DESIGN GUIDANCE IN THE POWER DIMENSION

Jan Rabaey, Lisa Guerra, and Renu Mehra
Department of EECS, University of California, Berkeley, USA

ABSTRACT

This work proposes an approach for high level design guidance for low power using properties of given algorithms and architectures. Several relevant properties (operation count, the ratio of critical path to available time, spatial locality, and regularity) are identified and discussed, with quantitative measures being proposed for the latter two. Significant emphasis is placed on exploiting the regularity and spatial locality algorithm properties for the optimization of interconnect power. Examples illustrate the large savings that can be attained through property-based guidance of algorithm selection and architecture composition. Though demonstrated for ASIC designs, this approach is extensible to different hardware platforms and performance metrics (e.g. speed, area).

1. INTRODUCTION

Recently, power consumption has become a key performance metric in ASIC design, especially for the growing area of portable applications. However, few CAD tools exist to aid in the behavioral level analysis and optimization of low power designs. Analysis techniques such as PFA [1] and SPA [2] cannot be used before the architectural level of the design is reached. Also, previous work on behavioral power analysis and optimization is tightly coupled to a specific implementation platform [3, 4].

This work proposes an approach for high level low power design guidance that is extensible to different hardware platforms and performance metrics. The underlying premise is that an efficient design involves a natural match between a particular algorithm and architecture, which can be traced to explicit features in the algorithms and architectures themselves. In other words, high quality designs are those that exploit algorithmic and architectural properties to the greatest extent. This general approach and detailed descriptions of several property measures in the context of ASIC area estimation were presented in [5]. Properties have been investigated in related work for performance analysis [6]. In particular, this effort considered properties relevant for analyzing high speed parallel processing designs. The property measures of interest in this paper are of a different nature due to both the type of designs targeted (ASIC), and the performance goal (low power).

Total chip power is analyzed by considering each architectural resource individually (i.e. execution units, memory, control, clock, interconnect). The switching power for any single resource is given by $P = C_{eff} \cdot V_{dd}^2 \cdot f$, where C_{eff} is the effective capacitance switched, V_{dd} is the supply voltage, and f is the sampling frequency. C_{eff} has two components, the average capacitance of the resource and the activity. For a behavioral level analysis, the activity corresponds to the number of accesses to the resource.

2. PROPERTIES FOR LOW POWER DESIGN

The first step in the proposed approach involves determining which properties are important indicators for low power design. Properties are measured from the control/data flowgraph representation of the algorithm.

Analysis and optimization of the execution unit fraction of the power budget has been investigated previously [3]. The most relevant property is the average number of operations, which includes accesses to memory. Similarly, the number of edges is relevant for the prediction of register and bus accesses. Another measure is the ratio of critical path to the sample period, which indicates the potential for power reduction by using parallelism for voltage scaling [3]. In section 3, several examples illustrating the use of these properties for algorithm selection will be shown.

This project is sponsored by ARPA grant J-FBI 93-153. L. Guerra is supported by a fellowship from AT&T and the Office of Naval Research.

0-7803-2431-5/95 $4.00

Another important power component that has not previously received much attention, primarily due to the difficulty in analysis and optimization, is the interconnect power. This component often constitutes a significant amount of the total chip power and can increase dramatically for multiple-chip designs due to large off-chip bus capacitance. Two non-trivial properties, spatial locality and regularity, are key indicators of the potential for interconnect power optimization. Below, an overview of these properties is presented. Section 4 illustrates their use in synthesis guidance.

2.1 Spatial locality

Spatial locality relates to the degree to which an algorithm has natural isolated clusters of operations with few interconnections between them. As an example, a cascaded biquad IIR (see Figure 3) has isolated clusters of computation (each biquad), with only a single connection between each one. In the decimation-in-time FFT algorithm, on the other hand, no clear clusters exist. Identification and use of spatially local sub-structures can be used to guide hardware partitioning (inter and intra-chip) resulting in the minimization of global buses. This leads to lower bus capacitance, and thus lower interconnect power. Analogously, locality of memory references can be used to partition memories for low power implementations.

The method used to identify clusters is based on the eigenvalues and eigenvectors of the Laplacian of the graph. This technique has been used successfully in partitioning layouts [7]. The eigenvector corresponding to the second smallest eigenvalue provides a 1-D placement of the nodes (normalized to lie between -1 and 1), which minimizes the mean-squared connection length. Distances between placed nodes that are greater than a threshold (e.g. $\mu + 2\sigma$, where μ and σ are the mean and standard deviation of the distance statistics) are used to indicate partition points. This is illustrated in the following figure, which shows the 1-D node placement for a 4th order parallel IIR consisting of 2 biquad clusters (the filter structure is the same as shown in Figure 2,

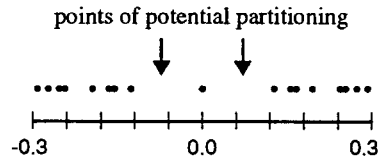

Figure 1: Node placement for the 4th order parallel IIR.

with one less biquad). The process is repeated hierarchically on clusters thus identified, and the cumulative number of points passing the threshold is an indicator of the total number of clusters.

2.2 Regularity

The idea behind regularity is to capture the degree to which common patterns appear in an algorithm. Common patterns enable the design of less complex architecture and therefore simpler interconnect structure (muxes, buffers, and buses). Additionally, regular designs often have less control hardware.

Simple measures of regularity include the number of loops in the algorithm and the ratio of operations to nodes in the data flowgraph. The statistics of the percentage of operations covered by sets of patterns is also indicative of an algorithm's regularity. Quantifying this measure involves first finding a promising set of patterns, large patterns being favored. The core idea is to grow pairs of as large as possible isomorphic regions from corresponding pairs of seed nodes.

Another measure of regularity is given by the following equation:

$$\text{Regularity} = \frac{\text{Size}}{\text{Descriptive Complexity}}$$

The *size* component is the number of operations and data transfers executed in the computation. The *descriptive complexity* is a measure of the shortest description from which the graph can be reproduced. The higher the regularity of a graph, the more repeated patterns of nodes and interconnections it has, and the more concisely it can thus be described. Motivated by the underlying ideas in descriptive complexity of strings (Kolmogorov complexity) [8], a heuristic measure of graph complexity has been developed. Given an encoding scheme (pseudo-language by which to describe the graph and a measure of the length of a program), the complexity of a graph is defined as the length of the shortest program which can represent the graph [5].

3. USING PROPERTIES IN ALGORITHM SELECTION

An important application of property-based prediction is to provide guidance for algorithm selection. In this section, we show how simple properties can go a long way in aiding this process.

3.1 Number of operations

As mentioned previously, the operation count property is directly related to the amount of execution unit power in an

implementation and can thus be used as a first order estimate of the relative power of the two algorithms. This is of course most useful when execution unit power is the dominant factor. Consider two alternate algorithms for a vector comparison (often used in vector quantization). The algorithms are used to compare the mean squared distance of a vector X from vectors C_a and C_b (constants):

$$MSE_a - MSE_b = \sum_{i=0}^{7} (C_{ai} - X_i)^2 - \sum_{i=0}^{7} (C_{bi} - X_i)^2$$

$$MSE_a - MSE_b = \left(C_a^2 - C_b^2\right) - \sum_{i=0}^{7} 2(C_{ai} - C_{bi}) \times X_i$$

Table 1 shows a summary of the operation counts and associated capacitance switched (voltage and sample period are fixed) for the two algorithms. An increased operation count not only effects EXU power, but also implies increased register and bus accesses due to the greater number of data transfers.

	Number of operations	EXU effective capacitance (pF)	Register effective capacitance (pF)
Algorithm 1	47	321	155
Algorithm 2	16	144	50
Percentage difference	66%	55%	68%

Table 1. Operation count and effective capacitances for vector comparison algorithms.

3.2 Critical path to sample period ratio

Similar to operation count, critical path can be used as a high-level guide of an algorithm's relative potential for low power implementation. Lower critical paths allow for more opportunity to lower the voltage while still meeting throughput requirements. Consider the 14-tap direct-form and transposed-form FIR filters. Table 2 shows the operating voltages, estimated areas, and estimated energies for each (sample rate is fixed).

Voltage	Area mm^2 (direct)	Area mm^2 (transposed)	Energy pJ (direct)	Energy pJ (transposed)
3	16.77	14.88	8.02	7.21
2.25	--	23.53	--	4.21
1.5	--	85.1	--	2.66

Table 2. Energy/Area of FIR direct and transposed form structures.

Both algorithms have the same number of operations; however, their critical path to sample period ratios are very different. At 5 volts, the direct form has a ratio of 1, while the transposed form has a ratio of 0.25. As a result, the latter can operate at much lower voltages (1.5V vs. 3.0V). Further, at a common voltage of 3V, the transposed version allows more complete utilization of resources, resulting in smaller total area and hence lower interconnect power. Both these factors together result in a 3x improvement in power, provided the area is not a factor. Alternatively, for a 1.4x area penalty, the transposed version gives a 1.9x improvement in power over the direct form.

4. USING PROPERTIES IN ARCHITECTURE COMPOSITION

Besides algorithm selection, there are a number of other applications of property-based prediction. This section highlights the potential for effectively guiding a number of tasks in the architectural composition process.

4.1 Spatial locality

As an example of how spatial locality can be exploited, consider the flowgraph of Figure 2. Using the technique described in Section 2.1, the algorithm is identified as having three strongly connected clusters of computation. This coincides with the expected clustering evident from the figure. In a direct implementation, hardware sharing between all operations might occur, destroying any locality of computation. To preserve this, the graph can be partitioned and implemented with the constraint that no

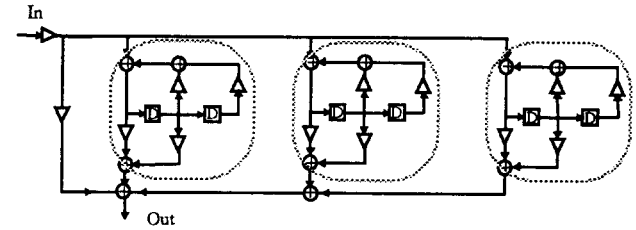

Figure 2: Spatial locality for partitioning guidance.

hardware sharing is done between operations in different clusters. Although the overall number of required execution units is likely to increase, the global interconnect is

	Direct implementation (pF)	Spatial locality driven implementation (pF)
Muxes	25	16
Buffer	13	9
Buses	491	160
Total	529	185
Improvement	1x	2.9x

Table 3. Effective capacitance with and without partitioning.

minimized, reducing average bus lengths and hence lowering bus capacitance.This leads to lower interconnect power and the minimization of the number of mux and buffer operations (which accompany high levels of bus multiplexing). For this example, a 2.9x improvement in interconnect power is achieved as shown in Table 3, which improves the total chip power by 1.5x.

Another advantage in this partitioning is that signal values within the clusters tend to be more highly correlated. This results in lower activities (e.g. switching capacitance of the execution units was reduced from 446 to 350 pF for this example).

4.2 Regularity

Similarly, preserving the algorithmic regularity in the design process can result in large power savings. Consider the flowgraph of Figure 3. The complexity of describing the graph using the encoding scheme referenced in Section 2.2 is 26, corresponding to a regularity of 4. However, if the presence of patterns is ignored, the complexity increases to 69, giving a regularity of 1.5. A direct assignment of operations to hardware units, without regard to the regularity of the algorithm, will result in an inefficient interconnect structure. Instead, if the corresponding operations in each of the four patterns are assigned to the same hardware, the

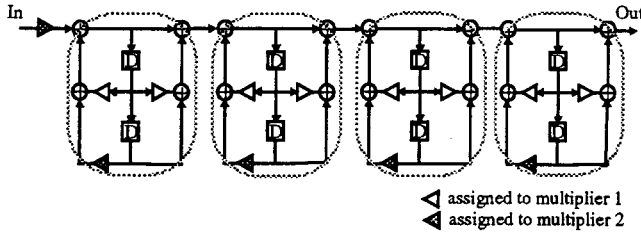

Figure 3: Regularity for assignment guidance.

interconnect structure is used more efficiently. In particular, the reduction in mux and buffer accesses is accompanied by a reduction in bus lengths, since fewer buses allows easier and more efficient routing. In this example, a 1.4x improvement in interconnect power is attained as shown in Table 4.

	Direct Implementation (pF)	Regularity-based implementation (pF)	Regularity-based implementation with chaining (pF)
Muxes	72	9	9
Buffer	29	31	22
Buses	349	284	232
Total	450	324	263
Improvement	1x	1.4x	1.7x

Table 4. Effective capacitance with and without regularity exploitation and chaining.

In addition to assignment, chaining can be used to further exploit regularity. In this example, the hardware corresponding to the regular add-add patterns are chained. The localization of interconnect between adders, and the corresponding reduction in the number of buffers results in further improving the interconnect power by 1.2x. Another advantage of chaining is that the removal of registers between the units results in the decrease of register access count, and thus corresponding register power. For the regularly assigned and chained versions, total chip power was improved by 1.1x and 1.2x, respectively.

5. CONCLUSION

This paper has presented the use of properties to aid in different aspects of the design process, with emphasis on power optimization as a performance goal. Relevant properties (operation count, critical path to sample period ratio, spatial locality, and regularity) have been identified and defined. Examples have illustrated the potential of these properties for guidance in the low power dimension.

6. REFERENCES

[1] S. Powell, P. Chau, "Estimating Power Dissipation of VLSI Signal Processing Chips: The PFA Technique," *VLSI Signal Processing IV*, IEEE Press, pp. 250 - 259, 1990.

[2] P. Landman, J. Rabaey, "Power Estimation for High Level Synthesis", *Proc. of EDAC-EUROASIC*, pp. 361-366, 1993.

[3] A. Chandrakasan, M. Potkonjak, R. Mehra, J. Rabaey, R. Brodersen, "Optimizing Power Using Transformations," to appear in *Transactions on CAD, Vol. 14, No. 1*, Jan. 1995.

[4] R. Mehra, J. Rabaey, "Behavioral Level Power Estimation and Exploration," *Proc. of the Int'l. Workshop on Low Power Design*, pp. 197-202, 1994.

[5] L. Guerra, M. Potkonjak, J. Rabaey, "System-Level Design Guidance Using Algorithm Properties," *VLSI Signal Processing VII*, IEEE Press, pp. 73-82, 1994.

[6] L. Jamieson, "Characterizing Parallel Algorithms," in *The Characteristics of parallel algorithms*, L. Jamieson, D. Gannon, R. Douglass (eds.), MIT Press, Cambridge, Mass., 1987.

[7] L. Hagen, A. Kahng, "New Spectral Methods for Ratio Cut Partitioning and Clustering," *Transactions on CAD*, Vol. 11, No. 9, pp. 1074-1085, Sept. 1992.

[8] M. Li, P. Vitanyi, "Kolmogorov Complexity and its Applications," in *Handbook of Theoretical Computer Science*, Jan van Leeuwen (ed.), pp. 187-254, MIT Press, Cambridge, MA, 1990.