Power Optimization for Simultaneous Scheduling and Partitioning with Multiple Voltages

Dongxin Wen Ling Wang
Department of Computer Science and Technology
Harbin Institute of Technology
Harbin, Heilongjiang, China

Yingtao Jiang Henry Selvaraj
Department of Electrical & Computer Engineering
University of Nevada, Las Vegas,
USA

Abstract: This paper presents a synthesis scheme based on simulated annealing, to minimize power consumption with resources operating at multiple voltages under timing and resource constraints. The scheme considers both scheduling and partitioning simultaneously to reduce power consumption with considering layout issues. Experimental results with a number of DSP benchmarks show that the scheme can achieve significant power reduction at the expense of running time.

Key-Words: low power, scheduling, partition, multiple voltages

1 Introduction

Low power design has become synonymous with mobile telecommunications and portable computing system designs. In order to lower power consumption and control leakage currents, for the past a few years, techniques at various levels of design (ranging from the processing technology, circuit, logic, architectural and algorithmic/behavioral levels, up to the system level) have been deployed. In this study, we focus on the power minimization at the behavioral level, where one promising approach to reduce power consumption is to use multiple supply voltages on the chip [17]. In a multiple voltage design, operations on the critical paths are assigned to the high-voltage resources to meet the timing constraints while operations on the non-critical path are assigned to low-voltage resources to reduce power consumption.

Most designs using multiple voltage supplies at the behavioral level follow a similar synthesis flow, with the emphasis is placed on scheduling and binding of the operations. In specific, the scheduling assigns the operations in the control data flow graph (CDFG) to specific time steps and binds the operations to specific functional units.

However, if only the scheduling and the binding are considered, multiple voltage synthesis design at the behavioral level can cause some physical layout problems, as illustrated in the example shown in Fig. 1.

After the schedule of all five operations (4 additions and 1 multiplication represented as nodes in Fig. 1.a) has been determined, binding will follow to assign the operations to appropriate functional units. In this particular example shown in Fig. 1, there are four adders and one multiplier (A1 through A4, M1) available. After performing ASAP scheduling and binding, nodes can be assigned to these units under specific timing constraints just as shown in Fig.1.a. After layout these functional units, we shall be able to see that longer interconnections exist between function units (A4, A3) and (M1, A4) as they are placed far apart. Even worse, five level shifters are needed as modules (A1, M1) (A1, A2), (A3, A4), (A2, A3) and (M1, A4) are assigned to different voltages in Fig.. 1.b.

On the other hand, by taking the layout issues into consideration at the behavior level, a different binding scheme may prevail. In fact, without changing timing constraints, only exchanging A3 with A2 (operation 3 is bounded to A3, 4 to A2, as shown in Fig. 1.c) may result in dramatically different layout. In this case, shorter interconnections and less level shifters (only 3 are actually needed) are possible, when all five operations are partitioned into three larger groups, namely (A1, A3), (A2, A4) and M1, and units that belong to the same partition can be placed close to each other, as shown in Fig.. 1.d.. That is, scheduling and binding with considering layout issue could result in reduced length of interconnections and less number of level shifters in the final layout.

Realizing the importance of partitioning at the behavioral level, we proposed the schemes to minimize power consumption under the timing constraints [19] and under the resource constraints [17], with each scheme performing scheduling, partitioning and binding in a sequential manner..

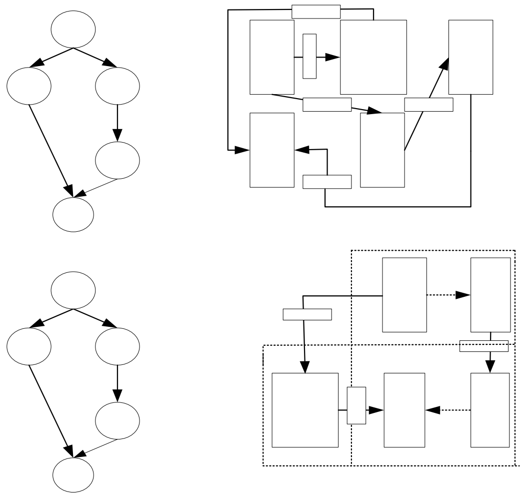

Fig..1 Illustrated Example..

Since scheduling and partitioning are not independent from each other, it is more desirable to perform these two tasks simultaneously to achieve greater power reduction. However, the complexity of performing two tasks simultaneously demands good algorithms to make a good balance between the computation time and the performance. In [16], a fast tabu search (TS)-based algorithm was proposed, and comparative studies show that this algorithm can achieve higher power reduction than those sequential algorithms [17][19].

In this paper, a simulated-annealing-based algorithm is proposed, which is also capable of performing the scheduling and the partitioning simultaneously. Experimental results have indicated that this algorithm can achieve even higher power reduction at a cost of more running time.

2 Problem Statement

2.1 Notions

Definition 1 (Data Flow Graph). A data flow graph DFG is a directed acyclic graph $G = (V, E)$, where V is a set of nodes, and E is a set of edges between nodes. Here each node represents an operation, and a directed edge from node v_i to node v_j means execution of v_i must precede that of v_j..

Definition 2 (Timing Constraint). Timing constraint is the time available to execute the nodes in a DFG.

Definition 3 (Resource Constraint). Resource constraint is a set of allowed functional units that should run at specified supply voltages. Note that we could have several instances of the same functional unit running at different/same supply voltages..

Definition 4 (Spatial Voltage Cluster). A spatial voltage cluster is a set of nodes that operate at the same supply voltage. The resources assigned to the nodes in a spatial voltage cluster shall be placed closer each other in the layout to minimize the communication costs.

2.2 Problem Statement

The multiple voltage scheduling and partitioning (MVSP) problems under the timing and the resource constraints are defined as follows:

Given a DFG and timing and resource constraints, find a schedule and a partition that satisfy the given timing and resource constraints, with minimum power consumption.

A solution to above stated problem must satisfy the following requirements:

· each node in the DFG is assigned to a specific resource at its corresponding control step and

· all the nodes are partitioned into spatial voltage cluster(s).

Here, a solution S is defined as:

$$S = \{\langle v_1, r_1, c_1\rangle, \langle v_2, r_2, c_2\rangle, \ldots \langle v_i, r_i, c_i\rangle, \ldots \langle v_n, r_n, c_n\rangle\} \quad (1)$$

where (i) node $v_i \in V$, $n = \|V\|$, (ii) r_i is the resource assigned to node v_i, and (iii) c_i is the cluster to which node v_i belongs and $c_i \in C$, C is the set of all spatial voltage clusters.

Our goal is to find a solution S that can minimize the following objective function:

$$\sum P_{\text{int}}^{(i,j)}(S) + \sum_{k=1}^{n} P_{res}^{k}(S) \qquad (2)$$

Subject to

$$T \le T_C \text{ and } R \le R_C \qquad (3)$$

where (1) $P_{\text{int}}^{(i,j)}(S)$ is the power consumed by the interconnections between clusters c_i and c_j in the solution S (2) P_{res}^{k} is the power consumed by kth functional unit. (3) T is the total sched total number of asule time, (4) T_C is the timing constraint, (5) R is thesigned resources at each control step, (6) R_C is the number of the available resources (resource constraints) at each control step, and (7) n is the number of nodes.

Lemma: Multiple voltage scheduling problem is NP-complete.

+1 A1

*2

M 1

+3 A 2

3.3V A1

2.4V A4

+4 A 3

+5 A 4

(a)

Theorem: Multiple voltage scheduling and partitioning problem is NP-complete.
Proof: It can be readily shown from the Lemma.

3 Simulated Annealing Algorithm

In this section, the synthesis scheme is presented by using simulated annealing to solve the multiple voltage scheduling and partitioning problem, since simulated annealing is suitable for parallel computing.

Simulated annealing is inspired by an annealing process, where matter is first melted, and then slowly cooled in a controlled way to obtain a certain arrangement of the atoms.. When the temperature is high, atoms can occasionally move to states with higher energy, but then, as the temperature drops, the probability of such moves is reduced.

The main advantage of simulated annealing is that it is able to improve upon the relatively poor performance of local search by simply replacing the deterministic (strict improvement) acceptance criterion by a stochastic criterion.

The inputs to the proposed algorithm consist of the DFG representation of a circuit, the timing constraints alone, or the resource constraints alone, or the timing and resource constraints, a design library with fully characterized resources. The framework of simulated annealing is shown as follows.

```
Input: DFG represented as G(V,E); // see Def. 1
Definitions:
$S$ : a feasible solution as Eq.(1);
$F$: Objective function as Eq.(2);
$N(S)$: Neighborhood of $S$; $T$: Temperature;
$r$: control factor; $Max$: number of iterations;
Initialization:
$i$=0; Generate an initial feasible solution $S_i$ ;
Set an initial temperature $T$>0;
$bests = S_i$ ; $bestcost = F(bestS)$; $besti$ = 0;
Body:
while( count< $Max$){// count is the number of iterations
    Choose a random solution $S_{i+1}$ from N( $S_i$ );
    Let $D$=$F$( $S_{i+1}$ )-$F$( $S_i$ );
    if $D$<0 //uphill move $S_i = S_{i+1}$ ;
    if $D \geq 0$ //downhill move
            $S_i = S_{i+1}$ with probability exp($D/T$);
   $T$=$r * T$; // reduce temperature
   Count=Count+1;
}
Output: the best $S$' visited
```

- **Step 1. Scheduling and partitioning by simulated annealing**

There are two main issues related to the adaptation of this general approach to the multiple voltage scheduling and partitioning problem. They are given as followings:

- Construct *move* and *neighborhood* structure:

A *move* from solution S to another S* is represented by $(v_i,(r_i,c_l),(r_i^*,c_l^*))$ where i) v_i is the only node in S that its cluster and resource are to be modified, ii) resource r_i is assigned to v_i while node v_i belongs to cluster c_i in S, and iii) resource r_i^* is assigned to v_i in S***,** and node v_i belongs to a different cluster c_i^* in S*. Note that resources r_i and r_i^* are of the same function type but operate at different supply voltages. As resource r_i^* operates at voltage V_j , c_i^* can be any of the clusters $c_{j,1},c_{j,2},...,c_{j,m}$ all having the same voltage V_j . In this way, the search space includes all the clusters of interest. The move is required to satisfy timing and resource constraints.

Neighborhood of a solution S, denoted as N(S), contains all the solutions obtained by a move of S.

- Design the *cooling schedule*:

As for the *cooling schedule* design, we made the following decisions.

1)The initial *temperature* T0 is chosen so that the initial accepting rate is in the range (0,1).

2)Temperature T is chosen as *A*/*log*(count+1), where count is the number of iterations and *A* is the parameter.

All the parameters for the simulated annealing algorithm are not independent. We adjust the parameters of our simulated annealing for each benchmark. We repeat the process until no perturbation of the parameters can improve the performance.

- **Step 2. Resource Binding**

After obtaining a satisfactory schedule and partition from the previous scheduling and portioning step, resource binding takes place (Fig. 1.c). As the resources assigned to the nodes with the same voltage supply will be placed close to each other in the physical layout, the interconnection power consumption between the clusters are significantly higher than that inside the clusters. Therefore, the binding algorithm proposed in 0 can be applied for each cluster to maximize the resource sharing among the nodes inside each cluster in order to reduce the interconnection cost.

The binding algorithm is listed here.

```
Input: bestS, R: a set of resources in design library;
Definitions:
st: the starting time of the node in bestS;
et: the ending time of the node in bestS;
V: the set of bounded nodes in a cluster;
L: the list of unbounded nodes in a cluster;
Body:
for ( each cluster in bestS ){
   Sort nodes in a list L in ascending order by their st;
   while(some nodes have not been bound){
      V=0;
      t=0;
      while (there exists a node in L such that its st>t){
          v=First node in the list L with its st>t;
          V=V+{v};
          t= node v ending time et;
          L=L-{v}; //Delete v from L;
      }
      Bind nodes in V to resources in R;
   }
}
```

4 Experimental Results

In this section, we present the results obtained by applying our algorithm on some high-level synthesis benchmarks (Lattice, Differential Equation, Ellipf, Wdf7, Volterra, 7th order IIR filter, Wavelet and DCT). The input format is VHDL code, which is translated to DFG (data flow graph) by CDFG tools developed by electrical engineering department of Seoul National University [20]. The output format is RTL which can be interfaced with commercial CAD tools. The proposed algorithm is coded in C and runs on a Linux workstation.

The number of nodes of each circuit is reported in column *N* of Table 1.

The design library, adopted from [2], as shown in Table 2, consists of two types of functional units: adder and multiplier, and both can operate in any of the four voltage levels: 5.0V, 3.3V, 2.4V, and 1.5V. The delay and the power consumption of each function unit have been fully characterized.

The timing constraint *Tc* (column *Tc* in Table 1) is set as 2T, where *T* is the delay obtained by ASAP scheduling when all the sources operate at 5V (column *Tc* in Table 1). The resource constraint R is set as an integer. For instance, if the resource constraints *R*=1, it means that, at each control step, maximum 4 multipliers (each operates at 5V, 3.3V, 2.4V, 1.5V) and 4 adders (each operates at 5V, 3.3V, 2.4V, 1.5V) shall be allowed. In all the experiments, the number of iterations is set to 1000.

Fig. 2 and 3 show the convergence of power consumption obtained by the proposed algorithm for DCT filter benchmark with *A*=10, and A=1000, respectively, where *P* is power consumption and *N* is the iteration number. It is seen that faster convergence is achieved when the parameter A is set as 10. As a matter of fact, this issue of the algorithm has also been observed for all other benchmark circuits.

The experimental results are tabulated in Table 1 with A=10. In Table 1, column E_{fun}^{5} reports the energy consumption when all the functional units operate at 5V supply voltage. Column E_{fun}_s and E_{fun}_t report the energy consumption obtained by simulated annealing algorithm (SA) and tabu search algorithm (TA), respectively. Column I_s and I_t report cutsize obtained by SA and TS, respectively. Column Nc_s and Nc_t report the number of clusters obtained by SA and TS, respectively. Column CPU_s and CPU_t report CPU time obtained by SA and TS, respectively.

From the experiment results, the following conclusions are concluded:

- It is shown that TS is average faster than SA and SA can achieve more power consumption reduction than TA except for Volterra and wavelet circuits. However, if A is set as 100, power consumption reduction by SA for Volterra and wavelet circuits is more than them by TS. Hence, the algorithm is parameter sensitive.
- At the same time adaptation of the SA strategy for a particular problem is relatively easy. On the contrary, development of a TS algorithm is more complex and has to consider particular aspects of the given problem.
- Running time obtained with TS is definitely superior in comparison to those given by SA. On the contrary, power reduction obtained with SA is more than those obtained by TS. This conclusion is important in the context, as no TS based or SA based Multiple voltage synthesis scheme has yet been reported.

5 Conclusion

In this paper, we have proposed a scheme based on simulated annealing algorithm to minimize power consumption under timing and resource constraints, which perform scheduling and partitioning simultaneously

Simulated annealing algorithm is applied to a number of DSP benchmarks. From these experimental results, it can be concluded that the resource power consumption obtained by TS is slightly more than those given by SA. On the contrary, running time obtained with SA is more than those obtained by TS.

References:

[1] P. Chandrakasan, S. Sheng, and R. W. Brodersen, *"Low Power CMOS Digital Design," IEEE J. Solid-State Circuits*. vol. 27, pp.473-483, Apr. 1992.

[2] J. M. Chang, and M. Pedram, "Energy Minimization Using Multiple Supply Voltages," *IEEE Trans. VLSI*, vol.5, pp.157-162, Dec. 1997.

[3]M. R. Garey and D. S. Johnson, Computers and Intractability: A guide to the Theory of NP-Completeness, Freeman, San Francisco, CA.

[4]F. Glover and C. Mcmillan, "The General Employee Scheduling Problem: an Integration of Management Science and Artificial Intelligence," *Computers Oper. Res.*, vol. 13, no. 3, pp.563-593, 1987.

[5]A. Hertz, and D. De Werra, "The Tabu Search Metaheuristic: How We Used It." *Ann. Mathematics and Artificial Intelligence*, vol. 1, pp.111-121, 1991.

[6]M. C. Johnson and K. Roy, "Datapath Scheduling with Multiple Supply Voltages and Level Converters," *ACM Trans. Design Auto. Electronic Systems*, vol.2, pp. 227-248, July 1997.

[7]Y. R Lin and C. T. Hwang, "Scheduling Techniques for Variable Voltage Low Power Designs," *ACM Trans. Design Auto. Electronic Systems*, vol.2, pp. 81-97, Apr. 1997.

[8]A. Manzak and C. Chakrabarti, "A Lower Power Scheduling Scheme with Resources Operating at Multiple Voltages," *IEEE Tran. VLSI*, vol.10, pp. 6-14, Feb. 2002

[9]R. Martin and J. Knight, "Power Profiler: Optimizing ASIC's Power Consumption at the Behavioral Level," *Proc. IEEE/ACM Design Automat. Conf.*, pp. 42-47, 1995.

[10]R. Mehra, L. M. Guerra, and J. M. Rabaey, "Low Power Architectural Synthesis and the Impact of Exploiting Locality," *J. VLSI. Signal Processing*, vol.13, pp.239-258, 1996.

[11]E. L. Mooney and R. L. Rardin, "Tabu Search for a Class of Scheduling Problems," *Ann. Oper. Res.*, vol. 41, pp.253-278, 1993.

[12]S. Raje and M. Sarrafzadeh, "Variable Voltage Scheduling," *Proc. Int. Symp. Low Power Design*, pp.9-14, Apr 1995.

[13]S. Raje and M. Sarrafzadeh, "Scheduling with Multiple voltages," *Integr. VLSI J.*, pp.37-60, Oct.1997.

[14]W. T. Shiue and C. Chaitali, "Low Power Scheduling with Resources Operating at Multiple Voltages," *IEEE Trans. CASII*, vol.47, pp.536-543, 2000.

[15]J. Skorin Kapov, "Tabu Search Applied to the Quadratic Assignment Problem", *ORSA J. Computing*, vol.2, pp.33-45, 1989.

[16]L. Wang, Y. Jiang, and H. Selvaraj, "Synthesis Scheme for Low Power Designs with Multiple Supply Voltages by Tabu Search," *Proc. ISCAS* 2004.

[17]L. Wang, Y. Jiang and H. Selvaraj, "Scheduling and Optimal Voltage Selection with Multiple Supply Voltages under Resources Constraints", *Proc. Int. VLSI Conf.*, pp. 272-278, June 23-26, 2003

[18]B. Hajek, "Cooling Schedules for Optimal Annealing," *Math. Oper. Res.*, vol.13, no.3, pp.53-61, 1990.

[19]L. Wang, Y. Jiang and H. Selvaraj, "A Synthesis Scheme for Low Power Designs with Multiple Voltages under Timing Constraints," *Proc. NASA 11th VLSI Symp.*, 2003, Idaho.

[20]http://poppy.snu.ac.kr/CDFG

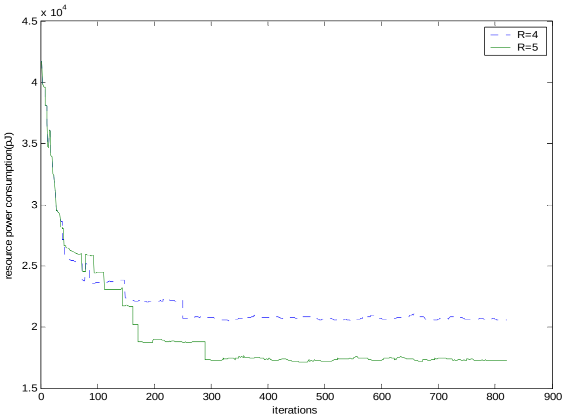

Fig. 2 DCT Circuit (power consumption P vs. number of iterations N) under different resource constraints with A=10

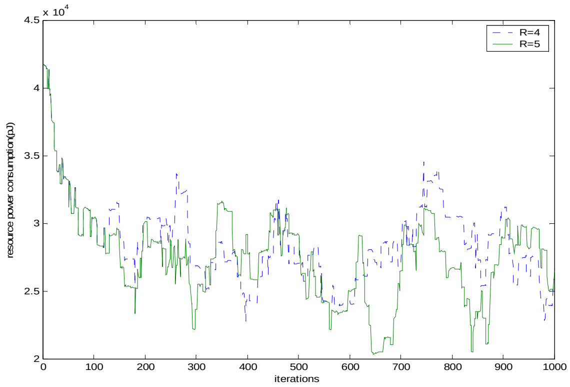

Fig.3 DCT Circuit (power consumption P vs. number of iterations N) under different resource constraints with A=1000

Table 1 Experimental Results on Benchmarks with A=10

Bench-mark	Nn	Tc	R	E^{5}_{fun}	E_{fun_s} (pJ)	E_{fun_t} (PJ)	I_s	I_t	Nc_ s	Nc_ t	CPU_s (second)	CPU_t (second)
Lattice	22	48	1	23598	11347.3	13467.5	37	34	5	5	0.72	0.06
			2		8559.1	10975.9	39	38	8	4	0.73	0.41
Diffeq	10	42	1	15614	7188.5	15152.6	29	25	4	4	0.87	0.18
			2		6070.1	12766.6	25	25	5	3	1.01	0.19
Ellipf	37	58	1	23100	13550.9	13463.7	62	51	4	5	1.19	0.24
			2		8984.8	12235.4	60	56	7	4	1.14	0.14
			3		9663.9	14046.7	56	57	5	4	1.3	0.15
			4		9170.2	11447.5	56	58	5	4	1.4	0.23
Iir7	36	40	2	39212	20015.7	20332.1	70	61	6	4	1.26	0.22
			3		15398.6	15740.6	61	64	4	4	1.41	0.64
			4		13471.5	14678.9	67	65	9	4	1.54	0.25
Wdf7	50	60	2	49464	22250.9	24604	79	72	4	4	1.6	0.66
			3		18558.9	19110.4	74	73	4	4	1.4	0.89
			4		14213.9	19827.2	80	71	10	4	1.42	0.84
			5		14851	18448	72	71	4	4	1.57	0.9
Dct	42	34	4	43132	20602.1	21977.8	75	79	4	4	1.38	0.65
			5		17350.9	20165.7	68	72	4	4	1.32	0.65
volterra	32	40	3	43748	22673.7	16814.8	60	55	6	4	1.05	0.37
			4		19186.5	15122.7	52	54	4	5	0.94	0.86
wavelet	67	42	4	73180	34434.9	28184.4	122	116	4	4	2.37	1.05
			5		31370.7	23596.5	118	114	4	4	2.28	1.15

Table 2 Design library [2]

	5.0V		3.3V		2.4V	
resource	Delay	Power (pJ)	Delay	Power (pJ)	Delay	Power (pJ)
Adder	1	118.0	2	51.4	3	27.2
Substracter	1	118.0	2	51.4	3	27.2
Multiplier	5	2504	9	1090	15	576.9