A High-level Interconnect Power Model for Design Space Exploration

Pallav Gupta, Lin Zhong, and Niraj K. Jha
Dept. of Electrical Engineering
Princeton University
Princeton, NJ 08544
{pgupta|lzhong|jha}@ee.princeton.edu

Abstract— In this paper, we present a high-level power model to estimate the power consumption in semi-global and global interconnects. Such interconnects are used for communications between logic modules, clock distribution networks, and power supply rails. The main purpose of our model is to set forward a simple methodology to efficiently obtain first-order estimates of interconnect power in early stages of the design process. Hence, the objective is to provide designers and/or high-level design automation tools with a way to quickly explore the design space and weed out architectures whose interconnect power requirements do not meet the allocated power budget. In addition to switching power, which includes inter-wire coupling, our model also considers power due to vias and repeaters. Our experimental results show that in comparison to an accurate low-level model, the error in our method in estimating total switching power is only 6% (while the speedup is three-to-four orders of magnitude), and an estimate of the numbers of vias (hence, via power) is within 3% agreement of that obtained for designs synthesized by commercial tools. Furthermore, we develop a probabilistic segment length distribution model for cases in which Rent's rule is inadequate. By analyzing the netlists of a set of complex designs, we have been able to validate our segment length distribution model. The novelty of this work lies in the introduction of a high-level interconnect modeling methodology in which it is possible to efficiently compute all the major sources of power consumption in interconnects and hence, enable interconnect-aware, high-level design space exploration.

I. INTRODUCTION

Rapid innovations in the semiconductor industry have enabled very large scale integrated (VLSI) circuits to migrate into nanometer technologies and operate in the multi-gigahertz frequency range [1]–[8]. This has led to semi-global and global interconnects, which comprise the power supply rails, the clock distribution networks, and the on-chip and off-chip communication links between logic modules, to dominate power consumption, execution time, cost, and manufacturability of a VLSI chip. Hence, many researchers have pointed out that economical design of present and future chips is limited by their wiring requirements [1], [6], [9], [10].

There are four main reasons why interconnects have become the center of attention with respect to power consumption of a circuit. First, interconnects have not scaled exponentially like transistors in sub-micron technologies. Therefore, interconnect capacitance now forms a larger proportion of the total chip capacitance [11], [12]. Second, the current problem of modeling deep sub-micron (DSM) effects was generally ignored in past technologies because transistors remained the focus due to their relatively large size. For example, DSM phenomenon such as mutual coupling between adjacent interconnects, which increases the switched capacitance, has become increasingly dominant [3], [4], [11]. Third, interconnects are now proportionately longer, which implies that the interconnect delay has increased. Finally, the introduction of large numbers of repeaters and vias to reduce wire delay almost doubles the power consumption in interconnects [7]. As shown in [13], interconnects are going to consume a larger proportion of total chip power in future technologies.

There has been considerable work done in developing power models for interconnects. In [1], [4], [14], the authors have discussed a suite of tools to analyze the interconnect requirements of a chip and provide the designer with estimates of power consumption. The authors in [3], [15]–[17] present closed-form expressions to estimate coupling power based on an analysis of lossy transmission lines and distributed RLC circuits. Although the above methods provide quite accurate results, a drawback is that the respective authors consider only a subset of the sources of power consumption

in interconnects in their models. In addition, their dependence on circuit-level parameters makes these models suitable for low-level power estimation only. Rent's rule is used in [9], [10] to derive a stochastic wire-length distribution to study the wiring limitations in present and future designs. Finally, a survey of advances in system-level interconnect prediction based on Rent's rule is presented in [18].

Since interconnect power has become an area of concern, recent work in high-level synthesis (HLS) and system-on-a-chip (SoC) synthesis has started to take it into account [19]–[22]. However, primitive interconnect power models are used. In [23], the authors present a comprehensive treatment of interconnect power consumption during HLS without addressing the problem of interconnect power modeling. This is an example of where our work can be applied for better design space exploration.

The purpose of this work is to propose a simple, yet efficient, high-level power model to estimate the power consumption in semi-global and global interconnects. For our purposes, these interconnects are limited to the data transfer wires between logic modules. We provide a comprehensive treatment of all the major sources of power consumption in interconnects. Since coupling capacitance is expected to dominate in nanometer technologies, we take it into consideration. We also target power consumed by repeater insertion and vias. The new contributions of this work are as follows:

- A power model for estimating interconnect power that is effective at the behavior and/or register-transfer level (RTL), unlike other models which are primarily suitable for low-level power estimation.
- Ability of our model to enable interconnect-aware, high-level design space exploration due to its simplicity and computational efficiency.
- Introduction of a global interconnect modeling methodology in which a probabilistic segment length distribution model is developed to estimate unit-length switching power.

The remainder of this paper is organized as follows. In Section II, we present some motivational examples for our work. Section III provides an overview of our interconnect modeling methodology and shows how it can be readily incorporated into traditional design flows. The detailed description of our methodology is developed in Sections IV and V. We discuss our experimental results in Section VI and conclude in Section VII.

II. MOTIVATIONAL EXAMPLES

We present examples of two application domains which can benefit immensely from the use of our proposed model. We should mention that the usage of an inadequate power model for interconnects by a tool in these application areas motivated this work.

In SoC synthesis, a set of tasks, often represented by a set of task graphs, must be mapped onto processing elements (PEs) (i.e., ASICs, IP cores, processors, etc.) of a system architecture such that some cost function (typically based on area, power, etc.) is minimized. MOCSYN [22] is a tool based on a genetic algorithm that is used to synthesize SoCs. It performs an initial task assignment, estimates the system cost, and maintains a set of solutions which evolve over time. In order to prune the search space, it relies on the fact that accurate area, power, and delay estimates for PEs are available. Interconnect power of the system is unknown *a priori* as it is highly dependent upon the number of PEs used and how they are interconnected. Thus, MOCSYN must estimate the interconnect power of an architecture at each iteration and it is imperative that such calculations be made quickly if tractable run-times are to be realized.
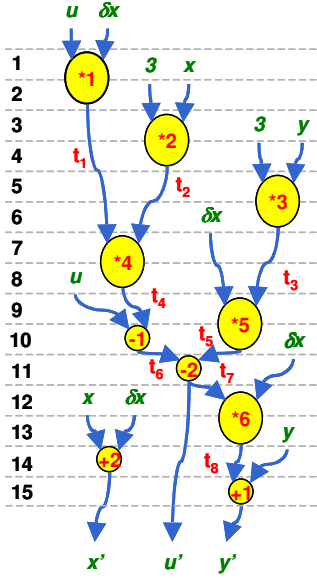
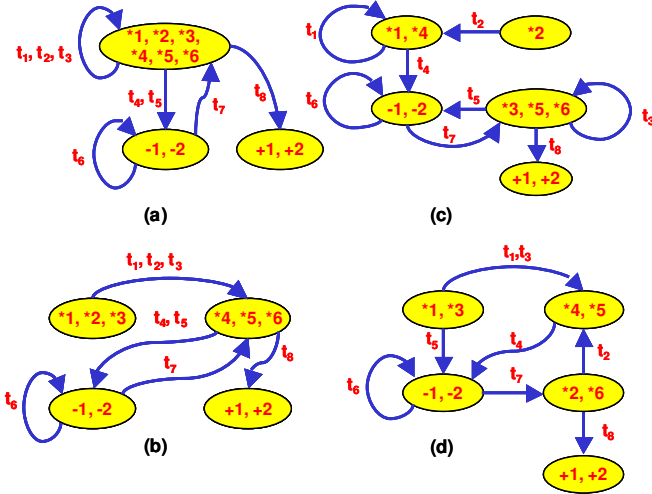Fig. 1.   A scheduled CDFG for the *Diffeq* benchmark.

Fig. 2.   Different bindings of the multiplication operations.

HLS for low power has been the focus of much research in the past decade [24]. It takes as its input a behavioral description in the form of a control-data flow graph (CDFG) and outputs a power-optimized RTL circuit. Iterative improvement algorithms can be used to apply a sequence of moves to an initial RTL architecture. The sequence of moves is accepted if the resulting architecture lowers power consumption.

As a motivational example, we consider a differential equation solver, *Diffeq*, from the NCSU CBL HLS benchmark suite [25]. One possible scheduled CDFG of the *Diffeq* benchmark is shown in Fig. 1. Each sample requires a processing time of 15 control steps with the variables in each sample becoming available in different control steps. For example, δx, x, and y are available in control steps 1, 3, and 5, respectively. Furthermore, multiplication operations *1 - *6, subtraction operations -1, -2, and addition operations $+1$, $+2$ require two, one, and one control steps, respectively.

Figs. 2(a)-(d) show four competing architectures that a high-level synthesis tool, such as SCALP [26], might consider in implementing the final RTL circuit. All of the architectures contain one adder and one subtracter (an oval represents a functional unit). Consequently, the bindings of the -1, -2, $+1$, and $+2$ operations are fixed. However, there are one, two, and three multipliers in the datapath of the architectures of Fig. 2(a), Fig. 2(b), and Figs. 2(c)-(d), respectively

(these correspond to different schedules). This results in different bindings of the multiplication operations to different functional units.

At first glance, it may seem that Fig. 2(a), due to its smallest area, is the best implementation. However, from a power point of view, it is inferior to the architectures in Figs. 2(b)-(d). Due to heavy data exchanges, there will be significant spurious switching activity on the interconnects which will result in higher power consumption [23]. Similar reasoning shows that the architecture in Fig. 2(d) is inferior to that in Fig. 2(c). However, given Fig. 2(b) and Fig. 2(c), it will be difficult for design automation tools to make a good judgment without having to spend considerable time performing detailed simulation and possibly going to layout to get accurate power consumption numbers. It is here where our model can assist the tools.

In HLS and SoC synthesis, design space exploration entails the evaluation of hundreds of competing architectures at each iteration. It is impractical to descend to the layout stage for each candidate to determine what its interconnect power consumption would be. The use of our model, which we now discuss, would enable tools to efficiently traverse and prune the design space during synthesis.

III. OVERALL METHODOLOGY

We briefly give an overview of our interconnect modeling methodology here before diving into its details in later sections. Fig. 3 presents the main steps needed in our flow to estimate the power consumption of an interconnect. It is assumed that a high-level RTL floorplan of the design, lookup tables for κ (Section V-B), the switching activity on the interconnects, and the switching power for interconnects of various lengths have been determined *a priori* and are available in a database. Except for the RTL floorplan, we will describe how all the required pieces of information can be constructed later.

The fidelity of any interconnect power model lies in its ability to accurately estimate wire length, metal layer, and switching activity. Our methodology consists of two models which we choose to call the global wire model and the power models, respectively. Based on some criteria to be discussed later, we estimate the value of a parameter that is used by our global wire model to calculate the segment length distribution and the mean total switching power of an interconnect. This model also estimates the number of repeaters and vias required for this interconnect. This information is then used by the power models to calculate the unit-length switched capacitance. Next, all the various components of interconnect power are estimated and summed to produce total power. The procedure is iterated for each semi-global and global interconnect present in the design.

To avoid any confusion in our discussion, we must emphasize that our model is targeted toward semi-global and global interconnects. For our purposes, these interconnects form the data transfers (e.g., buses) between logic modules. Our model does not consider local interconnects. We assume that the power consumed by such interconnects is computed as part of a module's total power consumption and is available in the RTL design library of components.

From Fig. 3, it can be seen that our methodology is very simple and hence, high-level design automation tools can easily integrate it within their design flow. Because our power models consist of closed-form mathematical expressions, they are computationally efficient and the overhead from their usage is minimal. When a tool has finished synthesizing a circuit, it can call our model within its inner loop to obtain power estimates. It can then use this information to drive its search for a better, alternative implementation of the circuit if given constraints are not met.

IV. POWER MODELS

In this section, we present our power models to estimate the power consumption of semi-global and global interconnects. The role of the models is to calculate the switching power (including power due to inter-wire coupling), the power due to vias, and the power due to repeaters. The power models have neither the knowledge of the length of a wire nor the number of vias and repeaters on that wire. They rely on the global wire model to obtain this information.

The total power, P_{total}, consumed by an interconnect is given by,

$$P_{total} = P_{sw} + P_{vias} + P_{repeaters}, \qquad (1)$$

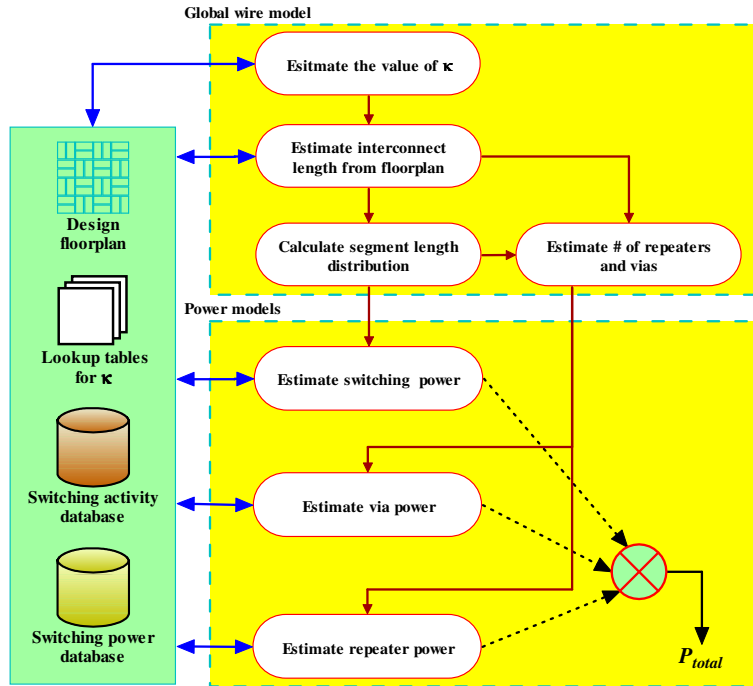where the quantities on the right hand side of (1) are defined as follows:

Fig. 3. Flow diagram providing an overview of our methodology for estimating interconnect power to enable high-level design space exploration.
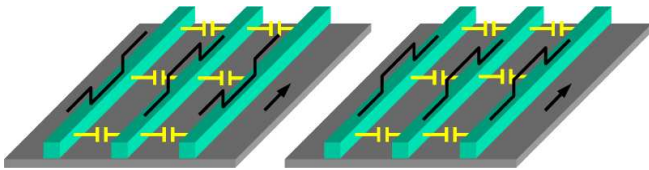

Fig. 4. (a) $180°$ out of phase, and (b) in phase signals.

P_{sw} :   Interconnect power resulting from switched interconnect capacitance and inter-wire coupling;

P_{vias} :   Power consumed by vias due to the use of multiple metal layers; and

$P_{repeaters}$:   Power consumed by repeaters inserted on an interconnect to minimize delay.

We discuss each of the above components separately in the following subsections.

A. Switching Power

In nanometer technologies, DSM effects such as mutual coupling become a significant source of power consumption because at small feature sizes, inter-wire capacitances dominate. This problem increases for wires in a multi-level interconnect structure because wires at higher levels are farther away from the substrate and run in parallel for longer distances [11], [12]. A variety of interconnect power models considering coupling effects have been proposed [3], [4], [16], [17].

In [27], a novel table-lookup method is presented where the total switching power is determined not by the number of transitions but by the *types* of transitions that can occur on an interconnect. This method implicitly assumes that transitions on the interconnects are synchronized and thus, the logic modules are flip-flop bounded. For example, it can be seen that due to switching activity on parallel interconnects, the switching power in Fig. 4(a) will be greater than that in Fig. 4(b) because in the former, the neighboring signals are $180°$ out of phase with respect to the center signal while in the latter, they are all in phase. Because coupling effects between interconnects decrease sharply the further apart they are, only an interconnect and its immediate neighbors need to be considered.

Table I shows the set, T, of all the various transitions that are possible on three-wire interconnects. The idea then is to use low-

level transistor simulation to construct *three-wire* lookup tables for minimally-spaced wires of various lengths that provide the switching power consumed for a given type of transition. This constitutes the switching power database in Fig. 3. The advantage of this method is that in using low-level simulation, it is possible to accurately model the electrical characteristics of a wire for a given process technology. Furthermore, since only three wires need to be simulated, the required time is negligible. Total switching power can then be estimated simply by counting the types of transitions on each interconnect and performing a table lookup. Note that this method does not consider the effect of glitches. We expect good synthesis tools to employ techniques such as those presented in [28] to suppress glitches. Furthermore, glitch estimation is very difficult at the behavior level or RTL and would make the method unnecessarily complex. We adopt the table-lookup method in our model. However, rather than counting the types of transitions, we propose a way of *estimating* them.

TABLE I
TYPES OF TRANSITIONS

S S S	S X S	S S X	S X O	S X X
X X X	X S X	X X O	O X O	X S O

s = stationary, x = transition, o = opposing transition.

The reason why we want to estimate, rather than count, the types of transitions is motivated by the fact that designers and/or high-level design automation tools will be considering multiple architectures to implement a design. Many of these architectures will be very similar to each other except in a few enhancements and modifications. Thus, it is useless and time-consuming to run a full-fledged simulation on each architecture to characterize the switching activity on its interconnects. Given that a particular architecture does not change drastically, it suffices to characterize the switching activity on this architecture only. The data, which form the switching activity database in Fig. 3, can then be used to obtain first-order estimates of switching activity on similar architectures.

Consider the diagram shown in Fig. 5. Let σ denote the logic value on the ith output line of a logic module. We calculate the transition probabilities on this output line by simulating the set of input traces, V. This simulation can be easily done at the behavior level or RTL. The representative input traces for a given application are generally
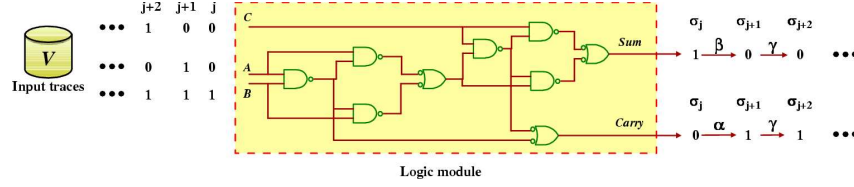
Fig. 5. Estimation of switching activity on the output lines of logic modules.

known. If not, a random vector sequence can be used. We define α, β, and γ to be the $0 \rightarrow 1$ transition, $1 \rightarrow 0$ transition, and no transition (i.e, $0 \rightarrow 0$ and $1 \rightarrow 1$) probabilities, respectively. Then, the transition probabilities of the ith output line are given as,

$$\alpha_i = \frac{1}{|\boldsymbol{V}|-1} \sum_{j,\, j+1 \in \boldsymbol{V}} (\sigma_j = 0) \wedge (\sigma_{j+1} = 1) \quad (2)$$

$$\beta_i = \frac{1}{|\boldsymbol{V}|-1} \sum_{j,\, j+1 \in \boldsymbol{V}} (\sigma_j = 1) \wedge (\sigma_{j+1} = 0) \quad (3)$$

$$\gamma_i = \frac{1}{|\boldsymbol{V}|-1} \sum_{j,\, j+1 \in \boldsymbol{V}} (\sigma_j = \sigma_{j+1}), \quad (4)$$

where σ_j and σ_{j+1} are the output responses to the input vectors, j and $j+1$, respectively. In (2)-(4), we simply count the number of transitions that occur on each line as a result of the application of the input vectors. The total count is divided by the number of input vectors applied to obtain the transition probabilities. This characterization is a one-time cost, unless the set of input traces or the architecture changes significantly. In that case, the transition probabilities will have to be recomputed.

Once the output transition probabilities have been determined, the probabilities for the different types of transitions in Table I can be calculated. For example, the probability of an *sxs* type transition occurring on a three-wire interconnect is the probability that a transition (i.e., $0 \rightarrow 1$ or $1 \rightarrow 0$) occurs on the center interconnect and no transition occurs on the adjacent interconnects. Similarly, the probability of an *ssx* transition occurring is the probability that no transitions occur on the first and second interconnects and a transition occurs on the third interconnect *or* a transition occurs on the first interconnect and no transitions occur on the second and third interconnects.

Assuming that the output lines are independent, we can calculate the probability of a given transition type. Although this assumption is not strictly valid, it is still a reasonable assumption because any correlation that does exist between the output lines has already been accounted for by the transition probabilities to a certain extent. We will validate this assumption when we present our experimental results in Section VI. The probability, $p(t)$, for a given transition type, $t \in \boldsymbol{T}$, on the ith output line is given by one of the following:

$$p(s,s,s) = \gamma_{i-1} \cdot \gamma_i \cdot \gamma_{i+1} \quad (5)$$

$$p(s,x,s) = \gamma_{i-1} \cdot (\alpha_i + \beta_i) \cdot \gamma_{i+1} \quad (6)$$

$$\begin{aligned} p(s,s,x) = \ &\gamma_{i-1} \cdot \gamma_i \cdot (\alpha_{i+1} + \beta_{i+1}) \\ &+ (\alpha_{i-1} + \beta_{i-1}) \cdot \gamma_i \cdot \gamma_{i+1} \end{aligned} \quad (7)$$

$$\begin{aligned} p(s,x,o) = \ &\gamma_{i-1} \cdot (\alpha_i \cdot \beta_{i+1} + \beta_i \cdot \alpha_{i+1}) \\ &+ (\alpha_{i-1} \cdot \beta_i + \beta_{i-1} \cdot \alpha_i) \cdot \gamma_{i+1} \end{aligned} \quad (8)$$

$$\begin{aligned} p(s,x,x) = \ &\gamma_{i-1} \cdot (\alpha_i \cdot \alpha_{i+1} + \beta_i \cdot \beta_{i+1}) \\ &+ (\alpha_{i-1} \cdot \alpha_i + \beta_{i-1} \cdot \beta_i) \cdot \gamma_{i+1} \end{aligned} \quad (9)$$

$$p(x,x,x) = \alpha_{i-1} \cdot \alpha_i \cdot \alpha_{i+1} + \beta_{i-1} \cdot \beta_i \cdot \beta_{i+1} \quad (10)$$

$$p(x,s,x) = \alpha_{i-1} \cdot \gamma_i \cdot \alpha_{i+1} + \beta_{i-1} \cdot \gamma_i \cdot \beta_{i+1} \quad (11)$$

$$\begin{aligned} p(x,x,o) = \ &(\alpha_{i-1} \cdot \alpha_i + \alpha_{i-1} \cdot \beta_i) \cdot \beta_{i+1} \\ &+ (\beta_{i-1} \cdot \beta_i + \beta_{i-1} \cdot \alpha_i) \cdot \alpha_{i+1} \end{aligned} \quad (12)$$

$$p(o,x,o) = \alpha_{i-1} \cdot \beta_i \cdot \alpha_{i+1} + \beta_{i-1} \cdot \alpha_i \cdot \beta_{i+1} \quad (13)$$

$$p(x,s,o) = \alpha_{i-1} \cdot \gamma_i \cdot \beta_{i+1} + \beta_{i-1} \cdot \gamma_i \cdot \alpha_{i+1}. \quad (14)$$

The total switching power of a single interconnect (i.e., an output line), i, of length, l, is calculated as the sum of the total number of each type of transition multiplied by its corresponding switching power, P_{t_l}, from the lookup table. Note that P_{t_l} may need to be extrapolated using the data for the lengths that are closest in proximity to l. The number of transitions of a particular type can be estimated by its probability, $p(t)$, and by the number of input vectors, N, applied. Summing over all interconnects, $\boldsymbol{I}$ (i.e., all output lines), gives the total switching power, P_{sw}, as

$$P_{sw} = (N-1) \sum_{i \in \boldsymbol{I}} \sum_{t \in \boldsymbol{T}} p_i(t) P_{t_l}. \quad (15)$$

In (15), it is necessary to estimate the length, l, of an interconnect. In the industry, it is very unlikely that any chip design project begins without an approximate, initial floorplan. This is because a floorplan provides the minimum amount of information needed by high-level estimation tools to provide designers with estimates of various parameters like clock speed, power, area, etc. Thus, we assume that a high-level RTL floorplan is always available for determining the approximate distance between logic blocks. Accordingly, we use center-to-center Manhattan distance between the blocks to estimate the length of an interconnect. Experiments with commercial routing tools indicate that in cell-based designs, approximately only 2% of semi-global and global interconnects violate Manhattan distance routing when performing automatic place and route.

B. Power Due to Vias

Vias or contacts serve two purposes. They form the bridge between (a) transistors residing on the substrate and the interconnect connecting these transistors, and (b) interconnects running on multiple metal layers. Traditionally, the power consumed by vias has been totally ignored in estimating interconnect power. However, because the use of repeaters is a very popular design practice, the power consumed by vias must be taken into account to get a more accurate picture of total interconnect power consumption.

The power consumed by vias, P_{vias}, is estimated by the product of the number of vias, V_N, and the power consumed by a single via, P_{via}. We will describe how to estimate the total number of vias in the next section. Thus, we have,

$$P_{vias} = V_N P_{via}. \quad (16)$$

P_{via} is heavily dependent upon the layer in which it resides. Currently, our model does not take this into account because we have not addressed the metal layer assignment problem (Section V-E). Hence, P_{via} is approximated either by taking an average of the power consumed by vias of different configurations or by assigning a weighting factor to each via configuration which represents its proportionate contribution to all the vias present in the layout of a circuit. Note that the power consumed by vias used in repeaters is not estimated by (16). This will be accounted for in the following subsection.

C. Power Due to Repeaters

Due to the shrinking feature size, interconnect wires are proportionately getting longer as they are not scaling as well as transistors. A wire can be modeled by a simple RC network. It is known that its delay is a quadratic function of length because its resistance and its capacitance are functions of its length [11]. Thus, the delay of a wire has been increasing and the insertion of repeaters (a.k.a buffers) has become common [6], [12], [23]. It is an accepted practice to insert repeaters when the repeated wire delay is less than the unrepeated delay [2], [7].

The global model determines the number of repeaters, N_R, required on an interconnect. We will describe how this is estimated in the next section. Once N_R is known, the number of repeater vias, V_R, can be obtained as twice the number of repeaters because separate

paths are needed to descend to and ascend from the substrate (where repeaters reside). Total repeater power is then obtained by summing repeater power over all interconnects, as follows:

$$P_{repeaters} = C_{repeater} V_{dd}^2 f \sum_{i \in I} \rho_i N_{R_i} + P_{via} \sum_{i \in I} V_{R_i}, \quad (17)$$

where the capacitance of a single repeater, $C_{repeater}$, is given by the equations in [7], ρ_i is the switching activity, V_{dd} is the operating voltage, f is the clock frequency, and P_{via} is as described in the previous subsection. We observe that since repeaters are inserted on interconnects responsible for communication between logic modules, their switching activity is determined by the switching activity on the output lines of the modules. We also note that if the designer so desires, he or she can add a term in (17) to incorporate leakage power in the repeaters.

V. GLOBAL WIRE MODEL

In this section, we present a global modeling methodology for interconnects. Global wire modeling attempts to develop a statistical model that describes the characteristics of a single interconnect, given its initial and terminal coordinates in the floorplan. It is differentiated from the power models presented in the last section in that, it takes a global perspective to estimate the topology and the layout of an interconnect in a probabilistic way. Its primary purpose is to determine the segment length distribution of an interconnect and the number of vias and repeaters on this interconnect. As shown in Fig. 3, this information is passed to the power models to obtain actual power estimates. As mentioned earlier, accurate wire length estimation indirectly validates our methodology since it is the most important variable in our model. We will verify our global wire model and hence, our overall methodology this way in Section VI.

Power models and global wire modeling techniques are relatively independent of each other and the designer can combine different methods from each domain in his or her model. Nevertheless, both are equally important to wire modeling. As seen previously, there are a variety of power models to which we have added ours. However, there does not exist any high-level global wire modeling technique that we are aware of. The following is the first step in the direction of developing such a model.

A. Intermediate Point Model

Traditionally, Rent's rule [12] has been applied to estimate the length of an interconnect [7], [9], [14], [18]. It is an empirical power law that holds well for modular designs with blocks containing 50,000 - 75,000 gates on average. It is stated as follows:

$$M = kG^p, \quad (18)$$

where M is the number of terminals of a block, G is the number of gates per block, and k and p are empirical constants. Generally, one proceeds by making an assumption about the values of k and p and that they are constant. Unfortunately, this assumption is not applicable in areas like HLS and SoC synthesis where high-level design automation tools produce different circuit configurations for a given high-level description due to different binding and scheduling scenarios [23]. Since one of our objectives is to target such domains, we choose not to use Rent's rule. Instead, we seek to develop an alternative methodology for determining the segment length distribution of an interconnect.

To determine the probabilistic segment length distribution of an interconnect, consider the diagram shown in Fig. 6. We must route between points A and B by possibly going through n intermediate points, D_1, D_2, ..., D_n. These intermediate points can represent some obstacles or other constraints that must be satisfied. We make the following assumptions in our analysis:

1) we consider the Manhattan distance between points A and B;
2) we must route through *all* intermediate points and at each point we change directions (i.e., switch metal layers); and
3) the occurrence of intermediate points can be modeled as a Poisson process.

We have been able to verify all of these assumptions experimentally.

We apply our analysis to the horizontal axis only, since the analysis for the vertical axis is similar due to orthogonality. Let A be the
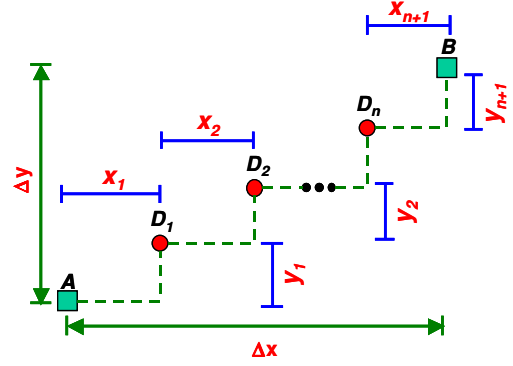

Fig. 6. Routing point A to point B via intermediate points D_1, ..., D_n.

reference point and Δx represent the horizontal Manhattan distance to B. The occurrence of the number of intermediate points, N, is a sequence of discrete events that can be modeled by a Poisson process and is given as,

$$p(N = n) = \frac{(\kappa \Delta x)^n e^{-\kappa \Delta x}}{n!}, \quad (19)$$

where κ is an empirical constant that will have to be determined on a per design basis. The segment length distribution, L, between two consecutive intermediate points, say D_1 and D_2 (with n intermediate points on the interconnect), is given by,

$$p(L|N = n) = \begin{cases} \frac{n}{\Delta x}\left[1 - \frac{L}{\Delta x}\right]^{n-1} & n \geq 1 \\ \delta(L - \Delta x) & n = 0. \end{cases} \quad (20)$$

Now, the joint probability of the segment length distribution and the number of intermediate points is the product of (19) and (20). Thus,

$$\begin{aligned} p(L, N) &= p(L|N = n) \cdot p(N = n) \\ &= \begin{cases} \frac{n}{\Delta x}\left[1 - \frac{L}{\Delta x}\right]^{n-1} \frac{(\kappa \Delta x)^n e^{-\kappa \Delta x}}{n!} & n \geq 1 \\ \delta(L - \Delta x)e^{-\kappa \Delta x} & n = 0. \end{cases} \end{aligned} \quad (21)$$

To calculate the segment length distribution of the first segment, we sum (21) for $N = 0 \rightarrow \infty$ to get,

$$p(L) = \sum_{N=0}^{\infty} p(L, N) = \delta(L - \Delta x)e^{-\kappa \Delta x} + \kappa e^{-\kappa L} \quad 0 < L \leq \Delta x. \quad (22)$$

The term $\delta(L - \Delta x)e^{-\kappa \Delta x}$ in (22) takes into account those interconnects which have zero intermediate points. As can be seen, this probability is very small when Δx is large.

The mean power, W_j, of segment j is given by,

$$\begin{aligned} m(W_j|N = n) &= \begin{cases} P_u^j \int_0^{\Delta x} w_j \frac{n}{\Delta x}\left[1 - \frac{w_j}{\Delta x}\right]^{n-1} dw_j \\ P_u^j \Delta x \end{cases} \\ &= \begin{cases} P_u^j \frac{\Delta x}{n+1} & n \geq 1 \\ P_j^j \Delta x & n = 0, \end{cases} \end{aligned} \quad (23)$$

where P_u^j is the unit-length total switching power of segment j given by the switching power model described in Section IV-A. The total mean switching power of an interconnect (with n intermediate points), W_{sw}, is the sum of the means of the total switching power of its segments. That is,

$$m(W_{sw}|N = n) = \sum_{j=0}^{n} m(W_j|N = n) = \frac{\Delta x}{n + 1} \sum_{j=0}^{n} P_u^j. \quad (24)$$

The expectation of the total switching power is then,

$$m(W_{sw}) = \sum_{n=0}^{\infty} m(W_{sw}|N=n)p(N=n)$$

$$= \Delta x e^{-k\Delta x} \sum_{n=0}^{\infty} \frac{(k\Delta x)^n}{(n+1)n!} \sum_{j=0}^{n} P_{u,n}^j, \qquad (25)$$

where $P_{u,n}^j$ is decided by the switching power model and metal layer assignment.

B. Estimating the Value of κ

In our model, κ is the only parameter that needs to be empirically determined. We propose a few ideas that can be used to obtain its value. We believe κ will be highly dependent upon the following factors:

1) routing tool and algorithm;
2) total chip area of the design; and
3) design style.

Different routing tools and algorithms will produce different interconnect structures depending upon their sophistication. Though the segment length distribution will still follow an exponential distribution, κ will be different. A bigger chip will require longer semi-global and global interconnects that will affect κ. Finally, the type of design style will also have an impact. The interconnect structures for ASICs, full-custom, and semi-custom designs are quite different. For example, in digital signal processing (DSP) applications, data transfers tend to occur on dedicated interconnects and are multiplexed. They are usually called multiplexer-based interconnects. However, in microprocessors, shared interconnects in the form of busses exist between components that wish to communicate with each other. Such architectural and "design style" differences will manifest themselves in the need for different amounts of interconnect.

To overcome some of the above issues, we propose that a series of lookup tables be built with respect to design styles for each type of routing tool that is being used. Within each of these tables, each chip of a particular area will have a value of κ ascribed to it. This is the value that is to be used in our model. For chip areas not present in the tables, one will have to resort to interpolation to obtain the value of κ. The experience of the designer will be needed to help select the appropriate design style table.

The task of building design style tables may seem daunting, but it is quite reasonable in practice. First, most companies have a few set of design styles and/or routing tools. Thus, the tables will be quite small. Second, the entries in the lookup tables need not be exhaustive. Finally, companies can use their old designs to characterize κ and use the obtained data in the lookup tables. These tables constitute the lookup tables for κ in Fig. 3.

C. Estimating Number of Vias

In estimating the number of vias, we again consider Fig. 6. Since we have imposed the restriction that we must change directions every time we encounter one of the intermediate points, it is obvious that there exist only two possible paths in routing from A to B. The number of vias required in each case is $2n+1$. If we removed the above restriction, there would be 2^{n+1} possible paths from A to B. Of these, 2^{n-1}, 2^{n-1}, and 2^n paths would have $2+n$, $2+2n$, and $4+2n$ vias, respectively. The average via per path is the total number of vias divided by the total number of possible paths. This yields an average of $\lceil 3 + 1.75n \rceil$ vias per path. This result shows that our original restriction is reasonable. The total number of vias, V_N, is estimated by calculating and summing the number of vias at each point A that needs to be routed to each point B.

D. Estimating Number of Repeaters

In estimating the number of repeaters, we use the approach presented in [7]. For a given process technology, the optimal distance, l_{opt}, between repeaters was originally formulated in [12] and restated in [7] as,

$$l_{opt} = 3.24\sqrt{\frac{r_o C_{nmos}}{R_w C_w}}, \qquad (26)$$

where r_o and C_{nmos} are the resistance and capacitance of a minimum-sized nMOS transistor, respectively, and R_w and C_w are the resistance and capacitance per unit length of the wire, respectively. The number of repeaters, N_R, and the number of repeater vias, V_R, on a wire of length, Δx, is given by:

$$N_R = \frac{V_R}{2} = \begin{cases} 0 & \text{if } \Delta x < l_{opt} \\ \lceil \frac{\Delta x}{l_{opt}} \rceil - 1 & \text{otherwise.} \end{cases} \qquad (27)$$

E. Metal Layer Assignment

We devote this subsection to discussing the problem of assigning metal layers to interconnects. In a multi-layer interconnect scheme, it is important to determine the probability that an interconnect will reside on a particular layer as well as determine the criteria under which it will be promoted to the next layer. This will aid in understanding how interconnects are distributed across metal layers. It will also help in estimating the capacitance of an interconnect since different segments residing on different layers will have different capacitances. Intuitively, we can say that the longer the interconnect, the greater its probability of being situated in a higher metal layer.

This is a difficult question that is open for further research. We state the problem as follows. Given m levels of metal layers, l_1, l_2, ..., l_m, in a specific process technology that run in a particular direction (either horizontally or vertically), what is the probability, $p(l_1)$, $p(l_2)$, ..., $p(l_m)$, that the segments of an interconnect of length Δx with n intermediate points reside at a particular metal layer l_k ($k \in \{1, 2, \cdots, m\}$)?

F. Summary

To summarize the global model, the total mean power of an interconnect, W_{total}, is given as,

$$m(W_{total}) = m(W_{sw}) + m(W_v) + m(W_r), \qquad (28)$$

where $m(W_{sw})$ is the total mean switching power given in Section V-A, $m(W_v)$ is the total mean via power given in Section V-C, and $m(W_r)$ is the total mean repeater power given in Section V-D.

VI. EXPERIMENTAL RESULTS

We present experimental results to validate our proposed methodology in this section. Table II shows a brief description of each benchmark in our benchmark suite. The first eight benchmarks are DSP applications while the remaining seven benchmarks are microprocessor core designs [25], [29]. Each benchmark was synthesized using Design Compiler from Synopsys [30] and the VTVT library [31] based on TSMC 0.25μm technology. The circuits were placed and routed automatically using Silicon Ensemble from Cadence [32]. The longest time required to synthesize and place and route a circuit was about 12 hours and 20 minutes. All of our experiments were conducted on a SPARC workstation with four 333 MHz processors and 4GB of main memory.

To our knowledge, there exists no tool to solely provide estimates of interconnect power. Thus, it is not useful for us to present interconnect power estimates because we have no basis for comparison. Consequently, we validate our overall methodology indirectly by verifying the individual components that comprise it. In the future, we plan to implement our model as a package within an HLS framework and report on its estimation results.

A. Switching Power Model

We measured the error between counting and estimating the types of transitions. The transition probabilities were first characterized using input traces of 25,000 vectors. Then, vector sets of length in multiples of 500 (up to 5,000 vectors) were applied to estimate the number of each transition type and the associated switching power given by our proposed model. Fig. 7 compares the resulting error of our method against the method in [27] for benchmarks 1 - 6 using three different sets of vectors (similar results were obtained for the other benchmarks and vector sets). We see that the average error is about 6%. This verifies our assumption that the output lines are independent. Since our method uses equations, its runtime (after initial characterization) is independent of the size of the input traces. Fig. 8 plots the speedup obtained by our method on vector sets of different lengths. We observe that speedup over

TABLE II
BENCHMARK CHARACTERISTICS

| #	Benchmark	Description	Area (μm^2)	Cell count	κ ($\frac{1}{\mu m}$)	R^2
1	Chemical	IIR filter	492,617	6,495	1.067 ± 0.022	0.99412
2	DCT Lee	DCT algorithm	498,986	6,754	1.061 ± 0.013	0.99721
3	DCT Wang	DCT algorithm	551,207	7,269	1.015 ± 0.052	0.95974
4	DCT Dif	DCT algorithm	335,069	4,377	0.941 ± 0.044	0.97227
5	Elliptic	Elliptic wave filter	266,023	3,184	1.035 ± 0.027	0.98844
6	Paulin	Differential equation solver	155,238	1,882	1.031 ± 0.069	0.93601
7	JPEG Encoder	JPEG core	9,302,175	125,870	0.979 ± 0.045	0.97363
8	Biquad Filter	Biquad IIR filter	326,839	4,263	1.069 ± 0.039	0.97822
9	DES	Data Encryption Standard core	6,882,506	83,869	0.782 ± 0.079	0.93248
10	FPU	IEEE 754 floating point unit core	1,098,412	16,315	0.833 ± 0.061	0.95469
11	PLASMA	MIPS core	1,331,585	18,442	0.902 ± 0.029	0.98701
12	RISC	Mini RISC core	190,958	2,097	0.827 ± 0.071	0.94365
13	UART	16550 UART μcontroller core	384,730	4,159	0.939 ± 0.018	0.99523
14	AES Cipher	Rijndael encrypt core	1,266,766	19,751	0.698 ± 0.053	0.96936
15	AES Decipher	Rijndael decrypt core	2,085,187	27,393	0.770 ± 0.044	0.97653
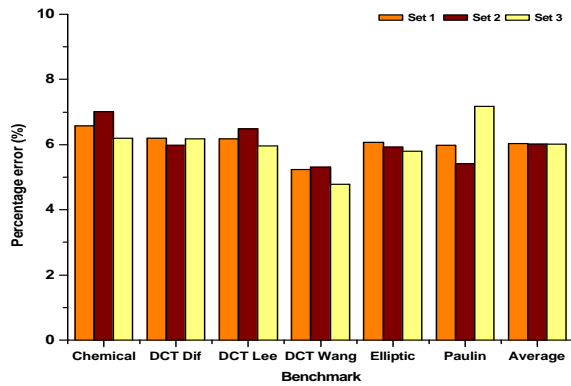
Fig. 7. Error in switching power estimation by estimating the number of transition types rather than explicitly counting.

Fig. 9. Segment length distribution for the benchmarks in Table II. The x-axis represents binned data (in units of 10^3) to get a meaningful value of κ during data regression.
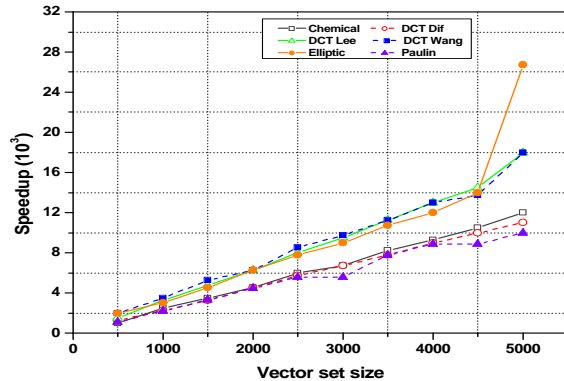

Fig. 8. Total speedup. (*Note*: Initial characterization time is not considered when calculating speedup, as speedup is intended to reflect the advantage gained when using our method iteratively.)

Fig. 10. Value of κ as a function of the length of a chip. The rectangles represent DSP applications (benchmarks 1-8) while the circles represent microprocessor designs (benchmarks 9-15).

four orders of magnitude is attainable for vector sizes of just 5,000. In calculating the speedup, we have not factored in the time for initial characterization because the speedup is intended to reflect the advantage that will be gained when a high-level design automation tool iteratively estimates switching power for similar architectures inside its inner loop as it explores the design space.

B. Global Wire Model

Table II shows the chip size, the total cell count, the value of κ, and the quality of regression fit, R^2, on κ obtained for each of the circuits after place and route. We wrote Perl scripts to analyze the GDSII database which contained all the low-level mask information about interconnects. Essentially, we counted the number of segments
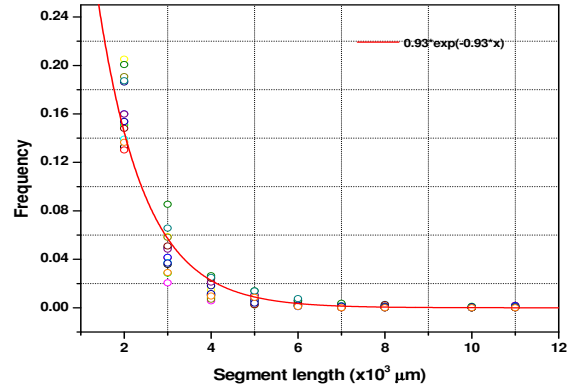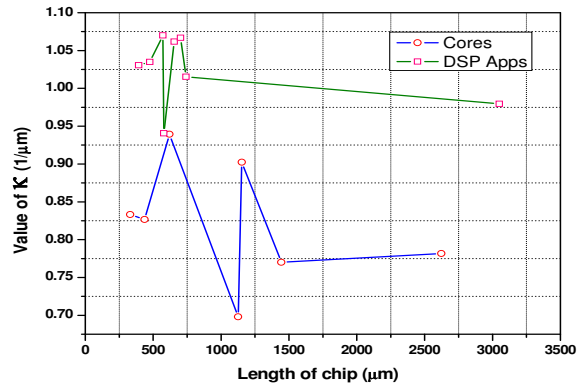
and the length of each segment. Fig. 9 shows a plot of the length of the segment (not to be confused with total interconnect length) versus its frequency for each individual benchmark. The x-axis has been binned (in units of 10^3) to get meaningful values of κ during data regression. The plotted curve with $\kappa = 0.93$ represents the average of all κ in Table II. We can see that the raw data follows an exponential distribution quite nicely. Furthermore, R^2 is high for the majority of benchmarks indicating a good regression fit by Origin [33]. This validates our assumption that we can model the number of intermediate points as a Poisson process.

We also ran tests to see how many of the interconnects violated the Manhattan distance and whether they changed direction at the

TABLE III
ESTIMATION OF VIAS

Benchmark	Actual	Projected	Error (%)
Chemical	31,454	32,763	4.2
DCT Lee	32,701	34,245	4.7
DCT Wang	36,501	38,142	4.5
DCT Dif	22,806	23,422	2.7
Elliptic	17,953	17,934	−0.1
Paulin	10,201	10,537	3.3
JPEG Encoder	606,936	648,771	6.9
Biquad Filter	20,371	21,601	6.0
DES	487,234	491,323	0.8
FPU	87,274	89,916	3.0
PLASMA	117,008	117,405	0.3
RISC	11,084	11,805	6.5
UART	24,025	23,866	−0.7
AES Cipher	118,927	120,383	1.2
AES Decipher	164,593	166,757	1.3

intermediate points. We observed that only about 2% of the semi-global and global interconnects (i.e., routed on metal2 and higher) did not follow a Manhattan distance while every interconnect changed direction at each intermediate point. This verifies our other two assumptions in our global wire model.

From Table II, it can be seen that κ varies from 0.698 to 1.069. However, if we look at Fig. 10, which shows the value of κ as a function of the length of the chip (i.e., square root of area), we can make some interesting observations. First, the value of κ for rectangular points is tightly centered around 1.0 and seems to be independent of circuit size. Furthermore, all these data points represent DSP applications (benchmarks 1-8). The circular points represent microprocessor designs (benchmarks 9-15) and we can see that the situation is reversed here in that, κ varies significantly. This seems to give strong support to our original hypothesis that the value of κ will be dependent upon the "design style." We believe that the variance in κ in the microprocessor designs, is due mainly to the unique architecture of each design. Also, microprocessor designs tend to have a higher proportion of semi-global and global interconnects which should, as observed experimentally, result in a smaller value for κ. The overall conclusion is that the use of lookup tables seems to be an appropriate way to estimate κ.

C. Number of Vias

We used our global wire model to estimate the number of vias that were required in each benchmark. Table III shows the number of vias reported by Silicon Ensemble, the estimate given by our model, and the resulting error. The average absolute error is approximately 3%. It is here that we see the importance of taking via power into consideration. For example, we note that JPEG Encoder has over half a million vias.

VII. CONCLUSIONS

As technology scales, power in semi-global and global interconnects is beginning to have a significant impact on the total power consumption of a chip. It is important to identify the major sources of power consumption in interconnects and quantify them with simple, first-order models. To that end, we have presented a computationally efficient, high-level power model which provides a comprehensive study of the major sources of power consumption. Based on switching probabilities, we have introduced a novel way of estimating, rather than counting, the type of transition to determine switching power. Our model also considers the often ignored via power due to multi-level metal layers and repeater insertion. Furthermore, we have proposed a high-level global wire model in which we have derived a simple segment wire-length distribution for cases in which Rent's rule may be inadequate. We have also posed a question of metal layer assignment to interconnects that needs to be addressed in the near future. Through experimental results, we have been able to validate our methodology. We hope that our methodology can be easily integrated into current high-level design automation tools to get estimates of interconnect power to enable interconnect-aware synthesis.

REFERENCES

[1] J. Cong, "An interconnect-centric design flow for nanometer technologies," *Proc. IEEE*, vol. 89, no. 4, pp. 505–528, Apr. 2001.
[2] J. Cong and Z. Pan, "Interconnect performance estimation models for design planning," *IEEE Trans. Computer-Aided Design*, vol. 20, no. 6, pp. 739–752, June 2001.
[3] M. Kuhlmann and S. S. Sapatnekar, "Exact and efficient crosstalk estimation," *IEEE Trans. Computer-Aided Design*, vol. 20, no. 7, pp. 858–866, July 2001.
[4] T. Uchino and J. Cong, "An interconnect energy model considering coupling effects," *IEEE Trans. Computer-Aided Design*, vol. 21, no. 7, pp. 763–776, July 2002.
[5] A. Deutsch *et al.*, "On-chip wiring design challenges for gigahertz operation," *Proc. IEEE*, vol. 89, no. 4, pp. 529–555, Apr. 2001.
[6] R. Ho, K. W. Mai, and M. A. Horowitz, "The future of wires," *Proc. IEEE*, vol. 89, no. 4, pp. 490–504, Apr. 2001.
[7] P. Kapur, G. Chandra, and K. C. Saraswat, "Power estimation in global interconnects and its reduction using a novel repeater optimization methodology," in *Proc. Design Automation Conf.*, June 2002, pp. 461–466.
[8] T. N. Theis, "The future of interconnection technology," *IBM J. Res. Develop.*, vol. 44, no. 3, pp. 379–390, May 2000.
[9] J. A. Davis, V. K. De, and J. D. Meindl, "A stochastic wire-length distribution for gigascale integration (GSI) – Part I: Derivation and validation," *IEEE Trans. Electron Devices*, vol. 45, no. 3, pp. 580–589, Mar. 1998.
[10] ——, "A stochastic wire-length distribution for gigascale integration (GSI) – Part II: Applications to clock frequency, power dissipation, and chip size estimation," *IEEE Trans. Electron Devices*, vol. 45, no. 3, pp. 590–597, Mar. 1998.
[11] J. M. Rabaey, *Digital Integrated Circuits*. Englewood Cliffs, NJ: Prentice Hall, 1996.
[12] H. B. Bakoglu, *Circuits, Interconnects, and Packaging for VLSI*. Reading, MA: Addison-Wesley, 1990.
[13] D. Sylvester and C. Hu, "Analytical modeling and characterization of deep-submicrometer interconnect," *Proc. IEEE*, vol. 89, no. 5, pp. 634–664, May 2001.
[14] D. Sylvester and K. Keutzer, "System-level performance modeling with BACPAC - Berkeley advanced chip performance calculator," in *Proc. System-level Interconnect Prediction Conf.*, Apr. 1999, pp. 109–114.
[15] D. Sylvester, O. S. Nakagawa, and C. Hu, "An analytical crosstalk model with applications to ULSI interconnect scaling," in *Proc. SRC Technical Conf.*, June 1998.
[16] P. Heydari and M. Pedram, "Interconnect energy dissipation modeling in high-speed ULSI circuits," in *Proc. Asia and South Pacific Design Automation Conf.*, Jan. 2002, pp. 132–140.
[17] P. P. Sotiriadis and A. Chandrakasan, "A bus energy model for deep sub-micron technology," *IEEE Trans. VLSI Syst.*, vol. 10, no. 3, pp. 341–350, June 2002.
[18] D. Stroobandt, "Recent advances in system-level interconnect prediction," *IEEE Circuits and Systems Newsletter*, vol. 19, no. 9, pp. 4–20, Dec. 2002.
[19] R. Mehra, L. M. Guerra, and J. M. Rabaey, "Low-power architectural synthesis and the impact of exploiting locality," *J. VLSI Signal Processing*, vol. 13, no. 8, pp. 877–878, Aug. 1996.
[20] S. Hong and T. Kim, "Bus optimization for low-power data path synthesis based on network flow method," in *Proc. Int. Conf. Computer-Aided Design*, Nov. 2000, pp. 312–317.
[21] P. Prabhakaran and P. Banerjee, "Simultaneous scheduling, binding, and floorplanning in high-level synthesis," in *Proc. Int. Conf. VLSI Design*, Jan. 1998, pp. 428–434.
[22] R. P. Dick and N. K. Jha, "MOCSYN: Multiobjective core-based single-chip system synthesis," in *Proc. Design Automation and Test in Europe Conf.*, Mar. 1999, pp. 263–270.
[23] L. Zhong and N. K. Jha, "Interconnect-aware high-level synthesis for low power," in *Proc. Int. Conf. Computer-Aided Design*, Nov. 2002, pp. 110–117.
[24] A. Raghunathan, N. K. Jha, and S. Dey, *High-level Power Analysis and Optimization*. Norwell, MA: Kluwer Academic Publisher, 1998.
[25] "The benchmark archives at CBL." http://www.cbl.ncsu.edu
[26] A. Raghunathan and N. K. Jha, "SCALP: An iterative-improvement based low power data path synthesis system," *IEEE Trans. Computer-Aided Design*, vol. 16, no. 11, pp. 1260–1277, Nov. 1997.
[27] C. N. Taylor, S. Dey, and Y. Zhao, "Modeling and minimization of interconnect energy dissipation in nanometer technologies," in *Proc. Design Automation Conf.*, June 2001, pp. 754–757.
[28] A. Raghunathan, S. Dey, and N. K. Jha, "Register transfer level power optimization with emphasis on glitch analysis and reduction," *IEEE Trans. Computer-Aided Design*, vol. 18, no. 8, pp. 1114–1131, Aug. 1999.
[29] "Opencores website." http://www.opencores.org
[30] "Synopsys website." http://www.synopsys.com
[31] "VTVT library." http://www.ee.vt.edu/~ha
[32] "Cadence design systems website." http://www.cadence.com
[33] "Origin lab website." http://www.originlab.com