Methods for True Energy-Performance Optimization

Dejan Marković, *Student Member, IEEE*, Vladimir Stojanović, *Student Member, IEEE*, Borivoje Nikolić, *Member, IEEE*, Mark A. Horowitz, *Fellow, IEEE*, and Robert W. Brodersen, *Fellow, IEEE*

***Abstract*—This paper presents methods for efficient energy-performance optimization at the circuit and micro-architectural levels. The optimal balance between energy and performance is achieved when the sensitivity of energy to a change in performance is equal for all the design variables. The sensitivity-based optimizations minimize energy subject to a delay constraint. Energy savings of about 65% can be achieved without delay penalty with equalization of sensitivities to sizing, supply, and threshold voltage in a 64-bit adder, compared to the reference design sized for minimum delay. Circuit optimization is effective only in the region of about $\pm 30\%$ around the reference delay; outside of this region the optimization becomes too costly either in terms of energy or delay. Using optimal energy–delay tradeoffs from the circuit level and introducing more degrees of freedom, the optimization is hierarchically extended to higher abstraction layers. We focus on the micro-architectural optimization and demonstrate that the scope of energy-efficient optimization can be extended by the choice of circuit topology or the level of parallelism. In a 64-bit ALU example, parallelism of five provides a three-fold performance increase, while requiring the same energy as the reference design. Parallel or time-multiplexed solutions significantly affect the area of their respective designs, so the overall design cost is minimized when optimal energy–area tradeoff is achieved.**

***Index Terms*—Adders, circuit optimization, circuit topology, digital circuits, energy–performance tradeoff, leakage currents, parallel architectures, pipelines.**

I. Introduction

THE nature of integrated circuit design has experienced a major change in recent years due to continued scaling of the underlying technology. In the past, the amount of functionality that could be integrated on chip was limited by area; today, power dissipation is the primary limiting factor. The characteristics of power constraints are different for desktop processors and mobile devices, but in both cases, the maximum achievable performance depends on the efficiency of computation per unit of energy. Focusing primarily on performance for high-speed circuits will result in too much power dissipation. Focusing only on energy for mobile applications is equally inadequate, since this approach rarely achieves the required performance. The correct optimization either minimizes energy consumption subject to a throughput constraint, or maximizes the amount of computation for a given energy budget.

Manuscript received December 11, 2003; revised April 15, 2004. This work was supported in part by MARCO Contracts CMU 2001-CT-888, GSRC 98-DT-660, and Georgia Tech B-12-D00-S5.

D. Marković, B. Nikolić, and R. W. Brodersen are with the Berkeley Wireless Research Center, University of California, Berkeley, CA 94704 USA (e-mail: dejan@eecs.berkeley.edu).

V. Stojanović and M. A. Horowitz are with Stanford University, Stanford, CA 94305 USA.

Digital Object Identifier 10.1109/JSSC.2004.831796

This new relationship between performance and energy forces a change in design techniques. Using traditional approaches, architects attempt to create a machine organization that has the "best" performance. Block designers take this organization and try to build each block such that it achieves peak performance. If energy efficiency is the key in achieving high performance, optimizing each layer individually for speed will not lead to an optimal design; rather, it will lead to a design that dissipates too much power. Instead, the most power-efficient optimization techniques must be applied to the design first, followed by others, until the desired performance or power consumption goal is reached.

"True energy-performance optimization" methods explore a multidimensional search space across various layers of the design abstraction allowing for a comparison of power and performance of different solutions. Optimization is performed at three layers of abstraction: system architecture optimization (outer layer), micro-architectural optimization (intermediate layer), and fixed circuit topology optimization (inner layer). Inner layer optimizations deal with circuit-specific supply voltage, threshold voltage, and gate sizes, which must be propagated to higher layers of abstraction to yield globally optimal solutions.

We will show that this drive for energy-efficient designs leads to much higher leakage currents. Furthermore, as the ratio of leakage-to-active power increases, the optimal architecture and circuits also change. From a power budget perspective, leaky gates are expensive when they are inactive, so they must be kept as active as possible leading to deeply pipelined rather than parallel architectures.

Section II is an overview of common optimization techniques that generally consider one variable in the optimization such as gate size, supply or threshold voltage. However, all variables must be jointly considered in the optimization to yield the highest energy efficiency. Before we describe the joint optimization in Section III, we present energy and delay models which we use to develop a sensitivity-based infrastructure for circuit optimization. This framework is used to optimize a 64-bit adder by jointly tuning sizing (W), supply (V_{dd}), and transistor threshold (V_{th}) during the optimization. In Section IV, we show how results from the circuit-level optimization provide insight for the micro-architectural optimization. The best choice of circuit topology and optimal level of parallelism is investigated by combining optimal energy–delay tradeoff curves corresponding to various circuits. Design issues such as determination of optimal balance between leakage and switching power, optimal V_{dd} and V_{th}, and investigation of energy–area tradeoffs are described. Section V concludes the paper.

II. Optimization Methods

The methods for achieving optimum performance are well explored. Establishing the balance between performance and power consumption has been a popular research topic in the past as well. An optimum in the energy–delay space has been searched for through minimization of objective functions that combine energy and delay. Minimizing the energy–delay product (EDP) [1], [2] of a circuit results in a particular design point in the energy–delay space where 1% of energy can be traded off for 1% of delay. Although the EDP metric is useful for comparison of different implementations of a design, the design optimization points targeting EDP may not correspond to an optimum under desired operating conditions. Metrics in a general form of $E \cdot D^{\mathrm{n}}$ [3] or energy–performance ratio by Hofstee [4] have been used instead. For example, the $E \cdot D^2$ metric [5] puts more weight on the delay than the energy, and is a V_{dd}-invariant metric. Minimizing $E \cdot D^{\mathrm{n}}$, however, has limited applicability since it gives only one (E, D) point in the energy–delay space at which the energy E is minimized for a fixed delay D. A complete understanding of the energy–delay tradeoff for a design is obtained by minimizing the energy subject to an arbitrary delay constraint. In this paper we use sensitivities to formalize the tradeoff between energy and performance. Sensitivity is defined as the absolute gradient of energy to delay with respect to a change in some design variable.

There are usually several tuning variables that can be exploited to trade off energy for performance at various levels of design hierarchy. The tradeoff achieved by tuning some design variable x is given by the sensitivity to variable x:

$$S_x(X) = \left.\frac{\partial E/\partial x}{\partial D/\partial x}\right|_{x=X}. \tag{1}$$

This quantity represents the amount of energy that can be traded for delay by tuning variable x. As pointed out by Zyuban and Strenski [6], the energy-efficient design is achieved when the marginal costs of all the tuning variables are balanced.

Gate size W, supply voltage V_{dd}, and change in threshold voltage ΔV_{th} are considered as tuning knobs in the circuit optimization. The true energy minimization method always exploits the tuning variable with the largest capability for energy reduction. A fixed point in the optimization is reached when the energy reduction potentials of all tuning variables are equal.

Sizing optimization of digital circuits has been explored extensively resulting in several optimization tools such as TILOS [7], JiffyTune [8], and EinsTuner [9]. Most such tools can at least approximate energy-constrained sizing by constraining the total transistor width available for the circuit. In addition, a number of researchers derived analytical solutions for area and energy optimization through gate sizing. The analysis is typically restricted to simple logic gates and inverter chains [10], [11]. Like TILOS, we use a simple analytical timing model, so we can guarantee a convex optimization problem, but we explicitly model the delay dependence on V_{dd} and V_{th} allowing us to perform multivariable optimization.

Supply voltage scaling is another common technique that is used to minimize energy under performance constraints. It was one of the key techniques in the low-power DSP work of Chandrakasan *et al.* [12] and has been practically demonstrated in [13] and [14]. With the emerging importance of leakage power consumption, threshold voltage becomes a critical tuning variable and is generally considered together with supply voltage. Liu and Svensson hinted about the existence of optimal supply and threshold for a given design [15]. Gonzalez *et al.* [2] investigated joint supply and threshold voltage scaling for energy–delay product minimization. Kuroda *et al.* [16] and Nose and Sakurai [17] extended this work and proposed closed-form expressions for optimum supply, threshold, and leakage-to-switching power ratio.

We expand prior work by developing a sensitivity-based optimization framework that is applied to multivariable optimizations across several layers in the design abstraction. Extensive treatment of the circuit-level optimization has been reported in [18] and [19]. We review some of the key concepts here and apply them to explore tradeoffs at the micro-architectural level, which is the focus of this paper.

III. Circuit-Level Optimization

The efficiency of W, V_{dd}, and ΔV_{th} optimizations can be estimated from the profile of energy dissipation in the circuit by analyzing sensitivities. Circuit topologies are distinguished by two key features: off-path loading and path reconvergence. An optimization using these topological properties was analyzed in [18] and [19]. Here, we introduce our optimization framework and use an adder example to illustrate the effectiveness of tuning W, V_{dd}, and ΔV_{th}. The first step toward the sensitivity-based optimization is developing energy and delay models for our technology.

A. Technology Calibration

The energy and the delay of a logic gate are functions of its size, supply voltage, and transistor threshold voltage. In order to calculate the sensitivities of larger logic blocks comprised of simple logic gates, it is necessary to develop simple and accurate models of the energy and delay of the gate.

Delay Model: While there are many different models that can be used, we follow our prior work [18], [19] and use the alpha-power law model of [20] as a baseline for derivation of the gate delay formula:

$$\begin{aligned} t_p &= \frac{K_d \cdot V_{dd}}{(V_{dd} - V_{on} - \Delta V_{th})^{\alpha_d}} \cdot \left(\frac{w_{out}}{w_{in}} + \frac{w_{par}}{w_{in}}\right) \\ &= \tau_{ref} \cdot g \cdot \left(h + \frac{p}{g}\right) \\ &= \tau_{ref} \cdot d. \end{aligned} \tag{2}$$

This is a curve-fitted expression and parameters V_{on} and α_{d} are intrinsically related, yet not necessarily equal, to the transistor threshold voltage and velocity saturation index. ΔV_{th} is the change from the standard threshold voltage given by technology; K_d is a fitting parameter; $h = w_{out}/w_{in}$ is the electrical fan-out of a gate; and w_{par}/w_{in} is a measure of its intrinsic delay. Gate and parasitic capacitance are both made linear and the effects of transistor capacitance nonlinearities are lumped into the fitting parameters V_{on}, α_{d}, and K_d. The delay model

fits SPICE simulated data within 5% over a range of supply voltages from $0.4V_{dd}^{max}$ to V_{dd}^{max} and fanout factors from 2 to 10, assuming equal input and output rise and fall times [18].

Using the linear delay model from the method of logical effort [21], the delay formula can be expressed simply as a product of the process-dependent time constant τ_{ref} and the unit-less delay d. This delay d consists of the intrinsic delay p due to the self-loading of the gate, and the fanout delay $g \cdot h$ which is the product of the logical effort g and the fanout h. Logical effort g represents the relative ability of a gate to deliver current for a given input capacitance. Fanout h is the ratio of the total output to input capacitance. The simple linear delay model naturally extends to logic paths and multiple-supply voltages [18]. The delay of a logic path is simply $D = d_{path} \cdot \tau_{ref}$, where d_{path} is the sum of the normalized gate delays along the path.[1]

Energy Model: We consider two components of energy: switching and leakage. The switching component is the standard dynamic energy term given by

$$e_{Sw} = \alpha \cdot K_e \cdot (w_{out} + w_{par}) \cdot V_{dd}^2 \tag{3}$$

where $K_e \cdot w_{out}$ is the load capacitance, $K_e \cdot w_{par}$ is the self-loading of the gate, and α is the probability of an energy-consuming transition at the output of the gate.

Static leakage of a logic gate at $V_{gs} = 0$ is modeled as

$$e_{Lk} = D \cdot w_{in} \cdot I_0(S_{in}) \cdot e^{-\frac{\left(V_{th}^{ref} + \Delta V_{th} - \gamma \cdot V_{dd}\right)}{V_0}} \cdot V_{dd} \tag{4}$$

where $D = d_{critical-path} \cdot \tau_{ref}$ is the cycle time, $I_0(S_{in})$ is the normalized leakage current of the gate with inputs in state S_{in}, V_{th}^{ref} is the standard threshold voltage provided by technology, and $V_0 = n \cdot kT/q$ and γ account for the sub-threshold slope and DIBL factor, respectively. The model in (4) is calibrated in HSPICE over the full range (defined by lower and upper bounds) of design parameters V_{dd}, V_{th}, W, and also over the entire set of states S_{in} for each of the gates.

In large circuit blocks, the logic state and the switching probability of the internal gates are obtained through logic simulation. This way, gate-level models from (3) and (4) are extended to compute the total circuit energy.

Reference Design: Our baseline design is optimized for minimum delay D_{min} through gate size optimization, under the maximum supply voltage V_{dd}^{max} specified by the technology reliability limit, and nominal threshold voltage V_{th}^{ref} for this technology. Our nominal threshold voltage is the low-V_{th} from a standard dual-V_{th} 0.13-μm technology, and we label this voltage as reference V_{th}^{ref}, corresponding to $\Delta V_{th} = 0$. In our technology, $V_{th}^{ref} = 0.34$ V, $V_{dd}^{max} = 1.2$ V. The minimum delay D_{min} is achieved for some specified output load C_L and a fixed input capacitance C_{in}. All capacitances are normalized to the input capacitance of a unit inverter. In the optimization procedure, we specify some percentage incremental change in delay, d_{inc}, relative to the reference point. The energy is minimized for the new target delay $D = D_{min}(1 + d_{inc}/100)$, by using supply voltage, threshold voltage, gate size, and optional buffering as optimization variables.

[1]In this paper, we use small letters to label gate parameters and capital letters to label circuit and system parameters.

The delay-constrained energy minimization via transistor sizing represents a geometric program which has a convex formulation [7]. In supply optimization, our investigations include global supply reduction and the use of two discrete supplies. We limit supply voltage to decrease from input to output of a block assuming that low-to-high level conversion is done in registers. Sizing is allowed to change continuously. Conceptually, an energy-efficient solution attempts to maintain balance in the sensitivities to all individual tuning variables.

B. Sensitivity to Gate Sizing, Supply, and Threshold Voltage

The sensitivity of circuit energy to delay due to a change in size of a gate in stage i is given by (5), where $e_{C,i} = \alpha \cdot K_e \cdot (w_{in,i} + w_{par,i}) \cdot V_{dd}^2$ represents the switching energy due to capacitances of stage i (this is not the energy consumed at the output of stage i), $e_{Lk,i}$ is the leakage energy of stage i as given by (4), E_{Sw} and E_{Lk} are the total switching and the total leakage energy, respectively. Parameter $h_{eff,i} = g_i \cdot h_i$ is the effective fanout of stage i.

$$\frac{\partial E_{Sw}/\partial w_i}{\partial D/\partial w_i} = -\frac{e_{C,i}}{\tau_{ref} \cdot (h_{eff,i} - h_{eff,i-1})} \tag{5.a}$$

$$\frac{\partial E_{Lk}/\partial w_i}{\partial D/\partial w_i} = \frac{E_{Lk}}{D} - \frac{D \cdot e_{Lk,i}}{\tau_{ref} \cdot (h_{eff,i} - h_{eff,i-1})}. \tag{5.b}$$

Equation (5) shows that the largest potential for energy savings occurs at the point where the design is sized for minimum delay with equal effective fanouts h_{eff}, resulting in infinite sensitivity. Intuitively, the delay cannot be reduced beyond the minimum achievable delay, regardless of how much energy is spent. While decreasing gate size decreases the leakage current, it also increases the cycle time D, which increases the leakage energy. At the point where the sensitivity in (5b) becomes positive, the leakage energy will start increasing with further gate size reduction due to longer cycle time. In order to achieve equal sensitivity in all stages, the difference in the effective fanouts must increase in proportion to the energy of the gate, which closely ties the circuit energy profile with optimal sizing [18]. For example, this matches with the variable taper result of Ma and Franzon [11] for energy minimization of a delay constrained inverter chain.

The sensitivity of total circuit energy to delay increase due to global supply reduction is given by (6). Similar to the sizing approach, the design sized for the minimum delay at maximal supply voltage offers the greatest potential for energy reduction. This potential diminishes with the reduction in supply voltage since the energy $E = E_{Sw} + E_{Lk}$ decreases, cycle time D increases, and the V_{on}/V_{dd} ratio increases. Power supply reduction has a two-fold impact on the leakage energy, (6b): the leakage energy E_{Lk} increases because of increase in cycle time D, but it also decreases because of the supply reduction and because of the reduced DIBL effect γ. The resulting tendency is a decrease in the leakage energy with supply reduction, which results in negative sensitivity of the leakage energy to delay.

$$\frac{\partial E_{Sw}/\partial V_{dd}}{\partial D/\partial V_{dd}} = -\frac{2E_{Sw} \cdot (1 - V_{on}/V_{dd})}{D \cdot (\alpha_d - 1 + V_{on}/V_{dd})} \tag{6.a}$$

$$\frac{\partial E_{Lk}/\partial V_{dd}}{\partial D/\partial V_{dd}} = -\frac{E_{Lk}}{D} \cdot \left(\frac{(1 - V_{on}/V_{dd}) \cdot (1 + \gamma \cdot V_{dd}/V_0)}{\alpha_d - 1 + V_{on}/V_{dd}} - 1 \right). \tag{6.b}$$

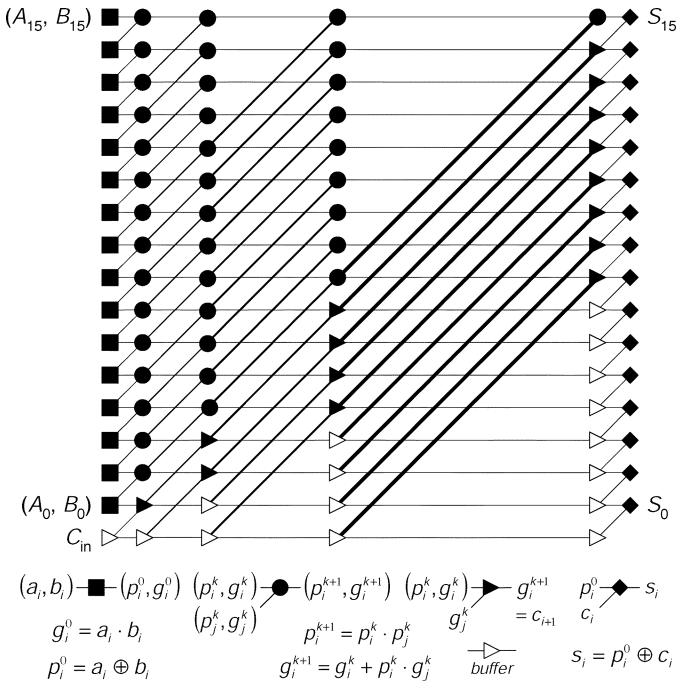


Fig. 1. Diagram of a 16-bit Kogge–Stone tree adder.

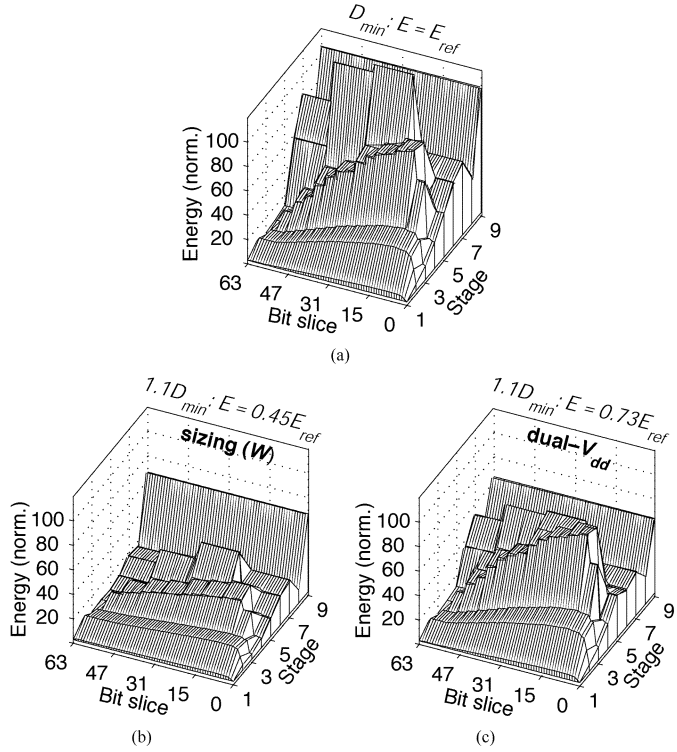


Fig. 2. Energy distribution in a 64-b adder. (a) Design sized for minimum delay at V_{dd}^{max} and V_{th}^{ref}. (b) Design with 10% additional delay, optimal sizing. (c) Design with 10% additional delay, dual-V_{dd} optimization. Each sum output is loaded with $C_L = 32$, input data activity is 10%.

In dual-supply voltage optimization, the same formula holds, where E_{Sw}, E_{Lk} and D represent the total switching energy, the total leakage energy, and the delay of the stages under the reduced supply voltage, respectively.

The sensitivity of energy to delay due to the change in threshold voltage is given by (7). This sensitivity decays exponentially with the increase in ΔV_{th} because E_{Lk} is an exponential function of ΔV_{th}, as in (4).

$$\frac{\partial E/\partial(\Delta V_{th})}{\partial D/\partial(\Delta V_{th})} = -\frac{E_{Lk}}{D} \cdot \left(\frac{V_{dd} - V_{on} - \Delta V_{th}}{\alpha_d \cdot V_0} - 1 \right). \quad (7)$$

The exponential dependence of the leakage energy on ΔV_{th} limits the optimization range. Lowering the threshold voltage while maintaining circuit speed for designs with very low leakage allows for a reduced V_{dd} and therefore reduced switching energy. The total energy is minimized when the leakage and switching components of energy are comparable [22].

C. Optimization Example: A 64-bit Adder

The circuit energy profile is crucial in providing insight to choosing the tuning variable that is most effective in reducing the total energy of the circuit. We illustrate this on a 64-bit Kogge-Stone carry-lookahead tree adder [23]. For brevity, Fig. 1 shows a 16-bit tree as an example. Various symbols in the figure correspond to different logic operations [24], as indicated in the figure. Dot operators compute propagate and generate signals in a parallel-prefix tree. Significant features of this adder topology include reconvergent fanouts inside propagate-generate blocks, long wires, and multiple active outputs.

The initial sizing of the reference adder attempts to make all the paths in the adder equal to the critical path for a fair comparison. We allocate each gate in the adder to a bit slice, which is the natural partitioning for tree adders. Fig. 2 shows the resulting energy map for minimum delay, as well as the case when a 10% delay increase is allowed. In this type of adder, the switching activity of propagate logic diminishes rapidly with the number of stages, and most of the switching energy is consumed by the generate logic in the later stages. The internal energy peak in Fig. 2(a) occurs due to the large activity of the propagate logic that is comparable to that of generate logic close to the input of the adder, and also due to the large load presented by the gates which drive long wires in the final stages of the adder.

The adder energy map of Fig. 2(b) shows that the gate size optimization is very effective in circuit topologies in which energy peaks occur inside the block. In such cases, gate sizes have not yet reached their bounds which allows for energy reduction by optimizing gate sizes. The data indicates that for a 10% excess delay, a 55% decrease in energy is possible using transistor sizing under V_{dd}^{max}, while only 27% is saved by using two supplies without resizing, as shown in Fig. 2(c). Reducing the supply over the whole block yields even lower energy reduction at only 17%. Using multiple supplies is therefore less effective than sizing in designs where the peak of energy consumption occurs inside the block. In order for the supply optimization to affect the energy peak, the delay of all stages following the peak needs to increase, thus reducing the marginal return. On the other hand, sizing can selectively target energy peaks, by focusing on downsizing of the selected internal gates first, yielding higher energy returns than the discrete supply optimization.

Plots in Fig. 3 for an inverter chain, a memory decoder, and an adder provide some insight into which parameters are more ef-

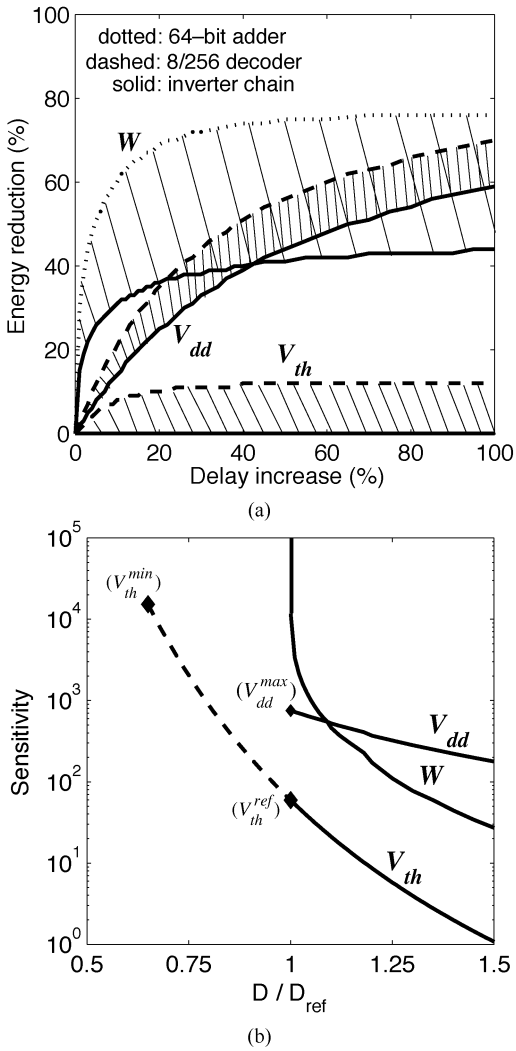


Fig. 3. (a) Energy reduction due to W, V_{dd} and V_{th} in inverter chain, 8/256 memory decoder, and 64-bit adder. Only cases with min and max energy reduction are shown. (b) Sensitivity to V_{dd}, V_{th}, and W in the adder example. V_{dd}^{max} and V_{th}^{min} are bounds on V_{dd} and V_{th}.

fective in different regions of delay. The data in Fig. 3(a) shows the potential energy reduction due to W, V_{dd}, and ΔV_{th} in the example circuits. The general trend is the superior performance of sizing at small incremental delays stemming from infinite sensitivity at the reference point which is still large at the 10% increment point. This can be observed in Fig. 3(b) which plots the energy–delay sensitivity to each of the tuning variables in the adder example. At larger delays, sensitivity to sizing diminishes and supply voltage becomes more effective providing larger energy savings. The threshold voltage primarily affects leakage energy which is significant in designs with lots of inactive gates, such as memory decoders [19]. We can take the advantage of V_{th}^{ref} being too high in most circuits and reduce V_{th} to speed up the circuit even beyond the reference delay. This in turn creates the opportunity for other variables such as V_{dd} or W to exploit the timing slack for overall energy reduction. The sensitivity gap between W, V_{dd}, and ΔV_{th} can be exploited to effectively perform multivariable optimization, leading to the most energy- efficient solution.

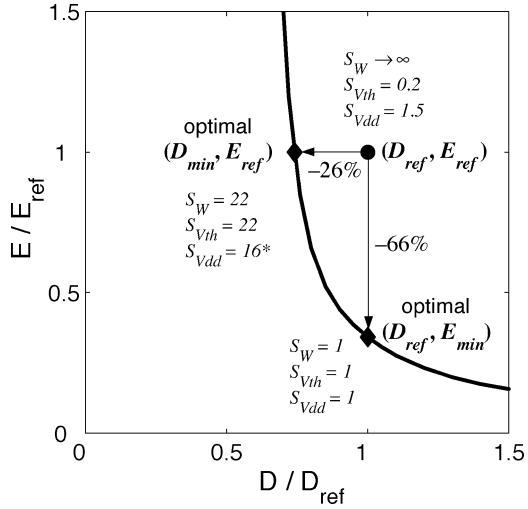


Fig. 4. Optimal energy–delay tradeoff in a 64-bit adder after performing $V_{dd} - V_{th} - W$ optimization. Reference is the design sized for minimum delay under V_{dd}^{max} and V_{th}^{ref}. Sensitivity to each of the tuning variables is marked on the graph.

Let us investigate the use of all three optimization variables together in the adder example. The plot in Fig. 4 illustrates the position of the reference design point for the adder relative to the optimal energy–delay tradeoff curve obtained by jointly optimizing gate size, supply and threshold voltages. The reference is the design sized for minimum delay under V_{dd}^{max} and V_{th}^{ref}. As seen in the plot, there is still significant room for improvement starting from the reference design, because the sensitivities differ at that point. In the reference design, the sizing sensitivity is infinite, the supply sensitivity is 50% higher, and the threshold sensitivity is five times smaller than the sensitivity at the optimal point, which is used as the baseline case in Fig. 4.

After balancing the sensitivities by downsizing the gates and decreasing supply and threshold, about 65% of energy is saved without any delay penalty. This is illustrated in Fig. 4, where the reference design (D_{ref}, E_{ref}) moves down on the y-axis to the optimal design point (D_{ref}, E_{min}) on the energy-efficient curve. Alternatively, we can maintain the energy and achieve the speedup of about 25%. Although we are still on the energy-efficient curve, the data in Fig. 4 shows that the sensitivities are not the same in this case, because V_{dd} has reached its upper limit, V_{dd}^{max}. Under such conditions, the circuits achieve the optimal solution under a constraint when some of the variables reach their limits.

Typically, only a subset of tuning variables is selected for optimization. With a proper choice of the two variables, the designer can obtain nearly the minimal energy for a given delay. In our case, for delays close to D_{ref}, these variables are sizing and threshold voltage since there is the largest gap between the sizing and threshold sensitivities around the nominal delay point, as illustrated in Fig. 3(b). The data in Fig. 4 shows that circuit optimization is really effective only in the region of about $\pm 30\%$ around the reference delay, D_{ref}. Outside this region, optimization becomes costly either in terms of delay or energy, and a more efficient variable must to be introduced at another level in the design hierarchy. This naturally expands the optimization to the micro-architectural level.

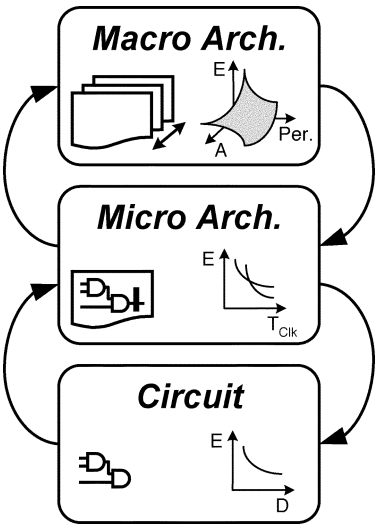


Fig. 5. Block diagram illustrating various abstraction layers in the optimization. Energy is the objective function at the circuit and micro-architectural layers, while achieving proper energy–area tradeoff is the objective at the marco-architectural layer.

IV. Micro-Architectural Optimization

Energy savings of about 65% in the adder example are possible without any delay penalty by simply choosing appropriate values of supply, threshold, and circuit size. However, individual circuit examples may be misleading. For example, if the energy of the adder, or some other functional-unit block, is a much smaller fraction of the total processor energy than that of registers and clocking, then it might be more beneficial to lower the power of the registers (make the latches slower) and increase the power of the adder (make the adder faster).

The optimal energy–delay tradeoff curves from the circuit level are used to hierarchically extend our optimization to larger blocks, as illustrated in Fig. 5. These tradeoff curves coupled with optimal W, V_{dd}, and V_{th} are strategically combined to obtain the optimal energy–performance tradeoff for circuit macros. Along this optimal energy–performance curve, we can select appropriate circuit topology in the pipeline or choose optimal level of parallelism based on the circuit optimization results. However, optimal V_{dd} and V_{th} in the individual circuits, when combined as a pipeline, rarely coincide due to their differing topology. In order to achieve the most energy-efficient solution under some global V_{dd} and V_{th}, several iterations may be required to optimize all the circuits under that particular V_{dd} and V_{th}. This includes finding optimal V_{dd} and V_{th} for a given architecture based on balancing the leakage and switching components of energy under some performance constraint.

The nature of performance constraints is different at various abstraction layers in the optimization. For instance, the performance of a circuit is measured by the circuit delay, while the performance of micro- or macro-architecture is related to the cycle time or the number of instructions per cycle. Each new layer in the optimization introduces more degrees of freedom, such as level of parallelism at the micro-architectural layer or area of the macro-architecture. However, designs of higher complexity can be still optimized based on the optimal energy–performance tradeoffs of their building blocks. This is computationally much more efficient than performing large-scale optimization at the gate level.

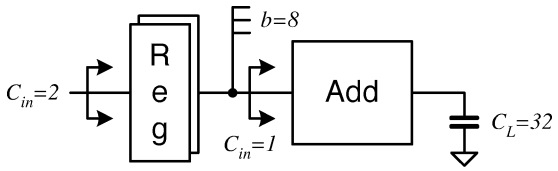


Fig. 6. Simplified model of one bit-slice of a 64-bit ALU.

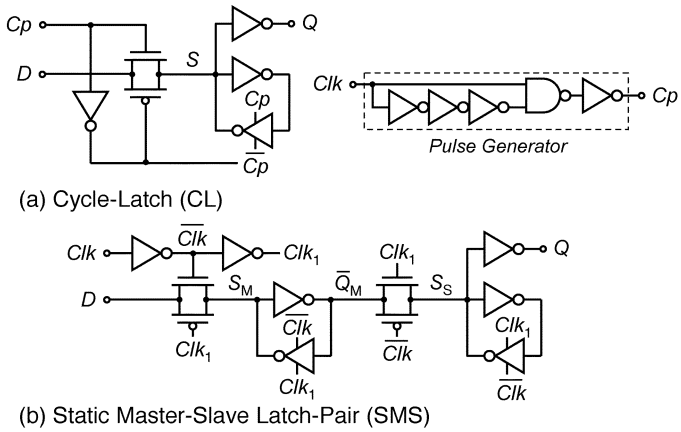


Fig. 7. Flip-flops used in implementation of the ALU register in Fig. 6. (a) High-performance cycle latch (CL). (b) Low-energy static master–slave latch pair (SMS).

A. Choosing Optimal Circuit Topology

We demonstrate our modular approach on the optimization of a pipeline that jointly optimizes registers and logic. When cascading heterogeneous circuit blocks, such as registers and logic, the total available delay has to be optimally divided among the circuit blocks to achieve minimal energy. As an example, we analyze a simplified model of an ALU as shown in Fig. 6. It consists of two registers that drive a 64-bit Kogge–Stone tree adder. The register is comprised either of simple cycle latches (CL) [25] or static master–slave latch pairs (SMS) [26], as shown in Fig. 7. The output load C_L is due to registers, wire, and bus capacitances; term b is the branching effort [21] at the output of the gate.

The input capacitance of the adder $C_{in}(Add)$ is fixed in our optimization in order to reduce search space in the global optimization. Without a fixed C_{in} constraint, optimal $C_{in}(Add)$ would be larger than minimum only in a very narrow range around the minimum delay. For delays farther away from the minimum delay, $C_{in}(Add)$ would quickly reach the lower bound due to large branching at the output of the register. Therefore, fixing the input capacitance of the adder is a good heuristic which also allows for a modular design.

The register and adder significantly differ in their switching activity. The register has higher average activity primarily due to a large activity factor of the clocked nodes. This results in a lower initial E_{Lk}/E_{Sw} ratio in the registers than in the adder. For this reason, the optimal value of V_{dd} and V_{th} would tend to be lower in the registers than in the adder. In reality, however, we are usually constrained to one core-level V_{dd} and V_{th} making

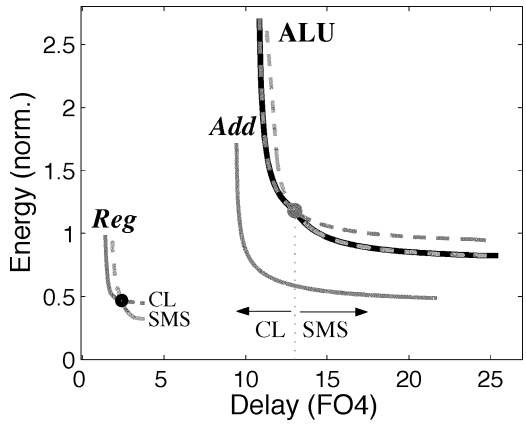


Fig. 8. Energy-efficient curves in register, adder, and ALU after performing gate size optimization. The dots indicate transition between CL- and SMS-based register.

them *global* variables. Hence extra effort will be spent in sizing the register during the $V_{dd}-V_{th}-W$ optimization of the ALU. The goal is to equalize sensitivities to W for both the *Reg* and *Add* blocks to obtain the most energy-efficient solution. With V_{dd} and V_{th} fixed, sizing the gates inside each of the blocks simply compensates for the intrinsic mismatch in sensitivity due to logic topology and activity.

Combining results of individual optimizations of *Reg* and *Add* blocks from the circuit level, the total energy of the ALU is minimized subject to a cycle time constraint. Fig. 8 shows the energy of the ALU after optimal sizing under V_{dd}^{max} and V_{th}^{ref}. Energy-efficient curves for registers comprised of CL or SMS latches combine to define a composite energy-efficient curve for the *Reg* block, as shown in Fig. 8 by the solid line. For each target ALU delay, points from optimal *Add* and *Reg* curves (solid lines) are chosen to minimize the overall energy of the ALU. Dashed lines show sub-optimal designs using an incorrect choice of the register topology. The optimal solution confirms that high-performance designs naturally use fast cycle latches, while simple SMS latch pairs are suitable for low-energy designs.

The scope of energy-efficient ALU optimization is extended through the selection of optimal register topology. This can be best illustrated by observing the energy–delay sensitivity in the register and in the adder. The goal of the sizing optimization at the circuit level—that sensitivity in all stages should be equal—applies here as well: the sensitivity of the adder block has to be the same as the sensitivity of the register. Because of the fixed interface between the blocks, the sensitivity of each block simply reduces to the ratio of the change in energy, ΔE_{block}, to the change in delay, ΔD_{block}, due to resizing the block:

$$S_{block}(W)=\frac{\Delta E_{block}(W)}{\Delta D_{block}(W)}. \tag{8}$$

For ALU delays greater than 13FO4, the solution with an SMS-based register becomes more energy efficient because the benefits from sizing of the CL-based register are utilized. Timing slack created by a faster adder can be exploited in the optimization of the register. This leads to an overall energy reduction of the ALU due to higher register sensitivity to sizing.

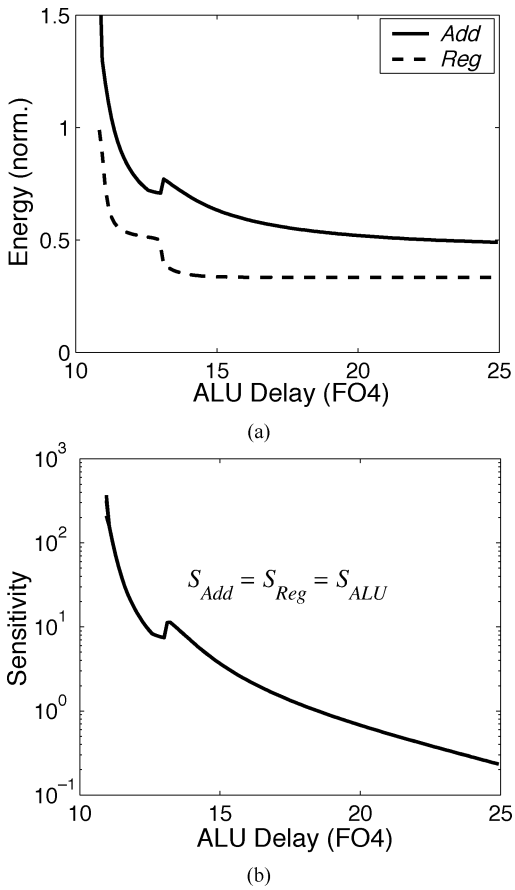


Fig. 9. Plots after optimal sizing and change in register topology. (a) Energy of adder and register when they are combined as the ALU. (b) Corresponding sensitivity of register, adder, and ALU.

The process described above is illustrated in Fig. 9(a), which plots energy of the adder and register when they are combined within an ALU. In an optimized design, sensitivities are balanced, so the sensitivities of the adder, the register, and the ALU are equal, as illustrated in Fig. 9(b). Higher energy efficiency of the ALU due to a change in register topology means higher sensitivity around the point where the change in register topology occurs, extending the range of energy-efficient optimization.

Circuit topology and intrinsic node activity have a large effect on the optimization result. At the optimum tradeoff point, high-activity gates with large numbers of transistors per stage, such as registers, are downsized, while the slack is consumed by upsized and lower-activity units, such as adders. This agrees with the result from [6]: the hardware intensity of the register should be smaller than that of the adder. In other words, at the optimal point, percent energy per percent delay in the register is smaller than that in the adder. Therefore, the operating point of the register is pushed out toward longer delays.

Both fixed topology circuit optimization and cascading heterogeneous blocks have limited scope due to limits on optimization variables. One effective way to extend the scope of optimization is to introduce more tuning knobs. This involves optimization at the micro-architectural level, where the amount of parallelism level or pipelining depth can be exploited as tuning variables.

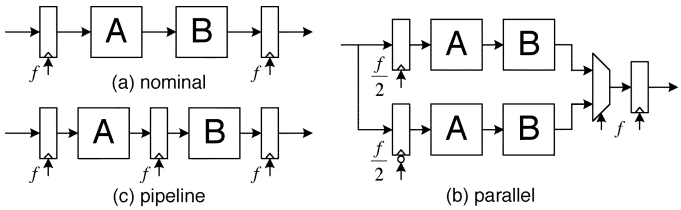


Fig. 10. Micro-architectural design options. (a) Nominal. (b) Parallel. (c) Pipeline.

B. Parallelism Versus Pipelining

This section revisits the example from Chandrakasan *et al.* [12] that evaluates the energy efficiency of a parallel and a pipelined design. We build on previous work by introducing the threshold voltage as an additional tuning variable in the optimization. Schematics of the nominal, parallel, and pipelined circuits are shown in Fig. 10.

Parallelism and pipelining are employed to relax timing constraints on the underlying blocks A and B when the energy of the reference design becomes too costly. In pipelining, an extra register is inserted between blocks A and B, effectively doubling the available computation time for each of the blocks. In a parallel design, the area is doubled by operating the two blocks in parallel. However, the available computation time is also doubled for each block since every other input operand is evaluated in an interleaved fashion.

The nominal design is an add-compare unit which uses the adder described in Section III-C for both the adder (block A) and the comparator (block B). In this example, the SMS latch pairs of Fig. 7(b) are used. The nominal design is first optimized through gate sizing to achieve minimum delay under V_{dd}^{max} and V_{th}^{ref}. Using the throughput of this design as a constraint and energy–delay tradeoffs of the adder and comparator blocks from the inner layer, we can estimate the energy needed for the nominal design and its parallel or pipelined implementation.

In all the designs of Fig. 10, we find the optimal value of the supply and threshold voltage that results in minimum energy for a given throughput constraint; we also find the corresponding E_{Lk}/E_{Sw} ratio. Minimal energy is found by $V_{dd} - V_{th}$ optimization, in which ΔV_{th} is swept from -200 mV to 0 in steps of 5 mV. Each time V_{th} is modified, V_{dd} is adjusted to achieve the target throughput with minimal energy, using the multivariable sensitivity information from the lower-level blocks. The goal of this sweep is to find the optimal point $(V_{dd}^{opt}, V_{th}^{opt})$ for each micro-architecture and to illustrate the trend around the optimal point, as shown in Fig. 11. Energy-per-operation in all three designs is compared to the nominal case which operates at V_{dd}^{max} and V_{th}^{ref}. For each design, the optimal $(V_{dd}^{opt}, V_{th}^{opt})$ point is reached when the supply and threshold voltage sensitivities of the underlying blocks are equal.

It has been shown that parallelism is more energy efficient than pipelining when the leakage energy is about an order of magnitude smaller than the switching energy [12]. However, as devices become leakier, the larger area of parallel design causes the balance between the switching and the leakage energy to occur at a higher supply voltage than in a pipelined design. This is due to lower effective activity of the parallel design. Equivalently, parallelism decreases the amount of time that a device

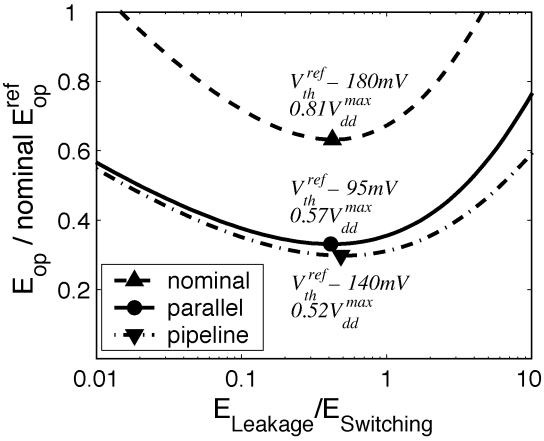


Fig. 11. Energy-per-operation as a function of the leakage-to-switching energy ratio in nominal, parallel, and pipeline designs. All designs operate at the throughput of the nominal design sized for minimum delay under V_{dd}^{max} and V_{th}^{ref}.

spends on computations, thereby increasing the ratio of wasted (leakage) to useful (switching) energy. Hence, a parallel implementation achieves smaller energy savings though the difference is very small. A parallel implementation may still be preferable since the energy saving in the pipelined design depends on determining the ideal locations for pipeline latches. In many systems, these points are hard to find.

We observe that the energy-per-operation as a function of the leakage-to-switching energy ratio has a very shallow minimum, as shown in Fig. 11. This follows from the logarithmic dependence of the E_{Lk}/E_{SW} ratio on the logic depth and activity [19]. In this example, the optimal E_{Lk}/E_{Sw} ratio is around 40% for all thee implementations, roughly corresponding to that of its main sub-block, the adder. When considering extreme circuit examples with significantly different switching activity, such as inverter chains and memory decoders [19], we found that the optimal E_{Lk}/E_{Sw} ratio of these circuits ranged from 0.2 to 0.8. Since the minima of the energy curves are very shallow in the range of leakage-to-switching ratio from 0.1 to 1 (Fig. 11), we conclude that the total energy is minimized at the point where the leakage energy is about half of the active energy.

C. Choosing Optimal Vdd and Vth

The fact that the optimal E_{Lk}/E_{Sw} is around 0.5 provides a way to quickly estimate the optimal V_{dd} and V_{th} in a function block. Using the dependence of critical-path delay and the E_{Lk}/E_{Sw} ratio on V_{dd} and V_{th}, we obtain the result in (9).

$$\Delta V_{th} = V_0 \cdot \ln\left(\frac{(E_{Lk}/E_{Sw})_{init}}{(E_{Lk}/E_{Sw})_{opt}} \cdot \frac{D_{opt}}{D_{init}}\right) \tag{9.a}$$

$$V_{dd}^{opt} = \frac{V_{on}^{init} + \Delta V_{th} + \chi \cdot \frac{\alpha_d - 1}{\alpha_d} \cdot \left(V_{dd}^{init}\right)^{\frac{1}{\alpha_d}}}{1 - \frac{\chi}{\alpha_d} \cdot \left(V_{dd}^{init}\right)^{\left(\frac{1}{\alpha_d} - 1\right)}} \tag{9.b}$$

$$\chi = \frac{V_{dd}^{init} - V_{on}^{init}}{\left(V_{dd}^{init}\right)^{\frac{1}{\alpha_d}}} \cdot \left(\frac{D_{init}}{D_{opt}}\right)^{\frac{1}{\alpha_d}} \tag{9.c}$$

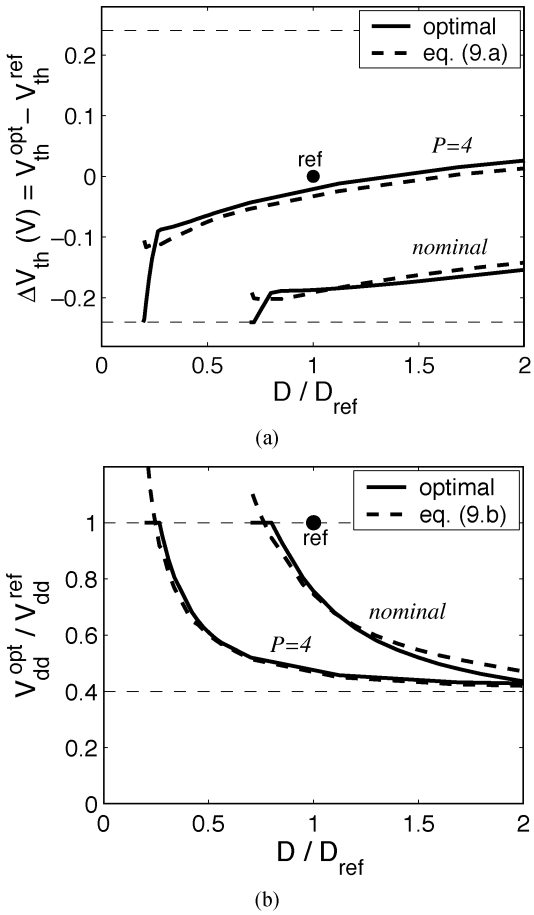


Fig. 12. Plot of (a) change in threshold ΔV_{th} and (b) optimal supply voltage $V_{dd}^{opt}/V_{dd}^{max}$ after performing energy-efficient $V_{dd}-V_{th}-W$ optimization on nominal and parallel-4 designs. Dot represents initial sizing for minimum delay at V_{dd}^{max}, V_{th}^{ref}.

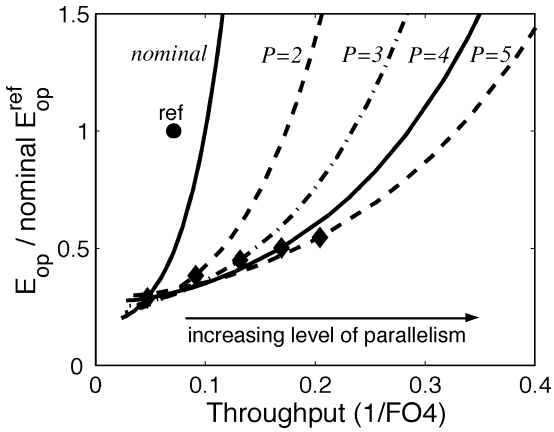


Fig. 13. Energy-per-operation as a function of throughput in energy-efficient designs with levels of parallelism from $P = 2$ to $P = 5$. Delay and energy penalty due to multiplexers is included. Diamond dots indicate min EDP, circle indicates nominal design initially sized for minimum delay at V_{dd}^{max}, V_{th}^{ref}.

where indices $init$ and opt indicate the initial and optimal design points, respectively. The calculation in (9) consists of first changing the threshold voltage to force the leakage energy to be about 50% of the dynamic energy, and then changing V_{dd} to achieve the desired performance.

Equation (9.a) finds the change in V_{th} by estimating the required change in leakage current, and can be easily derived by noticing that the leakage current is equal to the leakage energy divided by cycle time, and assuming that the change in the switching energy is small. Equation (9.b) follows the analysis in [17] and linearizes the alpha power law equation by taking Taylor expansion about V_{dd}^{init}. This expression relates performance to V_{dd} and V_{th} to derive optimal V_{dd} needed to achieve the desired performance D_{opt} under the new V_{th}.

The optimal design point determined above can then be used as an initial point for a new performance constraint, and in this way we can obtain the full energy–delay tradeoff curve for the design. As an example, we calculate optimal V_{dd} and ΔV_{th} for the nominal topology and the topology with parallelism of four across a wide performance range, as shown in Fig. 12. The plots are obtained using Taylor expansion about 1 V in our 0.13-μm technology. The values obtained from (9) and by optimization closely match, thus verifying the analysis. Deviation from ideal ΔV_{th} is from the fact that (9.a) assumes negligible change in V_{dd} and also due to sub-optimal E_{Lk}/E_{Sw} at V_{dd}^{max}. This analysis also provides insight about the tunable range of V_{dd} and V_{th}.

Among the circuit examples we analyzed in [19], the memory decoder is the closest to achieving the optimal E_{Lk}/E_{Sw} ratio of 0.5 under V_{dd}^{max} and V_{th}^{ref} ($E_{Lk}^{ref}/E_{Sw}^{ref} = 10\%$), so optimal V_{th} in the decoder is close to standard V_{th}^{ref}. In the adder ($E_{Lk}^{ref}/E_{Sw}^{ref} = 1\%$) and inverter chain ($E_{Lk}^{ref}/E_{Sw}^{ref} = 0.1\%$), V_{th}^{ref} is about 200 mV higher than optimal, because of the lower E_{Lk}/E_{Sw} in their respective reference designs. These three circuit examples span about three orders of magnitude in the leakage-to-switching energy ratio under V_{dd}^{max} and V_{th}^{ref} and, as such, they can serve as good examples for the V_{dd} and V_{th} tuning range in a particular technology.

The scope of optimization for each topology is limited to about a two-fold increase in its delay relative to the minimum achievable delay, as discerned from Fig. 4. Hence, it is desirable to choose the circuit topology whose minimum achievable delay is positioned relatively close to the desired throughput. Starting from this point, the energy can be best traded off for performance by varying a design knob such as level of parallelism. At this point the marginal costs of decreasing delay in the reference design are the largest yielding the highest potential for energy reduction.

D. Optimal Level of Parallelism

Parallelism is most efficient when the target delay is lower than the minimum achievable delay of the underlying blocks. Parallelism of level P implicitly increases the allowable delay of the underlying circuits about P times. The graph in Fig. 13 shows the energy–performance space for designs exhibiting parallelism from $P = 2$ to $P = 5$, together with our nominal design. The results are obtained from joint supply-threshold-size optimization, with the external load of $C_L = 32$. Delay and energy overhead due to the additional multiplexer at the output is included in the optimization. The data shows that a parallel architecture provides an increase in performance at a very small marginal cost in energy. As a result, by adding more parallel

units, it is possible to improve the performance/throughput beyond the maximal throughput of the nominal design. For instance, parallelism of five provides about a three-fold performance increase at unit energy, as shown in Fig. 13. Parallelism is also an option for energy reduction, which is a well known result [12]. The biggest energy savings due to parallelism are achieved when the sensitivity to circuit parameters of the reference design become very large.

Minimum EDP is the point at which any given architecture has equal marginal cost in energy and delay, allowing for energy-efficient optimization around that point. At the minimum EDP point in Fig. 13, the performance of our design increases with increasing levels of parallelism. This indicates that added parallelism is suitable for boosting performance. Additionally, allowing more levels of parallelism gives us a wider range of performance over which energy may be optimized. As a practical rule, for performance targets below minimum EDP, it is most energy-efficient to choose the micro-architecture with a reduced level of parallelism; for performance targets above minimum EDP we should increase the level of parallelism. While parallelism is a very efficient technique for improving the performance, the area of the parallel design is, to a first order, in linear proportion with the level of parallelism P. Therefore, one must always keep in mind the energy–area tradeoff of parallel solutions.

E. Energy-Area Tradeoff

Both area and energy affect the overall cost of a design. Area is most commonly related to the dollar cost of fabricating a chip, while energy is associated with chip cooling or battery capacity in portable designs. Intuitively, the cost of a design is minimized when an optimal tradeoff between energy and area is reached. We can formulate the design cost function $C(x)$ such that it considers both energy and area as shown in (10).

$$\begin{aligned} &\text{minimize} \quad C(x) = E(x) + \beta \cdot A(x) \\ &\text{subject to} \quad D(x) \leq D_{con}. \end{aligned} \tag{10}$$

The quantity $x \in R^{\mathrm{n}}$ is an n-dimensional vector of tuning variables; $E(x)$ and $A(x)$ are the total energy and area of the design, respectively. Parameter β is the weight-factor that defines contribution of area in the design cost, and D_{con} is the delay or performance constraint. Some of the optimization variables do not affect area, for example the supply and threshold voltages affect only energy. Some other variables such as parallelism or time-multiplexing affect both energy and area, with area impact being much larger when these techniques are applied to large blocks.

We use the concept of time-multiplexing illustrated in Fig. 14 to highlight the tradeoff between energy and area. Time-multiplexing reduces the area at the expense of some increase in energy. The energy increase is due to multiplexing and increased speed of processing element A, which is assumed to be the 64-bit adder. Therefore, reduction in area or energy cannot be the only goal in the optimization since there is a tradeoff between energy and area.

The tradeoff is clearly observed in Fig. 15. Designs with different levels of parallelism and time-multiplexing in Fig. 15(a)

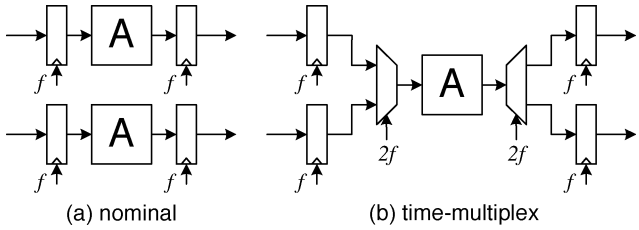


Fig. 14. Micro- and marco-architectural design options. (a) Nominal. (b) Time-multiplex.

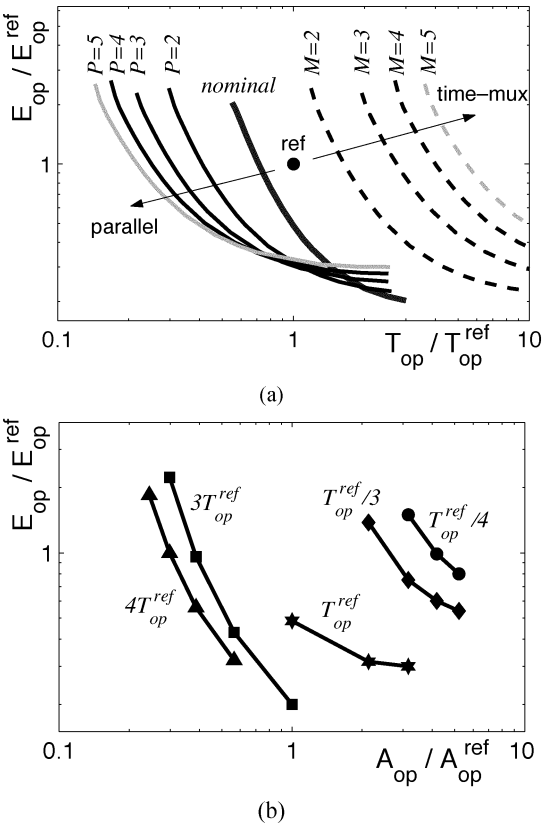


Fig. 15. (a) Energy-delay space for designs with various levels of parallelism and time-multiplexing. (b) Corresponding energy–area tradeoff under performance constraint. All parameters are normalized to the nominal design sized for minimum delay under V_{dd}^{max} and V_{th}^{ref}. T_{op} is time-per-operation.

span a wide performance range. Each of the performance targets can be achieved with several micro-architectural choices differing in energy and area. The optimal choice depends on the energy–area tradeoff, as illustrated in Fig. 15(b). Contours of constant performance in Fig. 15(b) indicate that the high-throughput designs generally require larger area than the low-throughput designs since parallelism must be employed for improving speed. The variations in energy and area in Fig. 15(b) are quite large for our simple adder-based blocks. However, these circuit blocks are just some of the components in a large system, so their impact on the total energy and area of the design would be less significant. The energy–area tradeoff indicates that a larger energy budget allows for smaller circuit area, so the optimum is found at the point where the overall chip cost, (10), is minimized.

V. Conclusion

In order to truly minimize the power in a chip, it is necessary to optimize all design layers simultaneously to achieve the optimal balance between energy and performance. In circuit-level optimizations, the energy can be traded off for delay by the choice of gate sizes, supply and threshold voltages. Relative effectiveness of each of the optimization variables depends on the circuit topology. Simultaneous balancing of W, V_{dd}, and V_{th} sensitivities achieves an optimum design. For example, the energy of a carry-lookahead adder can be reduced by 65% without any delay penalty, relative to the reference design sized for minimum delay. The total performance range at this level is small, only about a factor of two.

Extra degrees of freedom at the micro- and macro-architectural layers, such as choice of circuit topology or level of parallelism, allow for energy-efficient optimization over a wider performance range. In particular, for functions with parallelism, the tradeoff space is quite large. While parallel and pipelined designs can both improve the performance at very small marginal cost in energy, increasing leakage gives pipelined solutions a small advantage if we can find optimum locations for the pipeline latches. Of course, exploiting parallelism costs area, so energy–area tradeoff at a desired performance is often the true metric in minimizing the overall cost of the design.

In this optimization study, we ignored any variability in transistor and wire parameters, which is clearly not the case in practice. In fact, with the advent of new technologies, the impact of process and voltage variations begin to play a more significant role in the overall optimization process [15], [27]. Furthermore, conventional optimization intrinsically increases circuit sensitivity to variations since the optimizer forces all paths and parameters to be critical. It is desirable to augment our framework with tuning knobs that trade off yield with performance and power. This type of optimization is quite challenging, and is an area of ongoing research.

Acknowledgment

The authors thank anonymous reviewers for their helpful suggestions.

References

[1] J. Burr and A. M. Peterson, "Ultra low power CMOS technology," in *Proc. NASA VLSI Design Symp.*, Oct. 1991, pp. 4.2.1–4.2.13.

[2] R. Gonzalez, B. Gordon, and M. A. Horowitz, "Supply and threshold voltage scaling for low power CMOS," *IEEE J. Solid-State Circuits*, vol. 32, pp. 1210–1216, Aug. 1997.

[3] P. I. Penzes and A. J. Martin, "Energy-delay efficiency of VLSI computations," in *Proc. Great Lakes Symp. VLSI*, Apr. 2002, pp. 104–111.

[4] H. P. Hofstee, "Power-constrained microprocessor design," in *Proc. Int. Conf. Computer Design*, Sept. 2002, pp. 14–16.

[5] A. J. Martin, "Toward an energy complexity of computation," *Inform. Process. Lett.*, vol. 77, pp. 181–187, Feb. 2001.

[6] V. Zyuban and P. Strenski, "Unified methodology for resolving power-performance tradeoffs at the microarchitectural and circuit levels," in *Proc. Int. Symp. Low Power Electronics and Design*, Aug. 2002, pp. 166–171.

[7] J. P. Fishburn and A. E. Dunlop, "TILOS: a posynomial programming approach to transistor sizing," in *Proc. Int. Conf. Computer-Aided Design*, Nov. 1985, pp. 326–328.

[8] A. R. Conn, R. A. Haring, C. Visweswariah, P. K. Coulman, and G. L. Morrill, "Optimization of custom MOS circuits by transistor sizing," in *Proc. Int. Conf. Computer-Aided Design*, Nov. 1996, pp. 174–180.

[9] A. R. Conn *et al.*, "Gradient-based optimization of custom circuits using a static-timing formulation," in *Proc. Design Automation Conf.*, June 1999, pp. 452–459.

[10] H. C. Lin and L. W. Linholm, "An optimized output stage for MOS integrated circuits," *IEEE J. Solid-State Circuits*, vol. SC-10, pp. 106–109, Apr. 1975.

[11] S. Ma and P. Franzon, "Energy control and accurate delay estimation in the design of CMOS buffers," *IEEE J. Solid-State Circuits*, vol. 29, pp. 1150–1153, Sept. 1994.

[12] A. P. Chandrakasan, S. Sheng, and R. W. Brodersen, "Low-power CMOS digital design," *IEEE J. Solid-State Circuits*, vol. 27, pp. 473–484, Apr. 1992.

[13] T. Kuroda *et al.*, "Variable supply-voltage scheme for low-power high-speed CMOS digital design," *IEEE J. Solid-State Circuits*, vol. 33, pp. 454–462, Mar. 1998.

[14] T. Burd, T. Pering, A. Stratakos, and R. W. Brodersen, "Dynamic voltage scaled microprocessor system," in *IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers*, Feb. 2000, pp. 294–295.

[15] D. Liu and C. Svensson, "Trading speed for low power by choice of supply and threshold voltage," *IEEE J. Solid-State Circuits*, vol. 28, pp. 10–17, Jan. 1993.

[16] T. Kuroda *et al.*, "Variable supply-voltage scheme for low-power high-speed CMOS digital design," *IEEE J. Solid-State Circuits*, vol. 33, pp. 454–462, Mar. 1998.

[17] K. Nose and T. Sakurai, "Optimization of $\mathrm{V_{DD}}$ and $\mathrm{V_{TH}}$ for low-power and high-speed applications," in *Proc. Asia South Pacific Design Automation Conf.*, Jan. 2000, pp. 469–474.

[18] V. Stojanovic, D. Markovic, B. Nikolic, M. Horowitz, and R. Brodersen, "Energy-delay tradeoffs in combinational logic using gate sizing and supply voltage optimization," in *Proc. Eur. Solid-State Circuits Conf.*, Sept. 2002, pp. 211–214.

[19] R. Brodersen, M. Horowitz, D. Markovic, B. Nikolic, and V. Stojanovic, "Methods for true power minimization," in *Proc. Int. Conf. Computer-Aided Design*, Nov. 2002, pp. 35–42.

[20] T. Sakurai and R. Newton, "Alpha-power law MOSFET model and its applications to CMOS inverter delay and other formulas," *IEEE J. Solid-State Circuits*, vol. 25, pp. 584–594, Apr. 1990.

[21] I. Sutherland, B. Sproul, and D. Harris, *Logical Effort: Designing Fast CMOS Circuits*, 1st ed. San Francisco, CA: Morgan Kaufmann, 1999.

[22] J. Burr and J. Shott, "A 200 mV self-testing encoder/decoder using Stanford ultra-low-power CMOS," in *IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers*, Feb. 1994, pp. 84–85.

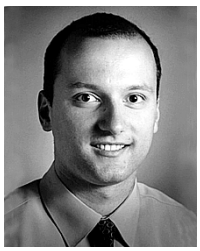
[23] P. M. Kogge and H. S. Stone, "A parallel algorithm for the efficient solution of general class of recurrence equations," *IEEE Trans. Comput.*, vol. C-22, pp. 786–793, Aug. 1973.

[24] Z. Huang and M. D. Ercegovac, "Effect of wire delay on the design of prefix adders in deep-submicron technology," in *Proc. 34th Asilomar Conf. Signals, Systems and Computers*, Oct. 2000, pp. 1713–1717.

[25] J. Tschanz *et al.*, "Comparative delay and energy of single edge-triggered and dual edge-triggered pulsed flip-flops for high-performance microprocessors," in *Proc. Int. Symp. Low Power Electronics and Design*, Aug. 2001, pp. 147–152.

[26] G. Gerosa *et al.*, "A 2.2 W, 80 MHz superscalar RISC microprocessor," *IEEE J. Solid-State Circuits*, vol. 29, pp. 1440–1454, Dec. 1994.

[27] J. Tschanz *et al.*, "Adaptive body bias for reducing impacts of die-to-die and within-die parameter variations on microprocessor frequency and leakage," in *IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers*, Feb. 2002, pp. 422–423.



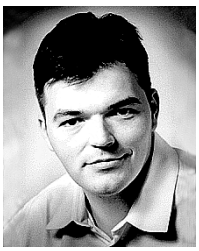
Dejan Marković (S'96) received the Dipl.Ing. degree from the University of Belgrade, Yugoslavia, in 1998 and the M.S. degree from the University of California at Berkeley in 2000, both in electrical engineering. He is currently working toward the Ph.D. degree at the University of California at Berkeley, where he is a member of the Berkeley Wireless Research Center.

He was a Visiting Scholar at the University of California at Davis in 1998. He held internship positions with the Lawrence Berkeley National Laboratory, Berkeley, CA, in 1999, where he worked on pixel-array IC for X-ray spectroscopy, and Intel Corporation, Hillsboro, OR, in 2001, investigating low-energy clocked storage elements. His current research is focused on energy-efficient digital integrated circuits and VLSI architectures for adaptive multiple-input multiple-output wireless communications.

Mr. Marković was awarded the CalVIEW Fellow Award in 2001 and 2002 for excellence in teaching and mentoring of industry engineers through the UC Berkeley distance learning program. In 2004, he received the Best Paper Award at the IEEE International Symposium on Quality Electronic Design. His work is most recently funded by an Intel Ph.D. Fellowship.

Vladimir Stojanović (S'96) received the Dipl.Ing. degree from the University of Belgrade, Yugoslavia, in 1998 and the M.S. degree in electrical engineering from Stanford University, Stanford, CA, in 2000. He is currently working toward the Ph.D. degree at Stanford University, where he is a member of the VLSI Research Group. He has also been with Rambus, Inc., Los Altos, CA, since 2001.

He was a Visiting Scholar with the Advanced Computer Systems Engineering Laboratory, Department of Electrical and Computer Engineering, University of California, Davis, during 1997–1998. His current research interests include design, modeling and optimization of integrated systems, from standard VLSI blocks to CMOS-based electrical and optical interfaces. He is also interested in design and implementation of digital communication techniques in high-speed interfaces and high-speed mixed-signal IC design.



Borivoje Nikolić (S'93–M'99) received the Dipl.Ing. and M.Sc. degrees in electrical engineering from the University of Belgrade, Yugoslavia, in 1992 and 1994, respectively, and the Ph.D. degree from the University of California at Davis in 1999.

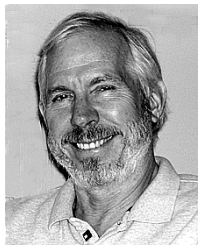
He was on the faculty of the University of Belgrade from 1992 to 1996. He spent two years with Silicon Systems, Inc., Texas Instruments Storage Products Group, San Jose, CA, working on disk-drive signal processing electronics. In 1999, he joined the Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, as an Assistant Professor. His research activities include high-speed and low-power digital integrated circuits and VLSI implementation of communications and signal-processing algorithms. He is coauthor of the book *Digital Integrated Circuits: A Design Perspective*, 2nd ed (Prentice-Hall, 2003).

Dr. Nikolić received the NSF CAREER award in 2003, College of Engineering Best Doctoral Dissertation Prize and Anil K. Jain Prize for the Best Doctoral Dissertation in Electrical and Computer Engineering at University of California at Davis in 1999, as well as the City of Belgrade Award for the Best Diploma Thesis in 1992.

Mark A. Horowitz (S'77–M'78–SM'95–F'00) received the B.S. and M.S. degrees in electrical engineering from the Massachusetts Institute of Technology, Cambridge, in 1978, and the Ph.D. degree from Stanford University, Stanford, CA, in 1984.

He is the Yahoo Founder's Professor of Electrical Engineering and Computer Science at Stanford University. His research area is in digital system design, and he has led a number of processor designs including MIPS-X, one of the first processors to include an on-chip instruction cache, TORCH, a statically scheduled, superscalar processor that supported speculative execution, and FLASH, a flexible DSM machine. He has also worked in a number of other chip design areas including high-speed and low-power memory design, high-bandwidth interfaces, and fast floating point. In 1990 he took leave from Stanford to help start Rambus Inc., Los Altos, CA, a company designing high-bandwidth chip interface technology. His current research includes multiprocessor design, low power circuits, memory design, and high-speed links.

Dr. Horowitz received the Presidential Young Investigator Award and an IBM Faculty development award in 1985. In 1993, he received the Best Paper Award at the IEEE International Solid-State Circuits Conference.



Robert W. Brodersen (M'76–SM'81–F'82) received the Ph.D. degree from the Massachusetts Institute of Technology, Cambridge, in 1972.

He was then with the Central Research Laboratory at Texas Instruments Inc. for three years. Following that, he joined the Electrical Engineering and Computer Science faculty of the University of California at Berkeley, where he is now the John Whinnery Chair Professor and Co-Scientific Director of the Berkeley Wireless Research Center. His research is focused in the areas of low-power design and wireless communications and the CAD tools necessary to support these activities.

Prof. Brodersen has won best paper awards for a number of journal and conference papers in the areas of integrated circuit design, CAD and communications, including in 1979 the W.G. Baker Award. In 1983, he was corecipient of the IEEE Morris Liebmann Award. In 1986, he received the Technical Achievement Awards in the IEEE Circuits and Systems Society and in 1991 from the Signal Processing Society. In 1988, he was elected to be a member of the National Academy of Engineering. In 1996, he received the IEEE Solid-State Circuits Society Award and in 1999 received an honorary doctorate from the University of Lund in Sweden. In 2000, he received a Millennium Award from the Circuits and Systems Society, the Golden Jubilee Award from the IEEE. In 2001 he was awarded the Lewis Winner Award for outstanding paper at the IEEE International Solid-State Circuits Conference and in 2003 was given an award for being one of the top ten contributors over the 50 years of that conference.