Methods for True Power Minimization

Robert W. Brodersen[1], Mark A. Horowitz[2], Dejan Markovic[1], Borivoje Nikolic[1], Vladimir Stojanovic[2]

[1]University of California, Berkeley; [2]Stanford University

Abstract

This paper presents methods for efficient power minimization at circuit and micro-architectural levels. The potential energy savings are strongly related to the energy profile of a circuit. These savings are obtained by using gate sizing, supply voltage, and threshold voltage optimization, to minimize energy consumption subject to a delay constraint. The true power minimization is achieved when the energy reduction potentials of all tuning variables are balanced. We derive the sensitivity of energy to delay for each of the tuning variables connecting its energy saving potential to the physical properties of the circuit. This helps to develop understanding of optimization performance and identify the most efficient techniques for energy reduction. The optimizations are applied to some examples that span typical circuit topologies including inverter chains, SRAM decoders, and adders. At a delay of 20% larger than the minimum, energy savings of 40% to 70% are possible, indicating that achieving peak performance is expensive in terms of energy. Energy savings of about 50% can be achieved without delay penalty with the balancing of sizes, supplies, and thresholds.

1. Motivation

During the past few years the nature of integrated circuit design has slowly changed; the continued scaling of the underlying technology has moved designs from being limited by the amount of functionality on a chip, to being power-constrained. The nature of the power constraints may be different (i.e., the chips in cell phones vs. desktop processors), but in many cases today, and in most cases in the future, the performance one can achieve will depend on the how efficiently that computation can be done per unit of energy. While historically for CMOS circuits there has always been a strong relationship between power and performance, the power of the chip remained within the allowable power envelope; in this scenario, designers focused primarily on achieving the needed performance. Power, if considered, was only checked to ensure that it was not too high. In order to achieve the highest performance in the power-limited scaling regime, one must use the most energy efficient method available, otherwise one will overrun the specified power/energy budget.

This new relationship between peak achievable performance and energy efficiency changes the way one tends to think about design. Traditionally, architects try to create a machine organization that has the "best" performance. This design is then passed to the block designers, who again try to build the blocks in order to achieve the peak performance. If energy efficiency is the key in achieving high performance, optimizing each layer individually will not lead to an optimal design, rather, it will lead to a design that dissipates too much power. Instead, one needs to optimize the design by using techniques that are the most power efficient first, until the desired performance or power is reached.

Merely optimizing for the most energy efficient design is misleading, since this approach rarely achieves needed performance. Thus, the correct optimization typically either minimizes the energy consumption, subject to a throughput constraint, or maximizes the amount of computation for a given amount of energy. Both these design optimizations can be achieved if the tradeoffs between the energy and delay are known.

The dramatic increase in leakage currents in today's (and future) technologies adds another factor to the optimization problem. Since some of the leakage power can be traded off for the dynamic power of the design, the optimization needs to select the correct balance here, as well. 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 since they cost watts when they are inactive. Thus, for leaky technologies, one wants to keep the gates as active as possible, leading to deeply pipelined, rather than parallel, architectures.

Design methods that explore "true power minimization" need to work in a large dimension search space, where power and performance of different solutions are compared. This includes system architecture optimization (outer loop), block-level optimization (intermediate loop), and fixed topology optimization (inner loop). Given that the inner loop optimizations deal with continuous variables, one needs some way to guide the optimization to yield globally optimal solutions. The key is to use the energy-delay tradeoff to piece many different optimizations together. This paper explores this problem, and uses inner loop examples to provide some insight into the method.

2. Power Minimization

Circuit optimization is not a new area and there is now a large research record on energy-constrained performance optimization. Since we utilize many of these techniques, we will briefly review some here, focusing on inner loop optimizations. An inner loop tool does not alter the circuit topology, so the principle variables it affects are transistor sizes, supply voltages, and threshold voltages.

Sizing optimization has been explored extensively resulting in several optimization tools such as TILOS [4] and EinsTuner [11]. Almost all of these tools can at least approximate energy-constrained sizing by constraining the total transistor width available for the circuit. In addition, a

0-7803-7607-2/02/$17.00

number of researchers derived analytical solutions for area and energy sizing optimizations. The analysis is typically restricted to simple logic gates and inverter chains [2], [8].

For many years, using supply to change the energy and performance of circuits has been utilized and was one of the key techniques in the low-power DSP work of Chandrakasan et al. [6]. Changing the supply to optimize the energy for a particular application was proposed in [10], [13].

With the emerging importance of leakage power consumption, threshold voltage becomes an important tuning variable and is mostly considered together with supply voltage. Liu and Svensson hinted about the existence of optimal supply and threshold for a given design [7]. Gonzalez et al. presented a framework for deeper understanding of joint supply and threshold voltage scaling for energy-delay product minimization [9]. Nose and Sakurai extended this work and derived closed-form formulas for optimum supply, threshold, and leakage-to-switching power ratio, minimizing power dissipation for a given technology and operating frequency [14].

By using multiple supply voltages in the circuit, one can optimize energy consumption by reducing the voltage on gates that drive large loads. Hamada et al. derived a set of practical expressions for optimal number and values of discrete supplies, thresholds, and gate sizes. They concluded that no more than three discrete values are needed for each tuning variable [15]. Their analysis was, however, limited to single-variable optimizations.

More recently, researchers have looked at doing multiple optimizations at once. In modern logic design, high threshold transistors are placed in non-critical paths to trade leakage energy for available timing slack. In addition, sizing optimization is combined with dual threshold to exploit the remaining timing slack [16], [17]. An interesting approach has been taken by Zyuban and Strenski [18]. They introduce a method of balancing optimizations at different levels within the design using a unified relative gradient metric they call *hardware intensity*.

In this paper we formalize the tradeoff between energy and delay via sensitivities (i.e. absolute gradients of energy to delay), which are very similar to the hardware intensity. We develop practical expressions for the sensitivity of sizing, supply voltage, and threshold voltage, relating the potential energy savings to the topology and energy profile of the circuit. The analysis of sizing, supply, and threshold sensitivities further reveals that the performance of joint optimizations can be predicted by knowing the energy reduction potential of each tuning variable and by applying the concept of sensitivity balancing. The energy-delay tradeoff information is then passed from the circuit level up to the block and micro-architectural levels.

3. Circuit Sensitivities

In order to find the circuit sensitivities, we first need to obtain equations that relate energy and delay to the transistor sizes, the supply voltage, and the threshold voltage. While there are many different models that one can use, we follow our prior work [19] and use the alpha-power law model of [5], as a baseline for the derivation of the gate delay formula:

$$t_d = \frac{K_d \cdot V_{dd}}{\left(V_{dd} - V_{on}\right)^{\alpha_d}} \cdot \left(\frac{W_{out}}{W_{in}} + \frac{W_{par}}{W_{in}} \right) \tag{1}$$

where W_{out}/W_{in} is the electrical fan-out of a gate, and W_{par}/W_{in} is a measure of its intrinsic delay [12]. While this model does have some fitting parameters (V_{on} is not exactly V_{th}, and α_d and K_d must be fit), it does fit the SPICE simulated data quite nicely.

We consider two components of energy: switching and leakage. The switching component is the standard dynamic energy term shown in Eq. (2),

$$E_{Sw} = \alpha_{0\to1} \cdot K_e \cdot \left(W_{out} + W_{par}\right) \cdot V_{dd}^2 \tag{2}$$

where $K_e W_{out}$ is the load capacitance, $K_e W_{par}$ is the self-loading of the gate, and $\alpha_{0\to1}$ is the probability of energy consuming transition at the output of the gate. Static gate leakage at $V_{gs} = 0$ is modeled as

$$E_{Lk} = \tau_{nom} \cdot d \cdot W_{in} \cdot I_0(S_{in}) \cdot e^{\frac{-(V_{th} - \gamma V_{dd})}{V_0}} \cdot V_{dd} \tag{3}$$

where $\tau_{nom} d$ is the delay of the logic block, $I_0(S_{in})$ is the normalized leakage current of the gate with inputs in state S_{in}; V_0 and γ represent sub-threshold slope and DIBL factor.

3.1. Sensitivity Overview

In energy-delay optimization, the objective is to utilize available timing slack for maximal energy reduction. There are usually several tuning variables that can be used to trade off energy and timing slack at various levels in design hierarchy. As pointed out by Zyuban and Strenski [18], the energy-efficient design is achieved when the marginal costs of all the tuning variables are balanced. Each of these variables carries a certain energy reduction potential per delay cost at each point of energy-delay space Eq. (4). This term (called hardware intensity in [18]) simply represents percent power per percent performance for an energy-efficient design.

$$\theta(X) = -\frac{D}{E} \cdot \left. \frac{\partial E / \partial x}{\partial D / \partial x} \right|_{x=X} \tag{4}$$

The true power minimization method always exploits the tuning variable with the largest capability for energy reduction. This ultimately leads to the point where the energy reduction potentials of all tuning variables are equalized. In order to further develop the understanding of these relative gradients, we will derive practical expressions (sensitivities), for different tuning variables. We consider gate size W, supply voltage V_{dd}, and change in threshold voltage ΔV_{th} as knobs in the optimization. By analyzing sensitivities, the efficiency of W, V_{dd}, and ΔV_{th} optimizations can be estimated from the energy profile of the logic block. Further, understanding the relationship between the logic block topology and the energy profile is necessary in order to identify the most efficient tuning variables without an exhaustive search.

3.2. Sensitivity to Gate Sizing

The sensitivity of energy to delay due to the sizing of stage i within a logic block is given by Eq. (5). There, ec_i represents the switching energy introduced by stage i, $p_{Lk,i}$ is

the leakage power of stage i and P_{Lk} is the total leakage power. Parameter $h_{eff,i}$ is the effective fanout of stage i [12].

$$\frac{\partial E_{Sw}/\partial W_i}{\partial D/\partial W_i} = -\frac{ec_i}{\tau_{nom}\cdot\left(h_{eff,i}-h_{eff,i-1}\right)} \tag{5a}$$

$$\frac{\partial E_{Lk}/\partial W_i}{\partial D/\partial W_i} = P_{Lk} - \frac{d\cdot p_{Lk,i}}{h_{eff,i}-h_{eff,i-1}} \tag{5b}$$

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, since the h_{eff} terms will be equal. This extends the variable taper result for an inverter chain [8], to more complex logic gates and topologies. Equation (5b) also suggests that at certain delay, leakage energy will start increasing with further size reduction.

3.3. Sensitivity to Supply Voltage

The sensitivity of total energy to delay, due to global supply reduction, is given by Eq. (6). Again, the design sized for the minimum delay at a nominal supply offers the greatest potential for energy reduction. This potential diminishes with the reduction in supply voltage. Supply reduction has a two-fold impact on the leakage energy: the leakage energy increases because of increase in delay, while on the other hand, it decreases because of the supply reduction and the reduced DIBL effect. The resulting tendency is the decrease in the leakage energy with supply reduction, which results in negative sensitivity to delay, Eq. (6b).

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

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

Parameter $x_v = (V_{on}+\Delta V_{th})/V_{dd}$; parameters V_{on} and α_d capture the DIBL effect on delay but are fixed across the range of supply voltages of interest. The same formula can be applied to dual supply voltage optimization. In that case, E and D would represent the total energy and delay of stages under low supply voltage.

3.4. Sensitivity to Threshold Voltage

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

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

Since the leakage power is exponential in ΔV_{th}, threshold voltage optimization has a limited range. For designs with very low leakage, lowering the threshold voltage is very attractive since it decreases delay with a very small energy cost.

4. Circuit Optimization – Examples

The sensitivities discussed in the previous section are derived for individual gates. What is more interesting is the sensitivity for whole circuit blocks. To evaluate blocks, this section looks at a few circuits to relate significant topological properties of logic paths—single path, off-path load, reconvergence—to the effectiveness of each of the optimization variables. Circuit block examples include a simple inverter chain, a memory decoder, and a tree adder.

In all of the examples, the nominal circuit is optimized for minimum delay d_{min} at nominal supply voltage V_{dd}^{nom} and nominal threshold V_{th}^{nom} ($\Delta V_{th} = 0$), as a reference. Starting from the nominal circuit, delay increment d_{inc} is specified and energy is then minimized under the delay constraint $d = d_{min}(1+d_{inc})$. The delay-constrained energy minimization represents a geometric program, which can be formulated in a convex form [4]. Optimization parameters are gate sizing W, supply voltage V_{dd}, change in threshold voltage ΔV_{th}, and optional buffer insertion. Energy-constrained delay minimization is a dual problem to delay-constrained energy minimization.

4.1. Inverter Chain – Single Path

4.1.1. Sizing Optimization – Mechanisms

The use of gate sizing to minimize the energy of a fixed length inverter chain is shown in Fig. 1a. Initially, when the circuit is sized for minimum delay, all stages have the same delay. Due to the geometric progression in size, most of the energy is dissipated in the last few stages, with the largest energy stored in the final load. Starting from the minimal delay point where all of the sensitivities are infinite, we change the gate sizes along the chain so that all of the sensitivities decrease equally. This, in turn, leads to an increase in effective fanout toward the output where most of the energy is consumed, as shown in Eq. (5). Therefore, the biggest energy savings for a fixed delay increase are achieved by downsizing the largest gates in the chain first. The optimal size of stage i is derived in Eq. (8). The expression is similar to that in [8] and directly follows from Eq. (5a).

$$W_i = \frac{W_i^{nom}}{\sqrt{1+\frac{ec_{i-1}}{S_W\cdot\tau_{nom}}}} \tag{8}$$

In the above formula, W_i^{nom} is the size of stage i that results in the minimum delay of the chain [3]. In an energy-efficient design, sizing sensitivity of all stages S_W is equal and also a function of the delay constraint.

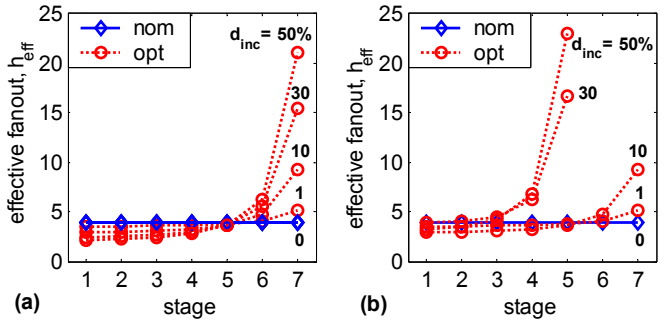

Fig. 1. Sizing: a) fixed, b) variable number of stages

If the number of stages can be varied, the delay constraint may be met with a fewer number of stages leading to a greater energy reduction. Intuitively, as the final stage is downsized to gain the biggest energy savings for a given delay increase, the size and number of the remaining stages adjust to meet the delay constraint, Fig. 1b [19]. It is important to realize, due to geometric progression in size in an inverter chain, that most of the energy is consumed in driving the fixed final load, and the maximum energy saving from sizing is limited to about 30%.

4.1.2. Supply Optimization – Mechanisms

Unlike sizing, scaling the supply directly affects the energy needed to charge the final load capacitance, and therefore can have a larger effect on the total energy. For illustration, we show supply optimization on a per-stage basis in the inverter chain. Our assumption here is that the supply voltage can only decrease from the input toward the output to avoid level conversion inside the block. In the nominal case, in which the delay of each stage is equal, the supply sensitivity of each stage depends only on the energy of that stage, as indicated in Eq. (6a). As in sizing, supply voltage optimization adds incremental delays, first to the stages with the highest energy consumption (stages toward the end of the chain), while increasing the effective fanout of these stages by lowering their supply voltage. Figure 2 shows the optimized per-stage supply and the resulting effective fanout.

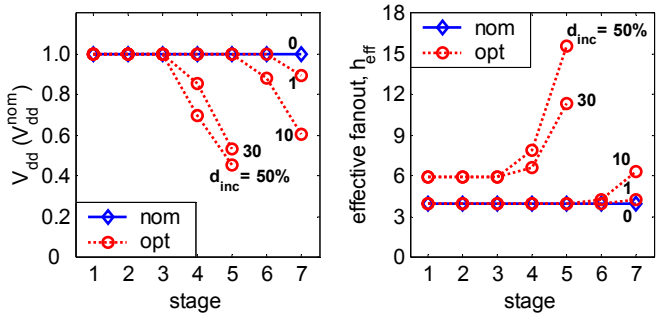

Fig. 2. Per-stage supply, variable number of stages

Compared to sizing, the supply optimization requires less change in the effective fanout for the same energy reduction. In practical designs, the effective fanout of the gate is bounded by the signal slope constraints to around 10-15.

4.2. Memory Decoder – Off-path Load

A buffer chain has a particularly simple energy distribution, one which increases geometrically until the final stage. This type of profile drives the optmization over sizing and supply to focus on the final stages first. Most practical circuits have a more complex energy profile due to off-path loads and varying activities per logic stage, for example, a SRAM decoder.

The decoder shares some characteristics with a simple inverter chain; the total capacitance at each stage grows geometrically, but the number of active paths decreases geometrically, as well. As a result of this, the peak of the energy distribution is often in the middle of the structure. For example, the 256 wordline SRAM decoder shown in Fig. 3 has the energy peak at the output of the predecoder because of the path properties shown in Table 1.

Table 1. Activity map of the 8→256 wordline SRAM decoder

SRAM decoder gates	predriver	predecoder		word driver
	Inv	Nand-inv	Nand-inv	Nand-inv-buff
Active	16	4	2	1
Total	16	16	32	256

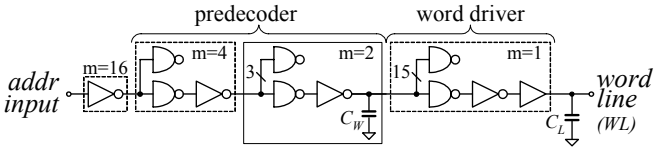

Fig. 3. Critical path, 256 wordline SRAM decoder

Figure 3 shows the critical path of this SRAM decoder. The multiplication factor m denotes the number of active gates at each stage. Branching occurs at the input of each NAND gate and the number of active gates per stage decreases in a geometric fashion to select only one wordline at the output.

4.2.1. Sizing and Buffer Insertion

Sizing optimization effectively reduces the internal energy peaks through direct gate sizing or buffer insertion, as shown in Fig. 4. The initial sizing for minimum delay does not require an extra buffer at the output of the decoder, thus the total number of stages is seven. Inserting a buffer stage at the output reduces the effective load presented by the 256 decoder/word driver cells. Alternatively, optimization by direct gate sizing minimizes the size of the word driver input and produces the same effect, as shown in Fig. 4b. This essentially divides the sizing problem into two sub-problems: a) sizing of predecoder logic to drive the minimum word driver input, and b) sizing of word driver to drive the wordline. This is readily seen from the per-stage sensitivity expression with branching:

$$\frac{\partial E_{Sw} / \partial W_i}{\partial D / \partial W_i} = -\frac{b_{i-1} \cdot ec_i}{\tau_{nom} \cdot (h_{eff,i} - h_{eff,i-1})} \qquad (9)$$

in which b_{i-1} is the branching factor of the stage i-1. Downsizing the gates driven by the stage with the highest branching factor yields the biggest energy savings for the given delay cost. In the decoder example this situation occurs at the output of the predecoder, as shown in Fig. 4a.

While the peak of switching energy is inside the block, the peak of leakage energy occurs at the output, due to the

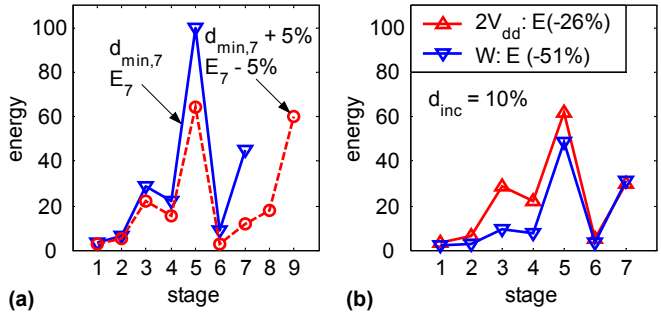

Fig. 4. Energy profile in SRAM decoder: a) d_{min} design, b) d_{inc}=10% (WL=128, input activity is 15%)

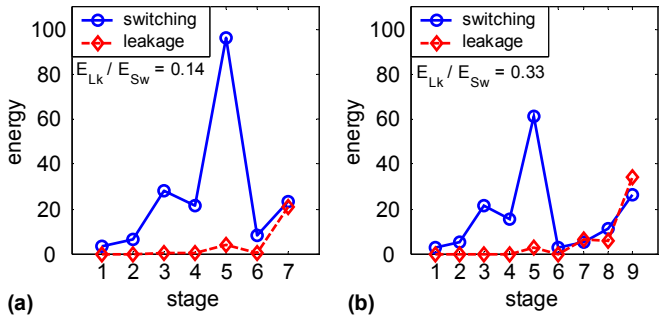

Fig. 5. Energy profile in SRAM decoder with a) 7 stages, b) 9 stages ($d=d_{min}$, WL=128, input activity is 15%)

activity profile of the decoder. This is illustrated in Fig. 5, where the energy profile in a min-delay sized decoder is shown for cases with seven and nine stages. Output gates are the largest and the majority of the gates are inactive, resulting in the largest leakage at the output. Although inserting a buffer stage reduces the size of the predecoder, the leakage energy of the word driver increases relative to the switching energy as shown in Fig. 5b, because only one output buffer is active at a time.

4.2.2. Supply Optimization

The supply optimization is less effective in designs where the peak of energy consumption occurs inside the block. Because of the assumption that the supply voltage can only decrease from input to output, in order for the supply to affect the energy peak, the delay of all stages after the peak needs to increase, thus reducing the marginal return, as shown in Fig. 4b. Sizing optimization is more effective than discrete supply optimization because sizing can selectively reduce dominant energy peaks inside the block by paying the price of increased delay, in stages only right after the energy peak, therefore increasing the marginal return. Contrarily, the supply starts from the output of the block and works backwards.

4.3. Adder – Off-path Load and Reconvergence

More complex designs may have reconvergent fanouts and multiple active outputs qualified by paths with various logic depth. As an example, we analyze a 64-bit Kogge-Stone tree

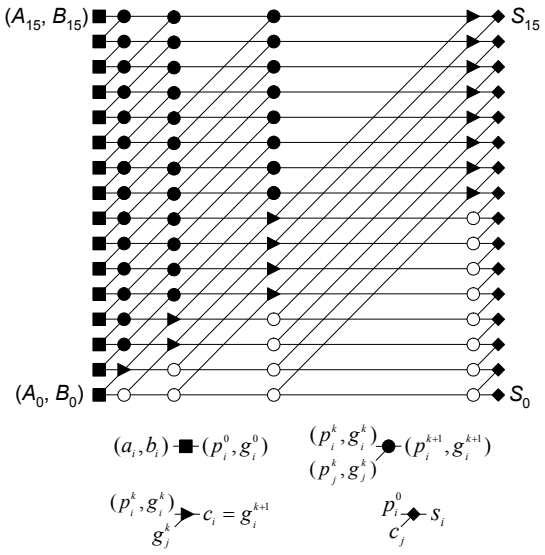

Fig. 6. Kogge-Stone tree adder

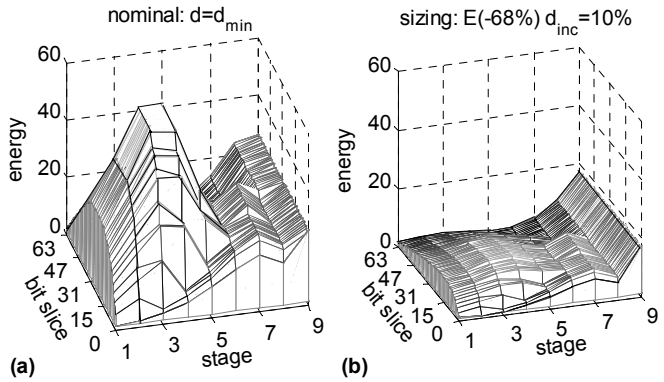

Fig. 7. Energy map in 64-b adder a) d_{min} design, b) d_{inc}=10%, gate sizing (WL=32, input activity is 15%)

adder [1]. The structure of this adder is shown on 16-b example in Fig. 6. There are many paths through an adder, and unlike the decoder, not all of these paths are balanced. To be fair, the initial sizing makes all the paths in the adder equal to the critical path. As a result, further reductions in size would cause the delay of the adder to increase. Since the paths through an adder roughly correspond to different bit slices, we allocate each gate in the adder to a bit slice. This partition works well for tree adders, and Fig. 7 shows the resulting energy map for the minimum delay, as well as the situation when a 10% delay increase is allowed [19]. Like the decoder, the dominant energy peaks are internal, which makes transistor sizing more effective than V_{dd} scaling. The data indicates that a 68% decrease in energy is possible using transistor sizing, while only 32% is saved by using two supplies. Reducing the supply over the whole block yields only 17% of energy reduction.

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. Therefore, the peak in Fig. 7a occurs close to the input of the adder, where the activity of propagate logic is still comparable to that of generate logic. Like in the decoder, the adder energy maps show that sizing optimization is very effective when energy peaks occur inside the block.

5. Joint Optimizations

To achieve the most energy-efficient design, the energy reduction potentials of all the tuning variables must be balanced, otherwise one would tune the variable with low energy cost rather than the variable with high energy cost. Optimization of V_{dd} and V_{th} has been explored by many researchers. Here, it is reviewed first, before looking at the more general problem.

5.1. Optimal V_{dd} and V_{th}

The nominal supply V_{dd}^{nom} and threshold V_{th}^{nom} ($\Delta V_{th} = 0$), given by the technology, are rarely optimal for all applications from the energy-throughput standpoint. To illustrate this, one should assume that required throughput is achieved with the nominal transistor at the nominal supply voltage. The same throughput can be achieved with less energy by adjusting V_{dd} and V_{th} [7], [9], [14]. In the nominal case, the leakage energy is not significant, so V_{th} is lowered. This, in turn, creates timing slack that enables the reduction of

V_{dd}, which targets the dominant component of energy consumption—switching energy.

The results indicate that the values of optimum V_{dd} and V_{th} depend on nominal process parameters, required throughput, block function, and topology. Expressions for optimal V_{dd} and V_{th} were derived by Nose and Sakurai in [14], leading to optimal ratio of leakage and switching energy. Simplifying their expression by using the linear current model ($\alpha_d = 1$), we obtain the dependence of optimal leakage-to-switching ratio in terms of process and architectural parameters,

$$\left.\frac{E_{Lk}}{E_{Sw}}\right|_{Opt} = \frac{2}{\ln\left(\frac{L_d}{\alpha_{avg}}\right) - K_{tech}} \tag{10a}$$

$$K_{tech} = \ln\left(\frac{2C \cdot V_0}{d_{FO4}^{nom} \cdot I_0} \cdot \frac{V_{dd}^{nom}}{V_{th}^{nom} - \gamma V_{dd}^{nom}}\right) \tag{10b}$$

where α_{avg} is the average block switching activity, L_d is logic depth of the critical path, d_{FO4}^{nom} is the delay of a FO4 inverter at the nominal supply V_{dd}^{nom} and the nominal threshold V_{th}^{nom}, and C is capacitance per micron of gate width.

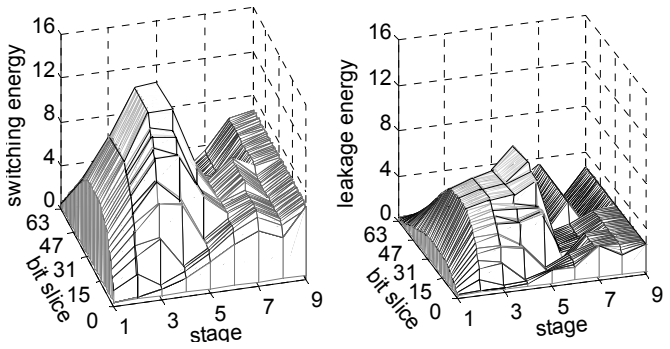

Fig. 8. Energy map in adder, d=d_{min}(ΔV_{th}=0), E_{Lk}/E_{Sw}=0.45, V_{dd}^{opt}=0.78V_{dd}^{nom}, ΔV_{th}^{opt}=-190mV (WL=32, input activity is 15%)

From the architectural standpoint, logic depth can vary by less than an order of magnitude, but average activity can change by several orders of magnitude, depending on the function. As a consequence, in designs with similar activity, optimal leakage-to-switching ratio will be almost constant due to the logarithmic dependence [9]. In our example, the adder circuit has a much higher average activity than the decoder. Due to the high switching activity in the adder, leakage energy is small relative to the switching component for the nominal case, so the optimization lowers the threshold to create timing slack and then scale down V_{dd} to reduce switching energy, Fig. 8. This results in an optimal leakage to switching ratio of 45% in the adder, compared to 33% in the decoder, matching the predictions from [14] of about a 40% ratio. Unfortunately, the range of threshold adjustment through substrate bias is small, which often prevents one from achieving optimal performance.

5.2. Other Combinations and Examples

The joint optimization has the additional degree of freedom to choose a more efficient direction at each point toward the optimal solution. In general, it is very difficult to predict the contribution of each tuning variable in a multi-variable optimization. In cases where sensitivity of one tuning

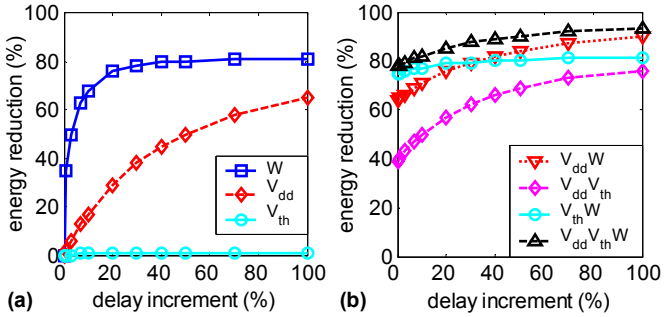

Fig. 9. Energy reduction techniques in 64-b tree adder, (WL=32, input activity is 15%, V_{dd} and V_{th} are adjusted per block, W is continuous)

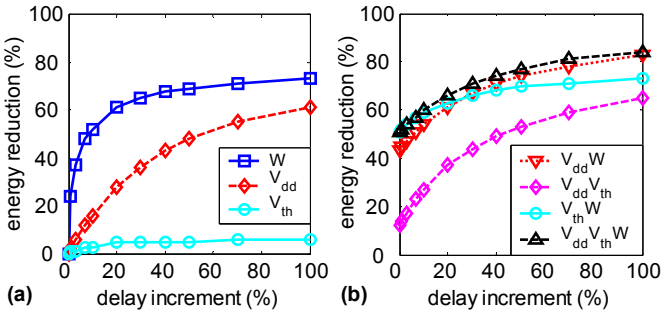

Fig. 10. Energy reduction techniques in SRAM decoder, (WL=128, input activity is 15%, V_{dd} and V_{th} are adjusted per block, W is continuous)

variable is significantly larger than those of other variables, the optimization trajectory can be approximated by the trajectory along that variable. When sizing, supply and threshold voltage are used as optimization variables, sizing has the largest sensitivity for small delay increments from the minimum delay design. Threshold sensitivity diminishes quicker than that of supply or sizing, leaving the supply sensitivity dominant one in the region of large delay increments. This is shown for adder and decoder examples, in Figs. 9a and 10a, respectively. Such properties allow for the analysis of joint optimizations based on the behavior of single-variable optimizations. Furthermore, the performance of single-variable optimizations can be predicted from the topological properties and the energy profile of a block without actually performing the optimization.

We begin the joint optimization analysis with a sizing and supply example. Since the energy reduction potential of sizing and supply is not equal at the min-delay point, it is possible to save energy without a delay penalty by simply balancing sensitivities. Raising the supply voltage and changing the sizes reduces the power. Due to the large gap between sizing and supply sensitivity, the increase in supply voltage results in the increase in energy, but creates slack,

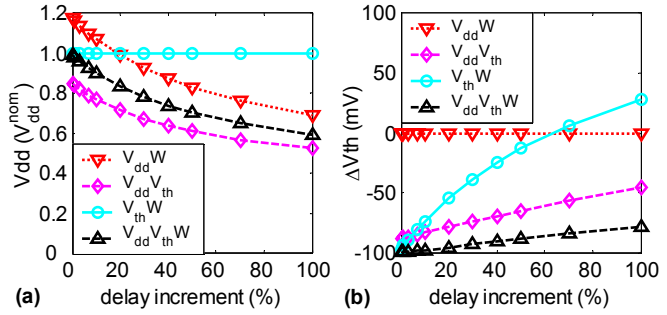

Fig. 11. Values of a) V_{dd}, and b) ΔV_{th} for cases in Fig. 10

which is then exploited with more efficient sizing optimization for overall energy reduction. The bigger the initial gap between the two sensitivities, the greater the energy reduction that can be obtained. For example, at minimum delay, the gap between the sizing (W) and the threshold (V_{th}) is the largest, followed by that between the sizing and the supply (V_{dd}), while the gap between the supply and the threshold is the smallest. This results in the largest initial energy reduction when the sizing is combined with threshold ($V_{th}W$), and the smallest reduction when the supply is combined with threshold ($V_{dd}V_{th}$).

By analyzing Fig. 11, we can help better understand the role of the supply and threshold variables in joint optimizations. Since the supply has a smaller sensitivity than the sizing, the supply is increased to a create timing slack which can be more efficiently utilized by sizing, the $V_{dd}W$ case in Fig. 11a. On the other hand, since the supply has a larger sensitivity than the threshold, its decrease is more energy efficient while the threshold reduction serves to preserve timing, as shown in $V_{dd}V_{th}$ case in Fig. 11. In addition to the initial gap between the sensitivities, which determines the energy reduction at the starting point, the resulting balanced sensitivity value determines the potential for energy reduction as the delay is further increased from the starting point.

Energy savings of about 60% in the adder and 40% in the decoder are possible without any delay penalty by simply choosing appropriate values of supply, threshold, and circuit size, as shown in Figs. 9b and 10b. However, individual circuit examples may be misleading. The marginal costs of the overall system are really what matters, and that is the reason why sensitivities are important. For example, if adder, or some other functional-unit energy is a much smaller percentage of the total processor energy than that of latches/clocking, than it might actually pay off to lower the power of the latches (make the latches slower) and increase the power of the adder (make the adder faster).

6. Micro-Architectural Optimization – Examples

In order to give an example of the system-level optimization, this section will revisit the example from Chandrakasan et al. [6] to compare a pipelined system design to a parallel system design for minimizing power. A schematic of the circuit is shown in Fig. 12. The reference design is an Add-Compare unit which uses the adder described in Sec. 4.3, for both the adder (block A) and comparator (block B). The reference design is optimized for minimum delay under V_{dd}^{nom} and V_{th}^{nom}. Using the throughput

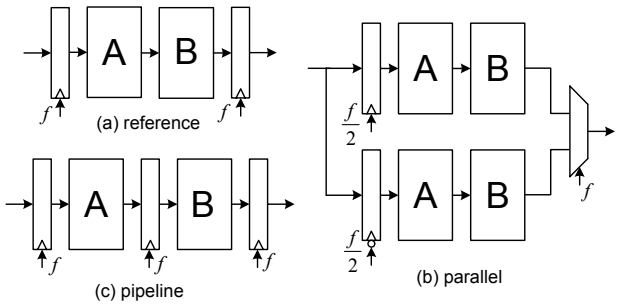

Fig. 12. Micro-architecture example: a) reference design, b) parallel design, c) pipeline design

of this design as a constraint and information about energy reduction tradeoffs of the adder and comparator blocks from the inner loop, we can estimate the energy needed for the reference design and its parallel or pipelined implementation.

In each of these three designs, the goal is to find the optimal value of the supply and the threshold voltage that result in a minimum energy for the given throughput constraint. This value is found by optimization, in which V_{th} is swept from 0 to -200mV in steps of 5mV. Each time V_{th} is modified, V_{dd} in all three designs is adjusted to achieve the target throughput with minimal energy, using the multi-variable sensitivity information from the lower level blocks. The goals of this sweep are to find the optimal (V_{dd}, V_{th}) point for each implementation and to illustrate the trend around the optimal point, as shown in Fig. 13. For each design, optimal (V_{dd}, V_{th}) point is reached when the voltage and threshold sensitivities of all the underlying blocks are balanced. As seen in Fig. 13, although the optimal (V_{dd}, V_{th}) points are different for each implementation, they all roughly correspond to the same value of leakage-to-switching energy ratio. This is in line with Eq. (10), since the logic depth and activity in these implementations do not vary significantly. In this example, the optimal ratio of leakage-to-switching energy is around 40% for all the implementations, which roughly corresponds to that of its main sub-block—the adder. In fact, all the curves are very flat around their optimal point in a range from 20% to 100% of leakage-to-switching energy ratio.

Energy-per-operation in all three designs is compared to the reference case which operates at V_{dd}^{nom} and V_{th}^{nom}. The switching energy-per-operation decreases approximately by the same factor from voltage scaling in both the parallel and the pipeline designs. The leakage energy increases from increasing the area of the design, and decreases from scaling down the supply voltage. Therefore, the leakage energy of the parallel design is larger than that of the pipelined design because of the larger area. 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 [6]. 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 that for a pipeline design. For this reason, a parallel implementation achieves smaller energy savings. Equivalently, the introduction of parallelism decreases the amount of time that a device spends on computations, thereby decreasing the ratio of useful (switching) to wasted (leakage) energy.

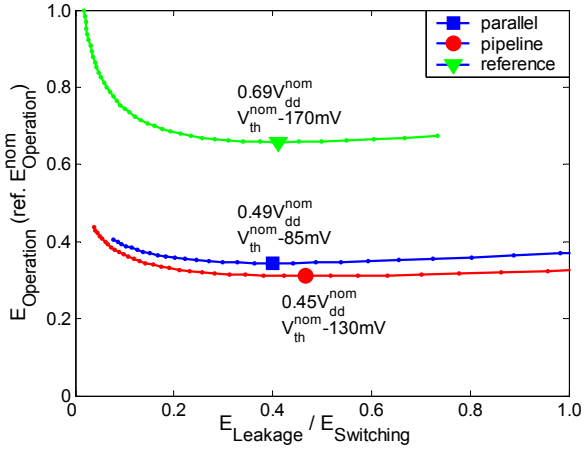

Fig. 13. Parallel, pipeline, and reference designs: energy-per-operation vs. leakage-to-switching energy ratio

Optimizations of sizing, supplies, and thresholds have limited scope due to the physical or functional constraints of these tuning variables. Each topology has a range within its energy-delay space where the energy and delay can be traded for each other. At either extreme, the marginal cost of decreasing the energy/delay becomes too large. For a specific topology, about a two-time increase in the delay exploits almost all the available energy savings, so the marginal energy saving for an additional delay is very low. Architectural changes like parallelism and pipelining implicitly increase the delay of the underlying block about two-times, leaving little space for additional optimizations. This is illustrated in Fig. 14, in which the energy of the design (reference, parallel and pipelined) is shown as a function of the delay increment from the nominal value, for (a) the supply optimization, and (b) the supply and threshold optimization.

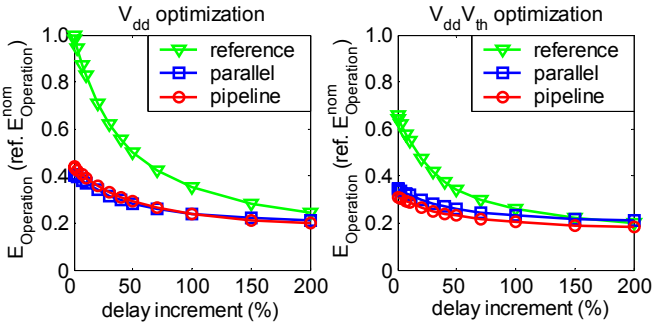

Fig. 14. Energy vs. delay increment for reference, parallel and pipeline design under a) the supply optimization, b) the supply and threshold optimization

The throughput requirement set by the application determines the choice of the most efficient circuit topology (for example, the type of adder). Given that the scope of optimization for each topology is limited to about a two-time increase in its delay (from the minimum set by technology), it is desirable to choose the circuit topology whose minimum achievable delay is positioned relatively close to the throughput requirement determined by the application. Once the topology is chosen, the optimization of the sizing, supply, and threshold can be efficiently exploited.

7. Conclusions

Creating energy efficient circuits is becoming an increasingly important priority. In order to truly minimize the power in a chip, it requires that the different layers of the design all work toward achieving the same balance in trading energy for performance. We examined the lowest level optimization issues in this paper, exploring how optimizing the gate size, the supply voltage, and the threshold voltage affect circuit performance. In topologies with a monotonic increase in energy towards the output (such as an inverter chain), supply reduction achieves the largest energy savings, with sizing being much less effective. If, however, an off-path load and a reconvergent fanout are present, sizing optimization will be the most effective since the peak of energy consumption is internal to the block.

In looking at a design optimized for speed, the nominal clock cycle should be set about 10% higher than the theoretical minimum, due to the large energy benefit offered by a small delay penalty; but the returns from sizing quickly fall off, and above 20% the return is very small. In contrast, a global supply reduction is the least effective energy reduction technique for small delay increments, but it is quite useful when the delay increment is sizeable. It is found that for the circuits analyzed at a delay increment of 20%, at least a 30% energy savings can be achieved by sizing, and a 30%-60% by supply optimization. A combination of sizing and supply or threshold voltage can provide a 40-70% savings. Proper balancing of the tuning variables provides an energy savings of about 50%, with no delay penalty. Future work is needed to see if similar tradeoffs exist at the block and micro-architecture levels.

The designs that are truly power-optimized will have higher leakage current than what is common today. By increasing the leakage energy, pipelining begins to have advantages over parallel solutions, and has already begun to affect how high-performance chips are designed.

Acknowledgments

This research is supported in part by MARCO contracts: CMU 2001-CT-888, GSRC 98-DT-660, and Georgia Tech B-12-D00-S5.

References

[1] P.M. Kogge and H.S. Stone, "A Parallel Algorithm for the Efficient Solution of General Class of Recurrence Equations," *IEEE Trans. Computers*, vol. C-22, no. 8, pp. 786-793, Aug 1973.

[2] H.C. Lin and L.W. Linholm, "An Optimized Output Stage for MOS Integrated Circuits," *IEEE JSSC*, vol. SC-10, no. 2, pp. 106-109, Apr. 1975.

[3] C. Mead and L. Conway, "Introduction to VLSI Design," Reading, MA: *Addison-Wesley*, 1980.

[4] J.P. Fishburn and A.E. Dunlop, "TILOS: a posynomial programming approach to transistor sizing," *in Proc. ICCAD*, Nov. 1985, pp. 326-328.

[5] T. Sakurai and R. Newton, "Alpha-Power Law MOSFET Model and its Applications to CMOS Inverter Delay and Other Formulas," *IEEE JSSC*, vol. 25, no. 2, pp. 584-594, Apr. 1990.

[6] A.P. Chandrakasan and R.W. Brodersen, "Low-power CMOS digital design," *IEEE JSSC*, pp. 473-484, vol. 27, no. 4, Apr. 1992.

[7] D. Liu and C. Svensson, "Trading Speed for Low Power by Choice of Supply and Threshold Voltage," *IEEE JSSC*, vol. 28, no. 1, pp. 10-17, Jan. 1993.

[8] S. Ma and P. Franzon, "Energy Control and Accurate Delay Estimation in the Design of CMOS Buffers," *IEEE JSSC*, vol. 29, no. 9, pp. 1150-1153, Sept. 1994.

[9] R. Gonzalez, B. Gordon, and M.A. Horowitz, "Supply and Threshold Voltage Scaling for Low Power CMOS," *IEEE JSSC*, vol. 32, no. 8, pp. 1210-1216, Aug. 1997.

[10] T. Kuroda et al., "Variable Supply-Voltage Scheme for Low-Power High-Speed CMOS Digital Design," *IEEE JSSC*, pp. 454-462, vol. 33, no. 3, Mar. 1998.

[11] A.R. Conn et al., "Gradient-Based Optimization of Custom Circuits Using a Static-Timing Formulation," *in Proc. DAC*, June 1999, pp. 452-459.

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

[13] T.Burd et al., "Dynamic Voltage Scaled Microprocessor System," *in Proc. ISSCC*, Feb. 2000, pp. 294-295.

[14] K. Nose and T. Sakurai, "Optimization of V_{DD} and V_{TH} for Low-Power and High-Speed Applications," *in Proc. ASP-DAC*, Jan. 2000, pp. 469-474.

[15] M Hamada, Y. Ootaguro, and T. Kuroda, "Utilizing Surplus Timing for Power Reduction," *in Proc. CICC*, May 2001, pp. 89-92.

[16] S. Sirichotiyakul et al., "Duet: An Accurate Leakage Estimation and Optimization Tool for Dual-*Vt* Circuits," *IEEE TVLSI*, vol. 10, no. 2, pp. 79-90, Apr. 2002.

[17] J. Tschanz et al., "Design optimizations of a high performance microprocessor using combinations of dual-V_T allocation and transistor sizing," *in Proc. Symp. VLSI*, June 2002, pp. 218-219.

[18] V. Zyban and P. Strenski, "Unified Methodology for Resolving Power-Performance Tradeoffs at the Microarchitectural and Circuit Levels," *in Proc. ISLPED*, Aug. 2002, pp. 166-171.

[19] V. Stojanovic, D. Markovic, B. Nikolic, M. Horowitz, R. Brodersen, "Energy-Delay Tradeoffs in Combinational Logic using Gate Sizing and Supply Voltage Optimization," to appear *in Proc. ESSCIRC*, Sept. 2002.