Interconnect Delay Estimation Models for Logic and High Level Synthesis

Jason Cong and David Z. 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 delay estimation models with consideration of various interconnect optimization techniques, 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 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

As VLSI circuit design advances to deep sub-micron (DSM) technologies, interconnect delay has become the dominating factor for circuit performance. 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 very late stages. Consequently, accurate interconnect delay, especially that for global interconnects is not known to logic/high level syntheses. Since interconnect optimization may improve interconnect performance by a factor of 5 to 6 times [1], it is less likely for synthesis engines to make correct decision without proper modeling of the impact of interconnect optimization.

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

- Inefficiency: Most interconnect optimization algorithms use either iterative local refinement operations or dynamic programming based approaches, which are too costly to be used repeatedly by synthesis engines.
- 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 during the synthesis level.
- 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) under various optimization techniques, namely optimal wire sizing (OWS), simultaneous driver and wire sizing (SDWS), and buffer insertion, sizing and wire sizing (BISWS). Our DEMs effectively overcome all the difficulties listed above: (i) their running time is very low (constant time in practice), (ii) they provide high level abstraction and (iii) they can easily be embedded into synthesis tools. Moreover, our DEMs provide explicit relation to enable design decision at high levels.

*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.

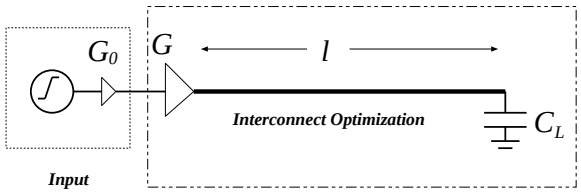

Figure 1: An interconnect wire of length l and loading capacitance of C_L. It is driven by a gate G with input waveform provided by the nominal gate G_0, which is connected with a ramp voltage input.

The rest of the paper is organized as follows. Section 2 states the problem formulation and parameters used for our study. Sections 3 to 5 present the 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.

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 to be considered. It is driven by a gate G, and has 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 notations of key parameters are listed below.

- W_{min}: the minimum wire width, in μm
- S_{min}: the minimum wire spacing in μm
- r: the sheet resistance, in $\Omega/\square$
- c_a: the unit area capacitance, in $fF/\mu m^2$
- c_f: the unit fringing capacitance, in $fF/\mu m$
- t_g: the intrinsic device delay in ps
- c_g: input capacitance of a minimum device, in fF

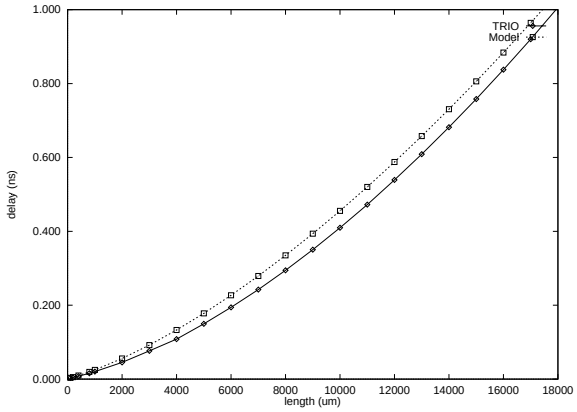

Figure 2: Comparison of DEM with running TRIO under OWS using the 0.18 μm technology. $R_d = r_g/100$, $C_L = c_g \times 100$.

- r_g: output resistance of a minimum device, in $k\Omega$

The values of these parameters for our study are shown in Table 1. They are based on the *1997 National Technology Roadmap for Semiconductors* (NTRS'97) [2].

Tech. (μm)	0.25	0.18	0.15	0.13	0.10	0.07
W_{min}	0.25	0.18	0.15	0.13	0.10	0.07
S_{min}	0.34	0.24	0.21	0.17	0.14	0.10
r	0.073	0.068	0.073	0.081	0.092	0.095
c_a	0.059	0.060	0.054	0.046	0.053	0.056
c_f	0.082	0.064	0.054	0.043	0.045	0.040
t_g	86.6	66.4	65.5	54.4	50.1	29.8
c_g	0.282	0.234	0.220	0.135	0.072	0.066
r_g	16.2	17.1	17.3	22.1	23.4	22.1

Table 1: Parameters based on NTRS'97.

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

In this section, we present the delay estimation model under OWS. Proper wire sizing has been shown to be very effective to reduce interconnect delay (e.g., [3]). For OWS, the size of driver G in Fig. 1 is fixed. Let R_d be the effective resistance of G, $T_{ows}(R_d, l, C_L)$ be the delay under OWS for an interconnect l with driver resistance R_d and loading capacitance C_L. We have performed extensive analytical and numerical studies on the complex optimal wire shaping function [4] and obtained the following simple DEM under OWS. Due to the length limitation, its justification is left in [5].

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

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 [4] defined as the value of w that satisfies $we^w = x$. From the definition of the convex function and the characteristics of the W function, we can easily show that

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

We have tested the 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. In these experiments, we have wire width set being $\{W_{min}, 2W_{min}, \ldots, 20W_{min}\}$ and the wire is segmented into $10\mu m$-long segments.

Delay Estimation Model under SDWS

Input: R_{d0}, l, C_L, c_a, c_f, r, and driver set D of size between $[k_{min}, k_{max}]$

1. Calculate the best driver size k_{opt} that minimize $T(k)$ of Eqn. (2)
   - calculate the root k^* of $dT(k)/dk = 0$
   - **if** $k_{min} < k^* < k_{max}$, k_{opt} is one of $\lfloor k^* \rfloor$ or $\lceil k^* \rceil$ which gives smaller $T(k)$
   - **else** k_{opt} is one of k_{min} or k_{max} which gives smaller $T(k)$
2. Compute T_{sdws} using Eqn. (3)

Figure 3: The delay estimation model under SDWS.

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

This section presents the delay estimation model under SDWS, which sizes both the wires and the driver [6]. To obtain an accurate delay estimation model, we use the OWS delay estimation model from the previous section. Note that 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 device. 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

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

Note that the input stage delay $(t_g + R_{d0} \cdot C_d)$ is included for overall delay minimization. Substitute the delay formula of T_{ows} from (1) and calculate the best driver size k_{opt} that minimizes $T(k)$, we can obtain the delay estimation under optimal SDWS,

$$T_{sdws}(D, l, C_L) = t_g + T_{ows}(r_g/k_{opt}, l, C_L) \tag{3}$$

Recall that we do not include the input stage delay in our delay estimation. The delay estimation model under SDWS is outlined in Fig. 3. To solve the root k^* of $dT(k)/dk = 0$, we can use the efficient numerical approach such as bisection method [7]. 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, $\epsilon_0 < 1000$ and $\epsilon \geq 1$, so ten or less iterations are usually sufficient for the root-finding. Therefore, the procedure in Fig. 3 runs in constant time.

Fig. 4 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 matches the experimental results very well, with over 90% accuracy on the average.

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 [8, 9]. 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 a delay estimation model for buffer insertion and wire sizing (BIWS, no buffer sizing), and for buffer insertion, sizing and wire sizing (BISWS).

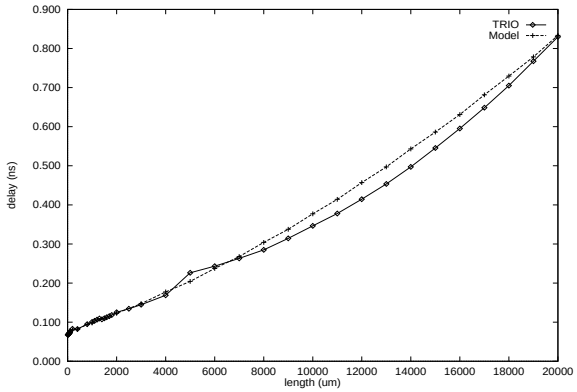

Figure 4: Comparison of DEM with running TRIO under SDWS. G_0 and C_L are from 10×min device.

5.1 Critical Length for Buffer Insertion under Optimal Wire Sizing

Given the delay estimation model $T_{ows}(R_d, l, C_L)$ in (1), we can analyze the longest wire that can run without the benefit from buffer insertion. For a buffer b with intrinsic delay of T_b, input capacitance of C_b and output resistance of R_b, denote $T_{1buf}(\alpha, R_d, l, C_L)$ to be the delay by inserting one buffer at the position of αl from the source ($0 \leq \alpha \leq 1$). Then

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

is the delay after inserting the buffer with OWS in the resulting two wires before and after the buffer. 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 α_{opt}. Then it is beneficial to insert such a buffer if and only if the resulting delay is smaller than the OWS delay, i.e.,

$$T_{1buf}(\alpha_{opt}, R_d, l, C_L) < T_{ows}(R_d, l, C_L) \quad (5)$$

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

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

gives the critical length for buffer insertion, i.e., $l_{crit}(b, R_d, C_L)$. The critical length computation procedure is outlined in Fig. 5. Similar to SDWS, we use very fast bisection method [7] to obtain the root for Eqn. (6). Let ϵ_0 be the initial root range and ϵ be the error tolerance for l_{crit}, the root can be computed in $log_2(\epsilon_0/\epsilon)$ iterations in step 2. In practice, a conservative estimation of $\epsilon_0 = 2cm$ and a error tolerance of $\epsilon = 10\mu m$ are usually sufficient for our delay estimation purpose, which leads to 11 or less iterations for computing $l_{crit}(b, R_d, C_L)$. Therefore, in practice, the procedure in Fig. 5 also runs in constant time.

Table 2 shows the critical lengths for some typical buffer sizes under various NTRS'97 technology generations. It is interesting to observe that: (i) As l_{crit} decreases and chip size increases [2], more buffers shall be inserted for performance optimization as the technology scales down. Therefore, for DSM circuit designs, a long interconnect is no longer a simple metal line but indeed a complex circuitry and needed to be planned carefully! (ii) l_{crit} under OWS tends to increase as buffer size gets larger.

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

Procedure to Compute $l_{crit}(b, R_d, C_L)$

Input: R_d, C_L, and buffer b with characteristics of R_b, C_b and T_b

/* Use bisection (binary search) method */
1. initialize l_{crit}'s range $[l_{min}, l_{max}]$, where $f(l_{min}) > 0$ and $f(l_{max}) < 0$
2. **while** $l_{max} - l_{min} > \epsilon$
   $l_{mid} \leftarrow (l_{min} + l_{max})/2$
   if $f(l_{mid}) > 0$, $l_{min} \leftarrow l_{mid}$
   else $l_{max} \leftarrow l_{mid}$
3. return l_{mid}

Figure 5: The procedure to compute critical length for buffer insertion.

Tech. (μm)	0.25	0.18	0.15	0.13	0.10	0.07
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 2: Critical length l_{crit} (in mm) for buffer insertion under OWS with some typical buffer sizes from 10× to 500× min gate.

(including the driver) are of the same given size. We prove that

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

For simplicity, we denote $l_{crit}(b, R_b, C_b)$ as l_c. Then 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:

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

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

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

The model in (7) can be approximated by the following linear model with respect to l, which is usually accurate enough for delay estimation purpose.

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

The delay estimation model under BIWS is summarized in Fig. 6. In practice, $l_c = l_{crit}(b, R_b, C_b)$ can be computed in constant time. (7), (8) and (9) can also be computed easily in constant time, so our estimation model under BIWS again takes only constant time. The model in either (7) or (9) gives very good matches compared with running TRIO as shown in Fig. 7, having about 90% accuracy.

Delay Estimation Model under BIWS

Input: R_{d0}, l, C_L, c_a, c_f, r, and buffer b

1. Compute $l_c = l_{crit}(b, R_b, C_b)$
2. Compute τ_{biws} using Eqn. (8)
3. Compute T'_{biws} using Eqn. (7) or T_{biws} using Eqn. (9)

Figure 6: The delay estimation model under BIWS.

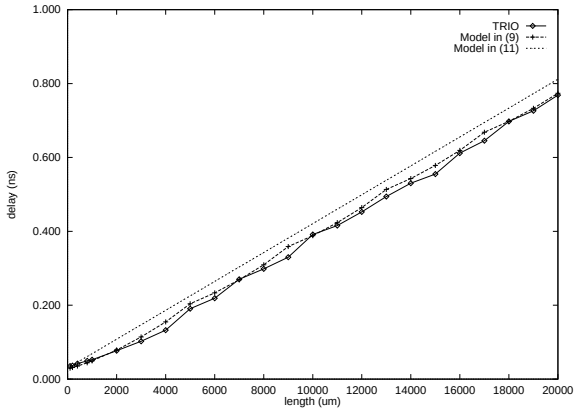

Figure 7: Comparison of DEM with TRIO under BIWS using 0.07 μm technology. G_0 and C_L are from 10× min device.

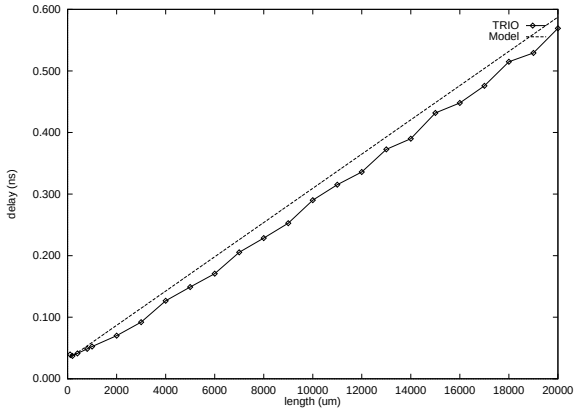

Figure 8: Comparison of DEM and TRIO under BISWS using 0.07 μm technology. G_0 and C_L are from 10× min device.

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

The BISWS technique is the most complete and powerful optimization to reduce delay for global interconnects. We observe from extensive TRIO experiments that a 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. Therefore, the delay under BISWS can be estimated from the best BIWS solution.

$$T_{bisws} = \tau_{bisws} \cdot l + t_g \tag{10}$$

where $\tau_{bisws} = \min_{b \in B}\{\tau_{biws}\}$ from available buffer set B. 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. 8. The estimation model again closely matches the experimental results.

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 match the experimental results very well (with about 90% accuracy on average) and run extremely fast 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:

- Layout-driven synthesis and mapping. One may keep a companion placement during synthesis and technology mapping [10]. For every logic synthesis operation, the companion placement will be updated. Once the cell positions are known, one can use our DEMs to accurately predict interconnect delay for the synthesis engine.
- RTL and physical level floorplan: During the sizing and placement of functional blocks, one can use our models to accurately predict the impact on the performance of global interconnects.
- 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: (i) evaluate different optimization alternatives; (ii) plan routing and silicon resource beforehand for interconnect layout optimization.

We plan to extend our interconnect performance estimation models to handle nets with multiple-pin topology in the future. Also, we plan to look at the performance estimation models under area/power constraints.

Acknowledgments

The authors would like to thank Prof. D. F. Wong, C.-P. Chen, C. Chu, and Y. Gao from U.T. Austin, M. K. Mohan from Intel, and L. He, K.-Y. Khoo, and C.-K. Koh 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] Semiconductor Industry Association, *National Technology Roadmap for Semiconductors*. 1997.

[3] 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.

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

[5] J. Cong and D. Z. Pan, "Interconnect performance estimation models for synthesis and design planning," Tech. Rep. 980017, UCLA CS Dept, 1998.

[6] 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.

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

[8] 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.

[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] 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.