Reprint from: Custom Integrated Circuit Conference, 1996.

Exploiting Locality for Low-Power Design

Renu Mehra, Lisa Guerra, and Jan Rabaey

Department of Electrical Engineering and Computer Sciences
University of California at Berkeley
Berkeley, CA 94720

Abstract

We propose a new high-level synthesis technique for the low-power implementation of real-time applications. The technique uses algorithm partitioning to preserve locality in the assignment of operations to hardware units. This results in reduced usage of long high-capacitance buses, fewer accesses to multiplexors and buffers, and more compact layouts. Experimental results show average reductions in bus and multiplexor power of 62.9% and 38.5%, respectively, resulting in an average reduction of 18.5% in total power.

1. Introduction

High-level synthesis is steadily making an inroad into the digital design community. Most work to date has focused on techniques for area and speed optimization [1]. Recently, there has been significant interest in techniques and tools for power optimization. While for area optimization, high resource utilization through hardware sharing is one of the main goals, for power optimization, reduced hardware sharing often gives better results.

Consider Wu's comparison of an automatically-generated maximally time-shared and a manually-generated fully-parallel implementation of a QMF sub-band coder filter [2]. In the manual design, a number of optimizations were used to obtain power savings in the various components. The power consumption of both versions is documented in Table 1. For the same supply voltage, an improvement of a factor of 10.5 was obtained at the expense of a 20% increase in area.

Note that the interconnect elements (buses, multiplexors, and buffers) consume 43% and 28% of the total power in the time-shared and parallel versions, respectively. Further, these elements contribute the most to the power reduction achieved in the parallel version. Power improvement factors of 16.9, 15.1, and 12.5 were obtained for buses, multiplexors, and buffers, respectively, mainly due to dedicated communication and reduced usage of multiplexors and buffers. This points to the large opportunity available for interconnect power reduction and highlights its significance.

While in this example, the fully-parallel implementation resulted in large power gains with low area overhead, this may not always be the case. Parallel implementations may be too area intensive and may not necessarily result in reduced interconnect power. If the area overhead is too high, the increase in the required bus lengths may offset the power gains due to other factors.

Table 1. Power consumption (mW) in the maximally time-shared and fully-parallel versions of the QMF sub-band coder filter.

Component	Time-shared	Fully-parallel	Improvement factor
Functional units	8.52	1.03	8.3
Registers	9.76	1.08	9.0
Buses	23.69	1.40	16.9
Multiplexors	3.77	0.25	15.1
Buffers	4.36	0.35	12.5
Others	23.99	2.92	8.2
Total	74.09	7.03	10.5

In this work, techniques are presented to achieve low-power designs by reducing the interconnect power while incurring low area overhead. The approach aims to capture some of the optimizations of the above example in an automated way while maintaining a balance between the maximally time-shared and the fully-parallel implementations. The next section illustrates the main idea behind our proposed low-power synthesis technique.

2. The impact of exploiting locality

The main idea behind our approach is to synthesize designs with localized communications. We achieve this by dividing the algorithm into *spatially local clusters* and performing a *spatially local assignment.* A spatially local cluster is a group of algorithm operations that are close to each other in the flowgraph representation. A spatially local assignment is a mapping of the algorithm operations to specific hardware units such that no operations in different clusters share the same hardware. Partitioning the algorithm into spatially local clusters ensures that the majority of the data transfers take place within clusters and relatively few occur between clusters. The spatially local assignment restricts intra-cluster data transfers to buses that are local to a subset of the hardware (local buses); thus only inter-cluster data transfers use buses that are shared by all resources (global buses). The combined result is that local buses which are

shorter are used more frequently than longer highly-capacitive global buses.

Consider the two different assignments for maximum throughput implementations of the fourth-order parallel-form IIR filter shown in Fig. 1. Indicated beside each operation is the hardware resource that it is assigned to. (A_i are adders and M_i are multipliers). In Fig. 1a, the graph is divided into two spatially local clusters and the operations in each cluster are mapped to mutually exclusive sets of hardware resources (A_1, A_2, and M_1 are used for operations in cluster 1 and A_3, A_4, and M_2 are used for those in cluster 2). As a result, a large number of the communications are restricted to only a subset of the hardware. In Fig. 1b, however, the hardware is not partitioned and all communications are global. The number of global data transfers (shown with solid lines in both cases) for the local and the non-local assignments are 2 and 20, respectively.

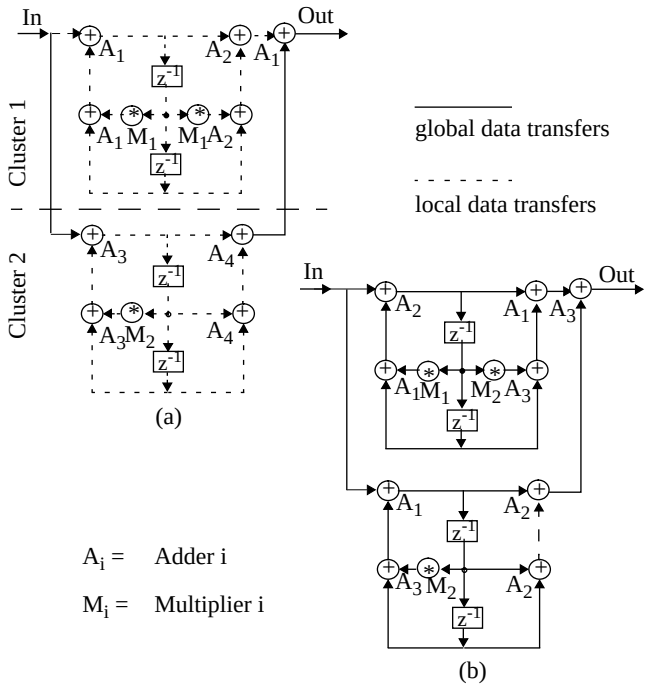

Fig. 1. A fourth-order parallel-form IIR filter: (a) Local assignment, (b) Non-local assignment.

Notice that the local version needs 4 adders and 2 multipliers whereas the non-local assignment requires just 3 adders and 2 multipliers. This increase in the number of functional units does not necessarily translate into a corresponding increase in the overall area since localization of interconnect makes the design more conducive to compact layout. Furthermore, reduced hardware sharing results in additional power savings due to fewer accesses to multiplexors and buffers.

Varying the number of clusters trades off local and global bus power. This is because, as the number of clusters is increased, the number of inter-cluster communications increases while the local bus lengths decrease.

3. Low-power synthesis system

The high-level synthesis process generates an architectural level netlist from a behavioral description and a set of performance constraints. In this section we present a new high-level synthesis strategy based on exploiting the locality of algorithm operations. The core of the approach is a partitioning and assignment scheme. The techniques have been integrated into the *Hyper-LP* system [3].

While the basic synthesis flow of the *Hyper-LP* system is the same as that of the *Hyper* system [4], a new partitioning step is added preceding the assignment phase and the assignment algorithm itself is modified to exploit spatial locality.

3.1 Partitioning methodology

Previous works in partitioning for high-level synthesis have targeted area minimization, with a significant portion of the gains resulting from interconnect reduction [5, 6]. For power minimization, however, it is better to have two global buses each accessed twice rather than one bus accessed six times. The goal for reducing interconnect power, therefore, is to minimize the *number of accesses* to long global buses.

Our partitioning methodology consists of two phases — the first phase generates several candidate partitioning solutions and the second phase evaluates them and selects the best one.

The generation of candidate partitions is based on a spectral partitioning technique used in a number of partitioning algorithms [7, 8, 9]. The technique was introduced by Hall [7] who proved that the second smallest eigenvector of the Laplacian of a graph gives a one-dimensional placement of graph nodes such that the sum of squares of edge lengths is minimized. Large gaps in this ordering are used to delimit the clusters. For example, Fig. 2 shows an eighth order cascade filter and the corresponding eigenvector placement. The spacing between nodes in the placement clearly indicates four distinct clusters which are also evident from the structure. We use $m+3\sigma$ as the threshold for detecting these gaps, where m is the mean of all the distances between the nodes and σ is the standard deviation of the distances. In the cascade example of Fig. 2, this threshold delimits the expected four clusters.

In our scheme, several different candidate partitions are generated by varying the targeted number of clusters. For example, in the cascade filter of Fig. 2, in addition to the 4-cluster partition, a 2-cluster partition may also be proposed. As discussed before, varying the number of clusters trades off between global and local bus power.

In the second phase a rough estimate of the total bus power is used to evaluate and compare the candidate partitions. The metric used as a measure of the global bus power is the number of global data transfers times the estimated global bus

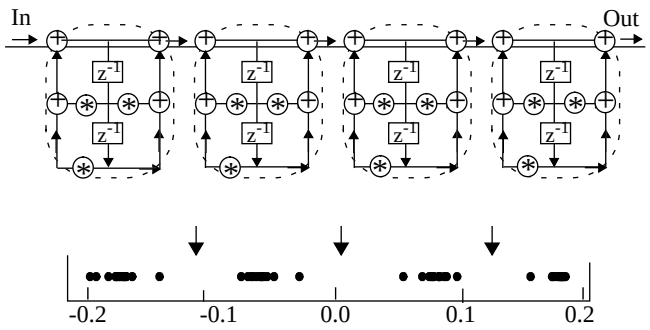

Fig. 2. An eighth-order cascade-form IIR filter and the corresponding eigenvector placement.

length. Similarly, the measure used for the local bus power of each cluster is the number of data transfers local to it times the cluster's estimated bus length.

Since the lengths of the buses have been shown to be proportional to the square root of the area, the cluster and total chip area estimates are used as measures of the local and global bus lengths, respectively. Estimates of these areas are in turn given by the maximum value of the weighted concurrency distribution graph [10]. The distribution graph gives the amount of concurrent hardware needed by the computation in each time slot.

At the culmination of the partitioning phase, the single most promising candidate partition is applied to the algorithm.

3.2 Assignment methodology

Once the partitioning is complete, all operations have an associated cluster number. Also, each data transfer is classified as either an inter- or intra-cluster transfer. The functional unit assignment performs the random assignment with iterative improvement approach of the *Hyper* system [4] and uses the clustering information to ensure that each cluster is assigned to mutually exclusive hardware.

Graph coloring is a commonly used technique to assign data transfers to shared buses such that there are no timing conflicts. In our scheme, we assign data transfers to buses avoiding not only timing but also clustering conflicts. Local buses are shared only among transfers within a cluster. Global buses are used only by inter-cluster data transfers. This ensures that intra-cluster data transfers occur on short local buses and only inter-cluster ones use the long highly-capacitive global buses.

4. Results

In this section we present the results of our partitioning-based synthesis scheme. Implementations generated using the *Hyper-LP* and the *Hyper* systems are compared. The SPA architectural power estimation tool [11] is used for power estimations. Estimates of the total chip area and bus lengths are obtained using models presented in [12]. The bus length model was enhanced to estimate the local and global bus lengths separately.

4.1 Cascade filter

The first result compares the *Hyper-LP* and *Hyper* implementations of the eighth-order cascade filter (Fig. 2). Given a throughput constraint of 21 clock cycles, the *Hyper* implementation uses 4 adders and 3 shifters while the *Hyper-LP* implementation uses one adder and one shifter for each cluster resulting in a total of eight units. Layouts of the two implementations are shown in Fig. 3. In the *Hyper* implementation, 2 of the 7 functional units are merged by the layout tool. In the *Hyper-LP* implementation the 4 datapaths pictured correspond to each of the 4 clusters.

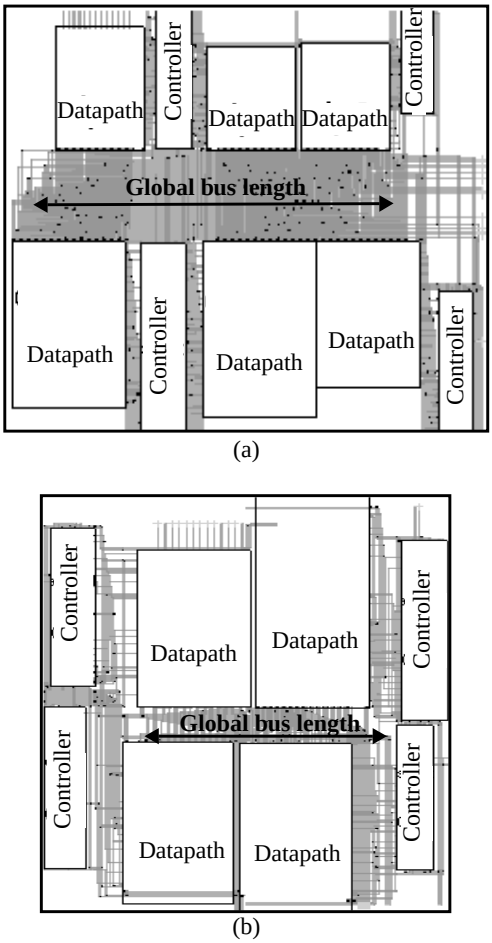

Fig. 3. Cascade filter layouts: (a) Non-local implementation from *Hyper,* (b) Local implementation from *Hyper-LP.*

Table 2 compares the power dissipated in the two implementations. An overall reduction of 34% in the power consumption was realized by the *Hyper-LP* approach. The average length of the global buses reduced by approximately 55%, from 2100 to 950 microns and the bus power reduced 5-fold, from 2 mW to only 0.4 mW. The multiplexor power

reduced by 60% as the reduced time-sharing of units results in lower usage of multiplexors. Notice that the contribution of interconnect (buses, multiplexors, and buffers) to the total power dissipation was reduced from 30% to 17%.

Table 2. Comparison of power consumption (mW) in the *Hyper* and *Hyper-LP* implementations of the cascade filter.

Component	*Hyper*	*Hyper-LP*	Percentage reduction
Buses	2.0	0.4	80.0
Multiplexors	3.7	1.5	59.5
Buffers	1.0	0.9	10.0
Others	8.7	7.4	14.9
Total	15.4	10.2	33.77

4.2 Other examples

This section summarizes our experimental results for the cascade and several other DSP filter and transform examples. Some are in their original form (DCT, FFT, and parallel-form IIR) and others are transformed using either constant multiplication expansion (cascade-form IIR, direct-form IIR, and wavelet) or retiming (wave digital filter).

Table 3. Comparison of power consumption (mW) in the *Hyper* and *Hyper-LP* implementations.

Design	*Hyper*			*Hyper-LP*			
	Bus	Mux	Total	Clusters	Bus	Mux	Total
Cascade	2.0	3.2	21.3	4	0.4	1.5	16.3
Direct form	29.8	38	144.6	3	10.3	21.1	110.4
Wave digital	1.5	3.2	20.4	2	0.5	1.7	18.0
DCT	9.0	3.8	41.5	2	4.5	2.3	37.2
FFT	17.6	4.7	48.6	2	5.2	3.8	36.8
Parallel IIR	15.1	2.8	57.5	2	3.2	2.2	48.8

Table 3 shows the bus, multiplexor, and the overall power dissipation for both implementations of each example. Table 4 summarizes the percentage power improvements. The *Hyper-LP* implementations uniformly dissipate less power than the *Hyper* implementations. Power consumed by buses is reduced drastically in all examples (up to 80%). Up to 60% reduction in multiplexor power is seen due to reduced and more localized hardware sharing. The average reduction in bus, multiplexor, and total power is 62.9%, 38.5%, and 18.5%, respectively.

The power reduction comes at the cost of an increase in the number of units. However, since the communications are localized, the designs are more conducive to compact layout. Further, overhead elements such as multiplexors and buffers are reduced. Table 4 shows the estimated area penalty obtained in the *Hyper-LP* designs.

Table 4. Overall power reduction and area overhead.

Design	Percentage power reduction			Percentage change in area
	Bus	Mux	Total	
Cascade	80.0	59.6	25.9	-15.1
Direct form	65.4	44.5	23.7	+2.1
Wave digital	33.3	46.9	11.8	+7.2
DCT	50.0	39.5	10.4	-23.4
FFT	69.7	19.1	24.3	+0.8
Parallel IIR	78.7	21.4	15.1	+22.1
Average	62.9	38.5	18.5	-1.05

5. Conclusions

We have presented a technique for power reduction based on exploiting the locality in a given application. At the core of the approach is a partitioning and assignment strategy. It was seen that the proposed scheme improves the implementation in a variety of ways. The predominant effect is the reduction of accesses to highly-capacitive global buses. Our results showed average bus, multiplexor, and overall power reductions of 62.9%, 38.5%, and 18.5%, respectively, and low associated area overheads. The partitioning and assignment techniques have been integrated into the *Hyper-LP* system.

6. References

1. D.D. Gajski, *High-Level Synthesis: Introduction to Chip and System Design*, Boston, Kluwer Academic, 1992.
2. S. Wu, "A Hardware Library Representation for the Hyper Synthesis System," *Masters' Thesis*, University of California, Berkeley, Memorandum No. UCB/ERL M94/47, June 1994.
3. A. Chandrakasan, M. Potkonjak, R. Mehra, J. Rabaey, and R. W. Brodersen, "Optimizing Power Using Transformations," *IEEE Trans. on CAD*, Vol. 14, No. 1, Jan. 1995, pp. 12-31.
4. J. M. Rabaey, C. Chu, P. Hoang, and M. Potkonjak, "Fast Prototyping of Datapath-Intensive Architectures," *IEEE Design & Test of Computers*, June 1991, pp. 40-51.
5. M.C. McFarland and T.J. Kowalski, "Incorporating Bottom-up Design into Hardware Synthesis," *IEEE Trans. on CAD*, Vol. 9, No. 9, Sept. 1990, pp. 938-949.
6. E.D. Lagnese and D.E. Thomas, "Architectural Partitioning for System Level Synthesis of Integrated Circuits," *IEEE Trans. on CAD*, Vol. 10, No. 7, July 1991, pp. 847-860.
7. K. M. Hall, "An r-Dimensional Quadratic Placement Algorithm," *Management Science*, Vol. 17, No. 3, Nov. 1970, pp. 219-229.
8. L. Hagen and A. B. Kahng, "New Spectral Methods for Ratio Cut Partitioning and Clustering," *IEEE Trans. on CAD*, Vol. 11, No. 9, Sept. 1992, pp. 1074-1085.
9. B. Hendrickson and R. Leland, "The Chaco User's Guide, V. 1.0," *Tech Report SAND93-2339, Sandia National Lab*, Oct. 1993.
10. P. G. Paulin and J.P. Knight, "Force-Directed Scheduling for Behavioral Synthesis of ASIC's," *IEEE Trans. on CAD*, Vol. 8, No. 6, June 1989, pp. 661-679.
11. P. E. Landman and J. M. Rabaey, "Architectural Power Analysis: The Dual Bit Type Method," *IEEE Trans. on VLSI Systems*, Vol.3, No.2, June 1995, pp. 173-87.
12. R. Mehra and J. M. Rabaey, "Behavioral Level Power Estimation and Exploration," *Proc. of the Int'l. Workshop on Low-Power Design*, April 1994, pp. 197-202.