Power-Driven Simultaneous Resource Binding and Floorplanning: A Probabilistic Approach

Azadeh Davoodi, *Student Member, IEEE,* and Ankur Srivastava, *Member, IEEE*

***Abstract*—Floorplanning information is integrated during resource binding for better modeling of the interconnect effects on timing and power. Although this integration improves the estimation of the interconnect effects, nonavailability of exact net-lengths can result in suboptimal solutions, because global routing is not yet performed. In this work we propose a probabilistic approach to integrate floorplanning and resource binding by modeling the distribution of the net-lengths from a given floorplan. The advantage of this approach is that a probabilistic technique can better capture the inaccuracy associated with net-length estimation, and consequently, the inaccuracy in estimation of net-delay and net-power. The result is higher chance of successful synthesis, and therefore faster timing closure. Additionally, due to better management of uncertainty, it has a better overall post-synthesis power. These results are illustrated in our experiments that were conducted using state of the art commercial and academic tools.**

I. Introduction

POWER dissipation is a crucial optimization objective. At higher levels of the design, power optimization tremendously affects the lower levels power. Low power resource binding assigns operations to the available resources such that the correct functionality is maintained and the overall power is minimized. Low power resource binding has been an active topic of research during the past decade [9], [11]. Due to the ever decreasing device feature sizes, when doing resource binding, it is extremely important to consider the interconnect effects on performance and power. This is because interconnects consume a significant fraction of the overall power [6]. Also the interconnect delay tremendously affects the system performance. To do so, many researchers proposed extracting interconnect information from the low-level design stages [10], [12], [16]–[18] for better resource binding. The floorplanning information is used in [18] to provide a rough estimate of the interconnect delay. It performs resource binding and floorplanning iteratively by using the floorplan information for more accurate resource binding, which is then re-floorplanned.

The inherent problem with the existing approaches is in the estimation of the net information. In the low power resource binding problem, net-delay and net-power are two important features that are determined from the floorplan, expressed as a function of the net-length.

The net-length is usually estimated from the floorplan using models such as the half-perimeter bounding box or minimum Steiner tree (MST). Using such approximations results in suboptimal optimization. As an example the half-perimeter bounding box is a lower bound on the actual net-length. Using the half-perimeter bounding box would result in underestimating the net-delay and net-power. Similarly, upper-bound estimates of net-length results in overestimating the delay and power. These estimates generate suboptimal resource binding solutions.

To better consider the inaccurate estimation of the net parameters, we present a probabilistic approach to power-driven simultaneous floorplanning and resource binding. Rather than fixed estimates, we model the distribution of a net-length. This allows considering a range of values for the net-length together with their associated probabilities. Also distributions for the net-delay and net-power are consequently obtained. Using such probabilistic models, the optimization is then performed to maximize the likelihood of meeting the design constraints while minimizing the overall power, thus higher probability of successful synthesis, and faster timing closure. In addition, the probabilistic models result in better estimation of power during optimization, and better design quality. Similar probabilistic methods have been investigated in [2], [4], [15].

In our experimental results, we show that our approach results in faster timing closure when compared to traditional approaches that use half-perimeter bounding box or MST as the net-length estimate. In our case, all or most of the functional modules are successfully synthesized after routing. Even when both types of approaches satisfy the timing requirements, the actual power, including power of the nets, functional modules and registers, after the routing stage, is smaller using our probabilistic approach.

The rest of the paper is organized as follows. Section II explains the traditional approaches and their shortcomings. Section III reviews a probabilistic net-length model. Section IV-A describes the computation of the probability distribution of the entire data flow graph (DFG). This is followed by the algorithm for probabilistic optimization in Section IV-B. Experimental results are in Section V.

II. Traditional Low-Power Binding and Floorplanning

In this section, the low power resource binding and floorplanning problem is defined and the traditional approach to this problem is explained. Then the shortcomings of the existing approaches are explained.

A. Problem Definition and Motivation

Given an initial resource binding of a scheduled DFG with a clock duration of T and its corresponding floorplan, improve the resource binding such that the overall power is minimized.

Manuscript received April 28, 2004; revised September 13, 2004 and March 3, 2005.

The authors are with the Department of Electrical and Computer Engineering, University of Maryland, College Park, MD 20742 USA (e-mail: azade@eng.umd.edu; ankurs@eng.umd.edu).

Digital Object Identifier 10.1109/TVLSI.2005.853618

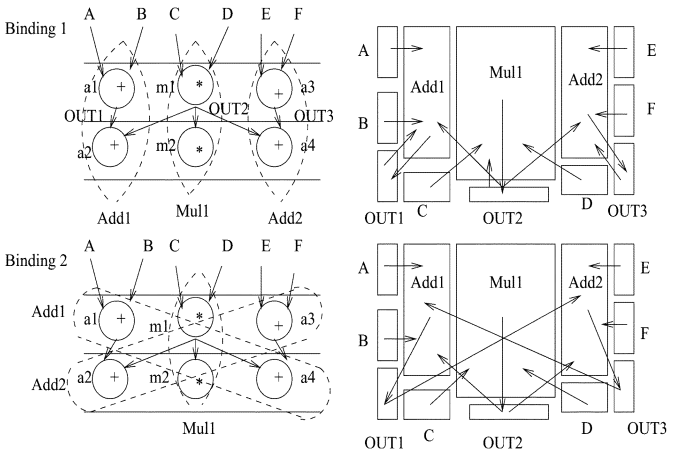

Fig. 1. Motivation: Simultaneous resource binding and floorplanning.

For a given initial resource binding, its corresponding floorplan is used to extract the net-lengths in order to compute the net-delay and net-power. This allows for improving the existing resource binding, because the net-power is considered as part of the overall power. The net-delay is taken into account to ensure that the synthesis of the resources is successful for the desired clock duration.

Fig. 1 illustrates the basic motivation behind simultaneous floorplanning and resource binding for low power. A scheduled DFG is shown here with two possible resource bindings. In the first case, shown on the top of the figure, operations a1 and a2 (of type ADD) are bound together and a3 and a4 are bound together. A different binding is shown on the bottom of the figure. For each binding, the resulting nets and floorplans are shown in the right-hand side of the figure. In the second case, there are two long nets going from registers OUT1 and OUT3 to ADD2 and ADD1, respectively. This potentially results in higher power since the net-power is increased. In addition the net-delays are higher in the second case. Given a clock duration of T, all operations in a consecutive register to register path must be performed within this clock duration. In the second case with longer nets, ADD1 and ADD2 have a smaller time to perform their additions and must consume more power. Intuitively, the first binding is better when considering the net effects.

B. Traditional Approach and Overall Design Flow

Many researchers have used low-level information for more accurate high-level synthesis and resource binding [10], [12], [16]–[18]. In [18], floorplanning information are used to provide rough estimates of the net-delay. Given an initial resource binding and floorplan the following steps are performed in the low-power resource binding and floorplanning:

1) extract the net-lengths from the current floorplan;
2) compute the power and delay of each net;
3) improve the existing resource binding with the provided net-power and net-delay information;
4) generate the floorplan of the new resource binding;
5) go to step 1 if convergence does not occur.

Extracting the net-lengths from the floorplan at step 1, enables computing the delay and power of the nets. Then at step 3, the current binding is improved for lower power using more accurate delay and power information. The improvement is by doing a set of moves. At each move, a new binding with smaller power is generated. This is done by traversing the DFG one clock at a time. At each clock, potential exchanges of the bindings of the same-type operations in that clock are considered. The set of exchanges that results in smaller overall power are performed in that clock to generate a new binding. When computing the overall power, the power of the registers, resources and the nets are added. The power of the registers are assumed to be proportional to their switching activities. The power of the nets are expressed as a function of their lengths which is extracted from the floorplan. For each module there is a tradeoff curve between its delay and power reflecting different ways that it can be synthesized. The power of each module is determined from this tradeoff curve based on its delay. The delay of the module is determined based on the delay of its incoming/outgoing interconnects and the duration of the clock. Once all the clock steps are traversed, the final binding is floorplanned in step 4. The above procedure iterates until the generated binding converges. The overall design flow, that takes advantage of this iterative floorplanning and resource binding is illustrated in Fig. 2. The scheduled DFG along with resource and register constraints are taken as input. Scheduling also decides a target clock duration (T). This is followed by iterative floorplanning and resource binding.

For a given resource binding, based on the net delays of each register to register path, the corresponding module on that path has a specific *delay budget* that it must be synthesized in, to satisfy the clock duration. The delay budget of each module, critically depends on the nets connected to its inputs and outputs.

As an example in Fig. 1, for the resource binding in the bottom, ADD1 has a long incoming net in clock 2. At the same time, ADD1 has a limited amount of time to get its input through the long incoming net, perform the addition and provide the output, which goes through the delay of its outgoing net. All of these should finish within T. Therefore, the delay budget of ADD1 is the maximum amount of time it is assigned, to do its operations based on its incoming/outgoing nets and T.

The key issue is the estimation accuracy of the net information from the floorplan. Although the floorplan is available, the exact lengths of the nets are not known. Typically, global routing is performed once the iterative resource binding and floorplanning has finished, to calculate the exact net-lengths as shown in Fig. 2. The existing approaches do not perform floorplanning and routing iteratively due to high run time complexity. After global routing, accurate net delay information are available which enables the computation of the actual delay budgets of all the modules. These functional modules are then synthesized with the generated delay budgets using popular synthesis tools. If all modules successfully get synthesized, the binding successfully passes synthesis.

C. Shortcomings of the Conventional Approaches

The traditional approaches to low-power resource binding and floorplanning, somehow estimate the net-length from the floorplan, which results in calculating the net-delay and

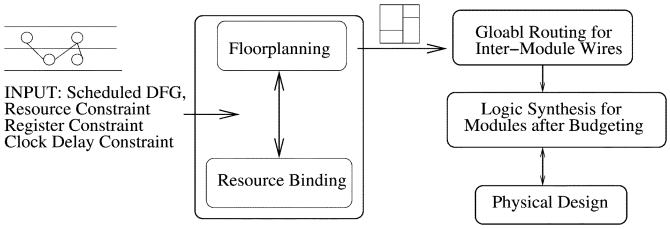

Fig. 2. Integrating floorplanning and resource binding.

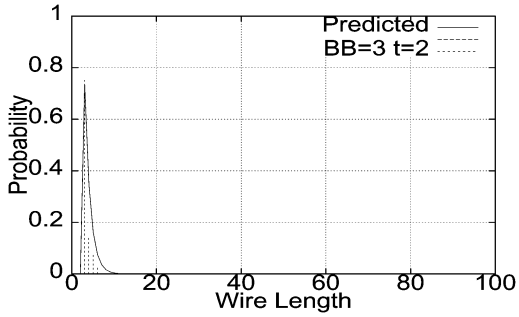

Fig. 3. Net-Length density function for half-perimeter bounding box: 3; net terminal count: 2.

net-power. Accurate estimation of these parameters is crucial during optimization. On one hand the net-power contributes to the overall power and inaccurate calculation of net-power results in inaccurate evaluation of the cost function during optimization. On the other hand the net-delay determines the delay budget of the modules which determines a module's power.

Inaccurate modeling of the net-length can have severe effects on the overall design quality and design closure. For example if all the net-lengths are modeled by their half-perimeter bounding box, there will be high chances of failing synthesis as shown in our experiments. Such a situation occurs when many nets are routed with lengths higher than their half-perimeter bounding box. This in turn occurs if the design is heavily connected, and contains congestion hot-spots. On the other hand, the net-lengths can also be modeled using worst case values. In this case, the post-routing net-length is always less than the estimate. However, this leads to a situation of false alarm where we predict a design to not satisfy the delay constraint, while in reality it does satisfy the constraint. Such an approach is an overkill. Accurate net-length cannot be determined until routing has been performed. One exhaustive approach is to run a router every time a move needs to be evaluated. But this would be computationally very expensive.

These shortcomings mainly arise from estimating the net-lengths with fixed values. In order to address the issue, in this work, we present a probabilistic approach to performing floorplanning and resource binding simultaneously. We consider the net-length as a random variable and model its probability density function. This means that a range of potential lengths together with their associated probabilities are considered for each net, rather than a fixed estimate. This probabilistic model is incorporated in a novel technique that optimizes the power probabilistically in this problem. Our approach is also iterative and move-based similar to the conventional approach of [18]. In Section III, we present the probabilistic model for the net-length.

III. Probabilistic Net-Length Model

The key to the success of any probabilistic approach is accurate models that capture the distribution of the cost function. We have developed empirical models to estimate the distribution of net-lengths in the post placement/floorplanning and prerouting stage [3]. The model was formalized by looking at empirical correlations between various parameters in the layout before and after routing. The model is briefly presented in this section. Given a placed netlist, our model estimates the probability for a given net to be routed with a certain length after routing. The proposed model has the following properties:

1) it is parameterizable for different routing tools or different routing modes of a certain tool;
2) the parameters of our model are determined by the benchmark type and router constraints and can be computed very fast.

In this model, we assign the same probability distribution to all the nets that have the same half-perimeter bounding box and number of terminals. We did extensive experimentation on standard benchmarks (high-level MediaBench [5] and IBM[1]) using commercial and academic tools (Cadence and Labyrinth [13]). We collected many sample distributions for the nets with the same half-perimeter bounding box and number of terminals and obtained a general form for modeling the distribution of any net. We modeled the probability density function as a curve that rises and falls after reaching a certain peak. Fig. 3 illustrates a sample plot of the actual and predicted values for a two-terminal net with bounding box of 3. With this form of density function we are particularly interested in modeling the length that has the maximum probability for each net and its associated probability. We refer to these as PeakLoc and Peak respectively. We investigated the effects of the empirical parameters in the system such as routing-grid granularity, capacity of the routing channels and number of available metal layers. We finally obtained empirical equations for Peak and PeakLoc. The model is outlined as follows:

$$P(x) = \begin{cases} 0, & x \leq \mathrm{BB} \\ \left(\dfrac{\mathrm{Peak}}{\mathrm{PeakLoc}-\mathrm{BB}}\right)(x-\mathrm{BB}), & \mathrm{BB} < x < \mathrm{PeakLoc} \\ \mathrm{Peak} \times e^{-l(x-\mathrm{PeakLoc})}, & x \geq \mathrm{PeakLoc}. \end{cases} \quad (1)$$

Here $p(x)$ is the probability for a net to have length x. The unit for length is in terms of the number of routing grids. The probability density function is modeled as an increasing ramp for lengths between half-perimeter bounding box (BB) to the peak location (PeakLoc) and after that it follows an exponentially decaying trend (1). The model for Peak and PeakLoc considers empirical parameters and is presented as follows:

$$\mathrm{Peak} = Ke^{-k1\times(\mathrm{BB}/C)+k2\times(\mathrm{BB}/G)+k3\times\mathrm{BB}\times n}t^{k4} \quad (2)$$

$$\mathrm{PeakLoc} = a1 \times \mathrm{BB} + a2 \times t + a3. \quad (3)$$

In (2) and (3), t is the number of terminals in the pertinent net and n is the total number of nets in the design. Also K, k_1,

[1][Online] Available: http://er.cs.ucla.edu/benchmarks/ibm place2.

k_2, k_3, k_4, a_1, a_2, and a_3 are parameterizable constants. BB is the half-perimeter bounding box of the considered net. G is the grid dimension of the layout and C is the routing capacity of each edge of the routing grid. The parameter Peak models the peak probability and PeakLoc estimates the net length that has this peak probability. The final unknown in (1) is the parameter l which captures the rate of decay of the exponential region of the density function. This is computed using the fact that overall area under this density function curve is 1.

This is a parameterizable model for any routing tool. In the model we used the Labyrinth router [13] and Cadence W-Route to validate our claims. In this paper, we use the parameters for Labyrinth, which are: $K = 122.56, k1 = 0.0019, k2 = 0.5358, k3 = 0.000\,059, k4 = -0.5, a1 = 1, a2 \approx 0, a3 \approx 0$.

The main purpose of this paper is to propose probabilistic floorplanning and resource binding. In general this can be shown using other models for net-length probability distribution as well. However, we are more interested to use our empirical model [3].

IV. Probabilistic Low-Power Resource Binding and Floorplanning

In this section, initially we define the probabilistic version of the problem. Then we explain how the probability distribution of the power of the design is calculated for a given resource binding, based on the probabilistic net-length model. This is incorporated in a probabilistic algorithm, which is similar to the deterministic one that iteratively optimizes the resource binding.

A. Probabilistic Problem Definition and Motivation

Here all the net-lengths are assumed to be random variables having probability distributions as explained in Section III. This results in the net-delays, net-powers, and consequently, the delay budgets and power dissipation of the modules to be probability distributions as well. Unlike the deterministic approach that only fixed values for the power and delay budgets of each module is calculated, here a range of possibilities together with their corresponding probabilities are considered. The probabilistic version of the low-power simultaneous resource binding and floorplanning problem is defined as below:

Given an initial binding for a scheduled DFG with a clock duration of T, together with its corresponding floorplan, and a delay violation constraint denoted by D_{cons}, and a power constraint denoted by P_{cons}, improve the resource binding such that:

1) the probability of failing synthesis for each module is less than D_{cons};
2) the probability of violating P_{cons}, the power constraint, is minimized.

Fig. 4 elaborates the above objectives. The delay budget of each module has a density function such as shown in Fig. 4(a), because of the delay distributions of the nets. In practice, for each module a minimum delay value is known, below which it cannot be synthesized. Such information is provided by the library. The probability of failing synthesis for a module is the probability of having a delay budget less than the minimum synthesizable delay budget for that module. This is shown as the shaded area in Fig. 4(a). The first optimization objective is that this shaded area for all the modules be less than D_{cons}.

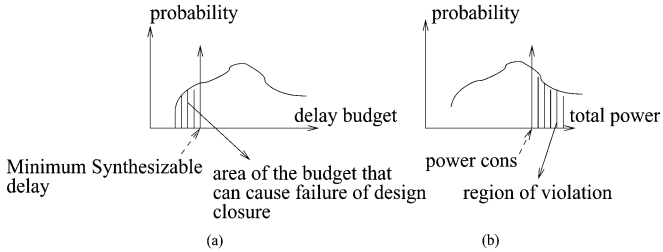

Fig. 4. Delay and power violation constraints.

The second optimization objective is to minimize probability of violating P_{cons}. In the probabilistic approach, a power distribution for the entire design is obtained for a given resource binding, such as in Fig. 4(b). The probability of failing power constraint is calculated from power distribution of Fig. 4 based on power constraint. This corresponds to the shaded area. The second goal is to minimize this shaded area.

Minimizing the probability of failing P_{cons} addresses the shortcoming of the deterministic approach that minimizes the power based on fixed estimates, which might be pessimistic or optimistic. On the other hand defining an input delay violation probability, is equivalent to a maximum risk of failing synthesis that the designer defines. If the designer is not willing to risk, the delay violation constraint is set to be very small. This in turn results in a resource binding with a large power.

The algorithm for the probabilistic version of the problem is similar to the deterministic one. It's a move-based algorithm that improves an existing binding using the floorplan information. However, the probability distribution of all the quantities are estimated and improvement is defined as minimizing the probability of failing power constraint. For a given binding, computing power distribution of the design is explained next.

B. Probabilistic Power Estimation

Given a resource binding, in this section we explain how the probability density function (pdf) of the design's power is calculated using the pdf of the net-lengths. The design's power comprises of the power of registers, nets and modules. Each of these will be explained next.

1) Register Power: Fig. 5(a) shows a scheduled DFG that has been bound to three resources and six registers. The variables that are bound to registers decide the corresponding switching activity. The switching activity is proportional to an average number of transitions from 0 to 1 or *vice versa* of the register content. We assume the register power to be a fixed deterministic value proportional to the switching activity of each register.

2) Net Power: The net power is expressed as a function of the net-lengths: $\sum_{\forall\ \text{net}_i}(0.5 \times V_{\text{dd}}^2 \times \text{SwitchCap}_i)$ where $\text{SwitchCap}_i = \text{length}(i) \times c \times \text{switchingActivity}$. Here c is the capacitance per unit length. Similarly, the switching activity is caused by the data transitions on the net, which in turn is decided by the register and functional module that is sourcing the net. The pdf of net-length defines potential lengths that a net can have together with their associated probabilities.

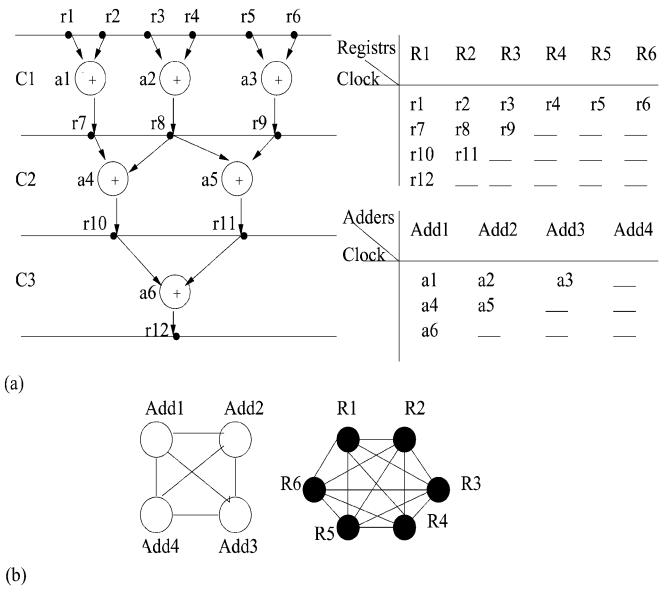

Fig. 5. Construction of exchange graph. (a) Binding of operations and registers. (b) Maximal matching for resource exchange at clock C1.

Each net-length also has an associated power that is calculated using the above formula. Therefore, the pdf of the net power is calculated using the pdf of its length.

3) Functional Module Power: Power of each functional module is determined by its delay budget. Therefore, we initially explain how the delay budget is calculated. Initially, we explain these for fixed net-lengths, and the discussion is then expanded to the probabilistic version, assuming net-length pdfs.

Fig. 6 illustrates a module with the operations bound to it. For each operation, the edges that provide it with data inputs are considered to determine the nets that source the datum. This information is determined from the floorplan. We also consider the nets at the output of each operation and determine their lengths from the floorplan. The delay budget of an operation, illustrated in left side of Fig. 6, is calculated as

$$R - \max(\text{delay (inNets)}) - \max(\text{delay (outNets)}) \quad (4)$$

where R equals the number of clock cycles it needs to do an operation multiplied by the clock duration (T). Note that delay of each net is a function of its length and is roughly estimated here as the multiplication of the *RC* of the unit length by the overall length.

The delay budget of the module is the minimum of the delay budgets of all operations bound on the module and is defined as follows:

$$\text{modDelayBud} = \text{Min}_{\forall\ op_i \in \text{module}}(\text{delayBudget}(i)). \quad (5)$$

For a set of operations bound to a functional module, each operation has a delay budget which is obtained from (4). For a resource that performs all these operations, the delay budget is defined as the minimum of its operations delay budgets. This is because, if any delay budget higher than the minimum is chosen, the module will fail to finish the operation that had the smallest delay budget on time. This results in violating the input/output relationship.

Using this delay budget of each functional module, calculation of the power dissipation of the module is explained next.

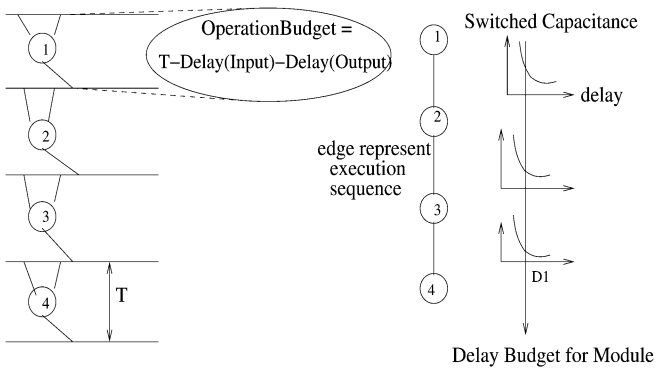

Fig. 6. Calculating the module power.

Resource binding specifies a sequence in which the operations are executed on each module. In Fig. 6 assume operations 1, 2, 3 and 4 are bound on the same resource and are performed in the stated order. This sequence determines the switching activity of the functional modules. Typically, the binding sequence is represented as a path [8] (Fig. 6 right side). Each edge on the path is associated with a tradeoff curve between a module's delay budget and amount of switched capacitance between the two operations that the edge is connecting. We assume that for each edge in this path we know the delay vs switched capacitance tradeoff curve (as illustrated in Fig. 6 right side). Such a tradeoff curve is obtained in a preprocessing step similar to [2] and [11]. For different delay budgets that a module can be synthesized in, a tradeoff curve exists between switching capacitance and module's delay budget at each binding edge. Therefore, for a fixed delay budget of a module, each edge on the path is characterized with a cost which represents the amount of switched capacitance that occurs in the module when the two operations at the edge end points are executed one after the other. The summation of these switched capacitances is proportional to the module power [11]. For example in Fig. 6 at delay value $D1$ there is a corresponding switched capacitance for each of the 3 tradeoff curves. Summation of these switched capacitances is proportional to the module power for delay budget of $D1$. The total module power is then calculated as $0.5\ V_{\text{dd}}^2 \times \text{switchedCap}$.

So far we explained the calculation of the module power deterministically for fixed net-length values. Next we expand the discussion assuming net-length pdfs.

In the probabilistic case the net-length pdf is modeled as explained in Section III. The net-length pdf results in a net-delay pdf since delay of each net is expressed as a function of its length. The pdf of net-delay is consequently calculated. The net-delay pdfs result in delay budget pdf for each operation [obtained from (4)] and delay budget pdf for each module [obtained from (5)]. In (4) and (5), subtractions and min are done probabilistically on random variables.

Once the delay budget pdf of each functional module is obtained, next the power distribution of functional modules are calculated. To find the power distribution of a module, recall that summation of switched capacitances of the operations bound on the module was calculated (Fig. 6). Now since the delay

budget is a distribution, we will end up with a distribution in switched capacitance for each edge in the binding path of Fig. 6. The power distribution of the module is now proportional to the probabilistic summation of these switched capacitance distributions for each binding edge.

Once the fixed register powers and distributions of net and functional module powers are calculated, the overall power distribution of the DFG is calculated as the probabilistic summation of these three powers.

Note in calculation of the total power, the effect of multiplexors have been ignored. Also we assume an unshared interconnect architecture, hence there are no buses in the design. Please note that these are merely simplifying assumptions. Our philosophy of probabilistic estimation and optimization would hold even if these assumptions are relaxed. Also note that there would be covariance between the net and functional module powers since the net-delay determines the module power distribution. While evaluating the total power distribution such covariance can be taken into account. This is by considering the covariance in the equations when doing probabilistic summation and probabilistic min. Such covariance has been ignored here however it is important to note that ignoring covariances results in an overestimation of module power and delay. Therefore, meeting the design constraints is not underestimated. The reported experiments show the effectiveness of this probabilistic paradigm after routing and synthesis stages.

C. Probabilistic Algorithm

In the previous subsection, we explained for a given resource binding and floorplan, how the pdf of the power of the design is calculated. Next we explain the details of the probabilistic low-power simultaneous resource binding and floorplanning algorithm that relies on the probabilistic power computation of Section IV-B. Details of the probabilistic algorithm is explained next. The proposed algorithm is a move-based one, similar to [18], that improves upon a current resource binding based on the floorplan information.

1) Algorithm Details: Improvement in the initial binding is done by iterative moves from one valid binding to another that is more probable to meet the power constraint while it is meeting the delay violation constraint. This is done until no substantial improvement is obtained. At each point, for a given resource binding and floorplan, the pdf of the power of the design is calculated as explained in Section IV-B.

The algorithm comprises of a local and a global phase. In the local phase, improving the binding at a specific clock cycle is considered. However, in the global phase, the algorithm traverses the DFG in many rounds. In a local step, considering a clock step, potential exchanges of the bindings of the operations of the same type (exchanges of functional module bindings) are considered. Potential exchanges between variable bindings which provide inputs to the operations in that clock step (exchanges of register bindings) are also considered. As an example consider Fig. 5(a). Here, at clock C1, three additions and their bindings are available. This clock also has six input variables whose bindings can also be exchanged. The exchange of the bindings of any pair of these additions/variables will result in a new valid binding. A move comprises of either exchanging two operations/variables or moving an operation/variable to a free module/register. Next these two phases are explained in detail.

Local Optimization: In the local phase exchanging the bindings of the operations of the same type at a certain clock and exchanging the variable bindings feeding these operations are considered. During the local phase, once a set of moves are done, the state of the DFG is recalculated for the new moves and if D_{cons} is violated, the moves are undone. After the local optimization at a clock step, a new binding with a better total power distribution of the design in terms of meeting the power constraint is obtained. Local optimization is done one clock step at a time iteratively until no further improvement is obtained.

At each clock step, the algorithm builds a weighted *exchange* graph for variables and operations separately. As illustrated in Fig. 5(b) such a graph has been initialized for clock step C1. Each node represents a functional module/register. An edge between two nodes implies that the corresponding operations/variables can potentially be exchanged. Hence, functional modules with different types will not have a common edge. Note that the nodes could represent free functional modules/registers too. Each edge has a cost which implies the amount of gain in the global cost function. The idea is to identify the maximum number of independent edges in the corresponding graphs such that the total gain is maximized. This is solved using maximal matching [7].

The weight of an edge between nodes i and j is an indication of the amount of improvement in power if node i and node j are exchanged. More specifically, the edge weights can be of two types depending on how power improvement is defined. Fig. 7(a) illustrates the first type of edge weights. Assume at a clock step before considering any binding exchanges, the initial binding results in power distribution of the design to be as indicated in the figure. Assume two potential exchanges are possible resulting in power distributions of the design to be as solution A and solution B in the figure. Given a power constraint, the first type of edge weights are defined as follows. For an edge between nodes i and j, w_{ij} is the amount of improvement in the probability distribution of the design if the bindings of i and j are exchanged. For a potential move there is a probability of meeting the power constraint described as follows:

$$\text{Prob}(P_{\text{cons}}) = \int_{-\infty}^{P_{\text{cons}}} p(x)dx \tag{6}$$

where $p(x)$ is the probability distribution of the power of the design. In Fig. 7(a), this probability is the dashed area for solution A and is 0 for solution B and the original solution. This definition of edge costs sounds logical but can limit optimization. As an example assume a situation like Fig. 7(b) where the current binding is far away from meeting the power constraint. The two potential moves result in power distributions for solutions A and B as shown in the figure. Comparing with the power constraint, each potential move is not *strong* enough to meet the power constraint and all the edge weights will end up to be zero. Therefore, the second type of edge weights are defined as the improvement in the *expected value* of power distribution of the DFG. A potential move corresponds to a new power distribution of the DFG with a new expected value. If this expected value is

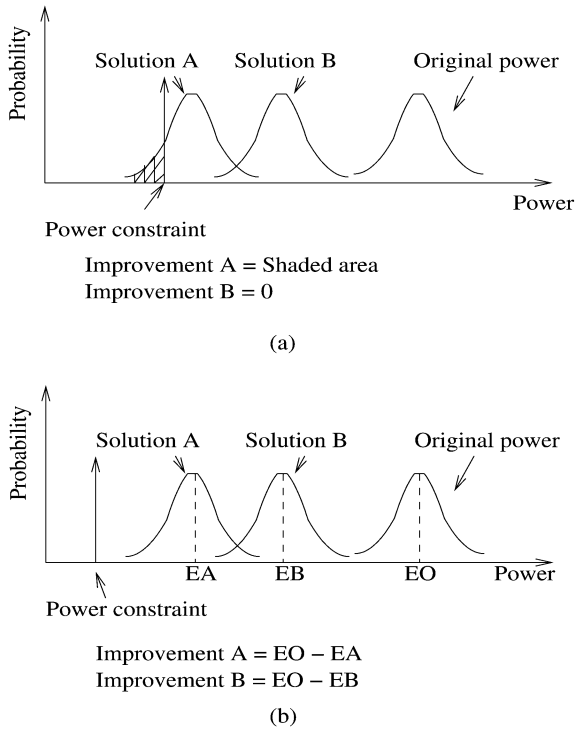

Fig. 7. Defining different edge types. (a) Defining edge type 1. (b) Defining edge type 2.

smaller than the original case, this potential move results in improvement and the edge weight is the difference between these two expected values as shown in Fig. 7(b).

Given the exchange graphs at a clock step, we consider the edge costs of type two if all the edge costs of the first type are zero. This implies that the current solution is far away from the power constraint and none of the moves can bring it close enough to meet the power constraint.

Since exchanging registers and modules of the same type cannot occur simultaneously, a separate exchange graph is constructed for each case. Maximal matching is done separately and depending on the set of moves that results in a better power distribution, register or module exchange is done. The solution with higher gain is given priority and the two sets of exchanges are made one after the other. Once a set of moves are done, if D_{cons} is violated, the moves are undone. These steps are the following.

1) Identify exchange graphs for registers and modules.
2) Calculate the maximal matching for all graphs and pick the set with greater gain.
3) Perform the set of moves for the selected graph.
4) Recalculate the state of the design and undo the move if it violates the delay violation constraint.
5) Calculate the maximal matching for the other graph and make the corresponding moves.
6) Recalculate the state of the design and undo the move if it violates the delay violation constraint.

Global Optimization: The local phase is applied iteratively, one clock step at a time in the global phase. After each iteration the new binding has a better probability of meeting the power constraint. This iteration is stopped until no significant improvement in power is possible. This corresponds to the situation

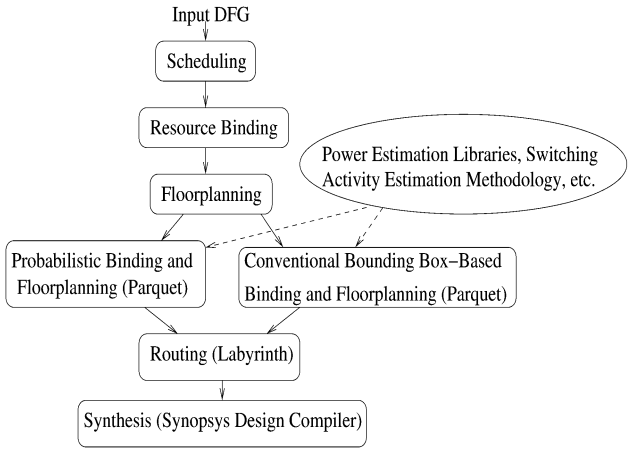

Fig. 8. Experimental flow.

where all the edge costs in the exchange graphs are 0 or that no solution can be found to meet the delay violation constraint.

Once the iteration stops, another round of floorplan is done using the new generated binding. New net-lengths, obtained from the floorplan, are used in next round of binding. This flow is iterated until convergence occurs.

V. Experimental Results

We validated our claims through extensive experimentation with the following flow: We consider traditional high level synthesis benchmarks and the MediaBench benchmarks [5] and schedule the corresponding DFGs into minimum possible clock steps. The latency of an adder is assumed to be one clock cycle and multiplier to be one clock cycles. This is followed by resource binding for functional and registers for minimum number of resources as in [1]. The generated register transfer level (RTL) design is floorplanned using Parquet [14]. The floorplanned RTL design is optimized for power using the proposed simultaneous resource binding and floorplanning methodology. The generated floorplanned RTL design is routed using the Labyrinth [13] global router. Using the accurate net-delay values (post global-routing), the functional modules are budgeted and synthesized for the budgeted delay constraint using Synopsys Design Compiler. Fig. 8 illustrates the experimental setup.

There are two approaches to performing simultaneous resource binding and floorplanning: the conventional approach and our proposed probabilistic methodology. The conventional approach is similar to [18] and uses half-perimeter bounding box or MST as the net-length estimate. The conventional approach is similar to the probabilistic one, only the power estimation is done deterministically since the net-length values are fixed. Solutions of both approaches are routed. Using the real net-delay values the modules are budgeted and synthesized using Synopsys Design Compiler.

Table I illustrates the experimental results on some MediaBench and traditional high-level synthesis benchmarks. Column 2 is T, the timing constraint, that was chosen very stringent. The number of functional modules on which the synthesis engine results in timing failure are reported in columns 4, 5, and 6 for different approaches. These are for probabilistic and

TABLE I
NUMBER OF UNSYNTHESIZED MODULES

Bench	T(ns)	#FU	Prob.	BBox	MST
fft1	2.3	30	0	3	2
fft2	2.1	27	1	2	2
fir	1.7	15	0	1	1
dct	2.3	20	0	3	2
jdmerge1	2.1	23	0	2	1
jdmerge2	1.8	13	3	4	4
jdmerge3	1.6	18	0	3	2
jdmerge4	1.9	21	0	1	1
motion2	1.6	28	1	2	1
motion3	1.6	26	2	2	2
ellipt	3.2	18	1	1	1
ecbnc4	1.0	12	1	1	1
noisest2	1.6	16	3	4	4

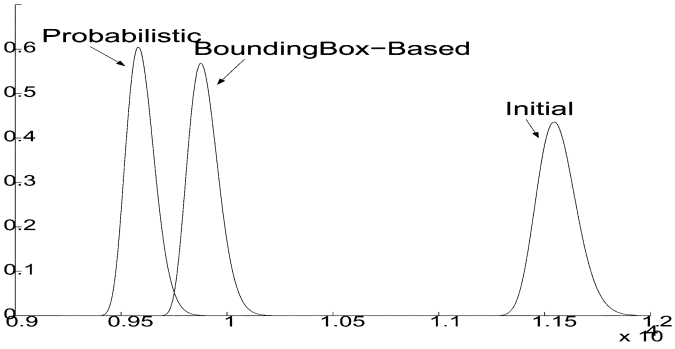

Fig. 9. Comparison of power density functions.

TABLE II
POST SYNTHESIS POWER

Bench	T(ns)	Init.	Prob.	BBox	MST
fft1	2.5	588.82	535.78	543.73	540.78
fft2	2.5	137.68	84.21	52.81	45.62
fir	2.5	57.01	49.37	50.84	50.84
dct	3.0	72.34	34.51	52.81	45.62
jdmerge1	2.5	70.52	79.71	50.84	64.82
jdmerge2	2.5	187.84	168.48	169.32	169.32
jdmerge3	2.5	63.67	52.51	59.42	57.66
jdmerge4	2.5	85.85	63.69	71.71	68.70
motion2	2.5	125.56	101.55	111.34	108.70
motion3	2.5	125.58	108.33	107.10	107.90
ellipt	4.0	105.85	81.95	96.23	82.34
ecbnc4	1.8	35.67	34.96	34.96	34.96
noisest2	2.5	106.25	95.39	98.89	95.87

TABLE III
RUN-TIME COMPARISON (SECONDS)

Bench	Prob.	BBox	MST
fft1	1510	1420	1481
fft2	713	497	513
fir	537	280	310
dct	312	240	246
jdmerge1	836	652	695
jdmerge2	846	804	824
jdmerge3	780	567	570
jdmerge4	590	459	498
motion2	1077	761	898
motion3	1724	1650	1700
ellipt	562	479	491
ecbnc4	117	115	117
noisest2	169	149	159

traditional approaches using half-parameter bounding box and MST, respectively. The total number of modules are reported in column 3. It can be seen that for most of the benchmarks the conventional approaches result in timing failure for many modules whereas our probabilistic approach gave a feasible design or smaller number of unsynthesized modules. Hence, our approach is better suited at achieving design closure. This was primarily happening because our strategy was considering the fact that the net could become longer than what they were estimated and accepted solutions which had a higher chance of meeting the clock constraint even in the presence of this variability. The conventional bounding box-based approach could not make such decisions and was always underestimating the net-length, hence overestimating the module delay budget. This directly resulted in larger number of unsynthesized modules. MST was slightly more successful than half-perimeter bounding box.

Fig. 9 shows typical power density functions in the designs generated by the conventional bounding box-based approach and our probabilistic approach. The distribution of the bounding-box based approach was obtained as follows. The bounding box-based approach generated a resource binding for which the net-length pdfs were calculated using the model of Section III. Consequently, the pdf of the power of the DFG was calculated for this final bounding box-based solution as explained in Section IV-B. In Fig. 9, in most of the cases, the power from the probabilistic approach was slightly better than the bounding box-based approach. Also the power distribution in the probabilistic case stays slightly before the bounding box-based approach. Both of the distributions are better than the distribution of the initial solution in which no optimization was performed.

Table II compares our probabilistic approach with conventional bounding box-based and MST-based approaches with respect to total power after routing and synthesis. In this case all designs passed synthesis. Here the second column reports the clock duration constraint which was set to be larger than in Table I. This is because in Table I, T was smaller and some modules were not synthesized after routing. Hence, no power could be reported. However in Table II, T is slightly increased so that both approaches result in design closure after routing and synthesis. The power of the probabilistic approach was as good as the conventional approach. The bounding-box based approach underestimates the net-lengths resulting in overestimation of delay budgets of the modules. Consequently, this results in underestimation of the module powers and inaccurate estimation of the DFG power and inaccurate optimization.

The novelty in our approach is in the higher chance of ending with a feasible design that is synthesizable without degradation in power when compared to the traditional case. The runtime of different approaches are compared in Table III. The probabilistic approach is the most time-consuming one since it needs to compute the total power of the DFG at each step probabilistically.

We could have provided a worst case estimate for wire-length in order to ensure that the timing constraint is always satisfied. We did experimentation for the same benchmarks having the conventional wire-length estimate as $5 \times$ the half-perimeter bounding box value. In such case, the conventional approach

was unable to generate a feasible solution satisfying the stringent timing constraint. Such an approach is an overkill since many of the times if the delay constraint is stringent, worst case estimation would ignore moves that violate the timing constraint (which in reality do not violate timing).

VI. Conclusion

A novel approach for low-power simultaneous resource binding with floorplanning is presented. Inaccuracy in net-length estimation results in failure of timing closure. Unlike traditional methods that half-perimeter bounding box of a net is taken as its fixed length estimate, in this paper the distribution of each net-length is modeled. Using these distributions, a novel approach is proposed to do resource binding. Each potential resource binding solution results in a power distribution of the DFG, which includes power of functional modules, registers and interconnects. Post-routing experiments show higher chance of successful synthesis. As part of the future work, more comprehensive power estimation, including power of the multiplexors, is planned.

References

[1] A. Davoodi and A. Srivastava, "Effective graph theoretic techniques for the generalized low power binding problem," in *Proc. Int. Symp. Low Power Electronics and Design*, Aug. 2003, pp. 152–157.

[2] ——, "Voltage scheduling under unpredictabilities: A risk management paradigm," in *Proc. Int. Symp. Low Power Electronics and Design*, Aug. 2003, pp. 302–305.

[3] A. Davoodi, V. Khandelwal, and A. Srivastava, "Empirical models for net-length probability distribution and applications," *Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 12, no. 10, pp. 1066–1075, Oct. 2004.

[4] A. Srivastava and M. Sarrafzadeh, "Predictability: Definition, analysis and optimization," in *Proc. Int. Conf. Computer Aided Design*, Nov. 2002, pp. 118–121.

[5] C. Lee, M. Potkonjak, and W. H. Mangione-Smith, "Mediabench: A tool for evaluating and synthesizing multimedia and communications systems," in *Proc. Int. Symp. Microarchitecture*, Dec. 1997, pp. 330–335.

[6] D. Liu and C. Svensson, "Power consumption estimation in CMOS VLSI chips," *IEEE J. Solid-State Circuits*, vol. 29, no. 10, pp. 663–670, Jun. 1994.

[7] G. Chartrand and O. R. Oellermann, *Applied and Algorithmic Graph Theory*. New York: McGraw-Hill, 1993.

[8] J. M. Chang and M. Pedram, "Low power register allocation and binding," in *Proc. Design Automation Conf.*, Jun. 1995, pp. 29–35.

[9] L. Kruse, E. Schmidt, G. Jochens, A. Stammermann, A. Schulz, E. Macii, and W. Nebel, "Estimation of lower and upper bounds on the power consumption from scheduled data flow graphs," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, pp. 3–14, Feb. 2001.

[10] 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.

[11] M. Pedram, "Low Power CMOS Digital Design," *IEEE J. Solid-State Circuits*, pp. 472–484, Apr. 1992.

[12] M. Xu and F. J. Kurdahi, "Layout-driven RTL binding techniques for high-level synthesis using accurate estimators," *ACM Trans. Des. Autom. Electron. Syst.*, pp. 312–343, Oct. 1997.

[13] R. Kastner, E. Bozorgzadeh, and M. Sarrafzadeh, "Predictable routing," in *Proc. Int. Conf. Computer Aided Design*, Nov. 2000, pp. 110–114.

[14] S. Adya and I. Markov, "Fixed-outline floorplanning through better local search," in *Proc. Int. Conf. Computer Aided Design*, 2001, pp. 321–334.

[15] S. Borkar *et al.*, "Parameter variations and impact on circuits and microarchitecture," in *Proc. Design Automation Conf.*, Jun. 2003, pp. 338–342.

[16] S. Park, K. Kim, H. Chang, J. Jeon, and K. Choi, "Backward-annotation of post-layout delay information into high-level synthesis process for performance optimization," in *Proc. Int. Conf. VLSI and CAD*, Oct. 1999, pp. 25–28.

[17] W. E. Dougherty and D. E. Thomas, "Unifying behavioral synthesis and physical design," in *Proc. Design Automation Conf.*, Jun. 2000, pp. 756–761.

[18] Y. M. Fang and D. F. Wong, "Simultaneous functional-unit binding and floorplanning," in *Proc. Int. Conf. Computer Aided Design*, Nov. 1994, pp. 317–321.

Azadeh Davoodi (S'02) received the B.S. degree in electrical engineering from University of Tehran, Tehran, Iran, in 1999. She is currently pursuing the Ph.D. degree at the University of Maryland, College Park.

Her research interests include low-power design automation for deep submicron VLSI designs, considering process variations, and uncertainties.

Ankur Srivastava (S'00–M'02) received the B.S. degree from Indian Institute of Technology, Delhi, India, in 1998, the M.S. degree from Northwestern University, Evanston, IL, in 2000, and the Ph.D. degree from the University of California at Los Angeles (UCLA), in 2002.

Since the fall of 2002, he has been an Assistant Professor at University of Maryland College Park. His main area of interest include all aspects of design automation including logic and high-level synthesis, low power issues and predictability. He is also working on low power issues pertaining to sensor networks.

Dr. Srivastava was awarded the Outstanding Ph.D. Award from the Computer Science Department of UCLA in 2002. He is a Member of the Association for Computing Machinery (ACM).