Power-Delay Optimization in VLSI Microprocessors by Wire Spacing

KONSTANTIN MOISEEV
Technion
SHMUEL WIMER
Intel Corporation
and
AVINOAM KOLODNY
Technion

The problem of optimal space allocation among interconnect wires in a VLSI layout, in order to minimize the switching power consumption and the average signal delay, is addressed in this article. We define a Weighted Power-Delay Sum (WPDS) objective function and derive necessary and sufficient conditions for the existence of optimal interwire space allocation, based on the notion of capacitance density. At the optimum, every wire must be in equilibrium of its line-to-line weighted capacitance density on its two opposite sides, and the WPDS of the whole circuit is minimal if and only if capacitance density is uniformly distributed across the entire layout. This condition is shown to be equivalent to all paths of the layout cross-capacitance graph having the same length and all cuts having the same flow. An implementation which has been used in the design of a recent commercial high-end microprocessor and yielded 17% power reduction and 9% delay reduction in top-level interconnects is presented.

Categories and Subject Descriptors: B.7.2 [**Integrated Circuits**]: Design Aids—*Layout placement and routing*; J.6 [**Computer Applications**]: Computer-Aided Engineering—*Computer-aided design*

General Terms: Algorithms, Design

Additional Key Words and Phrases: Wire spacing, power optimization, delay-optimization, interconnect optimization

ACM Reference Format:

Moiseev, K., Wimer S., and Kolodny A. 2009. Power-Delay optimization in VLSI microprocessors by wire spacing. ACM Trans. Des. Autom. Electron. Syst. 14, 4, Article 55 (August 2009), 28 pages. DOI = 10.1145/1562514.1562523 http://doi.acm.org/10.1145/1562514.1562523

Authors' addresses: K. Moiseev, Department of Electrical Engineering, Technion—Israel Institute of Technology, 33000, Haifa, Israel; S. Wimer, Intel Corporation, Israel Development Center, 31015, Haifa, Israel; A. Kolodny, Department of Electrical Engineering, Technion—Israel Institute of Technology, 33000, Haifa, Israel; email: kolodny@ee.technion.il.

DOI 10.1145/1562514.1562523 http://doi.acm.org/10.1145/1562514.1562523

1. INTRODUCTION

Power-delay optimization has received increased attention in the last decade. The reason for this is that power dissipation has become a significant factor in the design of new microprocessors and other digital products. The main reason for increased power consumption is the growing logic complexity, with integration of multiple computational cores on a single die. The dissipation of power has become a major concern because of the growing awareness to environmental heating, the drive to deliver lighter mobile computers with longer battery life, and the emerging demand for portable consumer electronic products. Hence, new design methods for reducing power are sought by the industry, and every opportunity to contribute to the power saving is considered. Power reduction methods cannot neglect timing constraints imposed on the circuits; and therefore simultaneous optimization of power and delay has special importance in the design of modern circuits.

Power reduction was addressed at various design levels [Borkar 2001; Devadas et al. 1995], from architecture and system level through RTL synthesis, signal encoding, circuit implementation, and layout implementation, which is the focus of this article. The interconnect power dissipated because of charging and discharging wire capacitances is a dominant component in processors [Magen et al 2004]. A typical breakdown of dynamic power dissipation of a high-end microprocessor designed in 65 nanometer process technology is illustrated in Figure 1, indicating that global wires at the top metal layers generate 20% of the total dynamic power, and about half of this power is due to cross-coupling between adjacent wires in the same metal layer. In a similar manner, cross-capacitances between wires in interconnect structures have a major effect on circuit timing. The wire delays at the top metal layers are typically dominated by cross-capacitances between adjacent wires, since the aspect ratio of wire thickness to wire width tends to grow with nonuniform technology scaling [Mui et al. 2004]. Therefore, delays can be optimized by allocation of interwire spaces. We show in this article how delay and power can be significantly reduced by optimizing interwire spacing in the completed layout.

Commercial routing tools and manual artwork of mask designers tend to produce congested wires. Tools and humans do not always take advantage of the entire area available for layout implementation. This is quite natural, since routing is usually a sequential process. Therefore, the more area is saved at any routing step, the better is the chance to complete all required interconnections [Li et al. 2007]. However, this approach results in nonuniform area utilization, leaving islands of "white areas" in the layout. Unfortunately, such inefficiency can be observed only after the routing job is done, as shown in Figure 2.

Based on this observation, we propose to eliminate the white space by spreading-out wires in the final layout, using a post-processing algorithm. This operation balances interwire spaces in order to reduce excessive capacitances, save power, and decrease wire delays. A similar post-processing approach has been employed in wire spacing for improved manufacturability. It is assumed that interconnects have been routed (manually or automatically), and their relative locations are not subject to any change (i.e., layout topology is unchanged).

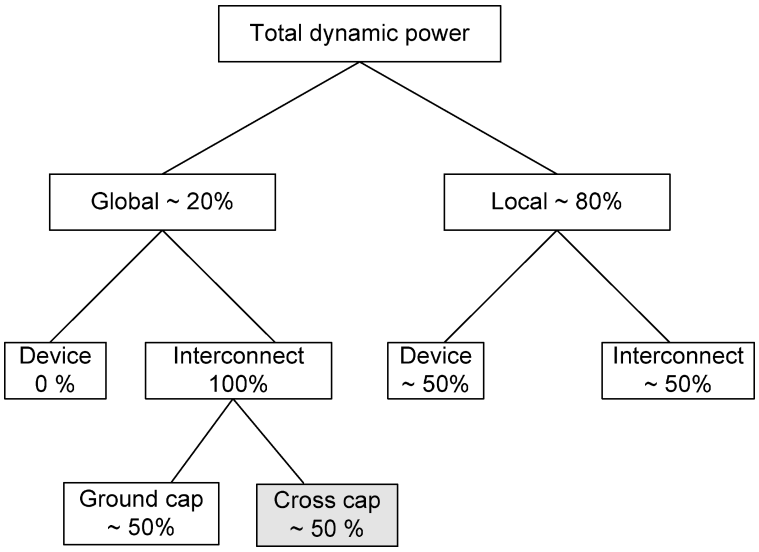

Fig. 1. Typical breakdown of dynamic power into local blocks and global interconnects. As can be seen, the cross-capacitances between global wires at the top routing layers contribute about 10% of the total dynamic power.

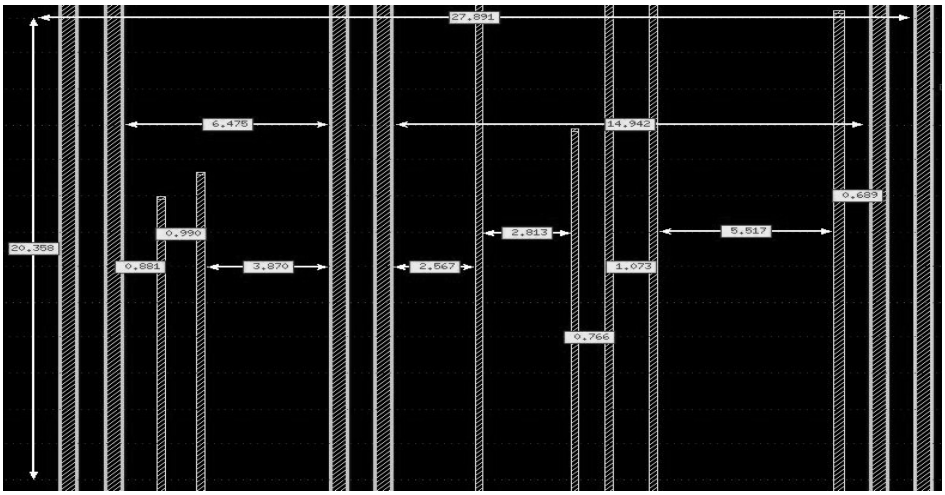

Fig. 2. An example layout of 7th metal layer taken from a high-end microprocessor. The wide wires are VCC/VSS and are fixed. The narrow wires are signals routed automatically. The figure demonstrates the amount of white space found in the layout and its inefficient distribution among signal wires.

It is also assumed that wire widths have been set to satisfy signal delay and other design goals such as reliability, and shield wires have been inserted to eliminate crosstalk noise on sensitive nodes. Hence, the method aims only to modify line-to-line capacitance densities across the whole layout in each of the top-level interconnect layers.

Design optimization by wire spacing has been discussed by many authors, for different purposes: Signal delay optimization [Cong et al. 2001; He et al. 1998; Wimer et al. 2006; Hanchate et al. 2006], power consumption minimization [Macii et al. 2003], cross-coupling noise reduction [Chaudhary et al. 1993; Saxena et al. 2000], and yield enhancement [Chiluvuri 1995] are just a few. The authors in Macii et al. [2003], Chaudhary et al. [1993], and Saxena et al. [2000] used local optimization. The optimization approach in this article is reminiscent of that in Saxena et al. [2000] in the sense that both rely on the convexity of line-to-line cross-coupling capacitance. However, unlike Macii et al. [2003], Chaudhary et al. [1993], and Saxena et al. [2000] which treat the problem locally, this article looks at the entire layout at once, and finds a provable global optimal solution. An iterative solution was used in Saxena et al. [2000] to find the optimal spacing for a single wire. Here we deal with a global problem involving thousands of wires simultaneously. The authors of Saxena et al. [2000] used convexity arguments to prove the existence of minimum cross-coupling noise in a single net, followed by a method to find the minimum without solving explicitly any cross-coupling noise equations. They further proposed improvement of noise immunity by local perturbations of signal wires. Cross-coupling noise, which is a "local" phenomenon, imposes a local optimization problem. In contrast, dynamic power consumption is a cumulative effect, thus a global solution for all the wires is required, which is the essence of the present article. Another difference is that the solution in Saxena et al. [2000] addresses two-dimensional routing for channel and switchbox routing styles. This work addresses the simultaneous optimization of the entire top-level microprocessor routing comprising many thousands of nets. Wire spacing optimization in the global routing layers of a processor is a collection of several, almost independent, one-dimensional problems. We exploit the one-dimensionality and the independency to obtain an effective global optimization approach.

Another important difference of this article from previous works discussing layout optimization by wire spacing is the simultaneous consideration of power and delay. We demonstrate that signal power and delay behave similarly with respect to interwire spaces and define a new Weighted Power-Delay Sum (WPDS) optimization problem. This formulation allows a global view of interconnect power and interconnect delay, while taking into account the delay criticality of individual wires.

The rest of the article is organized as follows. In the next section, circuit and layout models are presented. A necessary and sufficient WPDS minimization condition for each wire is proven in Section 3. In Section 4, a graph model of wire spacing and line-to-line capacitance is introduced, and is used to prove that the weighted capacitance density must be constant for all the wires in each layer at the global minimum WPDS solution. An iterative algorithm that guarantees convergence to the optimum is presented in Section 5. Practical considerations of power-delay optimization are discusses in Section 6. Results obtained for a recent high-end microprocessor designed in 65 nanometer process technology are presented in Section 7 and the article is concluded in Section 8.

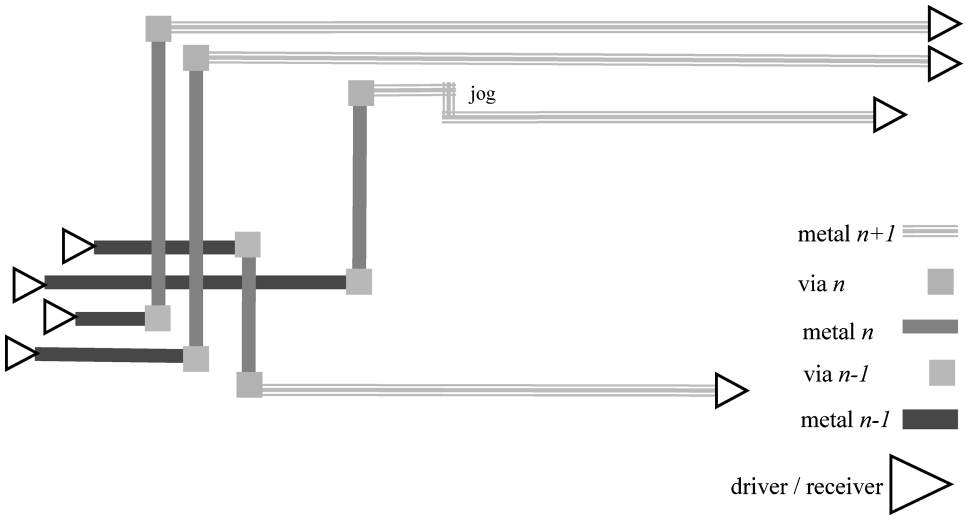

Fig. 3. Typical interconnect patterns: A driver transmits a signal which propagates through interconnecting wires on various layers. Consecutive layers route wires in alternating orthogonal directions. Connections from layer to layer are made by vias. Some wires may have jogs.

2. INTERCONNECT MODELING ASSUMPTIONS

The interconnecting wires at high metal layers typically run in alternating orthogonal directions, for example, wires residing in even layers are vertical and wires in odd layers are horizontal, as shown in Figure 3. Sometimes wires going in the main layer direction are connected by short jogs in the perpendicular direction. Such jogs are rarely used in high metal layers and they are ignored in the optimization discussion.

Spacing optimization is carried out at each layer independently of the other layers as follows: Let the vertical wires of an even layer l be subject to optimization. Connectivity must be maintained under any horizontal shift of vertical wires. As shown in Figure 3, shifting wires in one layer doesn't affect spacing of the orthogonal wires in the layers above it and below it. The lengths of horizontal wires in layers $l-1$ and $l+1$ usually reach hundreds of microns, while the typical wire shift during the optimization in layer l is less than a micron. Thus, lengths of horizontal wires in the adjacent layers usually change by less than 1%. The statistical average of these small changes is zero, such that these variations are negligible for all practical cases. Odd layers behave similarly.

Without loss of generality, we limit the following discussion to even (vertical) layers only.

The model we use to derive optimal spacing conditions is shown in Figure 4. There, n wires corresponding to signals $\sigma_1, \ldots \sigma_n$ run in parallel and the entire bundle is shielded on both sides by power supply wires, which are not allowed to move. The two side shield nets σ_0 and σ_{n+1} do not make logical transitions and are not connected to any driver.

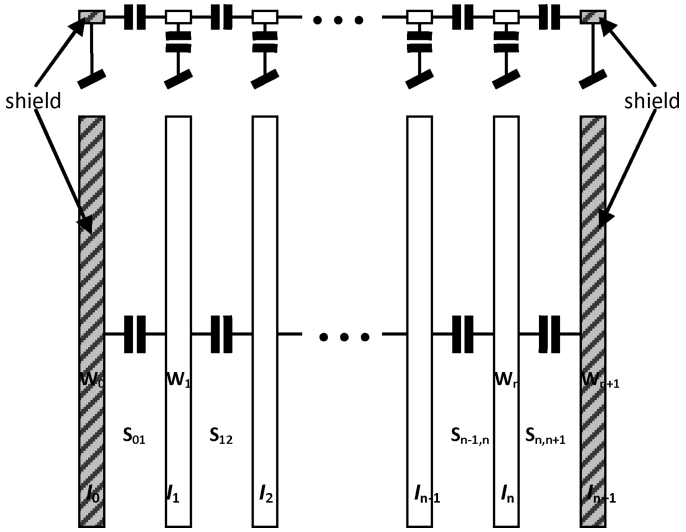

Fig. 4. Fundamental cross coupling and ground capacitance. Wires run in parallel and the entire bundle is shielded on both sides by wires connected to ground.

Assuming full voltage swing, the dynamic power consumed by toggling signal σ_i is given by

$$P_i = \alpha_i \left(C_i^a + kC_i^{ll}\right) V_{dd}^2 f, \tag{1}$$

where V_{dd} is supply voltage and f is the clock frequency. C_i^a is the wire capacitance to ground planes above and below and C_i^{ll} is the nominal line-to-line capacitance to other wires at the same routing layer. α_i denotes the amount of signal's switching relative to the clock signal, called the signal's *activity factor* [Genossar et al. 2003], ranging from $\alpha_i = 0$ if it never switches (e.g., shields or power delivery wires), to $\alpha_i = 1$, if it toggles twice at every cycle (e.g., clocks). This model of dynamic power is broadly used in the industry and provides good correlation to silicon. Signal activity factors are derived by using an industrial-power simulator which checks the signal activity in different scenarios, and then averaging activities over all cases [Bakoglu 1990; Genossar et al. 2003]. The power contributed by the line-to-line capacitance between σ_i and σ_j depends on α_i, α_j and the Miller Coupling Factor (MCF) between σ_i and σ_j (denoted by k in Eq. (1)). According to Miller's theorem, the simultaneous switching of two signals in identical and opposite directions yields MCF of 0 or 2, respectively, or -1 to 3 if worst-case transition slopes are assumed [Chen et al. 2000]. Assuming that the signals are logically independent, simultaneous transitions in identical and opposite directions are equally likely. Hence, the energy dissipated over multiple simultaneous switching transitions can be calculated using the average MCF for power, which is equal to 1 (same as the MCF for nonsimultaneous switching, when the adjacent signal is stable). Under this assumption the power contributed by the line-to-line capacitance between σ_i and σ_j is proportional to $\alpha_i + \alpha_j$. For the side nets σ_0 and σ_{n+1}, the MCF is

also 1, since the sidewall wires are shields which are always stable, and power contributed by space between side nets and shields is proportional to α_1 and α_n, respectively.

Signal delays are expressed by an Elmore model using simple approximations for wire capacitances and wire resistance. Using a π—model of the interconnect, the delay of signal σ_i is given by

$$D_i = R_i^{ed}\left(C_i^a + kC_i^{ll} + C_i^{el}\right) + R_i^{wire}\left(\frac{1}{2}\left(C_i^a + kC_i^{ll}\right) + C_i^{el}\right), \tag{2}$$

where R_i^{ed} and C_i^{el} are *effective driver resistance* and *effective load capacitance*. A signal's effective driver resistance represents the actual driving logic gate in series with interconnect resistance leading to the wire's near end. Similarly, the wire's far end is connected to the signal's effective load capacitance, accounting for the actual receiver in parallel with the capacitance of wires connected downstream from the far end. Figure 3 illustrates this model. Although the Elmore model is a first-order approximation and it does not account for input waveform slope [Kahn et al. 1996], it is widely used in the industry for interconnect optimization due to its high-fidelity property [Boese et al. 1993]. The absolute accuracy of the model can also be improved, by using parameter fitting as described in Abou-Seido et al. [2002]. The simple model is used here because of its simplicity in mathematical analysis.

In Eq. (2), k is the Miller Capacitance Factor. For delay calculation, typically MCF = 2 is assumed when neighbor wires switch in the opposite direction causing increased delays, while MCF = 0 is assumed for same-direction switching and reduced delay. We assume that MCF = 1 for all of the signals, yielding nominal delay values. Therefore, in the delay equation (2), cross-capacitance C_{ll} will appear with $k = 1$.

Let's consider an arbitrary layout shown in Figure 5. We say that two wires of the same layer are "visible" to each other if they have some common span. For a given wire, line-to-line capacitances to its visible wires can influence the dynamic power or delay associated with the wire. The progression of VLSI process technology has made the line-to-line term dominant over others [Ho et al. 2001; Sylvester et al. 1998], and its importance is expected to grow in future generations [ITRS 2005]. The line-to-line capacitance between two adjacent wires is proportional to the length of their common span where they are "visible" to each other, and inversely proportional to some positive exponent of their space to each other [Saxena et al. 2000].

Let $I_0, I_1, \ldots, I_n, I_{n+1}$ be $n + 2$ parallel wires, where I_0 and I_{n+1} are leftmost and rightmost shields, $\alpha_0 = 0, \alpha_1, \ldots, \alpha_n, \alpha_{n+1} = 0$ are their corresponding activity factors, $R_0^{ed} = 0, R_1^{ed}, \ldots, R_n^{ed}, R_{n+1}^{ed} = 0$ their corresponding effective driver resistances, and $C_0^{el} = 0, C_1^{el}, \ldots, C_n^{el}, C_{n+1}^{el} = 0$ their corresponding effective load capacitances. A partial order $\prec$ is defined on wires $I_0, \ldots, I_{n+1}$ as follows. We say that $I_i \prec I_j$ if I_i and I_j satisfy the following conditions: (1) the intersection of their vertical span is nonempty, (2) x_i and x_j, the abscissas of I_i and I_j, respectively, satisfy $x_i < x_j$, and I_i and I_j are visible to each other.

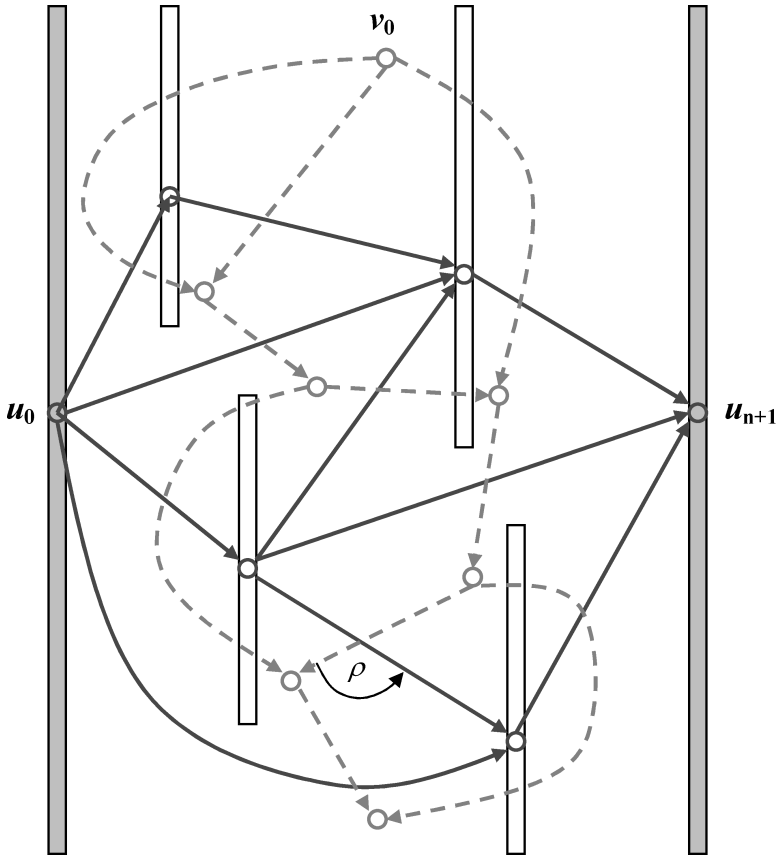

Fig. 5. Spacing visibility graph overlaying its corresponding layout. Solid arcs comprise the primal graph corresponding to wires and their spacing. Dashed arcs comprise its dual graph corresponding to capacitances between visible wires.

This is a left-to-right topological order of the wires, and in the rest of the article we will assume that they are topologically ordered. Wire spacing optimizations preserve the order of the wires.

We assume that the widths $w_0, w_1, \ldots, w_n, w_{n+1}$ of the wires are set by wire sizing optimization performed earlier to satisfy timing requirements [Cheng et al. 1999] and thus are not subject to change in the spacing optimization. This assumption matches VLSI design practice, where wire widths are set very early in the design flow according to signal propagation delay goals. Optimal spacing, however, is more opportunistic and is addressed late in the design. There, all interconnects are already implemented with their specified width, so the unused "white area" can be redistributed in order to allocate more space to highly active or timing-critical wires, hence reducing their capacitance.

Let l_{ij} be the common span of I_i and I_j in which they are visible to each other. If I_i and I_j are not visible to each other l_{ij} is undefined, but for the mathematical discussion we set it to be identically zero. The space $x_j - x_i$ between I_i and I_j is defined if and only if $l_{ij} > 0$. It needs to satisfy the following constraint,

which accounts for the predefined wire widths and the minimum wire spacing dictated by the process technology.

$$x_j - x_i - (w_j + w_i)/2 \geq \quad S_{min}, \quad I_i \prec I_j \tag{3}$$

Inequality (3) means that the order of two visible wires is not allowed to change and they must be apart of each other in at least S_{min}, called the *minimum spacing rule*.

The line-to-line capacitance c_{ij} associated with I_i and I_j is given by

$$c_{ij} = \kappa l_{ij}^{\eta}/[x_j - x_i - (w_j + w_i)/2]^{\gamma}. \tag{4}$$

The factor κ depends only on process technology, whereas $\eta \geq 1$ and $\gamma \geq 1$. Various papers used different values of η and γ. A setting of $\eta = 1$ and $\gamma = 1$ is assumed in Gao et al. [1996], Miyoshi et al. [1995], and Wang et al. [1998]. Other authors use the setting $\eta = 1$ and $\gamma = 1.34$ [Onazawa et. al. 1995; Jhang et al. 1994]. In the following discussion we assume $\eta = 1$, however, the results of this article are applicable for any setting of the previous parameters.
Signal delay can be decomposed into two components.

$$D_i = D^{self}\left(C_i^a, C_i^{el}, R_i^{ed}, R_i^{wire}\right) + D^{cross}\left(C_i^{ll}, R_i^{ed}, R_i^{wire}\right) \tag{5}$$

One is associated with the ground capacitance of the wire and the effective capacitive load. We call it "*self delay*." The second component of the delay is associated with the wire's line-to-line capacitances to other wires residing in the same layer. We call it "*cross delay*." Redistribution of spaces between wires affects only the second component. Using a π—model for individual wire segments, the cross delay of the signal is proportional to

$$D^{cross} \propto \sum_{k=1}^{n_{left}} c_k \left(R^{ed} + r_{1\to k} + \frac{1}{2}r_k\right) \quad + \sum_{k=1}^{n_{right}} c_k \left(R^{ed} + r_{1\to k} + \frac{1}{2}r_k\right), \tag{6}$$

where n_{left} and n_{right} are the numbers of left and right adjacent segments visible by the wire, R^{ed} is the effective driver resistance, c_k and r_k are the capacitance of the kth visible adjacent segment and the resistance of appropriate wire segment and $r_{1\to k}$ is the resistance of the part of the wire from the effective driver near end to the kth visible segment. Figure 6 illustrates calculation of signal cross delay according to (6).

The resistance of a wire segment of length l and width w is given by Abou-Seido et al. [2002]. We have

$$r = \beta \frac{l}{w^{\tau}}, \tag{7}$$

where β is the sheet resistance of the wire and τ is a constant.

Let's denote by $l_{ij,k}$ the length of the kth segment among m_{ij} segments which are visible and shared by I_i and I_j, namely

$$l_{ij} = \sum_{k=1}^{m_{ij}} l_{ij,k}. \tag{8}$$

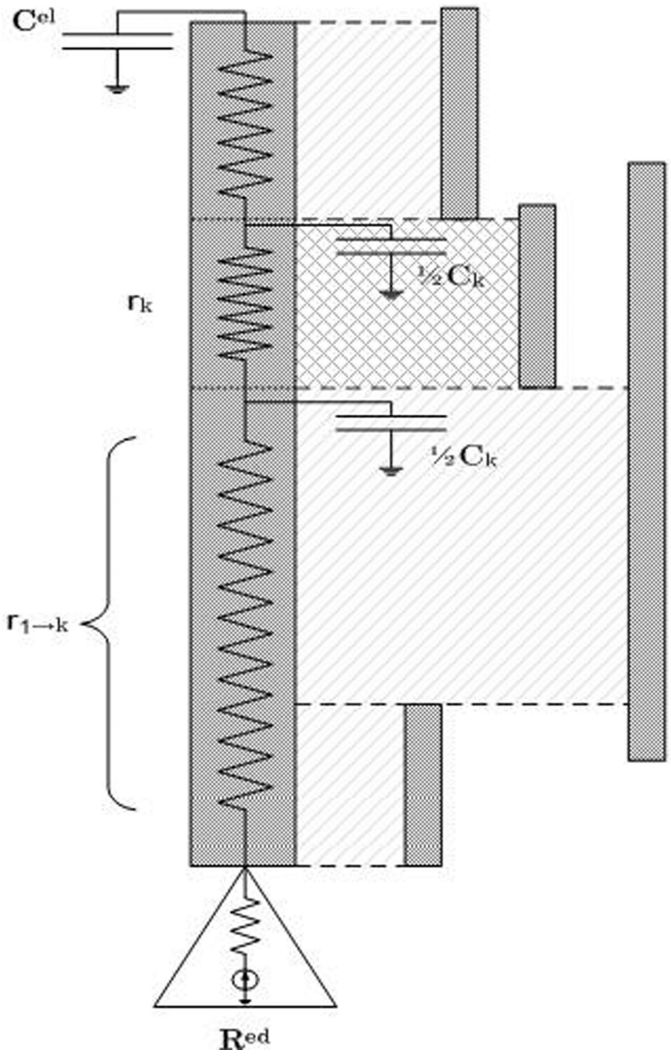

Fig. 6. Calculation of delay using an Elmore approximation. The contribution of the $k-$ spacing segment is equal to $(R^{ed} + r_{1\to k})C_k + \frac{1}{2}r_k C_k$.

In the following discussion we will use a similar notation for resistances and capacitances. Substituting (7) and (4) into (6), the cross delay of the wire I_i is expressed as follows.

$$D_i^{cross} = \sum_{j=0, j\neq i}^{n+1} \sum_{k=1}^{m_{ij}} \frac{l_{ij,k}}{\left[x_j - x_i - (w_j + w_i)/2\right]^{\gamma}} \left(R_i^{ed} + \beta \frac{l_{ij,1\to k}}{(w_i)^{\tau}} + \frac{1}{2}\beta \frac{l_{ij,k}}{(w_i)^{\tau}} \right)$$
$$= \sum_{j=0, j\neq i}^{n+1} \frac{l_{ij}}{\left[x_j - x_i - (w_j + w_i)/2\right]^{\gamma}} \Re_{ij}, \quad (9)$$

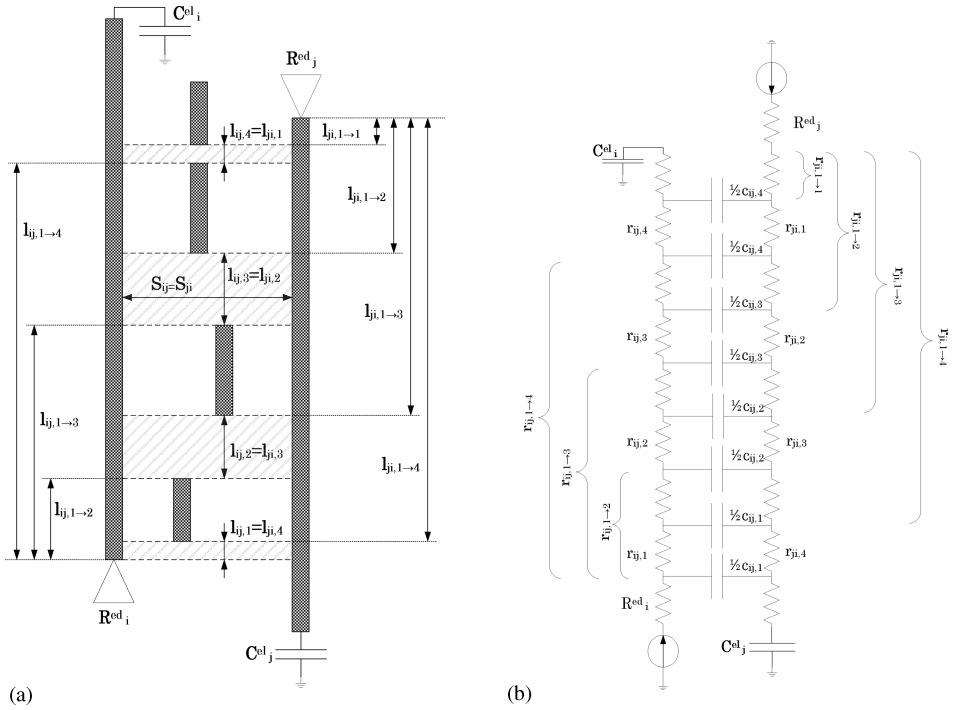

Fig. 7. Example of calculation of cross delay for wires i and j contributed by their shared spaces. (a) layout view (b) corresponding RC model.

For wire i:

$D_{ij}^{cross} = (R_i^{ed} + \frac{1}{2}r_{ij,1})c_{ij,1} + (R_i^{ed} + r_{ij,1\to 2} + \frac{1}{2}r_{ij,2})c_{ij,2} + (R_i^{ed} + r_{ij,1\to 3} + \frac{1}{2}r_{ij,3})c_{ij,3} + (R_i^{ed} + r_{ij,1\to 4} + \frac{1}{2}r_{ij,4})c_{ij,4}$

For wire j:

$D_{ji}^{cross} = (R_j^{ed} + r_{ji,1\to 1} + \frac{1}{2}r_{ji,1})c_{ji,1} + (R_j^{ed} + r_{ji,1\to 2} + \frac{1}{2}r_{ji,2})c_{ji,2} + (R_j^{ed} + r_{ji,1\to 3} + \frac{1}{2}r_{ji,3})c_{ji,3} + (R_j^{ed} + r_{ji,1\to 4} + \frac{1}{2}r_{ji,4})c_{ji,4}$

where we introduced *effective signal resistance* $\Re_{ij}$.

$$\Re_{ij} = \frac{1}{l_{ij}} \sum_{k=1}^{m_{ij}} l_{ij,k} \left(R_i^{ed} + \beta \frac{l_{ij,1\to k} + 0.5 l_{ij,k}}{w_i^{\tau}} \right) \tag{10}$$

$\Re_{ij}$ is a normalized sum of all resistances affecting the delay related to the segments where wires I_i and I_j are visible to each other.

The summation in (9) is done over all wires. Notice that if two wires I_i and I_j are not visible to each other, then $l_{ij} = 0$ and the corresponding sum is zero. An example for calculation of a term in sum (9) is shown in Figures 7(a) and 7(b). 7(a) illustrates the layout of two wires I_i and I_j having 4 distinct visibility segments. Figure 7(b) shows the corresponding RC model.

Similarly to delay, the dynamic power associated with the wire capacitance consists of two terms.

$$P_i = P^{self} + P^{cross} = \alpha_i C_i^a V_{dd}^2 f + \alpha_i C_i^{ll} V_{dd}^2 f \tag{11}$$

In (11) P^{self} denotes wire *"self power,"* contributed by wire area and fringe capacitance, and P^{cross} denotes wire "*cross power*," contributed by line-to-line capacitances of the wire to other wires in the same routing layer. Using the notation of (8) and substituting (4) into (11), the cross power of the wire I_i is expressed as

$$\begin{aligned} P_i^{cross} &= \alpha_i k \sum_{j=0, j\neq i}^{n+1} \sum_{k=1}^{m_{ij}} \frac{l_{ij,k}}{\left[x_j - x_i - (w_j + w_i)/2\right]^{\gamma}} \\ &= \alpha_i k \sum_{j=0, j\neq i}^{n+1} \frac{l_{ij}}{\left[x_j - x_i - (w_j + w_i)/2\right]^{\gamma}}, \end{aligned} \tag{12}$$

where the coefficient k incorporates supply voltage, clock frequency, and technology-dependent constants.

Our goal is optimization of power with consideration of timing. The commonly used objective functions incorporating both power and delay are the power-delay product or similar multiplicative metrics. However, these functions are not handy for mathematical analysis. Instead, an objective function based on a weighted sum rather than a product of delay and power is used.

Consider the problem of minimizing a weighted sum of cross power and cross delay (Weighted Power-Delay Sum—WPDS).

$$E^{cross}(\mathbf{x}) = \lambda \mathbf{1} \cdot \mathbf{P^{cross}}(\mathbf{x}) + \mu \cdot \mathbf{D^{cross}}(\mathbf{x}) \tag{13}$$

Here $\lambda \in \mathbb{R}$ is scalar and $\boldsymbol{\mu}, \mathbf{P^{cross}}, \mathbf{D^{cross}}, \mathbf{x}$ are vectors of real numbers. $\mathbf{1}$ represents the unit vector. λ and μ are coefficients which set the relative importance of the power and delay terms for each signal. Note that while power is equally additive from all nets, delays of different nets may have different criticality and hence we use a vector of weights for the delays. The goal is to find a vector of wire locations $\mathbf{x}$ that minimizes (13). Notice that vectors $\mathbf{P^{cross}}$ and $\mathbf{D^{cross}}$ have only n elements, since I_0 and I_{n+1} are tied to constant voltages. Since the objective function (13) is defined as a weighted sum of power and delay characteristics, the power and delay should be normalized to make them comparable. Normalization factors $P_{tot} = \sum_i P_i^{cross}(\mathbf{x})$ and $D_{tot} = \sum_i D_i^{cross}(\mathbf{x})$ calculated at the preoptimization design state can be used. It is convenient to use $\lambda = 1$ and to set elements of the coefficient vector $\boldsymbol{\mu}$ according to timing criticality of individual signals.

3. NECESSARY AND SUFFICIENT CONDITION FOR MINIMAL WPDS

LEMMA 1. *The minimum of (13) subject to (3) is global.*

PROOF. Let us define $s_{ij} = x_i - x_j - (w_i + w_j)/2$ to be the spacing between two visible wires. Substitution of s_{ij} into (13) yields the following minimization problem.

$$\text{minimize } \lambda \mathbf{1} \cdot \mathbf{P^{cross}}(\mathbf{s}) + \boldsymbol{\mu} \cdot \mathbf{D^{cross}}(\mathbf{s}), \tag{14}$$

$$\text{subject to } s_{ij} \geq S_{min}, \quad I_i \prec I_j, \tag{15a}$$

$$s_{ij} - x_j + x_i + \left(w_j + w_i\right)/2 = 0, \quad I_i \prec I_j, \tag{15b}$$

$$0 < x_i < A, \quad 1 \le i \le n \tag{15c}$$

The objective function (14) is convex (see the appendix of Saxena et al. [2000]) and so are the constraints (15a) through (15c). Consequently, there is a single minimum which is global [Luenberger 1984]. □

Let us ignore for the moment the requirement (15a) of minimum spacing, and replace it by $s_{ij} > 0$. Although it is not feasible for a VLSI layout, it simplifies the characterization of the optimal spacing yielding minimum WPDS. We'll return to (15a) and take it into account in the real implementation of wire spacing. Formally, (15a) is replaced by

$$s_{ij} > 0, \quad I_i \prec I_j. \tag{15d}$$

Consider now the abscissa x_i of a wire I_i whose width is w_i, $1 \le i \le n$. Denote all of its left and right visible wires by I_{ij}^{left} and I_{ij}^{right}, respectively, where the superscript designates left and right sides of I_i and in the subscript j is varying. Let's denote by s_{ij}^{left} and s_{ij}^{right} spaces between wires I_i and I_j on the left and right sides of I_i, respectively. In addition, we denote by $l_{ij,k}^{left}$ and $l_{ij,k}^{right}$ the length of the kth spacing interval between wires I_i and I_j on the left and right sides of I_i. We use similar indexing notation for the corresponding abscissas, widths, lengths of wires, visibility span, activity factors, and signal driver resistances.

THEOREM 1 (NECESSARY AND SUFFICIENT CONDITION FOR MINIMAL WPDS). *A necessary and sufficient condition so that the WPDS expression in (14) is minimized subject to the constraints (15b), (15c), (15d) is that every wire* I_i, $1 \le i \le n$ *satisfies*

$$\sum_j \frac{l_{ij}^{left}\left(\mu_i \Re_{ij}^{left} + \mu_j \Re_{ji}^{left} + \lambda\left(\alpha_i + \alpha_j^{left}\right)\right)}{\left[x_i - x_j^{left} - \left(w_i + w_j^{left}\right)\right]^{\gamma+1}} = \sum_j \frac{l_{ij}^{right}\left(\mu_i \Re_{ij}^{right} + \mu_j \Re_{ji}^{right} + \lambda\left(\alpha_i + \alpha_j^{right}\right)\right)}{\left[x_j^{right} - x_i - \left(w_i + w_j^{right}\right)\right]^{\gamma+1}}. \tag{16}$$

Summation on the left- and righthand sides of (16) is performed over all left and right visible wires.

PROOF. By substitution of (15b) into (14), it follows that the WPDS for wire I_i is proportional to

$$\sum_j \frac{l_{ij}^{left}\left(\mu_i \Re_{ij}^{left} + \mu_j \Re_{ji}^{left} + \lambda(\alpha_i + \alpha_j)\right)}{\left[x_i - x_j^{left} - \left(w_i + w_j^{left}\right)\right]^{\gamma}} + \sum_j \frac{l_{ij}^{right}\left(\mu_i \Re_{ij}^{right} + \mu_j \Re_{ji}^{right} + \lambda(\alpha_i + \alpha_j)\right)}{\left[x_j^{right} - x_i - \left(w_i + w_j^{right}\right)\right]^{\gamma}}. \tag{17}$$

The minimum of (14) is obtained at an internal point of the region defined by (15d). Otherwise, there would be some $s_{ij} = 0$. This, however, will result in (14) going to infinity, hence not a minimum.

Since the minimum is obtained at an internal point, and by Lemma 1 the minimum is global, a necessary and sufficient condition to minimize (14) is that its gradient with respect to $\mathbf{x}$ is zero. The only term of the sum (14) which contains x_i is expressed by (17). Therefore, differentiation of (14) by $\mathbf{x}$ is equivalent to differentiation of (17) by x_i. Such differentiation yields (16). □

The physical interpretation of Theorem 1 is that it is necessary and sufficient for minimum WPDS that every wire is in "equilibrium," where the sum of its left side weighted capacitance derivatives is equal to that of its right side.

Solving (14) for all wires together with the constraints (15b), (15c), (15d) involves a large number of nonlinear equations and linear inequalities. Its solution for a typical VLSI layout can be very tedious. The next section describes a representation which addresses all nets simultaneously, yielding the optimal solution.

4. GRAPH REPRESENTATION OF POWER MINIMIZATION

This section presents a planar graph model of the problem, which projects the "local equilibrium" necessary and sufficient condition of Theorem 1 into a global consequence related to the entire layout. This representation leads to an algebraic formulation of the solution, and provides an interesting insight about the nature of the optimum, given as a corollary at the end of the section, that the capacitance density in an optimally spaced metal layer is uniform throughout the layout.

Let us build a wire visibility graph and show how minimal WPDS can be captured by satisfaction of some properties of that graph. The *Spacing visibility graph* $G(U, E, \boldsymbol{\xi})$ is a directed graph whose vertices U correspond to wires and arcs E correspond to spaces between wires visible to each other. An arc $e_{ij} \in E$ connecting $u_i \in U$ with $u_j \in U$ exists if $I_i \prec I_j$ (I_i is residing left to I_j and they are visible to each other, namely $l_{ij} > 0$ and $s_{ij} > 0$). In this definition G is a planar directed acyclic graph having one source u_0 and one sink u_{n+1}, corresponding to I_0 and I_{n+1}, respectively. The solid arcs in Figure 5 illustrate the graph overlaying the original layout.

An arc e_{ij} is assigned with the real positive number $\xi_{ij} = s_{ij} + (w_i + w_j)/2$ which is the distance between the centerlines of I_i and I_j. In this setting, the length of all paths from source to sink is equal to the distance from the leftmost to the rightmost wire, which is the block width A. Let $\Gamma = \{\gamma_k\}$ be the set of all source-to-sink paths of $G(U, E, \boldsymbol{\xi})$, then

$$\sum_{e_{ij} \in \gamma_k} \xi_{ij} = \sum_{e_{ij} \in \gamma_k} s_{ij} + \big(w_i + w_j\big)/2 = A, \quad \forall \gamma_k \in \Gamma. \tag{18}$$

It follows from planarity of G that there exists a dual graph $H(V, F, \boldsymbol{\eta})$, illustrated in Figure 5 by dashed arcs. We call it the *weighted capacitance derivative graph*. It is defined as follows. Define a source vertex $\boldsymbol{v}_0$ and sink vertex v_{n+1} of

H, located in the infinite faces of G. The vertices of H are assigned each inside a distinct face of G. Let F be the arcs of H. Such a graph representation occurs in floor planning. A study of their algebraic properties can be found in Wimer et al. [1988].

To every dual arc $f_{ij} \in F$ crossing the primal arc $e_{ij} \in E$ we assign the following weight.

$$\eta_{ij} = \frac{1}{s_{ij}^{\gamma+1}} l_{ij}[\mu_i \Re_{ij} + \mu_j \Re_{ji} + \lambda(\alpha_i + \alpha_j)] \tag{19}$$

The expression in (19) is the absolute value of the derivative of c_{ij} by any of the abscissas x_i or x_j, weighted by the effective activity factors and the effective signal resistances of the wires forming the space s_{ij}. We refer to η_{ij} as *weighted cross-capacitance density*, since it represents the value of weighted cross-capacitance divided by the spacing between two wires. The direction of an arc $f_{ij} \in F$ is set such that a counterclockwise rotation of f_{ij} towards e_{ij} by the angle $\rho < \pi$ leads to overlap of arc heads, as shown in Figure 5. The graph $H(V, F, \eta)$ thus defined is also directed and acyclic, having one source and one sink. Figure 5 illustrates the overlay of the dual graphs.

In the aforesaid representation the topology of G is invariant of the abscissas of the wires, as long as the left-to-right relations between visible wires are maintained. The interpretation of paths in H is of vertically stacked capacitors, and the path length is the sum of weighted capacitance derivatives.

It follows from the invariance of the topology of G under repositioning wires and from duality that the topology of H is also invariant. This implies that any vertical stack of capacitors, corresponding to a source-to-sink path in H is preserved in the layout, regardless of the abscissas of $I_0, \ldots, I_{n+1}$. This is shown in Figure 8, where the gray areas represent the line-to-line capacitances. Note that a face of H always encloses a vertex in G corresponding to a vertical wire. The left (right)-side path corresponds to the vertical stack of capacitors on its left (right) side, as illustrated in Figure 8.

All source-to-sink paths of H can be ordered "left to right" by applying a depth-first traversal which expands all the paths from v_0 to v_m [Cormen et al. 2001]. Paths are exhausted such that any two successively issued paths δ' and δ'' are constructed as follows. Both paths emanate from v_0 and share the same arcs up to v_r, where they split into two subpaths $\rho' \subset \delta'$ and $\rho'' \subset \delta''$ extending between v_r and v_s. At $v_s\delta'$ and δ'' merge again up to v_m, as illustrated in Figure 9. The physical interpretation of ρ' and ρ'' is of the left- and right-side stacked capacitors shown in Figure 8.

LEMMA 2. *All source-to-sink paths in H are critical (having same length) if and only if for every internal face the left and right subpaths have the same length.*

PROOF. Figure 9 illustrates the proof. Let all source-to-sink paths in H be critical. Assume on the contrary that there exists an internal face of H which left and right subpaths have different lengths. Then, two successive source-to-sink paths must exist in the previously defined order; one is longer than the

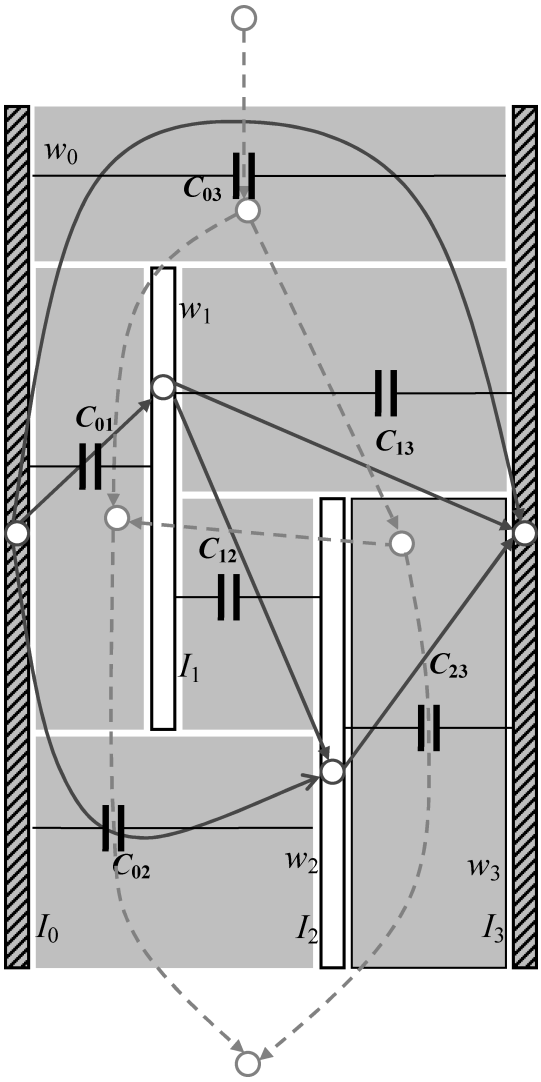

Fig. 8. Cross-capacitance layout model with the corresponding spacing visibility graph, and its weighted capacitance derivative dual. Gray areas correspond to line-to-line capacitors. Faces of the dual graph correspond to capacitors residing on the two sides of a signal wire.

other, since except the two distinct subpaths they share common arcs, hence a contradiction.

Conversely, let left and right subpaths of any face of H have the same length. Assume on the contrary that not all source-to-sink paths in H are critical. There exist then two successive source-to-sink paths δ' and δ'' whose lengths are different. Paths δ' and δ'' coincide in all their arcs, except in those arcs forming $\rho' \subset \delta'$ and $\rho'' \subset \delta''$, which are the left and right sides of an internal face in H. But then these must have different lengths, a contradiction. □

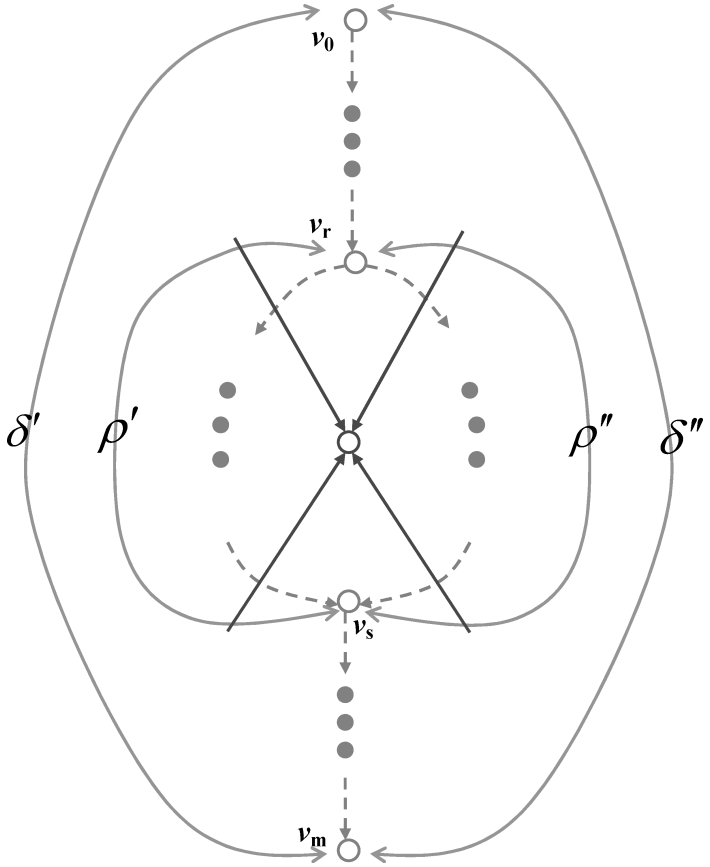

Fig. 9. Proof of Lemma 2. Two "left to right" ordered paths from v_0 to v_m in H consist of two common parts and a face enclosing a vertex of G.

THEOREM 2 (PATH CRITICALITY CONDITION FOR MINIMUM WPDS). *The WPDS of all signals in a layout is minimized if and only if all paths in the weighted capacitance derivative graph are critical.*

PROOF. According to Lemma 2 all paths in H are critical if and only if the left and right paths of any internal face have same length. The weights of H's arcs are the derivatives of line-to-line capacitances. Consequently, the sums of derivatives of line-to-line capacitances stacked on the two opposite sides of every wire are equal to each other. By Theorem 1 this equality is a necessary and sufficient condition for minimal interconnect WPDS. □

Let $\Delta = \{\delta_k\}$ be the set of all source-to-sink paths of $H(V, F, \eta)$, then according to Theorem 2 there exists at minimum a positive real number B satisfying

$$\sum_{f_{ij}\in\delta_k} \eta_{ij} = \sum_{f_{ij}\in\delta_k} l_{ij}\big(\mu_i \Re_{ij} + \mu_j \Re_{ji} + \lambda(\alpha_i + \alpha_j)\big) \Big/ s_{ij}^{\gamma+1} = B, \quad \forall \delta_k \in \Delta. \tag{20}$$

The graph representation of the WPDS problem can provide an algebraic solution method as follows. Let **K** and **L** be the coefficient matrices of (18) and (20), respectively. Then, combining the two in one matrix representation, they can be rewritten as

$$\begin{pmatrix} \mathbf{K} & \mathbf{0} \\ \mathbf{0} & \mathbf{L} \end{pmatrix} \begin{pmatrix} \xi \\ \eta \end{pmatrix} = \begin{pmatrix} \mathbf{A} \\ \mathbf{B} \end{pmatrix}, \tag{21}$$

where $\mathbf{A} = A\mathbf{1}$ and $\mathbf{B} = B\mathbf{1}$ are corresponding vectors of the righthand side constants A and B in (18) and (20).

According to Wimer et al. [1988] and Seshu et al. [1961], it can be shown that that the rank of the combined matrix in (21) is $N_{space} + 1$. Hence the number of independent equations is linear in the size of the layout.

There is still the question of how to effectively derive the $N_{space}+1$ equations. To this end we'll interpret (18) and (20) as network cuts and flows [Hu 1969]. It follows from the duality that there is a one-to-one correspondence between paths in G and cuts in H and vice versa. Let us exchange the weights of dual arcs in $G(U, E, \xi)$ and $H(V, F, \eta)$, thus creating new graphs $G'(U, E, \eta)$ and $H'(V, F, \xi)$. Then, the lengths equality of all paths in G translates to equality of all cut flows in H' and similarly for H and G'.

The equality of all cut flows in a graph implies that the total length of incoming arcs of a vertex is equal to the total length of its out-going arcs. This holds for both H' and G', thus yielding $|U|+|V|$ vertex equations. Substituting $|U|$ and $|V|$ which have been used in finding the rank of (21) yields a total of $N_{space} + 3$ equations, which can replace (18) and (20).

An interesting consequence of Theorem 2 is that at the optimum, weighted line-to-line capacitance density is uniformly distributed across the whole layout. Consider an imaginary vertical line scanning the layout from left to right. Define $C(x)$ to be the cumulative line-to-line capacitance from the left side of the block, and $c(x) = dC(x)/dx$ is its derivative (or *density*), namely $C(x) = \int_{\xi=0}^{\xi=x} c(\xi)d\xi$. Using this terminology, with the interpretation of a vertical scan-line as a source-to-sink path in H, the follows from Theorem 2.

COROLLARY 1 (UNIFORMITY OF CAPACITANCE DENSITY). *The total interconnect WPDS in a layout is minimized if and only if its underlying line-to-line weighted capacitance density is constant.*

5. ITERATIVE ALGORITHMS FOR MINIMIZATION OF WPDS

While the graph representation gives an effective algebraic solution method and a useful insight about the uniformity of capacitance density in an optimal layout, we can use a simple yet robust iterative algorithm which relies on the convexity of the problem. It has been implemented and successfully used in the design of a commercial 65 nanometer high-end microprocessor, and the results are shown in Section 7. The iterative algorithm is based on the equilibrium condition for minimum WPDS stated in Theorem 1. The algorithm uses a modified form of the balancing technique described in Cederbaum et al. [1992], which had been applied for wire spacing in a commercial tool for manufacturing yield enhancement [XTREME].

The algorithm works on a single wire at a time while maintaining a global view of the other wires. It repositions a wire between its left and right visible wires, such that the equilibrium in (16) is achieved. According to Theorem 1, at a nonminimum point there exists at least one wire which is not in equilibrium. We then shift it to the abscissa x which satisfies (16). It has been proven in Cederbaum et al. [1992] that such iterations converge to a configuration where all wires are in equilibrium, such that (16) is satisfied for all wires.

The path lengths expressed in the constraints (18) are invariant under repositioning of a single wire. Since initially the layout is legal, thus satisfying (18), it is automatically satisfied through all the iterations.

It has yet to be shown that the repositioning of a single wire indeed reduces the total WPDS. Considering (9) and (12), the only affected terms are those which involve the shifted wire and its left and right visible neighbors. These terms are expressed in (17). The amount of delay and power appears only once in (9) and (12) and their weighted sum after repositioning has been lowered, hence the net WPDS change is negative. We can conclude in the following theorem.

THEOREM 3. *The iterative algorithm which equilibrates wires one at a time converges to the global minimum of WPDS.*

PROOF. The infinite sequence of WPDS values obtained by the iterative algorithm is positive and monotonically decreasing, hence converging to a limit where all wires are in equilibrium. Theorem 1 ensures that this limit is indeed the global minimum. □

Following is a pseudocode of the algorithm.

In order to ensure fast convergence of the iterative algorithm, wires are put into a heap [Cormen et al. 2001] in decreasing order of their distance from equilibrium. This is implemented in lines 1 and 2 of the pseudocode. Assuming that the number of visible wires of any wire is bounded, which is the practical situation in VLSI layouts, equilibration calculations consume $O(1)$ time per wire. Building the heap consumes $O(n \log n)$ time.

Algorithm.

```
1.  initialization: for every wire calculate "distance" from equilibrium by equation (16)
2.  put all wires into a heap
3.  while top of heap is greater than ε > 0 (measure of accuracy) do {
4.     solve (16) for the wire at the top of the heap
5.     locate the wire at abscissa found in line 4
6.     re-enter top wire to heap
7.     for every visible wire do {
8.        update "distance" from equilibrium by (16)
9.        re-enter the wire into heap
10.    }
11. }
12. retain connectivity by stretching all orthogonal wires according to the shift made
    to the vertical wire they are connected to.
```

The equilibration of the top wire modifies the equilibrium of other wires visible to it. In the outer loop at line 3 wires are popped from the top of the heap one at a time, repositioned at their equilibrium abscissa in line 5, and then re-entered to the heap in line 6 (they are located at the bottom by definition since their distance from equilibrium is zero). This takes $O(\log n)$ time.

The inner loop in lines 7 through 10 handles all the wires visible to the previous top wire that just has been re-entered into the heap. Their distance from equilibrium is recalculated and their location in the heap is updated accordingly by re-entering. Assuming that the number of visible wires of any wire is bounded, this operation also consumes $O(\log n)$ time. Finally, line 12 retains layout connectivity by shortening or extending orthogonal wires in adjacent layers that are connected to the ends of shifted wires in the layer being spaced. This operation is $O(1)$ per wire, hence $O(n)$ altogether.

Once the convergence criterion in line 3 is met, it follows from the very definition of a heap that all wires are at distance ε or less from equilibrium, where ε is some predefined accuracy. The dependence of runtime on ε has been analyzed in Cederbaum et al. [1992]. According to Cederbaum et al. [1992], the runtime complexity of the algorithm is $O(n \log n \log \frac{1}{\varepsilon})$. Although the number of vertices in $G(U, E, \boldsymbol{\xi})$ is large, Figure 2 shows that practical VLSI layouts are sliced by a fixed mesh of VCC/VSS power grid, hence $G(U, E, \boldsymbol{\xi})$ at the top metal layers is separable into many independent small graphs. The number of movable wires between two fixed power rails doesn't typically exceed 12. Assuming a manufacturing grid $\varepsilon = 0.001$ micron for a 65 nanometer process technology, the worst-case number of iterations is a few hundreds per VCC/VCC trunk. Considering the entire chip area which incorporates a few hundreds or thousands of such trunks, the iterative balancing algorithm converges within several minutes of computation time per layer.

So far, the S_{min} constraint in (3) has been ignored. Practical layout must account for it, of course. The iterative algorithm supports it as follows. Once the equilibrium position of the wire is found by solving (16), it is checked whether S_{min} is violated. If this is the case then the wire stops at S_{min}. The iterative algorithm still yields the minimum, although it is achieved at the boundary of the feasible region rather than at an internal point as assumed in the proof of Theorem 1. The optimality can be verified from Lemma 1.

6. PRACTICAL CONSIDERATIONS IN POWER-DELAY OPTIMIZATIONS

The objective function (13) can be refined to suit a specific practical application. Such refinement can be done using coefficients μ_i and λ. Typical applications are shown in Table I, describing various design stages during process migration. Assume that a circuit implemented in a previous generation of technology is being redesigned for a new process technology. In the early stages of design migration there are no firm timing specifications for individual internal nodes of the circuit. Therefore, the initial goal is to reduce all signal delays. Thus, both power and delay are given the same weights $\boldsymbol{\mu} = \mathbf{1},\ \lambda = 1$.

At a later stage, realistic time budgeting is calculated such that each signal is assigned a required arrival time. The slack of each signal is defined as the

Table I. Possible Settings of Weighting Coefficients for Different Optimization Objectives

Application	Setting of Parameters
Reduction of power and average signal delay	$\boldsymbol{\mu} = \mathbf{1},\ \lambda = 1$
Reduction of power and total sum of slacks	$\boldsymbol{\mu} = \mathbf{1},\ \lambda = 1$
Power reduction of signals with positive slack and delay reduction of signals with negative slack	$\lambda = 1, \mu_i = 0$ for signals with positive slack and $\lambda = 0, \mu_i = 1$ for signals with negative slack
Power reduction with consideration of signal criticality	$\lambda = 1$ and μ, according to signal criticality
Measuring maximum power improvement possible by spacing	$\boldsymbol{\mu} = 0, \boldsymbol{\lambda} = 1$
Measuring maximum delay improvement possible by spacing	$\boldsymbol{\mu} = 1, \boldsymbol{\lambda} = 0$

difference between the required arrival time and its actual delay. Negative slack indicates a violation of specifications. The total sum of slacks indicates the potential for increasing the operating frequency of a chip (by reducing the required times). Notice that, mathematically, optimizing the total sum of slacks is equivalent to optimizing the total sum of delays. Therefore, $\boldsymbol{\mu} = \mathbf{1}$ and $\lambda = 1$ are used as before.

Separation to sum of negative slacks and sum of positive slacks is very useful in design migration. While the total negative slack reflects the amount of expected circuit design effort for timing closure, the sum of positive slacks indicates opportunities for power saving. Thus, in this optimization scenario, different weights are assigned to nets with negative and positive slacks. We set $\lambda = 1$, $\{\mu_i = 0\}_{i \in I_p}$ (I_p is the index set of positive slack nets) to focus optimization on power saving, and $\lambda = 0$, $\{\mu_j = 1\}_{j \in I_n}$ (I_n is the index set of negative slack nets) to focus optimization on delay reduction. Further refinement is also possible: If nets with a small positive slack need to be protected from turning into negative slack, three types of settings are defined. For nets with negative slack, $\lambda = 0$, $\{\mu_j = 1\}_{j \in I_n}$, for nets with large positive slack, $\lambda = 1$, $\{\mu_i = 0\}_{i \in I_l}$, and finally, for nets with small positive slack, $\lambda = 1$, $\{\mu_k = 1\}_{k \in I_s}$, where I_s is the corresponding index set.

At the final stage of timing closure, critical paths are treated for eliminating negative slacks and delay reduction. In that case the objective is delay minimization of the signal with the worst slack. To this end, corresponding coefficients μ_i are set according to the criticality of the signal: The most critical signals will be assigned the largest values μ_i.

7. EXPERIMENTAL RESULTS

A pictorial example of spacing optimization is shown in Figure 10, where the activity factor is written next to each wire. As shown in Figure 10(b), the optimization algorithm has distributed the spaces according to the relative weight of wire activity.

The iterative algorithm presented in Section 6 has been applied to the entire global routing layers in a 65 nanometer high-end microprocessor. Due to the large size of the data, the layout of the processor was divided into five portions. The number of wires varied from ∼44000 to ∼118000 wires in the

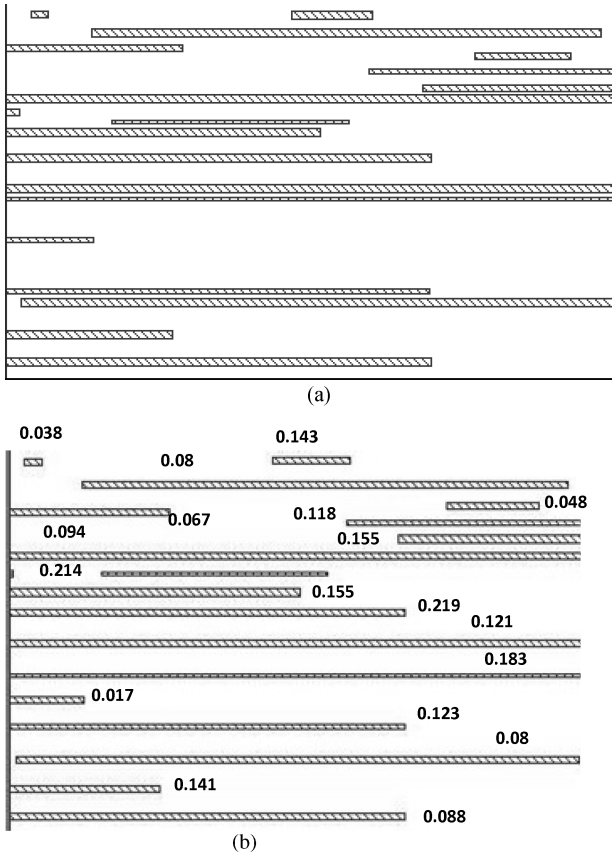

Fig. 10. An example of spacing optimization for reduced interconnect power, implied by activity factors: (a) before optimization (b) after optimization. Activity factors are shown next to the wires.

portion. Optimization was performed on each portion separately while maintaining boundary conditions to obtain proper interface and connectivity. All top-level metal layers from 5^{th} to 8^{th} were optimized, whereas all connectivity and design rules were perfectly maintained. The algorithm ran on a 2.4 GHz Pentium IV machine with 8GB of memory, consuming about thirty minutes per layer.

In each layout portion the optimization was performed simultaneously for all wires, where power grid wires were not allowed to move. This is equivalent to performing the optimization in each power grid slice separately. All the results reported in the following were obtained using a fixed power grid. We also ran experiments without this limitation on the power grid, thus allowing more freedom in spacing optimization, and got power savings higher by about 40% in comparison with the fixed power grid. However, shifting the power grid can be too disruptive for a conservative design methodology. Hence we discuss the fixed power grid results only.

Optimization was performed in several modes. First, power minimization by wire spacing (ignoring delays) was performed by setting $\lambda = 1$, $\boldsymbol{\mu} = \mathbf{0}$. Then

Table II. Power and Delay Reduction Obtained for Entire Global Routing by Optimizing Power Only: $\lambda = 1$, $\boldsymbol{\mu} = \mathbf{0}$

Portion No.	Power Improvement,%	Average Delay Improvement,%
1	18.73	−8.77
2	19.28	0.88
3	16.24	0.43
4	16.03	−2.86
5	20.10	3.99
Entire Chip	**18.55**	**−2.75**

Table III. Power and Delay Reduction Obtained for Entire Global Routing by Optimizing Delay Only: $\lambda = 0$, $\boldsymbol{\mu} = \mathbf{1}$

Portion No.	Power Improvement,%	Average Delay Improvement,%
1	4.52	10.30
2	3.26	13.18
3	1.73	10.50
4	3.96	9.71
5	6.22	12.70
Entire Chip	**4.22**	**11.30**

average delay minimization (ignoring power) was performed by setting $\lambda = 0$, $\boldsymbol{\mu} = \mathbf{1}$. The results are shown in Tables II and III, respectively. As can be seen in Tables II, ignoring timing in the optimization yielded 18.55% reduction of the interconnect power, while average delay has increased (degraded) by 2.75%. On the other hand in Table III, focusing on delay yielded when only 4.22% power saving, while the average delay was reduced by 11.30%. These extreme cases define the "power-delay optimization envelope," and any other setting of λ and $\boldsymbol{\mu}$ will result in power and delay improvements within these ranges.

In the following series of experiments $\boldsymbol{\mu}$ was set uniformly for all wires, ranging in $\mu = 0, 1, 2, 5, 10, 100$, while maintaining $\lambda = 1$. The resulting power and delay reductions are plotted relative to the power-delay optimization envelope in Figure 11. As seen from the chart, delay improvement increases rapidly with increasing μ, while power improvement decreases slowly. Table IV shows detailed results for $\lambda = 1$, $\boldsymbol{\mu} = \mathbf{1}$. The total global interconnect dynamic power was reduced by 16.85%, and the average delay was reduced by 9.62%. According to Figure 11, further increase of μ hardly improves delay, but yields some power improvement. Therefore, setting $\lambda = 1$, $\boldsymbol{\mu} = \mathbf{1}$ is reasonable for power-delay optimization. Comparison of improvements obtained by the settings $\{\lambda = 1, \boldsymbol{\mu} = \mathbf{0}\}$, $\{\lambda = 0, \boldsymbol{\mu} = \mathbf{1}\}$, and $\{\lambda = 1, \boldsymbol{\mu} = \mathbf{1}\}$ is shown in Figure 12. The power-delay tradeoffs demonstrated in Figures 11 and 12 and in Tables II, III, and IV are common in VLSI design optimization problems, but reallocation of spaces can often improve both power and delay. According to Magen et al. [2004] and Figure 1, the total interconnect power reduction of 16.85% translates to 1.7% reduction in the total dynamic power of the processor we have worked on. This is considered a significant improvement in industrial terms, obtained by a simple layout post-processing step, and its results have been used in silicon.

In the last experiment we optimized power while considering individual signal criticality. This was done by setting individual delay weights μ_i's to critical

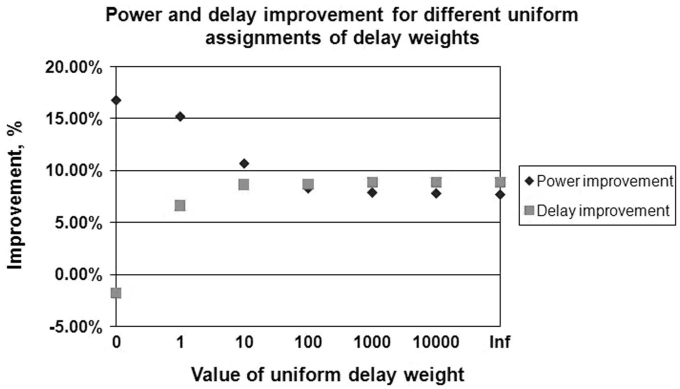

Fig. 11. Power and delay improvement for different uniform assignments of delay weights: $\mu_i = \mu$ for all $1 \leq i \leq n$, while $\lambda = 1$. The rightmost points correspond to optimization of delay only ($\lambda = 0; \mu_i = 1$).

Table IV. Power and Delay Reduction Obtained for Entire Global Routing by Optimizing Equaly Weighted Power and Delay: $\lambda = 1$, $\boldsymbol{\mu} = \mathbf{1}$

Portion No.	Power Improvement,%	Average Delay Improvement,%
1	16.01	8.99
2	17.99	11.27
3	15.13	8.88
4	14.57	7.34
5	18.79	10.75
Entire Chip	**16.85**	**9.62**

nodes and checking how it affects optimization results. For power we set $\lambda = 1$. The results are shown in Table V. The first row shows the initial timing state where Worst Negative Slack (WNS), and hence the Total Negative Slack (TNS) as well, are zero. Since power optimization by wire spacing is performed after timing closure, TNS at this stage is usually zero. The second row shows the results when power minimization was done (ignoring timing). Though power was reduced by 17%, average delay had increased by 1.82% and TNS jumped to -257.5. The third row shows the results obtained when power and delay were equally treated at optimization. Though power gain is slightly worsened, delay got improved by 6.63%. TNS is still -22.3 units due to a few critical paths whose delay was harmed by the new spacing. To repair timing, the nets which turned critical by the former setting were assigned higher delay weight, based on the amount of their negative slack. Delay weights were taken in the range of 1 to 100 according to the amount of negative slack. Signals with positive slack were still assigned a weight of 1. As shown in the fourth row of the table, TNS was reduced by half using this setting, while power and delay improvements were almost not affected. The distributions of negative slacks created in these optimizations are plotted in Figure 13.

In a final experiment, those nets that turned critical by the former experiment were "frozen" and their wires together with their neighbors were not allowed to move, in order to eliminate any effects on their delays. This indeed

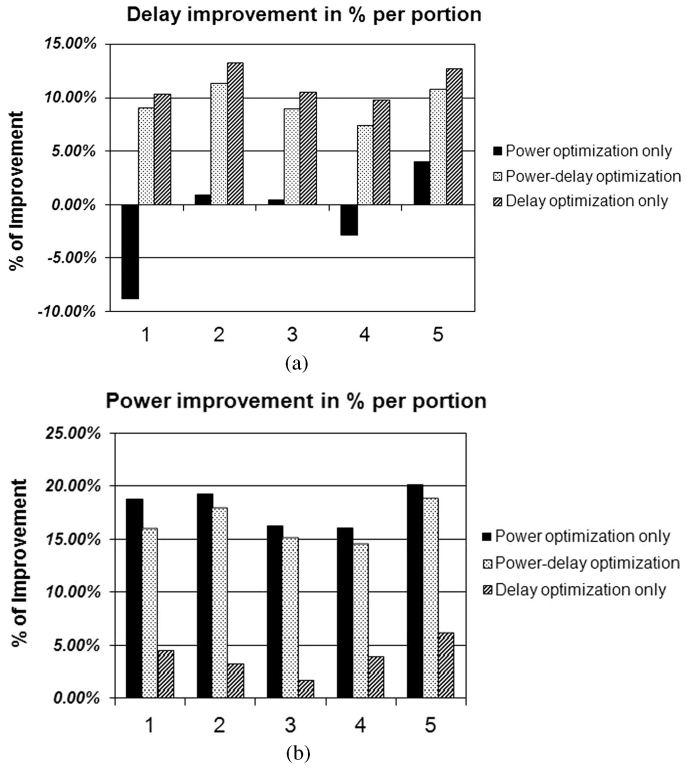

Fig. 12. Power and delay improvement in different kinds of optimization for different portions of the processor layout.

Table V. Optimization Results for Single Routing Portion with Different Weight Assignment

Optimization	Power Improvement, %	Delay Improvement,%	Total Negative Slack (TNS)	Worst Negative Slack (WNS)
Initial state	—	—	0	0
Power optimization only, $\lambda = 1$, $\boldsymbol{\mu} = \mathbf{0}$	16.71	−1.82	−257.5	−6.09
Power−delay optimization, $\lambda = 1$, $\boldsymbol{\mu} = \mathbf{1}$	15.20	6.63	−22.3	−2.21
Criticality−driven optimization	15.01	6.70	−11.3	−1.14
Signal freezing (final)	12.82	6.16	0	0

resulted in zero TNS (i.e., no critical paths were affected), while 12.8% power and 6.16% delay reduction were still achieved.

8. CONCLUSIONS

The problem of optimizing wire spacing in order to reduce the interconnect switching power and wire delays is addressed for the global routing metal layers of VLSI systems. A mathematically proven algorithm based on necessary and sufficient conditions and capacitance density interpretation has been proposed.

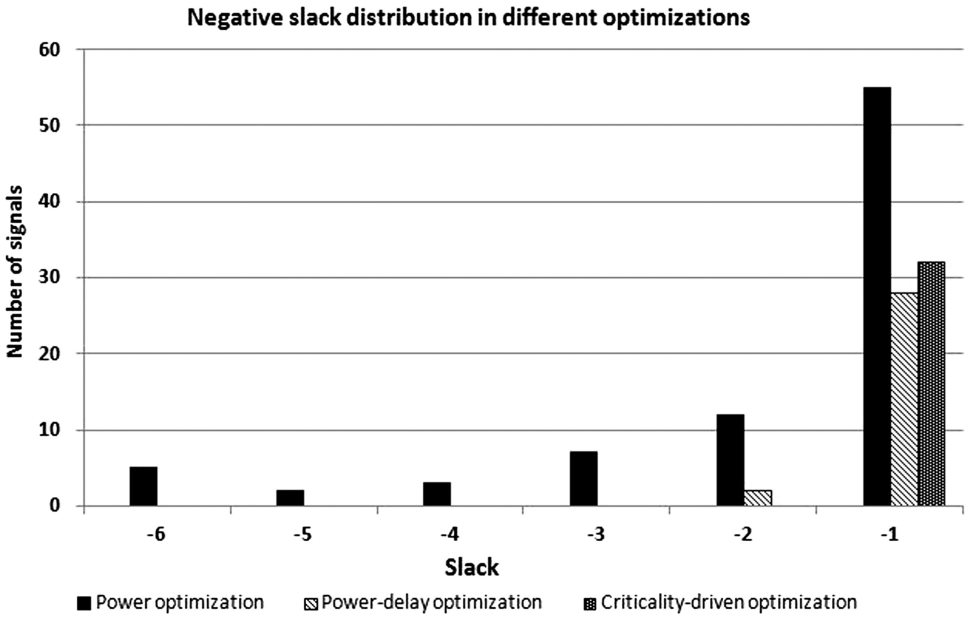

Fig. 13. Negative slack distributions created in different optimization types. In the initial circuit there were no signals with negative slack. The negative slacks were eliminated in the final experiment by "freezing" critical wires.

The algorithm was applied in the design of a 65 nanometer process technology high-end microprocessor, and yielded considerable dynamic power reduction. The technique is applicable as a post-processing step after detailed routing, and the achievable power saving depends on the density and style of the original layout. Signal delays are treated as constraints, but they can be optimized by modifying the power optimization techniques and then offer a systematic exploration in the power-delay design space.

Further dynamic power reduction is potentially possible by optimizing wire spacing in the underlying lower-level functional blocks. Unfortunately, in 45 nm technologies and beyond, the spacing design rules of low-level metal layers have been drastically changed from continuous to discrete. Although continuous methods can be used to obtain approximate solutions, discrete optimization techniques are more appropriate, which are currently explored by the authors.

ACKNOWLEDGMENTS

The authors wish to thank A. Erez of Intel Corporation and J. Pogerov of Sagantec for implementing the spacing algorithm and W. Hendawi of Intel Corporation for useful comments.

REFERENCES

Abou-Seido, A. I., Nowak, B., and Chu, C. 2002. Fitted Elmore delay: A simple and accurate interconnect delay model. In *Proceedings of IEEE International Conference on Computer Design*, 422–427.

BAKOGLU, H. 1990. *Circuits, Interconnections, and Packaging for VLSI*. Addison-Wesley.

BOESE, K. D., KAHNG, A. B., MCCOY, B. A., AND ROBINS, G. 1993. Fidelity and near-optimality of Elmore-based routing constructions. In *ICCAD Digest of Technical Papers*, 81–84.

BORKAR, S. 2001. Low-power design challenges for the decade. *Proceedings of the Conference on Asia South Pacific Design Automation*, 293–296.

CEDERBAUM, I., KOREN, I., AND WIMER, S. 1992. Balanced block spacing for VLSI layout. *Disc. Appl. Math. 40*, 3, 308–318.

CHAUDHARY, K., ONOZAWA, A., AND KUH, E. 1993. A spacing algorithm for performance enhancement and cross-talk reduction. In *Proceedings of the IEEE/ACM International Conference on CAD*, 697–702.

CHILUVURI, V. K. R. AND KOREN, I. 1995. Layout-synthesis techniques for yield enhancement. *IEEE Trans. Semiconductor Manufactur. 8*, 2, 178–187.

CHEN, P., KIRKPATRICK, D. A., AND KEUTZER, K. 2000. Miller factor for gate-level coupling delay calculation. In *Proceedings of the ICCAD*, 68–74.

CHENG, C.-K., LILLIS, J., LIN, S., AND CHANG, N. 1999. *Interconnect Analysis and Synthesis*. Wiley-Interscience.

CONG, J., HE, L., KOH, C. K., AND PAN, Z. 2001. Interconnect sizing and spacing with consideration of coupling capacitance. *IEEE Trans. Comput.-Aid. Des. Integr. Circ. Syst. 20*, 9, 1164–1169.

CORMEN, T. H., LEISERSON, C. H., AND RIVEST, R. L. 2001. *Introduction to Algorithms*, 2nd Ed. MIT Press.

DEVADAS, S. AND MALIK, S. 1995. A survey of optimization techniques targeting low-power VLSI circuits. *Proceedings of the 32nd ACM/IEEE Conference on Design Automation*, 242–247.

GAO, T. AND LIU, C. L. 1996. Minimum cross-talk channel routing. *IEEE Trans. Comput.-Aided Des. Integr. Care Syst. 15*, 5, 465–474.

GENOSSAR, D. AND SHAMIR, N. 2003. Intel® Pentium® M processor power estimation, budgeting, optimization, and validation. *Intel Technol. J. 7*, 43–50.

HANCHATE, N. AND RANGANATHAN, N. 2006. A linear-time algorithm for wire sizing with simultaneous optimization of interconnect delay and cross-talk noise. In *Proceedings of the 19th International Conference on VLSI Design*, 283–290.

HE, J-A AND KOBAYASHI, H. 1998. Simultaneous wire sizing and wire spacing in post-layout performance optimization. In *Proceedings of the ASP-DAC Design Automation Conference*, 373–378.

HO, R., MAI, K., AND HOROWITZ, M. 2001. The future of wires. *Proc. IEEE 89*, 4, 490–501.

HU, T. C. 1969. *Integer Programming and Network Flows*. Addison Wesley.

ITRS. 2005. ITTS report. http://www.itrs.net/reports.html

JHANG, K., HA, S., AND JOHN, C. 1994. A segment rearrangement approach to channel routing under the cross-talk constraints. In *Proceedings of the Asia-Pacific Conference on Circuits and Systems*, 536–541.

KAHNG, A., MASUKO, K., AND MUDDU, S. 1996. Analytical delay models for VLSI interconnects under ramp input. In *Proceedings of the IEEE International Conference on Computer-Aided Design (ICCAD)*, 30–36.

LI, C., XIE, M., CONG, C-K KOH, J., AND MADDEN, P. H. 2007. Routability-driven placement and white space allocation. *IEEE Trans. Comput.-Aided. Des. Integr. Circ. Syst. 25*, 5, 858–871.

LUENBERGER, D. G. 1984. *Linear and Nonlinear Programming*. Addison Wesley, Chapter 6.5.

MACII, E., PONCINO, M., AND SALERNO, S. 2003. Combining wire swapping and spacing for low-power deep-submicron buses. In *Proceedings of the 13th ACM Great Lakes Symposium on VLSI*, 198–202.

MAGEN, N., KOLODNY, A., WEISER, U., AND SHAMIR, N. 2004. Interconnect-Power dissipation in a microprocessor. In *Proceedings of the International Workshop on System Level Interconnect Prediction*, 7–13.

MIYOSHI, T., WAKABAYASHI, S., KOIDE, T., AND YOSHIDA, N., 1995. An MCM routing algorithm considering crosstalk. In *Proceedings of the International Symposium on Circuits and Systems*, 211–214.

MUI, M. L., BENERJEE, K., AND MEHORTRA, A. 2004. A global interconnect optimization scheme for nanometer scale VLSI with implications for latency, bandwidth, and power dissipation. *IEEE Trans. Electon. Devices 51*, 2, 195–203.

ONAZAWA, A., CHAUDHARY, K., AND KUH, E. S. 1995. Performance driven spacing algorithm using attractive and repulsive constraints for submicron LSI's. *IEEE Trans. Comput.-Aided. Des. Integr. Circ. Syst. 14*, 707–719.

SAXENA, P. AND LIU, C. L. 2000. An algorithm for cross-talk-driven wire perturbation. *IEEE Trans. Comput.-Aided. Des. Integr. Circ. Syst. 19*, 6, 691–702.

SESHU, S. AND REED, M. B. 1961. *Linear Graphs and Electrical Networks*. Addison-Wesley, Reading, MA.

SYLVESTER, D. AND KEUTZER, K.. 1998. Getting to the bottom of deep submicron. In *Proceedings of the IEEE/ACM International Conference on CAD*, 203–211.

WANG, D. AND KUH, E. S. 1998. A performance driven MCM router with special considerations of cross-talk reduction. In *Proceedings of the Design Automation and Test in Europe*, 466–470.

WIMER, S., KOREN, I., AND CEDERBAUM, I. 1988. Floorplans, planar graphs, and layout. *IEEE Trans. Circ. Syst. 35*, 3, 267–278.

WIMER, S., MICHAELY, S., MOISEEV, K., AND KOLODNY, A. 2006. Optimal bus sizing in migration of processor design. *IEEE Trans. Circ. Syst. I 53*, 5, 1089–1100.

XTREME. *A Wire Spacing Tool for Manufacturing Yield Enhancement*. Sagantec.

Received October 2008; revised January 2009; accepted March 2009