# Interconnect Performance 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**

The objective of this work is to provide simple, efficient, yet reasonably accurate interconnect performance estimation models for synthesis and design planning under various complex interconnect optimization techniques. We have developed a set of closed-form delay estimation models as functions of interconnect length as well as some other key interconnect and device parameters with the consideration of various interconnect optimization techniques, which include 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 when compared with the delays obtained by running complex optimization algorithms directly followed by HSPICE simulations. Moreover, our models run in constant time for all practical purposes. 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 has become the dominating factor in determining the overall circuit performance. As a result, it impacts all aspects of the design flow. In recent years, many interconnect optimization techniques, including interconnect wire sizing, driver sizing, buffer insertion and sizing, etc., have been proposed and shown to be very effective for interconnect delay reductions (see [1, 2] for comprehensive surveys). For example, it was shown in [3] that for a 2 cm global interconnect in the 0.07 $\mu m$ technology generation in the *1997 National Technology Roadmap for Semiconductors* (NTRS'97) [4], applying the optimization technique by simultaneous driver sizing, buffer insertion, buffer sizing, and wire sizing can significantly reduce delay by a factor of 5 to 6$\times$ when compared with nominal wire-sizing.

However, in the current VLSI design flow, interconnect optimization is usually done at very late stages. Consequently, accurate interconnect performance, especially that for global interconnects is not known to logic/high level syntheses and floorplanning tools. Since interconnect optimization may improve interconnect performance significantly, it is less likely for synthesis and design planning tools 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 and design planning levels may 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. Although shown to be polynomial in time complexity, they are still too costly to be used repeatedly by logic and high level synthesis engines and/or design planning tools.

*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.

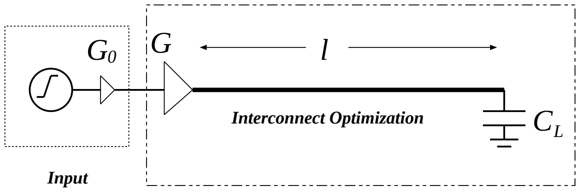


Figure 1: An interconnect wire of length $l$ and loading capacitance of $C_L$. It is driven by a gate $G$ with input waveform provided by the nominal gate $G_0$, which is connected with a ramp voltage input.

- 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 and planning 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 under various optimization techniques, namely optimal wire sizing (OWS), simultaneous driver and wire sizing (SDWS), and buffer insertion, sizing and wire sizing (BISWS). Note that most previous layout-driven logic/high level synthesis works, such as [5, 6], often use over-simplified models which assume that the interconnect delay is proportional to the square of wire length, without considering the interconnect optimization. We observe that this quadratic delay model (based on some nominal wire width) can be 5× larger than the final optimized delay [2, 3]. Therefore, it is no longer accurate to guide the synthesis tools in deep submicron designs.

Our closed-form delay estimation models overcome all three difficulties listed above: (i) their running time is very low (in constant time in practice), mainly by the complexity of finding the root of some non-linear equations; (ii) they provide abstraction and do not need detailed information such as wire segmentation, number of wire width, etc.; (iii) they can be easily embedded into any synthesis engine. More detailed discussion of applications of our performance estimation models will be given in Section 6.

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 delay estimation models under OWS, SDWS and BISWS respectively, and compare them with HSPICE simulations after running corresponding optimization algorithms using the UCLA Tree-Repeater-Interconnect-Optimization (TRIO) package [2]. 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 during the optimization so that it acts as a constraint to avoid over-sizing $G$ in the optimization of the stage delay $T(G, l, C_L)$. Our goal is to develop simple closed-form formula and procedure to efficiently estimate $T(G, l, C_L)$ with consideration of various interconnect optimization techniques such as OWS, SDWS and BISWS.

A long wire may be divided into a number of segments during the interconnect optimization. A wire segment is then modeled as a $\pi$-type resistor-capacitor (RC) circuit and a buffer is modeled as a switch-level RC circuit (e.g., see [1] for details). Therefore, it will form a distributed RC tree. The well-known Elmore delay model [7] is used to guide the delay optimization and estimation.

The notations of key parameters are listed below.

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

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) [4] and derived in [3]. In the table, the interconnect capacitances are obtained using a 3D capacitance solver FASTCAP [9] for an interconnect wire of 5× minimum width and with two parallel neighboring wires of 5× minimum spacing. The device used for our study is a buffer, made up of two cascaded inverters with stage ratio of 1:5. The device parameters are obtained through HSPICE simulations (with minor adjustments to the data in [3]).

| Tech. ($\mu m$) | 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 |
| $r$ | 0.0733 | 0.0679 | 0.0733 | 0.0806 | 0.0917 | 0.0952 |
| $c_a$ | 0.0589 | 0.0596 | 0.0542 | 0.0461 | 0.0531 | 0.0558 |
| $c_f$ | 0.0819 | 0.0641 | 0.0538 | 0.0433 | 0.0448 | 0.0404 |
| $t_g$ | 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: Interconnect and device 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. It was first shown in [10, 11] that when wire resistance becomes significant, as in deep sub-micron designs, proper wire-sizing can effectively reduce the interconnect delay. Assuming that each wire has a set of discrete wire widths, their work presented an optimal discrete wire-sizing (DWS) algorithm for a single-source RC interconnect tree to minimize the sum of weighted delays from the source to timing-critical sinks under the Elmore delay model. The algorithm is very efficient by using the technique of iterative local refinement to compute lower and upper bounds of the optimal wire widths. The lower and upper bounds usually meet, which leads to an optimal wire-sizing solution. It was extended to optimize a routing tree with

[1] It is defined as the sum of the fringing and coupling capacitances, as introduced in [8].

multiple sources [12], without a priori segmentation of long wires [12], and to minimize the maximum delay using Lagrangian relaxation [13]. An alternative approach to perform wire-sizing optimization is through bottom-up dynamic programming [14] and it can be combined easily with routing tree construction and buffer insertion [15]. Later on, optimal continuous wire sizing (CWS) for a wire segment was studied. Closed-form wire shaping functions was obtained to minimize the Elmore delay, first without fringing capacitance [16, 17], later on with fringing capacitance [18, 19], and recently for a bi-directional wire [20].

Our detailed study first shows very close matches between CWS and DWS in terms of the resulting delays after optimization (see Appendix for details). Therefore, in the following discussions, we will just use OWS for either CWS or DWS. 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 $C_L$. For given technology, the interconnect parameters of $r$, $c_a$, and $c_f$ are fixed, thus are not included in the $T_{ows}$ notation. We have performed extensive analytical and numerical studies on the original complex wire shape and delay functions under CWS [18, 20] and extracted the key terms that contribute to the optimal OWS delay as follows.

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

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 [18] defined as the value of $w$ that satisfies $we^w = x$. The justification for Eqn. (1) can be found in Appendix. When the fringing capacitance is zero, it consists of the first two terms and degenerates into the closed-form optimal Elmore delay formula under CWS without fringing capacitance, as shown below:

$$T_{ows}(R_d, l, C_L)|_{c_f=0} = \alpha_1 l^2 / W^2(\alpha_2 l) + 2\alpha_1 l^2 / W(\alpha_2 l) \tag{2}$$

The last two terms in Eqn. (1) are the adjustment terms when considering the fringing capacitance.

$W(x)$ function is shown in Fig. 2. Notice that for $W(x)$ to be valid, $x$ must be larger or equal to $-1/e$. From $W(x)e^{W(x)} = x$, we can take the differential to $x$

$$W'(x)e^{W(x)} + W(x)e^{W(x)}W'(x) = 1$$

Then, we have

$$\begin{aligned} W'(x) &= \frac{1}{e^{W(x)} + W(x)e^{W(x)}} \\ &= \frac{1}{x/W(x) + x} \\ &= \frac{W(x)}{x\left[1 + W(x)\right]} \end{aligned} \tag{3}$$

**Definition 1** *[21] A function $f$ defined on a convex set $\Omega$ is said to be* convex *if, for every $x_1, x_2 \in \Omega$ and every $\alpha$, $0 \leq \alpha \leq 1$, there holds*

$$f(\alpha x_1 + (1-\alpha)x_2) \leq \alpha f(x_1) + (1-\alpha) f(x_2).$$

Geometrically, a function is convex if the line joining two points on it graph lies nowhere below the graph. It is well known that a necessary and sufficient condition for a function $f(x)$ to be convex is that $f''(x) > 0$. Then from the characteristics of the $W$ function, we have the following two lemmas.

**Lemma 1** *Let $f(x) = \frac{x^2}{W(x)}$ $(x > 0)$. Then $f(x)$ is a convex function.*

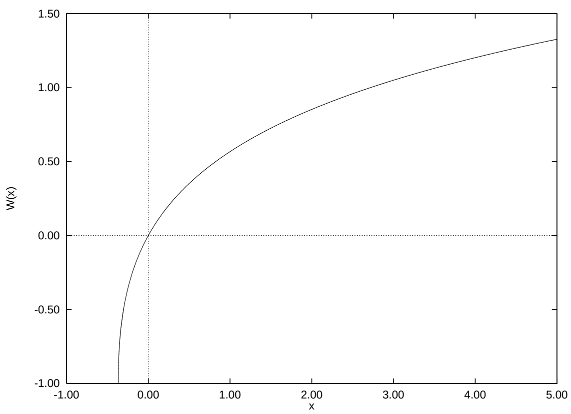


Figure 2: Euler's Lambert function.

**Proof:** Take the differential of $f(x)$ w.r.t. $x$,

$$\begin{aligned} f'(x) &= \frac{2xW - x^2W'}{W^2} \\ &= \frac{2xW - x^2\frac{W}{x(W+1)}}{W^2} \\ &= \frac{x(2W+1)}{W(W+1)} \end{aligned} \tag{4}$$

Then,

$$\begin{aligned} f''(x) &= \frac{(2W+1+x\cdot 2W')\cdot W(W+1) - x(2W+1)\cdot(2W+1)W'}{[W(W+1)]^2} \\ &= \frac{(2W+1+x\cdot 2\frac{W}{x(W+1)})\cdot W(W+1) - x(2W+1)\cdot(2W+1)\frac{W}{x(W+1)}}{[W(W+1)]^2} \\ &= \frac{(2W+1)W(W+1)^2 + 2W^2(W+1) - W(2W+1)^2}{W^2(W+1)^3} \\ &= \frac{(2W+1)[W(W+1)^2 - W(2W+1)] + 2W^2(W+1)}{W^2(W+1)^3} \\ &= \frac{(2W^2+3W+2)}{(W+1)^3} \\ &> 0 \end{aligned} \tag{5}$$

Therefore, $f(x)$ is a convex function. $\square$

**Lemma 2** *Let* $g(x) = \frac{x^2}{W^2(x)}$ *($x > 0$). Then $g(x)$ is a convex function.*

**Proof:** Take the differential of $g(x)$ w.r.t. $x$,

$$\begin{aligned} g'(x) &= \frac{2xW^2 - x^2 2WW'}{W^4} \\ &= \frac{2xW(W - x\cdot\frac{W}{x(W+1)})}{W^4} \end{aligned}$$

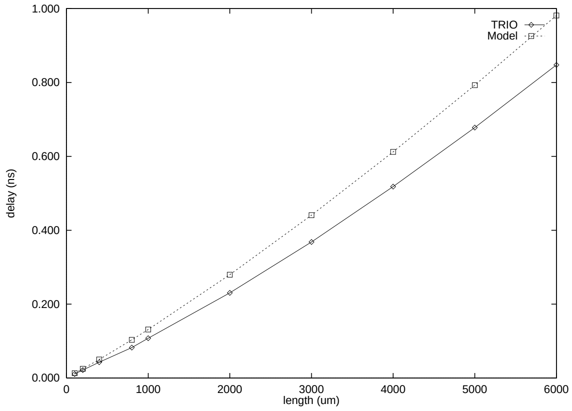


Figure 3: Comparison of the delay estimation model with running TRIO under OWS using 0.18 $\mu m$ technology. $R_d = r_g/10$, $C_L = c_g \times 10$.

$$= \frac{2x}{W(W+1)} \tag{6}$$

Then,

$$\begin{aligned} g''(x) &= \frac{2W(W+1) - 2x \cdot W'(2W+1)}{[W(W+1)]^2} \\ &= \frac{2W(W+1) - 2\frac{W}{W+1}(2W+1)}{[W(W+1)]^2} \\ &= \frac{2W}{(W+1)^3} \\ &> 0 \end{aligned} \tag{7}$$

Therefore, $g(x)$ is a convex function. □

**Proposition 1** *$T_{ows}$ is a sub-quadratic convex function with respect to the interconnect length $l$.*

**Proof:** It is obvious that $T_{ows}$ is sub-quadratic, since all the terms $l^2/W^2(l)$, $l^2/W(l)$, $l$ and $l^{3/2}$ are sub-quadratic. We can easily show that both $l$ and $l^{3/2}$ are convex function of $l$ by taking the second-order derivatives. From Lemma 1 and Lemma 2, $l^2/W(l)$ and $l^2/W^2(l)$ are also convex functions of $l$. Since the positive linear combination of convex functions are still convex [21], we have the above proposition. □

This characteristic of $T_{ows}$ will be useful to perform optimal buffer insertion and wire sizing (BIWS) in Section 5.

We have tested the closed-form delay estimation model in Eqn. (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. The model usually provides more conservative delay than that from the optimization program. Some examples with typical interconnect parameters are shown in Figs. 3 and 4. 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.

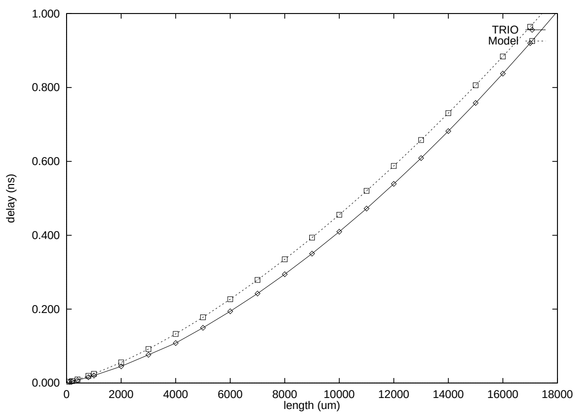


Figure 4: Comparison of the delay estimation model with running TRIO under OWS using the 0.18 $\mu m$ technology. $R_d = r_g/100$, $C_L = c_g \times 100$.

# 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. The SDWS problem was first studied in [22]. It was shown that the dominance property presented in [11] still holds for SDWS problem and the local refinement operations (as used by OWS) can be used iteratively to compute tight lower and upper bounds of the optimal widths for the driver and wires efficiently. However, it has not been possible to develop a closed-form solution from the local refinement based iterative algorithm.

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

$$\begin{aligned} T(k) &= (t_g + R_{d0} \cdot C_d) + t_g + T_{ows}(R_d, l, C_L) \\ &= (t_g + R_{d0} \cdot kc_g) + t_g + T_{ows}(r_g/k, l, C_L) \end{aligned} \tag{8}$$

under the constraint of given driver choices. 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,

$$T_{sdws}(D, l, C_L) = t_g + T_{ows}(r_g/k_{opt}, l, C_L) \tag{9}$$

Recall that we do not include the input stage delay in our delay estimation. The simple delay estimation procedure under SDWS is outlined in Fig. 5. To solve the root $k^*$ of $dT(k)/dk = 0$, we can use the efficient numerical approach such as bisection method [23]. Let $\epsilon_0$ be the initial range that $k^*$ lies in and $\epsilon$ 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, ten or less iterations are usually sufficient for the root-finding. Therefore, for all practical purposes, the procedure in Fig. 5 can be considered to run in constant time.

Fig. 6 compares the optimal delay from our estimation model and that from running TRIO package under SDWS using the 0.18 $\mu m$ technology. Our delay estimation model matches the experimental results very well, with over

**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. (8)
   - 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. (9)

Figure 5: The Delay Estimation Model under Simultaneous Driver and Wire Sizing.

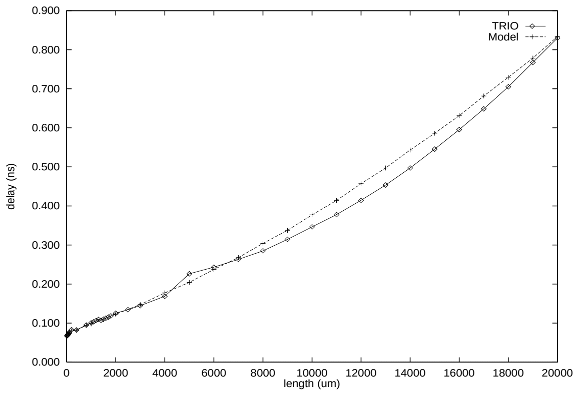


Figure 6: Comparison of the estimation model with running TRIO under SDWS using the 0.18 $\mu m$ technology. $G_0$ and $C_L$ are based on a 10×minimum device.

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. A polynomial-time dynamic programming based algorithm was first presented in [24] to find the optimal buffer placement and sizing for RC trees under the Elmore delay model. It was further extended to include simultaneous buffer insertion and wire-sizing [14]. 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).

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

Figure 7: The procedure to compute critical length for buffer insertion.

### 5.1 Critical Length for Buffer Insertion under Optimal Wire Sizing

Given the delay estimation model $T_{ows}(R_d, l, C_L)$ in Eqn. (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 such buffer at the position of $\alpha l$ from the source ($0 \leq \alpha \leq 1$). Then

$$\begin{aligned} T_{1buf}(\alpha, R_d, l, C_L) &= T_{ows}(R_d, \alpha l, C_b) \\ &+ T_b + T_{ows}(R_b, (1-\alpha)l, C_L) \end{aligned} \tag{10}$$

is the delay after inserting one buffer into the wire with OWS of the resulting two wire segments. We can find the $\alpha$ that minimizes the $T_{1buf}(\alpha, R_d, l, C_L)$ by solving the root of $dT_{1buf}/d\alpha = 0$ under $0 \leq \alpha \leq 1$, denoted as $\alpha_{opt}$. Then it is beneficial to insert such a buffer into a wire of length $l$, loading capacitance of $C_L$ and driver resistance of $R_d$ if and only if the resulting delay is smaller than the optimal wire sizing delay, i.e.,

$$T_{1buf}(\alpha_{opt}, R_d, l, C_L) < T_{ows}(R_d, l, C_L) \tag{11}$$

We define the *critical length* for inserting buffer $b$ to be the minimum $l$ that satisfies Eqn. (11) 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

$$f(l) = T_{1buf}(\alpha_{opt}, R_d, l, C_L) - T_{ows}(R_d, l, C_L) = 0 \tag{12}$$

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. 7. Similar to SDWS, we use very fast bisection method [23] to obtain the root for Eqn. (12). Let $\epsilon_0$ be the initial root range and $\epsilon$ 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 enough for our delay estimation purpose, which leads to about 11 iterations for computing $l_{crit}(b, R_d, C_L)$. Therefore in practice, the procedure in Fig. 7 can be considered to run in constant time.

We notice that in a very recent work by [25], critical length concept was also introduced but based on a different assumption. In [25], buffer insertion was performed on a *uniform* wire line, whereas ours is performed together with OWS. In [25], an important observation 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 2 shows the critical lengths for various NTRS'97 technology

| Tech. ($\mu m$) | | 0.25 | 0.18 | 0.15 | 0.13 | 0.10 | 0.07 |
|---|---|---|---|---|---|---|---|
| [25] | | 2.52 | 2.23 | 2.14 | 1.94 | 1.50 | 1.43 |
| Ours | 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 uniform min wire width based on [25] and under OWS based on Eqn. (12) with some typical buffer sizes from 10× to 500× min device.

| Tech. ($\mu m$) | 0.25 | 0.18 | 0.15 | 0.13 | 0.10 | 0.07 |
|---|---|---|---|---|---|---|
| 2-input NAND ($\mu m^2$) | 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× | 2.4 | 3.88 | 5.52 | 6.33 | 7.87 | 7.88 |
| 500× | 3.19 | 5.12 | 7.21 | 8.42 | 9.93 | 10.6 |

Table 3: Logic volume (x$10^6$) in numbers of 2-input NAND gates (area estimated based on NTRS'97) that can be packed in the square $\frac{l_{crit}}{2} \times \frac{l_{crit}}{2}$.

generations using the formula of [25] under uniform minimum wire width (MIN) and using our procedure in Fig. 7 under some typical buffer sizes. It is interesting to observe from that

1. The critical length $l_{crit}$ decreases as technology further advances. But $l_{crit}$'s decrease is not as fast as cell size's scaling down as the feature size decreases. For example, we can 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. $\frac{1}{4}l_{crit}^2$. The area of a 2-input minimum NAND gate is estimated based on NTRS'97 parameters. As shown in Table 3, the logic volume within the range of critical length actually increases due to the scaling down of devices.

2. As $l_{crit}$ decreases and chip size increases [4], more buffers shall be used for performance optimization as the technology scales. So 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!

3. In contrast to [25], our $l_{crit}$ under OWS is no longer independent of buffer size. Indeed, 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 $4.12mm$.

4. Our $l_{crit}$ under OWS may be significantly larger than that based on MIN. For example in the $0.25\mu m$ technology, $l_{crit}$ under OWS with 500× minimum device is 9.98 $mm$, about 4 times of that based on MIN, which is 2.52 $mm$.

It is worthwhile to note that for the same buffer characteristics, a larger $C_L$ will lead to a smaller $l_{crit}$, and a smaller $R_d$ will lead to a longer $l_{crit}$. For the same $R_d$, $C_L$, $R_b$ and $C_b$, Fig. 8 shows that $l_{crit}$ increases monotonically as the intrinsic delay increases. It implies that smaller intrinsic delays will encourage to insert more buffers.

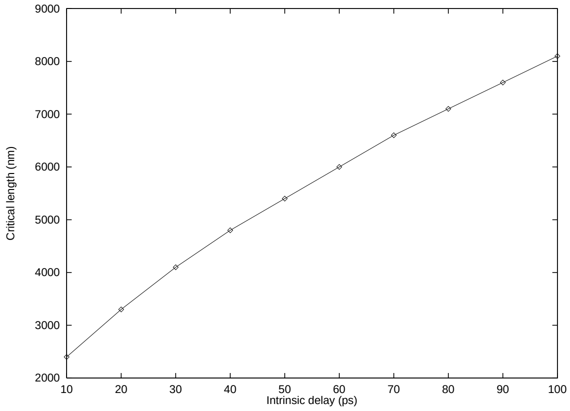


Figure 8: Sensitivity of critical length $l_{crit}$ for buffer insertion with different intrinsic delays. $R_d$, $C_L$, $R_b$ and $C_b$ are all based on a 100× minimum-sized buffer of 0.18 $\mu m$ technology as shown in Table 1.

## 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 size and the size is given *a priori*. We will first provide some theoretical analysis, then give the delay modeling for BIWS.

**Proposition 2** *For optimal BIWS solution to an interconnect wire, the distance between adjacent buffers is equal.*

**Proof:** We just need to prove that for any internal buffer $b$ (i.e., neither the first or the last), it should be inserted in the middle position of its two neighboring buffers. Since all buffers are of the same size, we just need to minimize the sum of two OWS solutions before and after the inserted buffer. According to Proposition 1, $T_{ows}$ is a convex function of $l$. Then, from the definition of the convex function (i.e., $\lambda f(x) + (1-\lambda)f(y) \geq f(\lambda x + (1-\lambda)y)$, $\lambda \in [0,1]$), we have

$$\begin{aligned}
& \frac{1}{2}T_{ows}(R_b, \alpha l, C_b) + \frac{1}{2}T_{ows}\left(R_b, (1-\alpha)l, C_b\right) \\
\geq\ & T_{ows}\left(R_b, \frac{1}{2}\alpha l + \frac{1}{2}(1-\alpha)l, C_b\right) \\
=\ & T_{ows}(R_b, \frac{l}{2}, C_b)
\end{aligned} \tag{13}$$

So the best location for the buffer $b$ will be $\alpha_{opt} = 1/2$. That is to say, the buffers will be equally spaced. □

**Remark 1:** In [26] and [25], buffer insertion was performed at equal spacing for an interconnect with uniform wire width. It was stated that the number of buffers as well as the delay are linear functions of the interconnect length. The justification of such a conclusion was recently presented in [27]. However, none of [26], [27] and [25] performed optimal wire sizing while doing buffer insertion. From our proof above, it is clear that as long as the wire segment delay is a convex function of $l$, which is the case for both uniform wire width and optimal wire sizing, we should insert buffers at equal spacing. □

Given Proposition 2, it is easy to show that

**Proposition 3** *The optimal distance between adjacent buffers under optimal BIWS is $l_{crit}(b, R_b, C_b)$.* □

We simply 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 $T_{crit}$ =

| **Delay Estimation Model under BIWS** |
|---|
| **Input:** $R_{d0}$, $l$, $C_L$, $c_a$, $c_f$, $r$, and buffer $b$ |
| 1. Compute $l_c = l_{crit}(b, R_b, C_b)$<br>2. Compute $\tau_{biws}$ using Eqn. (15)<br>3. Compute $T'_{biws}$ using Eqn. (14)<br>or $T_{biws}$ using Eqn. (16) |

Figure 9: The Delay Estimation Model under Optimal Buffer Insertion and Wire Sizing.

| Tech. ($\mu m$) | 0.25 | 0.18 | 0.15 | 0.13 | 0.10 | 0.07 |
|---|---|---|---|---|---|---|
| $l_c$ | 7.5 | 6.8 | 7.0 | 6.4 | 5.3 | 3.9 |
| $\tau_{biws}$ | 0.53 | 0.44 | 0.41 | 0.39 | 0.38 | 0.39 |
| $\lceil l/l_c \rceil$ | 3 | 3 | 3 | 4 | 4 | 6 |
| #BT | 4 | 4 | 4 | 4 | 4 | 7 |

Table 4: Results from delay estimation model and running TRIO under BIWS. Buffer size is 100× min. $l_c$ is in $mm$, and $\tau_{biws}$ is in $10^{-4} ns/\mu m$. $\lceil l/l_c \rceil$ and #BT are numbers of buffers inserted for a 2 cm interconnect estimated from the model and obtained by running TRIO, respectively.

$t_g + T_{ows}(R_b, l_c, C_b)$ (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:

$$\begin{aligned} T'_{biws} &= T_{crit} \cdot (n_b - 1) + T_{last} \\ &= \tau_{biws} \cdot (n_b - 1) l_c + T_{last} \end{aligned} \tag{14}$$

where $\tau_{biws}$ is given by the delay estimation model under OWS:

$$\begin{aligned} \tau_{biws} &= t_g / l_c + \alpha_1 l_c / W^2(\alpha_2 l_c) + 2\alpha_1 l_c / W(\alpha_2 l_c) \\ &+ R_b c_f + \sqrt{R_b r c_a c_f l_c} \end{aligned} \tag{15}$$

The model in (14) can be further approximated by the following linear model with respect to $l$, which usually will be accurate enough for delay estimation purpose.

$$T_{biws} = \tau_{biws} \cdot l + t_g \tag{16}$$

The delay estimation model under BIWS is summarized in Fig. 9. In practice, $l_c = l_{crit}(b, R_b, C_b)$ can be computed in constant time. Eqns. (14), (15) and (16) can also be computed easily in constant time, so our estimation model under BIWS again takes only constant time. For buffer size of 100× minimum, we list $l_c$ and $\tau_{biws}$ in Table 4. It shows that the estimated buffer numbers match those given by running BIWS algorithm [2] in TRIO very well, differing by at most one for a global 2cm interconnect. In terms of delay, the model in either (14) or (16) gives very good matches compared with running TRIO as shown in Fig. 10, having about 90% accuracy.

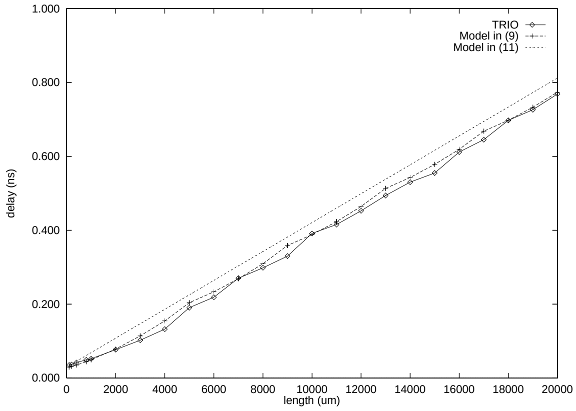


Figure 10: Comparison of delay estimation model with running TRIO under BIWS optimization for the 0.07 $\mu m$ technology. $G_0$ and $C_L$ are based on a 10× minimum device.

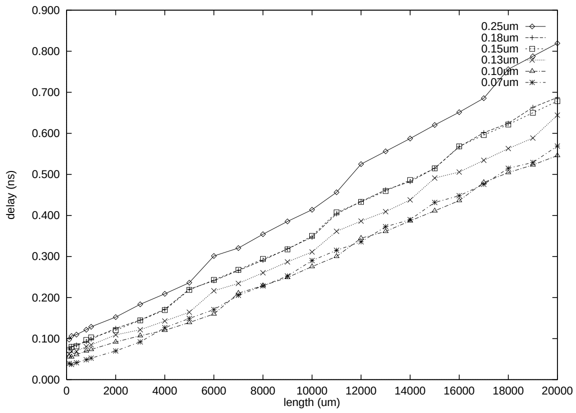


Figure 11: The delay vs. length relationship by using BISWS of TRIO package for various NTRS'97 technology generations from 0.25 $\mu m$ to 0.07 $\mu m$. For all technologies, $G_0$ and $C_L$ in Fig. 1 are based on a 10× minimum.

### 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. The optimal delay will be largely dependent of the given buffer choices, which makes the delay estimation modeling more difficult. However, as seen in BIWS, buffer insertion divides the original long wires into smaller segments according to some critical length $l_c$ and shields the quadratic behavior. Therefore we expect that a similar linear relationship between delay and length will still hold for BISWS. This hypothesis is confirmed by Fig. 11, which shows the optimized delay by running TRIO package under BISWS optimization for various technologies in NTRS'97. For all technologies, the buffer size is up to 1000× minimum feature size, and the wire width is up to 20× minimum width.

Then the only unknown to be determined is the slope of the linear function. For a long wire with many buffers, we observe from our running of TRIO package that the internal buffers will be approximately of the same size and of equal distance between adjacent buffers due to their symmetric structure. Therefore, the slope can be estimated using the best BIWS solution from available buffer selections. The delay estimation model for BISWS is thus proposed as follows:

$$T_{bisws} = \tau_{bisws} \cdot l + t_g \tag{17}$$

| Delay Estimation Model under BISWS |
|---|
| **Input:** $R_{d0}$, $l$, $C_L$, $c_a$, $c_f$, $r$, and the buffer set $B$ |
| 1. Compute the $\tau_{bisws} = \min_{b \in B}\{\tau_{biws}\}$<br>2. Compute $T_{bisws}$ using Eqn. (17) |

Figure 12: The Delay Estimation Model under Optimal Buffer Insertion, Sizing and Wire Sizing.

| Tech. ($\mu m$) | 0.25 | 0.18 | 0.15 | 0.13 | 0.10 | 0.07 |
|---|---|---|---|---|---|---|
| $l_c$ | 9.9 | 9.1 | 9.3 | 8.6 | 7.1 | 5.2 |
| $\tau_{bisws}$ | 0.41 | 0.33 | 0.33 | 0.32 | 0.27 | 0.28 |
| $\lceil l/l_c \rceil$ | 3 | 3 | 3 | 3 | 3 | 4 |
| #BT | 3 | 3 | 3 | 4 | 3 | 5 |

Table 5: Results from delay estimation model and running TRIO under BISWS. For all technologies, $G_0$ and $C_L$ are based on the 10× minimum buffer. The units are the same as in Table 4.

where $\tau_{bisws} = \min_{b \in B}\{\tau_{biws}\}$ from available buffer set $B$. The critical length $l_c$ for BISWS is then estimated by the critical length of BIWS with minimum $\tau_{biws}$. The delay estimation model under BISWS is summarized in Fig. 12. It is easy to show that the time complexity of the model is $O(|B|)$, where $B$ is the set of available buffer sizes (usually no more than 20, so the BISWS estimation model can also be considered to run in constant time for practical purpose). The results from the delay estimation model and from running BISWS algorithm in the TRIO package are shown in Table 5 and Fig. 13. The estimation model again closely matches the experimental results.

**Remark 2:** In [28], the closed-form optimal BISWS solution without consideration of fringing capacitance was derived. From their delay formula, we have done analysis and shown the linear relationship between delay and length, which confirms our results. Yet [28] is a special case of our BISWS as we consider both area and fringing capacitances. It is still an open problem to provide a theoretical justification of the linear relation in Eqn. (17) under BISWS in the general case. We are currently studying this problem. □

# 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). These models 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:

- Placement-driven synthesis and mapping: One may keep a companion placement during synthesis and technology mapping [29, 30]. For every logic synthesis operation, the companion placement will be updated. Once the cell positions are known, one can use our delay estimation models to accurately predict interconnect performance and feed it into the synthesis engine.

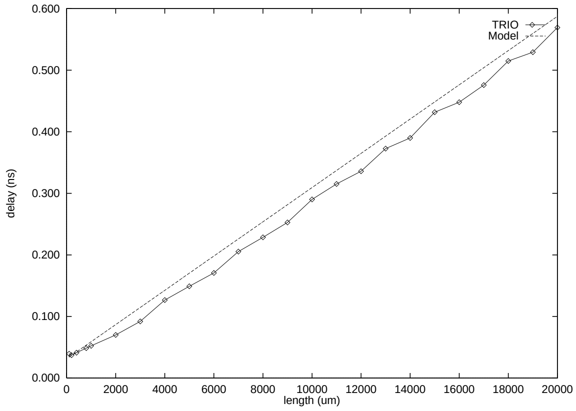


Figure 13: Comparison of delay estimation model and TRIO using BISWS optimization for 0.07 $\mu m$ technology. $G_0$ and $C_L$ are from $10\times$ min device.

- 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 technology 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) Various interconnect optimization alternatives can be evaluated for front-end engines to make decision during RTL synthesis and floorplanning. (ii) Based on the back-end interconnect optimization requirements from our estimation models, routing resource as well as silicon real estate (e.g. for buffer insertion) can be planned beforehand.

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. Martin Wong, Chung-Ping Chen, Youxin Gao from U.T. Austin for providing the continuous wire sizing program, and Mosur K. Mohan, Krishnamurthy Soumyanath, Ganapati Srinivasa from Intel for their input and support of this project. We also thank Chris Chu from U.T. Austin, Lei He, Kei-Yong Khoo, and Cheng-Kok Koh from UCLA for their helpful discussions.

# References

[1] J. Cong, L. He, C.-K. Koh, and P. H. Madden, “Performance optimization of VLSI interconnect layout,” *Integration, the VLSI Journal*, vol. 21, pp. 1–94, 1996.

[2] 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.

[3] J. Cong, “Challenges and opportunities for design innovations in nanometer technologies.,” in *SRC Working Paper, http://www.src.org/prg_mgmt/frontier.dgw*, Dec. 1997.

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

[5] C. Ramachandran, F. Kurdahi, D. Gajski, A.-H. Wu, and V. Chaiyakul, "Accurate layout area and delay modeling for system level design," in *Proc. Int. Conf. on Computer Aided Design*, pp. 355–361, 1992.

[6] Y. Chen, W. Tsai, and F. Kurdahi, "Layout driven logic synthesis system," *IEE Proc.-Circuits Devices Syst.*, vol. 142, no. 3, pp. 158–164, June 1995.

[7] W. C. Elmore, "The transient response of damped linear networks with particular regard to wide-band amplifiers," *Journal of Applied Physics*, vol. 19, no. 1, pp. 55–63, Jan. 1948.

[8] 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.

[9] K. Nabors and J. White, "Fastcap: A multipole accelerated 3-d capacitance extraction program," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, pp. 1447–1459, Nov. 1991.

[10] J. Cong, K. S. Leung, and D. Zhou, "Performance-driven interconnect design based on distributed RC delay model," in *Proc. Design Automation Conf.*, pp. 606–611, 1993.

[11] 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.

[12] J. Cong and L. He, "Optimal wiresizing for interconnects with multiple sources," *ACM Trans. on Design Automation of Electronic Systems*, vol. 1, pp. 478–511, Oct. 1996.

[13] 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.

[14] 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.

[15] T. Okamoto and J. Cong, "Buffered Steiner tree construction with wire sizing for interconnect layout optimization," in *Proc. Int. Conf. Computer Aided Design*, pp. 44–49, Nov. 1996.

[16] J. P. Fishburn and C. A. Schevon, "Shaping a distributed-RC line to minimize Elmore delay," *IEEE Trans. on Circuits and Systems-I: Fundamental Theory and Applications*, vol. 42, pp. 1020–1022, Dec., 1995.

[17] C. P. Chen, Y. P. Chen, and D. F. Wong, "Optimal wire-sizing formula under the Elmore delay model," in *Proc. Design Automation Conf.*, pp. 487–490, 1996.

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

[19] J. P. Fishburn, "Shaping a VLSI wire to minimize Elmore delay," in *Proc. European Design and Test Conf.*, 1997.

[20] Y. Gao and D. F. Wong, "Optimal shape function for a bi-directional wire under elmore delay model," in *Proc. Int. Conf. on Computer Aided Design*, pp. 622–627, 1997.

[21] D. G. Luenberger, *Linear and Nonlinear Programming.* Addison-Wesley, 1984.

[22] 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.

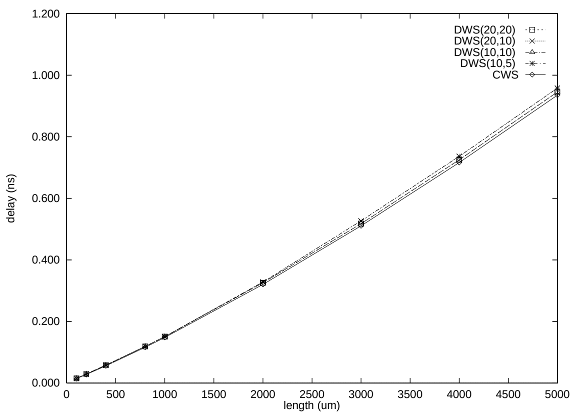


Figure 14: The comparison of delay optimization using continuous and discrete wire sizings using different driving and loadings for 0.18 $\mu m$ technology. $R_d = r_g/10$, $C_L = 10 \times c_g$.

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

[24] 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.

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

[26] H. B. Bakoglu, *Circuits, Interconnections, and Packaging for VLSI.* Addison-Wesley, 1990.

[27] C. J. Alpert and A. Devgan, "Wire segmenting for improved buffer insertion," in *Proc. Design Automation Conf*, pp. 588–593, june 1997.

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

[29] 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.

[30] M. Pedram, "Logical-physical co-design for deep submicron circuits: challenges and solutions," in *Proc. Asia and South Pacific Design Automation Conf.*, pp. 137–142, Feb. 1998.

# Appendix: Comparison of CWS and DWS and Justification of Delay Estimation Model under OWS

First, it is interesting to observe that although the shapes between optimal continuous wire sizing (CWS) and discrete wire sizing (DWS) may be very different, their optimized Elmore delays matches suprisingly well as shown by our experiments. Figs. 14 and 15 show the comparison of the continuous and discrete wire sizing for two typical driver-loading pairs. For discrete case DWS, the first number in the parenthesis is the maximum allowable wire width, in unit of $W_{min}$, the second number is the number of wire choices. For example, DWS(20, 10) has width choices from $W_{min}$ to $20 \times W_{min}$, with the increment to be $2W_{min}$. The wire is segmented in every $10\mu m$. We can see that in both figures, for all DWS cases, they achieve almost the same delay as CWS, which is the theoretical lower bound.

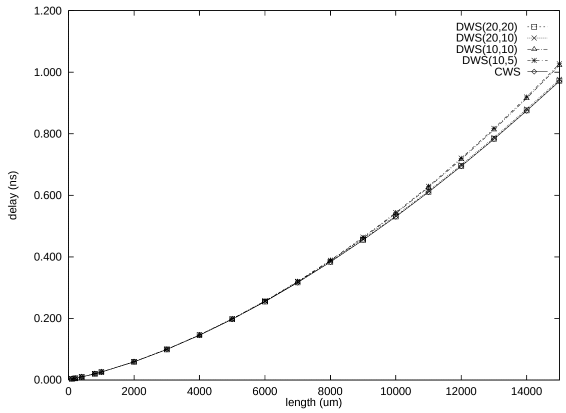


Figure 15: The comparison of delay optimization using continuous and discrete wire sizings using different driving and loadings for 0.18 $\mu m$ technology. $R_d = r_g/100$, $C_L = 100 \times c_g$.

Therefore, in the following, we will use the continuous wire sizing for delay estimation model. [16, 17, 18, 19] are mostly interested in the optimal wire shaping. They did not provide closed-form optimal delay formula. Using the *calculus of variations* but with an alternative Elmore delay expression (being the sum of capacitance times upstream total resistance, rather than the sum of resistance times downstream total capacitance), the closed-form delay formula has been obtained [20] for a bi-direction wire. Since we are mostly interested in the problem of delay estimation, we will rewrite it for uni-direction wire as follows. Denote $T_{cws}(R_d, l, C_L)$ to be the best delay using optimal continuous wire-sizing for an interconnect of length $l$, with driver resistance of $R_d$ and loading capacitance of $C_L$.

Without considering the fringing capacitance, the optimal wire-sizing is an exponential tapering [17] and the optimal delay can be written as

$$T_{cws}(R_d, l, C_L) = R_d C_L e^{2rc_a L/\tau_1} + \frac{1}{2}\tau_1 l \tag{18}$$

where $\tau_1$ is given by $\tau_1 = \frac{rc_a L}{W(\frac{l}{2}\sqrt{\frac{rc_a}{R_d C_L}})}$, in which $W(x)$ is Lambert's $W$ function defined as the value of $w$ that satisfies $we^w = x$. Therefore, we can rewrite Eqn. (18) as:

$$\begin{aligned} T_{cws}(R_d, l, C_L) &= R_d C_L e^{2W(\frac{l}{2}\sqrt{\frac{rc_a}{R_d C_L}})} + \frac{1}{2}\frac{rc_a l^2}{W(\frac{l}{2}\sqrt{\frac{rc_a}{R_d C_L}})} \\ &= \alpha_1 l^2/W^2(\alpha_2 l) + 2\alpha_1 l^2/W(\alpha_2 l) \end{aligned} \tag{19}$$

where $\alpha_1 = \frac{1}{4}rc_a$, $\alpha_2 = \frac{1}{2}\sqrt{\frac{rc_a}{R_d C_L}}$.

The above closed-form expression assumes the fringing capacitance $c_f = 0$. When the fringing capacitance must be considered, the optimal delay formula for $T_{cws}$ will become much more complicated (based on [20]).

$$\begin{aligned} T_{cws}(R_d, l, C_L) = {} & R_d C_L + R_d c_f l + \frac{1}{2}\tau_2 l \\ & + \frac{rc_a}{b} w_1 l + \frac{2rc_a C_L}{bc_f}(w_1 - w_0) \\ & + \frac{rc_a}{2b^2}(w_1^2 - w_0^2) + \frac{rc_a}{b^2}(w_1 - w_0) \end{aligned} \tag{20}$$

where $w_1 = W(-ae^{-bl})$, $w_0 = W(-a)$, $a = \frac{R_d c_f}{R_d c_f + \tau_2} e^{\frac{2rc_a l - R_d c_f}{R_d c_f + \tau_2}}$, $b = \frac{rc_a}{R_d c_f + \tau_2}$, and $\tau_2$ is given by solving the following nonlinear equation:

$$
\begin{aligned}
&\frac{1}{2}l + (\frac{2C_L}{c_f} + l + \frac{1+w_1}{b}) \left[\frac{1}{2} - \frac{1}{4}\frac{b\tau_2}{rc_a(1+w_1)}\right] w_1 \\
&+\frac{1}{2b}(w_1 - w_0) - (\frac{2C_L}{c_f} + \frac{1+w_0}{b}) \\
&\cdot \left[\frac{1}{2} - \frac{1}{4}b(2l + \frac{\tau_2}{rc_a})\frac{1}{1+w_0}\right] w_0 = 0
\end{aligned}
\tag{21}
$$

As we can see, the nonlinear equation (21) is very complicated to get explicit closed-form root. So in [18, 20], it is obtained by using numerical method (such as Newton-Raphson or bisection method). To get an explicit optimized delay estimation under OWS, we have performed extensive experiments for a wide range of parameters, trying to identify the key terms in (20). One crucial observation is that the first three terms in (20) are usually the dominant terms for $T_{cws}$. To get a good approximation of $\tau_2$, we performed extensive numerical analysis to (21) and found that it can be simplified by the following equation.

$$b\tau_2 = 2rc_a(1 + w_1) \tag{22}$$

Based on (22), we can solve $\tau_2$. Combined with the closed-form delay without fringing capacitance in (19), we can identify the adjustment terms from the fringing capacitance and get the following delay estimation under OWS.

$$
\begin{aligned}
T_{ows}(R_d, l, C_L) &= \alpha_1 l^2/W^2(\alpha_2 l) + 2\alpha_1 l^2/W(\alpha_2 l) \\
&+(R_d c_f + \sqrt{R_d r c_a c_f l}) \cdot l
\end{aligned}
\tag{23}
$$