Power Constrained High-Level Synthesis of Battery Powered Digital Systems

S.F. Nielsen and J. Madsen
Informatics and Mathematical Modelling
Technical University of Denmark
{sfn,jan}@imm.dtu.dk

Abstract

We present a high-level synthesis algorithm solving the combined scheduling, allocation and binding problem minimizing area under both latency and maximum power per clock-cycle constraints. Our approach eliminates the large power spikes, resulting in an increased battery lifetime, a property of outmost importance for battery powered embedded systems. Our approach extends the partial-clique partitioning algorithm of [3] by introducing power awareness through a heuristic algorithm which bounds the design space to those of power feasible schedules. We have applied our algorithm on a set of dataflow graphs and investigated the impact on circuit area when applying different power constraints.

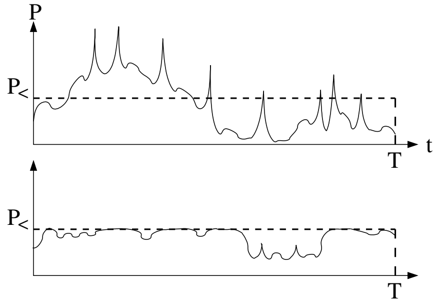

Figure 1. Undesired power schedule (top). Desired power schedule (bottom).

1. Introduction

Our target applications are low-cost portable embedded systems. Today, consumers demand portable applications so tiny that they go virtually undetected when not in use. An interesting aspect of this application area is the low-cost issue which puts focus on reducing the overall system cost, eg. a requirement to select a low-priced (low-quality) battery over a high-priced (high-quality) battery. Now, the amount of total energy/charge available from a battery, and thus its life-time, depends strongly on the current profile of the application [1, 2]. In particular if the peak-current exceeds a maximum-threshold the life-time starts dropping dramatically, this effect is more dominant on batteries of low quality, where up to a 20-30 percent extension of life-time has been reported when designing for battery powered systems [1]. In figure 1 is shown both an undesirable and a more desirable schedule.

So far the main efforts in low-power synthesis can be divided into two groups: *(a) Low level allocation and assignment* [4, 5, 6, 7]: Here the goal is to combine functional units and operations in such a way that the internal switching activity of the functional units is minimized. *(b) Task level scheduling* [1, 2, 8]: Here the goal is to schedule tasks in such an order the peak system power is minimized. The majority of these algorithms are either based on meta-heuristic algorithms, or two-step algorithms [1, 2] where in step one a traditional time constrained schedule is constructed and in step two the schedule is reordered to meet the power constraint.

In this paper we present a heuristic synthesis algorithm which solves: (i) scheduling, (ii) allocation and (iii) assignment simultaneously under both a time and power constraint. This enables us to expand the exploration of the design space to include different types of functional units eg. the speed and energy usage of an operator can be traded versus the area of the operator.

2. Power Heuristic Scheduling

The main idea of our algorithm is to heuristically “stretch” the classical **asap** schedule to fit the power constraint, ie. to schedule the operators as fast as possible, but only if there is power available meaning some operators will be delayed additional cycles. The power constrained **asap**

1530-1591/03 $17.00 © 2003 IEEE

Module	Oprs	Area	Clk-cyc.	P
add	$\{+\}$	87	1	2.5
sub	$\{-\}$	87	1	2.5
comp	$\{>\}$	8	1	2.5
ALU	$\{+,-,>\}$	97	1	2.5
Mult (ser.)	$\{*\}$	103	4	2.7
Mult (par.)	$\{*\}$	339	2	8.1
input	imp	16	1	0.2
output	xpt	16	1	1.7

Table 1. Functional unit library.

scheduling algorithm, **pasap** ($P_<$), is as follows:

Initialize: Schedule source start-time to zero and initialize the execution offset o_i (cycles) to zero for all operators.

step 1: Pick an unscheduled operator v_i

step 2: If v_i has unscheduled predecessors, goto 4.

step 3: if there is power available in the execution time interval $[(t_i+o_i)..(t_i+o_i+d_i)]$, where d_i is the execution delay of v_i and $t_i = \max\{t_j + d_j\}\ \forall v_j \rightarrow v_i$, schedule operation i at time t_i, otherwise increase o_i by one.

step 4: If unscheduled operators, goto step 1.

We have enhanced the formulation of a valid "time-extendend compatibility graph" ($V1$) in [3] to include power constraints using the **pasap** and its time-revesed **palap** algorithms. Then the solution to the synthesis problem with the minimum area and using least interconnect is the problem of finding the **Partial minimal cost clique partitioning of $V1$ which does not violate the power constraint**. As in [3] we will heuristically solve the clique partitioning problem, through a greedy approach ie. evaluate the $V1$ graph and pick a "best" decision, which is then scheduled, allocated and assigned and then repeat the process until no operators are left. During this we need to ensure feasibility, as **pasap** and **palap** are heuristic algorithms they depend on what operators have been scheduled, therefore a sequence of assignments might cause the deletion of unscheduled operators, causing an invalid schedule. The solution is to backtrack one step and lock the start time of all unscheduled operators to the **pasap** schedule (which was valid) and then continue.

We have benchmarked the algorithm on some CDFGs, where we have investigated the area of the resulting circuits as a function of time and power constraints. The results are shown in figure 2. The FU library used in the tests is shown in table 1.

3. Conclusion

In this paper we have presented an algorithm for time and power constrained synthesis of digital circuits. We have applied the algorithm on the traditional synthesis benchmark sets and investigated different regions in the time-power-constraint space. For our benchmarks we have found that we are able to trade in a small amount of area to obtain a solution which fits our power requirements.

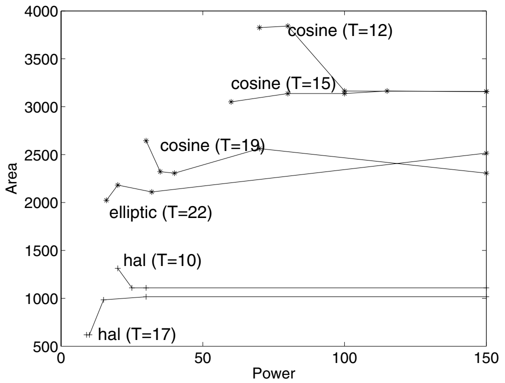

Figure 2. Power vs. area under different time constraints for our CDFGs.

References

[1] J. Luo and N. K. Jha. Battery-Aware Static Scheduling for Distributed Real-time Embedded Systems. In proceedings of 38th Design Automation Conference, 2001. Page(s): 444 -449.

[2] K.Lahiri, A. Raghunathan, S. Dey. Communication Architecture Based Power Management for Battery Efficient System Design. In proceedings of Design, Automation and Test in Europe. 2002. Page(s): 691 -696

[3] J. Jou; S. Kuang; R. Chen. Clique Partitioning Based Integrated Architecture Synthesis for VLSI Chips. In proceedings of International Symposium on VLSI Technology, Systems, and Applications, 1993, Page(s): 58 -62.

[4] Y. Choi and T. Kim. An Efficient Low-Power Binding Algorithm in High-Level Synthesis. Circuits and Systems, 2002. ISCAS 2002. IEEE International Symposium on , Volume: 4 , 2002 Page(s): 321 -324.

[5] M. S. Bright and T. Arslan. Synthesis of Low-Power DSP Systems Using a Genetic Algorithm. In IEEE Transactions on Evolutionary Computation, vol. 5, no. 1, Feb 2001.Page(s): 27 -40.

[6] F. Gruian and K. Kuchcinski. Operator Binding and Scheduling for Low Power Using Constraint Logic Programming. Euromicro Conference, 1998. Proceedings. 24th , Volume: 1 , 1998 Page(s): 83 -90 vol.1.

[7] M. Bhardwaj, R. Min, A. Chandrakasan. Power-Aware Systems. Very Large Scale Integration (VLSI) Systems, IEEE Transactions on , Volume: 9 Issue: 6 , Dec. 2001 Page(s): 757 -772.

[8] J. Liu, P. H. Chou, N. Bagherzadeh, F. Kurdahi. Power-Aware Scheduling under Timing Constraint for Mission-Critical Embedded Systems. In proceedings of the 38th Design Automation Conference, 2001. Page(s): 840 -845.