

Interconnect Delay and Area Estimation for Multiple-Pin Nets *

Jason Cong and David Zhigang Pan
Department of Computer Science
University of California, Los Angeles, CA 90095
Email: {cong,pan}@cs.ucla.edu

Abstract

This paper studies interconnect delay and area estimation for multiple-pin nets with consideration of interconnect optimizations, including optimal wire sizing (OWS), and simultaneous buffer insertion/sizing and wire sizing (BISWS), under two types of optimization objectives: one is to minimize delay to a single critical sink, and the other is to minimize the maximum delay to all critical sinks. Our estimation models are accurate, yet an order of ten thousand times faster when compared with running corresponding complex optimization algorithms directly. Therefore, they are expected to be very useful to guide the interconnect-centric synthesis and planning for deep submicron circuit designs.

1 Introduction

As VLSI technology moves to deep submicron (DSM) dimensions and giga-hertz clock frequencies, interconnects play a dominant role in determining the overall chip performance. Recently, many interconnect optimization techniques, including optimal wire sizing, buffer insertion and sizing, etc., have been proposed and shown to be very effective in reducing interconnect delays (e.g., by a factor of 5 to 6 times, as shown in a recent survey [1]). However, in the conventional VLSI design flow, interconnect optimization is usually performed at late stages during the design process. Consequently, accurate interconnect delay and area, especially those for global interconnects are not known to synthesis and planning tools. Without proper modeling of the impact of interconnect optimization, these tools are less likely to make correct high level decisions.

The problem in this study is to develop *efficient* yet *accurate* interconnect estimation models that are suitable for high level synthesis/planning tools with consideration of physical level interconnect optimizations. These estimation models provide exactly an enabling mechanism to effectively couple the synthesis/planning tools with layout optimizations, and to assure the design convergence.

So far, there has been very limited work on interconnect estimation that considers interconnect layout

optimization. [2] provides the first systematic study on interconnect delay estimation with interconnect optimization. It derived a set of simple delay estimation models (DEM) under various optimization techniques, e.g., optimal wire sizing (OWS), simultaneous driver and wire sizing (SDWS), and simultaneous buffer insertion/sizing and wire sizing (BISWS). These DEM's are shown to have about 90% accuracy when compared with running corresponding complex interconnect optimization algorithms, e.g., those from UCLA Tree-Repeater-Interconnect-Optimization (TRIO) package [1] directly. However, the DEM's in [2] are developed for 2-pin nets. They cannot be applied directly for multiple-pin nets with possibly many critical sinks. Moreover, [2] does not have wire area estimation, which shall also be planned beforehand at high levels to make sure that the planned interconnect optimization will be realizable at the layout level.

In this paper, we study interconnect delay and area estimations for multiple-pin nets with tree topologies, under two types of optimization objectives: (i) minimize the delay to a single critical sink, (ii) minimize the maximum delay for all critical sinks (i.e, the *tree delay*, as defined in [3]). Other objectives such as weighted delay are possible, but not in the scope of our current study. For each objective, we perform OWS or BISWS for performance optimization. Our main contributions include the following:

- Under objective (i), we formulate the original problem into a single-line-multiple-load (SLML) problem. We then transform SLML into an equivalent single-line-single-load (SLSL) problem, and obtain the delay and area estimation.
- Under objective (ii), we obtain a lower bound delay estimation for the optimal tree delay and show that in practice, this delay can be used to approximate the optimal tree delay.

The rest of the paper will be organized as follows. Section 2 formulates the problem and states some preliminaries. Section 3 studies delay and area estimation for a single critical sink, i.e., under objective (i). Section 4 studies delay estimation for multiple critical sinks, i.e., under objective (ii). The concluding remarks follow in Section 5.

*This research is partially sponsored by Semiconductor Research Corporation under Contract 98-DJ-605 and a grant from Avant! Corporation under the California MICRO Program.

2 Problem Formulation and Preliminaries

Given a multiple-pin interconnect network of tree structure with driver G and a set of sinks S_1, S_2, \dots, S_n , as shown in Figure 1, our problem is to seek efficient yet accurate delay/area estimations with consideration of various interconnect optimizations, such as OWS or BISWS. In Figure 1, G 's input waveform is generated by a nominal gate G_0 connected with an ideal voltage source. Each sink S_i has loading capacitance C_{si} . The delay to be minimized is from the input of G_0 , while the delay to be estimated is the stage delay from the input of G to a single or multiple critical sinks. The input stage delay is included during the optimization such that it acts as a constraint not to over-size G during the interconnect optimization. For OWS, the driver G 's size is fixed; for BISWS, G 's size is not fixed and will be determined optimally. In our study, we use the following two performance-driven optimization objectives: (i) minimizing delay from source to a single critical sink, (ii) minimizing maximum delay of all critical sinks, i.e., minimizing the tree delay.

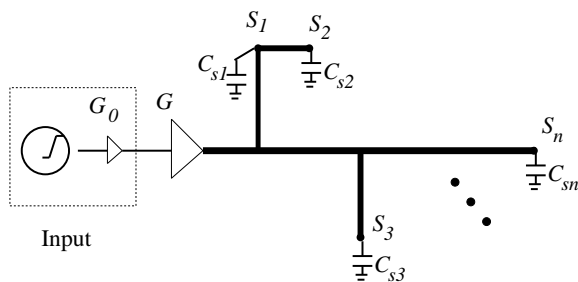


Figure 1: Problem formulation.

To derive efficient and accurate interconnect estimation models, we need simple but reasonably accurate delay computation models, as well as a set of key interconnect and device parameters. We model each gate (or buffer) as a switch-level RC circuit [1] and use the well-known Elmore delay model [4] for delay computation. Although Elmore delay model is not very accurate in DSM design, especially for delay calculation of near-source nodes due to the resistive shielding [5], it is still a proper metric for our delay estimation purpose to provide guidance to high-level design planning. The following are the key interconnect and device parameters for our study.

- W_{min} : the minimum wire width, in μm
- S_{min} : the minimum wire spacing in μm
- r : the sheet resistance, in Ω/\square
- c_a : the unit area capacitance, in $fF/\mu m^2$

- c_f : the unit effective-fringing capacitance¹, in in $fF/\mu m$
- t_g : the intrinsic device delay in ps
- c_g : input capacitance of a minimum device, in fF
- r_g : output resistance of a minimum device, in $k\Omega$

We derive these parameters based on *1997 National Technology Roadmap for Semiconductors* (NTRS'97) [6] (see [7] for details of these parameters).

3 Estimation for Single Critical Sink (SCS)

In this section, we study interconnect delay and area estimation under single critical sink (SCS) formulation, with consideration of two interconnect optimization techniques, OWS and BISWS.

3.1 Optimal Wire Sizing for SCS

For delay minimization to a single critical sink S_k , OWS will only size wire segments along the critical path (i.e., the path from G to S_k), and use minimum width for all other wire segments not on the critical path, so that the wire load from non-critical sinks is minimum. Since the wire load at each branch from the critical path can be pre-computed before performing OWS, we can transform the original OWS problem with tree topology into an equivalent *single-line-multiple-load* (SLML) problem, as shown in Figure 2. In the figure, R_d is the effective resistance of the driver G , and S_k is the single critical sink. At each branch i on the critical path, C_i is the total effective downstream capacitance (excluding that from the critical path).

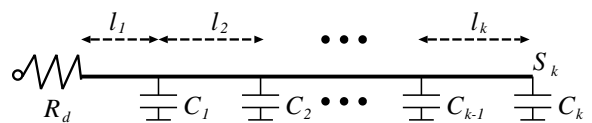


Figure 2: The single-line-multiple-load (SLML) problem for critical sink S_k . $l_1 + l_2 + \dots + l_k = l$.

In [7], it was shown that a simple wire sizing scheme that uses the best single width (i.e., the optimal 1-WS solution) can approximate the delay and area of an OWS solution with many wire width selections reasonably well. So we will first start with a single-width sizing. Under 1-WS, We can transform the SLML problem into a much simpler problem with only two loading capacitances, one is at the source and the other is at the critical sink. The transformation is formally described by the following theorem:

¹It is defined as the sum of fringing and coupling capacitances.

Theorem 1 Under the Elmore delay model, SLML in Figure 3 (a) is equivalent to SLDL in Figure 3 (b) for any wire width w , with

$$C_L = \sum_{j=1}^k \frac{\sum_{i=1}^j l_i}{l} \cdot C_j \quad (1)$$

$$C_0 = \sum_{j=1}^k C_j - C_L \quad (2)$$

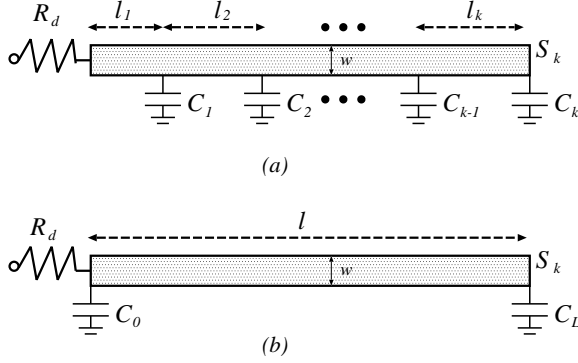


Figure 3: (a) single-line-multiple-load (SLML) under uniform wire sizing w is equivalent to (b) single-line-double-load (SLDL) under the Elmore delay model, where C_0 and C_L is obtained from Theorem 1 and $l_1 + l_2 + \dots + l_k = l$.

Proof: For the j -th wire segment, denote $c_j = (c_a w + c_f) \cdot l_j$ and $r_j = \frac{r}{w} \cdot l_j$. The Elmore delay of SLML problem in Figure 3 (a) can be written as

$$\begin{aligned} T &= R_d \left(\sum_{j=1}^k c_j + \sum_{j=1}^k C_j \right) + r_1 \left(\frac{1}{2} c_1 + \sum_{j=2}^k c_j + \sum_{j=1}^k C_j \right) \\ &\quad + r_2 \left(\frac{1}{2} c_2 + \sum_{j=3}^k c_j + \sum_{j=2}^k C_j \right) + \dots + r_k \left(\frac{1}{2} c_k + C_k \right) \\ &= R_d (c_f l + c_a w l + \sum_{j=1}^k C_j) \\ &\quad + r \left(c_a + \frac{c_f}{w} \right) \cdot \left[\frac{1}{2} \sum_{j=1}^k l_j^2 + \sum_{j=1}^{k-1} \sum_{i=j+1}^k l_j l_i \right] \\ &\quad + \sum_{j=1}^k (C_j \sum_{i=1}^j r_j) \\ &= R_d C_0 + R_d (c_f l + c_a w l + C_L) \\ &\quad + \frac{1}{2} r \left(c_a + \frac{c_f}{w} \right) \cdot l^2 + \frac{r l C_L}{w} \end{aligned} \quad (3)$$

Therefore, it is equivalent to the Elmore delay of SLDL in Figure 3 (b). \square

Intuitively, Theorem 1 transforms SLML into SLDL by redistributing any internal loading capacitance C_i into two parts. One part of C_i goes to C_L at the sink

S_k based on the ratio of C_i 's upstream wire resistance to total resistance on the critical path. And the other part of C_i goes to C_0 at the source to preserve the constant term $R_d C_i$. For the SLDL problem, since $R_d C_0$ is constant regardless of different wire sizes, we can further take it out without affecting wire-sizing solution and reduce SLDL to a single-line-single-load (SLSL) problem. Note that Theorem 1 holds for any wire width w . From (3), we can compute the best single-width w^* that minimizes the Elmore delay for (3).

$$w^* = \sqrt{\frac{r(c_f l + 2C_L)}{2R_d c_a}} \quad (4)$$

And the optimal Elmore delays for SLML and SLSL using w^* are the same,

$$\begin{aligned} T_{1ws/SLML} &= T_{1ws/SLSL} \\ &= R_d C_0 + R_d (c_f l + C_L) + \frac{1}{2} r c_a l^2 \\ &\quad + 2\sqrt{r R_d c_a \left(\frac{1}{2} c_f l + C_L \right)} \cdot l \end{aligned} \quad (5)$$

As mentioned earlier, we have observed in [7] that $T_{1ws/SLSL}$ is a reasonable estimation for $T_{ows/SLSL}$, as $T_{ows/SLSL}$ is usually between 0.8 to 0.95 times $T_{1ws/SLSL}$. Similarly, $T_{ows/SLML}$ is also found to be about 0.8 to 0.95 times $T_{1ws/SLML}$. Since $T_{1ws/SLSL} = T_{1ws/SLML}$ from (5), we can then use $T_{ows/SLSL}$ to estimate $T_{ows/SLML}$, with at most $(0.95-0.8)/0.8=18\%$ error². Note that in practice, the estimation error is usually much smaller than the maximum error because OWS for SLSL and SLML tends to reduce the delay in a similar manner.

$T_{ows/SLSL}$ is available from the previous work of 2-pin nets in [2]. Using [2]'s model, and taking the constant term $R_d C_0$ into consideration, we have the following delay estimation model for the critical path of a multiple-pin net using OWS optimization.

$$\begin{aligned} T_{ows} &= R_d C_0 + [\alpha_1 l / W^2 (\alpha_2 l) + 2\alpha_1 l / W (\alpha_2 l) \\ &\quad + R_d c_f + \sqrt{R_d r c_a c_f l}] \cdot l \end{aligned} \quad (6)$$

where $\alpha_1 = \frac{1}{4} r c_a$, $\alpha_2 = \frac{1}{2} \sqrt{\frac{r c_a}{R_d C_L}}$, and $W(x)$ is Lambert's W function defined as the value of w that satisfies $w e^w = x$.

For the wire area estimation, it was shown in [7] that the average wire width of OWS can be estimated accurately from the best single-width w^* in (4). Then, we use the following simple formula to estimate the total wire area on the critical path.

$$A_{ows} = w_{avg} \cdot l = \sqrt{\frac{r(c_f l + 2C_L)}{2R_d c_a}} \cdot l \quad (7)$$

²This estimation is especially robust when wire length is shorter than the *critical length* for buffer insertion [2].

Figure 4 summarizes the delay and area estimation procedure for a single critical sink using optimal wire sizing. Figures 5 and 6 show the delay and average wire width comparisons from our model and from those by running OWS algorithm in TRIO package. For ease of illustration, we show the case with one branching capacitance C_1 , located at different positions along the critical path. Three critical paths of lengths $l = 5mm$, $10mm$, and $20mm$ are shown for the comparison, with l_1 from $0.1l$ to $0.9l$. We can see that for all (l, l_1) combinations, the delay and area estimations from our model match those from TRIO very well.

Input: Interconnect network with tree structure, and certain critical sink S_k
1. Compute C_1, C_2, \dots, C_k at each branch, using minimum wire width;
2. Compute C_L and C_0 using (1) and (2);
3. Estimate critical path delay using (6);
4. Estimate critical path area using (7).

Figure 4: The delay and area estimation for SCS with OWS.

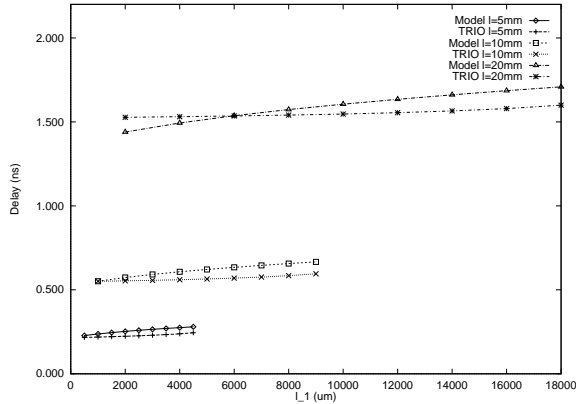


Figure 5: Delay comparison of our model with TRIO, using $0.18\mu m$ technology. $R_d = 180\Omega$, $C_1 = 100fF$, and $C_2 = 10fF$. TRIO uses 20 discrete wire width choices with maximum width of $20 \times W_{min}$, and wire segmentation of every $10\mu m$ (same for Figure 6).

In terms of run time, since we have the closed-form formula for the delay and area estimations, the run time for our model is constant. In fact, our estimation model is so fast that we have to call the estimation procedure many times using a loop to collect a measurable CPU time. The CPU time to run the estimation model 10,000 times (or equivalently, to estimate 10,000 nets) is just 0.8 second on a SUN UltraSPARC 1. However, using the efficient local refinement based OWS algorithm in TRIO for *one* net will take about 1 second. Therefore, our estimation model is an order of 10^4 faster! Note that this does not mean TRIO or other interconnect optimization algorithms are no longer needed. Our estimation model only gives the optimal delay and area

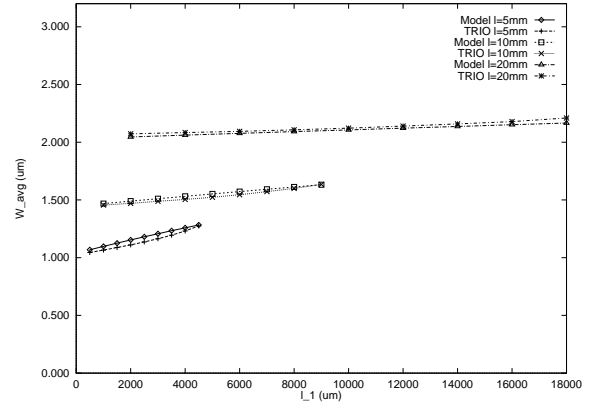


Figure 6: Average wire width comparison of our model with TRIO.

estimation to be used for high level synthesis and design planning (e.g., to screen out floorplanning candidates that cannot meet timing target even considering interconnect optimization), but not the sizing solution itself. TRIO or other interconnect optimization algorithms will still be needed to obtain the final optimal interconnect solution.

3.2 Buffer Insertion/Sizing and Wire Sizing for SCS

Optimizing single critical sink delay with BISWS can be formulated as a special case of the SLML problem by inserting minimum buffer at every branch on the critical path to shield all the downstream interconnect and device capacitances, as shown in Figure 7. For DSM designs, the gate capacitance c_g for minimum buffer is only about 0.1 to 0.5 fF, less than the interconnect capacitance of a $10\mu m$ short wire. Therefore, we can simply ignore these c_g 's for our delay estimation, and reduce the SLML problem into a SLSL problem, for which we can use the linear delay estimation model for BISWS developed in [2] to estimate the delay.

$$T_{bisws} = \tau_{bisws} \cdot l + t_g \quad (8)$$

where τ_{bisws} is the linear slope (see [2] for details). Note that this simple approach can also be used to estimate the best possible delay to *any* sink using BISWS. It will be useful to evaluate and screen out floorplanning and placement candidates.

Figure 8 shows the delay comparison using our model with running BISWS algorithm in TRIO package directly. The critical sink is $5mm$ to $20mm$ from the source. Our model has very accurate delay estimation. In terms of run time, our model is again extremely fast. The CPU time to run the model for 10,000 nets is just 8 seconds. However, using the bottom-up dynamic programming approach based BISWS in TRIO for *one* net will take about 14 seconds, using 10 different wire and buffer choices and wire segmentation in every $500\mu m$.

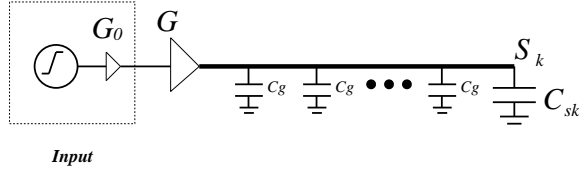


Figure 7: To estimate the best delay from source to sink S_k , we insert the minimum buffer at every branch on the critical path from source to sink S_k to shield the downstream capacitance at each branch.

So our estimation model is again an order of 10^4 times faster.

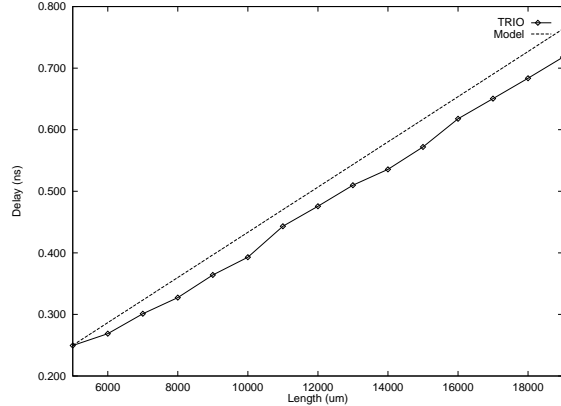


Figure 8: Delay comparison of our model with TRIO, using $0.18\mu\text{m}$ technology. $R_{d0} = r_g/10$, $C_{sk} = 10c_g$. TRIO uses 10 discrete wire width choices with maximum width being $20 \times W_{min}$, 10 buffer choices with maximum buffer size being $400 \times \text{min}$, and wire segmentation in every $500\mu\text{m}$.

4 Estimation for Multiple Critical Sinks (MCS)

In this section, we study interconnect delay estimation for multiple critical sinks under the optimization objective of minimizing the maximum delay of all critical sinks (i.e., the tree delay using [3]’s definition). To minimize the tree delay, [3] formulated it into a convex optimization problem and developed a sensitivity-based algorithm to solve it. The tree delay minimization can also be solved using weighted delay formulation through Lagrangian relaxation [8], or solved directly through bottom-up dynamic programming [9, 10]. In this work, we use the dynamic programming approach [10] implemented in TRIO package for the comparison with our estimation models.

4.1 Optimal Wire Sizing for MCS

Given a routing tree connecting multiple critical sinks, we have the following definitions.

Definition 1 An *internal critical sink* is a critical sink that is on the path from the source to another critical sink.

Definition 2 A *leaf critical sink* is a sink that is not on a path from the source to all other critical sinks.

The estimation for the optimal tree delay (i.e., the minimized maximum delay of all critical sinks) with MCS is much more difficult than the delay estimation with SCS because when we optimize the delay for one critical sink, it may affect all other critical sinks as well. That is why all optimization algorithms in [3, 8, 9, 10] used iterative-based approaches. However, we notice that there are some simple but very useful characteristics for the optimal tree delay. First, under the Elmore delay model, it can be easily shown that

Lemma 1 The critical sink that has the maximum delay from the source must be a leaf critical sink.

Now suppose we had performed OWS to a net minimizing the tree delay, then the pin-to-pin delay from source to any sink S_k must be larger than that by making S_k to be the *single* critical sink, and all other sinks to be non-critical (i.e., the SCS formulation). Since the tree delay is defined to be the maximum delay of all source-to-sink delays, we have the following theorem.

Theorem 2 The optimal delay to any critical sink under SCS formulation is a lower bound for the optimal tree delay.

From Theorem 2, we can obtain a lower bound estimation for the optimal tree delay by taking the maximum of all single critical sink delays, as shown in Figure 9.

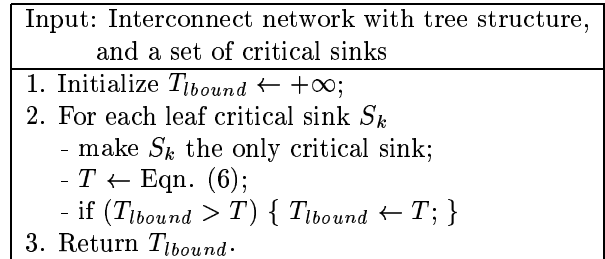


Figure 9: The lower bound delay estimation for the optimal tree delay using OWS.

Our experiments show that this lower bound delay estimation is indeed fairly tight and we can just use T_{lbound} to estimate the optimal tree delay. The explanation is as follows. Since our objective is to minimize the maximum delay, i.e., the delay to the most critical sink, we shall keep the wire load from less critical sinks as small as possible (but may not be too small; otherwise, they may become the most critical sink). To the most critical sink, the main difference between our model and the real optimal solution is that the former uses the minimum wire width to compute wire load while the latter uses ‘as small as possible’ widths to

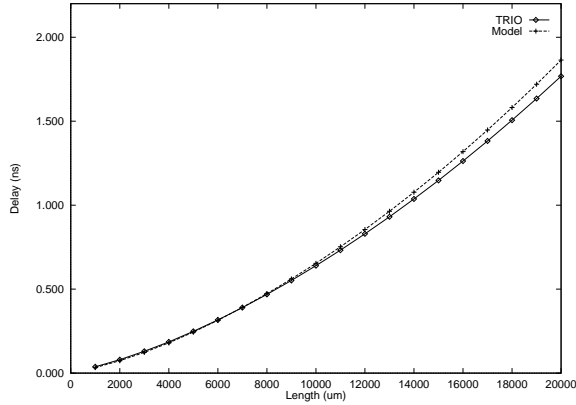


Figure 10: Comparison of our model with TRIO for optimal tree delay using OWS, under the $0.18 \mu\text{m}$ technology. $R_d = 180\Omega$, $C_s = 10fF$. The length from source to the maximum delay sink ranges from 1mm to 20mm. TRIO uses 10 discrete wire width choices with maximum width of $20 \times W_{min}$, and wire segmentation in every $500\mu\text{m}$.

compute wire load for non-critical paths. Meanwhile, for DSM designs, the area capacitance is usually dominated by effective-fringing capacitance [1]. Therefore, the two wire loads will not have much difference. Figure 10 shows the delay comparison of our model and TRIO for some random 4-pin nets using some typical parameters from $0.18\mu\text{m}$ technology. Our delay estimations match those from TRIO well. Note that for some lengths (e.g., $> 10,000\mu\text{m}$), our model, supposed to provide a tight lower bound, has slightly larger delay than that from TRIO. This is because our delay estimation model in (6) tends to have slightly more conservative delay estimation.

4.2 Buffer Insertion/Sizing and Wire Sizing for MCS

Similar to OWS, we find that the optimal tree delay under BISWS can be estimated by a tight lower bound delay from the leaf critical sink that has the maximum delay under the SCS formulation. Therefore, we can use our result in Section 3.2 to estimate the delay. Figure 11 shows the comparison of our model and TRIO. Again, our simple model gives accurate estimation for the optimal tree delay.

5 Concluding Remarks

This paper has developed fairly accurate yet extremely efficient interconnect delay/area estimation models for multiple-pin nets with consideration of two commonly used interconnect optimization techniques, OWS and BISWS, and under two optimization objectives, namely minimizing delay to single critical sink and minimizing the maximum delay to multiple critical sinks. Similar to [2], our OWS estimation model can be easily combined

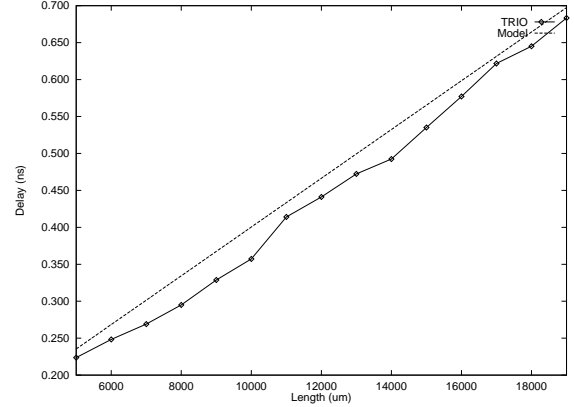


Figure 11: Comparison of our model with TRIO for optimal tree delay using BISWS, under the $0.18 \mu\text{m}$ technology. $R_{d0} = r_g/10$, $C_s = 10c_g$. TRIO uses 10 discrete wire width choices with maximum width being $20 \times W_{min}$, 10 buffer choices with maximum buffer size being $400 \times \text{min}$, and wire segmentation in every $500\mu\text{m}$.

with optimal driver size selection to perform simultaneous driver and wire sizing (SDWS) estimation.

We expect that these models will be very useful for interconnect-centric synthesis and design planning, e.g., performance-driven floorplanning, placement-driven synthesis and technology mapping, and interconnect planning.

Acknowledgments

The authors thank Lukas van Ginneken from Magma Design Automation for helpful discussion.

References

- [1] J. Cong, L. He, K.-Y. Khoo, C.-K. Koh, and Z. Pan, "Interconnect design for deep submicron ICs," in *Proc. Int. Conf. on Computer Aided Design*, pp. 478–485, 1997.
- [2] J. Cong and D. Z. Pan, "Interconnect delay estimation models for synthesis and design planning," in *Proc. Asia and South Pacific Design Automation Conf.*, Jan., 1999.
- [3] S. S. Sapatnekar, "RC interconnect optimization under the Elmore delay model," in *Proc. Design Automation Conf.*, pp. 387–391, 1994.
- [4] W. C. Elmore, "The transient response of damped linear networks with particular regard to wide-band amplifiers," *Journal of Applied Physics*, vol. 19, pp. 55–63, Jan. 1948.
- [5] L. Pileggi, "Timing metrics for physical design of deep submicron technologies," in *Proc. Int. Symp. on Physical Design*, pp. 28–33, 1998.
- [6] Semiconductor Industry Association, *National Technology Roadmap for Semiconductors*, 1997.
- [7] J. Cong and D. Z. Pan, "Interconnect estimation and planning for deep submicron designs," Tech. Rep. 980035, UCLA CS Dept., 1998.
- [8] C. P. Chen, Y. W. Chang, and D. F. Wong, "Fast performance-driven optimization for buffered clock trees based on Lagrangian relaxation," in *Proc. Design Automation Conf.*, pp. 405–408, 1996.
- [9] J. Lillis, C. K. Cheng, and T. T. Y. Lin, "Optimal wire sizing and buffer insertion for low power and a generalized delay model," in *Proc. Int. Conf. on Computer Aided Design*, pp. 138–143, Nov. 1995.
- [10] T. Okamoto and J. Cong, "Buffered Steiner tree construction with wire sizing for interconnect layout optimization," in *Proc. Int. Conf. on Computer Aided Design*, pp. 44–49, Nov. 1996.