Power-Performance Optimal DSP Architectures and ASIC Implementation

Farhana Sheikh[1], Melinda Ler[1], Radu Zlatanovici[1], Dejan Marković[2], Borivoje Nikolić[1]

[1]Dept. of Electrical Engineering & Computer Sciences, University of California, Berkeley, CA 94720
[2]Department of Electrical Engineering, University of California, Los Angeles, CA 90095
{farhana, melindal, zradu, bora}@eecs.berkeley.edu, dejan@ee.ucla.edu

Abstract—**A hierarchical, sensitivity-based ASIC design methodology is proposed and demonstrated in the implementation of power-performance optimal signal processing kernels for wireless applications. The design approach uses a systematic exploration of the power-performance design tradeoff space at the architecture, micro-architecture, and circuit levels. Energy-efficiency gains achieved via this methodology are exploited to accommodate flexibility to support multi-standard radio architectures. The methodology is exemplified in the selection of architecture and design of a flexible digital finite impulse response (FIR) filter. The flexible FIR filter consumes area and power that is only 2 to 4 times that of a dedicated ASIC FIR.**

I. INTRODUCTION

The accelerated deployment of multi-mode, multi-standard wireless systems is resulting in an exponential increase in algorithmic complexity that is outpacing the scaling benefits of Moore's Law [1]. The recent rapid advances in wireless communication demand extremely high levels of functionality and flexibility which cannot be simply obtained via technology scaling at little or no area or energy cost. It is necessary to design energy-efficient algorithms and architectures that consume the least power at the required performance. For example, a straight-forward implementation of multi-mode operation requires several parallel radios; this is an inefficient system in terms of energy and area cost. Ideally, the most efficient design is one where a single transceiver chain is shared among multiple modes and multiple standards. Wireless transceivers have tight power and area constraints, and numerous architectures can be conceived to achieve the required throughputs. The design tradeoff space is very large; thus significant effort is required to select the optimal architecture which will result in the lowest power consumption for the required performance. In this paper, we address this challenge by applying a hierarchical, sensitivity-based ASIC design methodology to create power-performance optimal signal processing architectures for wireless systems.

The proposed design approach systematically explores the power-performance-flexibility design tradeoff space, in a synthesis-based design environment, using sensitivity-based optimization at the architecture, micro-architecture, and circuit levels [2]. Energy-efficiency gains achieved via this hierarchical design methodology are exploited to accommodate flexibility.

The methodology is exemplified in the architecture selection and design of a finite impulse response (FIR) filter used in a multi-standard radio receiver front-end. The selected architecture and design is optimal in the power-performance design tradeoff space. The proposed design methodology allows for comprehensive exploration of a wide set of filter architectures to achieve flexibility at little cost in terms of area and power. The systematic use of sensitivity analysis in the architecture exploration phase, in the context of circuit and logic level constraints, enables short design times. The result is an optimal choice of architecture for the given constraints.

II. POWER-PERFORMANCE OPTIMIZATION

The multi-level hierarchical design methodology employed is founded on sensitivity-based optimization where energy efficiency is the primary design objective [2][3][4]. Sensitivity or hardware intensity is the ratio of the relative increase in energy and the corresponding relative gain in performance achieved by tuning a design parameter such as gate size or supply voltage. For example, if the energy-efficient curve (with respect to gate sizing) for a circuit is plotted in the energy-delay coordinate space, then a specific value of the hardware intensity is the normalized derivative taken at a specific point on this curve. Analytically, the sensitivity to a design parameter x is given as:

$$S(x) = -\frac{D\partial E}{E\partial D}\Big|_x, \quad 0 \le S(x) \le \infty \qquad (1)$$

An energy-efficient design is achieved when the marginal costs of all tuning variables are equalized [2][3]. This serves as the optimality condition for circuits. For system optimality, the conditions are similar but the optimal aggregate sensitivity is defined to be equal to weighted sensitivities of composite blocks, where the weights are ratios of their individual contribution to total system energy and total system delay [5].

In this research, sensitivity as defined in (1) is employed as a design metric since it encompasses all other metrics; its final value is dependent on the performance requirements.

III. Optimization Framework and Design Methodology

A. Overview

Traditional synthesis-based design environments target cycle time as the primary design objective. However, in a power-limited scaling regime, energy-efficiency must be given equal priority in the optimization criteria. Each digital function has its representative optimal energy-delay tradeoff curve which characterizes the minimal achievable energy for performing the required function under delay constraints. The derivative of the energy-delay tradeoff curve for each tuning variable provides the sensitivity of energy and delay to that specific tuning variable. An envelope of the composite plots with respect to the various tuning variables at each design abstraction layer provides the energy-efficiency boundary. This boundary is obtained by balancing sensitivities of all possible tuning variables at all levels of design hierarchy across all circuit blocks.

For example, by plotting the energy-delay tradeoff curves at each level of design abstraction as illustrated in Fig. 1, it can be deduced that Architecture 3 achieves higher performance than Architecture 1 for the given E_{max} energy constraint. The architecture composite curve is obtained by obtaining energy-delay tradeoff curves for the circuit tuning variables such as gate size and the tradeoff curves for the micro-architecture level tuning variables such as pipeline depth. The Circuit 1 tradeoff curve represents the circuit implementation of one of the system blocks at a particular supply voltage. At the intersection of the Circuit 1 tradeoff curve and the Micro-architecture 1 tradeoff curve, the sensitivity to gate sizing is equal to the sensitivity to pipeline depth. This optimality condition can be captured in an equation that gives the optimal aggregate sensitivity as a function of sensitivities to tuning variables at lower levels of design abstraction as detailed in [5].

Our methodology is both top-down and bottom-up as the greatest energy-efficiency in design is achieved when design decisions at the top level of hierarchy (architecture) are considered in the context of constraints at lower levels of hierarchy. We achieve this through composition. Constraints from higher levels of hierarchy are propagated down to lower levels, and sensitivities of tuning variables to energy and delay are balanced upward through the various design abstraction layers.

In the filter optimization example, the main circuit tuning variable is gate sizing. Supply voltage was fixed at nominal value from the technology to maximize the performance. At the micro-architecture level parallelism, folding and pipelining are the primary tuning variables. At the architecture level, conventional transpose and transverse styles, and the multiplier-less distributed arithmetic architecture [6][7] can be devised. The entire power-performance-flexibility tradeoff analysis is carried out in an experimental design framework using commercially available synthesis, simulation and backend place and route tools. The architecture optimization was performed using a standard 90nm CMOS process technology. The standard cell library contained over 400 cells – a fairly large library to limit quantization effects due to gate sizing.
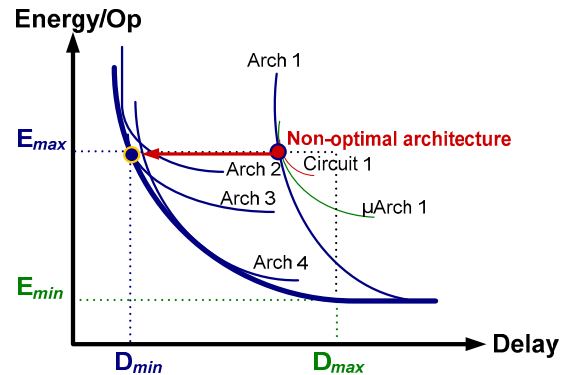

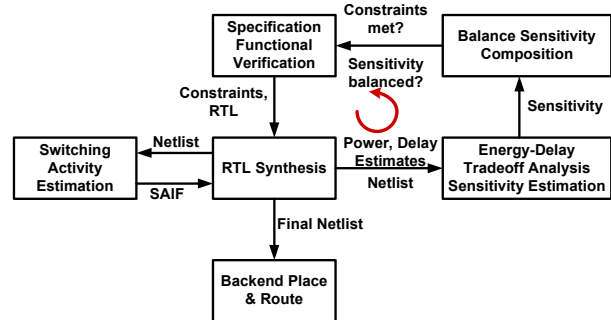Fig. 1. Energy-delay tradeoffs at multiple levels of design abstraction

Fig. 2. Hierarchical design methodology

Tcl scripts were created to synthesize various building blocks such as adders, multipliers and multiply-accumulate blocks for various delay and throughput targets. A combination of Simulink, and Module Compiler were used to model the various designs at the higher levels of abstraction. Lower level RTL netlists were generated for each delay target. Simulated switching activity was back-annotated into the gate-level design for power analysis and optimization. Composition rules derived from [5] for each of the different filter architectures allowed for quick power and throughput tradeoff analysis. A pictorial representation of the design methodology is given in Fig. 2.

B. Energy-Delay Tradeoffs and Sensitivity Estimation

Hierarchical energy-delay tradeoff analysis requires calculation or estimation of sensitivity at each level of hierarchy. This can be a compute-intensive and an overwhelming task if the design is large and there are numerous tuning variables. In our approach, we forgo calculation of derivatives and instead choose to estimate sensitivity using simple models derived from the use of a custom circuit optimizer [8] for critical small blocks such as adders and multipliers. Once enough energy-delay points are available via the optimizer, we can calculate the sensitivity to gate sizing and plot this against the ratio of total gate capacitance and wire capacitance (C_{gate}/C_{wire}).

In Fig. 3, the plot for a 64-bit Ling adder shows a linear relationship between sensitivity and C_{gate}/C_{wire} for all input loading conditions. When the same gate-level RTL netlist, is synthesized, we note that there is a discrepancy between the
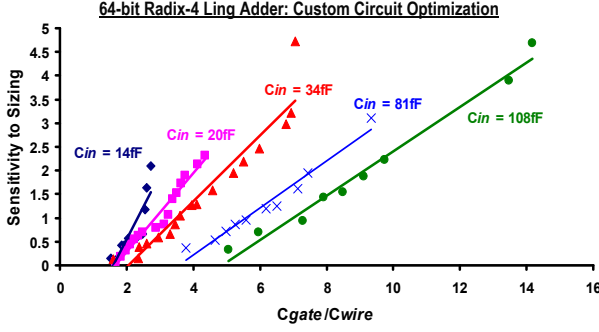
Fig. 3. Sensitivity to sizing estimation for 64-b radix-4 sparse-2 Ling adder
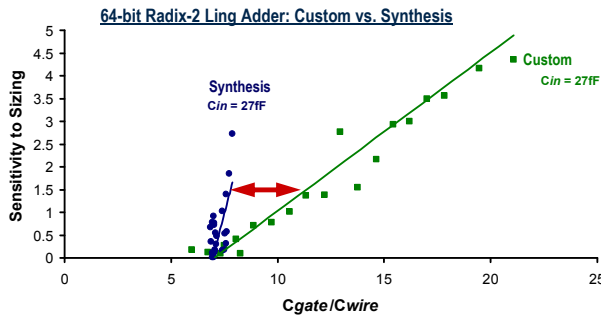

Fig. 4. Comparison of synthesized and custom optimized adder

custom design and the synthesized design (see Fig. 4). This custom-synthesis gap is due to the fact that synthesized designs require more routing area and gate sizes are quantized. This linear relationship extends to larger blocks for sensitivities around the knee of the energy-delay curve. For gate sizing, the model's slope and x-intercept can be derived in terms of C_{gate}, C_{wire}, C_{input} as shown in the equations given in (2). The variable n gives the number of different input capacitance loads. As n approaches infinity, the summation below becomes an integral.

$$x-intercept = \frac{C_{g-min}}{C_w} \; ;$$

$$Slope = \frac{C_{g-min}}{C_w} \cdot \frac{\sum_{i-1}^{n} C_{in_i}}{n} \cdot (C_{in})^{-(C_w/C_{g-min})} \qquad (2)$$

Sensitivity for higher level blocks are estimated using composition rules that give equations for optimal aggregate sensitivity in terms of sensitivity for lower level blocks as we will show in the next section. Hierarchical composition is a key factor in scaling this design methodology to very large designs.

C. Sensitivity Balancing Across Layers of Hierarchy

For system optimality, balancing block sensitivities means that the optimal aggregate sensitivity is either equal to a weighted sum of block sensitivities or is equal to "normalized" block sensitivity. The weight or normalization factor for a block is based on the ratio of its contribution to the total energy and to the total delay of a system. These balancing conditions lead to composition rules for optimal aggregate sensitivity as outlined in [5]. This optimal aggregate sensitivity provides an ideal

energy-efficiency boundary for the system. The boundary can be constructed for a synthesized or custom-optimized design depending on which models (synthesized or custom-optimized) are used at the building block level. An example of composition is shown in Fig. 5 for a 32-tap transpose filter; the ideal composition curves are compared with the synthesized results. Table 1 shows the composition rules derived for a number of different filter architectures explored. The number of taps is N. Fig. 7 shows the entire filter architecture tradeoff space that is generated using this methodology.

TABLE 1: COMPOSITION RULES FOR FILTERS

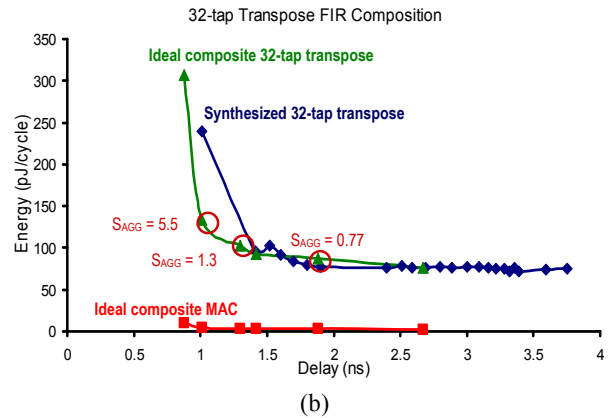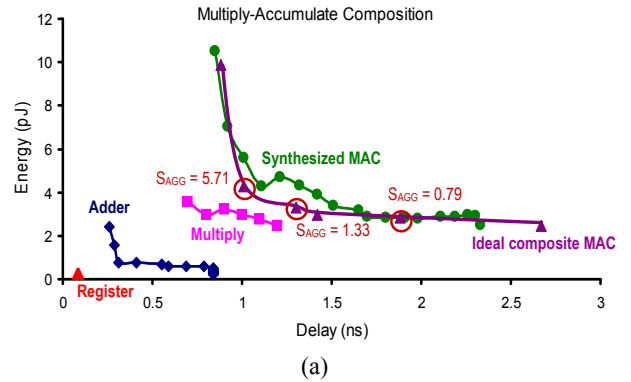| N-TAP FILTER	DELAY MODEL	ENERGY MODEL	OPTIMAL AGG. SENSITIVITY
Transpose	D_{MAC}	$(N-1)E_{MAC} + E_{Mult}$	$((N-1)/N)S_{MAC} + (1/N)S_{Mult}$
Transverse	$D_{Mult} + (N-1)D_{Add}$	$N(E_{Mult} + E_{Add}) + (N-1)E_{Reg}$	$\frac{N/(3N-1)}{1/(N)}S_{Mult} = \frac{N/(3N-1)}{(N-1)/(N)}S_{Add} = \frac{(N-1)/(3N-1)}{1}S_{Reg}$
P-Parallel Transpose	$D_{Transpose}/P$	$P \cdot E_{Transpose}$	$S_{Transpose}$
Pipelined Transverse	D_{MAC}	$(N-1)E_{MAC} + E_{Mult}$	$((N-1)/N)S_{MAC} + (1/N)S_{Mult}$

(a)

(b)

Fig. 5. Derivation of energy-efficiency boundary for 32-tap transpose filter: (a) Multipliy-accumulate composition; (b) Transpose FIR composition

Achieving the optimal aggregate sensitivity might not be possible in practice, due to a number of factors which include heuristic optimization in synthesis, quantization effects, poor estimation of wire capacitance, and inaccurate estimation of power/energy. A representation of this phenomenon is shown in Fig. 6. An actual example is shown in the difference between the ideal curve and the synthesized one in Fig. 5. Some of these issues can be addressed, but not completely eliminated: we can use simulation to capture switching activity for a design; an iteration of place and route can be performed to obtain a more accurate estimate of wire capacitance. However, the quantization effect can only be reduced by adding more cells. In addition, the approximated curve is based on models of sensitivity and approximation of the contribution of a block to the total energy and delay of the system, hence may not match the synthesized curve.

The ultimate goal is to minimize the difference between the synthesized energy-efficiency boundary and the ideal estimated energy-efficiency boundary for the system. This is shown in the optimization given in (3).

$$\min \left\| S_{C,X} - OptAggS(S_{A,X}, S_{B,X}) \right\| + \left\| S_{C,Y} - OptAggS(S_{A,Y}, S_{B,Y}) \right\|$$
$$\text{with respect to}: S_{C,X}, S_{C,Y}, X, Y \qquad (3)$$

The optimization in (3) is an example for a system C (see Fig. 6) which is comprised of two blocks A and B, and two tuning variables X and Y. The variables $S_{C,X}$ refer to the actual sensitivity to X of block C which is composed of block A and B. The constraints are minimum and maximum conditions on sensitivity, energy, and delay. The OptAggS(·) function refers to the calculated optimal aggregate sensitivity for the design C in terms of sensitivities of A and B to the respective tuning variables. The optimal aggregate sensitivity can be either the one derived for a custom circuit implementation using the linear model for sensitivity to sizing for a custom-designed critical block such as the 64-bit adder mentioned earlier; or it can be the one derived from the synthesized version of the model. When the difference between the previous and current iteration of the optimization is within a given threshold, the optimization is complete.

This is an elegant way to balance sensitivities across layers of hierarchy as the optimal aggregate sensitivity, which is our target, automatically provides us with a point on the ideal energy-efficiency boundary for the entire system. In addition, since the optimal aggregate sensitivity is computed in terms of sensitivities to tuning variables of lower level blocks, we automatically assess energy-delay tradeoffs at higher levels of abstraction in terms of lower level energy-efficiency constraints.

IV. PROGRAMMABLE FIR FILTER FOR MULTI-STANDARD WIRELESS COMMUNICATION

A. Flexible Digital Filter Tradeoffs

Future multi-mode, multi-standard wireless receivers will continue to move the boundary between analog and digital signal processing for increased flexibility [9].
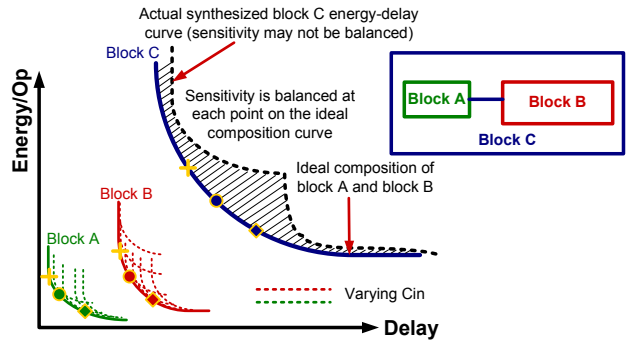

Fig. 6. Balancing sensitivity across hierarchy

In the digital front-end, the signal processing tasks include decimation, sampling rate conversion, and equalization. All of these must be supported by flexible filters that consume very little power but support low to high throughput rates, and varying number of bits in the word length of the coefficients and input stream. Essential to any of the flexible radio receivers (and transmitters) are FIR filters. Each standard dictates different requirements for the FIR filter. Some of these are shown in Table 2. An N-tap finite impulse response digital filter is described by the following equation (4), where y is the output sequence, x is the input sequence, and a_k is the unit-sample response.

$$y[n] = \sum_{k=0}^{N-1} a_k x[n-k] \qquad (4)$$

The goal of the design is to create a flexible digital FIR filter that supports 3G, wireless LAN, and digital television broadcast and consumes up to two to four times as much power as a filter dedicated to a single standard.

There are four separate design abstraction layers where power-performance-flexibility tradeoffs must be considered: circuit level, logic or arithmetic level, micro-architecture level, and the architecture level. Constraints from each layer must be propagated to the other layers to ensure an optimal tradeoff between power, performance, and area. The cost of flexibility is measured as the additional power and area required to support flexibility in terms of tap programmability; and in terms of programmability of input and coefficient word length. The authors in [10] present a detailed analysis of the various architecture, arithmetic level, and logic level choices available for a conventional filter design.

At the architecture level, a designer can choose to either time-multiplex or parallelize (e.g. multiplex in the space domain). At this abstraction layer, the designer also has the option of choosing between different filter structures: direct transversal filter, transposed direct form, multi-operand addition where the addition forms a tree, or using a distributed arithmetic structure that eliminates multiplication. At the logic and arithmetic level, decisions on number representation, sign processing, and adder and multiplier architectures can affect the number of partial products, the critical path delay, and power dissipation.

TABLE 2: FLEXIBLE FILTER REQUIREMENTS

STANDARD			FILTER REQUIREMENTS	
	Max. Throughput		No. of Taps	Word Length (bits)
3G				
WCDMA - UMTS	16 – 32 MSamples/s	1.92 Mbit/s (5 MHz)	8 – 64	6 – 8
WLAN				
802.11g	40 – 80 MSamples/s	54 Mbit/s	8 – 64	10 – 12
802.11n	40 – 160 MSamples/s	100 – 200 Mbit/s (40 MHz)	8 – 64	10 – 12
BROADCAST				
DVB-T/H	20 – 25 MSamples/s	4 – 30 Mbit/s (5 - 8 MHz)	32 – 64	10 – 12
ATSC Re-sample filter	15 – 25 MSamples/s	20 Mbit/s	32 – 64	10 – 12

At the circuit level, decisions on circuit implementation style, choice of supply and threshold voltages, clocking scheme, static or dynamic flip-flops that can be either edge-triggered or level-sensitive, and choice of gate sizes can impact the performance and power of a design.

B. Flexible Filter Architectures

Flexibility can be added to conventional architectures by combining parallelism and time multiplexing and by adding some control and memory. Two clocks are required: one for the filter and one for the control. The filter operates at the higher frequency so that time-multiplexing is possible.

Another option to consider for a flexible filter is the multiplier-less distributed arithmetic architecture which is described in detail in [6] and [7]. The distributed architecture structure is flexible in nature as a variable input word length can be accommodated by varying the accumulation cycles and the filter order can be varied by partitioning look up tables [11].

The power-performance-flexibility optimization is carried out by characterizing each different choice of filter architecture in the energy-delay tradeoff space. The cost of flexibility is measured by comparing fixed architecture area and power with that of flexible architectures.
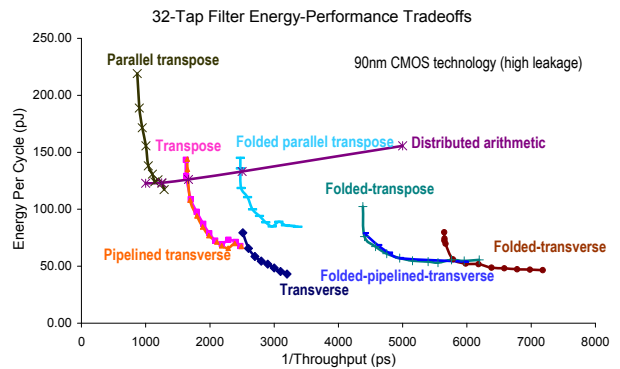
V. RESULTS

Results of the architecture tradeoff analysis are presented here for each of the various architectures considered. Fig. 7 shows the resulting architecture tradeoff space for two 90nm CMOS technologies with different transistor thresholds, supplies and foundries; one is optimized for high performance and the other for lower leakage. As one can see at lower throughputs, a flexible conventional architecture, such as the transpose or transverse, outperforms the distributed arithmetic FIR. However, at very high throughputs, the distributed arithmetic FIR is the most energy-efficient. Area can be re-

duced for the distributed arithmetic architecture by folding it in time. It is possible to do this and still meet throughput specifications because the filter can operate at such high throughput rates at low power.

The key results from the study show that flexibility requires a distributed arithmetic filter in a low leakage library or a hybrid parallel, time-multiplexed conventional filter in a high performance library. Fig. 7 shows that parallelism provides high throughput (see the parallel transpose curve) and that folding in time provides low energy (see the folded transverse plot). An 8 – 48 tap programmable FIR designed in the high performance technology is shown in Fig. 8. This filter supports both full-band and half-band mode operation. Folding in time was used to support tap programmability. It uses a parallel transverse architecture to meet the specified throughput constraints. The cost of flexibility of this filter over a fixed 27-tap dedicated half-band WLAN filter was determined to be 3 times in energy and 2 times in terms of area.

The distributed arithmetic filter is preferred for high throughput applications and flexibility in terms of tap programmability, variable input and coefficient word length.

(a)
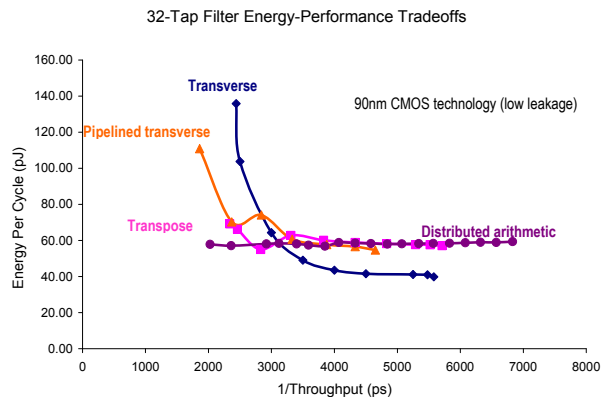
(b)

Fig. 7. Filter architecture tradeoff space: (a) High performance 90nm CMOS technology; (b) Low leakage 90nm CMOS technology.

For the given specifications of throughput and clock rate, results from the optimization using a low leakage technology showed that a time-multiplexed, parallel distributed arithmetic filter structure using an offset binary coding scheme for memory was the most energy-efficient flexible filter.

On average, adding tap programmability to a conventional filter resulted in an additional 25% overhead in energy-efficiency and 15% increase in area. However, this limited flexibility in a conventional architecture resulted in a reduction in throughput and increase in latency. The cost of flexibility, when using a distributed arithmetic architecture, was estimated to be 50% in terms of energy-efficiency and 60% to 80% in area, depending on the filter order. This cost is still much less than that required for a design that has one filter for each standard. The distributed arithmetic filter architecture is inherently flexible so there is no loss in throughput or increase in latency. In fact, it can be used to support very high throughput requirements efficiently.

VI. SUMMARY AND CONCLUSIONS

In this paper we have presented a hierarchical sensitivity-based design methodology that allows for a systematic exploration of the energy-delay tradeoff space at the architecture and micro-architecture level in the context of circuit level constraints, in an ASIC design environment. Within a week, it is possible to evaluate a wide range of flexible filter architectures that are suitable for a multi-standard radio. We have shown that sensitivit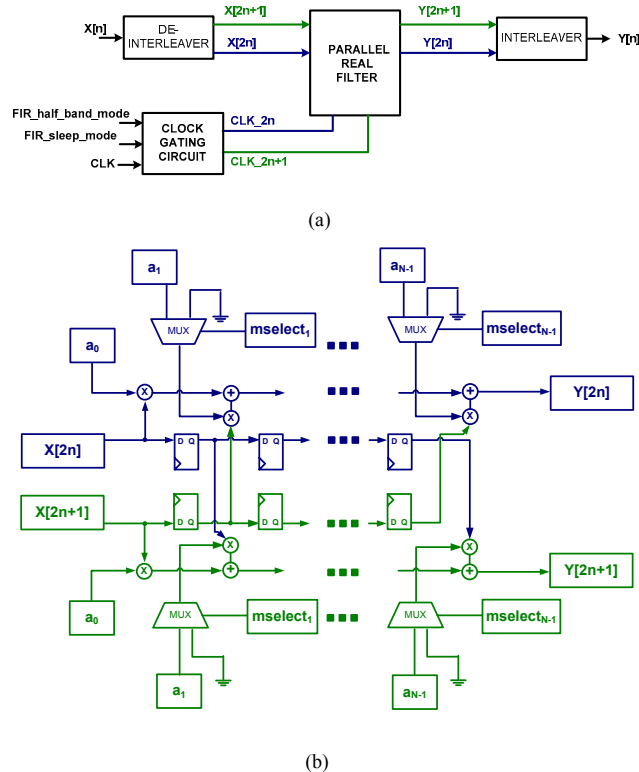y to sizing can be estimated using a linear approximation which is a function of input capacitance, total gate capacitance and total wire capacitance. This result holds in both custom and synthesis design environments. Balancing sensitivity across levels of hierarchy using optimal aggregate sensitivity composition rules generates an ideal energy-efficiency boundary which can be used to measure the optimality of the designed ASIC. The methodology is scalable to large designs due to its hierarchical nature. The methodology presented is exemplified in the architecture exploration and selection of energy-delay optimal flexible filters used in a multi-standard radio receiver. The resulting optimal architecture is dependent on the technology (high performance or low leakage) and the memory-to-logic ratio for a particular architecture. The cost of flexibility is determined to be a maximum of 2 to 4 times that of a filter dedicated to a single wireless standard.

REFERENCES

[1] G. E. Moore, "Cramming More Components onto Integrated Circuits", *Electronics Magazine,* Volume 38, Number 8, 19 April, 1965.

[2] D. Markovic, V. Stojanovic, B. Nikolic, M. A. Horowitz, R. W. Brodersen, "Methods for True Energy-Performance Optimization", *IEEE Journal of Solid-State Circuits*, Vol. 39, No. 8, August 2004, pp. 1282-1293.

[3] V. Zyuban and P. Strenski. "Unified Methodology for Resolving Power-Performance Tradeoffs at the Microarchitectural and Circuit Levels" in *Proceedings of ISLPED 2002.* August 12-14, 2002, Monterey, CA, USA.

[4] H. P. Hofstee. "Power-Constrained Microprocessor Design", in *Proceedings 2002 IEEE International Conference on Computer Design: VLSI in Computers and Processors.* September 2002, pp.14-16, Freiberg, Germany.

[5] V. Zyuban and P. Strenski, "Balancing hardware intensity in microprocessor pipelines", in IBM Journal of Research and Development, Vol. 47, No. 5/6, September/November 2003.

[6] K. Kim, K. Lee, "Low-Power and Area-Efficient FIR Filter Implementation Suitable for Multiple Taps", *IEEE Transactions on VLSI Systems*, Vol. 11, No. 1, February 2003, pp. 150-153.

[7] S. Rylov et. al., "A 2.3 GSample/s 10-tap Digital FIR Filter for Magnetic Recording Read Channels", *ISSCC Digest of Technical Papers*, 2001 IEEE International Solid-State Circuits Conference, 5-7 February 2001, pp. 190-191.

[8] R. Zlatanovici and B. Nikolić, "Power-Performance Optimization for Custom Digital Circuits", in Proceedings of PATMOS 2005, Leuven, Belgium, pp. 404 – 414, September 2005.

[9] L. Maurer, T. Burger, T. Dellsperger, R. Stuhlberger, M. Schmidt and R. Weigel, "On the Architectural Design of Frequency-Agile Multi-Standard Wireless Receivers", IST Mobile and Wireless Summit, Dresden, Germany, June 19 – 23, 2005.

[10] T. Gemmeke, M. Gansen, H. J. Stockmanns, T. G. Noll, "Design Optimization of Low-Power High-Performance DSP Building Blocks", *IEEE Journal of Solid-State Circuits*, Vol. 39, No. 7, July 2004, pp. 1131-1139.

[11] M. Ler, *An Energy Efficient Reconfigurable FIR Architecture for a Multi-Protocol Digital Front-End*, Master's Report, Dept. of EECS, University of California, Berkeley, March 2006.

(a)

(b)

Fig. 8. Parallel time-multiplexed 8 - 48 tap programmable FIR: (a) Time interleaving and clock gating; (b) Parallel transverse filter