Early Evaluation Techniques for Low Power Binding

Eren Kursun Ankur Srivastava Seda Ogrenci Memik Majid Sarrafzadeh

{kursun, apple, seda, majid} @cs.ucla.edu

Computer Science Department
University of California Los Angeles, CA

ABSTRACT

This paper presents effective metrics to evaluate the power dissipation of scheduled data flow graphs (DFGs). This enables early evaluation of schedules without performing the computationally expensive resource-binding step. Our metrics correlate heavily (as high as 0.95 and > 0.75 for most test cases) with power dissipation values obtained after resource binding and rescheduling for power optimization steps. An experimental flow that integrates path-based scheduling, power optimal binding and power driven iterative rescheduling stages is constructed. The flow integrates commercial tools like Synopsys, VSS and academic compilers like SUIF in a common optimization framework. Experimental results on DFGs from MediaBench suit also demonstrate the fact that metric evaluation is on average 42.6 times faster than performing optimal binding and iterative power improvement. Hence metric based evaluation enables fast design exploration at early stages.

Categories & Subject Descriptors: [Design] *High Level Synthesis, Power Optimization, Scheduling, Resource Binding.*

General Terms: Design

Keywords: Low Power Design, Scheduling, Resource Binding, Metric Evaluation.

1. INTRODUCTION

In recent years personal computing devices and wireless communication systems have gained remarkable popularity. Their demand for high-speed computation and complex functionality with low power consumption were among the major driving factors, inducing power to be a critical design concern. Not too long ago, performance, area and testability were considered as major design considerations, while power issues were of secondary importance. Number of on-chip transistors continued to increase and power dissipation reached excessive values, becoming limiting factor in many designs. This has led to an increase in the cost of packaging and cooling devices, drawing the attention of the industry as well as the research community to power optimization.

Power optimization techniques can be applied at various abstraction levels of design. Effective high level power optimization methodologies are desired at the early stages of the behavioral synthesis, as the decisions made at the earlier steps of the design cycle have more impact on the final implementation. An extensive body of work exists on behavioral synthesis for low power. Chang and Pedram [1] proposed a switching activity calculation technique for registers. They also solved the register assignment problem for minimum power consumption using a max-flow formulation optimally. Raghunathan *et al.* [14] presented an iterative improvement technique on switched capacitance matrices. Dasgupta *et al.* [5], [6] introduced simulated annealing based transition minimization algorithms. Recently Lyuh *et al* [2] proposed network flow based approach for low power data path optimization, which reduces the redundancy in flow computations. The behavioral level power optimization, iterative power improvement techniques, and network flow based formulation of related problems can be found in [1]-[6], [14]- [16].

Power driven high-level synthesis methodologies commonly incorporate a scheduler followed by binder [1], [4]. Rescheduling step that modifies the existing solution to improve the power dissipation follows the power driven binder [2]. The objective of rescheduling is to modify the original schedule to improve the power. The effectiveness of this extra optimization becomes lesser or even negligible if the initial schedule is wisely selected. However, the lack of criteria to evaluate the initial schedule is an important factor that hinders this. Hence iterative power optimization techniques are commonly used in practice.

In this paper we develop metrics to serve as optimization criteria leading to better initial schedules. Utilization of these metrics in initial schedule selection provides reduction in the design effort required for iterative improvement. Our experiments illustrate that in some cases, it even eliminates the need of iterative rescheduling. The proposed metrics exhibit high correlation with the post binding power dissipation and power driven rescheduling improvement, enabling prediction of the power dissipation at the prebinding-post scheduling stage. Post resource binding power dissipation values are reduced, when initial schedule selection is based on metric evaluation. As a result, design effort required for iterative improvement steps can be reduced significantly or even eliminated.

A design flow with the following properties 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 values are computed for the generated schedules after module characterization and simulation steps are completed. Subsequent to the optimal binding and iterative rescheduling-rebinding, correlation of the metric values with post binding power

ISLPED'02, August 12-14, 2002, Monterey, California, USA.
Copyright 2002 ACM 1-58113-475-4/02/0008...$5.00.

dissipation and rescheduling improvement is investigated for each schedule respectively. This correlation was found to be as high as 0.95 and higher than 0.75 for most test cases. These metrics enable quick power estimation without executing the computationally expensive resource binding and iterative rescheduling. Hence they can be employed in attaining better design space exploration and better power management.

The rest of the paper is organized as follows: Section 2 describes the 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 are reported in Section 6 and conclusion is in Section 7.

2. PRELIMINARIES

2.1 The Low Power Binding Problem

The low-power binding problem was optimally solved using the max-cost flow and matching techniques in [1] and [4] respectively. This study focuses on the max-cost flow formulation of the problem. In [1], compatibility graphs are generated from the scheduling information. Operations are represented as nodes in the compatibility graph. Every operation pair *u, v* that can potentially be executed on the same resource in succession, have 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 immediately 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 running 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 following sections. 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 skipped here for brevity. Chang *et al.* [1] present an in-depth discussion of the max-cost flow methodology for the low-power binding problem.

2.2 Rescheduling for Low Power

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 the moment more accurate information becomes available, initial design decision is further refined and improved upon. An example of this design paradigm is the methodology of Layout Driven Logic Synthesis. The synthesis decisions are improved when more accurate wire-length 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 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, change the data transfers, which invalidates some paths from the previous solution. One way to solve this problem is to perform the max-cost flow algorithm after each iteration step to refine the binding; nevertheless this is not desirable because of its high computational complexity. Recently Lyuh *et al* [2] proposed a method that addresses this running time 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, 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 step 1.

The algorithm offers both running time improvement and power efficiency. 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 operations are rescheduled and rebinded 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 the movements that satisfy the DFG and resource constraints, which guarantee a valid binding after every iteration step. Hence, the need to make the resource binding solution valid at the end of each iteration step is completely eliminated and provides possible improvement in speed over the methodology proposed by Lyuh *et al.* [2]. This iterative rescheduling algorithm is utilized during the experimentation along with the metrics formulated for estimating the power dissipation after resource binding. Schedules with favorable metric values lead to lower post binding power dissipation, hence reducing or even eliminating the effort needed for rescheduling. In Section 4, we describe these metrics in detail.

3. DESIGN FLOW

The design flow constructed for experimentation is illustrated in Figure 1. Benchmark DFGs used in our experiments are extracted from the MediaBench suit [12] using SUIF compiler infrastructure [13]. Scheduling is performed on these DFGs according to the timing and resource constraints. Algorithms such as the ones proposed in [7], [8], [11] can be used for this purpose. Alternative schedules that meet the constraints are generated. For each schedule generated, post-binding power dissipation and the correlation factors for the metrics will be computed in the subsequent steps. Based on the trace information, DFG simulation is performed. Internal DFG variables are calculated for the given input trace, which is a representative of the input statistics of the DFG. Subsequently module characterization and switching power computation is performed. Module characterization engine incorporates VSS and Synopsys Design Compiler tools. The input to the engine comes from the scheduler, module library, and simulation results of the scheduled DFG. Edge cost information is extracted based on this data. The outputted edge costs indicate switched capacitance of executing the corresponding operation pairs in succession on the same module.

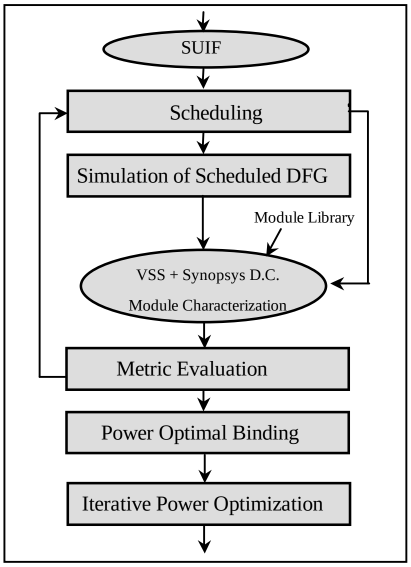

Figure 1: Design flow

Metric evaluation is performed at this stage based on the scheduling and module characterization information. Metric functions are formulated and discussed in Section 4. The correlation factors for the metrics are extracted at the final stage.

In the next step optimal binding is performed on the scheduled data flow graph for pre-estimated number of resources. A power optimal binder such as the one proposed by Chang *et al.* [1] can be used in this stage. Initial schedule selection has a significant effect on the power dissipation. As a result of this, only an optimal binder is not sufficient to find low power solutions by itself. Hence, the existing design flow incorporates an iterative rescheduling-rebinding step to improve the binding solution, as discussed in Section 2.

The iterative rescheduling-rebinding stage reschedules the operations and binds them to different resources as long as the power dissipation is improved. This step is repeated until no possible power improvement can be attained by the rescheduling-rebinding iterations. In the following sections we argue that if the initial schedule is optimized for the proposed metrics, the design effort on this rescheduling step can be minimized or even eliminated, based on the experimental results.

4. METRICS FOR LOW POWER BINDING

Proposed metric functions will be discussed in this section. These metrics are based on the network flow graph information, extracted from schedule and module characteristics.

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 the type *i*, are said to be compatible if it is possible execute them in succession on the same resource without violating the timing constraints and data dependencies. Two dummy nodes namely source *s* and the 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 covers 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 destination node *t*, where *R* signifies 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 the path represent the power consumption of that particular resource.

4.2 Metric Functions

The proposed metric functions utilize the intuition provided by this min-cost flow formulation of the low-power binding problem. The following formulation is used to represent the metrics: Let *v* represent a node in the network flow graph representing an operation in the DFG; *e* is an edge connected to node *v*.

E_v : Set of all edges connected to node *v*.
E_v^k : Set of edges with weights belonging to the minimum *k%* of 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*.

Design space of the binder includes the edges in the network flow graph. Both the number and the weights of these edges are important for the quality of the min-cost flow solution. As the number of edges in the network flow graph increase the design space of the binder is enlarged, increasing the probability of having optimal solution in the design space. We employ metric m_1 to account for this effect. Based on this intuitive idea *metric 1* is formulated as follows:

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

Higher m_1 is an indicator of an increased number of edges in the network flow graph; hence the design space is a less restricted one. Therefore the objective is to maximize m_1 to reduce the power dissipation. As previously discussed, the flow algorithm identifies paths in the network flow graph and each path signifies 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 as in [1]. In that case we consider the duplicated node pairs as single nodes. Furthermore, the algorithm tends to include the smallest edges in the solution, since the objective is minimization of the total cost. As a result, considering the average of the edge weights over all the edges may lead to an indicator for the quality of the final solution. Algorithm is provided with a set of edges that have favorable range of weight values. Moreover restricting the edges to those with weights that pertain to the minimum *k %* of all the edge weights on that particular node is possible as well.

Metrics m_2 and m_3 are based on the sum of edge weights in the network flow graph. Metric m_2 considers the sum of the edges with weight in the minimum *k %* of the edge weights range for that particular node. Since the flow algorithm tends to select edges with smaller weights, m_2 provides an indicator for the quality of the input of the flow algorithm. Lower m_2 values indicate a higher potential of yielding lower power binding solution. In m_3 all the edges related to a node are considered (i.e. *k: 100 %*). Metrics m_2 and m_3 should be minimized for minimum power. These metrics can be formulated as follows:

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

Changing variables like *k* might be performed for further tuning of the metrics. (*k = 60* for m_2 is chosen as a result of experiments.)

5. RESCHEDULING - REBINDING

A representative iterative rescheduling-rebinding algorithm is employed in the design flow. Resource constraints and the number of schedule steps are assumed to be predefined. As illustrated in Figure 1 this stage takes an already scheduled and binded DFG as its input.

The set of all possible rescheduling-rebinding movements with possible power improvement are considered as long as resource and DFG data dependencies are satisfied. The movement with 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-binded DFG.

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

```
1. ∀ Node v∈ 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 step3
5. Remove invalid edges; add necessary new edges to make flow valid.
6. Go back to step2
```

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 make the binding valid 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.

7. EXPERIMENTAL RESULTS

Experiments are performed on benchmarks selected from MediaBench Suite. Only the results for *ADD* operations are displayed and discussed in this section for the sake of brevity. However similar discussion is applicable to any other operation type. Figure 2 illustrates the variation in post binding power dissipation for different schedules of the same DFG, without rescheduling-rebinding along with the metrics m_1, m_2 and m_3. The values on the y-axis are the curve fitted versions of the data in Table 1. The correlation between the metrics and the power dissipation and iterative power improvement can be observed from the plot.

Table 1. Metric values, Post Binding power, Iterative Improvement (IPI), Power Dissipation after iterative power improvement of 16 different schedules for fft2.

m1	m2	m3	Power	I.P.I.	Overall
214	0.187550	0.223500	23.5211	0.34946	23.17164
210	0.195810	0.239660	26.1291	1.34500	24.78410
212	0.192644	0.238629	24.80364	0.69710	24.10654
222	0.182267	0.226375	24.17378	0.00000	24.17378
222	0.182160	0.226040	23.40554	0.25882	23.14672
198	0.208341	0.257466	29.73294	5.27448	24.45846
210	0.197059	0.242000	27.79720	2.99328	24.80392
220	0.185234	0.228337	24.60874	0.51598	24.09276
214	0.194940	0.238080	25.97200	2.77344	23.19856
212	0.192800	0.236100	24.33438	0.99670	23.33768
216	0.187050	0.233300	24.00300	1.20460	22.79840
224	***0.182700***	***0.224200***	***23.48700***	***0.69706***	***22.78994***
214	0.188700	0.235100	24.01048	0.38038	23.63010
216	0.190549	0.233409	25.89930	2.22700	23.67230
208	0.196491	0.242120	25.17900	1.40520	23.77380
212	0.192400	0.239200	25.88910	1.97850	23.91060

The plot demonstrates the fact that the data points with high post binding power values have high iterative power improvement as well. Even without executing the rescheduling step, the schedules selected by the proposed metric evaluation technique have power dissipation comparable to the minimum power schedule among all the experimented DFGs after rescheduling (indicated by the highlighted row in Table 1). This implies that rescheduling algorithm can improve the overall power dissipation of the designs with reduced effort on rescheduling. Similar observations are valid for the entire set of experimented DFGs and will be discussed later in this section. Results exhibit high correlation between the power dissipation values and the metrics. Tables 2 and 3 illustrate the correlation factors for post binding power dissipation and iterative rescheduling power improvement respectively. These values are computed only for the schedules with extreme metric values.

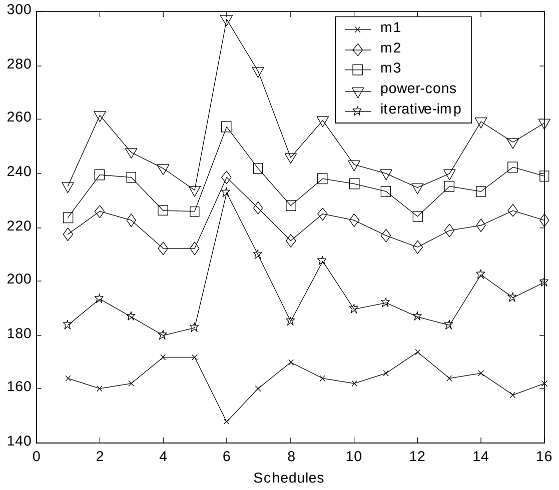

Figure 2. Curve fitted version for m$_1$, m$_2$, m$_3$, power dissipation, iterative power improvement gain variations for fft2

All the three metrics report high correlations with power and iterative power improvement (as high as 0.981) 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)*, post binding power dissipation and the iterative rescheduling improvement are high as well. (Similarly for iterative improvement as shown in Table 3.)

Table 2. Correlation of Metrics with Power Dissipation

	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_est	0.258	0.014	0.103

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. Table 5 tabulates the running time for metric evaluation with optimal binding and iterative power improvement. The ratios are as indicated in column 3. The results indicate that metric evaluation is on average 42.6 times faster than optimal binding and iterative improvement.

Speed improvement ratios as high as 62 can be attained by applying metric evaluation technique and selecting accordingly early in the design cycle as opposed to going through binding and iterative improvement stages to evaluate the quality of the initial schedule. The speed of the metric evaluation is another reason that emphasizes possible utilization of the technique.

Table 3. Correlation of Metrics with Iterative Improvement Gain

	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_est	0.077	0.020	0.040

Minimum power schedules after rescheduling and the power dissipation of schedules with the most favorable metric values without rescheduling are reported in Table 4. Columns 1, 2, 3 correspond to the post resource binding power dissipation for schedules with most favorable metric values. Column 4 reports the minimum power dissipation among all the different experimented schedules. Optimizing according to the metrics leads to schedules with power values very close to column 4.

Table 4. *P(m_i):* Minimum Power schedule selected by metric *i*, *Min:* Minimum power dissipation among all schedules.

	P(m1)	P(m2)	P(m3)	Min.
fft1	87.594	88.6	87.594	87.27
fft2	24.092	23.146	23.171	22.79
jctrans1	100.991	100.9916	100.9916	98.842
jctrans2	21.617	21.2882	21.2882	21.272
jdmerge1	62.653	62.653	62.144	62.144
jdmerge2	72.124	72.124	72.427	70.386
jdmerge3	39.547	38.371	38.403	38.258
jdmerge4	3.5740	3.5760	3.576	3.585
noise_est	17.952	17.694	17.952	17.602

Table 5. Running time of Metric evaluation (in msec), Iterative Power Improvement and Optimal Binding, ratio of the first 2 columns

	Metric	Iterative + Bind	Ratio
fft1	17.998	980.934	54.50239
fft2	3.144	108.52	34.51654
jctrans1	1.982	94.517	47.68769
jctrans2	2.357	96.015	40.73611
jdmerge1	3.742	152.237	40.68332
jdmerge2	17.175	1069.455	62.26812
jdmerge3	5.746	227.839	39.65176
jdmerge4	4.610	141.918	30.78482
noise_est	3.472	114.459	32.96630

7. CONCLUSIONS AND FUTURE WORK

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. Experiments with the Media Bench 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 close overall power dissipation.

Metric evaluation enables power estimation at 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 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 optimal binding and iterative improvement.

The immediate future work is to formulate scheduling algorithms that optimize these metrics. A study of the right tradeoff between post binding rescheduling and metric optimization is also imperative.

8. REFERENCES

[1] Chang J.M., Pedram M., Register Allocation and Binding for Low Power, Design Automation Conference, (DAC), 1995.

[2] Lyuh C.G., Kim T., Liu C.L., An integrated Data Path Optimization for Low Power Based on Network Flow Method, International Conference on Computer Aided Design, (ICCAD), 2001.

[3] Ku D., De Micheli G., Relative Scheduling Under Timing Constraints, Design Automation Conference, (DAC), 1990.

[4] Kruse L., Schmidt E., Jochens G., Stammermann A., Schulz A., Macii E. and Nebel W., Estimation of Lower and Upper Bounds on the Power Consumption from Scheduled Data Flow Graphs, IEEE Trans. on VLSI Systems, pp. 3-14, Feb-2001.

[5] Dasgupta A., Karri R., Simultaneous Scheduling and Binding for Power Minimization During Microarchitecture Synthesis, International Symposium on Low Power Electronics and Design, (ISLPED), 1995.

[6] Dasgupta A., and Karri R., High-Reliability, Low Energy Microarchitecture Synthesis, IEEE, TCAD, 1998.

[7] Hu T.C., Parallel Sequencing and Assembly Line Problems, Operations Research, No. 9, pp. 841- 848, 1961.

[8] Paulin P., Knight J., Force Directed Scheduling for the Behavioral Synthesis of ASIC's, IEEE Transactions on CAD, Vol. 8, No. 6, pp. 661- 679, 1989.

[9] Potasman R., Lis J., Nicolau A., Gajski D., Percolation Based Scheduling, Design Automation Conference, (DAC), 1990.

[10] Camposano R., Path-based Scheduling for Synthesis, IEEE Transactions on CAD, Vol. 10, No. 1, pp. 85 - 93, 1990.

[11] Ogrenci Memik S., Bozorgzadeh E., Kastner R., Sarrafzadeh M., A Super-Scheduler for Embedded Reconfigurable Systems, International Conference on Computer Aided Design, (ICCAD), 2001.

[12] Lee C., Potkonjak M. and Mangione-Smith W.H., MediaBench: A Tool for Evaluating and Synthesizing Multimedia and Communications Systems International Symposium on Microarchitecture, 1997.

[13] SUIF. http://suif.stanford.edu.

[14] Raghunathan A., Jha N.K., An Iterative Improvement Algorithm for Low Power Data Path Synthesis, International Conference on Computer Aided Design, (ICCAD), 1995.

[15] Gebotys C.H., Low Energy Memory and Register Using Network Flow, Design Automation Conference, (DAC), 1997.

[16] Hong S., Kim T., Bus Optimization for Low-Power Data Path Synthesis Based on Network Flow Method, International Conference on Computer Aided Design (ICCAD), November 2000.