Journal of
Low Power Electronics
Vol. 1, 1–13, 2005

Early Quality Assessment for Low Power Behavioral Synthesis

Eren Kursun,[1,*] Rajarshi Mukherjee,[2] and Seda Ogrenci Memik[2]

[1] *Computer Science Department, University of California Los Angeles, CA 900, USA*
[2] *Department of Electrical Engineering and Computer Science, Northwestern University, Evanston, IL 60208-3118, USA*

(Received: 17 April 2005; Accepted: 1 December 2005)

Fast and effective exploration at the early stages of the design flow can yield significant improvement in the quality of the design and substantial reduction in design time. In this paper, we present an efficient technique to evaluate the power dissipation of scheduled Data Flow Graphs (DFGs). Scheduling dictates the compatibility of operations with respect to their assignments to functional units. Generally for scheduled DFGs, this relation is captured in the form of a comparability graph. As a consequence, the topology of the comparability graph determines the solution space available to the subsequent binding stage. In this work, our main contribution is a technique to assess the inherent flexibility of the schedules we start with. We developed early evaluation metrics in order to assess the degree of flexibility inherent in an initial schedule that will eventually affect the quality of the binding solution. Every schedule is associated with a compatibility graph that represents the conflicts and compatibilities among operations with respect to possible binding decisions. Our metric based evaluation technique is based on several properties (such as edge connectivity, edge weight distribution, etc.) of these compatibility graphs. These metrics essentially reflect the amount of freedom that is provided to the binding stage, which enables early assessment and relative comparison of different possible schedules without actually performing the resource-binding step. Our experimental framework integrates scheduling, early metric-based power evaluation, low power binding and power driven iterative rescheduling stages. The correlation between early evaluation and the power measurements after binding is as high as 0.95 and greater than 0.75 for majority of test cases. Experimental results on DFGs from MediaBench suite demonstrate the fact that metric evaluation is on average 42.6 times faster than performing optimal binding and iterative power improvement. Our results show that low power schedule selection is fast and effective. On average, the schedules selected by metric evaluation have 43% less power dissipation than schedules with iterative power improvement, based on a study set of 320 schedules. We also examined the thermal profile of the corresponding solutions. We observed that schedules selected with our metric evaluation technique have on average 12 °C lower temperature, and the maximum on-chip temperatures are lower by 18 °C compared to the overall average of all schedules. These thermal profiles are obtained using a functional unit-level thermal simulator after block-level floorplanning.

Keywords: Power Optimization, Scheduling, Binding, Metric-Based Evaluation, Temperature Optimization, High-Level Synthesis.

1. INTRODUCTION

Power characteristics of today's electronic devices are creating significant challenges. Number of on-chip transistors is continually increasing, clock frequencies scale at dramatic rates and supply voltage reduction is expected to slow down in the near future. This is leading to ever increasing power densities. Furthermore, the cost of packaging and cooling devices is growing at exponential rates, drawing the attention of the industry as well as the academic research community to power and thermal optimization. Power has already become one of the most important design constraints today. As a result, power optimization techniques have been proposed at all stages and abstraction levels of the design flow. Estimation and modeling of performance characteristics (power consumption and thermal behavior are no exception to this) can be performed accurately only after sufficient levels of details on the physical

[*] Author to whom correspondence should be addressed.
Email: kursun@cs.ucla.edu

1546-1998/2005/1/001/013
doi:10.1166/jolpe.2005.028

properties of a circuit are obtained. However, it is important to note that early design decisions have a larger impact on the quality of the final implementation. In this work we focus on power optimization for behavioral synthesis stage.

Power driven high-level synthesis methodologies commonly incorporate a scheduler followed by a resource binding stage aiming at minimizing the total switching activity.[1,4] This can be further supported by a rescheduling step that modifies the initial scheduling solution to improve the power consumption.[2] However, the initial schedule impacts the subsequent binding step significantly, since the schedule dictates the compatibility among operations. Therefore, the initial scheduling step determines the extent of the available search space for the binding as well as the rescheduling phase. As a consequence, it would be highly desirable to assess the power efficiency of a given schedule. If a systematic method to evaluate schedules (and subsequently select the most power efficient schedule) was in place, it would benefit the following resource binding stage. Also, this would potentially eliminate the need for iterative rescheduling later on. However, due to lack of such systematic criteria to evaluate an initial schedule, iterative power optimization techniques are commonly employed in practice. Furthermore, most iterative rescheduling/rebinding techniques are computationally expensive.

In this work we develop a technique to identify initial schedules that will likely lead to final implementations with lower power consumption. Our approach is based on formulation of metrics to evaluate schedules for their potential impact on the subsequent resource binding stage. These metrics are used to compare alternative schedules and identify the best initial solution to be provided to the remainder of the high-level synthesis flow. The proposed metrics exhibit high correlation with the post-binding power dissipation and power-driven rescheduling improvement, thus enabling effective prediction of the power dissipation without actually performing resource binding. Utilization of these metrics in initial schedule selection yields reduction in the design effort required for the subsequent synthesis steps such as binding and iterative improvements (e.g., rebinding and rescheduling). Our experiments illustrate that in some cases, it even eliminates the need of iterative rescheduling and rebinding.

The following design flow is implemented for experimental validation: Data Flow Graphs from algorithm specifications in *C* are extracted using SUIF.[13] Alternative schedules are generated for the extracted DFGs such that the resource and timing constraints are met for each schedule. Metric based analysis is then performed on these schedules. This analysis provides an opportunity for quick and early assessment of power for a range of alternative schedules without executing the resource binding for each of them. Also, our results demonstrate that if an initial schedule is selected according to our metrics we can perform a single pass of low power binding without needing to execute iterative re-scheduling/re-binding and still obtain low power consumption. The initial schedule is important due its impact on the subsequent binding stage. Hence, it is desirable to provide the best schedule with the best prospect for a low power solution to the design flow.

There can be two approaches to this search. In the first case, the search space for possible schedules under a certain resource and/or latency constraint can be explored in an exhaustive manner. During this exploration the evaluation metrics can be used to identify potential candidates for a low power solution. Naturally, this search should not be overwhelmingly costly, hence, potentially overshadowing the benefit of the quick evaluation. We have formulated the scheduling problem using Binary Decision Diagrams (BDDs). Within this initial design space, which contains all schedules under a given latency and resource constraint, we have identified a subset of schedules that provide a diverse representation of the design space. Our criterion for this has been the distribution of time slack within each schedule. We have identified schedules that possess a varying amount of slack on a given operation, i.e., the mobility of an operation is different in each schedule. In this manner we have identified schedules that would present different amounts of flexibility with respect to different operations. Then, we applied our metric-based analysis onto this subset of schedules for a given DFG. Our experimental results have been generated by obtaining a pool of initial schedules using this approach.

The second approach is to compare schedules that were created by different heuristic scheduling algorithms or under different resource and/or latency constraints. This would be helpful in comparing the ability of different scheduling approaches in producing low power schedules. Similarly, comparing different resource constraints will enable the design space exploration along the resource dimension as well. Such a subset of schedules can be post-processed by our metrics and a good initial schedule can be identified for a given specification. Subsequent to the optimal binding[1] and iterative rescheduling-rebinding, we compare the correlation of the metric values with final power dissipation for each schedule respectively. This correlation was found to be as high as 0.95 and higher than 0.75 for most test cases.

Power dissipation has a close relationship with the on-chip temperature profile. Increasing logic density along with continuous technology scaling cause dramatic increases in power density. This manifests itself in rising on-chip temperatures, which calls for effective control on heat dissipation at each step of the design flow. Temperature has a significant impact on circuit performance. Increase in temperature has an adverse effect on carrier mobility, hence, switching speed of transistors. Interconnect resistance increases with temperature as well, the

[1] The binding is optimal with respect to the total switching activity.

variation in interconnect resistances causes timing errors that can compromise proper functionality. Circuit reliability is also heavily impacted by temperature. Regions on a chip that generate excessive amounts of heat are referred to as "hotspots." Hotspots can jeopardize correct execution by causing transient as well as permanent faults. Even if excessive heat does not lead to spontaneous damage, it accelerates electromigration, which can lead to permanent damage in the long run. As a result, thermal problems are becoming increasingly important. Thermally efficient architectures are more desirable than ever before. In this context, we also analyze potential thermal benefits of our early evaluation. We studied the correlation between our metric-based schedule selection and the temperatures reached by functional units after binding in each design. We found that schedules selected according to our metric evaluation technique have 12 °C lower temperature on average compared to alternative schedules generated under the same resource and latency constraint.

The rest of the paper is organized as follows: Section 2 describes the related work on low power binding and iterative rescheduling problem. Section 3 presents the design flow. Section 4 introduces the metrics and the representative implementation of a rescheduling algorithm is described in Section 5. Experimental results for power efficiency and thermal efficiency are reported in Section 6. Concluding remarks are given in Section 7.

2. PRELIMINARIES

2.1. Related Work

Power optimization has gained significant importance in the past decade; as a result, there has been a wide range of optimization and estimation techniques ranging from circuit, layout, logic, behavioral, and architectural levels.[17–20] Although power optimization can be achieved at various levels, we will discuss the past contributions in behavioral synthesis in the following.

2.1.1. Low Power Scheduling Problem

Power efficient scheduling has been studied extensively since a good initial schedule provides a much more efficient design space to the rest of the design flow. Scheduling for minimum switching activity is NP-Complete. Raghunathan and Jha[21] proposed an ILP formulation and heuristic techniques for scheduling for low power. They also introduced resource allocation, clock selection and binding heuristics. Lin et al.[22] studied a similar ILP model and heuristic for variable voltage scheduling problem. Su et al.[23] suggest a scheduling technique to reduce the switching activities of address access during scheduling. Similarly, using list scheduling Kim et al.[24] presented techniques to combine retiming with operand sharing scheduling. More recently, scheduling problem for low power design was formulated as a Traveling Salesman Problem (TSP) and the heuristics for TSP were used for the case of a single functional unit.[26] Shao et al.[25] proposed two heuristic techniques based on weighted bipartite matching. In Ref. [42] Tang et. al. investigate a power optimization approach for early stages of behavioral synthesis. They use an integer linear programming model to reduce the energy dissipation.

2.1.2. Power Efficient Binding Problem

The power efficient binding problem was optimally solved using the max-cost flow[1] and matching techniques.[4] Our framework employs the max-cost flow formulation for low power binding. In Ref. [1], compatibility graphs are generated from the scheduling information. Operations are represented as nodes in the compatibility graph. Between every operation pair u and v that can potentially be executed on the same resource in succession, there is a directed edge (u, v) connecting them. These operations are called compatible operations.

Optimal solution to the low-power binding problem is computed by executing max-cost flow algorithm on the network flow graph, constructed from this compatibility graph.[1] The max-cost flow algorithm finds a maximum cost set of cliques that cover the graph. Negating the cost of each arc in the network and solving the min-cost flow problem is a practical way of doing this. Hence, we will use min-cost flow problem formulation term instead in the remainder of the discussion. The flow values are all 1 on each path and the cost of each path is the power consumption of the corresponding resource. A minimum cost solution minimizes the overall switching activity. Details of the formulation are omitted here for brevity. Chang et al.[1] present an in-depth discussion of the max-cost flow methodology for the low-power binding problem.

Most synthesis methodologies utilize iterative refinement to improve the final design. Primary reason for this is the fact that in most of the cases an accurate estimate of design quality is not available at the initial stages. Therefore as soon as more accurate information becomes available, the initial design decision is further refined. An example of this design paradigm is the methodology of Layout Driven Logic Synthesis. The synthesis decisions are improved when more accurate wirelength and wire delay estimates are available. Rescheduling-rebinding exploits the same paradigm as well. The goal is to modify the existing schedule and resource binding solution by rescheduling-rebinding the operations to reduce the power dissipation. The related work on iterative power improvement techniques, and network flow based formulation can be found in Refs. [1–6, 14–16, 43].

In Ref. [4] Kruse et al. present a power estimation of data path resources from scheduled data flow graphs with a given input data stream, using Lagrange multipliers for resource constraints. Although our main goal or early scheduling assessment is quite similar to this technique, we are more interested in ordering the schedules according to their power dissipation than estimating the power dissipation value. Hence achieve faster initial design space exploration, and our post-binding power dissipation values are in close proximity of the minimum power solutions.

The lack of solid criteria required for choosing initial scheduling can be overcome by picking any initial schedule and improving the overall power consumption by modifying binding result iteratively. However, even local changes such as rescheduling an operation n from schedule step s_i to s_j, changes the data transfers, which invalidates some operation compatibilities and hence, the binding solution. One way to solve this problem is to perform the max-cost flow algorithm after each iteration step to update the binding. Nevertheless, this is not desirable because of its high computational complexity. Lyuh et al.[2] proposed a method that addresses this complexity problem. Their two-step iterative algorithm for bus binding can be extended to other components as well. In the first step the max-cost flow computation is performed and the algorithm retains the previous binding solution as much as possible. This avoids the unnecessary computation but still yields an optimal binding at the end of each iteration step. In the second stage the algorithm finds the negative cost cycles in the residual graph of the flow, which refines the solution of the first step.

The algorithm offers both running time improvement and power efficiency. However, the rectification at the end of each iteration step to validate the flow for the current schedule might still be time consuming. An alternative to this might be improving the running time of the iteration step and proceed until no power improvement can be attained by rescheduling-rebinding moves. In this paper we use such an iterative power improvement method, where we reschedule and rebind operations simultaneously to reduce the power dissipation. This representative iterative rescheduling algorithm puts more emphasis on the running time than optimality of each step, but still is able to generate good quality results. The algorithm runs until there is no rescheduling move with possible gain. The basic idea is to avoid the computational expense of reaching an optimal binding solution at each step and having an improved valid binding instead. In our methodology, modifications performed by the iterative rescheduling-rebinding algorithm are restricted to those movements that satisfy the DFG and resource constraints, which guarantee a valid binding after every iteration step. Hence, the need to validate the resource binding solution at the end of each iteration step is completely eliminated and this provides improvement in speed over the methodology proposed by Lyuh et al.[2] This iterative rescheduling algorithm is utilized in our experimental framework. In addition to our effort to modify the iterative rescheduling/rebinding stage to improve computational complexity, we have made an interesting observation while testing the effectiveness of our evaluation metrics. Schedules chosen with favorable metric values lead to lower post-binding power dissipation, hence reducing or even eliminating the effort needed for rescheduling altogether. In Section IV, we describe these metrics in detail. The impact of our evaluation metrics on the iterative improvement stage will be discussed in more detail in Section 6.1.

In this study we extend the analysis and results of our earlier work on metric evaluation techniques for scheduling.[45] The main contributions include:

- We extended our experimental analysis with schedules extracted using Binary Decision Diagrams. (In our early work we had worked with a small set of randomly generated schedules.) We selected 320 different schedules that are representative of the design space. We re-evaluated our proposed technique in this design space. Our new set of experimental results show that metric evaluation is very effective in rapid design space exploration. The schedules selected with metric evaluation technique have 43% less power dissipation compared to the average over all schedules.
- We improved our experimental framework to assess thermal behavior of each schedule. We augmented our experimental flow with HotSpot thermal models for a detailed temperature analysis of each design. We generated individual floorplans for each design, in order to achieve accurate localized heating information with Hotspot models. We collected data on maximum and average temperature for each individual resource. Our results indicate that metric evaluation technique is also effective in selecting schedules with lower temperatures, as well as lower power. Our experiments show that schedules selected with metric evaluation technique have 12 °C lower temperature on average, as well as a 18 °C reduction in maximum on-chip temperatures.
- We performed a complete study of all resource types including Multiplication operations and confirmed the correlation of metrics. (Our earlier work only included analysis for *Add* operations.)

2.1.3. *Thermal Analysis*

Thermal characteristics of modern electronic circuits are becoming increasingly challenging. There has been significant amount of research about design for thermal effects, thermal simulation, and evaluation of designs.[31–41] Several simulation tools have been proposed that enable the thermal modeling at higher levels. Tempest[27] by Dhodapkar et al. models the circuit temperature based on resistor capacitor equivalent of a given circuit. However, the entire circuit is modeled with a single pair of resistor and capacitor. Hence the localized heating in individual blocks cannot be detected and only an average chip temperature can be modeled. Skadron et al. proposed HotSpot thermal modeling for more detailed thermal analysis of processors.[28] It is based on the equivalent circuit of thermal resistance and capacitances, where each node in the equivalent RC network represents a building block in the circuit. This way the localized heating on processing blocks can be accurately identified. It takes as input the floorplan and the initial temperatures of the functional units, the heat spreader, and the heat sink. The power values of each functional unit are input in the form of a trace file where each line corresponds to a sampling interval. We have used a simulated annealing based floorplanner to generate floorplan for our designs. We have assumed the same chip-packaging configuration as modeled by HotSpot. The values of some of these parameters (such as chip area, sink, and spreader size, etc.) have been scaled to appropriate values for our synthesized designs. Finally, HotSpot simulates the activity on the chip and computes the steady state temperatures of the functional units. Recently, Yang et al.[44] proposed a thermal simulation framework that improves the speed of simulation significantly while maintaining accuracy. We have used HotSpot, which is a publicly available academic tool to generate the thermal profile of the datapaths synthesized in our experiments.

3. DESIGN FRAMEWORK

We have created an extensive design framework to embed and test our metric-based evaluation approach. We first describe this framework before we go into the details of our technique. Understanding the interaction between different design steps will be helpful in presenting the rationale behind our metric evaluation. The overall flow is illustrated in Figure 1. Benchmark DFGs used in our experiments are extracted from the MediaBench suit[12] using SUIF compiler infrastructure.[13] Initial scheduling is performed on these DFGs according to the timing and resource constraints. Algorithms such as the ones proposed in Refs. [7, 8, 11] can be used for this purpose. Based on this initial schedule we have established a latency constraint and a resource set for each benchmark. Next, alternative schedules that meet these constraints are generated. We have used the BDD package CUDD: CU Decision Diagram Package release 2.4.0[29] to create an initial collection of schedules.

The input DFGs have been simulated to generate switching probabilities for individual operations using a trace of 10,000 input values. For the experimental purposes these values have been randomly generated, however, actual application traces can be used if available. Functional modules (our resource sets contained ALUs to execute add, subtract, and logical operations and multipliers) have been synthesized using Synopsys Design Compiler onto a 180 nm technology library. We used scaling trends[2] to obtain switched capacitance and nominal power values for a switching activity of 0.5 at 130 nm technology node for each resource type. Capacitance values of modules have thereby been extracted to estimate switching power and this information was combined with bit toggle probabilities obtained through simulation to generate switched capacitance values. Compatibility graphs for each resource type for the scheduled DFGs have then been created, where edge weights are equal to the switched capacitances obtained as explained above. The edge costs indicate switched capacitance as a result of executing the corresponding operation pairs in succession on the same module. The compatibility graphs are given as input to the binding stage. Metric evaluation is performed before this stage based on the scheduling and switching activity information. Metric functions are introduced and discussed in Section IV.

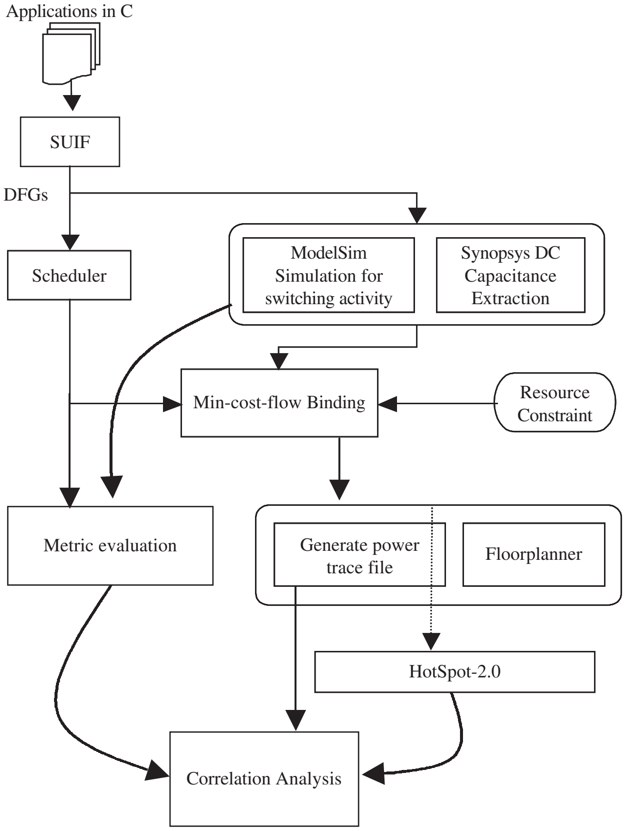

Fig. 1. Our design framework.

In the next step, optimal binding is performed on the scheduled data flow graph for a pre-allocated number of resources. We have used a software package developed by Goldberg[30] to solve the network flow formulation proposed by Chang et al.[1] Initial schedule selection has a significant effect on the power dissipation. As a result, an optimal binding alone is not sufficient to find the absolute low power solution.

Hence, our design flow incorporates an iterative rescheduling-rebinding step to further improve the binding solution. This step reschedules the operations and binds them to different resources as long as the power dissipation is improved. We refer to this stage as the iterative improvement stage and rescheduling-rebinding, interchangeably in the remainder of the discussion. This step is repeated until no possible power improvement can be attained by the rescheduling-rebinding iterations. The power estimation and thermal simulation of the final designs are performed at this point.

We used the thermal simulator HotSpot[28] to estimate the temperatures of functional units. HotSpot is originally developed to model the temperature of microprocessor architectures at the granularity of functional units by making use of the duality between heat flow and electricity. It constructs an equivalent RC network of thermal resistances and capacitances of the functional units and uses circuit-solving techniques to obtain the functional unit temperatures. It takes as input the floorplan and the initial temperatures of the functional units, thermal characteristics of the heat spreader, and the heat sink. The instantaneous power value of each functional unit is provided in the form of a trace file where each line corresponds to a sampling interval. Finally, HotSpot simulates the activity on the chip and computes the steady state temperature of the functional units.

Finally, we analyze the correlation between our metrics and the power and temperature profile of each synthesized datapath. Our analysis will be presented in Section VI in detail. In the following section we will describe our metrics for early evaluation of schedules.

4. A METRIC BASED EVALUATION TECHNIQUE

Our metric based evaluation technique will be discussed in this section. These metrics are closely related to the properties of the network flow graph extracted from a schedule and module characteristics within the resource set used. First, let us provide some basic definitions associated with the network flow graph used to represent the switching optimal resource-binding problem.

4.1. Min-Cost Flow Formulation

A compatibility graph G_i: (V, E) is defined for each operation type i in the DFG respectively such as *ADD, MUL*...etc. The network flow graphs F_i': (V', E') corresponding to each G_i can be constructed simply as follows: a node $v \in V$ in F_i', represents an operation of the type i in the DFG. Similarly, a directed edge e: $(u, v) \in E_i'$, implies that operations corresponding to the nodes (i.e., u and v) connected to this edge are compatible.

Operations u and v, of type i, are said to be compatible if it is possible to execute them in succession on the same resource without violating the timing constraints and data dependencies. Two dummy nodes source s and sink t are added to V', specifically for the network flow formulation. Let c_e denote the cost of edge e as previously discussed in Section 3. The edge costs represent the switching activity of the corresponding operation pair in succession on the same resource. A min-cost flow solution, which visits all the nodes exactly once, is the optimal solution to the power driven binding problem. Chang et al.[1] present an in-depth discussion of this methodology for the low-power binding problem.

This optimal algorithm minimizes the sum of the edge costs. Since the nodes represent the operations in the DFG, all the nodes have to be included in the solution. Essentially R units of flow is sent from the source node s to the sink node t, where R is equal to the number of available resources. Each unit of flow traces a path in the network flow graph. Hence, the final flow solution consists of R distinct paths. The nodes on the corresponding paths represent the operations executed on the same resource and the sum of the edge costs along each path represents the power consumption of that particular resource.

4.2. Metric Functions

The proposed metric functions utilize the intuition provided by the min-cost flow formulation of the low-power resource binding problem. The following formulation is used to represent the metrics:

Let v represent a node in the network flow graph corresponding to an operation in the DFG and let e be an edge connected to node v. Then, we define the following:

E_v: Set of all edges connected to node v.

E_v^k: Set of edges with weights within the minimum $k\%$ among all edge weights connected to node v.

w_{ve}^k: Weight of flow graph edge $e \in E_v^k$.

n_v: Total number of edges connected to node v.

The design space of the binding algorithm includes the edges in the network flow graph. Both the number and the weights of these edges are relevant for the quality of the min-cost flow solution. As the number of edges in the network flow graph increases the design space of the binding stage expands, increasing the probability of containing the absolute minimum power solution. Note that the overall optimal low power solution is not only dependent on the binding stage but also the schedule that is provided to the binding stage.

We define metric m_1 to account for this effect. Based on this intuitive idea metric m_1 is formulated as follows:

$$m_1 = \sum_{v \in \forall_{\text{nodes}}} n_v \tag{1}$$

A higher value for m_1 is an indicator of an increased number of edges in the network flow graph. Hence, the design space is likely to be less restricted. Therefore, it is desirable to maximize m_1 to reduce the power dissipation. In other words, an initial schedule which leads to a network flow graph with a high number of edges is likely to be a favorable schedule.

As discussed previously, the flow algorithm identifies paths in the network flow graph and each path denotes a resource instance. An important point to note is that for each node in the flow graph, the flow algorithm selects exactly one incoming and one outgoing edge in the solution. The basic idea remains the same when vertex duplication is applied.[1] Furthermore, the algorithm tends to include the edges with smallest weights in the solution, since the objective is to minimize the total cost. As a result, the distribution of the values of the edge weights available in the network flow graph may be an indicator for the quality of the final solution.

Metrics m_2 and m_3 are defined with this intention. They aim to capture the variation of edge weight values and provide a sense of the expected edge weight cost in the final binding solution. Metric m_2 considers the sum of the edges weights in the lowest $k\%$ of the value range for each node and computes the average of these sums over all nodes. Since the flow algorithm tends to select edges with smaller weights, m_2 provides an indication for the quality of the input presented to the flow algorithm. Lower m_2 values indicate a higher potential of yielding a lower power binding solution.

Metric m_3 is the generalization of metric m_2. Metric m_3 computes the average of all edge weights contained in the network flow graph. In essence, m_2 is the value of m_3 with $k = 100\%$. Metrics m_2 and m_3 should be minimized for minimum power. These metrics can be formulated as follows:

$$m_{2,3} = \frac{\sum_{v \in \forall_{\text{nodes}}} \sum_{e \in E_v^k} w_{ve}}{m_1} \tag{2}$$

The parameter k of metric m_2 can be tuned experimentally. Note that edge weights denote switching activities. For different expected input conditions and switching behavior the distribution of edge weights can be different. Therefore the value of k needs to be tuned. Within our framework, we have determined $k = 60\%$ experimentally for a given input trace and switching behavior.

4.3. Illustrative Example

In order to demonstrate the effect of scheduling on the rest of the design flow let us consider the alternative scheduling solutions illustrated in Figure 2. Assume the initial resource and latency constraints are 5 and 3 respectively. Schedules 1 and 2 are both feasible solutions in the design space.

Schedule 1 has characteristics similar to ASAP where each operation is scheduled at the earliest possible schedule step. However, as a result of the data dependencies both of the alternatives yield solutions with 3 schedule steps. The next step in the power optimal binding is to generate the corresponding network flow graph formulations of these schedules. The network flow graph formulation augments the initial node set with two special nodes: source and sink respectively. Then, the corresponding network flow graph edges are inserted. An edge (u, v) indicates the compatibility of the operations u and v. When two operations are compatible they can be executed by same resource. (i.e., execution intervals of the corresponding operations do not overlap).

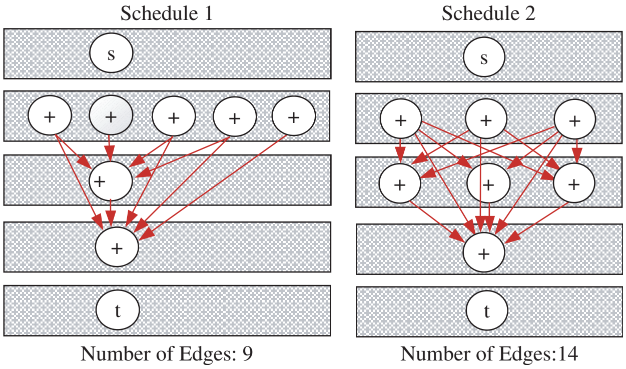

Fig. 2. Alternative schedules and the corresponding compatibility graph edges for a given DFG.

For the sake of simplicity we have excluded the edges from or to the source and sink nodes in Figure 2. By definition there would be a directed edge from the source to each node and from each node to the sink. As the numbers of edges that are connected to the source and sink are equal for both schedules this simplification does not affect the validity of the argument and yields clarity in illustration. We observe that Schedule 1 generates 9 compatibility edges compared to the 14 edges generated by Schedule 2.

The compatibility edges constitute the input design space of the following binding stage, where a power optimal binder mainly solves the max-cost flow problem on the edges. Although the same optimal binder is used for the two alternative schedules in the next stage, the two schedules impose different restrictions providing different search spaces to the binding stage. Hence, the optimal binder cannot converge to the absolute low power binding solution in each and every case.

Metric function m_1 focuses on this effect. We observe that the corresponding metric function value is favorable for Schedule 2. Hence, we expect the overall power consumption to be lower is Schedule 2 is selected for further binding. The importance of the metric functions in eliminating the schedules at the earlier stages of the design cycle is crucial in terms of the overall design quality. Our experimental results demonstrate that our metric functions indeed identify schedules with higher potential to yield a lower power solution. We will discuss our results in Section VI.

5. RESCHEDULING AND REBINDING

A representative iterative rescheduling-rebinding algorithm is employed in the design flow. Resource constraints and the maximum number of allowed schedule steps are assumed to be predefined. As illustrated in Figure 1 the iterative improvement stage takes an already scheduled and bound DFG as its input. The set of all possible rescheduling-rebinding moves with possible power improvement are considered as long as resource and DFG data dependencies are satisfied. The move with the maximum switching power gain is executed; thus the algorithm performs the locally optimal move for each step. A representative algorithm that implements this idea, is the following:

Input: Network Flow Graph G: (V, E) of an already scheduled-bound DFG.

Output: Power improved Network Flow Graph G': (V, E') with valid Scheduling and Binding.

1. $\forall$ Node $v \in G$, Repeat until Gain <0.
2. Consider all non occupied (s_i, r_i) that node v can be scheduled to/binded, checking the validity of the moves in terms of DFG and resource constraints.
3. Take the move with maximum switching power gain.
4. Perform the move (rescheduling-rebinding) to the position found in step 3.
5. Remove invalid edges; add necessary new edges to make flow valid.
6. Go back to step 2.

Our algorithm considers one operation at a time and evaluates whether it will improve power consumption if we move the operation to any of the vacant (permissible due to dependencies) control steps (provided that a resource is available at that step). This is the process of considering all permissible pairs of (s_i, r_i). If the algorithm deems beneficial to move the operation under consideration into a new location (s_i, r_i), this implies a rescheduling of the operation at the new control step s_i and rebinding of the operation to the resource r_i.

One of the reasons for high computational cost of iterative power improvement algorithms is the fact that the flow graph has to be refined at the end of each rescheduling step. By performing the valid moves only, while simultaneously rescheduling and rebinding, the algorithm eliminates the need to validate the binding at the end of each iteration step. The binding is a valid one after each iteration step. As a result, the speed of the iterative power improvement process is enhanced. Running time of the above algorithm is: $O(NRS)$ where N, R, S represent the number of operations, number of resources, and number of clock steps in the scheduled DFG respectively.

6. EXPERIMENTAL ANALYSIS AND RESULTS

We have performed experiments on benchmarks selected from the MediaBench Suite.[12] Only the results for ALU operations are displayed and discussed in this section for the sake of brevity. However similar discussion is applicable to any other operation/resource type. We have also experimented with multiplication operations and obtained consistent results. Our detailed design flow was illustrated in Figure 1. Here, we focus on the experimental analysis part of this flow, which is depicted in Figure 3.

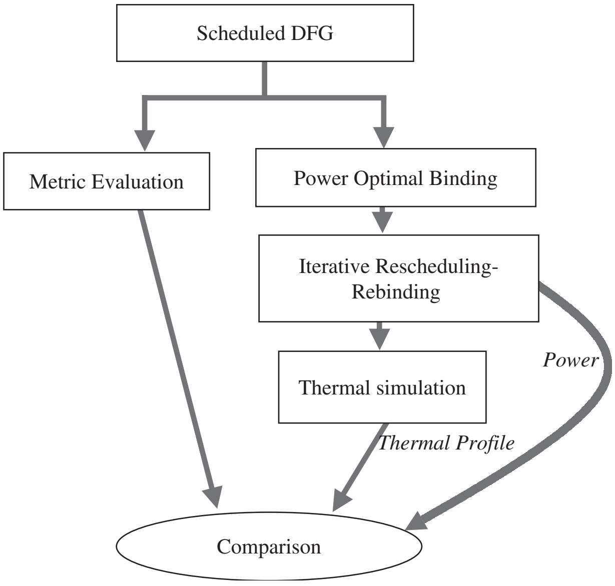

Fig. 3. The experimental analysis flow.

In this portion we generated 320 schedules for the Mediabench benchmark set. These schedules were then evaluated with our metric assessment engine. After all schedules go through the following power optimal resource binding, iterative rescheduling/rebinding, and thermal analysis steps metric evaluation results were compared with the power and temperature of the final synthesized designs.

6.1. Iterative Power Improvement Stage Gains

Table I displays the variation in post binding power dissipation and subsequent iterative power improvement for 16 different schedules of fft2. It is important to note that these 16 selected schedules and corresponding power consumption values are presented here for illustrative purposes. More extensive results over 320 schedules generated using BDDs for each DFG are reported in the next section. Fft2 was chosen for only illustration purposes. Fft2 has 78 operations (16 addition, 44 memory, 14 multiplication, 4 subtraction) and 74 edges. The rest of the data illustrates that same behavior exists for the rest of the benchmarks in the MediaBench suite.

The first 3 columns represent Metrics m_1, m_2, and m_3. The metric values are followed by the power dissipation for these schedules after resource binding (5th column). Reduction in power through iterative power improvement is given in the 6th column. Finally the overall power dissipation values after iterative improvement stage are listed.

Even without executing the rescheduling-rebinding step (i.e., iterative improvement step), the schedules selected

Table I. 16 selected schedules for the illustrative example, corresponding metric values, Post-Binding Power Consumption, Iterative Power Improvement, Power Dissipation at the end of the design flow for 16 different schedules for fft2.

Schedules (1–16)	m1	m2	m3	Post-Binding Power (mW)	Iterative Power Improvement (mW)	Overall Power (mw)
1	214	0.1875	0.2235	23.521	0.349	23.171
2	210	0.1958	0.2396	26.129	1.345	24.784
3	212	0.1926	0.2386	24.803	0.697	24.106
4	222	0.1822	0.2263	24.173	0.000	24.173
5	222	0.1821	0.2260	23.405	0.258	23.146
6	198	0.2083	0.2574	29.732	5.274	24.458
7	210	0.1970	0.2420	27.797	2.993	24.803
8	220	0.1852	0.2283	24.608	0.515	24.092
9	214	0.1949	0.2380	25.972	2.773	23.198
10	212	0.1928	0.2361	24.334	0.996	23.337
11	216	0.187	0.2333	24.003	1.204	22.798
12	**224**	**0.1827**	**0.2242**	**23.487**	**0.697**	**22.789**
13	214	0.1887	0.2351	24.010	0.380	23.630
14	216	0.1905	0.2334	25.899	2.227	23.672
15	208	0.1964	0.2421	25.179	1.405	23.773
16	**212**	**0.1924**	**0.2392**	**25.889**	**1.978**	**23.910**

by our metric evaluation technique have power dissipation comparable to the minimum power schedule among all the experimented DFGs after rescheduling (indicated by Schedule 12 in the first highlighted row of Table I).

These two highlighted rows represent Schedule 12 and Schedule 16 along with their corresponding power dissipation values for post-binding, iterative improvement and final stages. Schedule 12 is a schedule we selected using our metric evaluation. It has the highest metric 1 value for the schedules we looked at, a low metric m_2 for the range of metric m_2, and the lowest metric m_3 value. All these observations on metric values indicate that Schedule 12 is a favorable schedule that has potentially low power dissipation and almost no need for iterative rescheduling/rebinding. Our experimental results in the following columns are consistent with this prediction. The post-binding power dissipation is comparable to the power dissipation of most other schedules that went through iterative improvement. Finally, Schedule 12 yields the lowest power dissipation at the end of the design flow.

Figure 4 displays the comparison of post-binding power, iterative rebinding gain and overall power for Schedule 16 and Schedule 12. It is important to note that iterative improvement stage provides almost no power reduction for Schedule 12 (one of the lowest over all schedules). As a result the computationally expensive Iterative Power Improvement is not as critical to the design flow if Schedule 12 was selected as the initial schedule using our metric evaluation technique. Similar observations are valid for the entire set of experimented benchmarks. We observed that the iterative improvement stage could safely be removed for reaching optimal or near-optimal power dissipation if the initial schedule is selected using our metrics.

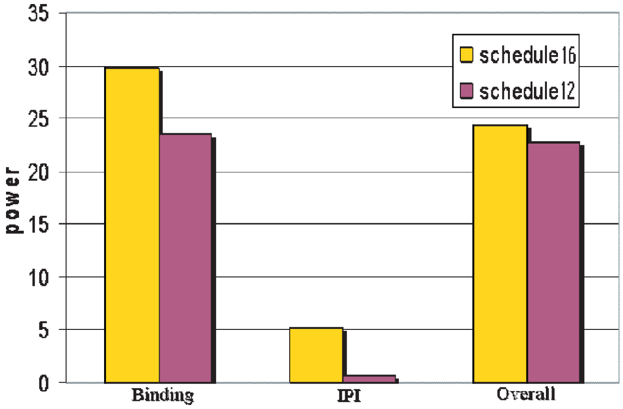

Fig. 4. Power dissipation after binding, iterative power improvement and overall power consumption for Schedule 12 and Schedule 16.

6.2. Correlation of Metrics with Post-Binding Power Dissipation

Figure 5 illustrates the correlation between the metrics and the corresponding power consumption values for fft2. The values on the y-axis are the curve fitted versions of the experimental results. The correlation between the metrics and the power dissipation can be observed from the plot. This is coherent with the aforementioned observations on the data from Table I. We can observe that the value distributions for metrics m_2 and m_3 follow the same trend as the value distribution of the power consumption. Metric m_1 on the other hand, displays a trend, which is the mirror symmetry of the power consumption curve.

Results exhibit high correlation between the power dissipation values and the metrics. Similar arguments hold for the rest of the benchmarks. Table II and Table III illustrate the correlation factors between our metrics and post binding power dissipation and iterative rescheduling power improvement respectively for all benchmarks. These values are computed only for the schedules with extreme metric values.

All the three metrics provide high correlations with power and iterative power improvement (as high as 0.98) and with post binding power dissipation (as high as 0.949). Majority of the correlation values are higher than 0.75 for both cases. For the data points with high m_2, m_3 (low m_1),

Table II. Correlation of metrics with power dissipation.

Benchmark	m1	m2	m3
fft1	0.877	0.934	0.949
fft2	0.872	0.940	0.946
jctrans1	0.603	0.598	0.617
jctrans2	0.722	0.890	0.728
jdmerge1	0.769	0.870	0.724
jdmerge2	0.888	0.849	0.934
jdmerge3	0.869	0.804	0.728
jdmerge4	0.646	0.788	0.760
noise est2	0.258	0.014	0.103

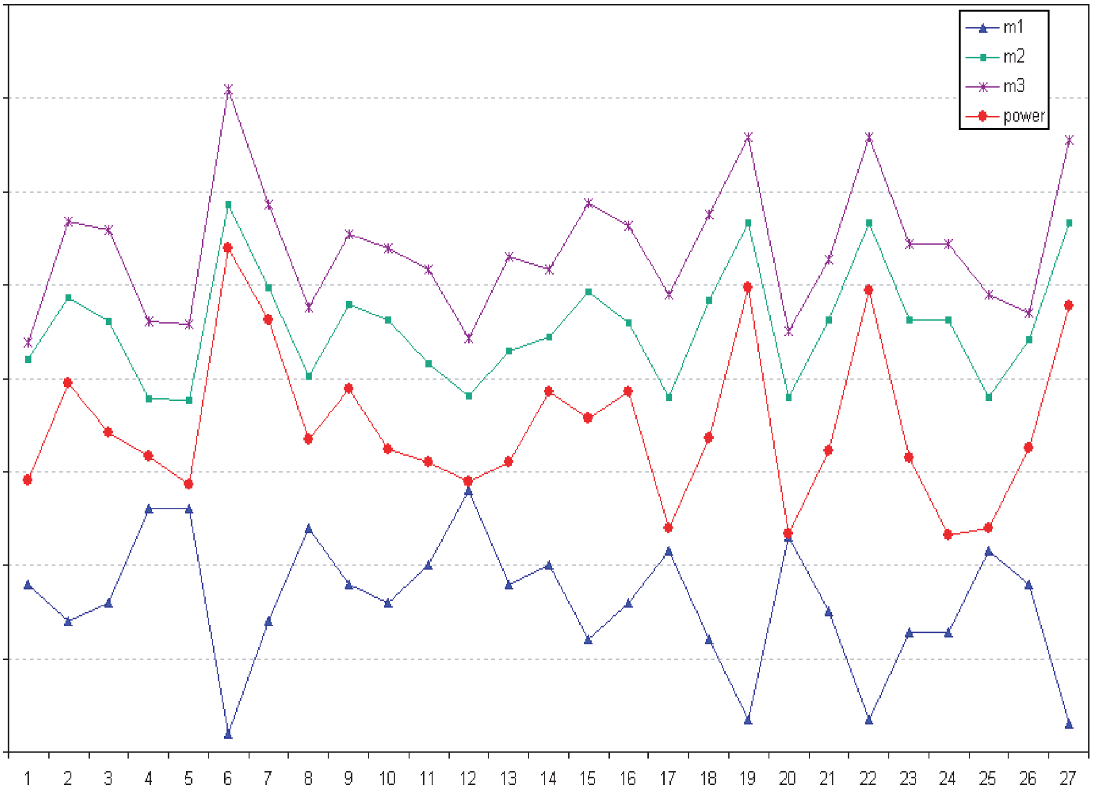

Fig. 5. Curve fitted data for m_1, m_2, m_3, power dissipation variations for fft2.

Table III. Correlation of metrics with iterative improvement gain.

Benchmark	m1	m2	m3
fft1	0.510	0.617	0.946
fft2	0.837	0.903	0.916
jctrans1	0.800	0.799	0.818
jctrans2	0.819	0.937	0.692
jdmerge1	0.566	0.666	0.725
jdmerge2	0.940	0.835	0.981
jdmerge3	0.645	0.577	0.478
jdmerge4	0.456	0.262	0.234
noise est2	0.077	0.020	0.040

post binding power dissipation and the iterative rescheduling improvement are high as well. (Similarly, correlation factors for the iterative improvement is shown in Table III).

The correlation of the metrics with post-binding power dissipation and iterative power improvement indicates another important point. The iterative power improvement gain is high for the schedules with high post binding values. This, along with the high correlation factors for the metrics imply that we can exploit the metric functions to select the schedules that have lowest power dissipation in post-binding step.

Power consumption values for these schedules are very close to the values of the other schedules after iterative power improvement step. Hence, the metric functions can be utilized in evaluation of the initial schedules in terms of the aforementioned qualities. Our experimental results indicate that schedules that are selected with metric evaluation have power dissipation within 3% proximity of the absolute minimal power scheduling.

We also compared the schedules that are selected with our metrics with the average of all 320 schedules we have experimented on. For each benchmark we have selected 3 schedules using our metric evaluation technique. Then, we compared power dissipation, maximum temperature reached by any resource and average temperature across all resources for these benchmarks with the averages over all the schedules for each benchmark. Our experimental results in Table IV show that power dissipation for the schedules selected with metric evaluation technique is 43% lower. This result is even more promising than the power dissipation for the (Schedule 12, Schedule 16)

Table IV. Power difference between schedules selected by our metric and all schedules.

Benchmark	Power (% Difference)
jdmerge1	72.86
jdmerge2	71.00
jdmerge3	70.38
jdmerge4	71.34
fft1	1.48
fft2	34.67
jctrans1	0.57
jctrans2	−1.68
noise est2	−1.62

Table V. Running time of metric evaluation (in msec), Iterative Power Improvement and Optimal Binding, speedup computed as the ratio of the 2nd and 3rd columns.

Benchmark	Metric (msec)	Iterative Scheduling-Binding	Speedup
fft1	17.99	980.93	54.50
fft2	3.14	108.52	34.51
jctrans1	1.98	94.51	47.68
jctrans2	2.35	96.01	40.73
jdmerge1	3.74	152.23	40.68
jdmerge2	17.17	1069.45	62.26
jdmerge3	5.74	227.83	39.65
jdmerge4	4.61	141.91	30.78
noise est2	3.47	114.45	32.96

example we have illustrated before. The reason is, with a larger design space to explore within the 320 representative schedules, our metric evaluation technique is more effective in selecting the promising low power candidates. In general the effectiveness of the technique is better for a larger set of initial schedules. Since metric-evaluation is a fast technique this kind of initial design exploration of all possible schedules can be performed, completely eliminating iterative improvement steps at the later stages of the design flow.

6.3. Speed of Metric Evaluation Technique

Table V tabulates the running time for metric evaluation with optimal binding and iterative power improvement. The ratios are as indicated in column 4. The results indicate that metric evaluation is on average 42.6 times faster than optimal binding and iterative improvement. Speedups as high as 62 can be attained by applying the metric evaluation technique and selecting accordingly early in the design cycle as opposed to going through binding and iterative improvement to evaluate the quality of the initial schedule. The speed of the metric evaluation is another reason that emphasizes the usefulness of our technique.

Finally, note that the resource binding process in itself might not take very long in a single pass. However, consider deploying such a behavioral synthesis tool within the inner loop of a design space exploration task. Integration of such tools with lower level steps such as logic synthesis and floorplanning is also becoming increasingly popular. Within the inner loop of such a synthesis engine, the run-time of a single iteration will become crucial. The re-synthesis step would be needed to repeat for a significantly large number of times and in addition the inputs to the synthesis stage can be much larger (the size of the DFG, number of resources, etc.) in practice. The run-times in this paper are reported for a single run. However, if we consider repeating this task several hundred times within a more complicated behavioral synthesis engine the benefit of the speedup will be much more emphasized.

Table VI. Difference between the average (and maximum) block temperatures for metric selected schedules and the remaining schedules.

Benchmark	Difference in Maximum Temp	Difference in Average Temp °C
jdmerge1	7.98	3.07
jdmerge2	47.57	11.64
jdmerge3	31.30	38.28
jdmerge4	9.45	11.56
fft1	8.10	1.87
fft2	52.32	40.19
jctrans1	7.87	0.33
jctrans2	−0.20	0.66
noise est2	−1.45	2.01

6.4. Thermal Benefits of Metric Evaluation

As temperatures correlate with the power dissipation of individual designs, we also explored possible thermal benefits of metric evaluation. It is important to note that, similar to power optimization, temperature optimization techniques are also needed in various stages of the design flow. In earlier stages of the design process, the critical decisions such as schedule selection affect the thermal characteristics of the final design. Note that, thermal-aware schedule selection is beyond the scope of this paper. Our main goal is power efficient scheduling selection. However, we intended to take the first step into investigating the potential thermal benefits of these power efficient scheduling solutions.

The reduction in power dissipation improves the on-chip temperatures as well. For the schedules selected with our metrics maximum on-chip temperatures are 18 °C less on average compared to the average temperature of the entire set. Furthermore, we observed that average temperatures reached by resources in the synthesized datapaths of these schedules are also 12 °C lower.

We did not assume any dynamic thermal management scheme for these synthesized designs. Therefore, the on-chip temperatures represent the theoretical high values. Table VI presents the differences between the schedules selected with metric evaluation and the average of all created schedules. For the schedules with favorable metric values: maximum temperature can be as much as 52 °C lower than another candidate schedule. Average temperatures for a favorable schedule can be as much as 40 °C lower.

7. CONCLUSION

In this paper we investigated the effects of scheduling on power dissipation. We proposed metrics that exhibit high correlation with power dissipation and iterative rescheduling power improvement, which can be exploited for initial schedule selection. With our fast and effective scheduling evaluation step, the need for iterative rescheduling/rebinding at later stages of the design flow can be reduced or even completely eliminated. Experiments with the MediaBench suite indicated that correlation factors are as high as 0.95 and higher than 0.75 for most cases. Comparing the lowest power schedules after iterative improvement with schedules that were optimized for the proposed metrics exhibited comparable overall power dissipation. Metric evaluation enables power estimation at the early stages of behavioral synthesis. This can be exploited in better power management and optimization. The results demonstrate that the design effort required in rescheduling and rebinding can be reduced significantly with this method. Optimizing for the proposed metrics can reduce the need for an aggressive rescheduling. Furthermore metric evaluation is on average 42.6 times faster than a faster version of a recently proposed optimal binding and iterative improvement algorithm. We have shown that our rapid scheduling evaluation step is very effective in exploring a design space created using BDDs to represent possible alternative schedules. Power dissipation for the schedules selected with metric evaluation technique is 43% lower, along with favorable thermal behavior. Average temperature for this set of schedules is 12 °C lower than the average of all schedules. Maximum temperatures are also 18 °C lower compared to the average maximum temperature for the entire set.

References

1. J. Chang J. M. and M. Pedram, Register allocation and binding for low power. *Design Automation Conference* (**1995**), pp. 93–98.
2. C. G. Lyuh, T. Kim, and C. L. Liu, An integrated data path optimization for low power based on network flow method. *International Conference on Computer Aided Design* (**2001**), pp. 553–559.
3. D. Ku and G. De Micheli, Relative scheduling under timing constraints, design automation conference (**1990**), pp. 59–6.
4. 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. on VLSI Systems* (**2001**), pp. 3–14.
5. A. Dasgupta and R. Karri, Simultaneous scheduling and binding for power minimization during microarchitecture synthesis. *International Symposium on Low Power Electronics and Design* (**1995**), pp. 69–74.
6. A. Dasgupta and R. Karri, High-Reliability, Low energy microarchitecture synthesis. *IEEE, Transactions on Computer Aided Design* (**1989**), pp. 661–679.
7. T. C. Hu, Parallel sequencing and assembly line problems. *Operations Research* (**1961**), pp. 841–848.
8. P. Paulin and J. Knight, Force directed scheduling for the behavioral synthesis of ASIC's. *IEEE Transactions on Computer Aided Design* (**1989**), Vol. 8, pp. 661–679.
9. R. Potasman, J. Lis, A. Nicolau, and D. Gajski, Percolation based scheduling. *Design Automation Conference* (**1990**), pp. 444–449.
10. R. Camposano, Path-based scheduling for synthesis. *IEEE Transactions on Computer Aided Design* (**1990**), Vol. 10, pp. 85–93.
11. S. Ogrenci Memik, E. Bozorgzadeh, R. Kastner, and M. Sarrafzadeh, A super-scheduler for embedded reconfigurable systems. *International Conference on Computer Aided Design* (**2001**), pp. 391–394.
12. C. Lee, M. Potkonjak, and W. H. Mangione-Smith, MediaBench: A tool for evaluating and synthesizing multimedia and communications systems. *International Symposium on Microarchitecture* (**1997**), pp. 330–335.
13. SUIF. http://suif.stanford.edu.
14. A. Raghunathan and N. K. Jha, An iterative improvement algorithm for low power data path synthesis. *International Conference on Computer Aided Design* (**1995**), pp. 329–336.
15. C. H. Gebotys, Low energy memory and register using network flow. *Design Automation Conference* (**1997**), pp. 435–440.
16. S. Hong and T. Kim, Bus optimization for low-power data path synthesis based on network flow method. *International Conference on Computer Aided Design* (**2000**), pp. 312–317.
17. M. Pedram, Power minimization in IC design: Principles and applications. *ACM Transactions on Design Automation and Electronic Systems* (**1996**), Vol. 1, pp. 3–56.
18. F. Najm, A survey of power estimation techniques in VLSI circuits. *IEEE Transactions VLSI Systems* (**1994**), Vol. 4, pp. 446–455.
19. F. Najm, Power estimation techniques for integrated circuits. *In Proceedings of the International Conference on Computer Aided Design* (**1995**), pp. 492–499.
20. S. Devadas and S. Malik, A survey of optimization techniques targeting low power VLSI circuits. *Design Automation Conference* (**1995**), pp. 242–247.
21. A. Raghunathan and N. K. Jha, Behavioral synthesis for low power. *In Proceedings of International Conference on Computer Design* (**1994**), pp. 318–322.
22. W. Lin, C.-T. Hwang, and A. Wu, Scheduling techniques for variable voltage low power designs. *ACM Transactions on Design Automation of Electronic Systems* (**1997**), Vol. 2, pp. 81–97.
23. C.-L. Su, C.-Y. Tsui, and A. M. Despain, Saving power in the control path of embedded processors. *IEEE Design and Test of Computers* (**1994**), Vol. 11, pp. 24–30.
24. D. Kim, D. Shin, and K. Choi, Low power pipelining of linear systems, a common operand centric approach. *In Proceedings of IEEE/ACM International Symposium on Low Power Electronics and Design* (**2000**), pp. 225–230.
25. Z. Shao, Q. Zhuge, E. H.-M. Sha, and C. Chantrapornchai, Analysis and algorithms for scheduling with minimal switching activities. *IEEE Circuits and Systems* (**2002**), Vol. 1, pp. I-372–5.
26. A. Chandrakasan, T. Sheng, and R. W. Brodersen, Low power CMOS digital design. *Journal of Solid State Circuits* (**1992**), pp. 473–484.
27. A. Dhodapkar, C. H. Lim, G. Cai, and W. R. Daasch, TEMPEST: A thermal enabled multi-model power/performance estimator. *In Workshop on Power Aware Computer Systems* (**2000**), pp. 94–125.
28. K. Skadron, M. Stan, W. Huang, S. Velusamy, K. Sankaranarayanan, and D. Tarjan, Temperature-aware microarchitecture. *In 30th Annual International Symposium on Computer Architecture* (**2003**), pp. 2–13.
29. CUDD: CU Decision Diagram Package 2.4.0, URL: http://vlsi.colorado.edu/~fabio/CUDD/cuddIntro.html.
30. A. V. Goldberg, An efficient implementation of a scaling minimum-cost flow algorithm. *Journal on Algorithms* (**1997**), 22, pp. 1–29.
31. M. N. Sabry, Dynamic compact thermal models: An overview of current and potential advances. *Proceedings of International Electronic Packaging Technical Conference Exhibition* (**2003**), pp. 6–11.
32. W. Huang, S. Ghosh, K. Sankaranarayanan, K. Skadron, and M. R. Stan, Compact thermal modeling for temperature-aware design. *Design Automation Conference* (**2004**), pp. 878–883.
33. W. Liao, F. Lei, and L. He, Microarchitecture level power and thermal simulation considering temperature dependent leakage model. *International Symposium on Low Power Electronics and Design* (**2003**), pp. 211–216.
34. D. Brooks, and M. Martonosi Dynamic thermal management for high-performance microprocessors. *International Symposium on High-Performance Computer Architecture* (**2001**), pp. 171–182.
35. W. Huang, J. Renau, S. M. Yoo, and J. Torellas, A framework for dynamic energy efficiency and temperature management. *International Symposium on Microarchitecture* (**2000**), pp. 202–213.

36. C. Tsai and S. Kang, Standard cell placement for even on-chip thermal distribution. *International Symposium on Physical Design* (**1999**), pp. 179–184.

37. C. C. N. Chu and D. F. Wong, A matrix synthesis approach to thermal placement. *International Symposium on Physical Design* (**1997**), pp. 163–168.

38. J. Cong, J. Wei, and Y. Zhang. A thermal-driven floorplanning algorithm for 3D ICs. *International Conference on Computer-Aided Design* (**2004**), pp. 306–313.

39. K. Banerjee, A. Mehrotra, A. Sangiovanni-Vincentelli, and C. Hu, On thermal effects in deep sub-micron VLSI interconnects. *Design Automation Conference* (**1999**), pp. 885–891.

40. T. Y. Chiang, K. Banerjee, and K. C. Saraswat, Analytical thermal model for multilevel VLSI interconnects incorporating via effect. *IEEE Electron Device Letters* (**2002**), 23, pp. 31–33.

41. K. Banerjee, S.-C. Lin, and V. Wason, Leakage and variation aware thermal management of nanometer scale ICs. *IMAPS-Advanced Technology Workshop on Thermal Management* (**2004**).

42. X. Tang, T. Jiang, A. Jones, and P. Banerjee, Behavioral synthesis of data-dominated circuits for minimal energy implementation. *IEEE International Conference on VLSI Design* (**2005**), pp. 267–273.

43. A. Raghunathan and N. K. Jha, Behavioral synthesis for low power. *Proceedings of International Conference on Computer Design* (**1994**), pp. 318–322.

44. Y. Yang, Z. Gu, C. Zhu, L. Shang, and R. P. Dick, Adaptive chip-package thermal analysis for synthesis and design. *To appear in Proc. Conf. on Design, Automation, and Test in Europe* (**2006**).

45. E. Kursun, A. Srivastava, S. Ogrenci Memik, and M. Sarrafzadeh, Early evaluation techniques for low power binding. *International Symposium on Low Power Design* (**2002**), pp. 160–165.

Eren Kursun

Eren Kursun *received her B.S. degree from Electrical Engineering from Bogazici University, Istanbul, in 2000. She received her M.S. degree from Computer Science Department, University of California Los Angeles in 2002. She is currently pursuing Ph.D. studies in Microarchitecture Research Laboratory in Computer Science Department at University of California Los Angeles.*

Rajarshi Mukherjee

Rajarshi Mukherjee *earned his B.E. in Electronics and Telecommunication Engineering, Jadavpur University, Calcutta, India 2000. He received his M.S. in Computer Engineering, ECE Dept, Northwestern University, Evanston, IL 2003. He is pursuing doctoral studies in Computer Engineering, EECS Dept, Northwestern University, Evanston, IL*

Seda Ogrenci Memik

Seda Ogrenci Memik *received her B.S. degree in Electrical and Electronic Engineering from Bogazici University, Istanbul, Turkey in 1998 her Ph.D. degree in Computer Science from UCLA in 2003. She is currently an Assistant Professor at the Department of Electrical Engineering and Computer Science of Northwestern University. Dr. Memik has authored two book chapters and over 30 technical papers. Her reserach interests include low power and thermal-aware design and low power reconfigurable systems.*