

# POWER MINIMIZATION IN DSP APPLICATION SPECIFIC SYSTEMS USING ALGORITHM SELECTION

*Miodrag Potkonjak*

C&C Research Laboratories, NEC USA, Princeton, NJ

*Jan M. Rabaey*

Department of EECS, University of California, Berkeley, CA

**ABSTRACT:** We introduce the algorithm selection problem for power minimization. After demonstrating a high impact of this synthesis task on the power consumption of the final implementation using case study, we studied its computational complexity. We present an efficient optimization intensive algorithm for power minimization using algorithm selection. On several DSP examples more than an order of magnitude reduction in power is demonstrated.

## 1. Motivation

Recently, the research and development in VLSI DSP domain received a new strong impetus due to two important emerging topics: power optimization [Cha92] and system level design [Kal93]. Power minimization is widely accepted as key enabling design technology for support of portable applications, by far the fastest growing DSP market segment [Cha92]. There is a wide consensus among CAD researchers and system designers that rapidly evolving hardware-software codesign research is a backbone of future CAD technology and design methodology [Kal93, Cho92, Ver94, Sri91].

We study the hardware-software codesign task from the point of view of a DSP application developer. It is recognized that for a specified functionality of the application, the user not only has a choice of implementation platform, but also may select among a variety of functionally equivalent algorithms [Pot94a, Pot94b]. As it is documented in the rest of this section, even on a small example, a properly

selected algorithm reduces a power of implementation by more than an order of magnitude for a given level of performance.

We first synthesized, under the identical throughput constraints using the Hyper high level synthesis system [Rab91], the following eight DCT algorithms: Lee - Lee's recursive sparse matrix factorization algorithm, Wang-Suehiro-Hatori's version of the Wang planar rotation-based sparse matrix factorization DCT, DIT - recursive decimation in time algorithm, DIF - recursive decimation in frequency algorithm, QR - QR decomposition based hybrid planar rotation algorithm, givens - Givens rotation-based algorithm, MCM - automatically synthesized algorithm, which applies only one multiple constant multiplication transformation on the generic and direct- the direct, generic definition of DCT algorithm [Rao90].

Table 1 shows the estimates of power requirements. We see that difference of required energy per sample by a factor of 6.5 (79.57 nJ per sample (direct) vs. 12.19 nJ per sample (lee) ) for the throughput constraints as required by the H261 standard. The results presented in the second column assume no voltage scaling. After the application of the voltage scaling technique as proposed in [Cha92] for the same throughput the power of the most efficient structure (mcm) is reduced to 3.93 nJ/sample, representing the improvement by a factor of more than 20 times over the non-optimized direct form DCT transform. It is interesting to note that the reduction in power consumption is achieved without increase in area

[Pot94].

algorithm	power [nJ/sample]	T critical path [nsec]	voltage scaling optimized power [nJ/S]
direct	79.57	380	16.84
DIF	13.39	600	5.31
DIT	16.26	620	6.66
wang	20.77	600	8.24
lee	12.19	560	4.24
QR	16.68	560	5.81
givens	27.17	600	10.78
mcm	21.64	340	3.93

**Table 1:** Power consumption for two dimensional 8X8 DCT. All numbers are for 1.2 micron technology.

The similar study was conducted on six different structures (direct form II, cascade, parallel, continued fraction, Gray-Markel's ladder form, and wave digital form) of the 8th order Avenhaus IIR bandpass filter [Cro75]. The maximally pipelined cascade form consumes more than 20 times less power than the direct form II of the filter [Pot94]. The importance of algorithm selection compounded with voltage scaling for power optimization is apparent.

Finally, we want to emphasize that for majority of important DSP algorithms numerous functionally equivalent algorithms are readily available for a given application [Bla85, Mit93]. The most important generic issues related to the algorithm selection problem and their relationship to algorithmic transformations and algorithm design are presented in [Pot94a, Pot94b].

## 2. Power Optimization Using Algorithm Selection: Problem Formulation and Computational Complexity

Before we formulate the power optimization using algorithm selection problem, we briefly outline the power modelling methodology. In CMOS technology, there are three sources of power dissipation: switching, short-circuit and leakage currents. The switching component, however, is the only one that can not be made negligible if proper design

techniques are followed. The switching power for a CMOS gate with a load capacitor,  $C_L$ , is given by the following formula [Cha92]:

$$P_{\text{switching}} = \text{Energy per Transition} \cdot f = C_{\text{avg}} \cdot V_{\text{dd}}^2 \cdot f = (p_t C_L) \cdot V_{\text{dd}}^2 \cdot f$$

where  $f$  is the clock frequency, and  $p_t$  is the probability of a power consuming transition (0 -> 1).

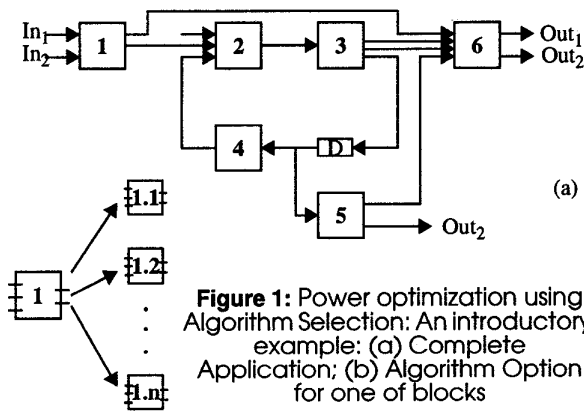
Starting from this formula, the experimentally validated behavioral level macroscopic model of power consumption is developed for ASIC custom designs [Cha92]. The model states that the power consumption is a quadratic function of the voltage and a linear function of effective capacitance. The length of the clock cycle is inversely proportional to the voltage; this relationship is accurately modeled using the Neville's algorithm for rational function interpolation and extrapolation [Cha95].

The effective capacitance is estimated using the statistically validated model which connects the behavioral level primitives and the available time to the power consumption of the final implementation [Cha92]. Therefore, by selecting an algorithm for a given application one can reduce power by either reducing the supply voltage at the expense of the longer execution time or by selecting the algorithm where the effective capacitance is lower (for examples, some algorithm for a given application have smaller number of operations and data transfers).

Until now we considered only applications which have only one procedure. Of course, a typical DSP application has significantly more complex structure.

Figure 1 shows illustrates the power minimization using algorithm selection problem for a common DSP scenario. The assumed computational model is synchronous data flow [Lee87] and a single thread of control with the synchronization at the beginning and the end of each block.

The overall application is depicted by a number of basic blocks which are interconnected in a specific, application dictated, manner. For each block, there are a number of different CDFGs (corresponding to different algorithms) as shown in Figure 1b. The number of options for block  $B_i$  is denoted by  $n_i$ . Each block is implemented on the corresponding separate platform. (We use in the rest of the paper custom ASIC implementation as provided by the Hyper high level synthesis system [Pot91], but the methodology and synthesis algorithms are general and can be applied on an arbitrary set of implementation platforms.) The goal is to select for each block a CDFG, so that the overall energy consumption per sample is minimized, while the required timing constraints (e.g. throughput) are satisfied.



**Figure 1:** Power optimization using Algorithm Selection: An introductory example: (a) Complete Application; (b) Algorithm Option for one of blocks

We proved that power optimization using algorithm selection is NP-complete optimization problem by using the polynomial transformation method from the equal subset problem [Pot94]. It is interesting to note that the power minimization using algorithm selection problem is NP-complete problem even when for each block only two algorithmic options are available, each block has only two inputs and two outputs, and each blocks sends data to only one block and receives data from only one block.

### 3. Power Optimization Using Algorithm Selection: Optimization Algorithm

The complexity of the power minimization using algorithm selection problem has two components. The problem is not just combinatorially intractable, but it is also associated with exploring a complex relationship between several degrees of freedom early in the synthesis process. For example, it is required to simultaneously consider different algorithmic options for each block and trade-offs between their timing and area (effective capacitance) requirements.

For the power optimization using algorithm selection problem we developed the marginal utility-based iterative optimization algorithm. The optimization (synthesis) algorithm can be described using the following pseudocode:

#### Power Optimization Using Algorithm Selection:

1. *Preprocessing Step ();*
2. *Initial Algorithm Selection();*  
*while there is a spare time to be allocated and power improvement is possible*
3. *Find Marginal Gain for each block();*
4. *Reselect the best algorithm for the block with the highest Marginal Gain Improvement();*

The preprocessing step provides characterization of each block and each algorithm and an initial solution. It has two phases. In the first phase for each block and each algorithm the table with information about the power consumption for all feasible available times and corresponding voltages is calculated using the Hyper estimation and synthesis tools [Rab91, Cha92]. Note that only voltages between 1V and 6 V are considered during this phase. In the second phase of the preprocessing step for each block the algorithm with the shortest critical path is preliminary selected as the initial solution.

If there is any unused time difference

between the user specified available time and the current critical path, for each block marginal gain (potential for the reduction of the power consumption) is calculated. The marginal gain has two components: positive and negative. The positive component is equal to the reduction in power which is achieved by selecting the best available algorithm (with the smallest power requirements) for the new available time for a given block. The new time is one clock cycle longer than the current available time for the block. The negative cost is equal to the sum of positive components of the marginal gains for all blocks in the transitive fan-out and transitive fan-in of the block. Only steps which reduce the current power budget are accepted. The marginal gain-based step of the optimization algorithm is repeated as long as there exists a block for which an additional clock cycle can be allocated without violating the throughput constraint.

#### 4. Experimental Results

We applied the marginal utility-based algorithm for algorithm selection on three DSP example Table 1 illustrates the effectiveness of the power optimization using algorithm selection on one audio (LMS DCT transform domain filter) and two video (NTSC formatter and DPCM coder) applications. The average reduction of power is by a factor of 129.9 times compared to the worst possible implementation.

Example	Worst Case Power [nJ/sample]	Optimized Power [nJ/sample]	Improvement Factor
LMS DCT filter	522	6.02	86.7
NTSC formatter	1557	9.87	157.7
DPCM coder	1201	8.26	145.4

**Table 2:** Optimizing Power using Algorithm Selection: Experimental Result. The power reduction is achieved without alternation of the initial throughputs.

#### 5. Conclusion

As a part of an effort to develop optimization intensive design methodology for system level

design, we introduced the algorithm selection problem for power minimization. After demonstrating a high impact of this synthesis task on the power consumption of the final implementation using case study, we studied its computational complexity. We introduced an efficient optimization intensive algorithm for power minimization using algorithm selection. On three DSP examples more than an order of magnitude reduction in power is demonstrated.

#### 6. References

- [Bla85] R.E. Blahut: "Fast Algorithms for Digital Signal Processing", Addison-Wesley, 1985.
- [Cha92] A.P. Chandrakasan, et. al.: "Hyper-LP: A Design System for Power Minimization using Architectural Transformations", IEEE International Conference on Computer-Aided Design, Santa Clara, CA, pp. 300-303, November 1992.
- [Cho92] P. Chou, R. Ortega, G. Borriello: "Synthesis of Mixed Hardware-Software Interfaces in Microcontroller-Based Systems", ICCAD-92, pp. 488-495, 1992.
- [Cro75] R.E. Crochiere, A. V. Oppenheim: "Analysis of Linear Networks", Proceeding of the IEEE, Vol. 63, No. 4, pp. 581-595, 1975.
- [Kal93] A. Kalavade, E.A. Lee: "A Hardware-Software Code-sign Methodology for DSP Applications", IEEE Design & Test of Computers, Vol. 10, No. 3, pp. 16-28, 1993.
- [Lee87] E. A. Lee and D. G. Messerschmitt: "Static Scheduling of Synchronous Dataflow Programs for Digital Signal Processing", IEEE Trans. on Computers, Vol. 36, No. 1, pp. 24-35, 1987.
- [Mit93] S.K. Mitra, J.F. Kaiser: "Handbook for Digital Signal Processing", John Wiley & Sons, Inc., New York, NY, 1993
- [Pot94a] M. Potkonjak, J.M. Rabaey: "Algorithm Selection: A quantitative computation-intensive optimization approach", ICCAD94, pp., 1994.
- [Pot94b] M. Potkonjak, J. Rabaey: "Power Minimization in DSP Application Specific Systems Using Algorithm Selection", Technical Report, NEC USA, 1994.
- [Rab91] J. Rabaey, et. al.: "Fast Prototyping of Data Path Intensive Architecture", IEEE Design and Test, Vol. 8, No. 2, pp. 40-51, 1991.
- [Rao90] K.R. Rao, P. Yip: "Discrete Cosine Transform", Academic Press, Inc., San Diego, CA 1990
- [Sri92] M.B. Srivastava, R. Brodersen: "Rapid-Prototyping of Hardware and Software in a Unified Framework", ICCAD-92, pp. 152-155, 1992.
- [Ver94] I. Verbauwhede, J. Rabaey: "Synthesis of Real-Time Systems: Solutions and Challenges", to appear in Journal of VLSI Signal processing, 1994.