Interconnect Delay Estimation Models for Synthesis and Design Planning *

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

Abstract

In this paper we develop a set of interconnect delay estimation models with consideration of various layout optimizations, including optimal wire-sizing (OWS), simultaneous driver and wire sizing (SDWS), and simultaneous buffer insertion/sizing and wire sizing (BISWS). These models have been tested on a wide range of parameters and shown to have about 90% accuracy on average compared with those from running complex optimization algorithms directly followed by HSPICE simulations. Moreover, our models run in constant time in practice. As a result, these simple, fast, yet accurate models are expected to be very useful for a wide variety of purposes, including layout-driven logic and high level synthesis, performance-driven floorplanning, and interconnect planning.

1 Introduction

In recent years, many interconnect optimization techniques, including wire sizing, driver sizing, buffer insertion and sizing, etc., have been proposed and shown to be very effective for interconnect delay reductions (e.g., [1]). However, in the current VLSI design flow, interconnect optimization is usually performed at late stages of the design process. Consequently, accurate interconnect delays, especially those for global interconnects are not known to higher level syntheses and design planning tools. Since interconnect optimization may improve interconnect delay by a factor of 5 to 6 times [1], it is less likely for synthesis and design planning tools to make correct decisions without proper modeling of the impact of interconnect optimization.

A brute-force integration that runs existing interconnect optimization algorithms directly at the synthesis and design planning levels will not be practical in designing complex deep submicron (DSM) circuits due to the following reasons:

- Inefficiency: Most interconnect optimization algorithms use either iterative local refinement operations or dynamic programming based approaches. Although they are efficient to optimize interconnects with respect to a given floorplan/placement, running them directly over tens of thousands of global nets is very costly to be used *repeatedly* by synthesis engines and/or design planning tools.
- Lack of abstraction: To make use of those optimization programs, a lot of detailed information is needed, such as the granularity of wire segmentation, number of wire widths and buffer sizes, etc. However, such information is usually not available at the synthesis and planning levels.
- Difficulty to interact synthesis engines with layout optimization tools.

To deal with these problems, we develop in this work a set of fast and accurate interconnect *delay estimation models* (DEM) with consideration of various optimization techniques, namely optimal wire sizing (OWS), simultaneous driver and wire sizing (SDWS), and buffer insertion, sizing and wire sizing (BISWS).

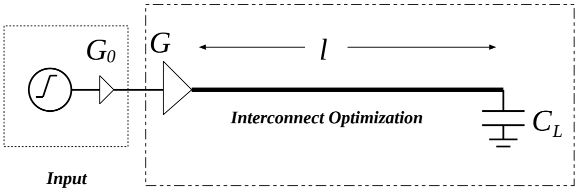

Figure 1: Problem formulation.

Our DEMs effectively overcome all the difficulties listed above: (i) they are very efficient (constant run time in practice), (ii) they provide high level abstraction and (iii) they can easily be embedded into synthesis and planning tools. Moreover, our DEMs provide explicit relation to enable design decision at high levels.

The rest of the paper is organized as follows. Section 2, states the problem formulation and parameters. Sections 3 to 5 present DEMs under OWS, SDWS and BISWS respectively, and compare them with HSPICE simulations after running corresponding optimization algorithms from UCLA Tree-Repeater-Interconnect-Optimization (TRIO) package [1]. Section 6 presents concluding remarks and possible applications of our models. Due to the length limitation, certain details are left in [2].

2 Problem Formulation and Parameters

The objective of our study is to quickly and accurately estimate interconnect delays with consideration of interconnect optimization. Fig. 1 shows such an interconnect wire of length l, driven by a gate G, and with loading capacitance C_L. G's input waveform is generated by a nominal gate G_0 connected with a ramp voltage input. The delay to be minimized is the overall delay from the input of G_0 to the load C_L, while the delay to be measured and estimated is the stage delay from the input of G to C_L, denoted as $T(G, l, C_L)$. The input stage delay is included so that it acts as a constraint not to over-size G during the interconnect optimization. Our goal is to develop simple closed-form formula and/or procedure to efficiently estimate $T(G, l, C_L)$ with consideration of various interconnect optimization techniques such as OWS, SDWS, and BISWS.

During interconnect optimizations, a long wire may be divided into a number of wire segments. Each wire segment is modeled by a π-type RC circuit and each buffer is modeled as a switch-level RC circuit [1]. The well-known Elmore delay model is used to guide the delay optimization and estimation. The following parameters are used by our estimation models.

- W_{min}: the minimum wire width, in μm
- S_{min}: the minimum wire spacing in μm
- r: the sheet resistance, in $\Omega/\Box$
- c_a: the unit area capacitance, in $fF/\mu m^2$
- c_f: the unit effective-fringing capacitance, in $fF/\mu m^1$,

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

[1] It is defined as the sum of fringing and coupling capacitances, as introduced in [3].

- 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 from *1997 National Technology Roadmap for Semiconductors* (NTRS'97) [4].

3 Delay Estimation Model under Optimal Wire-Sizing (OWS)

Proper wire sizing has been shown to be very effective in reducing interconnect delay (e.g., [5]). For OWS, the size of driver G in Fig. 1 is fixed. Let $T_{ows}(R_d, l, C_L)$ be the delay under OWS for an interconnect l with driver resistance R_d and loading capacitance C_L. Our comparison of discrete wire sizing (DWS) [5] and continuous wire shaping (CWS) [6] first shows that they have almost identical optimized delay (see [2] for details). We then perform extensive analytical and numerical studies on the complex optimal wire shaping function from [6] and obtain the following simple closed-form DEM under OWS.

$$\begin{aligned} T_{ows}(R_d, l, C_L) &= \Big(\alpha_1 l / W^2(\alpha_2 l) + 2\alpha_1 l / W(\alpha_2 l) \\ &\quad + R_d c_f + \sqrt{R_d r c_a c_f l}\Big) \cdot l \qquad (1) \end{aligned}$$

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 [6] defined as the value of w that satisfies $we^w = x$. Due to the length limitation, its justification is left in [2]. We can show that

Theorem 1 *T_{ows} is a sub-quadratic convex function of the interconnect length l.* □

Note that the wiring delay with uniform wire width (i.e., no OWS) is a quadratic function of l. The convexity of T_{ows} will be useful to perform optimal buffer insertion and wire sizing later on. We have tested our closed-form delay estimation model of (1) on a wide range of parameters. It matches the optimal delay very well from running TRIO package under OWS optimization, with about 90% accuracy on average. An example with typical interconnect parameters is shown in Fig. 2.

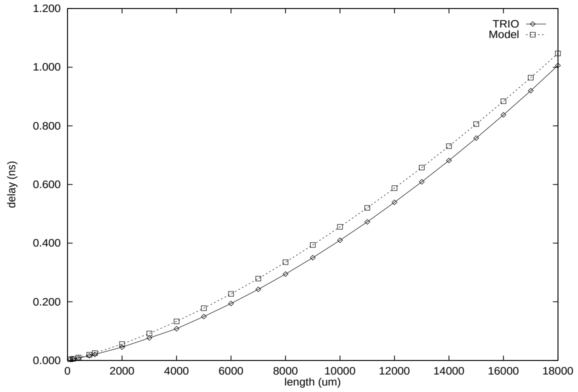

Figure 2: Comparison of our DEM with running TRIO for OWS under the 0.18 μm technology, with $R_d = r_g/100$, $C_L = c_g \times 100$. TRIO uses wire width set $\{W_{min}, 2W_{min}, \ldots, 20W_{min}\}$ and $10\mu m$-long segments (same for other figures).

4 Delay Estimation Model under Simultaneous Driver and Wire Sizing (SDWS)

This section presents the delay estimation model under SDWS, which sizes both wire and driver [7]. In our problem formulation, G_0 is fixed. But the driver G can be sized optimally to achieve the best performance from available driver set D. Denote R_{d0} and R_d to be the effective resistance of G_0 and G, and C_d to be the input capacitance of G. Suppose G's size is $k\times$ minimum gate. From the switch-level device model, we have $R_d = r_g/k$ and $C_d = kc_g$. Then the overall delay from the input of G_0 to C_L in Fig. 1 to be minimized is

$$\begin{aligned} T(k) &= (t_g + R_{d0} \cdot C_d) + t_g + T_{ows}(R_d, l, C_L) \\ &= (t_g + R_{d0} \cdot kc_g) + t_g + T_{ows}(r_g/k, l, C_L) \qquad (2) \end{aligned}$$

Note that the input stage delay $(t_g + R_{d0} \cdot C_d)$ is included for overall delay minimization but not in the one-stage delay estimation. Substitute the delay formula of T_{ows} from (1) and calculate the best driver size k^* that minimizes $T(k)$, we can obtain the following DEM under optimal SDWS:

$$T_{sdws}(D, l, C_L) = t_g + T_{ows}(r_g/k^*, l, C_L) \qquad (3)$$

To compute k^*, we set $dT(k)/dk = 0$, and compute its root. It can be solved efficiently by the bisection method [8]. Let ϵ_0 be the initial range that k^* lies in and ϵ be the error tolerance for k^*. Bisection method basically cuts the root search range by half at each iteration. So the number of iterations will be $log_2(\epsilon_0/\epsilon)$. In practice, $\epsilon_0 < 1000$ (determined by the maximum driver size) and $\epsilon \geq 1$ (minimum driver size), so ten or less iterations are usually sufficient for the root-finding. Therefore, k^* can be computed in constatn time.

Fig. 3 compares the delay from our estimation model and the optimal delay from running TRIO package under SDWS using the 0.18 μm technology. Our delay estimation model again matches TRIO very well, with over 90% accuracy on average.

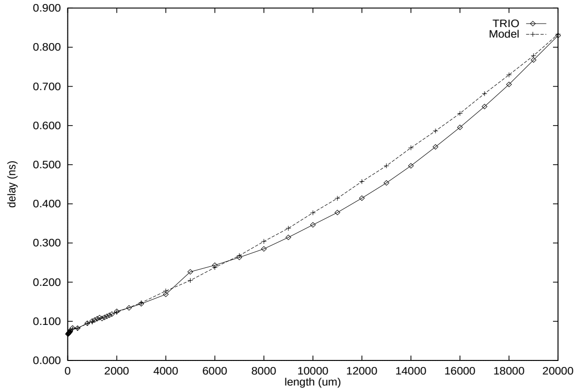

Figure 3: Comparison of our DEM with running TRIO for SDWS under $0.18\mu m$ tech., with G_0 and C_L of 10×min gate. Maximum driver for TRIO is set to be 200× min gate.

5 Delay Estimation Model under Buffer Insertion/Sizing and Wire Sizing (BISWS)

BISWS is a more powerful technique that can further reduce interconnect delay than SDWS by allowing buffer insertion to divide long wires into shorter ones. Dynamic programming based algorithms are often used for BISWS [9, 10]. However, they are not suitable for delay estimation. In this section, we will first introduce the concept of critical length for buffer insertion under OWS and give an analytical formula for it. Then we derive the DEMs for buffer insertion and wire sizing (BIWS, no buffer sizing), and for buffer insertion/sizing and wire sizing (BISWS).

5.1 Critical Length for Buffer Insertion under Optimal Wire Sizing

We first compute the longest length that a wire can run without the benefit from buffer insertion. Let $T_{1buf}(\alpha, R_d, l, C_L)$ denote the delay by inserting a buffer at the position of αl from the source ($0 \leq \alpha \leq 1$). Then

$$\begin{aligned} T_{1buf}(\alpha, R_d, l, C_L) &= T_{ows}(R_d, \alpha l, C_b) \\ &\quad + T_b + T_{ows}(R_b, (1-\alpha)l, C_L) \quad (4) \end{aligned}$$

is the delay after inserting the buffer and applying OWS to the two resulting wires separated by the buffer b with intrinsic delay of T_b, input capacitance of C_b and output resistance of R_b.

We can find the α that minimizes $T_{1buf}(\alpha, R_d, l, C_L)$ by solving the root of $dT_{1buf}/d\alpha = 0$ under $0 \leq \alpha \leq 1$, denoted as $\alpha^*(l)$. Then it is beneficial to insert such a buffer if and only if the resulting delay is smaller than the OWS delay, i.e.,

$$T_{1buf}(\alpha^*(l), R_d, l, C_L) < T_{ows}(R_d, l, C_L) \quad (5)$$

We define the *critical length* for inserting buffer b to be the minimum l that satisfies (5) and denote it as $l_{crit}(b, R_d, C_L)$.

Intuitively, when the wire length l is small, optimal wire sizing will achieve the best delay; whereas when the interconnect is long enough, the buffer insertion becomes beneficial. Thus, the root of l^* for the following equation

$$f(l) = T_{1buf}(\alpha^*(l), R_d, l, C_L) - T_{ows}(R_d, l, C_L) = 0 \quad (6)$$

gives the critical length for buffer insertion, i.e., $l_{crit}(b, R_d, C_L)$. Similar to SDWS, we use very fast binary search to obtain the root for Eqn. (6). Note that we need a two-level binary search for l^* and α^*. Let ϵ_{l0}, ϵ_l be the initial range and the error tolerance for l^*, and $\epsilon_{\alpha 0}$, ϵ_α be the initial range and the error tolerance for α^*. Then the root can be computed in $log_2(\epsilon_{l0}/\epsilon_l)$ iterations of l. For each l, we need another binary search for $\alpha^*(l)$, which takes $log_2(\epsilon_{\alpha 0}/\epsilon\alpha)$ steps. In practice, $\epsilon_{l0} = 2cm$, $\epsilon_l = 10\mu m$, $\epsilon_{\alpha 0} = 1$, and $\epsilon_\alpha = 0.01$ are usually sufficient for our delay estimation purpose, which leads to at most $log_2 2000 \times log_2 100 = 77$ steps for computing $l_{crit}(b, R_d, C_L)$. So in practice, $l_{crit}(b, R_d, C_L)$ can be computed in constant time.

In a recent work by [11], critical length concept was also introduced but on a *uniform-width* wire. An important observation from [11] is that l_{crit} is independent of buffer size. However, this is not the case for our l_{crit} where OWS is performed. As a comparison, Table 1 shows the critical lengths from the formula in [11] without OWS and from our formula with OWS using some typical buffer sizes. It is interesting to observe that:

1. In contrast to [11], our l_{crit} with OWS is no longer independent of buffer size. In fact, it tends to increase as buffer size gets larger. For example in $0.25\mu m$ technology, l_{crit} under 200× is $8.65mm$, more than the double of that under 10×, which is only $4.12mm$. Moreover, our l_{crit} with OWS is usually larger than that from [11] without OWS.
2. In general, l_{crit} decreases as technology further advances, which implies more buffers shall be used for performance optimization.
3. Although l_{crit} decreases as feature size scales down, this does not mean less logic cells can be reached by l_{crit}. We define the *logic volume* to be the number of 2-input minimum NAND gates that can be packed in the region spanned by the critical length, i.e. $\frac{1}{4}l^2_{crit}$. Table 2 shows that the logic volume actually increases due to the scaling down of devices.

5.2 Delay Estimation Model under Buffer Insertion and Wire Sizing (BIWS)

In this subsection, we derive the delay estimation model under optimal buffer insertion and wire sizing. We assume that all buffers (including the driver) are of the same given size. We prove that

Theorem 2 *For optimal BIWS solution to an interconnect wire, the distance between adjacent buffers is the same and equal to* $l_{crit}(b, R_b, C_b)$. □

Tech. (μm)	0.25	0.18	0.15	0.13	0.10	0.07
[11]	2.52	2.23	2.14	1.94	1.50	1.43
10×	4.12	3.80	3.97	3.61	2.92	2.08
50×	6.40	5.81	6.01	5.51	4.45	3.30
100×	7.47	6.83	7.04	6.39	5.30	3.91
200×	8.65	7.92	8.14	7.43	6.35	4.49
500×	9.98	9.10	9.30	8.57	7.13	5.21

Table 1: Critical length l_{crit} (in mm) for buffer insertion under uniform min wire width based on [11] and under our definition using OWS with some typical buffer sizes from 10× to 500× min gate.

Tech. (μm)	0.25	0.18	0.15	0.13	0.10	0.07
2-NAND (μm^2)	7.80	4.04	3.00	2.18	1.28	0.64
10×	0.55	0.89	1.31	1.49	1.66	1.69
50×	1.31	2.09	3.01	3.48	3.87	4.25
100×	1.79	2.88	4.13	4.68	5.48	5.97
200×	2.40	3.88	5.52	6.33	7.87	7.88
500×	3.19	5.12	7.21	8.42	9.93	10.6

Table 2: Logic volume (x10^6) in numbers of 2-input mininum NAND gates (area estimated based on NTRS'97) that can be packed in the square area of $\frac{l_{crit}}{2} \times \frac{l_{crit}}{2}$.

Note that previous works such as [12] and [11] also perform equally-spaced buffer insertion, but on uniform-width wires and without considering optimal wire sizing.

For simplicity, we denote $l_{crit}(b, R_b, C_b)$ as l_c. Then from Theorem 2 the total number of buffers (including the driver) will be $n_b = \lceil l/l_c \rceil$. They divide the original wire into n_b stages. Each stage has equal wire length of l_c and equal delay of $T_{crit} = t_g + T_{ows}(R_b, l_c, C_b)$ (defined as the *critical delay*), except the last one. Let the length of the last stage wire segment be l_{last}, then $l_{last} = l - (n_b - 1)l_c$, and the last stage delay is $T_{last} = t_g + T_{ows}(R_b, l_{last}, C_L)$. Therefore, the following accurate delay estimation model for BIWS is obtained:

$$\begin{aligned} T'_{biws} &= T_{crit} \cdot (n_b - 1) + T_{last} \\ &= \tau_{biws} \cdot (n_b - 1)l_c + T_{last} \quad (7) \end{aligned}$$

where τ_{biws} is given by the delay estimation model under OWS:

$$\begin{aligned} \tau_{biws} &= t_g/l_c + \alpha_1 l_c/W^2(\alpha_2 l_c) + 2\alpha_1 l_c/W(\alpha_2 l_c) \\ &\quad + R_b c_f + \sqrt{R_b r c_a c_f l_c} \quad (8) \end{aligned}$$

The model in (7) can be further approximated by the following linear model with respect to l, by ignoring the second order effects due to T_{last}.

$$T_{biws} = \tau_{biws} \cdot l + t_g \quad (9)$$

In practice, $l_c = l_{crit}(b, R_b, C_b)$ can be computed in constant time, which is also true for (7), (8) and (9). Thus, our estimation model under BIWS again takes only constant time. Fig. 4 shows the comparison of our DEMs with TRIO. Again, our DEM in (7) closely matches that from TRIO. The simple linear DEM in (9) approximates T_{last} by a linear interpolation of T_{crit}. It is accurate for long interconnects (longer than l_c), where the "bump" due to the T_{last} is negligible.

5.3 Delay Estimation Model under Buffer Insertion, Sizing and Wire Sizing (BISWS)

We observe from extensive TRIO experiments that a similar linear relationship between delay and length still holds for BISWS. Moreover, we observe that the internal buffers have about the same size and the adjacent buffers have about the same distance, mainly due to the internal symmetric structure. Thus the delay under BISWS can be estimated from the best BIWS solution.

$$T_{bisws} = \tau_{bisws} \cdot l + t_g \quad (10)$$

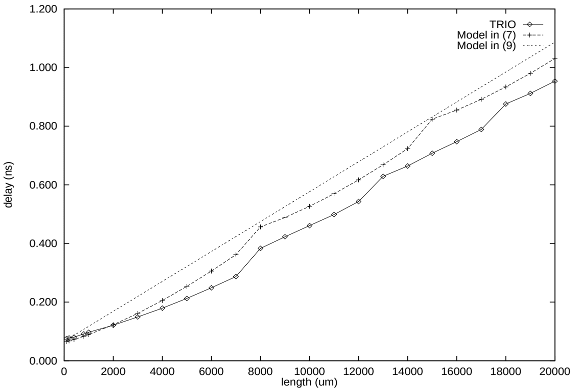

Figure 4: Comparison of DEM with TRIO under BIWS using 0.18 μm technology. G_0 and C_L are from 10× min. Buffer size is 100× min.

where $\tau_{bisws} = \min_{b \in B}\{\tau_{biws}\}$ from available buffer set B. In [13], the closed-form optimal BISWS solution *without fringing capacitance* was derived. We find that [13] as a special case of our BISWS, confirms our linear model. However, analytical justification of (10) remains open. The time complexity of the model is $O(|B|)$. Since $|B|$ is usually no more than 20, the BISWS model can also be considered to run in constant time for practical purpose. The results from the model and from running BISWS algorithm in TRIO package are shown in Fig. 5. The estimation model again achieves about 90% accuracy.

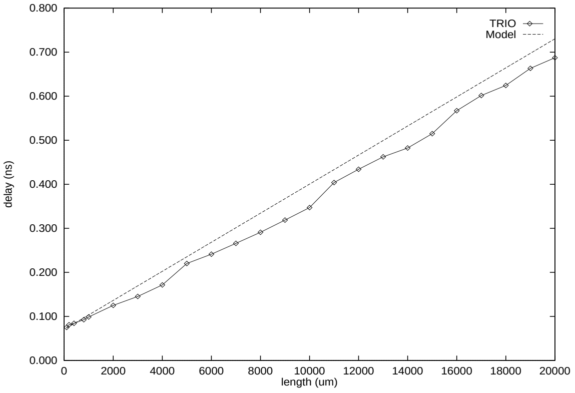

Figure 5: Comparison of our DEM and TRIO under BISWS using 0.18 μm technology. G_0 and C_L are from 10× min. 20 buffer choices are used from min to 400× min.

6 Conclusions and Applications

The main contribution of our work is a set of closed-form delay estimation models and very efficient computation procedures (constant time in practice) under various interconnect optimization techniques, such as OWS, SDWS, and BISWS, for both local wires (without buffer insertion) and global wires (with buffer insertion). They are shown to be very accurate and efficient compared with running complex interconnect optimization algorithms (e.g.,TRIO) directly. In addition, they can be easily embedded and coded into any synthesis engine and design planning tool.

We believe that these delay estimation models can be used in a wide spectrum of applications listed, but not limited, as follows:

- RTL and physical level floorplan: During the sizing and placement of functional blocks, our models can be used to accurately predict the impact on the performance of global interconnects.
- Placement-driven synthesis and mapping: A companion placement may be kept during synthesis and technology mapping [14]. For every logic synthesis operation, the companion placement will be updated. Once the cell positions are known, our DEMs can be used to accurately predict interconnect delay for the synthesis engine.
- Interconnect process parameter optimization: Interconnect parameters (e.g., metal aspect ratio, minimum spacing, etc.) may be tuned to optimize the delays predicted by our models for global, average, and local interconnects under certain wire-length distributions.
- Interconnect Planning: our models can also be used to evaluate different optimization alternatives and to plan routing and silicon resources beforehand for interconnect layout optimization.

In the future, we plan to extend our work to multiple-pin nets and investigate the delay/area/power tradeoffs.

Acknowledgments

This research makes use of the software donated from Avant! Corporation, whose generous donation is greatly appreciated. The authors would also like to thank D. F. Wong, C.-P. Chen, and Y. Gao from U.T. Austin for providing continuous wire sizing program, C.-K. Koh from Purdue Univ., M. K. Mohan from Intel, Lukas van Ginneken from Magma Design Automation, and L. He, K.-Y. Khoo, and D. Xu from UCLA for their helpful discussions.

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 Z. Pan, "Interconnect performance estimation models for synthesis and design planning," Tech. Rep. 980018, UCLA CS Dept, 1998.

[3] J. Cong, L. He, C.-K. Koh, and Z. Pan, "Global interconnect sizing and spacing with consideration of coupling capacitance," in *Proc. Int. Conf. on Computer Aided Design*, pp. 628–633, 1997.

[4] Semiconductor Industry Association, *National Technology Roadmap for Semiconductors*, 1997.

[5] J. Cong and K. S. Leung, "Optimal wiresizing under the distributed Elmore delay model," in *Proc. Int. Conf. on Computer Aided Design*, pp. 634–639, 1993.

[6] C.-P. Chen and D. F. Wong, "Optimal wire sizing function with fringing capacitance consideration," in *Proc. Design Automation Conf*, pp. 604–607, 1997.

[7] J. Cong and C.-K. Koh, "Simultaneous driver and wire sizing for performance and power optimization," *IEEE Trans. on Very Large Scale Integration (VLSI) Systems*, vol. 2, pp. 408–423, Dec. 1994.

[8] W. H. Press, S. Teukolsky, W. Vetterling, and B. Flannery, *Numerical Recipes in FORTRAN–The Art of Scienctfic Computing*. Cambridge University Press, 1992.

[9] L. P. P. P. van Ginneken, "Buffer placement in distributed RC-tree networks for minimal Elmore delay," in *Proc. IEEE Int. Symp. on Circuits and Systems*, pp. 865–868, 1990.

[10] 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.

[11] R. Otten, "Global wires harmful?," in *Proc. Int. Symp. on Physical Design*, pp. 104–109, Apr. 1998.

[12] C. J. Alpert and A. Devgan, "Wire segmenting for improved buffer insertion," in *Proc. Design Automation Conf*, 1997.

[13] C. C. N. Chu and D. F. Wong, "Closed form solution to simultaneous buffer insertion/sizing and wire sizing," in *Proc. Int. Symp. on Physical Design*, pp. 192–197, 1997.

[14] M. Pedram, N. Bhat, and E. Kuh, "Combining technology mapping and layout," *The VLSI Design: An Int'l Journal of Custom-Chip Design, Simulation and Testing*, vol. 5, no. 2, pp. 111–124, 1997.