Interconnect Power and Delay Optimization by Dynamic Programming in Gridded Design Rules

Konstantin Moiseev
Electrical Engineering Dept.
Technion
Haifa 32000, Israel
+972-4-865-1537
mkostya@tx.technion.ac.il

Avinoam Kolodny
Electrical Engineering Dept.
Technion
Haifa 32000, Israel
+972-4-829-4764
kolodny@ee.technion.ac.il

Shmuel Wimer
School of Engineering
Bar-Ilan University
Ramat-Gan 52900, Israel
+972-3-531-7208
wimers@macs.biu.ac.il

ABSTRACT
The lithography used for 32 nanometers and smaller VLSI process technologies restricts the admissible interconnect widths and spaces to a small set of discrete values with some interdependencies, so that traditional interconnect sizing by continuous-variable optimization techniques becomes impossible. We present a dynamic programming (DP) algorithm for simultaneous sizing and spacing of all wires in interconnect bundles (or bus structures), yielding the optimal power-delay tradeoff curve. It sets the width and spacing of all interconnects simultaneously, thus finding the global optimum. The DP algorithm is generic and can handle a variety of power-delay objectives, such as total power or delay, or weighted sum of both, power-delay product, max delay and alike. The algorithm consistently yields more than 10% dynamic power and 5% delay reduction for interconnect channels in industrial microprocessor blocks designed in 32 nanometer process technology, when applied as a post-layout optimization step to redistribute wires within interconnect channels of fixed width, without changing the area of the original layout.

Categories and Subject Descriptors
B.7.2 [**Integrated Circuits**]: Design aids – *layout, placement & routing.*

B.7.1 [**Integrated Circuits**]: Types and Design Styles - *VLSI*

F.2.2 [**Analysis of Algorithms and Problem Complexity**]: Nonnumeric Algorithms and Problems – *Routing and layout*

General Terms
Algorithms, Performance, Design, Theory.

Keywords
Interconnect optimization, interconnect sizing and spacing, power-delay optimization, dynamic programming, gridded design rules

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
ISPD'10, March 14–17, 2010, San Francisco, California, USA.
Copyright 2010 ACM 978-1-60558-920-6/10/03...$10.00.

1. INTRODUCTION AND MOTIVATION
Power and performance of VLSI systems and their tradeoff are important design considerations in state-of-the art technology. The traditional trend towards higher clock rate requires more power, while recent demand for mobile products is driving reduction of power dissipation [1], [2]. Unfortunately, power and speed are often in conflict with each other and their tradeoff is delicate and challenging, offering opportunities for new design methods and algorithms targeting simultaneous power reduction and delay reduction.

Interconnect delay and power consumption due to charging and discharging of wire capacitances are dominant components of total system performance and power [11][13]. The split of delay between devices and interconnects is discussed in [12]. While devices continue to improve in speed, interconnect capacitance keeps growing and determines the dominant portion of the delay [3]. A similar trend happens in power, where interconnects become the dominant contributor of dynamic power consumption [11]. A typical breakdown of dynamic power dissipation in a 65 nanometer high-end microprocessor is illustrated in Fig. 1, indicating that 60% of the consumed dynamic power is due to interconnect capacitance. This portion increases in 32 nanometers and more advanced process technologies.

This paper addresses the problem of power and delay reduction in interconnect channels, also known as wire bundles (bus-like structures of parallel wires, see Figure 2), under discrete-size design rules. Our objective is to vary the wire widths and the inter-wire spaces in the channel while keeping a fixed total width of the structure, to achieve the optimal power-delay tradeoff curve. At each point on this curve we obtain the minimum interconnect power for a given delay, and vice versa. Simultaneous wire sizing and spacing is effective because wire-to-wire capacitances, which are the dominant part of interconnect capacitance [11], are very sensitive to inter-wire spacing.

Several interconnect resizing algorithms were proposed to increase clock frequency [3][4][5][6], to reduce dynamic power [7][8], and to maintain some tradeoff between both [9]. Most of the techniques assume that interconnect width and space can vary in a continuous range allowed by design rules. This assumption was valid until the 65 nanometer process technology generation. Modern manufacturing process technologies restrict the admissible width and space of interconnect to very few discrete values. Moreover, not all width and space combinations are allowed and some interdependencies restrictions are imposed on their choice [1][10]. Design and optimization under such restrictions is a challenge. The first discrete design rules appeared

in the 45 nanometer technology node for low-level metal layers. This trend continued for 32 nanometers technology and will remain so for 22 nanometers and smaller feature sizes [1] where upper metal layers are also subject to discrete-size design rules. Usually, minimizing power and delay in continuous domain is computationally easy since the methods of convex programming can be applied in most cases. However, in the discrete domain continuous optimization techniques cannot be used, and combinatorial optimization is applicable. In the following we show that finding the optimal tradeoff between speed and power in an interconnect channel is an NP-complete problem; but since it possesses the optimal substructure property the problem can be solved by dynamic programming. In this paper we demonstrate a DP algorithm which derives all the feasible power-delay pairs that can be obtained such that neither the power nor the delay may be further decreased without increasing the counterpart. The tradeoff curve is also known as shape-function, which has been discussed by many authors [14][16][17] for the optimization of a single net by sizing its wires and inserting buffers. The main limitation of single-net optimization is its blindness to other adjacent nets, hence ignoring the cross-capacitance between nets, thus yielding sub-optimal results. Moreover, single-net optimization cannot account for the area resource available at the block level. A shape function has also been used by similar DP algorithms for floor planning [17][18]. These DP algorithms work bottom-up [19] due to the tree structure of the problem. A general approach for the solution of such problems by using efficient data structures has been reported in [20].

Simultaneous optimization of all nets in order to achieve minimum delay or power has been addressed by several authors [5][7][9]. Such optimizations account for the block's area constraint and obtain the provable minimum which stems from the convexity of the power and delay expressions. These algorithms assume a continuous range of admissible widths and spaces, which are independent of each other, assumptions holding up to 65 nanometer process technologies.

As compared to previous works in this field, in this paper we present a new combination of several approaches described above. First, we employ a global optimization technique, optimizing many nets simultaneously. Second, our technique optimizes two objective functions – power and delay – simultaneously, which helps to understand tradeoffs between them and derive practical implications. Third, instead of a single optimal solution we generate the optimal power-delay tradeoff curve which reveals the full design space. And finally, to the best of our knowledge, this is the first work which presents a multi-net interconnect optimization technique under discrete design rules.

The allocation of wire widths and spaces from a set of discrete admissible values is an NP-complete problem and naturally mapped into sequential decision making, for which a dynamic programming algorithm is very useful [19]. The development of the algorithm, the proof of its optimality and its implementation for VLSI interconnects are the main contributions of this paper, the rest of which is organized as follows. Section 2 presents interconnect and process technology models with their delay and power equations. Section 3 develops the DP solution and section 4 presents results and experiments obtained for real industrial design in 32 nanometer process technology.

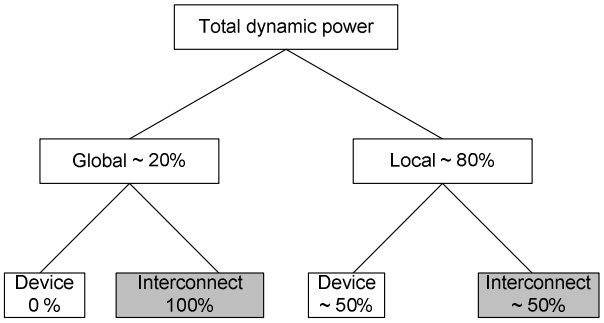

Figure 1: Breakdown of dynamic power into local blocks and global interconnects. As can be seen, the capacitances of global and local wires contribute 60% of the total dynamic power; hence have high potential for power save.

2. DELAY AND POWER MODELING OF INTERCONNECTS IN A BUNDLE

The interconnecting wires at high metal layers typically run in alternating orthogonal directions. Sometimes wires going in the main layer direction are connected by short jogs in the perpendicular direction. Such jogs are rarely used in metal layers higher that metal 1 and they are ignored in the optimization discussion. Connectivity must be maintained under any horizontal shift of vertical wires or vertical shift of horizontal wires. Shifting wires in one layer doesn't affect spacing / width of the orthogonal wires in the layers above it and below it. The lengths of wires in layers above and below the optimized layer usually reach hundreds of microns, while the typical wire shift during the optimization in a given layer is less than a micron. Thus, lengths of 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.

Let $\sigma_1,\ldots,\sigma_n$ be n signals of a wire bundle, and let $I_1,\ldots,I_n$ be their corresponding wires positioned between two shielding wires I_0 and I_{n+1} connected to ground, as shown in Fig 2. Let

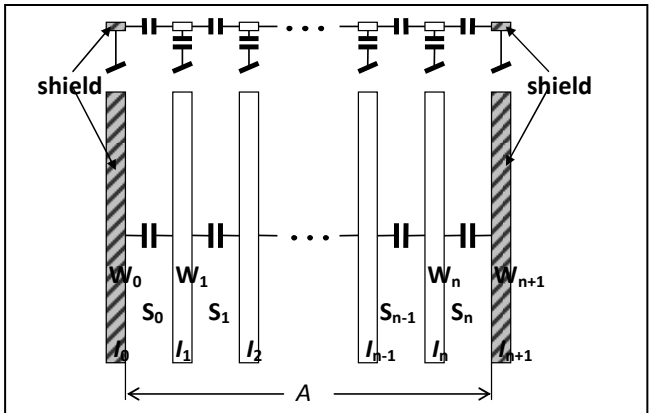

Figure 2: Fundamental cross coupling and ground capacitance model. Wires run in parallel and the entire bundle is shielded on both sides by wires connected to ground

$w_1,\ldots,w_n$ be wire widths and $s_0,\ldots,s_n$ be the spaces between them. It is assumed that admissible wire widths and spaces are taken from finite sets, whose cardinality is usually very small, representing gridded design rules

(2.1) $w_i \in \boldsymbol{W} = \{W_1,\ldots,W_q\}$, and

(2.2) $s_i \in \boldsymbol{S} = \{S_1,\ldots,S_p\}$.

Sometimes, a mix of discrete values with continuous ranges is allowed, but design practice usually employs only a limited set of values, turning the problem into pure discrete. Some technologies may also prohibit certain width and space combinations by imposing interdependencies between the values in (2.1) and (2.2). We shall ignore such restrictions as these do not affect the complexity or optimality of the problems and the proposed solutions. The area allocated for the wire bundle dictates a total width limit A,

(2.3) $\sum_{i=1}^{n} w_i + \sum_{i=0}^{n} s_i \le A$.

The delay of signal σ_i can be approximated by the Elmore model as follows:

(2.4) $$D_i(s_{i-1},w_i,s_i) = \alpha_i + \beta_i w_i + \gamma_i / w_i + \\ + (\delta_i + \varepsilon_i / w_i)(1/s_{i-1} + 1/s_i), \quad 1 \le i \le n$$

The coefficients $\alpha_i, \beta_i, \gamma_i, \delta_i$ and ε_i capture process parameters, driver's resistance and capacitive load, and interconnect length, which is fixed in this setting. The dynamic switching power P_i consumed by σ_i is given by:

(2.5) $P_i(s_{i-1},w_i,s_i) = \kappa_i w_i + \eta_i(1/s_{i-1} + 1/s_i), \quad 1 \le i \le n$.

The coefficients κ_i and η_i capture process parameters, signal activity factors and interconnect length. It is known that cross-capacitance to non-adjacent wires can be about 10% of total wire capacitance [27]. However, taking into account the effect of non-adjacent wires would significantly increase the run-time of the DP algorithm. Thus, only first-order cross-capacitance is taken into account. Generally, Delay and power models in (2.4) and (2.5) are commonly used in the literature[22], and the parameters in their expressions are not subject to optimization.

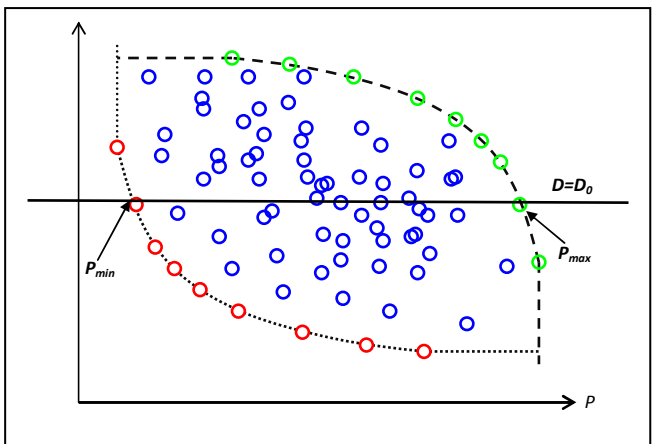

Figure 3: Power-delay design envelope. Circles represent all feasible width and space feasible allocations yielding some power-delay. The red circles are the optimal power-delay results, connected by a dotted curve called shape-function. The green circles are the worst power-delay results, connected by the dashed curve.

The total sum of delays, maximal delay and total interconnects power consumption are given respectively by:

(2.6) $D^{sum}(\bar{s},\bar{w}) = \sum_{i=1}^{n} D_i(s_{i-1},w_i,s_i)$,

(2.7) $D^{\max}(\bar{s},\bar{w}) = \max_{1\le i\le n} D_i(s_{i-1},w_i,s_i)$, and

(2.8) $P(\bar{s},\bar{w}) = \sum_{i=1}^{n} P_i(s_{i-1},w_i,s_i)$.

Once all parameters of the bundle are set, namely, drivers, capacitive loads and activity factors, the optimal sizing problem is equivalent to the following. Let "base" power and delay, called P_1 and D_1, respectively, be calculated for the setting in which all wire widths and spaces are minimum, namely, w_1 and s_1. We then seek an assignment of extra widths and spaces such that the total power (delay) is maximally reduced while total delay (power) change falls in a certain limit.

MIN_DLYPWR:

***Instance*:** A $n-$wire bundle with given drivers, capacitive loads and activity factors, whose wire widths and spaces are given in (2.1) and (2.2).

***Question*:** Is there a setting of the widths and spaces of the wires in bundle such that delay reduction from base delay D_1 is at least δD, while power increase from base power P_1 doesn't exceed δP?

It follows from the delay and power equations given in (2.4) and (2.5), respectively, that both are monotonic decreasing in spacing. Larger wire width always increases power, but may increase or decrease delay, depending on driver's resistance.

Theorem 1: MIN_DLYPWR in NP complete.

Proof sketch: We prove that the MIN_DLYPWR problem is NP complete by showing that any instance of a PARTITION problem, which is NP complete [25] can be transformed in polynomial time into a special instance of MIN_DLYPWR, such that the answer to PARTITION is YES if and only if it is so for the special MIN_DLYPWR instance. The proof follows some ideas used in [26] to prove that the problem of trading off area and delay by cell resizing is NP complete. ■

Being NP-complete, MIN_DLYPWR happens to possess the optimal substructure property. Thus, the problem can be efficiently solved using the dynamic programming approach. This is demonstrated in the next section.

3. DISCRETE WIDTH AND SPACE ALLOCATION IN INTERCONNECT BUNDLE

This section develops the computational model of the DP algorithm for the bundle shown in Fig. 2. We prove that it finds all the optimal power-delay combinations, and analyze its complexity.

3.1 Size allocation as a sequential decision problem

The total width A of the bundle in Fig. 2 is a ***resource*** being ***allocated*** to the space and width alternating sequence $\omega:\left(w_0,s_0,w_1,s_1,...,w_n,s_n\right)$. For the sake of convenience an artificial width $w_0=0$ is introduced, but it doesn't affect the feasibility of the problem and the calculations of power and delay.

Sequence ω needs to satisfy (2.1) and (2.2). It is assumed that a feasible allocation does exist, namely there exists at least one allocation satisfying,

(3.1) $\sum_{i=1}^{n} w_0+\sum_{i=0}^{n} s_i=A$.

For a subsequence $\left(w_0,s_0,w_1,s_1,...w_j,s_j\right)\subset\omega$ we define

(3.2) $D_{0..j}^{sum}=\sum_{i=1}^{j} D_i\left(s_{i-1},w_i,s_i\right)$,

(3.3) $D_{0..j}^{\max}\left(0,j\right)=\max_{1\le i\le j} D_i\left(s_{i-1},w_i,s_i\right)$,

(3.4) $P_{0..j}=\sum_{i=1}^{j} P_i\left(s_{i-1},w_i,s_i\right)$.

Equations (2.6), (2.7) and (2.8) can be calculated incrementally by (3.2), (3.3) and (3.4), respectively, which coincide at $j=n$.

Accumulated sum of delays in (3.2) and max delay in (3.3) are similar in terms of monotony and independence of their past calculation. Replacing the operations $+$ and $\max$ by $\oplus$, we obtain delay and power that get updated step-by-step as follows:

(3.5) $D_{0..j}=D_{0..j-1}\oplus D_j\left(s_{j-1},w_j,s_j\right)$,

(3.6) $P_{0..j}=P_{0..j-1}+P_j\left(s_{j-1},w_j,s_j\right)$.

At $j=n$ the objectives (2.5), (2.6) and (2.7) are completely defined.

The objective functions satisfy the following properties:

Property 1: The functions in (3.5) and (3.6) are monotonic non-decreasing in allocation step j.

Property 2: For any $1\le j\le n-1$,

(3.7) $D_{0..n}=D_{0..j}\oplus D_{j+1..n}$,

(3.8) $P_{0..n}=P_{0..j}+P_{j+1..n}$.

Property 3: After the first j allocations are done, optimization of the rest $n+1-j$ allocations depends only on s_j and $A_{j..n}=A-\left(\sum_{i=0}^{j} w_i+\sum_{i=0}^{j} s_i\right)$ which is available for the rest $n+1-j$ wires, and its optimization is ***independent*** of how the first j allocation decisions have been made.

Let Ω be the set of all possible allocations and define a partial order as follows:

Definition 1 *(dominancy):* Allocation $\omega':\left(w'_0,s'_0,...,w'_j,s'_j\right)\in\Omega$ is ***dominating*** allocation $\omega'':\left(w''_0,s''_0,...,w''_j,s''_j\right)\in\Omega$ if:

1. $A-\left(\sum_{i=0}^{j} s'_i+\sum_{i=0}^{j} w'_i\right)\ge A-\left(\sum_{i=0}^{j} s''_i+\sum_{i=0}^{j} w''_i\right)$,
2. $s'_j\ge s''_j$, and
3. $D_{0..j}\left(\omega'\right)\le D_{0..j}\left(\omega''\right)\wedge P_{0..j}\left(\omega'\right)\le P_{0..j}\left(\omega''\right)$.

It follows that ω'' cannot yield a better solution than ω', and can therefore be safely dropped from any further consideration of optimal solution. Sequence ω'' is called ***redundant***.

It follows that for every pair of $A_{j..n}$ and s_j there is a set of ***non-redundant*** $\left\{\left[P_k\left(A_{j..n},s_j\right),D_k\left(A_{j..n},s_j\right)\right]\right\}_k$ power-delay pairs. Therefore, the triplet $\left\langle A_{j..n},s_j,\left[P\left(A_{j..n},s_j\right),D\left(A_{j..n},s_j\right)\right]\right\rangle$ fully characterizes the first j allocations with their resultant power and delay, and is the only information required to yield the optimal allocation of all n wires. We code such a triplet in a so called ***state*** defined as follows:

Definition 2 *(state):* A triplet $\left\langle A_{j..n},s_j,\left[D\left(A_{j..n},s_j\right),P\left(A_{j..n},s_j\right)\right]\right\rangle$ is called ***state***.

A state is ***feasible*** if $A_{j..n}\ge 0$. It follows by definition that $A_{n..n}=0$ (all area is consumed). A ***stage*** Λ_j is the set of all feasible non-redundant states obtained by all possible size allocations of the first j wires. The states of a stage are totally ordered by lexicographic comparison of their *A*, *s* and *P*. Such order is important for efficient insertion, deletion and redundancy check of states in a stage, allowing access to states in logarithmic time, by using an appropriate data structure. It follows from non-redundancy that ordering by *P* implies reverse order by *D*.

3.2 State augmentation and satisfaction of optimality

Size allocation proceeds from I_j to I_{j+1} as follows. Stage Λ_{j+1} is initially empty. Every state of Λ_j is attempted for augmentation by every possible width and space pair $\left(w,s\right)$ satisfying (2.1) and (2.2). Only feasible augmentations satisfying $A_{j+1..n}=A_{j..n}-\left(w+s\right)\ge 0$ are considered and a new state $\left\langle A_{j+1..n},s,\left[D\left(A_{j+1..n},s\right),P\left(A_{j+1..n},s\right)\right]\right\rangle$ is thus defined.

If no state with the pair $A_{j+1..n}$ and s exists yet in Λ_{j+1} a new state is added to Λ_{j+1}. Otherwise, if it is found to dominate an

already existing state of Λ_{j+1}, the latter is deleted, and a new one is added. If it is found redundant then it is ignored. In this way Λ_{j+1} is built incrementally and maintains only non-redundant states, until all state augmentations of Λ_j are consumed.

Theorem 2 *(optimality):* Stage Λ_n of the DP algorithm contains all the feasible non-redundant, and hence optimal, power-delay pairs that can be obtained by any width and space allocation to *n* wires.

Proof: The proof proceeds in two steps. First we show that Λ_n is non empty. Then we show it must contain all optimal solutions. Assume on the contrary that Λ_n is empty. Let $\omega:\left(w_0,s_0,w_1,s_1,...,w_n,s_n\right)$ be a feasible allocation sequence. Let $\omega':\left(w_0,s_0,w_1,s_1,...,w_j,s_j\right)\subset\omega$ be the longest sub sequence yielding a state $\lambda\in\Lambda_j$ and $\omega'':\left(w_0,s_0,w_1,s_1,...,w_j,s_j,w_{j+1},s_{j+1}\right)\subset\omega$ does not yield a state in Λ_{j+1}. Such ω' must exist since $\left(w_0,s_0\right)\subset\omega$ obviously yields some state in Λ_0. Augment now $\lambda\in\Lambda_j$ by the pair $\left(w_{j+1},s_{j+1}\right)$, which is definitely feasible since $\sum_{i=0}^{j+1}w_i+s_i\le A$ by assumption. This yields a state in Λ_{j+1}, a contradiction to $\omega'\subset\omega$ being the longest subsequence having a corresponding state.

Having proven that $\Lambda_n\neq\phi$, we'll show similarly that any feasible non-redundant power-delay pair of a complete feasible allocation is obtained by some state in Λ_n. Assume on the contrary that $\left[P^*,D^*\right]$ is non-redundant power-delay obtained by $\omega^*:\left(w_0^*,s_0^*,w_1^*,s_1^*,...,w_n^*,s_n^*\right)$, but doesn't yield a state in Λ_n. Let $\omega^{*'}:\left(w_0^*,s_0^*,w_1^*,s_1^*,...,w_j^*,s_j^*\right)\subset\omega^*$ be the longest sub sequence yielding a state in Λ_j, while the subsequence $\omega^{*''}:\left(s_0^*,w_1^*,s_1^*,w_2^*,...,w_j^*,s_j^*,w_{j+1}^*,s_{j+1}^*\right)$ does not yield a state in Λ_{j+1}. Augmentation by $\left(w_{j+1}^*,s_{j+1}^*\right)$ results in the same contradiction as before.■

Knowing that the DP algorithm yields all power-delay non-redundant pairs, they define the power-delay envelope of the bundle. One can plot the power-delay curve as shown in Fig. 3. This curve is (by definition of dominancy) monotonic increasing in one parameter and monotonic decreasing in the other. The curve divides the first quadrant of the power-delay plane into an upper-right region where all feasible power-delay solution exist and a lower-left region where no feasible solutions exists. This envelope has the same nature of the well known shape-function in bottom-up buffer insertion and wire resizing algorithms [14][15][16].

We showed how a DP algorithm finds the power-delay Pareto curve [24] representing optimal design. Some papers proposed to minimize a weighted sum of the power and delay [9] or minimize a product of their powers [23]. It is a straightforward consequence that any two-variable function that is monotonic increasing in any of its variables will achieve its minimum at a point of the power-delay shape-function. For example, functions such as $f=\alpha P+\beta D$ and $f=P^\alpha D^\beta$, where $\alpha>0$ and $\beta>0$.

Theorem 3 (without proof): A power-delay function $f\left(P,D\right)$ which is monotonic increasing in P and D achieves its minimum on the boundary of the power-delay feasible region.

3.3 Time and memory bounds of the DP algorithm

Let $P_{\max}$ ($D_{\max}$) be the maximal power (delay) incurred by a wire. We define a power (delay) resolution as $\varepsilon P_{\max}$ ($\varepsilon D_{\max}$), where $\varepsilon<<1$ is an arbitrarily small accuracy parameter, and snap every calculated power (delay) to the nearest integral multiplication of this resolution. Then the following theorem defines time and storage bounds of the DP algorithm:

Theorem 4 *(time and storage bounds):* Given n-signal wire bundle and process technology having p admissible widths and q admissible spaces, the time complexity of the DP algorithm to find width and space allocation yielding the optimal power-delay curve in accuracy ε is bounded by $O\left(pq^2n^3\log n/\varepsilon\right)$. The storage is bounded by $O\left(qn^3/\varepsilon\right)$.

Proof sketch: The number of states at each stage is $O\left(n\right)\times q\times O\left(n/\varepsilon\right)=O\left(qn^2/\varepsilon\right)$ (number of distinct values of $A_{j..n}$ multiplied by number of admissible spaces multiplied by number of distinct non-redundant power-delay pairs). Each state is attempted $p\times q$ times for augmentation, where an augmentation consumes $O(\log n)$ time, since all the states of a stage are kept in an ordered balanced tree. The total number of stages is n (equals to number of signals in a bundle). ■

4. IMPLEMENTATION AND EXPERIMENTAL RESULTS

The DP algorithm described in Section 3 was coded in C++ under OpenAccess environment. It was applied to industrial layout blocks from a commercial full-custom processor design in 32nm process technology, which were generated by standard tools. Our algorithm was employed in an attempt to reduce the delay and dynamic power by resizing of interconnects and their spacing. The wire bundles were defined by power rails and clock wires, which were not allowed to move, such that the optimization preserved the original size of the whole layout. The power and delay values were calculated using the model presented in section 2.

Table 1. Optimization results for wire bundles derived from industrial layout.

Bundle	Number of wires	Bundle width, X	Bundle length, X	Metal layer	Precision	Run time, s	Number of states	Average diff. from the continuous solution, %
1	18	90	2115	3	LOW	0.87	25	27.75
					MEDIUM	6.56	89	5.24
					HIGH	359.03	202	1.15
					Continuous solution	3182.22	373	0
					- (INITIAL)	-	1	25.5
2	10	40	4007	2	LOW	0.21	16	58.69
					MEDIUM	0.91	23	6.62
					HIGH	29.23	97	1.36
					Continuous solution	2405.26	1006	0
					- (INITIAL)	-	1	88.5
3	14	38	1752	3	LOW	0.27	9	84.61
					MEDIUM	2.02	56	37.23
					HIGH	17.63	95	5.41
					Continuous solution	2242.49	277	0
					- (INITIAL)	-	1	58.7
4	12	40	3298	4	LOW	0.14	3	34.30
					MEDIUM	0.95	7	10.11
					HIGH	37.28	50	3.86
					Continuous solution	934.5	170	0
					- (INITIAL)	-	1	28.4

In the following discussion, we denote the minimum allowed wire width and minimum allowed inter-wire space by X: $W_{\min} = S_{\min} = X$. The maximum values of widths and spaces in the given technology is $3X$.

We first experimented with four different bundles consisting of 10-18 wires. Note that in typical industrial designs the layout is separated by power grid wires into stripes, each of which includes only a small number of wires (up to 20). Therefore, consideration of cases with a larger number of wires is unnecessary. Before applying the DP algorithm, the continuous optimization was performed where wire widths and inter-wire spaces were allowed to take any value between X and $3X$. It was done as follows: first, continuous minimization of the average wire delay was performed for the given bundle, the minimum delay $D_{\min}$ was determined and the power P was recorded. Then, continuous minimization of the total bundle power was performed and the total bundle power $P_{\min}$ was determined and the average wire delay D was recorded. The power and delay values in all the following experiments were normalized by dividing them by $D_{\min}$ and $P_{\min}$.

After continuous optimization, for each bundle we applied the DP algorithm three times for gridded design rules, with different sets of allowed wire widths and inter-wire spaces. The first set included just the minimum and maximum allowed values of spaces and widths, that is X and $3X$. The second set allowed three different values of widths and spaces: X, $2X$, $3X$. The third set contained the complete range of possible space and width values allowed in the given technology: X, $1.5X$, $2X$, $2.5X$, $3X$. For each run we generated the full set of Pareto optimal power-delay pairs and plotted them in the power-delay plane. The plots of two of them are shown at Fig. 4. The corresponding runs are denoted as "low", "medium" and "high" precision accordingly. On the same plane we showed also the initial layout state as it appears in the original layout, and the curve obtained by continuous optimization. The continuous optimization curve was obtained by uniformly choosing n delay points from the range $[D_{\min}; D]$ (n was chosen arbitrarily, but large enough to see the difference between the continuous case and the high precision curve). For each delay point, continuous minimization of power under delay constraint was performed, and the obtained power-delay pair was plotted. The experiment is summarized in Table 1. For the sake of consistency, the continuous minimization points are called "states" similar to real states in discrete optimizations. It can be seen that in all cases the original layout, created by commercial tools, is far from the optimal power-delay curve. This indicates that usually the main goal during the routing process is successful routing completion, and there is typically an opportunity to improve both power consumption and performance of the interconnections. The plots also demonstrate that using just two extreme values of widths and spaces or just three values of X, $2X$, $3X$ are insufficient for power-delay optimization. On the other hand, using 5 values of X, $1.5X$, $2X$, $2.5X$, $3X$ results in just about 1-5% difference in power and delay, as compared to continuous optimization.

It is interesting to explore where the original commercial routing tool falls in the entire feasible power-delay region as shown in Fig. 3. This is important since in some sense it evaluates the potential to improve standard routers by algorithms such as the one developed in this paper. The entire power-delay design space can be easily explored by reversing the dominancy in Definition

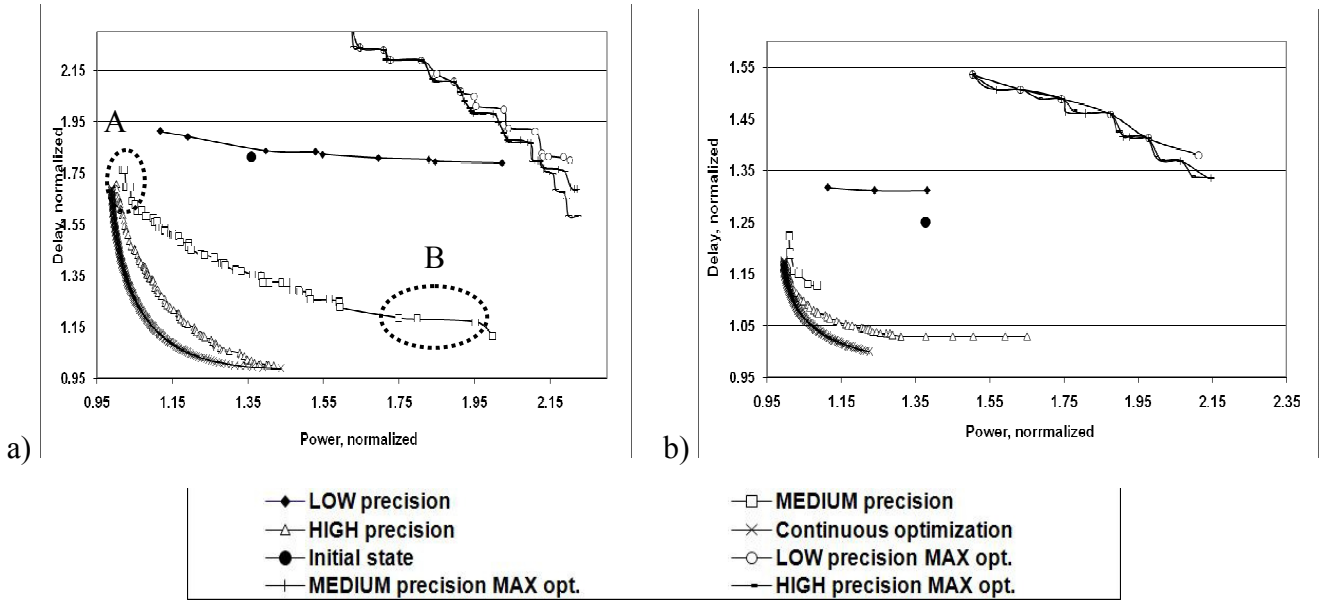

Figure 4: Power-delay curves for different bundles with entire power-delay envelope. Power and delay are normalized. a) 14 wires, metal 3; b) 12 wires, metal 4

1. The inequalities in 2 and 3 are reversed, so maximum is obtained instead of minimum. This way, the DP algorithm is modified to maximize power and delay of the wire bundle.

Maximum Pareto curves were generated for all bundles simulated in the previous example and are presented in Fig. 4. It can be seen that power and delay can vary by a factor of 1 to 4 from the corresponding minimum values.

In the vicinity of D_{min} and P_{min}, the sensitivity to one of the optimization objectives is high, while the sensitivity to the second is low. This means that there are layout configurations which differ in one of the objective values and are almost the same in the second. This characteristic has important design implications. For example, let's look at two areas emphasized in Fig. 4a). In area A there are two solutions with very high delay sensitivity, while in area B the situation is the opposite: there are two solutions with very high power sensitivity. Thus, tuning design to one of the corners is quite inefficient: slight improvement in one of the objectives causes a great loss in the other. From the design point of view, the best solution should be located near the middle of the power-delay curve (as close as possible to the origin). On the other hand, if the design had been tuned by some reason to one of the extreme areas, then there is a great opportunity for optimization: a major improvement of one of the objectives can be achieved by a slight increase of the other. Such improvement can usually be obtained by minor changes of wire width or space allocation in the layout.

All the results reported above were obtained using a fixed power grid. We also ran experiments without the limitation on the power grid, allowing more freedom in spacing optimization. We chose a typical bundle consisting of 16 data wires and 4 power grid wires distributed uniformly among data wires and ran the DP algorithm with high precision. The analysis of resulting power-delay curves for this setting shows an additional improvement of about 14 % as compared with fixed power grid (Fig. 5). However, shifting the power grid can be too disruptive for a conservative design methodology. Hence, we discuss fixed power grid results only.

Finally, we present the results obtained on real design blocks. As a follow-up of our research we heuristically extended the algorithm presented in this paper to arbitrary layout, which enabled us to apply it to complete design blocks. It was experimented on industrial random logic control blocks used in a full-custom processor design in 32nm process technology with placement and routing performed by a commonly used commercial vendor tool. Our algorithm was employed in an attempt to further reduce the delay and dynamic power by resizing

Table 2. Results of power-delay minimization in real blocks

	Block1		Block2	
size (microns)	69 x 68		101 x 150	
layers	metal 2	metal4	metal 2	metal4
Initial power	349.1	240.6	622	886.3
P_{min}	333	222.1	598	802.9
P_{max}	527	347.1	956.3	1238
power reduction (%)	**14.83**	**22.14**	**13.52**	**23.92**
Initial delay	5201	3880	8094	10635
D_{min}	5040	3633	7903	9802
D_{max}	6491	4614	10538	13159
delay reduction (%)	**6.33**	**8.76**	**6.94**	**11.65**

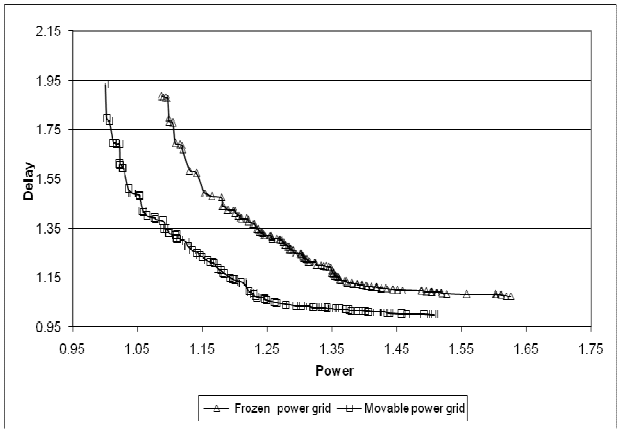

Figure 5: Optimization with fixed and movable power grid. An improvement of about 14% of power and delay in average is obtained when power grid is allowed to move.

interconnects and their spacing. Signals such as power rails and clocks were not touched and their position remained unchanged.

Table 2 presents the simultaneous power and delay results obtained for three typical blocks. These blocks use metal2, metal3 and metal4 for interconnections. The results are shown in relative units. As shown in the table, a significant simultaneous reduction of power and delay has been achieved. This is explained by the fact that commercial tools, though guiding the place and route for power-delay optimization by controlling the position of cells and specifying width and space for critical signals, do not perform global sizing optimization, which our algorithm does.

5. CONCLUSION AND FURTHER RESEARCH

In this paper we presented and solved the novel problem of simultaneous power-delay optimization of a bundle of signals with gridded (discrete value) design rules. We developed an efficient DP algorithm which solves the problem exactly. As a result, the power-delay Pareto curve is obtained which can be used by the designer to assess goodness of the current design state and derive important design implications. We showed that 5 values of available wire widths and spaces are enough to get to as close as 5% from the exact continuous solution, and that using just two or three values of widths and spaces is insufficient.

Finally, a variation of the algorithm developed in this paper has been deployed for optimizing layouts of complete functional blocks which use lower level metal layers subject to discrete value design rules. The application of the algorithm on real design blocks showed a reduction of 22% in interconnect power and 9% in interconnect delay on average. As process technology will progress to 22 nanometer feature size, more layers will turn to discrete rules, so application of the DP algorithm can cover full-chip routing as well, and further power-delay reduction would be achievable.

6. REFERENCES

[1] International Technology Roadmap for Semiconductors, *2007, available online, http://www.itrs.net/reports.html*

[2] S. Borkar, "Low power design challenges for the decade," *Proc. of the 2001 Conf. on Asia South Pacific design automation*, pp. 293-29

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] Arunachalam, R., Acar, E., and Nassif, S. R. 2003, 'Optimal shielding/spacing metrics for low power design", In *Proceedings of IEEE Computer Society Annual Symposium on VLSI*, 167-172.

[9] K. Moiseev, S. Wimer, A. Kolodny, "Power-Delay Optimization in VLSI Microprocessors by Wire Spacing", in *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, Volume 14, Issue 4, August 2009

[10] C. Webb, "45nm Design for Manufacturing.", *Intel Technology Journal*, 2008

[11] 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

[12] J. Cong, L. He, K.-Y. Khoo, C.-K. Koh, and D. Z. Pan, "Interconnect design for deep submicron ICs," in *Proc. Int. Conf. Computer-Aided Design*, Nov. 1997, pp. 478–485.

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

[14] W. Shi and Z. Li, "An $O(n \log n)$ time algorithm for optimal buffer insertion", *Proceedings of Design Automation Conference*, 2003, pp. 580-585

[15] J. Cong, C.-K. Koh, and K.-S. Leung, "Simultaneous buffer and wire sizing for performance and power optimization", *Proceeding on International Symposium on Low Power Electronics and Design*, 1996, pp. 271-276

[16] L. P. P. P. van Ginneken, "Buffer placement in distributed RC-tree networks for minimal Elmore delay", *Proceedings of International Symposium of Circuits and Systems*, 1990, pp. 865-868

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

[18] K. Chaudhary and M. Pedram, "A near optimal algorithm for technology mapping minimizing area under delay constraints", *Proceeding of Design Automation Conference*, July 1992, pp. 492-498

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

[20] R. Chen and H. Zhou, "An Efficient Data Structure for Maxplus Merge in Dynamic Programming", *IEEE Transaction on Computer-Aided Design of Integrated Circuits and Systems*, 2005, pp. 3004 – 3009.

[21] A. I. Abou-Seido, B. Nowak, and C. Chu, " Fitted Elmore Delay: A simple and Accurate Interconnect Delay Model", In *Proceedings of IEEE International Conference on Computer Design*, 2002, pp. 422-427.

[22] C.-K. Cheng, J. Lillis, S. Lin and N.H. Chang, *Interconnect Analysis and Synthesis*, John Wiley Press, 2000

[23] D. A. Hodges, H. G. Jackson and R. A. Saleh, Analysis and Design of Digital Integrated Circuits, Mc. Graw Hill, 3rd edition, 2004

[24] S. Boyd and L. Vandenberghe, Convex Optimization, Cambridge University Press, 2004.

[25] M. R. Garey and D. S. Johnson, Computers and Intractability, Freeman, 1979.

[26] W-N. Li, A. Lim, P. Agrawal and S. Sahani, "On circuit implementation problem", *IEEE Transaction on Computer-Aided Design of Integrated Circuits and Systems*, 1993, pp. 1147 – 1156.

[27] K. Sundaresan and N. R. Mahapatra, "Accurate Energy Dissipation and Thermal Modeling for Nanometer-Scale Buses", in *Proceeding of 11 International Symposium on High-Performance Computer Architecture*, 2005, pp. 51-60