

Interconnect Design Using Convex Optimization

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

ABSTRACT

Two wire sizing formulations for optimizing interconnect are presented. The first minimizes the delay under wire width constraints, while the second minimizes the wiring area under delay and width constraints. A convex programming formulation is proposed, and an efficient algorithm is used to perform the optimization. Experimental results show that the first formulation, which has been the prevalent one in the literature, provides bad engineering solutions, and that the second formulation leads to significantly better results.

1 Introduction

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 widths were minimal. This is no longer true, and hence, interconnect optimization has become significant.

The problem of sizing wires for high performance 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 crude delay model to minimize the delay of the interconnect under minimum and maximum wire width constraints. This was extended in [2], where the Elmore delay model was used to perform the timing optimization. The form of the delay function in this work, however, requires a great deal of user input. For a more general delay model [3], the property of separability discussed in [1, 2] does not hold, thereby making the wire sizing problem much harder, and rendering those algorithms unusable.

The work in [3] points out the deficiencies of that model and presents a heuristic sensitivity-based algorithm for the wire sizing problem. This work is an enhancement of that approach, whereby the problem is formulated as a convex programming problem, and an efficient technique is used to find the solution. Like [1–3], this work optimizes interconnect tree struc-

tures. We first state some properties of the wire sizing problem under this model, and describe two meaningful formulations of the problem in Section 2. Some further properties of the two suggested formulations are provided in Section 3, to lay the basis for an efficient algorithm to solve the problems, presented in Section 4. Finally, we present experimental results in Section 5, and conclude the paper in Section 6.

Apart from the deficiencies associated with the models used in [1, 2], the approach in [1] attempts to find the wire widths that provide the absolute minimum delay. This is a bad engineering solution, since a delay target of even 10% over the minimum delay can give a substantial savings in wiring area. The technique in [2] makes some attempt to resolve this by optimizing a weighted sum of the areas and the delays. This is an imprecise statement of the problem and depends on user input to provide the weighting functions. One of the formulations that we suggest minimizes the wiring area under delay constraints.

2 Formulation of the Problem

A. Modeling Interconnect Delay

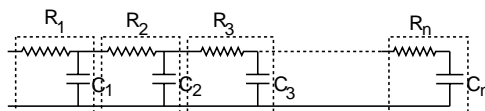


Figure 1: RC model of interconnect

A wire is modeled as a succession of RC segments, shown in Figure 1. The resistance, R_i , and capacitance, C_i , of the i^{th} segment are given by the formulæ

$$\begin{aligned} R_i &= \rho l_i / w_i \\ C_i &= \beta l_i \cdot w_i, \end{aligned} \quad (1)$$

where w_i and l_i are, respectively, the width and length of the i^{th} segment. Under this model, any interconnect tree can be modeled using an equivalent RC tree.

The delay $T_{d,i}$ of an RC tree is given by the well-known Elmore delay formula [4]. 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

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

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

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 sinks (we require no such user input), and a weighted sum of the Elmore delays to various sinks is minimized. The weights appear to be user-specified.

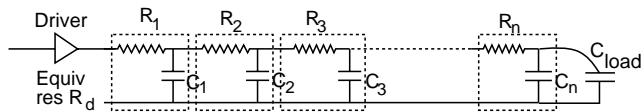


Figure 2: RC line driven by a gate

B. Properties of the General Wire Sizing Problem

We begin by stating a few results on the optimal wire widths; proofs are provided in [3]. At this point, we make minimal assumptions, so that Theorem 1 below is true for any reasonable definition of optimality.

Definition 1 A wire width assignment f for a tree T is a n -tuple $[w_1, \dots, w_n]$, 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, f' on a tree T , f dominates (is dominated by) f' iff. $w_i(f) \geq w_i(f')$ ($w_i(f) \leq w_i(f')$) 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 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 is open to interpretation under the Elmore delay model, and that we have not restricted ourselves to a strict definition of optimality at this point. However, under any reasonable definition of optimality, Definition 4 must hold.

Theorem 1 [3]: Any nonmonotonic wire width assignment f^* is suboptimal.

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

Problem P1 for min. delay under width constraints
 minimize $(\max_{i \in leafnode(T)} d_i)$
 subject to $W_{j,min} \leq w_j \leq W_{j,max} \quad \forall j = 1 \dots n$.

Problem P2 for minimum wire area under delay and width constraints
 minimize $\sum_{i \in T} w_i$
 subject to $d_i \leq D_{spec} \quad \forall i \in leafnode(T)$
 and $W_{j,min} \leq w_j \leq W_{j,max} \quad \forall j = 1 \dots n$.

Due to the general nature of Theorem 1, both problems **P1** and **P2** have the property that any nonmonotonic solution is suboptimal.

3 Properties of the Continuous Wire Sizing Problem

Definition 5: The *continuous wire sizing problem* is the problem of finding optimal wire widths to solve the wire sizing problem, such that wire widths may take on any real value. (In the *discrete wire sizing problem*, the wire widths must be integers.)

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

Property 2: The continuous wire sizing problems **P1** and **P2** are unimodal, i.e., any local minimum of these optimization 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* wire sizing problem that is convex. The (discrete) wire sizing problem is combinatorial; no such statements can be made about it. However, a solution to the continuous wire sizing problem gives a lower bound on the discrete solution.

4 The Optimization Algorithm

The optimization algorithm used here is an efficient convex programming algorithm [6]. Details of the implementation are provided in [7]. The algorithm works in an n -dimensional space, where n is the number of variables. It encloses the solution in an initial polytope, which is taken to be a box. In each iteration, the center of this polytope is determined, and an oracle is invoked to determine whether or not this center lies

```

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;
         $w_{i+} = \lceil width(i) \rceil$ 
         $w_{i-} = \lfloor width(i) \rfloor$ 
        if ( $|delay(w_{i+}) - delay(width(i))|$ 
             $< |delay(w_{i-}) - delay(width(i))|$ )
             $width(i) = w_{i+}$ ;
        else
             $width(i) = w_{i-}$ ;
        }
    }
}

```

Figure 3: Pseudocode for the mapping algorithm.

within the feasible region of the optimization problem. Feasibility checks may be performed by applying Theorem 1 to check for monotonicity; a nonmonotonic wire assignment is automatically infeasible. If the wire assignment is monotonic, a full delay calculation must be carried out to check for feasibility.

A hyperplane is then passed through the center such that the solution to the problem lies on one side of it; the equation of this hyperplane is determined using the gradient of the objective function if the center is feasible, or the gradient of a violated constraint if it is not. This hyperplane roughly halves the volume of the original polytope. The center of this polytope is then found and the procedure is continued until the volume of the polytope is sufficiently small.

The mapping algorithm is illustrated in Figure 3. 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 width of the current wire is an integer, its width remains unchanged. If not, the change in the delay to L caused by changing the wire width 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. In this algorithm, once a wire has been processed, its width remains unchanged. Thus, each wire is considered only once.

5 Experimental Results

The algorithm described above was implemented in C as the program COSI (Convex Optimization for Sizing Interconnect) on a DECstation 5000/240.

COSI was run on several test networks, using advanced MCM technology parameters from [1].

Experimental results for COSI/P1, in which the wire sizes that correspond to the minimum interconnect delay is found for each of the test circuits, are shown in Table 1. For each circuit, we show the cost (sum of wire sizes) and delay of the unsized circuit in which all wires have unit width. The remaining columns show the cost and delay(ns) after optimization, the number of iterations of the convex programming algorithm, and the CPU time. The results show that with some increase in wire sizes, a significant improvement in interconnect delay is possible.

The computation time of the algorithm is quite reasonable, especially when one considers that the algorithm finds the *exact* solution to the continuous sizing problem. The bulk of the CPU time is incurred by the continuous optimization problem, and only a small fraction (under 10%) is due to the mapping phase.

Next, we present results on COSI/P2, i.e., on minimizing interconnect delay under timing constraints, in Figure 4. In our experiments, a uniform timing constraint is applied at each leaf node. Note that the algorithm is general enough to allow different delay specifications at each of the leaf nodes rather than a uniform constraint.

The graphs show that the area overhead required to achieve the minimum possible delay is extremely high, for the last fraction of delay reduction. Therefore, the goal of minimizing delay is a not a very good engineering goal, and substantial savings can be accrued by relaxing the delay constraint slightly.

The delay corresponding to the mapped discrete solution was *always* seen to be within 5% of the continuous solution, thereby providing 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 in these cases, the difference is very likely to be due to discretization noise. Note also that the delay of the solution to the continuous sizing problem 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 considerably less.

6 Conclusion

A new algorithm for interconnect sizing has been described in this paper. This presents an improvement over the previously published work in [1, 2] since uses

Table 1: RESULTS OF COSI/**P1**: MINIMIZING INTERCONNECT DELAY.

Circuit	Unsize		$W_{max} = 3 \times W_{min}$				$W_{max} = 5 \times W_{min}$			
	Cost	Delay	Cost	Delay	# Iter	CPU time	Cost	Delay	# Iter	CPU time
Intcta	99	1.622	144	1.061	48	57.1s	147	0.967	73	90.6s
Intctb	99	2.526	155	1.393	35	37.6s	181	1.215	49	53.6s
Intctc	99	2.710	137	1.484	40	50.4s	171	1.267	51	64.7s
Intctd	499	0.872	601	0.666	56	347.7s	673	0.635	68	424.5s
Intcte	499	1.002	631	0.722	47	339.8s	711	0.674	70	497.2s
Intctf	499	1.297	684	0.837	43	381.2s	785	0.758	59	520.8s
Intctg	999	1.540	1231	0.981	47	1269.6s	1386	0.899	47	1263.9s
Intcth	999	2.387	1437	1.398	38	1468.1s	1567	1.147	47	1819.9s
Intcti	999	3.102	1598	1.768	37	1508.9s	1850	1.456	49	2147.1s

(a)

(b)

Figure 4: Results of COSI/**P2**: Cost vs. Delay for (a) Circuit Intcte (b) Circuit Intctg

a more accurate Elmore delay model that recognizes that subtrees of a tree cannot be sized independently of each other. Some theoretical results on this model are first stated, and then it is formulated as a convex programming problem. The problem is solved using an efficient optimization algorithm. Using this algorithm to solve problem **P2** in Section 2, that carries out optimization under delay constraints, rather than Problem **P1** that tries to minimize the delay, can lead to good engineering solutions.

REFERENCES

- [1] J. Cong, K.-S. Leung, and D. Zhou, "Performance-driven interconnect design based on distributed RC model," Tech. Rep. CSD-920043, UCLA CS Department, Oct. 1992. (Also see *Proc. DAC-93*).
- [2] J. Cong and K.-S. Leung, "Optimal wiresizing under the distributed Elmore delay model," Tech. Rep. CSD-930012, UCLA CS Department, Apr. 1993. (Also see *Proc. ICCAD-93*).
- [3] S. S. Sapatnekar, "RC interconnect optimization under the Elmore delay model," Tech. Rep. ISU-CPRE-93-SS12, Department of EE/CprE, Iowa State University, Oct. 1993.
- [4] J. Rubenstein, P. Penfield, and M. A. Horowitz, "Signal delay in RC tree networks," *IEEE Trans. Computer-Aided Design*, vol. CAD-2, pp. 202-211, July 1983.
- [5] J. Ecker, "Geometric programming: methods, computations and applications," *SIAM Rev.*, vol. 22, pp. 338-362, July 1980.
- [6] P. M. Vaidya, "A new algorithm for minimizing convex functions over convex sets," *Proc. IEEE Foundations of Computer Science*, pp. 332-337, Oct. 1989.
- [7] S. S. Sapatnekar, V. B. Rao, P. M. Vaidya, and S. M. Kang, "An exact solution to the transistor sizing problem for CMOS circuits using convex optimization," *IEEE Trans. Computer-Aided Design*, vol. 12, pp. 1621-1634, Nov. 1993.