

# Interconnect Performance Estimation Models for Design Planning

Jason Cong, *Fellow, IEEE*, and Zhigang (David) Pan, *Member, IEEE*

**Abstract**—This paper presents a set of interconnect performance estimation models for design planning with consideration of various effective interconnect layout optimization techniques, including optimal wire sizing, simultaneous driver and wire sizing, and simultaneous buffer insertion/sizing and wire sizing. These models are extremely efficient, yet provide high degree of accuracy. They have been tested on a wide range of parameters and shown to have over 90% accuracy on average compared to running best-available interconnect layout optimization algorithms directly. As a result, these fast yet accurate models can be used efficiently during high-level design space exploration, interconnect-driven design planning/synthesis, and timing-driven placement to ensure design convergence for deep submicrometer designs.

**Index Terms**—Buffer insertion and sizing, design planning, driver sizing, interconnect estimation, wire sizing.

## I. INTRODUCTION

AS THE very large scale integrated (VLSI) circuits are scaled into nanometer dimensions and operate in gigahertz frequencies, interconnect design and optimization have become critical in determining system performance, cost, and reliability. In recent years, many effective interconnect optimization techniques have been proposed for interconnect performance optimization, including wire sizing [1]–[6], device sizing [7]–[9], buffer/repeater insertion [8], [10]–[12], and various combinations of these techniques, such as simultaneous device and wire sizing [13]–[15], simultaneous buffer insertion/sizing and wire sizing (BISWS) [16]–[18] (see [19] and [20] for comprehensive survey/tutorial). It was shown in [21] that applying the optimization technique of simultaneous driver sizing, buffer insertion/sizing, and wire sizing can significantly reduce the global interconnect delay (of a 2-cm line) by a factor of five to six times when compared to using nominal wire width in the 0.07- $\mu\text{m}$  technology generation from [22].

Given such a great impact of interconnect layout optimization on the interconnect performance and, thus, on the overall chip performance, it is obvious that interconnect layout optimization must be considered properly at each design stage. However, in the current VLSI design flow, most interconnect layout optimization is performed at late stages such as global

routing or even postlayout phases. Consequently, accurate interconnect performance information, especially that of global interconnects, is not known to higher level synthesis and design planning tools. Since interconnect optimization will improve interconnect delay significantly, it is unlikely for synthesis and design planning tools to make correct decisions without proper modeling of the delays for *optimized* interconnects. Meanwhile, to make sure that the optimized interconnects are realizable at the layout level, proper area estimation of optimized interconnects is also needed beforehand at high level planning stages to ensure the overall design convergence.

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

- 1) Inefficiency: existing interconnect optimization algorithms use either iterative *local refinement* operations (e.g., [1] and [14]), *dynamic programming* approaches (e.g., [7] and [10]), or *mathematical programming* formulation (e.g., [6] and [18]). Although they are usually in polynomial time complexity and efficient to optimize interconnects associated with a *fixed* floorplan/placement, running these algorithms *repeatedly* over tens of thousands of global nets for each of thousands to millions of possible synthesis, floorplan, or placement solutions would be very costly during early evaluation stages;
- 2) 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, and so on. However, such information may not be available at early design planning stages;
- 3) Difficulty to directly integrate high-level design planning tools with interconnect optimization engines due to different levels of data abstraction.

To deal with these problems, we develop in this work a set of fast and accurate interconnect performance estimation models (IPEMs), with consideration of widely used interconnect layout optimization techniques, such as optimal wire sizing (OWS), simultaneous driver and wire sizing (SDWS), and BISWS. Once the design planning and/or synthesis tools generate a floorplan or placement configuration, IPEM can be used to obtain optimal (or near-optimal) interconnect delay and area information *without* going through the actual detailed interconnect optimization process. Note that most previous layout-driven synthesis works, such as [23] and [24], used oversimplified interconnect delay models that assume the interconnect delay be quadratically proportional to wire length without considering the in-

Manuscript received May 24, 2000; revised October 21, 2000. This work was supported in part by the Semiconductor Research Corporation under Contract 98-DJ-605 and by the Intel Corporation. This paper was recommended by Associate Editor M. Sarrafzadeh.

J. Cong is with the Computer Science Department, University of California, Los Angeles, CA 90095 USA (e-mail: cong@cs.ucla.edu).

Z. Pan is with the IBM T. J. Watson Research Center, Yorktown Heights, NY 10598 USA (e-mail: dpan@watson.ibm.com).

Publisher Item Identifier S 0278-0070(01)03546-1.

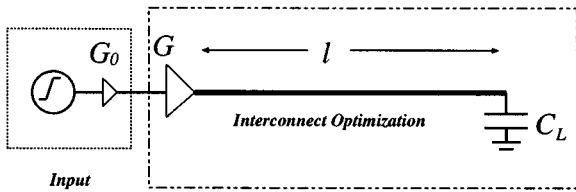


Fig. 1. Problem formulation for a two-pin net with wire length  $l$  and loading capacitance  $C_L$ .

interconnect optimization. Such a quadratic delay model (based on some nominal wire width) can be several times larger than the final optimized delay [20], [21] and is no longer accurate to guide synthesis and design planning tools in DSM designs. Our estimation models developed in this paper effectively overcome all of the aforementioned difficulties.

- 1) They are very efficient (constant runtime in practice).
- 2) They provide high-level abstraction, *explicit* relationship of interconnect performance with key design parameters, and high accuracy. Thus proper and accurate decision can be made at high-level design planning for interconnect design.
- 3) They can be easily embedded into any synthesis and design planning tool, due to the simplicity of the models.

The rest of the paper is organized as follows. Section II states the problem formulation and the parameters. Section III presents the IPEMs for two-pin nets. Section IV studies the IPEMs for multiple-pin nets, with one or multiple critical sinks (MCSs). These estimation models are validated by running detailed interconnect optimization algorithms from UCLA Tree-Repeater-Interconnect-Optimization (TRIO) package [20], [25]. The conclusion follows in Section V. Part of the preliminary results of this work were presented in [26]–[28]. A prototype software package based on this work has been developed. Interested readers can download it from [29].

## II. PROBLEM FORMULATION AND PRELIMINARIES

The objective of this study is to model the optimized interconnect performance (such as optimized interconnect delay and wiring area) under various interconnect optimizations in a simple, efficient, yet accurate manner. Both two- and multiple-pin nets are considered.

### A. Two-Pin Nets

Fig. 1 shows a two-pin interconnect wire of length  $l$  driven by a gate  $G$  and with loading capacitance  $C_L$ . The input waveform to  $G$  is generated by a nominal gate  $G_0$ , which is connected to a voltage supply. The delay to be optimized is the overall delay from the input of  $G_0$  to the sink, 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 overly size  $G$  during the interconnect optimization (e.g., driver sizing). 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.

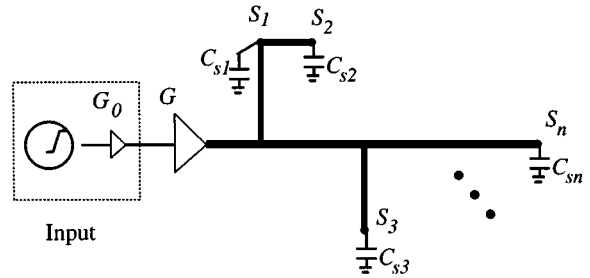


Fig. 2. Problem formulation for a multiple-pin net with tree topology. There are  $n$  sinks  $S_1, S_2, \dots, S_n$ , where each sink  $S_i$  has loading capacitance  $C_{S_i}$ .

### B. Multiple-Pin Nets

For multiple-pin nets, the IPEM problem takes a routing tree as input. The routing tree can be given by a user or obtained using existing topology generation algorithms such as A-tree [30] or P-tree [31]. Fig. 2 shows a multiple-pin net with driver  $G$  and a set of sinks  $S_1, S_2, \dots, S_n$ , where each sink  $S_i$  has loading capacitance  $C_{S_i}$ . In our study, we consider the following two performance-driven optimization objectives: (i) minimizing the delay from source to a give critical sink, (ii) minimizing the maximum delay to all critical sinks (i.e., the *tree delay* as defined in [2]). Other objectives (such as weighted delay) are possible, but are not in the scope of this study.

### C. Interconnect and Device Modeling

During interconnect optimizations, a long wire may be divided into a number of wire segments for wire-sizing and buffer-insertion purpose. To calculate delay, each wire segment is modeled by a  $\pi$ -type resistance-capacitance ( $RC$ ) circuit and each buffer is modeled as a switch-level  $RC$  circuit [20]. The well-known Elmore delay model [32], [33] is used to guide the delay optimization and estimation. Note that although Elmore delay model is not very accurate in DSM design, especially for delay calculation of near-source nodes due to the resistive shielding [34], it is still a proper metric for our delay estimation purpose to provide guidance to high-level design planning.<sup>1</sup>

The following parameters are used by our estimation models.

$W_{\min}$	Minimum wire width in $\mu\text{m}$ .
$S_{\min}$	Minimum wire spacing in $\mu\text{m}$ .
$r$	Sheet resistance in $\Omega/\square$ .
$c_a$	Unit area capacitance in $\text{fF}/\mu\text{m}^2$ .
$c_f$	Unit effective-fringing capacitance in $\text{fF}/\mu\text{m}$ , defined to be the sum of fringing and coupling capacitances, as introduced in [35].
$t_g$	Intrinsic gate delay in ps.
$c_g$	Input capacitance of a minimum-sized gate in fF.
$r_g$	Output resistance of a minimum-sized gate in $\text{k}\Omega$ .

We derive these parameters from [22]. Their values are listed in Table I. For capacitance extraction, we use a 2.5 D capacitance extraction methodology reported in [36], which uses a three-dimensional field solver to generate accurate capacitance values for interpolation and extrapolation.

<sup>1</sup>At high-level design and planning, other sources of errors, such as estimation of interconnect coupling capacitance due to unknown neighborhood structures, could easily outweigh the inaccuracy from the Elmore delay model.

TABLE I  
 PARAMETERS BASED ON NTRS'97

Tech. ( $\mu\text{m}$ )	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
$r$	0.073	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

### III. INTERCONNECT PERFORMANCE ESTIMATION MODELS FOR TWO-PIN NETS

In this section, we focus on interconnect performance (including delay and area) estimation models for two-pin nets, which are the majority of nets in a circuit (e.g., in application-specific integrated circuit designs, about 60%–70% of nets are two-pin nets [37]). The estimation models for two-pin nets in this section will also serve as a basis for developing IPEMs for multiple-pin nets in Section IV.

#### A. IPEM for OWS on Two-Pin Nets

It was first shown in [1] and [30] that when wire resistance becomes significant, as in DSM 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 weighted delay to all timing-critical sinks. It was later extended to optimize a routing tree with multiple sources [4] and to minimize the maximum delay using Lagrangian relaxation [38]. An alternative approach to perform wire-sizing optimization is through bottom-up dynamic programming [16] and it can be combined easily with routing tree construction and buffer insertion [17]. Later on, optimal continuous wire shaping (CWS) for a wire segment was studied. Closed-form wire-shaping functions were obtained to minimize the Elmore delay, first without fringing capacitance [3], [39], then with fringing capacitance [5], [40], and later to a bidirectional wire [41].

It is interesting to observe that one does not have to use very fine granularity of wire-width choice and/or segmentation for discrete wire sizing to obtain the theoretical optimal delay from the CWS. Fig. 3 shows the comparison of their comparison using a typical driver-loading pair. For the discrete case DWS, the first number in the parentheses is the maximum allowable wire width in unit of  $W_{\min}$  and 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  $10 \mu\text{m}$ . We can see that for all DWS cases, they achieve almost the same delay as CWS, which is the theoretical lower bound. So from now on, we only use the generic name of OWS.

1) *Delay Estimation for OWS:* For OWS, the size of driver  $G$  in Fig. 1 is fixed. Let  $T_{\text{ows}}(R_d, l, C_L)$  be the delay under OWS for an interconnect  $l$  with driver resistance  $R_d$  and loading ca-

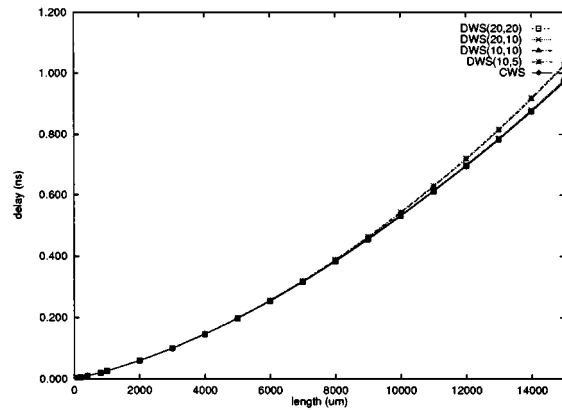


Fig. 3. Comparison of delay optimization using continuous and discrete wire sizings using different driving and loadings for 0.18- $\mu\text{m}$  technology.  $R_d = r_g/100$ ,  $C_L = 100 \times c_g$ .

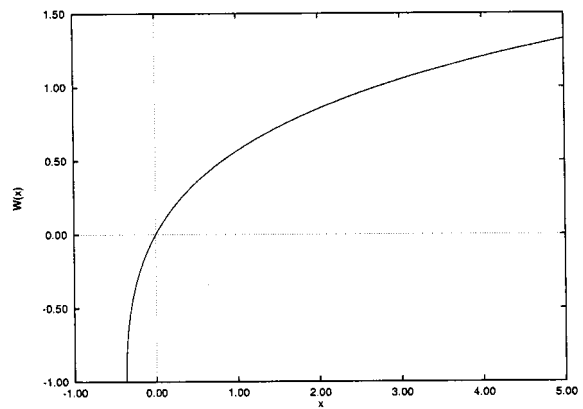


Fig. 4. Euler's Lambert function.

pacitance  $C_L$ . Under given technology, the interconnect parameters of  $r$ ,  $c_a$  and  $c_f$  are fixed and, thus, are not included in the  $T_{\text{ows}}$  notation.

In [3], an exponential wire tapering function was obtained to minimize Elmore delay, but with consideration of only area capacitance (that is,  $c_f = 0$ ). The optimal delay  $T_{\text{ows}}$  under  $c_f = 0$  is then

$$T_{\text{ows}}(R_d, l, C_L)|_{c_f=0} = \frac{\alpha_1 l^2}{W^2(\alpha_2 l)} + \frac{2\alpha_1 l^2}{W(\alpha_2 l)} \quad (1)$$

where  $\alpha_1 = (1/4)rc_a$ ,  $\alpha_2 = (1/2)\sqrt{rc_a/R_d C_L}$ , and  $W(x)$  is Euler's Lambert function [5] defined to be the value of  $w$  that satisfies  $w e^w = x$ .  $W(x)$  function is shown in Fig. 4. 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 have  $W'(x) = W(x)/(x[1 + W(x)])$ .

However, in DSM designs, the fringing and coupling capacitances often dominate the area capacitance and they should be taken into account. In [5], an optimal wire-shaping function with consideration of fringing capacitance was obtained, but it is in a complex formula that needs to solve some complicated nonlinear equations numerically. Computing the optimal delay will be an even harder task. We start from (1) and add the key adjustment terms due to the fringing capacitance by extensive analytical and numerical studies on the wire-shaping function in [5],

then obtain the following simple closed-form delay estimation model for OWS:

$$T_{\text{ows}}(R_d, l, C_L) = \left( \frac{\alpha_1 l}{W^2(\alpha_2 l)} + 2 \frac{\alpha_1 l}{W(\alpha_2 l)} + R_d c_f + \sqrt{R_d r c_a c_f l} \right) \cdot l. \quad (2)$$

One can easily verify that when  $c_f = 0$ , (2) degenerates into (1). In the following, we will show that the general formula of  $T_{\text{ows}}$  in (2) is a convex function of length  $l$ . A function  $f(x)$  is convex if and only if  $f''(x) > 0$ . From the characteristics of  $W$  function, we have the following two lemmas.

*Lemma 1:* Let  $f(x) = (x^2/W(x))$  ( $x > 0$ ). Then,  $f(x)$  is a convex function.

*Proof:* Take the differential of  $f(x)$  with respect to  $x$

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

Then

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

since  $W(x) > 0$  when  $x > 0$ . Therefore,  $f(x)$  is a convex function.  $\square$

*Lemma 2:* Let  $g(x) = (x^2/W^2(x))$  ( $x > 0$ ). Then,  $g(x)$  is a convex function.

*Proof:* Take the differential of  $g(x)$  with respect to  $x$

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

Then

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

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

*Theorem 1:*  $T_{\text{ows}}$  is a subquadratic convex function of the interconnect length  $l$ .

*Proof:* It is obvious that  $T_{\text{ows}}$  is subquadratic, since all the terms  $l^2/W^2(l)$ ,  $l^2/W(l)$ ,  $l$  and  $l^{3/2}$  are subquadratic. From Lemmas 1 and 2, we can easily show that  $T_{\text{ows}}''(l) > 0$ . So, we have the above theorem.  $\square$

Note that interconnect delay under uniform wire width (i.e., no OWS) is a quadratic function of wire length  $l$ . The convexity of  $T_{\text{ows}}$  will be useful to guide optimal buffer insertion and wire sizing (BIWS) estimation in Section III-C.

2) *Area Estimation for OWS:* For DSM designs, wiring area estimation shall also be needed so that adequate routing resource will be allocated at prerouting stages to make sure that the planned wire-sizing optimization will be realizable at the routing stage. To estimate the average wire width for OWS, we first compute the *single* best width using uniform wire sizing [28]. The Elmore delay using uniform wire width  $w$  for the two-pin net in Fig. 1 is

$$\begin{aligned} T(w) &= R_d [(c_a \cdot w + c_f) \cdot l + C_L] \\ &\quad + \frac{rl}{w} \cdot \left[ \frac{(c_a \cdot w + c_f) \cdot l}{2} + C_L \right] \\ &= (R_d c_f l + R_d C_L + \frac{1}{2} r c_a \cdot l^2) \\ &\quad + R_d c_a l \cdot w + \left( \frac{1}{2} r c_f l^2 + r l C_L \right) \cdot \frac{1}{w}. \end{aligned} \quad (3)$$

Thus, the best wire width to minimize  $T(w)$  is

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

We find that the above simple formula can be used to estimate the average wire width from OWS surprisingly well. This is because OWS tapering is usually wider than  $w^*$  near the source, yet narrower than  $w^*$  near the sink. Overall, the average wire width of OWS is still close to  $w^*$ . So the total wiring area using OWS optimization can be estimated by

$$A_{\text{ows}}(R_d, l, C_L) = \sqrt{\frac{r(c_f l + 2C_L)}{2R_d c_a}} \cdot l. \quad (5)$$

The above formula intuitively tells us how an OWS solution shall be under different driver/load and interconnect parameters. For example, we have the following.

- 1) A stronger driver (i.e., smaller  $R_d$ ) or a larger loading capacitance ( $C_L$ ) will lead to a larger optimal wire width. It confirms the driver/loading sizing and wire sizing relationships in [13], [42].
- 2) A larger effective-fringing capacitance coefficient  $c_f$  will lead to a larger optimal wire width, which confirms the effective-fringing property in [35].
- 3) A larger sheet resistance  $r$  will lead to a larger optimal wire width and a larger  $c_a$  will result in a smaller optimal wire width.

3) *Model Validation for OWS:* We have tested our closed-form delay and area estimation models under OWS on a wide range of parameters. They match those from running TRIO package under OWS optimization very well, with more

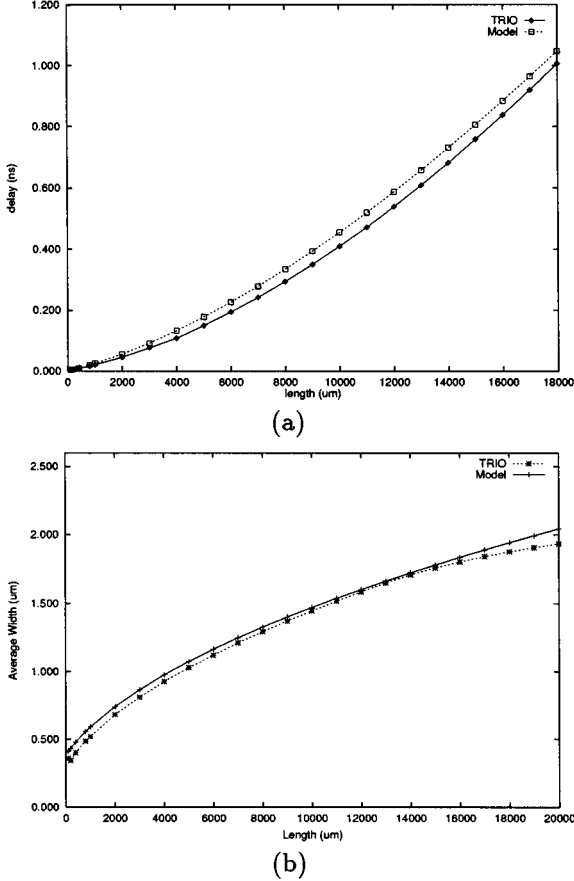


Fig. 5. Comparisons of (a) delay and (b) average wire width from our estimation model with TRIO for OWS optimization under the 0.18- $\mu\text{m}$  technology, with  $R_d = r_g/100$ ,  $C_L = c_g \times 100$ . TRIO uses wire-width set  $\{W_{\min}, 2W_{\min}, \dots, 20W_{\min}\}$  and 10- $\mu\text{m}$ -long segments.

than 90% accuracy in general. The OWS algorithm in TRIO is based on the iterative *local refinement* [1] operation, which is essentially a greedy algorithm that optimally sizes one wire segment at a time. A speed-up technique called *bundled refinement* [4] is also used in TRIO, which only divides a long wire into some smaller wire segments (for finer wire-sizing granularity) when necessary. In practice, the lower and upper bounds from the bundled refinement procedure usually meet. When they do not meet, a bottom-up dynamic programming algorithm is used to compute the optimal wire-sizing solution between the lower and upper bounds. Fig. 5 shows their delay and wiring area (equivalently, average wire width) comparisons to running OWS engine in the TRIO package for wire length range from 100  $\mu\text{m}$  to 2 cm. We can see that both delay and average wire-width estimations are very accurate.

In terms of runtime, since we have the closed-form formula for both delay and area estimations, the time complexity for our model is constant. In fact, our estimation model is so fast that we have to call the estimation procedure many times using a loop to collect a measurable central processing unit (CPU) time. The CPU time to run the estimation model 10 000 times (or equivalently, to estimate 10 000 nets) is just 0.8 s on a SUN UltraSPARC 1. However, using the efficient local refinement based OWS algorithm in TRIO for *one* net will take about 1 s. Therefore, our estimation model is an order of  $10^4$  faster. Note that

SDWS Estimation Model	
<b>Input:</b> $R_0, l, C_L, c_a, c_f, \tau$ , and driver set $D$ of size between $[k_{\min}, k_{\max}]$	
1. Calculate the best driver size $k_{\text{opt}}$ that minimize $T(k)$ of Eqn. (6) <ul style="list-style-type: none"> <li>- calculate the root <math>k^*</math> of <math>dT(k)/dk = 0</math></li> <li>- if <math>k_{\min} &lt; k^* &lt; k_{\max}</math>,  <math>k_{\text{opt}}</math> is one of <math>\lfloor k^* \rfloor</math> or <math>\lceil k^* \rceil</math> which gives smaller <math>T(k)</math></li> <li>- else  <math>k_{\text{opt}}</math> is one of <math>k_{\min}</math> or <math>k_{\max}</math> which gives smaller <math>T(k)</math></li> </ul>	
2. Compute $T_{\text{sdws}}$ from (7) and $A_{\text{sdws}}$ from (8).	

Fig. 6. Delay ( $T_{\text{sdws}}$ ) and wiring area ( $A_{\text{sdws}}$ ) estimation model under SDWS optimization.

this does not mean TRIO or other interconnect optimization algorithms are no longer needed. Our estimation model only provides the estimations of optimal delay and area for high level synthesis and design planning (e.g., to screen out floorplanning candidates that cannot meet timing target even considering interconnect optimization), but *not* the OWS solution itself. TRIO or other interconnect optimization algorithms are still needed to obtain the *final* optimal interconnect layout.

### B. IPEM for SDWS on Two-Pin Nets

This section presents the two-pin interconnect estimation model under SDWS, which sizes both wire and driver [13]. To obtain an accurate delay/area estimation, we will make use of the accurate OWS delay/area estimation model in (2) and (5). In our problem formulation, the input stage  $G_0$  is fixed and the driver  $G$  will be optimally sized to achieve the best performance from available driver set  $D$ . Denote  $R_0$  and  $R_d$  to be the effective resistance of  $G_0$  and  $G$  and  $C_d$  to be the input capacitance of  $G$ . Suppose the size of driver  $G$  is  $k \times$  of 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  $C_L$  (Fig. 1) to be minimized is

$$\begin{aligned} T(k) &= (t_g + R_0 \cdot C_d) + t_g + T_{\text{ows}}(R_d, l, C_L) \\ &= (t_g + R_0 \cdot kc_g) + t_g + T_{\text{ows}}(r_g/k, l, C_L). \end{aligned} \quad (6)$$

Note that the input stage delay ( $t_g + R_0 \cdot C_d$ ) is included for overall delay minimization, but not in the current-stage delay estimation. Substituting the delay formula of  $T_{\text{ows}}$  from (2) and calculating the optimal driver size  $k_{\text{opt}}$  from available driver set (to be explained soon), we can obtain the following delay and wiring area estimation models under optimal SDWS:

$$T_{\text{sdws}}(D, l, C_L) = t_g + T_{\text{ows}}(r_g/k_{\text{opt}}, l, C_L) \quad (7)$$

$$A_{\text{sdws}}(D, l, C_L) = \sqrt{\frac{r(c_f l + 2C_L)k_{\text{opt}}}{2r_g c_a}} \cdot l. \quad (8)$$

The SDWS estimation modeling procedure is outlined in Fig. 6. To compute  $k_{\text{opt}}$ , we first set  $dT(k)/dk = 0$  and compute its root  $k^*$ . It can be solved efficiently by the bisection method [43]. Let  $\epsilon_0$  be the initial range that  $k^*$  lies in and  $\epsilon$  be the error tolerance for  $k^*$ . The bisection method basically

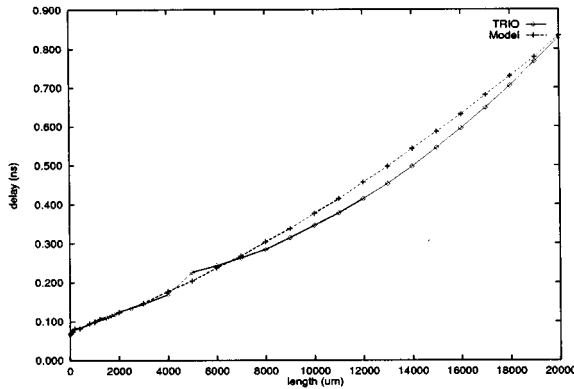


Fig. 7. Comparison of our delay estimation model with TRIO for SDWS under 0.18- $\mu\text{m}$  technology, with  $G_o$  and  $C_L$  of  $10\times$  min gate. Maximum driver for TRIO is set to be  $200\times$  min gate.

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 fewer iterations are usually sufficient for the root finding. Therefore,  $k^*$  can be computed in constant time. Then, we project  $k^*$  to the set of available drivers to obtain  $k_{\text{opt}}$ .

Fig. 7 compares the delay from our estimation model and the optimal delay from running TRIO package under SDWS using the 0.18- $\mu\text{m}$  technology. The SDWS algorithm in TRIO is based on that in [13]. Our delay estimation model again matches TRIO very well, with over 90% accuracy on average.

### C. IPEM for BISWS on Two-Pin Nets

BISWS is a widely used interconnect optimization technique for both delay and crosstalk reduction [8], [10]–[12]. Dynamic programming based algorithms are usually used for BISWS [10], [16]. However, they do not provide simple estimation of the optimized BISWS behavior.<sup>2</sup> In this section, we will derive effective estimation models for BISWS. We first introduce the concept of *critical length* for buffer insertion under OWS and give an analytical equation that characterizes it. Then, we derive IPEMs for BIWS (no buffer sizing) and for BISWS.

1) *Critical Length for Buffer Insertion Under OWS*: We first compute the longest length that a wire can run without the benefit from buffer insertion. Let  $T_{1\text{buf}}(\alpha, R_d, l, C_L)$  denote the delay by inserting a buffer at the position of  $\alpha l$  from the source ( $0 \leq \alpha \leq 1$ ). Then

$$T_{1\text{buf}}(\alpha, R_d, l, C_L) = T_{\text{ows}}(R_d, \alpha l, C_b) + t_g + T_{\text{ows}}(R_b, (1 - \alpha)l, C_L) \quad (9)$$

is the delay after inserting the buffer. Note that OWS is applied into the two wire segments separated by the buffer  $b$  with intrinsic delay of  $t_g$ , input capacitance of  $C_b$ , and output resistance of  $R_b$ .

We can find the  $\alpha$  that minimizes  $T_{1\text{buf}}(\alpha, R_d, l, C_L)$  by getting the root of  $dT_{1\text{buf}}/d\alpha = 0$  under  $0 \leq \alpha \leq 1$  denoted as

<sup>2</sup>When buffer insertion (with only one buffer size) is used alone for a uniform-width wire, closed-form formulae for delay were derived in [11], [44].

Procedure to Compute $l_{\text{crit}}(b, R_d, C_L)$	
<b>Input:</b> $R_d, C_L$ , and buffer $b$ with characteristics of $R_b, C_b$ and $T_b$	
/* Use bisection (binary search) method */	
1. initialize $l_{\text{crit}}$ 's range $[l_{\text{min}}, l_{\text{max}}]$ , where $f(l_{\text{min}}) > 0$ and $f(l_{\text{max}}) < 0$	
2. while $l_{\text{max}} - l_{\text{min}} > \epsilon$	
$l_{\text{mid}} \leftarrow (l_{\text{min}} + l_{\text{max}})/2$	
if $f(l_{\text{mid}}) > 0$ , $l_{\text{min}} \leftarrow l_{\text{mid}}$	
else $l_{\text{max}} \leftarrow l_{\text{mid}}$	
3. return $l_{\text{mid}}$	

Fig. 8. Procedure to compute critical length for buffer insertion.

$\alpha^*(R_d, l, C_L)$ , which by itself is a function of driver resistance, length, and loading capacitance. Then, it is beneficial to insert such a buffer if and only if the resulting delay is smaller than the OWS delay, i.e.,

$$T_{1\text{buf}}(\alpha^*(R_d, l, C_L), R_d, l, C_L) < T_{\text{ows}}(R_d, l, C_L). \quad (10)$$

We define the *critical length* for inserting buffer  $b$  to be the minimum  $l$  that satisfies (10) and denote it as  $l_{\text{crit}}(b, R_d, C_L)$ .

Clearly, when the wire length  $l$  is small, OWS will achieve the best delay; whereas when the interconnect is long enough, the buffer insertion becomes beneficial. Thus, the root for

$$f(l) = T_{1\text{buf}}(\alpha^*(R_d, l, C_L), R_d, l, C_L) - T_{\text{ows}}(R_d, l, C_L) \quad (11)$$

denoted as  $l^*$  gives the critical length for buffer insertion, i.e.,  $l_{\text{crit}}(b, R_d, C_L)$ . Similar to SDWS, we use a very fast binary search to obtain the root for  $f(l)$ . It is outlined in Fig. 8. Note that we need a two-level binary search for  $l^*$  and  $\alpha^*$ . Let  $\epsilon_{l0}, \epsilon_l$  be the initial range and the error tolerance for  $l^*$  and  $\epsilon_{\alpha0}, \epsilon_\alpha$  be the initial range and the error tolerance for  $\alpha^*$ . Then, the root for  $f(l)$  can be computed in  $\log_2(\epsilon_{l0}/\epsilon_l)$  iterations. For each  $l$ , we need another binary search for  $\alpha^*$ , which takes  $\log_2(\epsilon_{\alpha0}/\epsilon_\alpha)$  steps. In practice,  $\epsilon_{l0} = 2$  cm,  $\epsilon_l = 10$   $\mu\text{m}$ ,  $\epsilon_{\alpha0} = 1$ , and  $\epsilon_\alpha = 0.01$  are usually sufficient for our estimation purpose, which leads to at most  $\log_2 2000 \times \log_2 100 = 77$  steps for computing  $l_{\text{crit}}(b, R_d, C_L)$ .

In a recent work by [45], critical length concept was also introduced, but for a *uniform-width* wire. An interesting observation from [45] is that  $l_{\text{crit}}$  is independent of buffer size for a uniform-width wire. However, this is not the case for our  $l_{\text{crit}}$  where OWS is performed. As a comparison, Table II shows the critical length comparison from the formula in [45] under minimum wire width and from our formula with OWS, using some typical buffer sizes from  $10\times$  to  $500\times$  min gate. The driver and receiver are set to be the same size as the buffer.

From Table II, it is interesting to observe the following.

- 1) In contrast to [45], our  $l_{\text{crit}}$  with OWS is no longer independent of buffer sizes. It tends to increase as buffer size gets larger. For example, in the 0.25- $\mu\text{m}$  technology,  $l_{\text{crit}}$  for  $200\times$  buffer is 8.65 mm, more than double of that for  $10\times$  buffer (which is only 4.12 mm). Moreover, our  $l_{\text{crit}}$  with OWS is usually larger than that from [45] without consideration of OWS.

TABLE II  
 CRITICAL LENGTH  $l_{\text{crit}}$  (IN  $\mu\text{m}$ ) FOR BUFFER INSERTION

Tech. ( $\mu\text{m}$ )	0.25	0.18	0.15	0.13	0.10	0.07
[45]	2.52	2.23	2.14	1.94	1.50	1.43
10 $\times$	4.12	3.80	3.97	3.61	2.92	2.08
50 $\times$	6.40	5.81	6.01	5.51	4.45	3.30
100 $\times$	7.47	6.83	7.04	6.39	5.30	3.91
200 $\times$	8.65	7.92	8.14	7.43	6.35	4.49
500 $\times$	9.98	9.10	9.30	8.57	7.13	5.21

 TABLE III  
 LOGIC VOLUME (UNIT:  $10^6$ ) COMPARISON

Tech. ( $\mu\text{m}$ )	0.25	0.18	0.15	0.13	0.10	0.07
10 $\times$	0.55	0.89	1.31	1.49	1.66	1.69
50 $\times$	1.31	2.09	3.01	3.48	3.87	4.25
100 $\times$	1.79	2.88	4.13	4.68	5.48	5.97
200 $\times$	2.40	3.88	5.52	6.33	7.87	7.88
500 $\times$	3.19	5.12	7.21	8.42	9.93	10.6
$A_{\text{nand}}$ ( $\mu\text{m}^2$ )	7.80	4.04	3.00	2.18	1.28	0.64

- 2) In general,  $l_{\text{crit}}$  decreases as technology further advances. It implies more and more buffers are needed for future generation DSM circuit designs.
- 3) Although  $l_{\text{crit}}$  decreases as feature size scales down, this does not mean less logic cells can be reached by  $l_{\text{crit}}$ . To illustrate it, we define the *logic volume* to be the number of two-input minimum NAND gates that can be packed in the region spanned by  $(1/2)l_{\text{crit}} \times (1/2)l_{\text{crit}}$ , i.e.,  $(1/4)l_{\text{crit}}^2$ . In Table III, we show the logic volume under different technology generations and driver/buffer sizes. The area estimation of a two-input minimum NAND gate ( $A_{\text{nand}}$ ) is listed at the last row, which assumes that the spacing between adjacent gates is  $3\times$  feature size, and the island spacing between  $N$ -well and  $P$ -well is  $20\times$  feature size. It can be seen that the logic volume indeed increases as the device feature gets smaller, although the critical length  $l_{\text{crit}}$  becomes smaller. For an example of driver/buffer of  $100\times$  minimum size, as technology advances from  $0.25\text{-}\mu\text{m}$  to  $0.07\text{-}\mu\text{m}$ , more gates (i.e., functionality) can be packed within the critical length (from 1.79 million to almost 6 million).

2) *BIWS*: In this section, we derive the delay estimation model under optimal BIWS. We assume that all the buffers are of the same given size (buffer sizing is considered in Section III-C3). Denote  $l_c = l_{\text{crit}}(b, R_d, C_L)$ . From the definition of the critical length, it is obvious that if  $l \leq l_c$ , no buffer is needed and OWS alone achieves the best delay. For  $l > l_c$ , one or more buffers shall be inserted. In the case of only one buffer is inserted, we can compute the best insertion position  $\alpha^*(R_d, l, C_L)$  and then the optimal delay will be  $T_{1\text{buf}}(\alpha^*, R_d, l, C_L)$  from (9). In the following, we will focus on the buffer insertion case with two or more buffers.

*Lemma 3*: For an optimal BIWS solution, the distance between adjacent buffers is equal.

*Proof*: We only 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

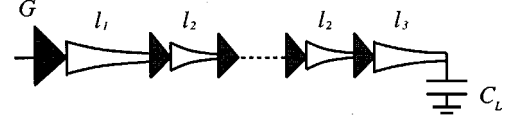


Fig. 9. Simultaneous BIWS optimization.

are of the same size, we only need to minimize the sum of two OWS delays before and after the inserted buffer. According to Theorem 1,  $T_{\text{ows}}$  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

$$\begin{aligned}
 & \frac{1}{2}T_{\text{ows}}(R_b, \alpha l, C_b) + \frac{1}{2}T_{\text{ows}}(R_b, (1 - \alpha)l, C_b) \\
 & \geq T_{\text{ows}}\left(R_b, \frac{1}{2}\alpha l + \frac{1}{2}(1 - \alpha)l, C_b\right) \\
 & = T_{\text{ows}}\left(R_b, \frac{l}{2}, C_b\right).
 \end{aligned} \tag{12}$$

So the best location for the buffer  $b$  will be  $\alpha_{\text{opt}} = 1/2$ , i.e., the buffers will be equally spaced.  $\square$

*Remark*: In [8] and [45], buffer insertion was performed at equal spacing for an interconnect with uniform wire width. It was stated that the number of buffers and also the delay are linear functions of the interconnect length. The justification of such a conclusion was recently presented in [11]. However, none of [8], [11], or [45] performed OWS 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 width wire and optimally sized wire, we should insert buffers at equal spacing.  $\square$

Based on Lemma 3, we only need to determine the following values for the optimal BIWS solution (with two or more buffers inserted, see Fig. 9): 1)  $l_1$ , the optimal distance from the source to the first buffer; 2)  $l_2$ , the optimal distance between adjacent buffers; and 3)  $l_3$ , the optimal distance from the last buffer to the sink. Note that OWS is performed for each wire segment. In the following, we will show how to estimate  $l_1$ ,  $l_2$ , and  $l_3$ .

Let  $l_{c1} = l_{\text{crit}}(b, R_d, C_b)$ , i.e., the critical length for the first buffer with driver resistance  $R_d$  and loading capacitance  $C_b$ , and similarly  $l_{c2} = l_{\text{crit}}(b, R_b, C_b)$  for internal buffers and  $l_{c3} = l_{\text{crit}}(b, R_b, C_L)$  for the last buffer. Then, we can obtain the following properties for  $l_1$ ,  $l_2$ , and  $l_3$ .

*Lemma 4*: For an optimal BIWS solution with two or more buffers inserted,  $l_1 + l_2 > l_{c1}$  and  $l_2 + l_3 > l_{c3}$ .

*Proof*: By the contradiction method. If  $l_1 + l_2 \leq l_{c1}$ , i.e., the second buffer is located within the range  $l_{c1}$  from the source, then the first buffer shall not be inserted at the beginning, which leads to contradiction. Similar proof for  $l_2 + l_3 > l_{c3}$ .  $\square$

*Theorem 2*: For an optimal BIWS solution with two or more buffers inserted, let  $\alpha_1 = \alpha^*(R_d, l_{c1}, C_b)$  and  $\alpha_3 = \alpha^*(R_b, l_{c3}, C_L)$  (see Section III-C1 for the definition of  $\alpha^*$  function). Then, we have: 1)  $\text{MAX}(\alpha_1 l_{c1}, l_{c1} - l_{c2}) < l_1 \leq l_{c1}$ ; 2)  $l_2 \leq l_{c2}$ ; and 3)  $\text{MAX}((1 - \alpha_3)l_{c3}, l_{c3} - l_{c2}) < l_3 \leq l_{c3}$ .

*Proof*: First, according to the definition of the critical length for buffer insertion, we have  $l_i \leq l_{ci}$  ( $i = 1, 2, 3$ ). From Lemma 4, the second buffer must be located at least  $l_{c1}$  from

<b>BIWS Estimation Model</b>	
<b>Input:</b> $R_d, l, C_L, c_a, c_f, r,$ and buffer $b$	
<ol style="list-style-type: none"> <li>1. Compute <math>l_c = l_{crit}(b, R_d, C_L)</math>, <math>l_{c1} = l_{crit}(b, R_d, C_b)</math>,  <math>l_{c2} = l_{crit}(b, R_b, C_b)</math>, and <math>l_{c3} = l_{crit}(b, R_b, C_L)</math>; { /* Figure III-C.1 */ }</li> <li>2. if (<math>l \leq l_c</math>) { /* No buffer needed */  <math>T_{biws} = T_{ows}(R_d, l, C_L)</math>; { /* Eqn. (2) */ }</li> <li>3. } else if (<math>l &lt; l_{c1} + l_{c3}</math>) { /* One buffer needed */  compute <math>\alpha_0 = \alpha^*(R_d, l, C_L)</math>;  <math>T_{biws} = T_{1buf}(\alpha_0, R_d, l, C_L)</math> as in Eqn. (9);  <math>A_{biws} = A_{ows}(R_d, \alpha_0 l, C_b) + A_{ows}(R_b, (1 - \alpha_0)l, C_L)</math>;</li> <li>4. } else { /* Two or more buffers needed */  compute <math>\alpha_1 = \alpha^*(R_d, l_{c1}, C_b)</math> and <math>\alpha_3 = \alpha^*(R_b, l_{c3}, C_L)</math>;  <math>l_{1min} = MAX(\alpha_1 l_{c1}, l_{c1} - l_{c2})</math>, <math>l_{3min} = MAX((1 - \alpha_3)l_{c3}, l_{c3} - l_{c2})</math>;  search the best locations of <math>l_1 \in (l_{1min}, l_{c1})</math> and <math>l_3 \in (l_{3min}, l_{c3})</math>;  compute <math>n_2 = \lfloor \frac{l - l_1 - l_3}{l_{c2}} \rfloor</math>, <math>l_2 = \frac{l - l_1 - l_3}{n_2}</math>;  <math>T_{biws} = T_{ows}(R_d, l_1, C_b) + (n_2 + 2)t_g + (n_2 + 1)T_{ows}(R_b, l_2, C_b) + T_{ows}(R_b, l_3, C_L)</math>;  <math>A_{biws} = A_{ows}(R_d, l_1, C_b) + n_2 A_{ows}(R_b, l_2, C_b) + A_{ows}(R_b, l_3, C_L)</math>;</li> </ol>	

Fig. 10. Delay ( $T_{biws}$ ) and wiring area ( $A_{biws}$ ) estimation model for optimal BIWS.

the driver. Since for an optimal BIWS solution, any buffer is in their local optimal position, then the first buffer must be at least  $\alpha_1 l_{c1}$  from the driver, i.e.,  $\alpha_1 l_{c1} < l_1$ . Again from Lemma 4, we also have  $l_1 > l_{c1} - l_2 \geq l_{c1} - l_{c2}$ . Thus, we prove that  $MAX(\alpha_1 l_{c1}, l_{c1} - l_{c2}) < l_1 \leq l_{c1}$ . Similarly, we can prove the inequality for  $l_3$ .  $\square$

Theorem 2 gives the lower and upper bounds for the optimal buffer insertion lengths  $l_1, l_2$ , and  $l_3$ . The procedure to estimate the optimal BIWS delay ( $T_{biws}$ ) and area ( $A_{biws}$ ) is outlined in Fig. 10. At step 1, it computes the critical lengths for insertion of buffer  $b$  under four different driver resistance and loading capacitance combinations, i.e.,  $(R_d, C_L)$ ,  $(R_d, C_b)$ ,  $(R_b, C_b)$ , and  $(R_b, C_L)$ , denoted to be  $l_c, l_{c1}, l_{c2}$ , and  $l_{c3}$ , respectively. Minimal necessary buffers are inserted for performance optimization. So, when  $l < l_c = l_{crit}(b, R_d, C_L)$ , no buffer is inserted and it degenerates to the OWS optimization. One buffer is inserted when  $l_c < l < l_{c1} + l_{c3}$ . In the situation when two or more buffers are needed, we first compute the ranges for  $l_1$  and  $l_3$  based on Theorem 2, then search for the best combination for them. Since the optimal BIWS delay is not sensitive to fairly large region near its optimal position [46], an optimal  $(l_1, l_3)$  pair can be obtained by a simple linear search with a coarse granularity (e.g., five by five).

For a wire length  $l$  significantly larger than all critical lengths, i.e.,  $l \gg l_{c1}, l_{c2}, l_{c3}$ , we will have asymptotically  $l_1 \approx l_{c1}$ ,  $l_1 \approx l_{c1}$ ,  $l_2 \approx l_{c2}$ , and  $n_2 \approx (l - l_1 - l_3)/l_{c2} \propto l$ . Thus, we have the following linear delay model versus length after optimal BIWS:

$$T_{biws} \approx \tau \cdot l + \beta \quad (13)$$

where  $\tau$  and  $\beta$  are some technology dependent constants [see [26], eq. (9)].<sup>3</sup>

Since the critical length for buffer insertion can be computed in constant time, our estimation model under BIWS again takes only constant time. Fig. 11 shows the comparison of our model with TRIO. Again, our delay estimation closely matches that

<sup>3</sup>Note that in [26], the driver is assumed to be of the same size as the buffer. However, such assumption is taken out in this paper and, thus, the model in Fig. 10 is more general and accurate than that in [26].

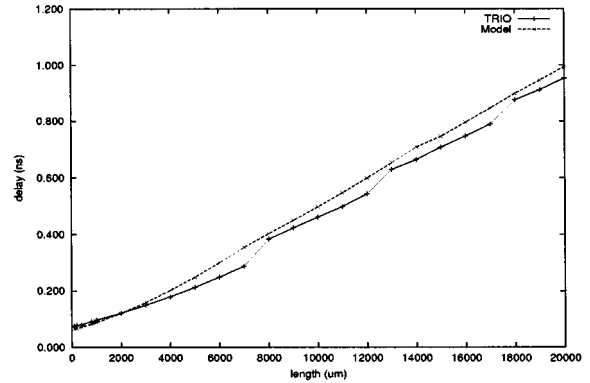


Fig. 11. Comparison of optimized delay from IPEM with TRIO under BIWS using 0.18- $\mu$ m technology.  $G_0$  and  $C_L$  are from  $10 \times$  minimum gate size. Buffer size is  $100 \times$  min.

from TRIO. It also verifies the asymptotic linear relationship in (13).

3) *BISWS*: The *BISWS* optimization also allows buffer sizing to further reduce delay for global interconnects. 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

$$T_{bisws} = \min_{b \in B} \{T_{biws}(b)\} \quad (14)$$

where  $B$  is the available buffer library and  $T_{biws}(b)$  is the best delay from BIWS using buffer  $b$ . In [47], the closed-form optimal *BISWS* solution *without fringing capacitance* was derived. The work in [47], as a special case of our *BISWS*, confirms our observation of asymptotic linear model using *BISWS*. 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 estimation model and from running *BISWS* algorithm in TRIO package are shown in Fig. 12. The estimation model again



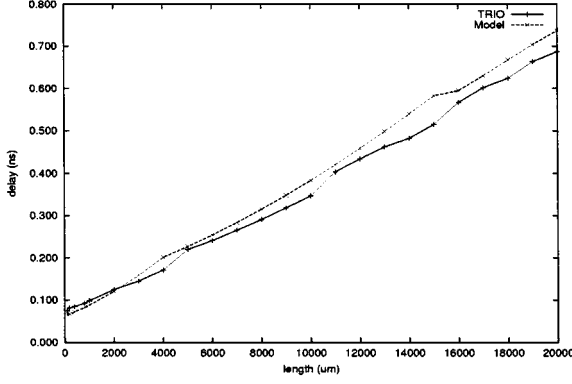


Fig. 12. Comparison of BISWS estimation model with TRIO using 0.18- $\mu\text{m}$  technology.  $G_0$  and  $C_L$  are from  $10 \times \text{min}$ . To run TRIO, 20 buffer choices are used with sizes from min to  $400 \times \text{min}$ .

achieves about 90% accuracy. In terms of runtime, our model is again extremely fast. The CPU time to run the model for 10 000 nets is just 8 s. However, using the bottom-up dynamic programming approach based BISWS in TRIO for *one* net will take about 14 s using ten different wire and buffer choices with wire segmentations in every 500  $\mu\text{m}$ . So, our estimation model is again an order of  $10^4$  times faster.

#### IV. IPEM FOR MULTIPLE-PIN NETS

So far, we have focused on the IPEM for two-pin nets, which on one hand, are the majority nets in a design and on the other hand, will serve as a basis for developing IPEM for multiple pin nets. The interconnect estimation problem for multiple-pin nets with tree topology is formulated in Section II-B. We aim to estimate the following two performance-driven design objectives by interconnect optimization: 1) minimizing delay to a single critical sink (SCS); and 2) minimizing maximum delay to MCSs.

For multiple-pin net estimation, the problem is much more difficult than that for two-pin nets because: 1) there are no closed-form wire-shaping functions like those for two-pin nets [3], [5], [39], [41]; and 2) all current interconnect optimization algorithms (e.g., TRIO) rely on iterative methods such as local/bound refinement and dynamic programming, which provides no intuition for simple closed-form like estimation. To overcome these difficulties, the key idea of our approach is to transform the multiple-pin net estimation problem into one or several two-pin net estimation problems and then employ the results from Section III.

##### A. IPEM for SCS

In this section, we study interconnect delay and area estimation under SCS formulation with consideration of two interconnect optimization techniques: OWS and BISWS.

1) *OWS for SCS*: For delay minimization to an SCS  $S_k$ , OWS will only size wire segments along the critical path (i.e., the path from  $G$  to  $S_k$ ) and use the minimum width for all other wire segments not on the critical path so that the wire load from noncritical sinks is minimum. Since the wire load at each branch from the critical path can be precomputed before performing OWS, we can transform the original OWS problem with tree

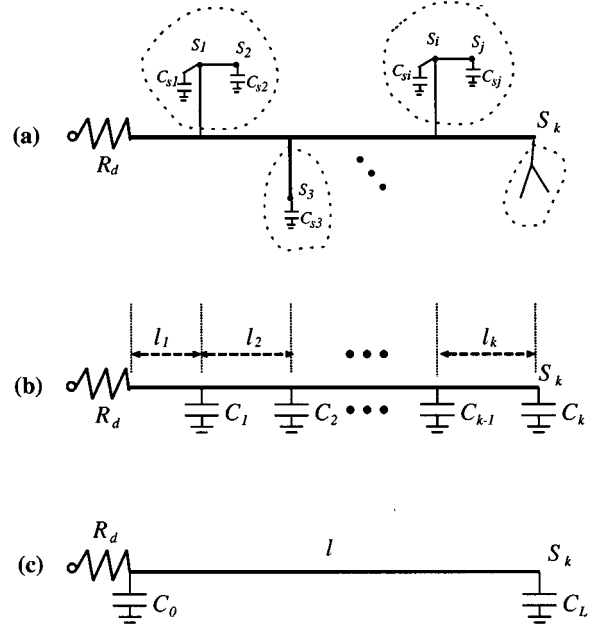


Fig. 13. (a) Transformation for a general routing tree to (b) an SLML problem for an SCS  $S_k$  and then (c) to an SLDL problem.  $l_1 + l_2 + \dots + l_k = l$  is the wire length from driver to the critical sink  $S_k$ .  $C_0$  and  $C_L$  are weighted summation of  $C_1, \dots, C_k$  given by Theorem 3.

topology into an equivalent single-line-multiple-load (SLML) problem, as shown in Fig. 13, from (a) to (b). In the figure,  $R_d$  is the effective resistance of the driver  $G$ , and  $S_k$  is the SCS. At each branch  $i$  on the critical path,  $C_i$  is the total effective downstream capacitance (excluding that from the critical path).

In [48], it was shown that a simple wire sizing scheme that uses the best single width (denoted as 1WS) can approximate the delay and area of OWS with many wire-width selections reasonably well. So, we will first start with a single-width sizing. Under single-width sizing, we can reduce the multiple-pin problem into a much simpler two-pin problem as shown in Fig. 13(c). The transformation is formally described by the following theorem.

*Theorem 3*: In terms of the Elmore delay from the source to the single-critical sink  $S_k$ , the multiple-pin problem in Fig. 13(b) is equivalent to the two-pin problem in Fig. 13(c) for any wire width  $w$ , where in Fig. 13(c)

$$C_L = \sum_{j=1}^k \frac{\sum_{i=1}^j l_i}{l} \cdot C_j \quad (15)$$

$$C_0 = \sum_{j=1}^k C_j - C_L. \quad (16)$$

*Proof*: The Elmore delay of Fig. 13(b) can be written as

$$T = R_d \left[ \sum_{j=1}^k (c_a w + c_f) l_j + \sum_{j=1}^k C_j \right] + \sum_{j=1}^{k-1} \frac{r l_j}{w} \\ \times \left[ \frac{1}{2} (c_a w + c_f) l_j + \sum_{i=j+1}^k (c_a w + c_f) l_i + \sum_{i=j}^k C_i \right] \\ + \frac{r l_k}{w} \left[ \frac{1}{2} (c_a w + c_f) \cdot l_k + C_k \right]$$

$$\begin{aligned}
&= R_d \left( c_a w l + c_f l + \sum_{j=1}^k C_j \right) + r \left( c_a + \frac{c_f}{w} \right) \\
&\quad \cdot \left[ \frac{1}{2} \sum_{j=1}^k l_j^2 + \sum_{j=1}^{k-1} \sum_{i=j+1}^k l_j l_i \right] + \frac{r}{w} \sum_{j=1}^k \left( l_j \sum_{i=j}^k C_j \right) \\
&= R_d \left( c_a w l + c_f l + \sum_{j=1}^k C_j \right) \\
&\quad + \frac{1}{2} r \left( c_a + \frac{c_f}{w} \right) \left( \sum_{j=1}^k l_j \right)^2 \\
&\quad + \frac{r}{w} \sum_{j=1}^k \left( C_j \sum_{i=1}^j l_i \right) \\
&= R_d (c_a w l + c_f l + C_0 + C_L) + \frac{1}{2} r \left( c_a + \frac{c_f}{w} \right) l^2 \\
&\quad + \frac{r l C_L}{w}. \tag{17}
\end{aligned}$$

Therefore, it is equivalent to the Elmore delay in Fig. 13(c).  $\square$

Intuitively, Theorem 3 transforms the original multiple-pin problem in Fig. 13(b) to a two-pin problem by redistributing each internal loading capacitance  $C_i$  into two parts. One part of  $C_i$  goes to  $C_L$  at the sink based on the ratio of  $C_i$ 's upstream wire resistance to total resistance on the critical path. The other part of  $C_i$  goes to  $C_0$  at the source to preserve the constant term  $R_d C_i$ . Note that Theorem 3 holds for any wire width  $w$ . From (17), we can compute the best single-width  $w^*$  that minimizes the Elmore delay for (17)

$$w^* = \sqrt{\frac{r(c_f l + 2C_L)}{2R_d c_a}}. \tag{18}$$

The optimal Elmore delay for Fig. 13(b) or (c) using  $w^*$  is the same, i.e.,

$$\begin{aligned}
T_{1ws} &= R_d C_0 + R_d (c_f l + C_L) + \frac{1}{2} r c_a l^2 \\
&\quad + 2 \sqrt{r R_d c_a \left( \frac{1}{2} c_f l + C_L \right)} \cdot l. \tag{19}
\end{aligned}$$

As mentioned earlier, we have observed in [48] that the delay using the single best width  $T_{1ws}$  is a reasonable estimation for  $T_{ows}$  for the two-pin case [see Fig. 13(c)], since  $T_{ows}$  is usually between 0.8 to 0.95 times  $T_{1ws}$ . This is also the case for the multiple-pin case of Fig. 13(b). Then, we can then use  $T_{ows}$  for Fig. 13(c) to estimate the optimal OWS delay for Fig. 13(b) with, at most,  $(0.95 - 0.8)/0.8 = 18\%$  error.<sup>4</sup> Note that in practice, the estimation error is usually much smaller than the maximum error because OWS for both two- and multiple-pin cases will reduce the delay in a similar manner.

$T_{ows}$  for Fig. 13(c) is available from the delay estimation of two-pin nets in (2). Using (2) and taking the constant term

<sup>4</sup>This estimation is especially robust when wire length is shorter than the *critical length* for buffer insertion, where uniform wire sizing has comparable delay to that by OWS [28].

<p>Input: Interconnect network with tree structure, and certain critical sink <math>S_k</math></p> <ol style="list-style-type: none"> <li>1. Compute <math>C_1, C_2, \dots, C_k</math> at each branch, using minimum wire width;</li> <li>2. Compute <math>C_L</math> and <math>C_0</math> using (15) and (16);</li> <li>3. Estimate critical path delay using (20);</li> <li>4. Estimate critical path area using (21).</li> </ol>
---

Fig. 14. Delay and area estimation for SCS with OWS.

$R_d C_0$  into consideration, we have the following delay estimation model for the critical path of a multiple-pin net using OWS optimization:

$$\begin{aligned}
T_{ows} &= R_d C_0 \\
&\quad + \left( \frac{\alpha_1 l}{W^2(\alpha_2 l)} + 2 \frac{\alpha_1 l}{W(\alpha_2 l)} + R_d c_f + \sqrt{R_d r c_a c_f l} \right) \cdot l. \tag{20}
\end{aligned}$$

The wiring area estimation for the *critical path*<sup>5</sup> can be obtained using the same formula as for a two-pin net, i.e.,

$$A_{ows} = \sqrt{\frac{r(c_f l + 2C_L)}{2R_d c_a}} \cdot l. \tag{21}$$

Fig. 14 summarizes the delay and area estimation procedure for an SCS using OWS. To demonstrate its effectiveness, we apply it to some randomly generated nets with one, two, four, or eight branches using typical parameter ranges ( $l$  from 1 to 10 mm,  $R_d$  from 50 to 1000  $\Omega$ ,  $C_i$  from 10 to 100 fF). Fig. 15 shows the scatter diagrams of the delay and the average width comparisons by our model and by running the OWS algorithm in TRIO package. We can see that both the delay and the area (i.e., average wire width) estimations from our model match those from TRIO very well (i.e., close to the  $y = x$  line). The average and maximum errors are 9.3% and 15.4% for delay estimation, and 4.5% and 19.9% for area estimation, respectively.

2) *BISWS for SCS*: Optimizing SCS delay using BISWS can be formulated as a special case of the SLML problem by inserting minimum buffer at every branch on the critical path (from source to the critical sink) to shield all the downstream interconnect and device capacitances, as shown in Fig. 16. For DSM designs, the input capacitance of a minimum-sized buffer is very small, just about the wire capacitance of 1 or 2  $\mu\text{m}$  (see Table I). So it can just be ignored during delay estimation. Then, our delay estimation model for two-pin nets as in (14) directly applies to the optimized BISWS delay estimation to the critical sink. Note that although simple, this estimation model is useful to estimate the best possible delay to *any* sink using BISWS, and to evaluate and screen out floorplanning and placement candidates. Due to its simplicity, we do not include experimental results here, since they are essentially the same as those for two-pin nets.

## B. IPEM for MCSs

In this section, we study interconnect delay estimation for MCSs under the optimization objective of minimizing the maximum delay of all critical sinks (i.e., the *tree delay* following

<sup>5</sup>For those wires on the noncritical path, the minimum wire width is used to minimize the wire load to the critical path.

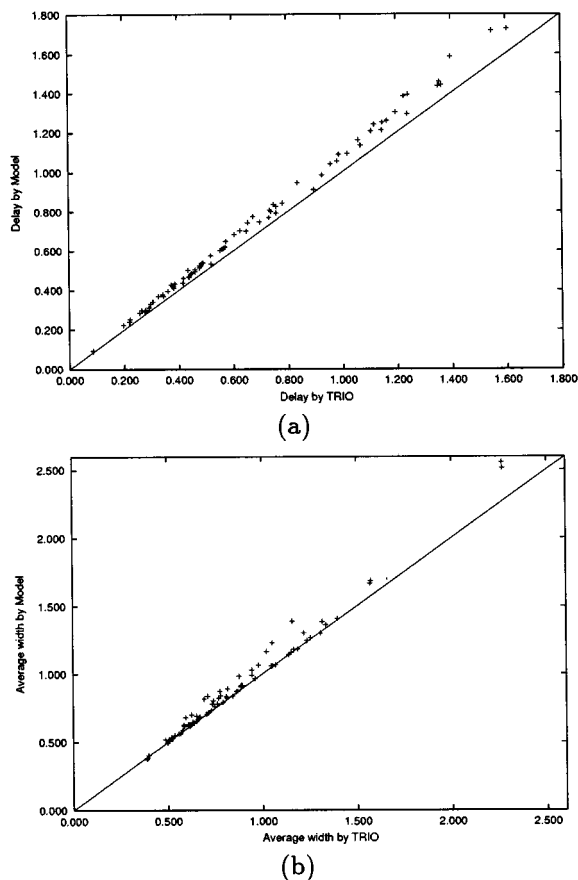


Fig. 15. Scatter diagrams of (a) delay and (b) average width comparison of our model with TRIO using 0.18- $\mu\text{m}$  technology. TRIO uses 20 discrete wire-width choices with maximum width of  $20 \times W_{\text{min}}$  and wire segmentation of every 10  $\mu\text{m}$ .

the definition in [2]). To minimize the tree delay, [2] formulated it into a convex optimization problem and developed a sensitivity-based algorithm to solve it. The tree delay minimization can also be solved using the weighted delay formulation through Lagrangian relaxation [38] or it can be solved directly through bottom-up dynamic programming [16], [17]. In this paper, we use the dynamic programming approach [17] implemented in TRIO package for the comparison with our estimation models.

1) *OWS for MCS*: Given a routing tree connecting MCSs, we have the following definitions.

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

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

The estimation for the optimal tree delay (i.e., the minimized maximum delay of all critical sinks) with MCS is much more difficult than the delay estimation with SCS because when we optimize the delay for one critical sink, it may affect all other critical sinks as well. That is why all optimization algorithms in [2], [16], [17], and [38] used iterative-based approaches. However, we notice that there are some simple, but very useful characteristics for the optimal tree delay. First, under the Elmore delay model, it can be easily shown that

**Lemma 5:** The critical sink that has the maximum delay from the source must be a leaf critical sink.

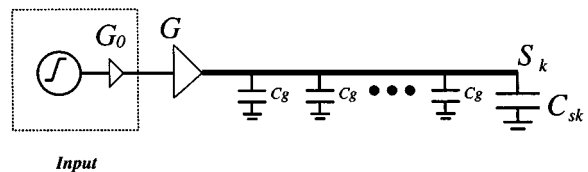


Fig. 16. To estimate the best delay from the source to the sink  $S_k$ , we insert the minimum buffer at every branch on the critical path from source to sink  $S_k$  to shield the downstream capacitance at each branch.

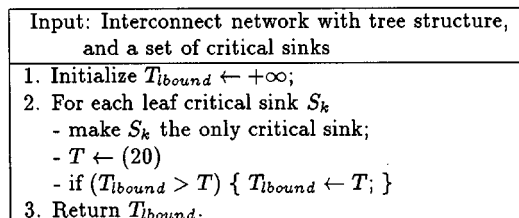


Fig. 17. Delay estimation for the optimal tree delay using OWS.

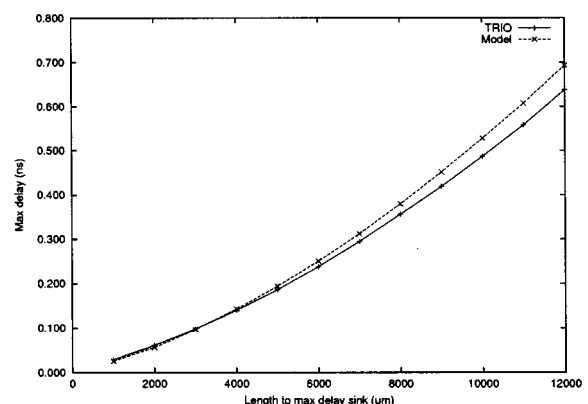


Fig. 18. Comparison of our model with TRIO for optimal tree delay using OWS under the 0.18- $\mu\text{m}$  technology.  $R_d = 180 \Omega$ ,  $C_s = 10 \text{ fF}$ . Length from source to the maximum delay sink ranges from 1–20 mm. TRIO uses ten discrete wire-width choices with maximum width of  $20 \times W_{\text{min}}$  and wire segmentation in every 500  $\mu\text{m}$ .

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

**Theorem 4:** The optimal delay to any critical sink under the SCS formulation is a lower bound for the optimal tree delay.

From Theorem 4, we can obtain a tight lower bound by taking the maximum delay for all leaf critical sinks under the SCS formulation and use it to estimate the optimal tree delay, as shown in Fig. 17.

Our experiments show that this lower bound delay estimation is indeed fairly tight and we can just use  $T_{\text{bound}}$  to estimate the optimal tree delay. The explanation is as follows. Since our objective is to minimize the maximum delay, i.e., the delay to the most critical sink, we shall keep the wire load from less critical sinks as small as possible (but may not be too small; otherwise, they may become the most critical sink). To the most critical

TABLE IV  
SUMMARY OF THE INTERCONNECT PERFORMANCE ESTIMATION MODELS UNDER DIFFERENT SCENARIOS

	Two-Pin Nets	Multiple-Pin Nets	
		Single Critical Sink	Multiple Critical Sinks
OWS	(2) and (5)	(20) and (21)	Figure IV-B.1
SDWS	Figure III-B	Combine Figure III-B and Eqn. (20)	Combine Figures III-B and IV-B.1
BIWS	Figure III-C.2	—	—
BISWS	(14)	Similar to 2-Pin case	Max of all 2-Pin cases

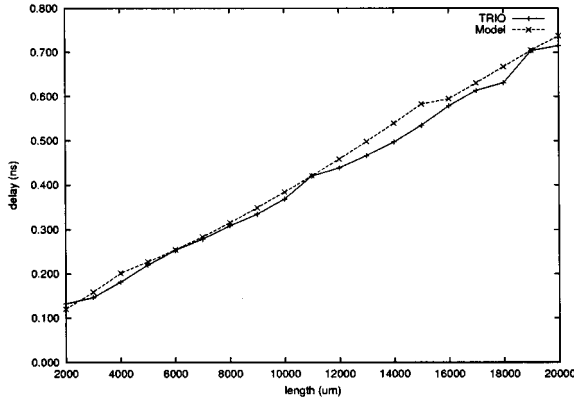


Fig. 19. Comparison of our model with TRIO for optimal tree delay using BISWS under the 0.18- $\mu\text{m}$  technology.  $R_{d0} = r_g/10$ ,  $C_s = 10c_g$ . TRIO uses ten discrete wire-width choices with maximum width being  $20 \times W_{\min}$ , ten buffer choices with maximum buffer size being  $400 \times \min$ , and wire segmentation in every 500  $\mu\text{m}$ .

sink, the main difference between our model and the real optimal solution is that the former uses the minimum wire width to compute the wire load while the latter uses *as small as possible* widths to compute the wire load for noncritical paths. For DSM designs, the area capacitance is usually dominated by effective-fringing capacitance [20]. Therefore, the two wire loads and then the resulting *optimized* delays to the *most critical* sink do not differ significantly. Fig. 18 shows the delay comparison of our model and TRIO for some random four-pin nets using typical parameters from the 0.18- $\mu\text{m}$  technology. Again, our delay estimations match those from TRIO well. Note that for some lengths (e.g.,  $>10000 \mu\text{m}$ ), our model has slightly larger delay than that from TRIO. This is because our delay estimation model in (2) tends to have slightly more conservative delay estimation (see Figs. 5 and 15).

2) *BISWS for MCS*: Similar to OWS, we find that the optimal tree delay under BISWS can be estimated by a tight lower bound delay from the leaf critical sink that has the maximum delay under the SCS formulation. That is to say we just need to evaluate a small number (i.e., the number of leaf critical sinks) of SCS configurations and then use the result from Section IV-A2 to estimate the optimal tree delay. Fig. 19 shows the comparison of our model and the BISWS algorithm TRIO. Again, our simple model gives accurate estimation for the optimal tree delay.

## V. CONCLUSION

This paper has developed a set of IPERMs (with closed-form formulae or simple computational procedures) under various interconnect optimization techniques such as OWS, SDWS, and

BISWS for both local wires (without buffer insertion) and global wires (with buffer insertion/sizing). Both two- and multiple-pin scenarios are studied. The models are summarized in Table IV for ease of reference. For multiple-pin nets, we only presented the models for OWS and BISWS in detail. For SDWS, one can easily use the OWS models and combine them with the optimal driver sizing in Fig. 6 to solve it.

Our estimation models are shown to be very accurate and extremely efficient (constant time in practice) compared with running complex interconnect optimization algorithms (e.g., those from TRIO package) directly. In addition, they can be easily embedded and coded into any synthesis engine and design planning tool. The IPERMs obtained in this paper have been integrated into a software package IPERM [29]. We expect that these delay estimation models can be used in a wide spectrum of applications, listed, but not limited, as follows.

- 1) *Timing-driven placement and floorplanning*: during the placement or floorplanning, our models can be used to accurately predict the behavior of the *optimized* global interconnects.
- 2) *Placement-driven synthesis and mapping*: a companion placement may be kept during synthesis and technology mapping [49]. For every logic synthesis operation, the companion placement will be updated. Once the cell positions are known, our models can be used to accurately predict interconnect performance for the synthesis engine.
- 3) *Interconnect process parameter optimization*: interconnect parameters (such as metal aspect ratio, wire width, and wire spacing) may be tuned to optimize the delays predicted by our models for global, average, and local interconnects under certain wire-length distributions, using different interconnect optimization techniques.
- 4) *Interconnect planning*: our models can also be used to evaluate different optimization alternatives and to plan routing and silicon resources beforehand for interconnect layout optimizations.

Our IPERMs have been successfully used in interconnect architecture planning [28], buffer block planning for interconnect-driven floorplanning [46], and MARCO GSRC technology extrapolation system (GTX) [50]. We plan to apply them throughout an interconnect-centric design flow [51] to achieve better design convergence in the future.

## ACKNOWLEDGMENT

The authors would like to thank Prof. M. D. F. Wong from the University of Texas, Austin, Dr. W. Donath, Dr. D. Kung, Dr. R. Puri and Dr. L. Stok from IBM, Dr. M. K. Mohan and Dr. P. Arunachalam from Intel, and Dr. W. Long from the University

of California at Los Angeles for their helpful discussions. They would also like to thank the anonymous reviewers for their constructive comments.

## REFERENCES

- [1] J. Cong and K. S. Leung, "Optimal wiresizing under the distributed Elmore delay model," in *Proc. Int. Conf. Computer-Aided Design*, Nov. 1993, pp. 634–639.
- [2] S. S. Sapatnekar, "RC interconnect optimization under the Elmore delay model," in *Proc. Design Automation Conf.*, June 1994, pp. 387–391.
- [3] J. P. Fishburn and C. A. Schevon, "Shaping a distributed-RC line to minimize Elmore delay," *IEEE Trans. Circuits Syst. I*, vol. 42, pp. 1020–1022, Dec. 1995.
- [4] J. Cong and L. He, "Optimal wiresizing for interconnects with multiple sources," *ACM Trans. Design Automat. Electron. Syst.*, vol. 1, no. 4, pp. 478–511, Oct. 1996.
- [5] C.-P. Chen and D. F. Wong, "Optimal wire sizing function with fringing capacitance consideration," in *Proc. Design Automation Conf.*, June 1997, pp. 604–607.
- [6] R. Kay and L. T. Pileggi, "EWA: Efficient wiring-sizing algorithm for signal nets and clock nets," *IEEE Trans. Computer-Aided Design*, vol. 17, pp. 40–49, Jan. 1998.
- [7] H. C. Lin and L. W. Linholm, "An optimized output stage for MOS integrated circuits," *IEEE J. Solid-State Circuits*, vol. SC-10, pp. 106–109, Apr. 1975.
- [8] H. B. Bakoglu, *Circuits, Interconnections, and Packaging for VLSI*. Reading, MA: Addison-Wesley, 1990.
- [9] D. Zhou and X. Y. Liu, "On the optimal drivers for high-speed low power ICs," *Int. J. High Speed Electron. Syst.*, vol. 7, no. 2, pp. 287–303, June 1996.
- [10] L. P. P. van Ginneken, "Buffer placement in distributed RC-tree networks for minimal Elmore delay," *Proc. IEEE Int. Symp. Circuits and Systems*, pp. 865–868, May 1990.
- [11] C. J. Alpert and A. Devgan, "Wire segmenting for improved buffer insertion," in *Proc. Design Automation Conf.*, June 1997.
- [12] C. J. Alpert, A. Devgan, and S. T. Quay, "Buffer insertion for noise and delay optimization," in *Proc. Design Automation Conf.*, June 1998, pp. 362–367.
- [13] J. Cong and C.-K. Koh, "Simultaneous driver and wire sizing for performance and power optimization," *IEEE Trans. VLSI Syst.*, vol. 2, pp. 408–423, Dec. 1994.
- [14] J. Cong and L. He, "Theory and algorithm of local-refinement based optimization with application to device and interconnect sizing," *IEEE Trans. Computer-Aided Design*, vol. 18, pp. 406–420, Apr. 1999.
- [15] C. C. N. Chu and D. F. Wong, "An efficient and optimal algorithm for simultaneous buffer and wire sizing," *IEEE Trans. Computer-Aided Design*, vol. 18, pp. 1297–1304, Sept. 1999.
- [16] 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. Computer-Aided Design*, Nov. 1995, pp. 138–143.
- [17] T. Okamoto and J. Cong, "Buffered Steiner tree construction with wire sizing for interconnect layout optimization," in *Proc. Int. Conf. Computer-Aided Design*, Nov. 1996, pp. 44–49.
- [18] C. C. N. Chu and D. F. Wong, "A quadratic programming approach to simultaneous buffer insertion/sizing and wire sizing," *IEEE Trans. Computer-Aided Design*, vol. 18, pp. 787–798, June 1999.
- [19] J. Cong, L. He, C.-K. Koh, and P. H. Madden, "Performance optimization of VLSI interconnect layout," *Integr. VLSI J.*, vol. 21, pp. 1–94, 1996.
- [20] J. Cong, L. He, K.-Y. Khoo, C.-K. Koh, and D. Z. Pan, "Interconnect design for deep submicron ICs," in *Proc. Int. Conf. on Computer Aided Design*, Nov. 1997, pp. 478–485.
- [21] J. Cong. (1997, Dec.) Challenges and opportunities for design innovations in nanometer technologies. [Online]. Available: [http://www.src.org/prg\\_mgmt/frontier.dgw](http://www.src.org/prg_mgmt/frontier.dgw)
- [22] Semiconductor Industry Association, *National Technology Roadmap for Semiconductors*. San Jose, CA: SIA, 1997.
- [23] C. Ramachandran, F. J. Kurdahi, D. D. Gajski, A. C.-H. Wu, and V. Chaiyakul, "Accurate layout area and delay modeling for system level design," in *Proc. Int. Conf. Computer-Aided Design*, Nov. 1992, pp. 355–361.
- [24] Y. Chen, W. K. Tsai, and F. J. Kurdahi, "Layout driven logic synthesis system," in *Proc. Inst. Elec. Eng. Circuits Devices Syst.*, vol. 142, June 1995, pp. 158–164.
- [25] UCLA Tree Repeater Interconnect Optimization Package (TRIO), J. Cong, L. He, C.-K. Koh, D. Z. Pan, and X. Yuan. (1999). [Online]. Available: [http://cadlab.cs.ucla.edu/software\\_release/trio/htdocs/](http://cadlab.cs.ucla.edu/software_release/trio/htdocs/)
- [26] J. Cong and D. Z. Pan, "Interconnect delay estimation models for synthesis and design planning," in *Proc. Asia South Pacific Design Automation Conf.*, Jan. 1999, pp. 97–100.
- [27] —, "Interconnect delay and area estimation for multiple-pin nets," in *Proc. IEEE/ACM Int. Workshop Timing Issues in Specification and Synthesis of Digital Systems (TAU)*, Mar. 1999, pp. 179–184.
- [28] —, "Interconnect estimation and planning for deep submicron designs," in *Proc. Design Automation Conf.*, June 1999, pp. 507–510.
- [29] Performance Estimation Models for Optimized Interconnects (IPEM), J. Cong, W. Long, and D. Z. Pan. (1999). [Online]. Available: [http://cadlab.cs.ucla.edu/software\\_release/ipem/htdocs/](http://cadlab.cs.ucla.edu/software_release/ipem/htdocs/)
- [30] J. Cong, K. S. Leung, and D. Zhou, "Performance-driven interconnect design based on distributed RC delay model," in *Proc. Design Automation Conf.*, June 1993, pp. 606–611.
- [31] J. Lillis, C. K. Cheng, T. T. Y. Lin, and C. Y. Ho, "New performance driven routing techniques with explicit area/delay tradeoff and simultaneous wire sizing," in *Proc. Design Automation Conf.*, June 1996, pp. 395–400.
- [32] W. C. Elmore, "The transient response of damped linear networks with particular regard to wide-band amplifiers," *J. Appl. Phys.*, vol. 19, no. 1, pp. 55–63, Jan. 1948.
- [33] J. Rubinstein, P. Penfield Jr., and M. A. Horowitz, "Signal delay in RC tree networks," *IEEE Trans. Computer-Aided Design*, vol. CAD-2, pp. 202–211, July 1983.
- [34] L. Pileggi, "Timing metrics for physical design of deep submicron technologies," in *Proc. Int. Symp. Physical Design*, Apr. 1998, pp. 28–33.
- [35] J. Cong, L. He, C.-K. Koh, and Z. Pan, "Global interconnect sizing and spacing with consideration of coupling capacitance," in *Proc. Int. Conf. Computer-Aided Design*, 1997, pp. 628–633.
- [36] J. Cong, L. He, A. B. Kahng, D. Noye, N. Shirali, and S. H.-C. Yen, "Analysis and justification of a simple, practical 2 1/2-d capacitance extraction methodology," in *Proc. ACM/IEEE Design Automation Conf.*, June 1997, pp. 40.1.1–40.1.6.
- [37] M. J. S. Smith, *Application-Specific Integrated Circuits*. Reading: Addison-Wesley, MA 1997.
- [38] 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.*, June 1996, pp. 405–408.
- [39] C. P. Chen, Y. P. Chen, and D. F. Wong, "Optimal wire-sizing formula under the Elmore delay model," in *Proc. Design Automation Conf.*, June 1996, pp. 487–490.
- [40] J. P. Fishburn, "Shaping a VLSI wire to minimize Elmore delay," in *Proc. Eur. Design Test Conf.*, Mar. 1997.
- [41] Y. Gao and D. F. Wong, "Optimal shape function for a bi-directional wire under elmore delay model," in *Proc. Int. Conf. Computer-Aided Design*, Nov. 1997, pp. 622–627.
- [42] J. Cong, C.-K. Koh, and K.-S. Leung, "Simultaneous buffer and wire sizing for performance and power optimization," in *Proc. Int. Symp. Low Power Electronics Design*, Aug. 1996, pp. 271–276.
- [43] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in FORTRAN—The Art of Scientific Computing*. Cambridge, U.K.: Cambridge Univ. Press, 1992.
- [44] R. H. J. M. Otten and R. K. Brayton, "Planning for performance," in *Proc. Design Automation Conf.*, June 1998, pp. 122–127.
- [45] R. H. J. M. Otten, "Global wires harmful?," in *Proc. Int. Symp. Physical Design*, Apr. 1998, pp. 104–109.
- [46] J. Cong, T. Kong, and D. Z. Pan, "Buffer block planning for interconnect-driven floorplanning," in *Proc. Int. Conf. Computer-Aided Design*, Nov. 1999, pp. 358–363.
- [47] C. C. N. Chu and D. F. Wong, "Closed form solution to simultaneous buffer insertion/sizing and wire sizing," in *Proc. Int. Symp. Physical Design*, Apr. 1997, pp. 192–197.
- [48] J. Cong and D. Z. Pan. (1998) Interconnect Estimation and Planning for Deep Submicron Designs. Dept. Comput. Sci., Univ. California, Los Angeles, CA. [Online]. Available: <http://cadlab.cs.ucla.edu/~pan/publications/>
- [49] M. Pedram, N. Bhat, and E. S. Kuh, "Combining technology mapping and layout," *VLSI Design Int. J. Custom-Chip Design, Simulation, Testing*, vol. 5, no. 2, pp. 111–124, 1997.
- [50] A. Caldwell, A. B. Kahng, F. Koushanfar, H. Lu, I. L. Markov, M. R. Oliver, and D. Stroobandt, "Gtx: The gsrc technology extrapolation system," in *Proc. Design Automation Conf.*, June 2000.
- [51] J. Cong, "An interconnect-centric design flow for nanometer technologies," in *Proc. Int. Symp. VLSI Technology Systems Applications*, June 1999, pp. 54–57.



**Jason Cong** (S'88–M'90–SM'96–F'00) received the B.S. degree in computer science from Peking University, Beijing, China, in 1985, and the M.S. and Ph.D. degrees in computer science from the University of Illinois at Urbana–Champaign in 1987 and 1990, respectively.

He is currently a Professor and Codirector of the VLSI CAD Laboratory in the Computer Science Department of the University of California, Los Angeles. He has been appointed as a Guest Professor of Peking University since 2000. His research interests

include layout synthesis and logic synthesis for high-performance low-power VLSI circuits, design and optimization of high-speed VLSI interconnects, and FPGA synthesis and reconfigurable architectures. He has authored or coauthored over 140 research papers and led over 20 research projects supported by DARPA, the National Science Foundation, and a number of industrial sponsors. He served as the General Chair of the 1993 ACM/SIGDA Physical Design Workshop, the Program Chair and General Chair of the 1997 and 1998 International Symposium on FPGAs, respectively, Program Cochair of the 1999 International Symposium on Low-Power Electronics and Designs, and on program committees of many major conferences, including DAC, ICCAD, and ISCAS.

Dr. Cong received the Best Graduate Award from the Peking University in 1985, the Ross J. Martin Award for Excellence in Research from the University of Illinois at Urbana–Champaign in 1989, the National Science Foundation Young Investigator Award in 1993, the Northrop Outstanding Junior Faculty Research Award from the University of California at Los Angeles in 1993, the IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS Best Paper Award in 1995, the ACM SIGDA Meritorious Service Award in 1998, and an SRC Inventor Recognition Award in 2000. He is an Associate Editor of the IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS and *ACM Transactions on Design Automation of Electronic Systems*.



**Zhigang (David) Pan** (S'97–M'00) received the B.S. degree in geophysics from Peking University, Beijing, China, in 1992, and the M.S. degree in atmospheric sciences, the M.S. degree in computer science, and the Ph.D. degree in computer science all from the University of California at Los Angeles in 1994, 1998, and 2000, respectively.

He was with Magma Design Automation, Inc. during summer 1999 and with the IBM T. J. Watson Research Center during summer 2000. He is currently a Research Staff Member at the IBM T. J.

Watson Research Center, Yorktown Heights, NY. His current research interests are focused on VLSI interconnect modeling, synthesis, planning, and their interaction with physical design and logic synthesis.

Dr. Pan received the Best Paper in Session Award from the SRC Techcon 1998, the IBM Research Fellowship in 1999, the Dimitris Chorafas Foundation Award in 2000, and the SRC Inventor Recognition Award in 2000.