

Formulae for Performance Optimization and Their Applications to Interconnect-Driven Floorplanning *

Nicholas Chia-Yuan Chang¹, Yao-Wen Chang², and Iris Hui-Ru Jiang³

¹Global Unichip Corp., Hsinchu 300, Taiwan

²Department of Electrical Engineering & Graduate Institute of Electronics Engineering, National Taiwan University, Taipei 106, Taiwan

³Department of Electronics Engineering, National Chiao Tung University, Hsinchu 300, Taiwan

Abstract

As the process technology advances into the deep submicron era, interconnect plays a dominant role in determining circuit performance. Buffer insertion/sizing and wire sizing are the most effective and popular techniques to reduce interconnect delay and are traditionally applied to post-layout optimization. As the SIA technology roadmap predicts, however, the number of interconnections among different blocks and that of buffers inserted in a chip for performance optimization will grow dramatically [17, 18]. It is obviously infeasible to insert/size hundreds of thousands buffers or wires during the post-layout stage when most routing regions are occupied. Therefore, it is critical to incorporate buffer-block and wire-size planning into floorplanning to ensure timing closure and design convergence. In this paper, we first derive continuous buffer insertion/sizing and wire sizing formulae for performance optimization under a more accurate wire model, and then apply the formulae to interconnect-driven floorplanning that considers not only the buffer-block planning addressed in [7], but also wire-size planning. Experimental results show that our approach achieves an average success rate of 93% of nets meeting timing constraints and consumes an average extra area of only 0.8% over the given floorplan, compared with the average success rate of 73% and extra area of 1.20% resulted from recent work in [7].

1 Introduction

As physical dimensions shrink into the deep submicron era, interconnect plays a dominant role in determining circuit performance. The SIA'97 and SIA'99 roadmaps [17, 18] reveal that interconnect may slow down the progress along Moore's curve. Hence, many techniques for interconnect optimization have extensively been investigated in the literature recently, e.g., buffer insertion/sizing, wire sizing/spacing/shaping, topology construction, and better interconnect materials [6, 14]. Among these techniques, buffer insertion/sizing and wire sizing are the most effective and popular ones to reduce interconnect delay.

Because the intrinsic delay of a single wire is quadratically proportional to the wire length, inserting buffers breaks a wire into shorter segments, and this operation can make the delay grow more linearly. Buffer insertion can be classified into two categories: uniform and nonuniform buffer insertion. Uniform buffer insertion distributes buffers of the same size (homogeneous buffers) in wires; this technique has obtained much attention in the literature, e.g., [1, 10]. However, when the loading capacitance is much larger than the buffer capacitance, uniform buffer insertion may not be effective since the buffer of the last stage still suffers from a large propagation delay due to its insufficient driving capability. It is, therefore, necessary to use a chain of buffers of increasing sizes, i.e., nonuniform buffer insertion, to optimize delay if the loading capacitance is far greater than the buffer capacitance. Traditional nonuniform buffer insertion ignores the effects of wires and considers only buffers and loads. The optimum increasing factor (*stage ratio*) has been derived for this simplified circumstance. If the diffusion capacitance of a buffer is ignored, the optimum stage ratio for delay minimization is equal to e [14]. Nevertheless, the traditional nonuniform buffer insertion scheme

without considering wire delay is no longer accurate in the deep submicron technology because most delay is incurred in wires. In addition to buffer insertion, wire sizing [3, 8] increases wire widths to improve the driving capability for large loads or decreases wire widths to reduce the capacitive loading for weak drivers. The leverage of wire sizing can thus reduce interconnect delay.

Recently, simultaneous buffer insertion/sizing and wire sizing has been explored to some degree in the literature [4, 5]. Chu and Wong in [4] provided a continuous closed form to solve this problem. However, their wire model does not consider fringing capacitance. They later in [5] incorporated fringing capacitance into their wire model, but manipulated only in the discrete domain. Unifying the considerations of wires for uniform buffer insertion, buffers and loads for nonuniform buffer insertion, and wire sizing, we derive continuous buffer insertion/sizing and wire sizing formulae for performance optimization under a more accurate wire model. (See the summary in Table 1 for the comparison.)

	Fringing capacitance	Continuous buffer sizes	Continuous wire sizes
[4]		✓	✓
[5]	✓		
Our work	✓	✓	✓

Table 1: Comparison with the related previous works.

According to the discussion in [4], the optimal buffer insertion/sizing and wire sizing may occur when we insert buffers into a wire at equidistance, where the forms of buffer sizes and wire widths are geometric progression. On the other hand, inspired by [14], the optimal buffer insertion/sizing is achieved when the delay associated with each stage is the same. Hence, thin wires correspond to large buffers so that the delay of each stage is relatively balanced. In our model for buffer insertion/sizing and wire sizing, buffers are inserted at equidistance, and buffer sizes are increasing, while wire sizes are decreasing.

Most existing buffer inserting/sizing and wire sizing algorithms are intended for post-layout interconnect optimization. As the SIA technology roadmap predicts, however, the number of interconnections among different blocks and that of buffers inserted in a chip for performance optimization will grow dramatically [17, 18]. It is obviously infeasible to insert/size hundreds of thousands buffers or wires during the post-layout stage when most routing regions are occupied. Therefore, it is critical to incorporate buffer-block and wire-size planning into floorplanning to ensure timing closure and design convergence. Cong, Kong, and Pan in [7] and Sarkar, Sundaraman, and Koh in [15] presented pioneering works on buffer block planning for interconnect-driven floorplanning. Given a floorplan, they clustered buffers into blocks and placed these buffer blocks into the dead spaces and channels in the floorplan to reduce circuit delay by using the uniform buffer insertion scheme.

In this paper, we develop an algorithm that applies our buffer insertion/sizing and wire sizing formulae for interconnect-driven floorplanning. We consider not only the buffer block planning problem formulated in [7], but also wire-size planning to optimize circuit performance. Experimental results show that our algorithm achieves an average success rate of 93% of nets meeting timing constraints and consumes an

* This work was partially supported by the National Science Council of Taiwan ROC under Grant No. NSC 89-2215-E-009-117.

average extra area of only 0.8% over the given floorplan, compared with the average success rate of 73% and extra area of 1.20% resulted from recent work in [7].

This paper is organized as follows. Section 2 introduces some notations. Section 3 derives nonuniform buffer insertion/sizing formulae. Section 4 develops simultaneous buffer insertion/sizing and wire sizing formulae. Section 5 applies the formulae to buffer-block planning for interconnect-driven floorplanning. Finally, Section 6 shows experimental results.

2 Preliminaries

We use the following notations in this paper.

- h : the width (size) of a wire.
- r_0 : the sheet resistance of a wire.
- c_0 : the unit-sized area capacitance of a wire.
- c_f : the unit-length fringing capacitance of a wire.
- $c(h)$: the unit-length capacitance of a wire of width h ; i.e., $c(h) = c_0h + c_f$.
- T_{in} : the intrinsic delay of a buffer.
- c_b : the unit-sized capacitance of a buffer.
- r_b : the unit-sized resistance of a buffer.
- C_L : the capacitance of the load.
- R_d : the resistance of the driver.
- β : the stage (size) ratio between two consecutive buffers.
- ω : the stage (size) ratio between two consecutive wire segments.

Throughout the rest of this paper, we will apply the parameters listed in Table 2 to facilitate our technical discussions. This set of parameters is based on the $0.18\mu\text{m}$ technology in the SIA'97 roadmap and was used in [7].

Parameter	Description (unit)	Value
h	unit wire width (μm)	0.18
r_0	wire sheet resistance (Ω)	0.068
c_0	wire sheet area capacitance ($fF/\mu\text{m}^2$)	0.06
c_f	wire fringing capacitance of a wire ($fF/\mu\text{m}$)	0.064
$c(h)$	wire unit-length capacitance as $h = 0.9\mu\text{m}$ ($fF/\mu\text{m}$)	0.118
T_{in}	intrinsic delay for each buffer (ps)	36.4
C_L	load capacitance (fF)	23.4
R_d	driver resistance (Ω)	180
c_b	input capacitance of a minimum size buffer (fF)	23.4
r_b	output resistance of a minimum size buffer (Ω)	180

Table 2: Parameters of the $0.18\mu\text{m}$ technology in SIA'97.

Figure 1 shows our circuit models for buffers and wire segments. We use a switch-level RC circuit to model a buffer, with the consideration of the intrinsic buffer delay; we adopt the π -model to model a wire segment, with the consideration of fringing capacitance. We use the Elmore delay model [9] to compute delay.

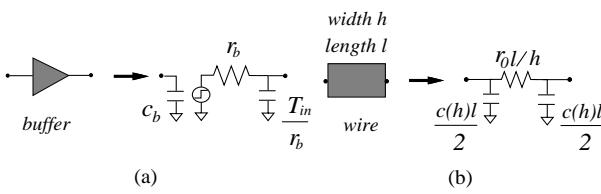


Figure 1: We model a buffer as a switch-level RC circuit and a wire segment as a π -model RC circuit, and apply the Elmore delay model for delay computation.

3 Nonuniform Buffer Insertion

Buffer insertion is currently considered as the most effective and popular way to optimize interconnect delay. In this section, we consider the problem of optimizing the interconnect delay by finding the best number of buffers to be inserted and the corresponding stage ratio. By “stage ratio,” we mean the size ratio of the $(i+1)$ -th buffer to the i -th one. As an example shown in Figure 2, we insert N buffers into a wire of length L at equidistance. Therefore, the wire is divided into $N+1$ segments (stages), and the length of each segment $l = \frac{L}{N+1}$.

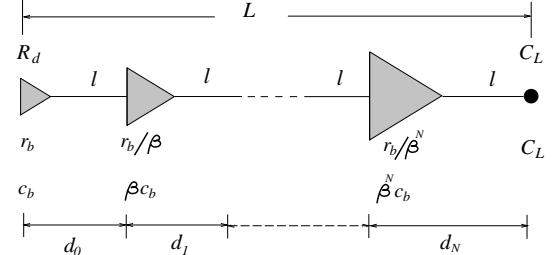


Figure 2: The model of nonuniform buffer insertion.

The *Nonuniform Buffer Insertion problem* is formulated as follows.

- **Problem:** The Nonuniform Buffer Insertion (NBI) Problem

Input: Wire length L , driver resistance R_d , load capacitance C_L , unit-sized area capacitance of a wire c_0 , unit-length fringing capacitance of a wire c_f , sheet resistance of a wire r_0 , input capacitance of a unit-sized buffer c_b , and resistance of a unit-sized buffer r_b .

Objective: Determine the stage ratio β for buffer sizes and the number of buffers N such that the wire delay is minimized.

The delay of each stage d_i , $0 \leq i \leq N$, is given as follows.

$$\begin{aligned} d_0 &= R_d(c_0h + c_f)l + \beta R_d c_b + \frac{1}{2}r_0 c_0 l^2 + \frac{r_0 c_f l^2}{2h} + \frac{\beta r_0 l c_b}{h}, \\ d_i &= T_{in} + \beta r_b c_b + \frac{1}{2}r_0 c_0 l^2 + \frac{r_0 c_f l^2}{2h} + \frac{r_b(c_0h + c_f)l}{\beta^i} \\ &\quad + \beta^{i+1} \frac{r_0 l c_b}{h}, \quad 1 \leq i \leq N-1; \\ d_N &= T_{in} + \frac{r_b C_L}{\beta^N} + \frac{r_0 c_0 l^2}{2} + \frac{r_0 c_f l^2}{2h} + \frac{r_b(c_0h + c_f)l}{\beta^N} + \frac{r_0 l C_L}{h}. \end{aligned}$$

Therefore, the total delay of the wire $D(N, \beta)$ is given by

$$\begin{aligned} D(N, \beta) &= R_d(c_0h + c_f)l + \beta R_d c_b + N T_{in} + \beta(N-1)r_b c_b \\ &\quad + \left(\frac{r_b}{\beta^N} + \frac{r_0 l}{h} \right) C_L + \frac{1}{2}(N+1) \left(r_0 c_0 l^2 + \frac{r_0 c_f l^2}{h} \right) \\ &\quad + \sum_{i=1}^N \left(\frac{r_b}{\beta^i} (c_0h + c_f)l + \frac{\beta^i c_b r_0 l}{h} \right). \end{aligned} \quad (1)$$

We have the following theorem to show the effect of buffer insertion/sizing on delay.

Theorem 1 *The delay of a wire is a convex function of the stage ratio β for buffer sizes and the number N of inserted buffers.*

Proof Sketch: Differentiating D_N twice with respect to β and N , we have

$$\begin{aligned} \frac{\partial^2 D(N, \beta)}{\partial \beta^2} &= \sum_{i=0}^N i(i+1) \frac{r_b(c_0h + c_f)l}{\beta^{i+2}} + \sum_{i=0}^N i(i-1) \frac{\beta^{i-2} r_0 l c_b}{h} \\ &\quad + N(N+1) \left(\frac{r_b C_L}{\beta^{N+1}} \right), \end{aligned}$$

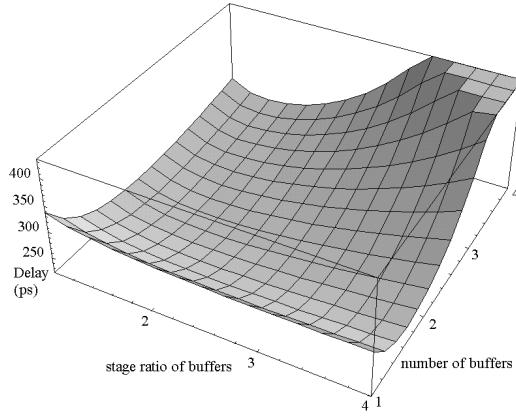


Figure 3: The effects of buffer insertion and sizing derived from Equation (1) based on the parameters listed in Table 2 and a wire of length 0.5 cm. The optimal number of buffers is around 4, and the stage ratio for buffers is around 2.

and

$$\begin{aligned} \frac{\partial^2 D(N, \beta)}{\partial N^2} &= \frac{2R_d(c_0h + c_f)l}{(N+1)^2} + \frac{r_0(c_0 + \frac{c_f}{h})l^2}{N+1} + \frac{r_bC_L}{\beta^{N-2}(ln\beta)^2} + \frac{2r_0lC_L}{h(N+1)^2} \\ &+ \frac{((ln\beta)(ln\beta-1)(N+1)\beta^N - 2\beta^N(ln\beta-1) - 2)r_0lc_b}{(\beta-1)(N+1)^2} \\ &+ \frac{((ln\beta)(ln\beta+1)(N+1)\beta^{-N} + 2\beta^{-N}(ln\beta+1) - 2)r_bc_0l}{(\beta^{-1}-1)(N+1)^2}. \end{aligned}$$

It can be shown that $\frac{\partial^2 D(N, \beta)}{\partial \beta^2}$, $\frac{\partial^2 D(N, \beta)}{\partial N^2}$, and $(\frac{\partial^2 D(N, \beta)}{\partial \beta^2}, \frac{\partial^2 D(N, \beta)}{\partial N^2})$ are greater than 0; the wire delay is thus always “curving upward”; in other words, the delay of a wire is a convex function of the stage ratio β and the number of inserted buffers N . \square

Since the delay function is convex, it is guaranteed that a local optimal solution equals the global optimum. Hence, we can apply any efficient search algorithm, such as the well-known gradient search procedure, to find a local optimal solution and thus the global optimum. Figure 3 shows the effects of the stage ratio for buffers and the number

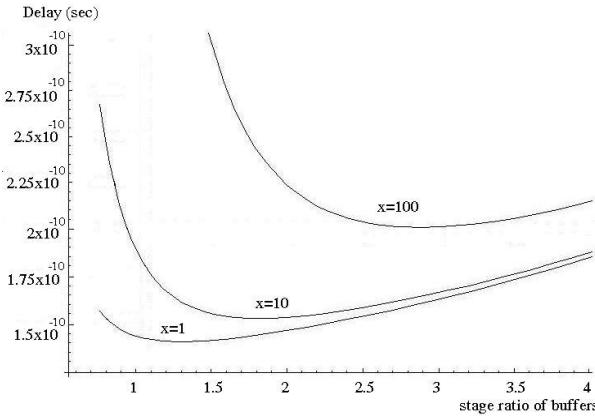


Figure 4: The delay functions for various $X = C_L/c_b$ based on the parameters shown in Table 2 (number of buffers: 3; wire length: 0.6 cm).

of buffers on the delay of a wire derived from Equation (1). In Figure 4, the delay of a wire is plotted as a function of the stage ratio for buffers based on the parameters shown in Table 2, three buffers inserted, and a wire of length 0.6 cm. Previous work shows that the optimum stage ratio for delay minimization is equal to e when the diffusion capacitance of a buffer and the delay of a wire are ignored [14]. However, diffusion capacitance and wire delay are of particular importance in the deep submicron technology. Figures 4 shows that the greater the loading capacitance, the greater the optimum stage ratio for buffers.

4 Simultaneous Buffer Insertion/Sizing and Wire Sizing

In this section, we intend to minimize the wire delay by simultaneous buffer insertion/sizing and wire sizing. *The Simultaneous Buffer Insertion/Sizing and Wire Sizing Problem* is formulated as follows. (Note that N is given here while it is unknown in the NBI problem.) Figure 5 shows our model for simultaneous buffer insertion/sizing and wire sizing. According to the discussion in [4], the optimal buffer insertion/sizing and wire sizing may occur when we insert buffers into a wire at equidistance, where the forms of buffer sizes and wire widths are geometric progression. On the other hand, inspired by [14], the optimal buffer insertion/sizing is achieved when the delay associated with each stage is the same. Hence, thin wires correspond to large buffers so that the delay of each stage is relatively balanced. In our model for buffer insertion/sizing and wire sizing, buffers are inserted at equidistance, and buffer sizes are increasing, while wire sizes are decreasing.

- **Problem:** The Simultaneous Buffer Insertion/Sizing and Wire Sizing (SBW) Problem

Input: Wire length L , driver resistance R_d , load capacitance C_L , unit-sized area capacitance of a wire c_0 , unit-length fringing capacitance of a wire c_f , sheet resistance of a wire r_0 , input capacitance of a unit-sized buffer c_b , resistance of a unit-sized buffer r_b , and the number of buffers inserted N .

Objective: Determine the stage ratio β for buffer sizes and the stage ratio ω for wire widths such that the wire delay is minimized.

In Section 3, we discussed the stage ratio β for buffers with a fixed wire width. Here, both wire widths and buffer sizes can be varied for the SBW problem.

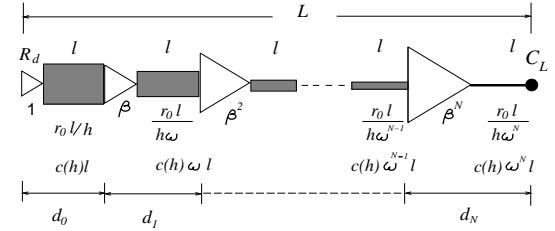


Figure 5: The model of nonuniform buffer insertion and wire sizing.

Inserting N buffers into a wire of length L at equidistance and considering the simultaneous buffer insertion/sizing and wire sizing, we can compute the delay from source R_d to sink C_L as follows (see Figure 5):

$$\begin{aligned} D_N(\omega, \beta) &= R_d(c_0h + c_f)l + \beta R_d c_b + NT_{in} + \beta(N-1)r_b c_b \\ &+ \frac{r_0 l C_L}{h \omega^N} + \frac{r_b}{\beta^N} C_L + \frac{1}{2}(N+1)(r_0 c_0 l^2 + \frac{r_0 c_f l^2}{h}) \\ &+ \sum_{i=1}^N \frac{r_b}{\beta^i} (c_0 \omega^i h l + c_f l) + \sum_{i=1}^N \frac{\beta^i}{\omega^i} \frac{r_0 l c_b}{h}. \quad (2) \end{aligned}$$

We have the following theorem.

Theorem 2 *The delay of a wire with N buffers inserted is a convex function of the stage ratio β for buffer sizes and the stage ratio ω for wire widths.*

Proof Sketch: Differentiating Equation (2) twice, we have

$$\begin{aligned} \frac{\partial^2 D_N(\omega, \beta)}{\partial \omega^2} &= \sum_{i=1}^N (\beta^{-i} r_b)(i(i-1)c_0 \omega^{i-2} h l + c_f l) \\ &+ \sum_{i=1}^N i(i+1)(\beta^i \omega^{-i-2}) \frac{r_0 l c_b}{h} + N(N+1)\omega^{N+2} \left(\frac{r_0 l C_L}{h} \right), \end{aligned}$$

and

$$\begin{aligned} \frac{\partial^2 D_N(\omega, \beta)}{\partial \beta^2} &= \sum_{i=1}^N i(i+1)\beta^{-(i+2)}(r_b c_0 \omega^i h l + r_b c_f l) \\ &+ \sum_{i=1}^N i(i-1)(\beta^{i-2} \omega^{-i}) \frac{r_0 l c_b}{h} + N(N+1)\beta^{-(N+2)} r_b C_L. \end{aligned}$$

It is not difficult to show that $\frac{\partial^2 D_N(\omega, \beta)}{\partial \omega^2}$, $\frac{\partial^2 D_N(\omega, \beta)}{\partial \beta^2}$, and $(\frac{\partial^2 D_N(\omega, \beta)}{\partial \omega^2} \frac{\partial^2 D_N(\omega, \beta)}{\partial \beta^2} - \frac{\partial^2 D_N(\omega, \beta)}{\partial \omega \partial \beta})$ are greater than 0. Thus the wire delay function D_N is always “curving upward”; in other words, the function is convex. \square

Figure 6 shows an example for Equation (2) based on the parameters listed in Table 2; it is obvious that the curve is convex. Let

$$\gamma = \sum_{i=1}^N \frac{\omega^{i+1}}{\beta^i} = \begin{cases} \frac{\omega}{\omega-\beta} \left(\frac{\omega^{N+1}-\beta^{N+1}}{\beta^N} \right) & \text{if } \omega \neq 1, \beta \neq 1 \\ \frac{\omega^{N+1}-\beta^{N+1}}{\beta^N-\beta} & \text{if } \omega = 1 \\ N & \text{if } \omega = \beta = 1, \end{cases} \quad (3)$$

$$\delta = \sum_{i=1}^N \frac{\beta^i}{\omega^i} = \begin{cases} \frac{1}{\beta-\omega} \left(\frac{\beta^{N+1}-\omega^{N+1}}{\omega^N} \right) & \text{if } \omega \neq 1, \beta \neq 1 \\ \frac{\beta^{N+1}-1}{\beta-1} & \text{if } \omega = 1 \\ N & \text{if } \omega = \beta = 1, \end{cases} \quad (4)$$

and

$$\theta = \sum_{i=1}^N \frac{1}{\beta^i} = N \quad \text{if } \beta = 1, \quad (5)$$

We have

$$\begin{aligned} D_N(\omega, \beta) &= R_d c_0 l + R_d c_b + N T_{in} + \beta(N-1)r_b c_b + \frac{(N+1)r_0 c_0 l^2}{2} \\ &+ \frac{r_b C_L}{\beta^{N-1}} + \frac{r_0 l C_L}{\omega^N} + (\gamma r_b c_0 h l + \theta r_b c_f l) + \delta \frac{r_0 l c_b}{h}. \end{aligned}$$

When ω equals 1, Equation (2) reduces to Equation (1) derived in Section 3, which considers only a uniform wire size. When β and ω are both equal to 1 and the fringing capacitance of wires are ignored, Equation (2) reduces to the optimal uniform buffer insertion equation derived in [1].

We can solve the SBW problem by using convex programming to obtain the optimal ratios for β and ω . Figure 6 shows the relationship among the delay $D_N(\omega, \beta)$, stage ratio for buffers β , and stage ratio for wire widths ω . The figure reveals that the optimal wire width is tapering while the optimal buffer size is increasing. This phenomenon coincides with the statements mentioned in previous work.

5 Application to Interconnect-Driven Floorplanning

In this section, we develop an algorithm that applies our buffer insertion/sizing and wire sizing formulae for interconnect-driven floorplanning. We consider not only the *buffer block planning* problem formulated in [7], but also wire size planning to optimize circuit performance.

5.1 Feasible Region for Buffer Insertion

We shall first discuss the feasible region for inserting buffers. According to [7, 15], the *feasible region (FR)* of a buffer is the maximum region where it can be placed while the net satisfies its timing constraint T_{req} , and other buffers along the net are placed within their respective feasible regions. The feasible regions of buffers are independent of each other. The work in [7, 15] considered only buffer insertion. Therefore, we shall derive feasible regions for the SBW problem with nonuniform buffers and wire sizes (i.e., consider β and ω). Recall that, in Section 4,

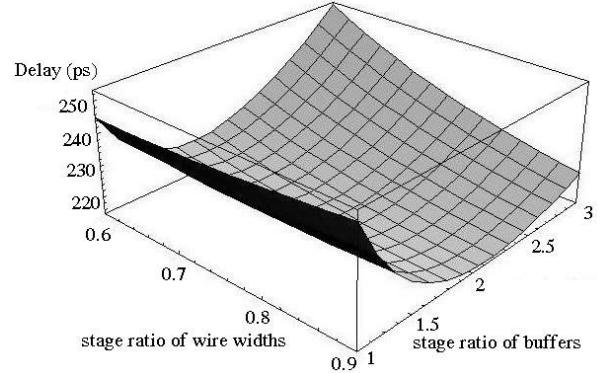


Figure 6: The effects of buffer sizing and wire width on delay ($N = 3$, $L = 0.5$ cm).

the optimal ω and β can be obtained by any efficient search algorithm, such as the well-known gradient search procedure.

Let x_i denote the distance from the driver to the i -th buffer. We have the following theorem to compute the feasible region for each buffer.

Theorem 3 For a wire of length L and with N buffers inserted, the feasible region for the i -th buffer ($1 \leq i \leq N$) is $x_i \in [x_{min}(N, i), x_{max}(N, i)]$ with

$$x_{min}(N, i) = \max \left(0, \frac{-\kappa_2 - \sqrt{\kappa_2^2 - 4\kappa_1\kappa_3}}{2\kappa_1} \right), \text{ and}$$

$$x_{max}(N, i) = \min \left(L, \frac{-\kappa_2 + \sqrt{\kappa_2^2 - 4\kappa_1\kappa_3}}{2\kappa_1} \right),$$

where κ_1 , κ_2 and κ_3 are functions of N and i with

$$\begin{aligned} \kappa_1(N, i) &= \frac{(N+1)r_0(c_0 + \frac{c_f}{h})}{2} \\ \kappa_2(N, i) &= \left(\sum_{i=0}^{N-1} \frac{\beta^i}{\omega^i} \right) \frac{r_0 c_b}{h} + \left(\sum_{i=0}^{N-1} \frac{\omega^{i+1}}{\beta^i} \right) r_b(c_0 h + c_f) \\ &+ R_d(c_0 h + c_f) + \frac{r_0 C_L}{h \omega^N} \\ \kappa_3(N, i) &= N T_{in} - T_{req} + R_d c_b + \beta(N-1)r_b c_b + \frac{r_b C_L}{\beta^{N-1}}. \end{aligned}$$

Note that we must have $\kappa_2^2 - 4\kappa_1\kappa_3 \geq 0$; otherwise, there exists no feasible region for inserting the i -th buffer, and the given floorplan/timing constraint must be changed. Figure 7 shows a 2-D feasible region with existing blocks acting as obstacles for buffer insertion.

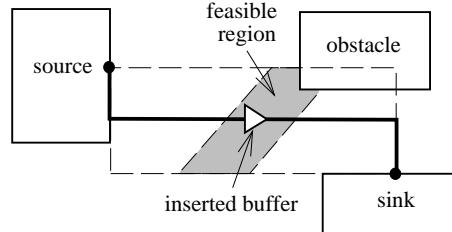


Figure 7: A 2-D feasible region (shaded region) with an existing block acting as an obstacle for buffer insertion.

5.2 Algorithm

In this subsection, we present an algorithm to demonstrate how to apply our formulae to interconnect-driven floorplanning.

In Figure 8, delay is plotted as a function of the number of inserted buffers. As shown in the figure, simultaneous buffer insertion and sizing (curve B) is clearly much more effective than buffer insertion alone (curve A) used in [7, 15] for timing optimization. Unlike the recent works [7, 15] that considered only uniform buffer insertion, our algorithm MBBP first tries to find available dead spaces in the feasible regions to insert and size buffers to reduce the delay for a net violating its timing constraint. If it cannot meet the timing constraint, MBBP will then try to size the net to improve the performance. At last it will resort to the simultaneous buffer insertion/sizing and wire sizing technique to further reduce the delay if none of the previous techniques can make the delay within the timing bound.

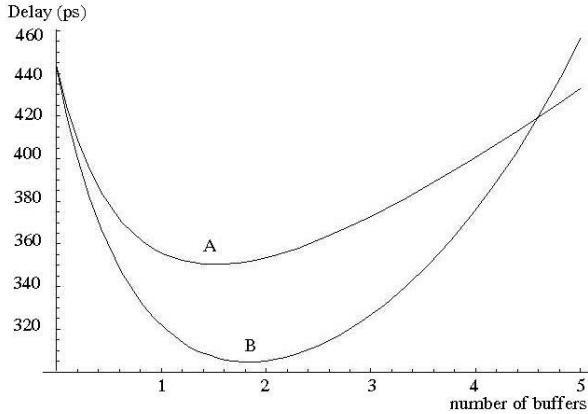


Figure 8: Delay comparison between uniform buffer insertion [1] (Curve A) and nonuniform one (Curve B). [7] and [15] applied the uniform buffer insertion formulated in [1]. Curve B is based on Equation (1) and $\beta = 2$. The wire length is equal to 0.6 cm, and other parameters are given in Table 2 for both curves.

```

Algorithm: Multi-phase Buffer Block Planning (MBBP)
Input: A floorplan  $F$  and a set of 2-pin nets.
Output:  $B_b$ —locations of buffer blocks.

1 Build the horizontal and vertical polar graphs for  $F$ ;
2 Build tile data structure;
3 For each tile, compute its area slack;
4 while (there exists a net violating its timing constraint) do
5   if (buffer insertion/sizing can meet the timing constraint
6       and there exists tiles in its FR)
7      $tile(n) \leftarrow Pick\_Tiles();$ 
8     Nonuniform_Buffer_Insertion( $tile(n)$ );
9     Update the polar graphs;
10    else if (wire sizing can meet the timing constraint and
11        there exists no tile in its feasible region)
12      Wire_Sizing();
13    else if (delay still violates the timing constraint)
14       $tile(n) \leftarrow Pick\_Tiles();$ 
15      Simultaneous_Buffer-IS_Wire-S( $tile(n)$ );
16      Update the polar graphs;
17 until (no net can be improved)

```

Figure 9: The Multi-phase Buffer Block Planning (MBBP) algorithm.

Figure 9 summarizes our MBBP algorithm. The first three lines are the data preparation stages, which are the same as those in [7]. We first build the horizontal and vertical polar graphs [16], denoted by G_H and G_V , for the given floorplan F . G_H and G_V are directed acyclic graphs.

For G_H (G_V), each vertical (horizontal) channel is represented by a vertex. A directed edge (u, v) denotes that u is the left (top) side of a block, and v is the right (bottom) side of the same block. Each edge is associated with a positive weight denoting the width (height) of the block. The longest path in G_H (G_V) is equal to the minimum required chip width (height). Therefore, G_H and G_V can be used to determine the area of the smallest enveloping rectangle of F . During processing, MBBP might need to shift blocks to make rooms for buffer blocks, and thus lengthen horizontal or vertical channels. Each channel and the dead space after floorplanning are divided into a set of rectangular tiles [7, 15] to facilitate data manipulation. For each tile, its area slack can be computed from the longest path path in G_H or G_V .

For each net, MBBP will try one of the following three procedures (in the order) until the net satisfies its timing constraint, if possible.

- Procedure 1: Buffer Insertion and Sizing (Line 8: Nonuniform_Buffer_Insertion())

If buffer insertion and sizing can satisfy the timing constraint and there exist enough tiles within dead space for buffer insertion, MBBP will insert all required buffers into the FRs in these tiles. Note that the total area will not increase for this operation.
- Procedure 2: Wire Sizing (Line 12: Wire_Sizing())

If there are no tiles within its FR, MBBP will try to size the wire to meet the timing constraint.
- Procedure 3: Simultaneous Buffer Insertion/Sizing and Wire Sizing (Line 15: Simultaneous_Buffer-IS_Wire-S())

If it still cannot meet the timing constraint after applying the aforementioned procedures, MBBP will use Equation (2) to size buffers and wires simultaneously to reduce the delay. If there exists no tile with dead space for buffer insertion, MBBP will expand the channel where its FR locates by one-tile width to make rooms for the inserted buffers. This operation certainly will increase the chip area.

After inserting buffers into tiles, MBBP has to update the polar graphs for further processing. If channel expansion occurs, MBBP will also update the timing information of the nets that have successfully been processed before. If the net cannot satisfy its timing requirement due to channel expansion, MBBP will process the net again. MBBP halts when no more nets can be improved by the above three procedures.

6 Experimental Results

The MBBP algorithm was implemented in the C++ language on a Sun UltraSPARC I workstation and tested on the benchmark circuits used in [7] (see Table 3 for the circuits). We compared our MBBP method with the buffer block planning algorithms BBP presented in [7] based on the data and parameters generated in [7]. For example, the parameters for interconnects and buffers were based on the $0.18\mu m$ technology given in the SIA'97 roadmap [17] (see Table 2 for the parameters), the nets were all 2-pin connections and the power/ground and single-pin nets were excluded, and the delay budgets were randomly generated by the authors in [7] from $1.05\text{--}1.20 T_{opt}$. The experiments of [15] are based on different parameters and delay bounds, so we compared results with only [7].¹

Table 4 gives the results obtained from the BBP and MBBP algorithms. Column 2 in Table 4 lists the number of nets (2-pin connections) that meet the timing constraint (# nets meet), the number of nets that do not meet the timing constraint (Not meet), and the total number of nets (Tot. # nets); Column 3 shows the percentage of the nets that meet the timing constraint (% Meet); Column 4 lists the total number of buffers inserted into the 2-pin nets (# buffers); Column 5 reports the increased area in percentage after the planning (Extra area (%)), which is computed by $(\text{new chip area} - \text{original chip area})/\text{original chip area}$; Column 6 gives the CPU times (in second) for the BBP, and MBBP algorithms based on an Intel Pentium-II, and a SUN UltraSPARC I machines, respectively. Note that we used the same delay budgets as those used in [7]

¹We were provided with 10 out of the 11 benchmark circuits used in [7].

Circuit	# modules	# nets	# pads	# 2-pin nets
apte	9	97	73	172
xerox	10	203	2	455
hp	11	83	45	226
ami33	33	123	43	363
ami49	49	408	22	545
playout	62	2506	192	2150
ac3	27	212	75	446
xc5	50	1005	2	2275
hc7	77	449	51	1450
a9c3	147	1202	22	1613

Table 3: Statistics of the MCNC benchmark circuits.

while those used were randomly generated by the authors (from the same interval as that in [7], i.e., $[1.05T_{opt}, 1.20T_{opt}]$).

Circuit algorithm	# nets meet / Not meet / Tot. # nets	% Meet	# buffers	Extra area (%)	CPU* time (sec)
apte	102/ 70/ 172	59.3	185	0.69	0.23
BBP	148/ 24/ 172	86.0	24	0.80	2.15
MBBP					
xerox	260/ 195/ 455	57.1	399	1.38	0.53
BBP	388/ 67/ 455	85.3	192	0.12	4.15
MBBP					
hp	131/ 95/ 226	58.0	280	1.24	0.48
BBP	198/ 28/ 226	87.6	49	1.39	1.56
MBBP					
ami33	305/ 58/ 363	84.0	667	1.36	1.63
BBP	360/ 3/ 363	99.2	49	4.18	3.38
MBBP					
ami49	412/ 133/ 545	75.6	946	0.78	3.25
BBP	542/ 3/ 545	99.4	202	0.11	7.00
MBBP					
playout	1533/ 617/ 2150	71.3	4263	0.84	13.98
BBP	2052/ 98/ 2150	95.4	242	0.13	54.38
MBBP					
ac3	369/ 77/ 446	82.7	733	1.11	1.39
BBP	415/ 31/ 446	93.0	131	0.04	3.38
MBBP					
xc5	1739/ 536/ 2275	76.4	3210	1.79	10.16
BBP	2216/ 59/ 2275	97.4	59	0.00	19.93
MBBP					
hc7	1068/ 382/ 1450	73.7	2693	1.92	15.88
BBP	1385/ 65/ 1450	95.5	184	1.22	9.98
MBBP					
a9c3	1446/ 167/ 1613	89.6	4265	0.89	29.20
BBP	1507/ 106/ 1613	94.0	277	0.00	25.67
MBBP					
Average				1.20	
BBP		72.77		0.80	
MBBP		93.28			

Table 4: Comparison with BBP. (*CPU times are based on different machines.)

We summarize the results shown in Table 4 as follows.

- Percentage of nets meeting timing constraints: The MBBP algorithm achieves an average success rate of 93% of nets meeting timing constraints, compared with the average success rate of 73% resulted from BBP [7].
- Extra area and number of buffers: MBBP consumes an average extra area of only 0.8% over the original floorplan, compared with extra area of 1.2% resulted from BBP [7]. A smaller number of buffers is used by nonuniform buffers than that by uniform buffers. (Of course, the sizes of the uniform buffers and the nonuniform ones are not the same.) Typically, only 1–4 buffers were inserted in a net to meet the timing constraints.
- CPU time: Although the CPU times reported in Table 4 are based on different machines, we still have the observation that the average runtime of MBBP is greater than BBP.

The results reveal that the buffer/wire sizing formulae are much more effective than previous work in improving timing performance by inserting even smaller numbers of buffers.

7 Conclusion

We have presented in this paper the formulae for performance optimization by sizing circuit components (buffers and wires) with fringing capacitance consideration and applied the formulae to buffer-block and wire-size planning for interconnect-driven floorplanning. Experimental results have revealed the effectiveness of our approaches.

Acknowledgments

We would like to thank Prof. Jason Cong, Mr. Tianming Kong, and Dr. David Zhigang Pan for providing us with the benchmark circuits. Special thanks go to Mr. Tianming Kong and Dr. David Zhigang Pan for their kind, prompt assistance and very helpful explanations on their data.

References

- [1] C. J. Alpert and A. Devgan, “Wire Segmenting for Improved Buffer Insertion,” *Proc. 34th DAC*, pp. 588–593, June 1997.
- [2] H. B. Bakoglu, *Circuit, Interconnections, and Packaging for VLSI*, Addison-Wesley, 1990.
- [3] C. P. Chen, Y. P. Chen, and D. F. Wong, “Optimal Wire-Sizing Formula Under the Elmore Delay Model,” *Proc. 33rd DAC*, pp. 487–490, June 1996.
- [4] C. C. N. Chu and D. F. Wong, “Closed Form Solution to Simultaneous Buffer Insertion/Sizing and Wire Sizing,” *Proc. ISPD*, pp. 192–197, Apr. 1997.
- [5] C. C. N. Chu and D. F. Wong, “A Quadratic Programming Approach to Simultaneous Buffer Insertion and Wire Sizing,” *IEEE TCAD*, vol. 18, No. 6, pp. 787–798, June 1999.
- [6] J. Cong et al., “Performance Optimization of VLSI Interconnect Layout,” *Integration, the VLSI Journal*, vol. 21, pp. 1–94, 1996.
- [7] J. Cong et al., “Buffer Block Planning for Interconnect-Driven Floorplanning,” *Proc. ICCAD*, pp. 358–363, Nov. 1999.
- [8] J. Cong and K. S. Leung, “Optimal Wiresizing Under the Distributed Elmore Delay Model,” *Proc. ICCAD*, pp. 110–114, Nov. 1993.
- [9] W. C. Elmore, “The Transient Response of Damped Linear Networks with Particular Regard to Wide Band Amplifiers,” *J. Applied Physics*, vol. 19, pp. 55–63, 1948.
- [10] L. P. P. P. van Ginneken, “Buffer Placement in Distributed RC-tree Network for Minimal Elmore Delay,” *Proc. ISCAS*, pp. 865–868, 1990.
- [11] F. S. Hillier and G. J. Lieberman *Introduction to Operations Research*, McGraw-Hill, 1995.
- [12] M. Kang et al., “Delay Bounded Buffered Tree Construction for Timing Driven Floorplanning,” *Proc. ICCAD*, pp. 707–712, Nov. 1997.
- [13] D. Li et al., “A Repeater-Optimization Methodology for Deep Sub-Micron, High-Performance Processors,” *Proc. ICCAD*, pp. 726–731, Nov. 1997.
- [14] J. M. Rabaey, *Digital Integrated Circuits: A Design Perspective*, Prentice Hall, 1996.
- [15] P. Sarkar et al., “Routability-Driven Repeater Block Planning for Interconnect-Centric Floorplanning,” *Proc. ISPD*, 2000.
- [16] S. Sait and H. Youssef, *VLSI Physical Design Automation*, IEEE Press, 1995.
- [17] Semiconductor Industry Association, *National Technology Roadmap for Semiconductors*, 1997.
- [18] Semiconductor Industry Association, *International Technology Roadmap for Semiconductors 1999 Edition*, Nov. 1999.