IRWIN AND JOAN JACOBS
CENTER FOR COMMUNICATION AND INFORMATION TECHNOLOGIES

Wire Spacing, Planar Graphs and the Minimization of Dynamic Power in VLSI Microprocessors

Konstantin Moiseev, Shmuel Wimer and Avinoam Kolodny

CCIT Report #695
April 2008

DEPARTMENT OF ELECTRICAL ENGINEERING
TECHNION - ISRAEL INSTITUTE OF TECHNOLOGY, HAIFA 32000, ISRAEL

CCIT Report #695 April 2008

Wire Spacing, Planar Graphs and the Minimization of Dynamic Power in VLSI Microprocessors

Konstantin Moiseev, Shmuel Wimer and Avinoam Kolodny

Abstract

The problem of optimal space allocation among interconnecting wires of VLSI chips, in order to minimize their switching power consumption is solved. Necessary and sufficient conditions for the existence of optimal space allocation are derived, stating that every wire must be in equilibrium of its line-to-line weighted capacitance density on its two opposite sides. Two proofs are presented, one based on convexity of the dynamic power objective, and another based on a graph representation of the problem. The notion of power density is introduced and it is proven that power is minimal if and only if its density is uniformly distributed across the entire layout. This condition is shown to be equivalent to all paths of the layout 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 is presented, and implications on circuit timing are discussed.

1. Introduction

The power consumed by VLSI systems is a significant factor in the design of new microprocessors and other products. The main reason for increased power dissipation 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 very low power 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 was addressed at various design levels [1][2], from architecture and system level through RTL synthesis, signal encoding, circuit implementation and layout implementation, which is the focus of this paper. The interconnect power dissipated because of charging and discharging wire capacitances is a dominant component in processors [3]. A typical breakdown of dynamic power dissipation of a high-end microprocessor designed in 65 nanometer process technology is illustrated in Fig. 1, indicating that global wires at the top metal layers generate 20% of the total dynamic power, and half of this power is due to cross-coupling between adjacent wires at the same layer. We show in this paper how it can be significantly reduced by optimizing inter-wire spacing in the layout.

Commercial routing tools and manual artwork of mask designers tend to produce congested wires. Tools and humans do not typically 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 [4]. However, this approach results in non-uniform 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 Fig. 2.

Based on this observation, we propose to eliminate the white space by spreading-out wires in a post processing algorithm, thus increasing and balancing inter-wire spaces in order to reduce capacitances and save power. It is assumed that interconnects have already been routed (manually or by some CAD tool), and their relative location is not subject to any change. It is also assumed that wire widths have been set to satisfy signal delay and other design goals such as reliability.

Design optimization by wire spacing has been discussed by many authors, for different purposes: Signal delay optimization [5] [6] [7] [8], power consumption minimization [9], cross-coupling noise reduction [10] [11] and yield enhancement [29], are just a few. The works in [9] [10] [11] address power minimization by local optimization. The optimization approach of this paper is somewhat reminiscent of that in [11] in the sense that they both rely on the convexity of line-to-line cross-coupling capacitance. However, unlike [11] which treats the problem locally, this paper looks at the entire layout at once, and finds a provable global optimal solution.

The authors in [11] used convexity arguments to prove the existence of minimum cross-coupling noise of a single net, followed by an effective method to find that 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 is required, which is the essence of this paper. Another difference is that the solution in

[11] is of two-dimensional routing and is suitable for channel and switchbox routing styles. This paper addresses the simultaneous optimization of the entire top-level microprocessor routing comprising many thousands of nets. Wire spacing optimization of the global routing layers in a processor is a collection of several, almost independent, one-dimensional problems. We take advantage of the one-dimensionality and the independency to obtain a robust and effective global optimization approach.

The rest of the paper is organized as follows. In the next section the circuit and layout model is presented. A necessary and sufficient power minimization condition is proven in section 3. In section 4 a graph model of wire spacing and line-to-line capacitance is introduced, yielding a graph-theoretical necessary and sufficient condition for global minimum power. The electrical meaning of this condition is discussed. Section 5 presents an algebraic solution of the problem based on the graph and network flow model. Section 6 presents an iterative algorithm that guarantees convergence to optimum. Results obtained for a recent high-end microprocessor designed in 65 nanometer process technology are presented in Section 7. Satisfaction of timing constraints and process technology design rules are discussed in section 8.

2. Interconnect modeling assumptions

The interconnecting wires of high metal layers typically run in alternating orthogonal directions, e.g. wires residing in even layers are vertical and wires in odd layers are horizontal, as shown in Fig. 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.

The switching of signals between voltage levels corresponding to logic 0 and 1 is the reason for dissipated interconnect power [12]. The interconnect power associated with a logic signal is proportional to its total capacitance and to its average amount of switching as compared to the clock signal, called the signal's ***activity factor*** [13].

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 Fig. 3, shifting wires in one layer doesn't affect spacing of the orthogonal wires in the layers above it and below it. Although the length of horizontal wires in layers $l-1$ and $l+1$ may slightly change, their variations are assumed negligible for any practical consideration. Odd layers behave similarly.

The fundamental model we use to derive optimal spacing conditions is shown in Fig. 4. There, a few wires run in parallel and the entire bundle is shielded on both sides by wires connected to power supplies, which are not allowed to move. Shielding wires do not make logical transitions; hence they do not consume any power.

Switching power consumed by a signal wire is associated with its capacitance to ground planes (representing the adjacent metal layers) and with line-to-line capacitance to other wires as shown in Fig. 4. We say that

two wires are "visible" to each other if they have some common span (See Fig. 5). For a given wire, only line-to-line capacitance to visible wires has influence on wire power. The progression of VLSI process technology has made this line-to-line term dominant over others [14] [15], and its importance is expected to grow in future generations [16]. The line-to-line capacitance between two adjacent wires is proportional to some power of their common span where they are "visible" to each other, and inversely proportional to some positive power of their space to each other [11].

Every signal σ_i assumes some activity factor α_i ranging from $\alpha_i = 0$ if it never switches (e.g., shields or power delivery network), to $\alpha_i = 1$, if it switches twice at every cycle (e.g., clocks). 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. 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 [30]. For calculating cumulative average power over many transitions, an average MCF of 1 is assumed. Under this assumption the power contributed by the line-to-line capacitance between σ_i and σ_j is proportional to $\alpha_i + \alpha_j$.

Let $I_0, I_1, ..., I_n, I_{n+1}$ be $n+2$ vertical wires, where I_0 and I_{n+1} are leftmost and rightmost shields and $\alpha_0 = 0, \alpha_1, ..., \alpha_{n,} \alpha_{n+1} = 0$ their corresponding activity factors. A partial order $\prec$ is defined on wires $I_0, ..., I_{n+1}$ as follows.

We say that $I_i \prec I_j$ if I_i and I_j satisfy: 1) the intersection of their vertical span is non empty, 2) x_i and x_j, the abscissas of I_i and I_i, respectively, satisfy $x_i < x_j$, and I_i and I_i are visible to each other. This is a left-to-right topological order of the wires, and in the rest of the paper we'll assume that they are ordered. Wire spacing optimizations preserve the initial order of the wires.

We assume that the widths $w_0, w_1, ..., w_{n,} w_{n+1}$ of the wires are predefined and thus are not subject to change in the 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 space, so the unused "white area" can be redistributed among wires in order to reduce their line-to-line 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 span l_{ij} may consist of several segments since two wires can be visible and hidden from each other several times. 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:

$$x_j - x_i - \left(w_j + w_i\right)/2 \geq S_{min}, \quad I_i \prec I_j \quad . \tag{1}$$

Inequality (1) 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 *minimum spacing rule*.

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

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

The factor κ depends only on process technology, while $\eta \geq 1$ and $\gamma \geq 1$. Various papers used different values of η and γ . A setting of $\eta = 1$ and $\gamma = 1$ is assumed in [17], [18] and [19]. Other authors use the setting $\eta = 1$ and $\gamma = 1.34$ [20] [21]. Similar to [11], the results of this paper are applicable for any setting of the above parameters.

The total switching power $P^{cross}\left(\overline{x}\right)$ resulting from line-to-line capacitance is therefore proportional to:

$$P^{cross}\left(\overline{x}\right) \propto \sum_{0 \leq i \leq n} \sum_{i < j \leq n+1} \left(\alpha_j + \alpha_i\right) c_{ij} = \kappa \sum_{0 \leq i \leq n} \sum_{i < j \leq n+1} \frac{\left(\alpha_j + \alpha_i\right) l_{ij}^{\eta}}{\left[x_j - x_i - \left(w_j + w_i\right)/2 \right]^{\gamma}} . \tag{3}$$

The goal is to find $\overline{x} = \left(x_1, ..., x_n\right)$ that minimizes (3). Recall that I_0 and I_{n+1} are fixed, hence we assume that $x_0 = 0$ and $x_{n+1} = A$.

3. Necessary and sufficient condition for minimal power

Lemma 1: The minimum of (3) subject to (1) is global.

Proof: Let us define $s_{ij} = x_j - x_i - \left(w_j + w_i\right)/2$ to be the spacing between two visible wires. Substitution s_{ij} into (1) and (3) yields the following minimization problem:

$$\text{minimize:} \quad \kappa \sum_{0 \le i \le n} \sum_{i < j \le n+1} \left(\alpha_j + \alpha_i\right) l_{ij}^{\eta} \Big/ s_{ij}^{\gamma} , \tag{4}$$

$$\text{subject to:} \quad s_{ij} \ge S_{min}, \quad I_i \prec I_j , \tag{5a}$$

$$s_{ij} - x_j + x_i + \left(w_j + w_i\right)/2 = 0, \quad I_i \prec I_j \text{, and,} \tag{5b}$$

$$0 < x_i < A, \quad 1 \le i \le n . \tag{5c}$$

The objective function (4) is convex (see appendix of [11]) and same are the constraints (5a)-(5c). Consequently there is one minimum which is global [22]. ☻

Consider now the abscissa x_i of a wire I_i whose width is w_i $1 \le i \le n$. Denote its left and right visible wires by $I_{i,j}^{l}$ and $I_{i,j}^{r}$, respectively, where the superscript designates left and right sides of I_i and in the subscript j is varying. We use the same indexing notation for the corresponding abscissas, widths, lengths of wires overlap and activity factors.

Let us ignore for the moment the requirement (5a) of minimum spacing, and replace it by $s_{ij} > 0$, which still guarantees the partial order preservation in (5b). Although it is not feasible for VLSI layout, it simplifies the characterization of the optimal spacing yielding minimum power. We'll

return to (5a) and take it into account in the real implementation of wire spacing. Formally, (5a) is replaced by

$$s_{ij} > 0, \quad I_i \prec I_j \,. \tag{5d}$$

We may assume that there exist no $I_i \prec I_j$ and $\alpha_i = \alpha_j = 0$ such that I_i is enclosed (nested) in I_j. Mathematically it creates an undefined solution (since then $\alpha_i + \alpha_j = 0$, and the denominator of terms in (6) below can be arbitrarily small). However, if this was the case in layout, it means that there are two shielding wires residing one next to the other, while one is shielding completely the other. Consequently the smaller (nested) one is redundant and could be dropped.

Theorem 1 *(necessary and sufficient condition for minimal interconnect power):* A necessary and sufficient condition so that the switching power expression in (4) is minimized subject to the constraints (5b)-(5d) is that every wire $I_i, 1 \le i \le n$ satisfies:

$$\sum_j \frac{\left(l_{i,j}^l\right)^\eta \left(\alpha_i + \alpha_{i,j}^l\right)}{\left[x_i - x_{i,j}^l - \left(w_i + w_{i,j}^l\right)\right]^{\gamma+1}} = \sum_k \frac{\left(l_{i,k}^r\right)^\eta \left(\alpha_i + \alpha_{i,k}^r\right)}{\left[x_{i,k}^r - x_i - \left(w_i + w_{i,k}^r\right)\right]^{\gamma+1}} \,. \tag{6}$$

Summation on left and right hand sides of (6) is taken on all left and right visible wires, respectively.

Proof: By substitution of (5b) into (4) it follows that the power consumed by wire I_i is proportional to:

$$\sum_j \frac{\left(l_{i,j}^l\right)^\eta \left(\alpha_i + \alpha_{i,j}^l\right)}{\left[x_i - x_{i,j}^l - \left(w_i + w_{i,j}^l\right)\right]^\gamma} + \sum_j \frac{\left(l_{i,k}^r\right)^\eta \left(\alpha_i + \alpha_{i,k}^r\right)}{\left[x_{i,k}^r - x_i - \left(w_i + w_{i,k}^r\right)\right]^\gamma}, \tag{7}$$

The minimum of (4) is obtained at an internal point of the region $s_{ij} > 0, \quad I_i \prec I_j$, defined by (5d). Otherwise, there would be some $s_{ij} = 0$. This however will result (4) 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 (4) is that its derivative by the abscissa of every wire is zero. Differentiation of (7) by x_i yields (6). ☻

The physical interpretation of Theorem 1 is that it is necessary and sufficient for minimum interconnect power that every wire will be in equilibrium, where the sum of its left side weighted capacitors derivatives is equal to that of the right side.

Notice that equilibrium property is preserved for any cross capacitance model that is a convex function of wires' abscissas. If the model in (2) is replaced by a more general, then (6) will take the form:

$$\sum_j \left(\alpha_i + \alpha_{i,j}^l\right) \frac{\partial C(x_i, x_{i,j}, l_{i,j})}{\partial x_{i,j}} = \sum_k \left(\alpha_i + \alpha_{i,k}^r\right) \frac{\partial C(x_i, x_{i,k}, l_{i,k})}{\partial x_{i,k}},$$

where $C\left(x_i, x_j, l_{ij}\right)$ is the cross-capacitance function.

Solving (6) for all wires together with the constraints (5b)-(5d) involves a large number of nonlinear equations and linear inequalities. Its solution for a typical VLSI layout can be very tedious. The next section presents two alternative solutions which address 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. Such consequence leads into two different algorithms. The first is an algebraic solution. The other is a combinatorial iterative algorithm. Both of them yield the optimal solution.

Let us build a wire visibility graph and show how minimal power consumption can be captured by satisfaction of some properties of that graph. ***Spacing visibility graph*** $G(U, E, \overline{\xi})$ is a directed graph whose vertices U correspond to wires and arcs E correspond to spacing 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 blue vertices and arcs in Fig. 5 illustrate the graph overlaying the original layout.

An arc e_{ij} is assigned with the real positive number $\xi_{ij} = s_{ij} + \left(w_i + w_j\right)/2$ 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 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 $G(U, E, \overline{\xi})$, then

$$\sum_{e_{ij} \in \gamma_k} \xi_{ij} = \sum_{e_{ij} \in \gamma_k} s_{ij} + \left(w_i + w_j\right)/2 = A, \ \gamma_k \in \Gamma \quad . \tag{8}$$

It follows from planarity of *G* that there exists a dual graph $H(V, F, \overline{\eta})$, illustrated in Fig. 5 in red color. We call it ***weighted capacitance derivative graph***. It is defined as follows. Define a source and sink vertices v_0 and sink 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 [23].

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

$$\eta_{ij} = \left(l_{ij}\right)^{\eta} \left(\alpha_i + \alpha_j\right) \Big/ s_{ij}^{\gamma+1} \ . \tag{9}$$

The expression in (9) is the absolute value of the derivative of c_{ij} by any of the abscissas x_i or x_j, weighted by the activity factors of the wires forming the space s_{ij}.

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 Fig. 5. The graph $H(V, F, \overline{\eta})$ thus defined is also directed and acyclic, having one source and one sink. Fig. 5 illustrates the overlay of the dual graphs.

In the above 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 capacitors derivatives.

It follows from the invariance of *G*'s topology under repositioning wires and duality that *H*'s topology is also invariant. This implies that any vertical stack of capacitors, corresponding to a source-to-sink path in *H* is preserved in layout, regardless of the abscissas of $I_0, ..., I_{n+1}$. This is shown in Fig. 6, where *H* is overlaying the layout and the gray areas are the line-to-line capacitances. Notice 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 Fig. 6.

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 [24]. 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 sub-paths $\rho' \subset \delta'$ and $\rho'' \subset \delta''$ extending

between v_r and v_s. At v_s δ' and δ'' merge again up to v_m, as illustrated in Fig. 7. The physical interpretation of ρ' and ρ'' is of the left and right side stacked capacitors shown in Fig. 6.

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 sub-paths have the same length.

Proof: Fig. 7 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 sub-paths have different lengths. Then, two successive source-to-sink paths must exist in the above defined order; one is longer than the other, since except the two distinct sub-paths they share common arcs, hence a contradiction.

Conversely, let left and right sub-paths 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. ☻

Theorem 2 *(necessary and sufficient condition for minimum interconnects power):* The total interconnect switching power 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 switching power. ☻

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

$$\sum_{f_{ij} \in \delta_k} \eta_{ij} = \sum_{f_{ij} \in \delta_k} l_{ij} \left(\alpha_i + \alpha_j\right) \Big/ s_{ij}^{\gamma+1} = B, \ \delta_k \in \Delta \quad . \tag{10}$$

A consequence of Theorem 2 is that at 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 $c(x) = dC(x)/dx$ be its derivative, namely $C(x) = \int_{\xi=0}^{\xi=x} c\left(\xi\right) d\xi$. In this terminology, with the interpretation of a vertical scan-line as a source to sink path in H, it follows from Theorem 2 that:

Corollary 1 *(necessary and sufficient condition for minimum power):* The total interconnect switching power consumed in a layout is minimized if and only if its underlying line-to-line weighted capacitance density is constant.

5. Algebraic solution for power minimization

Let K and L be the coefficient matrices of (8) and (10), respectively. Then, combining the two in one matrix representation, they can be rewritten as:

$$\begin{pmatrix} K & 0 \\ 0 & L \end{pmatrix}\begin{pmatrix} \bar{\xi} \\ \bar{\eta} \end{pmatrix}=\begin{pmatrix} \bar{A} \\ \bar{B} \end{pmatrix}, \tag{11}$$

where $\bar{A}$ and $\bar{B}$ are corresponding vectors of the right hand side constants A and B in (8) and (10).

Although the number of paths can grow exponentially with the number of arcs, and hence the number of rows of the combined matrix in (11), we show in the sequel that a far smaller number of equations is sufficient. The graphs defined for the power minimization are similar to those used for floorplan area minimization in [23], where the rank of such a matrix was studied in [25]. Following is a citation and adoption to our terminology.

Let N_{wire} denote the number of wires in G and N_{space} the number of line-to-line capacitors. By [25] there exists:

$$\text{rank } K = |E| - |U| + 1 = N_{space} - N_{wire} + 2 \ . \tag{12}$$

It follows from the duality of G and H that their number of arcs is equal, hence $|F| = |E| = N_{space} + 1$. Moreover, the number of vertices in H is equal to the number of faces in G. By Euler's formula for planar graphs, stating

that $\#\ faces = \#\ arcs - \#\ vertices + 2$ there exists $|V| = |E| - |U| + 2 = N_{space} - N_{wire} + 3$. Similarly to (12), there exists for H:

$$\text{rank } L = |F| - |V| + 1 = N_{wire} - 1 \tag{13}$$

Summing the ranks in (12) and (13) we conclude that the rank of the combined matrix in (11) equals $N_{space} + 1$. Hence the number of independent equations is linear in the size of the layout.

There's still the question of how to effectively derive the $N_{space} + 1$ equations. To this end we'll interpret (8) and (10) as network cuts and flows [26]. 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 $G\left(U, E, \bar{\xi}\right)$ and $H\left(V, F, \bar{\eta}\right)$, thus creating new graphs $G'\left(U, E, \bar{\eta}\right)$ and $H'\left(V, F, \bar{\xi}\right)$. 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 in-coming 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 (11) yields a total of $N_{space} + 3$ equations, which can replace (8) and (10).

6. Iterative algorithms for power minimization

Though (11) is linear in $\bar{\xi}$ and $\bar{\eta}$, and the number of equations is linear in the size of the problem, there's still the nonlinearity relation to the abscissas $\bar{x}$ of

the wires. So instead of solving the equations explicitly, we'll use a simple and efficient, yet robust iterative algorithm. An iterative solution was used in [11] to find the optimal spacing of a single wire. Here we deal with a global problem involving thousands of wires simultaneously. It has been implemented and successfully used for power reduction in the design of a commercial 65 nanometer high-end microprocessor. Power reduction results are shown in Section 7.

The iterative algorithm is based on the equilibrium condition for minimum stated in Theorem 1. An iterative algorithm which utilizes vacant areas of layout in order to enhance manufacturing yield has been used by a commercial tool [27]. It is based on the balancing algorithm described in [28], where the speed of convergence is analyzed. This paper adopts the same algorithm with appropriate modifications to address power reduction.

The algorithm works on one 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 (6) is achieved. According to Theorem 1, at a non minimum point there exists at least one wire which is not in equilibrium. We then shift it to the abscissa x which satisfies (6). Article [28] proved that such iterations converge to a configuration where all wires are in equilibrium, namely (6) is satisfied for all wires.

The path lengths expressed in the constraints (8) are by definition invariant under repositioning of a single wire. Since initially the layout is legal, thus satisfying (8), it is automatically satisfied through the entire iterations.

It has yet to be seen that the repositioning of a single wire indeed reduces the total power. Considering (3), the only affected terms are those which involve the shifted wire and its left and right visible ones. These terms are expressed in (7). This amount of power appears only once in (3) and its value after repositioning has been lowered, hence the net power change is negative. We can conclude in the following theorem:

Theorem 3: The iterative algorithm which equilibrate wires one at a time converges to the global minimum of switching power.

Proof: The infinite sequence of power values obtained by the iterative algorithm is positive and monotonic 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 the pseudo code of the algorithm.

1. *initialization: for every wire calculate "distance" from equilibrium by equation (6)*
2. *put all wires into a heap*
3. *while top of heap is greater than some predefined* $\varepsilon > 0$ *do {*
4. *solve equation (6) for the wire at the top of the heap*
5. *locate the wire at abscissa found in line4*
6. *re-enter top wire to heap*
7. *for every visible wire do {*
8. *update "distance" from equilibrium by equation (6)*
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 connect to

A few implementation and complexity comments follow. In order to ensure fast convergence of the iterative algorithm, wires are put into a heap [24] in decreasing order of their distance from equilibrium. This is implemented in lines 1 and 2 of pseudo code. Assuming that the number of visible wires of any wire is bounded, which is the practical situation in VLSI layout, equilibration calculations consume $O(1)$ time per wire. Building the heap consumes $O(n \log n)$ time.

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 7-10 handles all the wires visible to the previous top wire that just has been re-entered into 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.

Once the convergence criterion in line 3 is met, it follows by the very definition of a heap that all wires are at ε distance from equilibrium or less. The dependency of run time on ε has been analyzed in [28]. Finally, line 12 retains layout connectivity.

So far S_{min} constraint in (1) 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 (6), it is checked whether S_{min} is violated. If this is the case then the wire stops at S_{min} "wall". The iterative algorithm still yields the minimum, though it may now be achieved at the boundary of the feasibility region rather than at an internal point as assumed in the proof of Theorem 1. The optimality can be verified from Lemma 1.

7. Experimental results

A pictorial example of real spacing optimization is shown in Fig.8, where next to every wire its corresponding activity factor is written. As shown in Fig. 8(b) the optimization algorithm distributed the spacing according to the relative weight of wires' activities.

The iterative algorithm presented in Section 6 was applied to the entire global routing layers in a 65 nanometer high-end microprocessor. Due to the large size of the data, the routing of that processor is divided into five portions. Optimization was then applied to every portion separately while maintaining boundary conditions to obtain proper interface and connectivity.

All top-level metal layers from 5th to 8th were optimized, while all connectivity and design rules were perfectly maintained.

Results are summarized in Table 1. Fig. 9 shows the breakdown and optimization obtained for each portion of global routing and each metal layer. The total dynamic power charged to global interconnects was reduced by 16.8%, which according to [3] and Fig. 1 is 1.68% of the total dynamic power consumption. In a real industrial design environment where the algorithm was deployed, such a reduction is very significant.

The difference in power reduction among the various portions is explained by differences in signal activities, metal density and existing wire spaces at each portion.

Portion No.	Power before (% of total)	Improvement (% relative)	Improvement (% of total)
1	58.82	14.2	8.35
2	16.89	20.9	3.53
3	11.55	21.6	2.50
4	6.66	17.8	1.18
5	6.08	20.5	1.25
Total	**100**		**16.81**

Table 1: Power reduction obtained for entire global routing

8. Maintaining delay constraints while minimizing power

The optimal line-to-line spacing which minimizes power is not necessarily optimal for delay. Although the improvement in cross capacitance will statistically work in favor of reducing delays, the changes may also result in max and min delay violations. The wire spacing described in this paper was applied at the finals stage of a design where timing was already stable. We therefore couldn't allow delay violations. Two different approaches to tackle this problem are described. The first one is preventive and avoids any delay violation. It was the practice used for the design mentioned in this paper. The other is a corrective approach which fixes violations after they have occurred.

Fig. 10 shows the wire spacing flow which prevents delay violations. Spacing is optimized first and all parasitics are modified accordingly. A timing simulation then discovers max and min delay violations. Spacing optimization is executed again on the original input data, excluding wires identified as sources of violations, together with their other visible wires. These are not allowed to move. Another timing simulation then takes place in order to check whether other delay violations popped. The optimization-simulation iterations continue until convergence. Usually two iterations suffice. In this flow some power saving is sacrificed in favor of avoiding delay violations.

A more aggressive, but more complicated approach is to restore all the original delays by post-resizing drivers in order to fix max and min delay violations. The top-level interconnects which are the subject of optimization can be views as a driver-receiver pair, where the wire resides at top-level while its driver and receiver belong to some lower-level functional blocks.

Fixing of max and min delay problems works in opposite directions. Driver upsizing which fixes max delay violation may cost some layout area and increase dynamic power consumption. Driver downsizing to correct min delay violation has an opposite effect. In what follows we'll be more pessimistic and consider the impact of fixing all delay changes rather than just max and min delay violations.

As a first step we need to express driver size sensitivity to delay change. A simplified Elmore delay model [12] of the driver-receiver pair in Fig. 11 is given by $D = \left(R + a\,L/W\right)\left(C + bLW + cL\left(1/S' + 1/S''\right)\right)$ where R is driver's resistance, L is wire length and W is its width, C is the capacitive load of the receiver, S' and S'' are the spaces on the two sides of the interconnecting wire, and a, b and c are process technology parameters. The sensitivity is then given by $\left(dR/dD\right)/\left(R/D\right) = (1 + aL/WR)$.

The sensitivity depends therefore on wire length and width, process technology sheet resistance and driver's resistance. Fig. 11 plots the change in percents that needs to take place by driver size in order to restore the delay for one percent of delay change, as a function of driver size. We simulated minimum width wires of several top-level metal layers with appropriate sheet resistance of 65 nanometer process technology. Several lengths $L = 500\mu m$, $1000\mu m$ and $3000\mu m$ were measured for driver's resistance varying from 50Ω to $1.5\text{k}\Omega$. Fig. 11 shows the results for the worst metal layer. As shown in the plot, driver size is more sensitive in longer interconnect, and strong (low resistance) drivers are more sensitive than weak (high resistance) ones. As an example, a change of 10% of delay

incurred at a signal with a driver of 100Ω and wire length of $1000\mu m$ is recovered by a change of 20% in driver size.

The histogram in Fig. 12 illustrates the distribution of delay change incurred in the top-level interconnects as a result of spacing optimization. As can be clearly seen, for about 80% of the interconnects the amount of change is negligible and falls in the range of simulation accuracy. We have therefore to restore the delays of 20% of the top-level interconnects. Recall that this is still worst case analysis since the delay change of majority of those doesn't result in max or min delay violation.

In order to calculate the amount of driver size changes implied by delay restoration, the histogram in Fig. 12 is combined with the driver size sensitivity in Fig. 11, thus yielding a distribution of driver size change shown in Fig. 13. This data is further used to calculate the amount of power growth resulting from resizing (both upsizing and downsizing), which eventually yielded 0.1% of the total chip power consumption. Recalling that Table 1 yielded 1.68% power save, we are left with 1.58% net power saving.

9. Conclusions

This paper solved the problem of optimizing wire spacing in order to minimize the interconnect switching power incurred in global routing metal layers of VLSI systems. A mathematically provable algorithm based on necessary and sufficient conditions and power density interpretation has been proposed. It was applied in a 65namometer process technology high-end microprocessor design 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. Yet, further dynamic power reduction is potentially possible by optimizing of wire spacing in the underlying lower-level functional blocks. Unfortunately in recent process technologies of 45nanometer and beyond the spacing design rules of low-level metal layers have been drastically changed from continuous to discrete. Though the continuous methods can be used to obtain approximated solution, discrete optimization techniques are more appropriate, which are currently explored by the authors.

Acknowledgement

The authors wish to thank to Amit Erez of Intel Corp. and Julian Pogerov of Sagantec for implementing the spacing algorithm and to Wael Hendawi of Intel Corp. for useful comments.

References

[1] S. Borkar, “Low power design challenges for the decade,” *Proc. of the 2001 Conf. on Asia South Pacific design automation*, pp. 293-296.

[2] S. Devadas and S. Malik, “A survey of optimization techniques targeting low power VLSI circuits,” *Proc. of the 32nd ACM/IEEE Conf. on design automation*, 1995, pp. 242-247.

[3] N. Magen, A. Kolodny, U. Weiser and N. Shamir, "Interconnect-power dissipation in a microprocessor*", Int. Workshop on System-level interconnect prediction*, pp. 7-13, Paris, 2004

[4] C. Li, M. Xie, C-K Koh, J. Cong and P. H. Madden, "Routability-Driven Placement and White Space Allocation," *IEEE Trans. on CAD of IC and Systems*, Vol. 25, No. 5, 2007, pp 858-871.

[5] J. Cong, L. He, C. K. Koh, and Z. Pan, "Interconnect sizing and spacing with consideration of coupling capacitance," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 20, no. 9, pp. 1164–1169, Sep. 2001.

[6] J-A He and H. Kobayashi, "Simultaneous wire sizing and wire spacing in post-layout performance optimization," *Proc. Of the ASP-DAC Design Automation Conf.* 1998, pp 373-378.

[7] S. Wimer, S. Michaely, K. Moiseev and A. Kolodny, "Optimal Bus Sizing in Migration of Processor Design*," IEEE Trans. Circuits and Systems – I,* Vol. 53, No. 5, 2006, pp. 1089-1100.

[8] N. Hanchate and N. Ranganathan "A linear time algorithm for wire sizing with simultaneous optimization of interconnect delay and crosstalk noise," *Proc. of the 19th Intl. Conf. on VLSI Design*, 2006, pp. 283-290.

[9] E. Macii, M. Poncino and S. Salerno, "Combining Wire Swapping and Spacing for Low-Power Deep-Submicron Buses," *Proc. of the 13th ACM Great Lakes Symp. on VLSI*, 2003, pp. 198-202.

[10] K. Chaudhary, A. Onozawa and E. Kuh, "A spacing algorithm for performance enhancement and cross-talk reduction," *Proc. of the 1993 IEEE/ACM Intl. Conf. on CAD*, pp. 697-702.

[11] P. Saxena and C. L. Liu, "A algorithm for crosstalk-driven wire perturbation," *IEEE Trans on CAD of IC and Systems*, Vol. 19, No. 6, 2000, pp. 691-702.

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

[13] D. Genossar, and N. Shamir, "Intel® Pentium® M Processor Power Estimation, Budgeting, Optimization, and Validation". *Intel Technology Journal*, Vol. 7, 2003, pp. 43-50.

[14] R. Ho, K. Mai and M. Horowitz, "The future of wires," *Proc. of the IEEE*, Vol. 89, No. 4, 2001, pp. 490-501.

[15] D. Sylvester and K. Keutzer, "Getting to the bottom of deep submicron," *Proc. of the 1998 IEEE/ACM Intl. Conf. on CAD,* pp. 203-211.

[16] 2005 ITRS report, available online http://www.itrs.net/reports.html

[17] T. Gao and C. L. Liu, "Minimum crosstalk channel routing," *IEEE Trans. on CAD of IC and Systems*, Vol. 15, No. 5, 1996, pp. 465-474.

[18] T. Miyoshi, S. Wakabayashi, T. Koide and N. Yoshida, "An MCM routing algorithm considering crosstalk," *Proc. Intl. Symp. Circuits and Systems*, 1995, pp. 211-214.

[19] D. Wang and E. S. Kuh, "A performance driven MCM router with special considerations of crosstalk reduction," *Proc. Design Automation and Test in Europe*, 1998, pp. 466-470.

[20] A. Onazawa, K. Chaudhary and E. S. Kuh, "Performance driven spacing algorithm using attractive and repulsive constraints for submicron LSI's," *IEEE trans. CAD of IC and Systems*, Vol. 14, pp. 707-719, 1995.

[21] K-S. Jhang, S. Ha and C. S. Jhon, "COP: A crosstalk optimizer for gridded channel routing," *IEEE Trans. CAD of IC and Systems*, Vol. 15, 1996, pp 424-429.

[22] D. G. Luenberger, Linear and Nonlinear Programming, Chapter 6.5, Addison Wesley, 1984.

[23] S. Wimer, I. Koren, I. Cederbaum, "Floorplans, planar graphs, and layout," *IEEE Trans. on Circuits and Systems*, Vol. 35, No. 3, 1988, pp. 267-278.

[24] T. H. Cormen, C. H. Leiserson and R. L. Rivest, Introduction to Algorithms, MIT Press, 2nd Edition. 2001.

[25] S. Seshu and M. B. Reed, Linear Graphs and Electrical Networks, Addison-Wesley, Reading, Mass., 1961.

[26] T. C. Hu, Integer Programming and Network Flows, Addison Wesley, 1969.

[27] Sagantec, Xtreme-a wire spacing tool for manufacturing yield enhancement.

[28] I. Cederbaum, I. Koren and S. Wimer, "Balanced block spacing for VLSI layout," *Discrete Applied Mathematics*, Vol. 40, Issue 3, 1992, pp. 308-318.

[29] V.K.R Chiluvuri and I. Koren, "Layout-synthesis techniques for yield enhancement," *IEEE Trams. On Semiconductor Manufacturing*, Vol. 8, Issue 2, 1995, pp. 178-187.

[30] P. Chen, D. A. Kirkpatrick and K. Keutzer, "Miller factor for gate-level coupling delay calculation," *Proc. IEEE/ACM Intl. Conf. on Computer-aided design*, 2000, pp. 68-75.

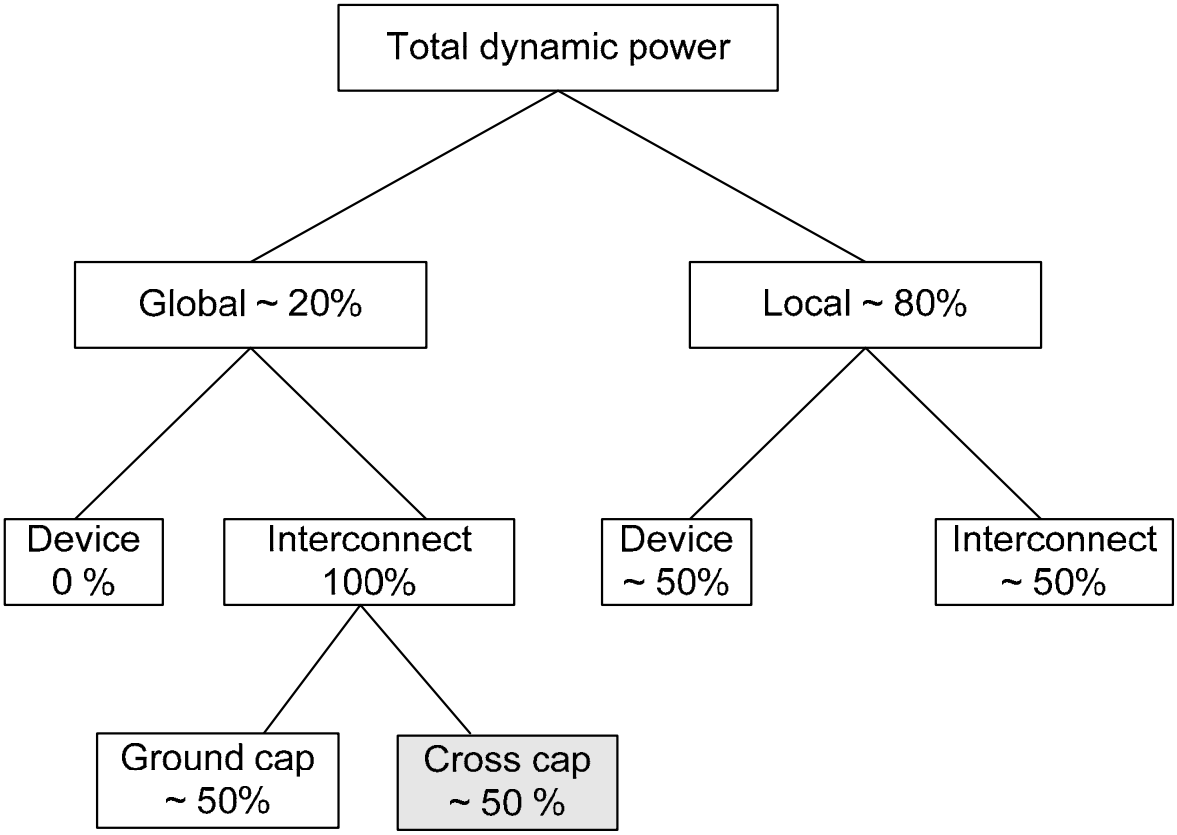


Figure 1: Breakdown of dynamic power into local blocks and global interconnects. As can be seen, the cross capacitances between global wires at the top routing layers contribute 10% of the total dynamic power.

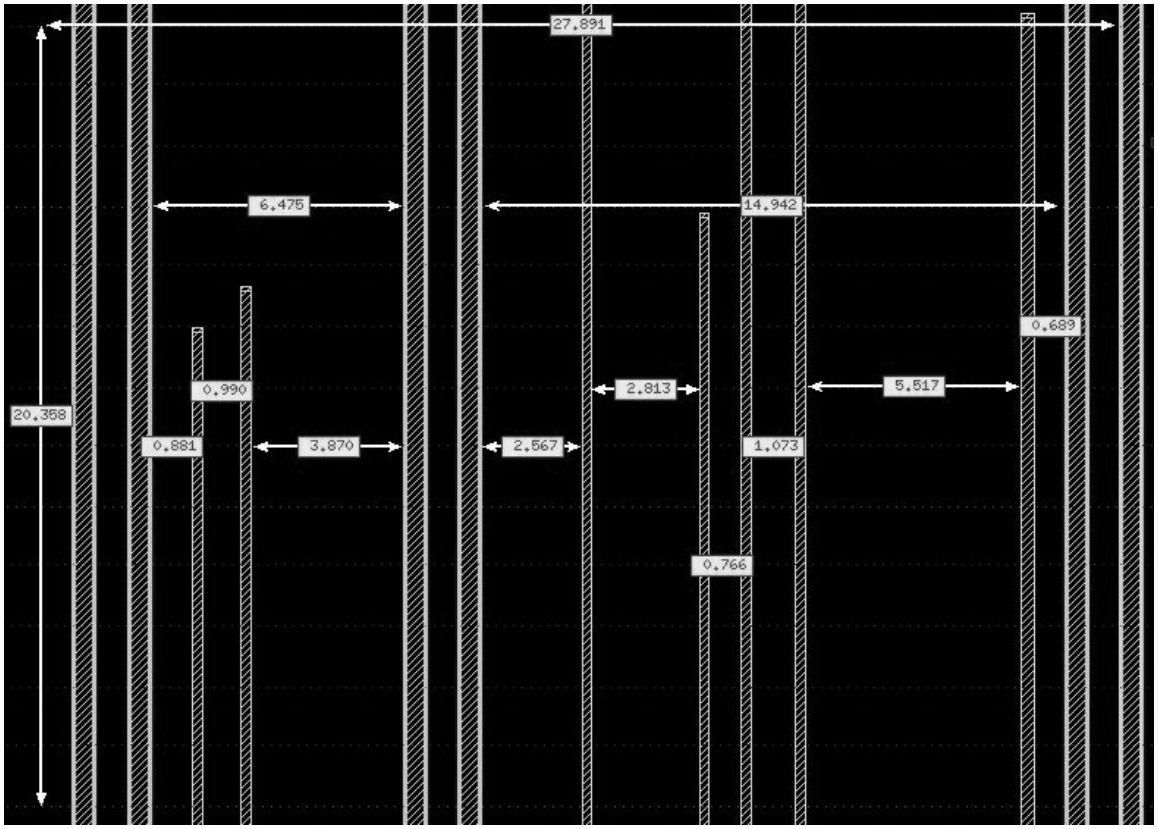


Figure 2: A clip of $20\mu\text{m} \times 28\mu\text{m}$ of 7^{th} metal layer routing taken from a $65n\text{m}$ process technology high-end microprocessor. The wide wires are VCC / VSS and are fixed. The narrow wires are signals routed automatically. The figure demonstrates the amount of white space found in layout and its inefficient distribution among signal wires.

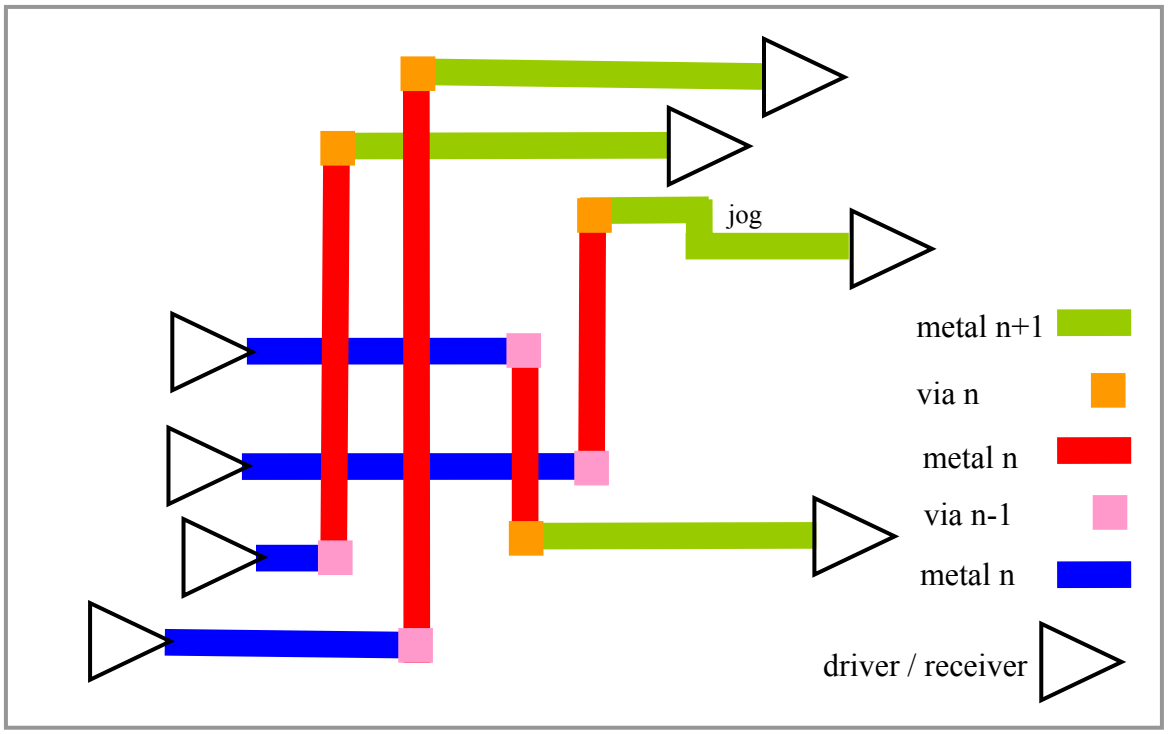


Figure 3: Typical interconnect patterns: A driver transmits a signal which propagates through interconnecting wires on various layers. Consecutive layers route wires in alternating orthogonal directions. Connections from layer to layer are made by vias. Some wires may have jogs.

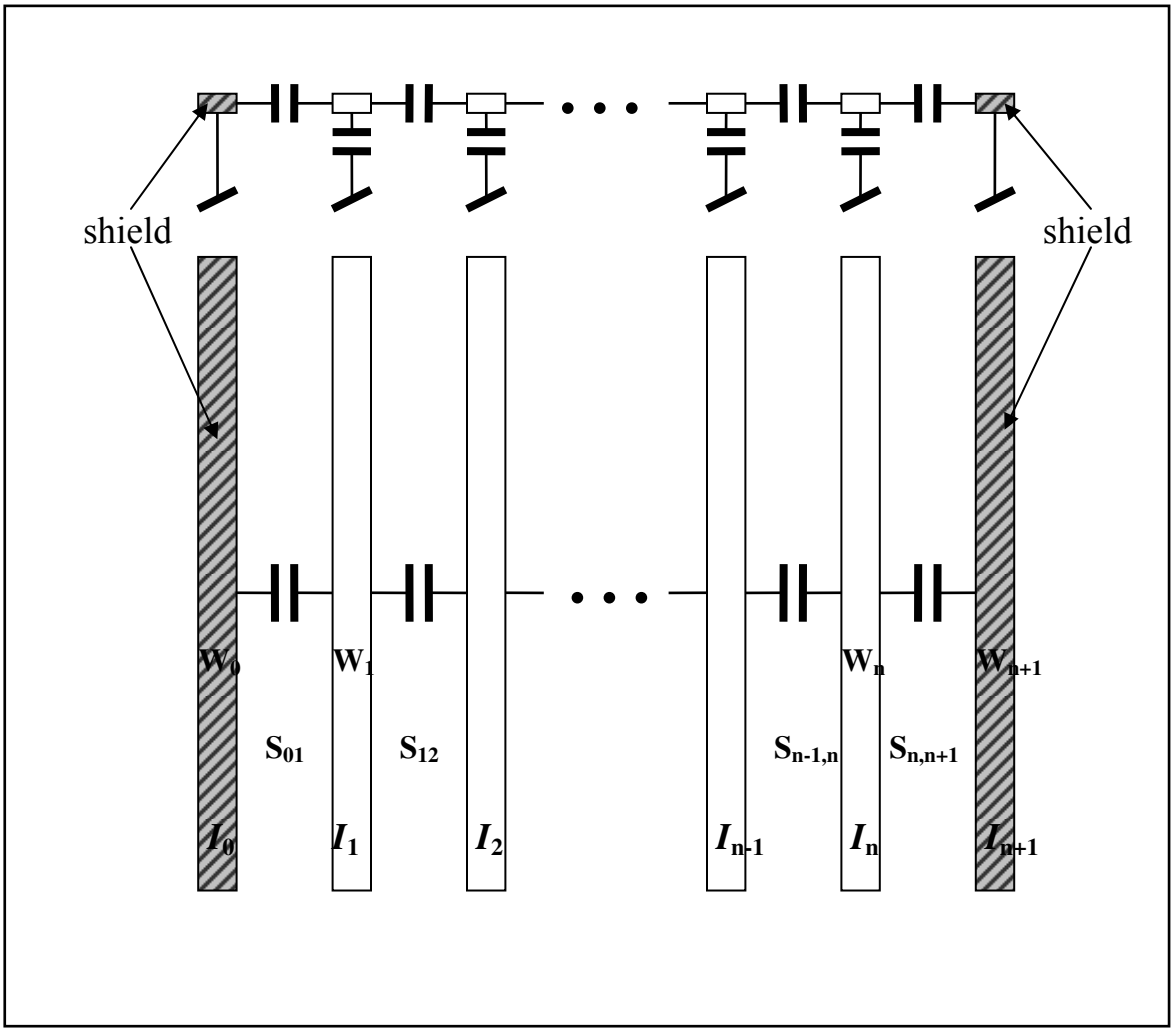


Figure 4: Fundamental cross coupling and ground capacitance. Wires run in parallel and the entire bundle is shielded on both sides by wires connected to ground.

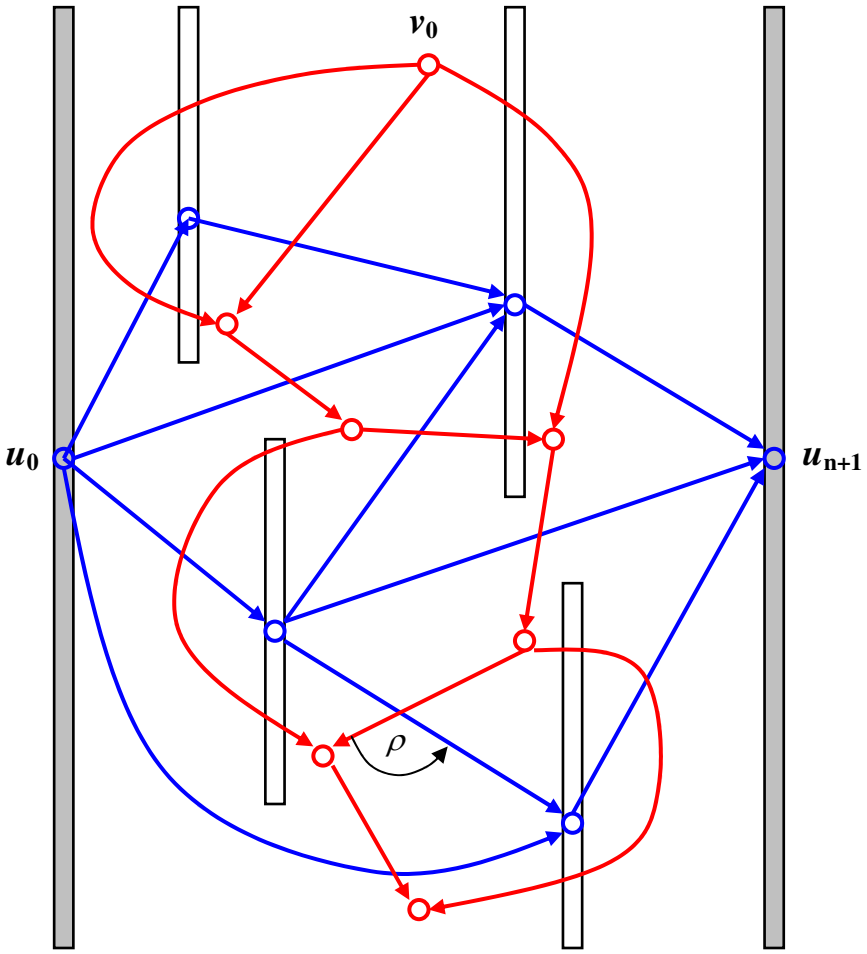


Figure 5: Spacing visibility graph overlaying its corresponding layout. Blue vertices and arcs comprise the primal graph corresponding to wires and their spacing. Red vertices and arcs comprise its dual graph corresponding to capacitances between visible wires. The preservation of the direction of ρ results in directed acyclic dual.

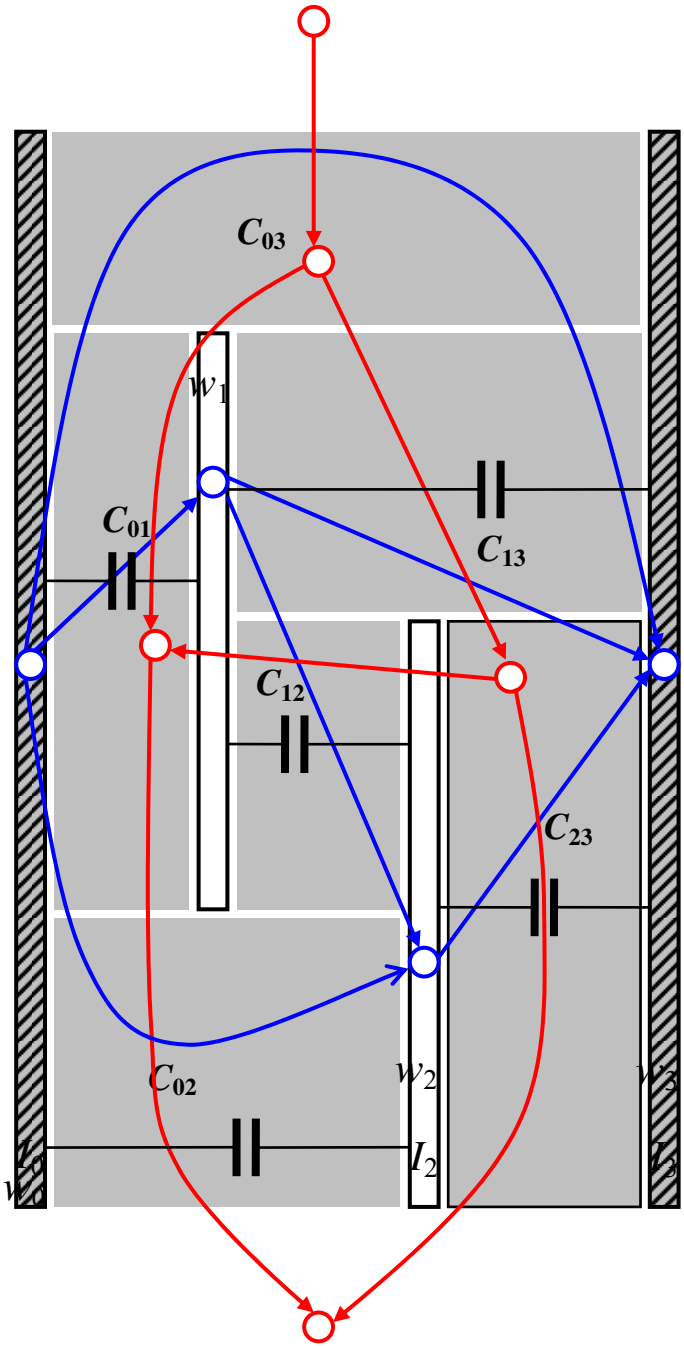


Figure 6: Cross-capacitance layout model with corresponding spacing visibility graph, and its weighted capacitance derivative dual. Gray areas correspond to line-to-line capacitors. Faces of the dual graph correspond to capacitors residing on the two sides of a signal wire.

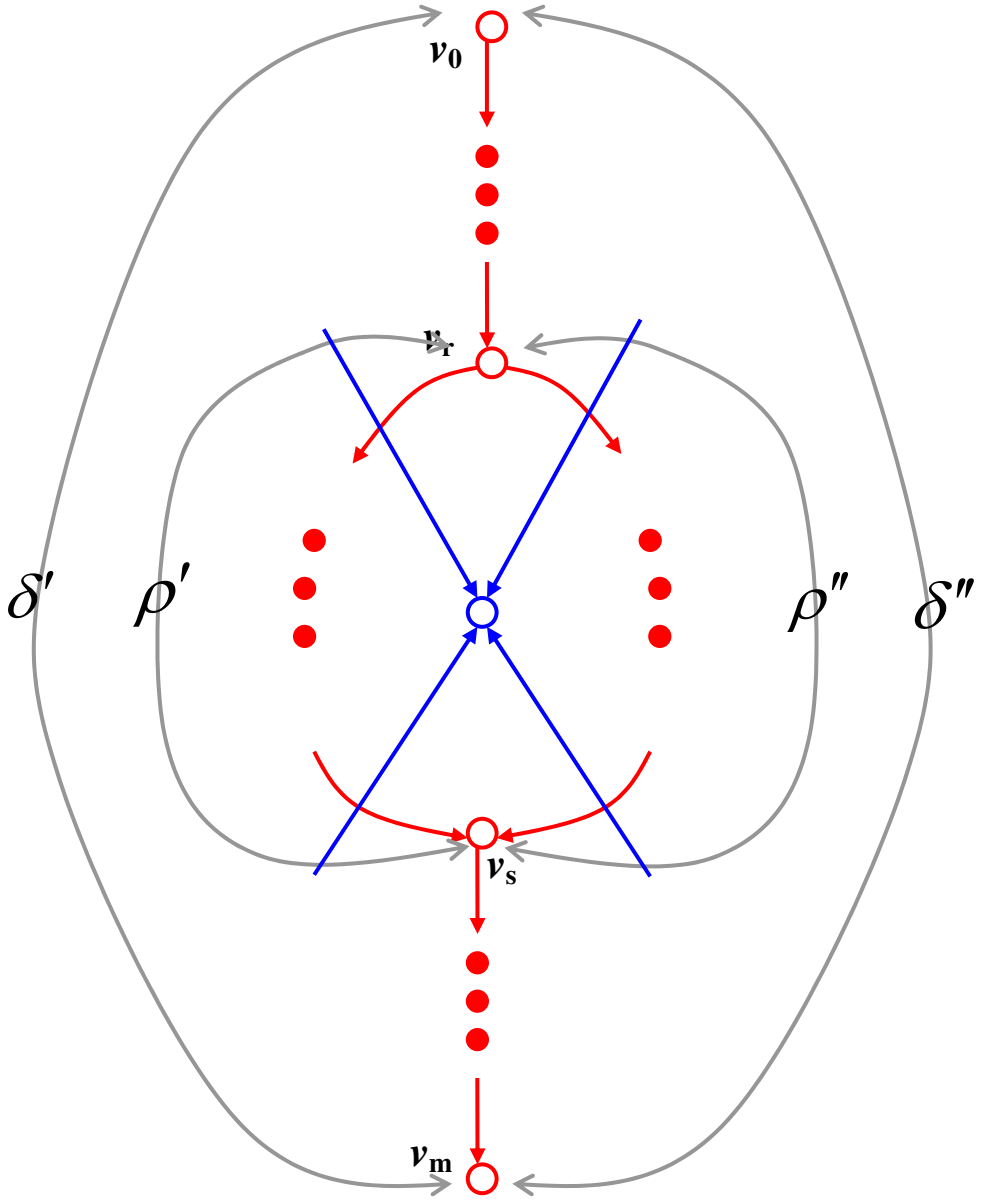


Figure 7: Proof of Theorem 2. Two "left to right" ordered path from v_0 to v_m in H consist of two common parts and a face enclosing a vertex of G.

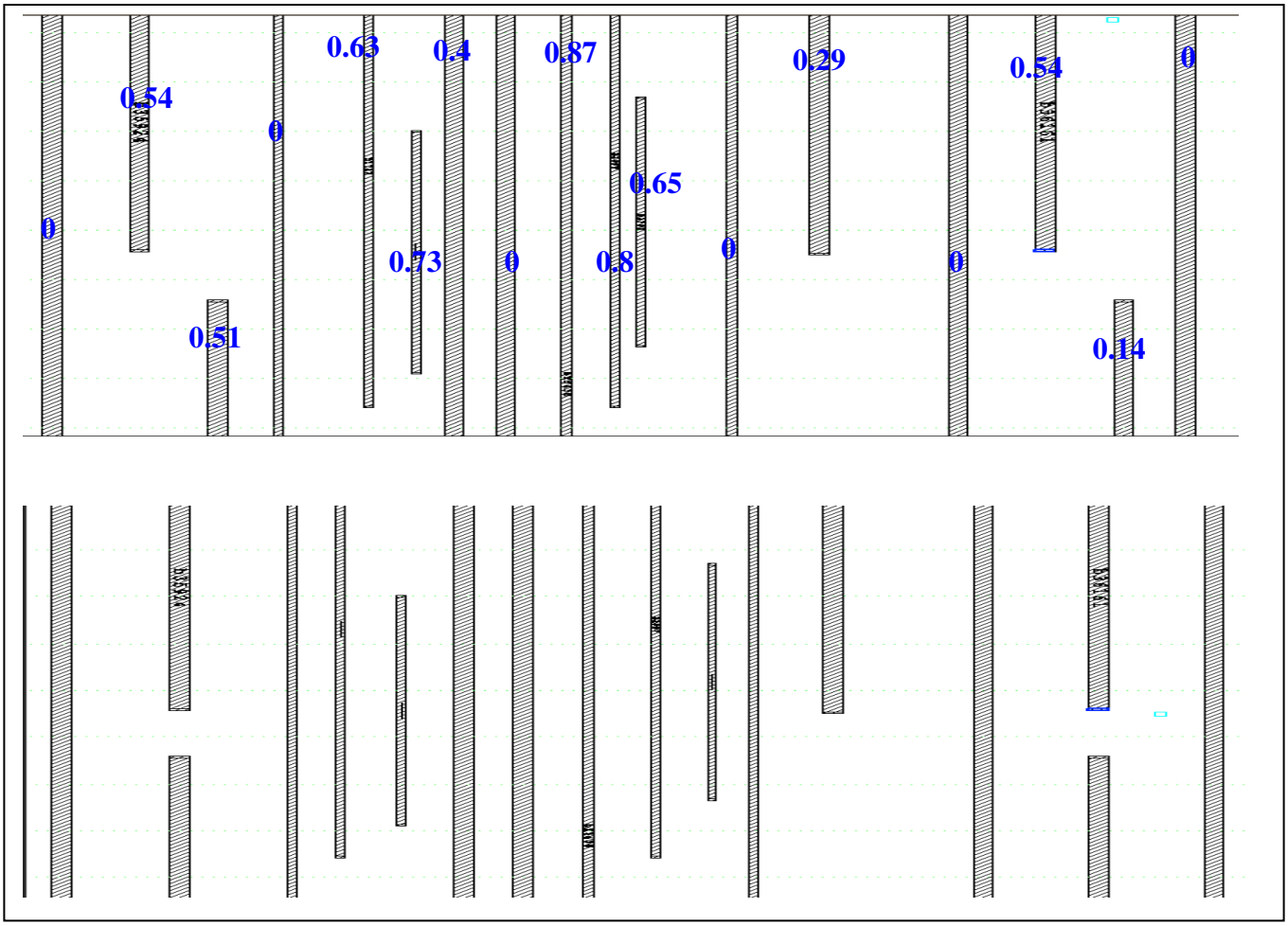
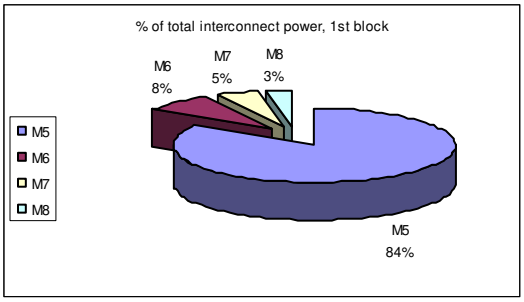
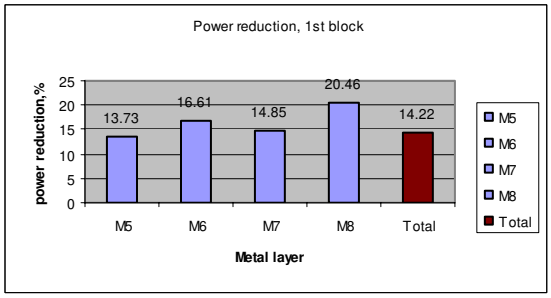
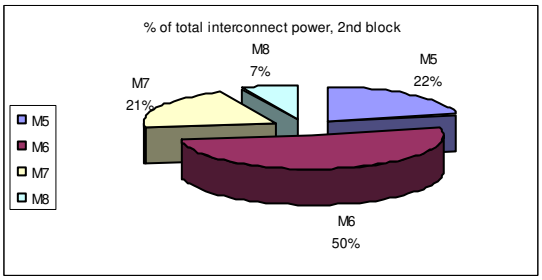


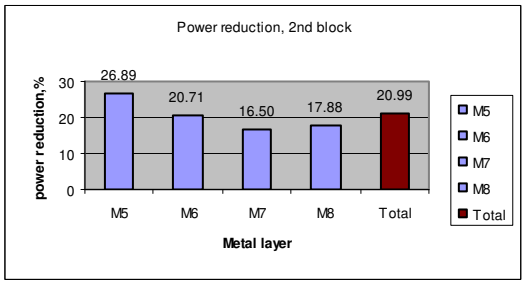
Figure 8: A clip of real spacing optimization and spacing distribution implied by activity factors.



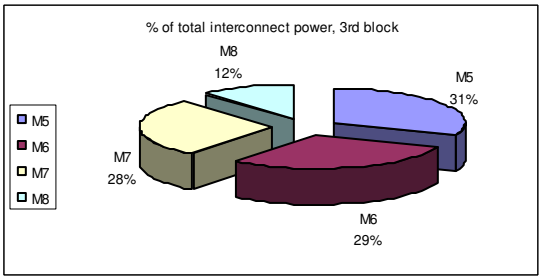


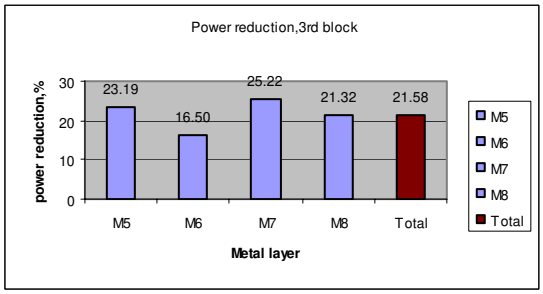
Block 2:



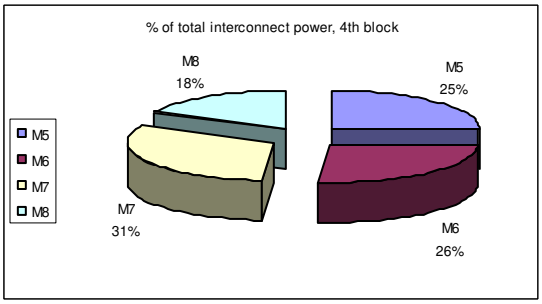


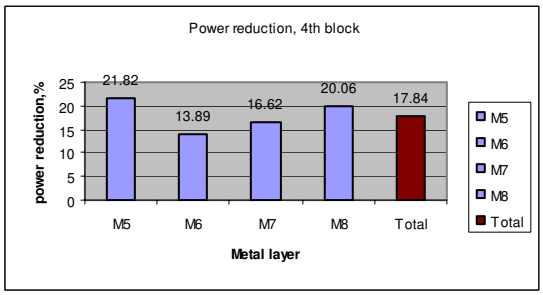
Block 3:





Block 4:





Block 5:

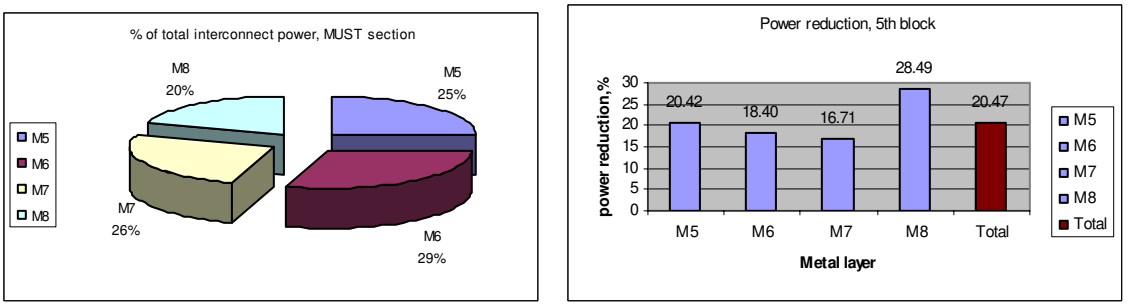


Figure 9: breakdown and optimization obtained for each portion of global routing and each metal layer.

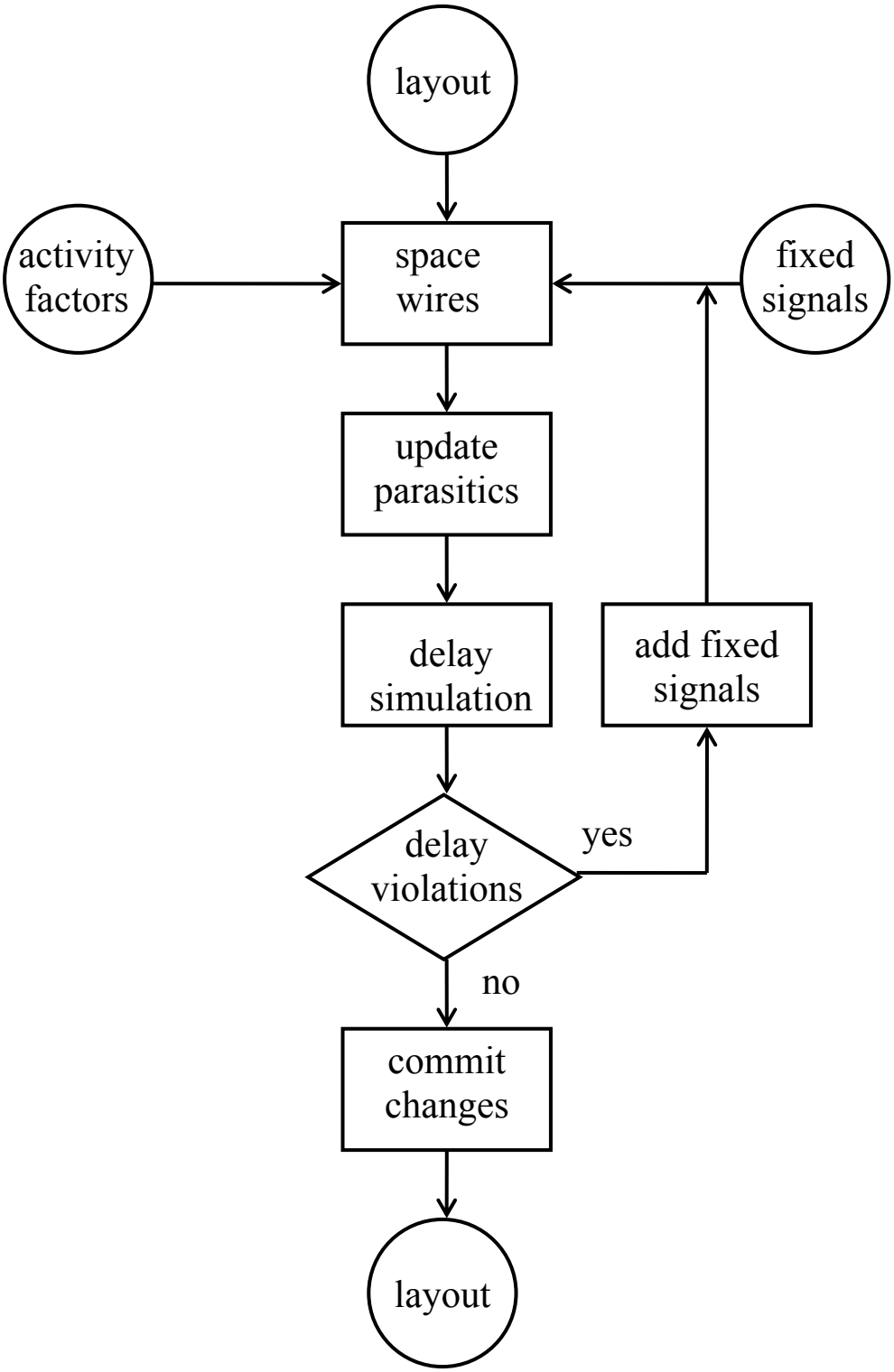


Figure 10: Power optimization flow which prevents delay violations.

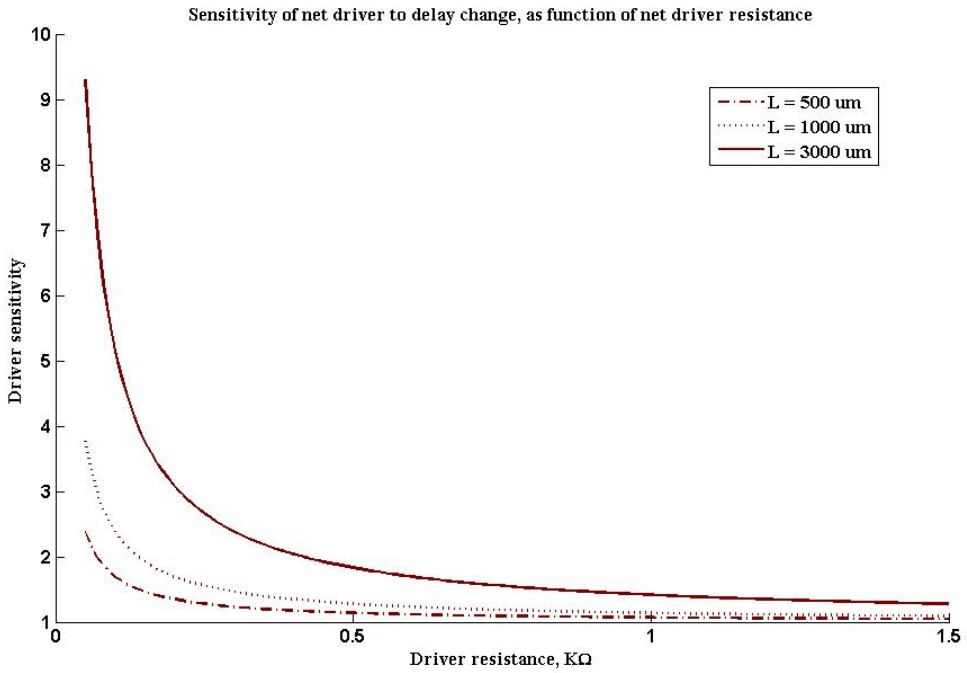


Fig. 11: Driver size sensitivity to delay change.

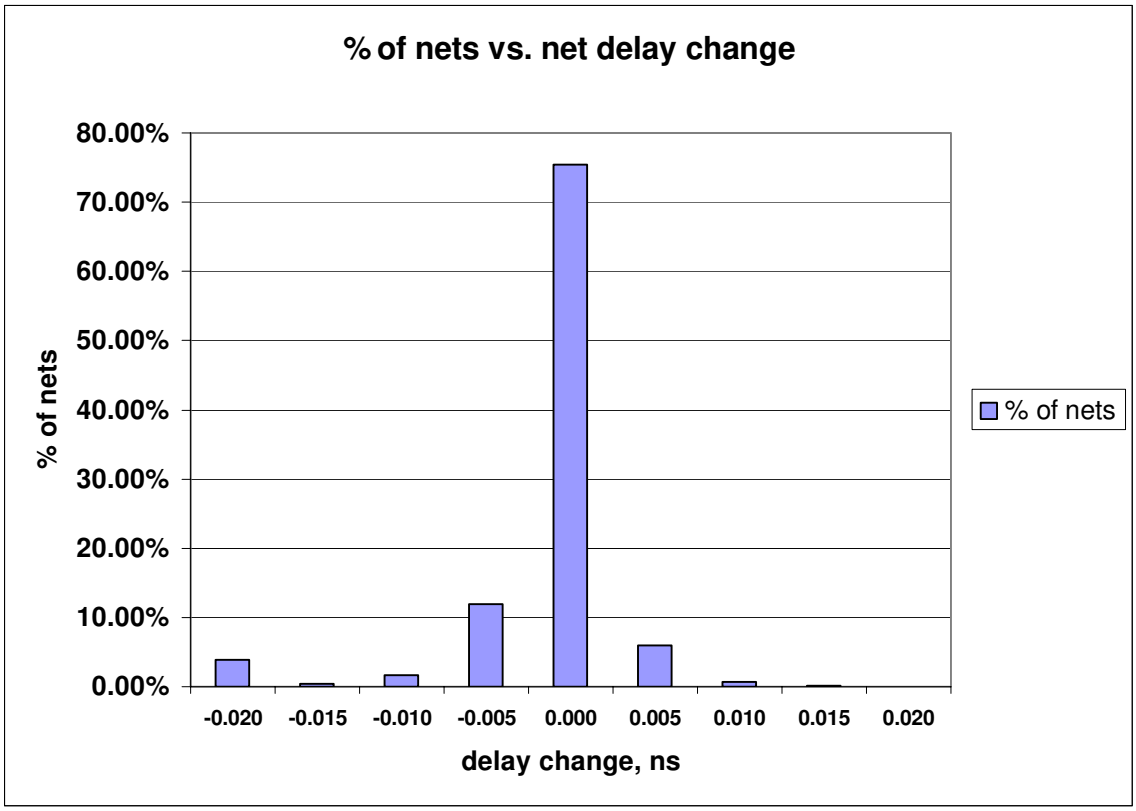


Fig. 12: Distribution of delay changes incurred by power minimization. The right tail corresponds to delay increase which may cause max delay violations. The left tail corresponds to delay decrease which may cause min delay violations.

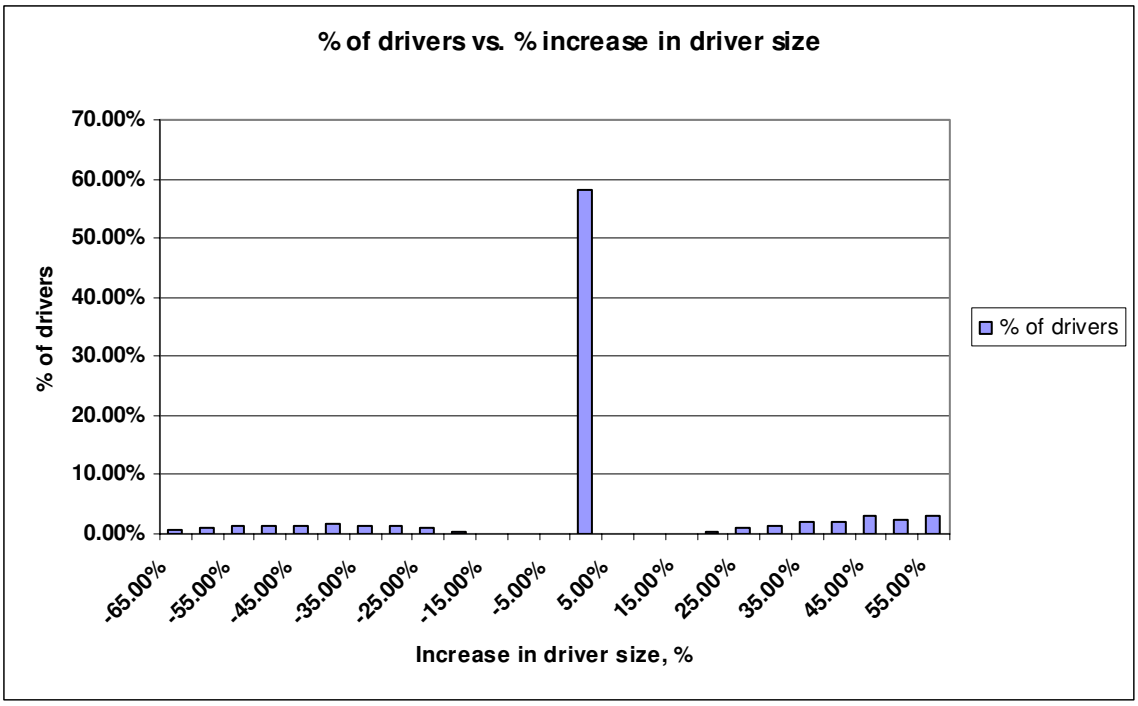


Fig. 13: Driver resizing distribution for recovering delay changes incurred by power minimization. The right tail is driver upsizing while the left tail is driver downsizing.