

Dynamic Programming Algorithm for Interconnect Channel Sizing in Discrete Design Rules

Abstract

The lithography used for 32 nanometers and smaller VLSI process technologies restricts the admissible interconnect widths and spaces to very few discrete values with some interdependencies, making traditional by continuous-variable optimization interconnect sizing techniques impossible. Single-net bottom-up power-delay optimization for discrete wire widths has been solved and got a lot of attention in literature. This article presents a dynamic programming (DP) algorithm for interconnect width and space allocation yielding the optimal power-delay tradeoff function. 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 real data blocks designed in 32 nanometer process technology.

1. Introduction and Motivation

Power and delay of VLSI systems and their tradeoff is important design consideration of microprocessors and other products. The traditional pace of higher clock rate consumes more power, while recent demand for cool products is driving power reduction [1], [2]. Unfortunately power and delay conflict each other and their tradeoff is delicate and challenging, offering opportunities for new design methods and algorithms for simultaneous power reduction and clock speedup.

Few interconnect resizing algorithms were proposed to speedup clock cycle, [3][4][5][6] to reduce dynamic power [7][8][9], and to maintain some tradeoff of both [10]. Most of the techniques assume that interconnect width and space can vary in a continuous range called *design rules*. This assumption holds up to 65 nanometers process technologies. First discrete design rules appeared in 45 nanometer node for lower metal layers up to third metal. This trend continued for 32 nanometers technology and will remain so for 22 nanometers and smaller feature size [1]. There, more discrete geometric rules have shown up, and upper metal layers are also subject to discrete design rules.

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][11]. Design and optimization under such restrictions is a challenge for both circuit designers and CAD.

Interconnect delay and power consumption due to charging and discharging of wire capacitances are dominant components [12][14]. The split of delay between devices and interconnects is discussed in [13]. While devices continue to speedup, interconnects cross-coupling capacitance was growing and turns to be the dominant portion of the delay [15]. Similar trend happens in power, where interconnects become the dominant contributor of dynamic power consumption [12]. A typical breakdown of dynamic power

dissipation in 65 nanometer high-end microprocessor is illustrated in Fig. 1, indicating that half of dynamic power consumed by functional blocks is due to cross-coupling capacitance between adjacent wires of same layer. This portion is increasing in 32 nanometers and smaller process technologies.

This paper finds the optimal power-delay tradeoff function, obtaining the minimum power for a given delay and vice versa. Finding this tradeoff is accomplished by a DP algorithm which derives all the feasible power-delay pairs that can be obtained such that neither the power nor the delay can be further decreased without increasing the counterpart. This is called also shape-function, which has been discussed by many papers [16][18][19] 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 incurring between nets, thus yielding sub-optimal results. Moreover, single-net optimization cannot account the area resource available at block level, which may lead to non feasible sizing. Shape function by similar DP algorithms for floor planning has also been widely used [19][20]. These DP algorithms are working bottom-up [21] due to the tree structure of the problem. A general approach for the solution of such problems by using efficient data structure has been lately reported in [22].

Simultaneous optimization of all nets in order to achieve minimum delay or power has been addressed by few papers [5][7][8][10]. Such optimizations account for the block's area constrained 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 65nanometer process technologies.

The width and space allocation are naturally mapped into sequential decision making for which dynamic programming algorithm is very useful [21]. The development of such algorithm, the proof of its optimality and its implementation for real VLSI interconnects is the essence of this paper, the rest of which is organized as follows. Section 2 presents interconnects and process technology models with their delay and power equations. Section 3 develops the DP solution and section 4 presents results obtained for real industrial design of 32 nanometers process technology.

2. Delay and power of interconnects in a homogeneous bus

Let $\sigma_1,...,\sigma_n$ be *n* signals of a homogeneous bus, and let $I_1,...,I_n$ be their corresponding wire segments positioned between two shielding wires I_0 and I_{n+1} connected to ground, as shown in Fig 2. Let $s_0,...,s_n$ be the spaces between the wires and $W_1,...,W_n$ be their widths. It is assumed that wire admissible widths and spaces are taken from the following sets:

$$(2.1) s_i \in S = \{S_1, ..., S_p\}, \text{ and }$$

$$(2.2) w_i \in W = \{W_1, ..., W_q\}.$$

p and q usually do not exceed 5. Sometimes, a mix of discrete values with continuous ranges is allowed, but design practice is usually using only 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'll

ignore such restrictions as those doesn't affect the optimality of the algorithm, but only narrows the solution space.

The delay of signal σ_i can be approximated by Elmore model [23] as follows:

$$(2.3) D_i(s_{i-1}, w_i, s_i) = \alpha_i + \beta_i w_i + \gamma_i / w_i + (\delta_i + \varepsilon_i / w_i) (1/s_{i-1} + 1/s_i), \quad 1 \le i \le n$$

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:

$$(2.4) P_i(s_{i-1}, w_i, s_i) = \kappa_i w_i + \eta_i (1/s_{i-1} + 1/s_i), \quad 1 \le i \le n$$

The coefficients κ_i and η_i capture process parameters, signal activity and interconnect length. The parameters of the delay and power expressions are fix and not subject to optimization. Delay and power models in (2.3) and (2.4) are commonly used in the literature [24].

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

$$(2.5) D^{sum} \left(\overline{s}, \overline{w}\right) = \sum_{i=1}^{n} D_i \left(s_{i-1}, w_i, s_i\right), \text{ and}$$

$$(2.6) D^{\max} \left(\overline{s}, \overline{w}\right) = \max_{1 \le i \le n} D_i \left(s_{i-1}, w_i, s_i\right).$$

$$(2.7) P\left(\overline{s}, \overline{w}\right) = \sum_{i=1}^{n} P_i \left(s_{i-1}, w_i, s_i\right).$$

An important property of the objectives in (2.5), (2.6) and (2.7) that makes DP algorithm viable is their being monotonic non-decreasing. The simultaneous power-delay optimization of (2.5) and (2.7) or (2.6) and (2.7)

is guaranteed to yield the power-delay shape function as shown in the subsequent.

3 Width and space allocation in homogeneous interconnect bus

This section develops the computational model of the DP algorithm for the homogeneous interconnect bus shown in Fig. 2. We'll prove that it finds all the optimal power-delay combinations and then analyze its complexity.

3.1 Size allocation as a sequential decision problem

The total width A of the bus in Fig. 2 is a *resource* being *allocated* to the space and width alternating sequence $\omega: (w_0, s_0, w_1, s_1, ..., w_n, s_n)$. 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 $(w_0, s_0, w_1, s_1, \dots, w_j, s_j) \subset \omega$ we define
- $(3.2) D_{0..j}^{sum} = \sum_{i=1}^{j} D_i \left(s_{i-1}, w_i, s_i \right),$
- $(3.3) D_{0...j}^{\max}(0, j) = \max_{1 \le i \le j} D_i(s_{i-1}, w_i, s_i),$
- $(3.4) P_{0..j} = \sum_{i=1}^{j} P_i(s_{i-1}, w_i, s_i).$

Equations (2.5), (2.6) and (2.7) can be calculated incrementally by (3.2), (3.3) and (3.4), respectively, which at j = n they are coinciding.

Cumulated sum of delay 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 are getting 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.

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 $1 \le j \le n-1$ there exists:

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

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

Property 3: After 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 $\omega': (w'_0, s'_0, ..., w'_j, s'_j) \in \Omega$ is **dominating** allocation $\omega'': (w''_0, s''_0, ..., w''_j, s''_j) \in \Omega$ 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. $D_{0..j}(\omega') \leq D_{0..j}(\omega'') \wedge P_{0..j}(\omega') \leq P_{0..j}(\omega'')$.

It follows that ω'' cannot yield better solution than ω'' does 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* $\left\{\left[P_k(A_{j,n},s_j), D_k(A_{j,n},s_j)\right]\right\}_k$ power-delay pairs. Therefore, the triple $\left\langle A_{j,n},s_j, \left[P(A_{j,n},s_j), D(A_{j,n},s_j)\right]\right\rangle$ fully characterizes the first *j* allocations with their outcoming power and delay, and is the only information required to yield the optimal allocation of all *n* wires. We code such triple in a so called *state* defined as follows:

Definition 2 (state): A triple $\langle A_{j.n}, s_j, [D(A_{j.n}, s_j), P(A_{j.n}, s_j)] \rangle$ 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 to access states in logarithmic time with 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 (w,s) satisfying (2.1) and (2.2). Only feasible augmentations satisfying $A_{j+1..n} = A_{j..n} - (w+s) \ge 0$ are considered and a new state $\langle A_{j+1..n}, s, [D(A_{j+1..n}, s), P(A_{j+1..n}, s)] \rangle$ 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, while 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 augmentation of Λ_j are consumed. Fig. 3 illustrates the progression from Λ_j to Λ_{j+1} .

Theorem 1 (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'll show that Λ_n is non empty. Then we'll show it must contain all optimal solutions. Assume in contrary that Λ_n is empty. Let $\omega: (w_0, s_0, w_1, s_1, ..., w_n, s_n)$ be a feasible allocation sequence. Let $\omega': (w_0, s_0, w_1, s_1, ..., w_j, s_j) \subset \omega$ be the longest sub

sequence yielding a state $\lambda \in \Lambda_j$ and $\omega'': (w_0, s_0, w_1, s_1, ..., w_j, s_j, w_{j+1}, s_{j+1}) \subset \omega$ does not yield a state in Λ_{j+1} . Such ω' must exist since $(w_0, s_0) \subset \omega$ obviously yields some state in Λ_0 . Augment now $\lambda \in \Lambda_j$ by the pair (w_{j+1}, s_{j+1}) , which is definitely feasible since $\sum_{i=0}^{j+1} w_i + s_i \leq A$ 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 proved that $\Lambda_n \neq \phi$, we'll show similarly that any feasible nonredundant power-delay pair of a complete feasible allocation is obtained by some state in Λ_n . Assume in contrary that $[P^*, D^*]$ is non-redundant powerdelay obtained by $\omega^* : (w_0^*, s_0^*, w_1^*, s_1^*, ..., w_n^*, s_n^*)$ but doesn't yield a state in Λ_n . Let $\omega^{*'} : (w_0^*, s_0^*, w_1^*, s_1^*, ..., w_j^*, s_j^*) \subset \omega^*$ be the longest sub sequence yielding a state in Λ_j , while the subsequence $\omega^{*''} : (s_0^*, w_1^*, s_1^*, w_2^*, ..., w_j^*, s_j^*, w_{j+1}^*, s_{j+1}^*)$ does not yield a state in Λ_{j+1} . Augmentation by (w_{j+1}^*, s_{j+1}^*) results in same contradiction as before.

Knowing that DP algorithm yields all power-delay non-redundant pairs, they define the power-delay envelop of the bundle. One can plot the power-delay curve as shown in Fig. 4. This curve is by the very definition of dominancy monotonic increasing in one parameter and monotonic decreasing in the other one. The curve divides the first quadrant of the power-delay plane into an upper-right region where all feasible power-delay solution exist and into lower-left portion where no feasible solution exists. Such envelop has the

same nature of the well known shape-function being dealt by the bottom-up buffer insertion and wire resizing algorithms [16][17][18].

The function plotted in Fig. 4 provides all the data needed to allocate the width and space to the wires of the bus. In order to enable sizing retrieval, the DP algorithm stores all the feasible non-redundant states at any stage Λ_i , $1 \le i \le n$. In addition, any new state at stage Λ_{j+1} points back to the state in Λ_j from which it was augmented. The back pointer stores the width and space that have been allocated at the augmentation. We can decide on the best power-delay tradeoff among all pairs of Λ_n and then retrieve the entire widths and spaces by backward traversals. Starting from (w_n, s_n) that correspond to the desired power-delay solution, we obtain with the aid of the above pointers all (w_i, s_i) , $n > i \ge 1$, until (w_0, s_0) is reached.

3.3 Generalization of power-delay optimization

We showed how a DP algorithm finds the power-delay Pareto curve representing optimal design. Some papers proposed to minimize a weighted sum of the power and delay [10] or minimize a product of their powers [25]. It is a straight forward 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. Functions as $f = \alpha P + \beta D$ and $f = P^{\alpha}D^{\beta}$, where $\alpha > 0$ and $\beta > 0$, are examples. s **Theorem 2:** A power-delay function f(P,D) which is monotonic increasing in *P* and *D* achieves its minimum on the boundary of the power-delay feasible region.

Proof: In contrary, if it was not the case then f(P,D) would have been minimized at some internal point of the power-delay feasible region, corresponding to redundant power-delay pair, say(P',D'). Therefore there exists a power-delay point (P'',D'') achievable by the DP algorithm satisfying P'' < P' and D'' < D'. It follows by elementary calculus that f(P,D) is monotonic decreasing along the closed interval connecting the internal point (P'',D'') with the point (P'',D''). Therefore, f(P'',D'') < f(P',D'), hence a contradiction. ■

3.4 Time and memory bounds of DP algorithm

The calculation of any of the objective functions in (3.5) and (3.6) requires O(1) time. It follows from (2.1) and (2.2) that a state is attempted for augmentation by $p \times q$ width-space combinations. The number of states at a stage is factored by the number of distinct $A_{j..n} = A - \left(\sum_{i=0}^{j} w_i + \sum_{i=0}^{j} s_i\right)$ values. Let *gcd* denote the greatest common divisor of the values in $W \cup S$ given in (2.1) and (2.2), so A/gcd bounds the number of distinct values of A(j,n). In modern VLSI processes *gcd* is typically half of minimum feature size. The area *A* of a homogeneous bus containing *n* wires is proportional to *n*, hence the number of distinct values of $A_{j..n}$ is O(n).

Let us consider the number of distinct non-redundant (essential) power-delay pairs $\left[D(A_{j,n},s), P(A_{j,n},s)\right]$ resulted for certain $A_{j,n}$ and s. Power and delay of a wire depend on two factors. The first involves process technology, drivers and receivers parameters, and wire length. These are not subject to change by optimization. The other involves wire width which has p values and left and right spaces which has q values each. Consequently, a wire can contribute to the total sum $O(p \times q^2)$ distinct power and delay values. The objectives functions (3.5) and (3.6) are non-decreasing, and may increase at any augmentation, so we wish to know how many distinct power-delay values can result in. If the power and delay values of a wire were arbitrary real number, the number of distinct power-delay values can be bounded by n/ε as described below, where $\varepsilon \ll 1$ is arbitrarily small accuracy parameter.

Let P_{\max} (D_{\max}) be the maximal power (delay) resulted by a wire. We define a power (delay) resolution as εP_{\max} (εD_{\max}), and snap every calculated power (delay) to the nearest integral multiplication of this resolution. The addition of power (delay) values will be then closed by definition, taking values from the set $\{k\varepsilon P_{\max} | 1 \le k \le n/\varepsilon\}$ ($\{k\varepsilon D_{\max} | 1 \le k \le n/\varepsilon\}$), whose size is n/ε . It follows that there are at most n/ε distinct non-redundant power-delay pairs since the very definition of non-redundancy implies reciprocal monotony of power and delay. Given $[P_1, D_1]$ and $[P_2, D_2]$ two non-redundant power-delay pairs, there exists $P_1 > P_2 \Leftrightarrow D_1 < D_2$. Power and delay accuracy can be controlled by setting ε to any desired small value. Notice also that the error occurring at a wire is randomly negative or positive due to snapping to nearest quantized value, so cumulative error stays very close to zero in practice.

A new state is checked for redundancy. We store all the states of a stage in an ordered balanced tree. With the above counting arguments, a stage has a total of $O(n) \times q \times O(n/\varepsilon) = O(qn^2/\varepsilon)$ distinct states $\langle A(j,n), s, [D(A(j,n),s), P(A(j,n),s)] \rangle$. The insertion of a new state with its redundancy test consumes therefore $O(\log n)$ time. In conclusion there are *n* stages, each has $O(qn^2/\varepsilon)$ states and each state is attempted $p \times q$ times for augmentation, where an augmentation consumes $O(\log n)$ time. We summarize in the following:

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

4 Implementation and experimental results

The DP algorithm described in Section 3 was coded in C++ under OpenAccess environment. It was experimented on industrial blocks used in full-custom processor design in 32nm process technology. Our algorithm was employed in an attempt to reduce the delay and dynamic power by resizing of interconnects and their spacing. Signals such as power rails and clocks were not touched and their position remained unchanged. We then tweaked the DP algorithm in order to maximize power and delay in order to explore where the commercial routing tool in use is falling in the power-delay envelop obtained by all wire sizing and spacing allocations as shown in Fig. 4. The exploration of maximum power-delay provides some idea about the entire power-delay envelop that can be achieved by resizing and re-spacing of interconnecting wires. Such data is important since in some sense it tells about the quality of the standard routing tool and the potential to improve it by algorithms as developed in this paper.

Power-delay envelop exploration can be easily done by reversing the dominancy in Definition 1. The inequalities in 2 and 3 are reversed, so maximum is obtained instead of minimum.

5 Conclusion and further research

The algorithm developed in this paper has been deployed for the benefit of functional blocks which use lower level metal layers, which in 32 nanometer and smaller feature size are obeying only discrete value design rules. As process technology will progress to 22 nanometer feature size, few upper level layers will turn to discrete rules as well, so the application of DP algorithm can cover full-chip routing as well and further power-delay reduction can be achieved.

This paper dealt with the blocks where the interconnecting wires aimed at optimization belong to one level of hierarchy, namely, data for optimization is flat. In custom design however, some of the blocks such those of data-path and register-file are bit oriented and may contain many level of hierarchy. DP paradigm well suits flat routing, and its enhancement for hierarchical routing is of high benefit and left as an open question whether it is possible.

References

- [1] International Technology Roadmap for Semiconductors, 2007, available online, <u>http://www.itrs.net/reports.html</u>
- [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 postlayout 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] P. Zuber, O.Bahlous, T. Ilnesher, M. Ritter, and W. Stechele, "Wire Topology Optimization for Low Power CMOS", *IEEE Transactions on VLSI*, 2009, pp. 1-11
- [9] 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.

- [10] K. Moiseev, S. Wimer, A. Kolodny, "Power-Delay Optimization in VLSI Microprocessors by Wire Spacing", in press
- [11] C. Webb, "45nm Design for Manufacturing.", *Intel Technology Journal*, 2008
- [12] 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
- [13] 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.
- [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] Cong, J., He, L., Koh, C. K., and Pan, Z. 2001. Interconnect Sizing and Spacing with Consideration of Coupling Capacitance. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 20, no. 9, 1164-1169.
- [16] W. Shi and Z. Li, "An O(nlogn) time algorithm for optimal buffer insertion", *Proceedings of Design Automation Conference*, 2003, pp. 580-585
- [17] 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
- [18] 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
- [19] I. Cederbaum, I. Koren and S. Wimer, "Balanced block spacing for VLSI layout," *Discrete Applied Mathematics*, Vol. 40, Issue 3, 1992, pp. 308-318.
- [20] 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

- [21] T. H. Cormen, C. H. Leiserson and R. L. Rivest, *Introduction to Algorithms*, MIT Press, 2nd Edition. 2001.
- [22] R. Chen and H. Zhou, "An Efficient Data Structure for Maxplus Merge in Dynamic Porgramming", *IEEE Transaction on Computer-Aided Design of Integrated Circuits and Systems*, 2005, pp. 3004 – 3009.
- [23] 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.
- [24] C.-K. Cheng, J. Lillis, S. Lin and N.H. Chang, Interconnect Analysis and Synthesis, John Wiley Press, 2000
- [25] D. A. Hodges, H. G. Jackson and R. A. Saleh, Analysis and Design of Digital Integrated Circuits, Mc. Graw Hill, 3rd edition, 2004



Figure 1: Breakdown of dynamic power into local blocks and global interconnects. As can be seen, the cross capacitances between local wires in functional blocks contribute 40% of the total dynamic power, hence have high potential for power save.



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.



Figure 3: The progression of DP from stage *j* to stage *j*+1 by state augmentation. Left circles represent states of stage Λ_j . The solid outgoing arcs represent allocation of a width *w* to wire I_{j+1} and the successive space *s* between I_{j+1} and I_{j+2} yielding feasible states at stage Λ_{j+1} . Dashed arcs represent non feasible state augmentations.



Figure 4: Power-delay design envelop. 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 curve called shapefunction. The green circles are the worst power-delay results.