid under the Elmore delay model that we have used.

Definition 1 A wire width assignment f for a tree T is an n-tuple $[w_1, \cdots, w_n]$, where n is the number of wires, and w_i is the width of wire i.

Definition 2 Given a routing tree T, a wire width assignment f on T is a monotonic assignment if $w_p \geq w_c$ whenever wire S_p is an ancestor of wire S_c.

Definition 3 Given two wire width assignments f and f' on the same tree T, f dominates (is dominated by) f' if and only if $w_i(f) \geq w_i(f')$ $(w_i(f) \leq w_i(f'))$ for all wires $i \in$ T.

Definition 4 A wire assignment f for a tree T is *suboptimal* if there exists another wire assignment f' for T, different from f, such that f dominates f', and the Elmore delay to *every* node in T under assignment f' is no greater than that under assignment f.

Note that the definition of an optimal assignment here is open to interpretation under any formulation that uses the Elmore delay model, and that we have not restricted ourselves to a strict definition of optimality at this point. Under any reasonable definition of optimality, Definition 4 must hold.

The result in Theorem 1 below is, therefore, similar to, but more general than the analogous results presented in [1, 2] due to the more general definition of optimality that has been used here.

Theorem 1 [4] Any nonmonotonic wire width assignment f^* is suboptimal.

Theorem 2 [4] Let i be a leafnode, and let P_i be the path from the root node to i. Then the delay from the root to node i cannot be decreased by increasing any wire size that does not lie on P_i.

C. Does Separability Hold for this Delay Model?

Under the delay models used in [1, 2], it is shown that the width of each wire depends only on the widths of its ancestors and descendants. As a result, if $T_{SS1}, T_{SS2} \cdots T_{SSk}$ are the single-stem subtrees [1] rooted at node N, it has been proven under their delay models that the optimal wire width assignments for T_{SSi} can be determined independently of $T_{SSj}, j = 1 \cdots k, j \neq i$. This has been referred to as *separability*. By using this property, for a tree with n wires and r possible wire widths, algorithms of worst-case complexity $O(n^{r-1})$ have been proposed.

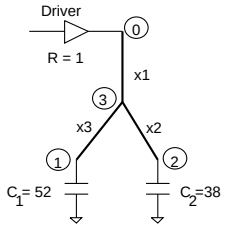


Figure 2: Counterexample for separability

Example: Consider the simple example shown in Figure 2. Assume, for simplicity, the following:

- Each branch resistance is related to the branch width by the relation, $R_i \propto 1/w_i$.
- Each branch capacitance is related to the branch width by the relation, $C_i \propto w_i$.
- The capacitive load at each branch is as shown in the figure.
- The maximum allowable wire size is 15 units.
- The driver has a resistance of 1 unit.

The delays to the two leaf nodes are given by the expressions:

$$D_1 = K \cdot (1 + \tfrac{1}{x_1})(x_1 + x_2 + x_3 + C_1 + C_2) + \tfrac{1}{x_3}(x_3 + C_1)$$
$$D_2 = K \cdot (1 + \tfrac{1}{x_1})(x_1 + x_2 + x_3 + C_1 + C_2) + \tfrac{1}{x_2}(x_2 + C_2)$$

where K corresponds to a proportionality constant. By enumeration, it was found that the minimum delay to leaf node 1 occurs when $x_1 = 10, x_2 = 1, x_3 = 7$, the minimum delay to leaf node 2 corresponds to the situation where $x_1 = 10, x_2 = 6, x_3 = 1$, while the maximum of the two delays was minimized at $x_1 = 10, x_2 = 4, x_3 = 5$, which shows that the single-stem subtrees cannot be optimized independently of each other.

The reason for this is easy to see. The delay to node 2 depends on the widths x_1 and x_2 (which act as both resistors and capacitors) *and* the width x_3 (which acts as a capacitive load). The optimal delay to node 2 implies that x_3 must be minimal; however, this could cause the delay to node 1 to be too large. At the optimum, there is a "balance" between the resistance of x_3 that causes a small delay to node 1, and the capacitance of x_3 that causes a small delay to node 2 as well. Thus, the sizing along the path from the root to node 2 is dependent on the sizes of branches that are off this path, and hence separability does not work. □

3 Statement of the WSP

As mentioned earlier, several viable definitions of optimality are possible. We now address two problems:

- Wire sizing for minimum delay
- Wire sizing under delay constraints

The corresponding optimization problems are:

Problem P1 minimize $(\max_{i \in leafnode(\mathrm{T})} d_i)$
subject to $w_j < w_{j,spec} \ \forall \ j = 1 \cdots n$.

Problem P2 minimize $\sum_{i \in \mathrm{T}} w_i$
subject to $d_i < D_{spec} \ \forall \ i \in leafnode(\mathrm{T})$
and $w_j < w_{j,spec} \ \forall \ j = 1 \cdots n$.

By Theorem 1, for both problems, any nonmonotonic solution is suboptimal.

4 Properties of the Continuous WSP

Definition 5: The *continuous WSP* is the problem of finding optimal wire widths to solve the WSP, such that wire widths may take on any real value. This is in contrast to the *(discrete) WSP* where the wire widths are constrained to be integers.

Property 1: The delay along any path of an RC tree is a posynomial [5] function of the sizes of wires in the tree.

Property 2: The continuous WSP's **P1** and **P2**, stated in Section 3, are unimodal, i.e., any local minimum of these problems is a global minimum.

To observe this, note that the simple transformation, $(w_i) = (e^{x_i})$, transforms any posynomial function of the w_i's to a convex function of the x_i's [5]. Hence, under this transformation, for both problems, the objective function as well as the constraints are convex. As a consequence of the fact that the mapping function is one-to-one, it is easy to see that the optimization problems **P1** and **P2** are unimodal.

It may be worthwhile to caution the reader here that it is only the *continuous* WSP that is unimodal; the (discrete) WSP is combinatorial, and no such statements can be made about it. However, a solution to the continuous WSP gives a lower bound on the solution to the discrete problem.

5 A Sensitivity-based Algorithm

Since the enumerative solution to the WSP with n wires and r permissible sizes is of complexity $O(r^n)$, we propose a heuristic.

The heuristic presented here is efficient and sensitivity-based. A solution to the continuous WSP is first found; next, the discrete solution is found by using a mapping algorithm to round off wire sizes to the next higher or lower integer. As shown in Section 6, this causes an insignificant degradation in the quality of the solution.

The pseudo-code representing the algorithm WIMIN is shown in Figure 3. In each iteration, the leafnode with the largest violation is identified; this

```
BEGIN ALGORITHM WIMIN()
 F = bumping factor;
 while (stopping criterion not met)
   current_leaf_node = leaf node with the
          largest delay violation;
   maxsensitivity = 0;
   maxsensitivity_wire = -1;
   for each wire i that is an ancestor
       of current_leaf_node
     if F * width(i) > width(predecessor[i])
       continue;
     if sensitivity S_i < maxsensitivity
       maxsensitivity = S_i;
       maxsensitivity_wire = i;
   if (maxsensitivity_wire == -1)
     /* minimum delay has been found */
     exit;
   width(maxsensitivity_wire) *= F;
 MAP();
END ALGORITHM WIMIN()
```

Figure 3: Pseudocode for continuous wire sizing.

will be referred to as the current leaf node. We calculate the sensitivity, S_i of wire i by finite differences as

$$S_i = \frac{Delay(F \cdot w_i) - Delay(w_i)}{(F-1) \cdot w_i} \qquad (3)$$

where *Delay* is the delay from the root node to the current leaf node, and F is a number just larger than 1. (Although the exact sensitivity of the delay function could have been computed here, since we will be taking steps of discrete sizes, it is more beneficial to compute the sensitivity as a finite difference.) By Theorem 2, the delay of the current leafnode can only be decreased by increasing the sizes of wires that lie on the path between the root node and that leafnode. The sensitivity of each such wire is identified, and the size of the single wire with the minimum negative sensitivity is bumped up by multiplying it by the same constant factor, $F > 1$, as in Equation (3) (typical values of F are 1.2 or 1.5). This ensures that the delay to the current leafnode is reduced in every iteration.

Note that due to the monotonicity property, it is unnecessary to compute the sensitivity for any wire for which the bumping operation violates monotonicity.

The stopping criterion for the iterations is satisfied when no wire has a negative sensitivity, which gives the solution to the unconstrained Problem **P1**, or until the delay specifications at all leaf nodes are met, which provides the solution to the constrained Problem **P2**.

The mapping algorithm is illustrated in Fig. 4. It starts from the leafnode, L, with the largest delay, and processes each wire on the path between node L and the root node. If the size of the current wire is an integer, its size remains unchanged. If not, the change in the delay to L caused by changing the wire size to the closest higher (lower) integer, w_{i+} (w_{i-}) is computed, and one that creates a smaller delay fluctuation is selected. L is now marked as "processed" and the algorithm proceeds iteratively with the unprocessed leafnode that has the largest delay. Note that in the mapping phase, each wire is considered only once.

```
BEGIN ALGORITHM MAP()
 Mark all wires as unprocessed;
 Mark all leafnodes as unprocessed;
 while (all leafnodes not processed)
   current_leaf_node = unprocessed leaf node
         with the largest delay;
   for each unprocessed wire i that is an
         ancestor of current_leaf_node
     if (width(i) is an integer) continue;
     w_{i+} = ⌈width(i)⌉
     w_{i-} = ⌊width(i)⌋
     if (| delay(w_{i+}) − delay(width(i)) |
         <| delay(w_{i-}) − delay(width(i)) |)
       width(i) = w_{i+};
     else
       width(i) = w_{i-};
END ALGORITHM MAP()
```

Figure 4: Pseudocode for the mapping algorithm.

6 Experimental Results

The WIMIN algorithm was run on twelve test networks. The technology parameters used are those presented in [1, 2].

The algorithm is implemented in C on a DECstation 5000/133. Experimental results for Problem **P1**, in which the wire sizes that correspond to the minimum interconnect delay are found for each of the test circuits, are shown in Table 1. The value of the multiplicative factor, F, is set to 1.2 here.

During our experiments, an additive factor was tried instead of a multiplicative factor; however, this was found to give poorer results. This may be attributed to the fact that wires near the source need to be sized more than those near the leaf nodes, and the general profile of the correctly sized wires resembles a geometric, rather than an arithmetic progression.

For each circuit, we show the cost and delay of the unsized circuit, i.e., the circuit in which all wires have unit width. As mentioned earlier, the cost is taken as the sum of wire sizes. The next two three-column sets show the cost, RC delay, and the execution time for the optimization, when the maximum allowable wire size is 2 and 6, respectively. Note that the computation time of the algorithm is very reasonable. With some increase in wire sizes, it can be seen that the interconnect delay can be improved significantly.

The bulk of the CPU time is incurred by the continuous optimization problem, and only a small fraction (under 10%) is attributable to the mapping phase. The run times are reasonable even for large circuits.

In the last two columns of Table 1, for the case when the maximum allowable wire size is 6, the delay constraint is relaxed to 15% over the minimum delay, and problem **P2** is solved. We apply a uniform timing constraint on each leaf node of the tree. Note that the nature of the algorithm is such that there may be different delay specifications at each of the leaf nodes for Problem **P2**, and not a uniform specification. For no reason in particular, however, we restrict ourselves to a uniform timing constraint for all leaf nodes in this section. It must be stressed, however, that the algorithm is general enough to handle nonuniform timing constraints too. The corresponding cost and run times are shown. The figures in brackets under the "Cost" column represent the % cost reduction compared to the minimum delay case. Improvements of as much as 46% are seen; note that the actual improvement in chip area may be even better, since our cost function is a very simple measure of routing expense.

Next, we present results on Problem **P2**, i.e., on minimizing interconnect delay under timing constraints, graphically on two specific circuits in Figure 5. This picture serves to illustrate the area-delay tradeoff made during wire sizing. As before, the value of the factor F in Algorithm 3 is set to 1.2.

The results plotted in Figure 5 show the true utility of using the problem formulation **P2**. It is observed that the interconnect area overhead required to achieve the minimum possible delay is extremely high, for the last fraction of delay reduction. While some of this is attributable to suboptimality of the sensitivity-based algorithm, the same characteristics were found to hold when the factor F was very close to 1, when the solution is close to optimal. This explains why, in Table 1, substantial improvements in the cost functions are achieved when the constraints are relaxed by a small amount.

It was found that the delay corresponding to the mapped discrete solution is *always* within about 10% of the continuous solution, thereby providing us with an upper bound on the deviation of the solution from the optimum. The larger errors are in the cases where the amount of sizing is relatively small and it is possible that in these cases, a large portion of the difference between the continuous and discrete solutions is due to discretization noise. If the factor $F = 1 + \delta$, then the quality of the continuous solution can be enhanced by making δ smaller.

The continuous sizing solution is, by the construction of the algorithm, less than the specification. However, the discrete solution delay is not always so, and may provide a solution that has slightly larger delay than the specification. This is not critical, since the Elmore delay model is known to be accurate only up to 10 or 20 %, whereas the discrepancy between the discrete solution delay and the specification is less, and some is attributable to discretization noise.

In the experiments above, it was assumed here that each wire segment is sized at a time. Experiments were also conducted where individual *grid* segments were sized (a grid segment corresponds to a single RC segment from Figure 1), so that the width of a wire segment is not uniform along its entire length. In such a case, it was found (as expected) that greater amounts of delay reduction were possible [4]. It was also seen that the difference between the continuous solution and the mapped solution was within 5% for all tested cases; this reduction in the gap occurs because the discretization noise for this case is smaller.

The drawbacks of sizing a grid segment at a time instead of a wire segment at a time are twofold: firstly, the run times are much larger (about an hour for Intct12, which has 999 segments), and secondly and more seriously, the nonuniform wire sizes could have serious repercussions on the routability of the layout. Nevertheless, this serves to illustrate the fact that the proposed algorithm can easily be generalized to handle this case, and any intermediate sizing strategies.

Table 1: Results of Minimizing Interconnect Delay.

Circuit	Unsized		Maxsize =2			Maxsize = 6				
			Minimum delay			Minimum delay			$D_{spec} = 1.15 \times D_{min}$	
	Cost	Delay (ns)	Cost	Delay	CPU	Cost	Delay	CPU	Cost	CPU
Intct1		1.622	118	1.161	1.1s	161	0.931	2.3s	128 (26%)	0.9s
Intct2		2.526	128	1.652	0.8s	189	1.182	1.3s	143 (32%)	0.6s
Intct3	99	2.710	120	1.787	0.8s	182	1.186	1.7s	144 (26%)	0.5s
Intct4		1.759	120	1.288	1.2s	180	1.087	2.4s	123 (46%)	0.9s
Intct5		2.231	115	1.650	0.4s	223	1.214	0.6s	163 (37%)	0.4s
Intct6		0.872	551	0.715	5.0s	672	0.633	13.1s	527 (28%)	1.4s
Intct7	499	1.002	565	0.774	5.2s	739	0.664	12.0s	552 (34%)	2.1s
Intct8		1.297	609	0.935	6.0s	864	0.740	13.1s	643 (35%)	3.1s
Intct9		1.236	700	0.865	4.0s	1072	0.689	4.8s	732 (46%)	3.1s
Intct10		1.540	1108	1.132	11.1s	1376	0.903	29.2s	1168 (18%)	5.8s
Intct11	999	2.387	1226	1.601	15.9s	1712	1.123	34.1s	1385 (24%)	7.4s
Intct12		3.102	1178	2.012	14.7s	2033	1.369	27.4s	1529 (33%)	7.4s

7 Conclusion

A new algorithm for interconnect sizing has been described in this paper. The WSP is solved under an Elmore delay model that does not require the critical leaf nodes to be specified. The problem of obtaining the optimal wire sizes under delay constraints is addressed for the first time and area-delay tradeoff curves are shown. Further, it is shown experimentally that achieving the absolute minimum delay for a net involves a wasteful use of resources; instead, a delay target of even 10-15% over the minimum delay provides a good engineering solution.

Acknowledgments

The author would like to thank the anonymous reviewers and Prof. Jason Cong for their constructive comments on the manuscript.

References

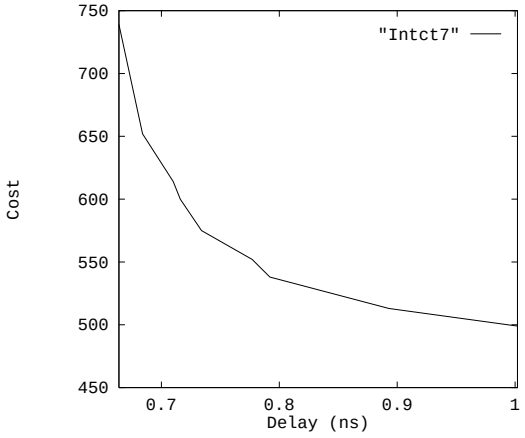
[1] J. Cong, K.-S. Leung, and D. Zhou, "Performance-driven interconnect design based on distributed RC model," in *Proceedings of the ACM/IEEE Design Automation Conference*, pp. 606–611, 1993.

[2] J. Cong and K.-S. Leung, "Optimal wiresizing under the distributed Elmore delay model," in *Proceedings of the IEEE International Conference on Computer-Aided Design*, pp. 634–639, 1993.

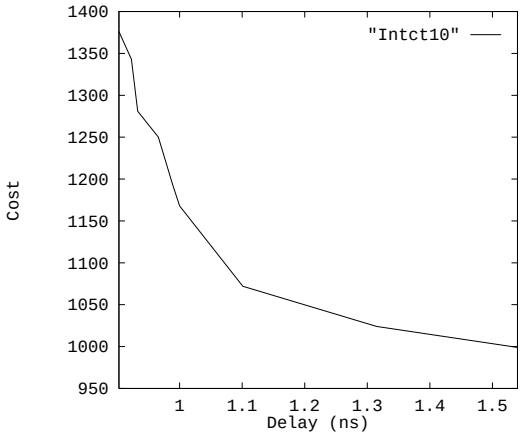
[3] J. Rubenstein, P. Penfield, and M. A. Horowitz, "Signal delay in RC tree networks," *IEEE Transactions on Computer-Aided Design*, vol. CAD-2, pp. 202–211, July 1983.

[4] S. S. Sapatnekar, "RC interconnect optimization under the Elmore delay model," Tech. Rep. ISU-CPRE-94-SS03, Iowa State University, Ames, IA, 1994.

[5] J. Ecker, "Geometric programming: methods, computations and applications," *SIAM Review*, vol. 22, pp. 338–362, July 1980.



(a)



(b)

Figure 5: Cost vs. delay for (a) Intct7 (b) Intct10