

The Limit of Dynamic Voltage Scaling and Insomniac Dynamic Voltage Scaling

Bo Zhai, *Student Member, IEEE*, David Blaauw, *Member, IEEE*, Dennis Sylvester, *Senior Member, IEEE*, and Krisztian Flautner, *Member, IEEE*

Abstract—Dynamic voltage scaling (DVS) is a popular approach for energy reduction of integrated circuits. Current processors that use DVS typically have an operating voltage range from full to half of the maximum V_{dd} . However, there is no fundamental reason why designs cannot operate over a much larger voltage range: from full V_{dd} to subthreshold voltages. This possibility raises the question of whether a larger voltage range improves the energy efficiency of DVS. First, from a theoretical point of view, we show that, for subthreshold supply voltages, leakage energy becomes dominant, making “just-in-time computation” energy-inefficient at extremely low voltages. Hence, we introduce the existence of a so-called “energy-optimal voltage” which is the voltage at which the application is executed with the highest possible energy efficiency and below which voltage scaling reduces energy efficiency. We derive an analytical model for the energy-optimal voltage and study its trends with technology scaling and different application loads. Second, we compare several different low-power approaches including MTCMOS, standard DVS, and the proposed Insomniac (extended DVS into subthreshold operation). A study of real applications on commercial processors shows that Insomniac provides the best energy efficiency. From these results, we conclude that extending the voltage range below $V_{dd}/2$ will improve the energy efficiency for many processor designs.

Index Terms—Dynamic voltage scaling, energy efficiency, insomniac, low power, subthreshold design.

I. INTRODUCTION

DUE TO technology scaling, microprocessor performance has increased tremendously, albeit at the cost of higher power consumption. Energy-efficient operation has therefore become a very pressing issue, particularly in mobile applications which are battery operated. Dynamic voltage scaling (DVS) was proposed as an effective approach to reduce energy use and is now used in a number of commercial low-power processor designs [1]–[3].

Most applications do not always require the peak performance from the processor. Hence, in a system with a fixed performance level, certain tasks complete ahead of their deadline and the processor enters a low-leakage sleep mode [4] for the remainder of the time. This operation is illustrated in Fig. 1(a). In DVS systems, however, the performance level is instead reduced during periods of low utilization such that the

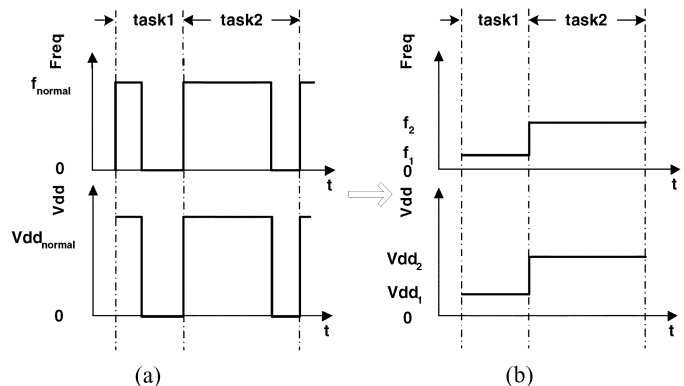


Fig. 1. Illustration of optimal task scheduling.

processor finishes each task “just in time,” stretching each task to its deadline, as shown in Fig. 1(b). As the processor frequency is reduced, the supply voltage can also be reduced. As shown by the equations below,¹ the reduction in frequency [5] combined with a quadratic reduction from the supply voltage results in an approximately cubic reduction of power consumption. However, with reduced frequency, the time to complete a task increases, leading to an overall quadratic reduction in the energy required to complete a task

$$\text{Delay} = \frac{1}{f} = \frac{C_s V_{dd}}{I_{dsat}} \propto \frac{V_{dd}}{(V_{dd} - V_{th})^{1.3}}$$

$$\text{Power} \propto f V_{dd}^2 \quad \text{Energy} \propto V_{dd}^2.$$

DVS is therefore an effective method to reduce the energy consumption of a processor, especially under wide variations in workload that are increasingly common in mobile applications. Hence, extensive work has been performed on how to determine voltage schedules that maximize the energy savings obtained from DVS [4], [9].

In most current DVS processor designs, the voltage range is limited from full V_{dd} to approximately half V_{dd} at most. In Table I, the available range of operating voltages and associated performance levels are shown for three commercial designs. The lower limit of voltage scaling is typically dictated by noise-sensitive circuits, such as pass-gate structures, phase-locked loops (PLLs), sense amps, and results from applying DVS to a processor “as is” with minimum redesign. However, significant design effort is required to accommodate operation over a wide range of voltage levels [6]. It is well known that CMOS circuits

Manuscript received November 12, 2004; revised June 7, 2005. This work was supported in part by MARCO/DARPA GigaScale Systems Research Center, ARM, and the National Science Foundation.

B. Zhai, D. Blaauw, and D. Sylvester are with the University of Michigan, Ann Arbor, MI 48109 USA (e-mail: bzhai@umich.edu; blaauw@umich.edu; dmcs@umich.edu).

K. Flautner is with ARM Ltd., Cambridge CB1 9NJ, U.K. (e-mail: krisztian.flautner@arm.com).

Digital Object Identifier 10.1109/TVLSI.2005.859588

¹The 1.3- μm power [5] scaling of current is only valid for high supply voltages when carrier velocity saturates. Subthreshold scaling of the supply voltage with performance for low-voltage operation will be extensively discussed in Section III.

TABLE I
COMMERCIAL PROCESSOR DESIGNS AND RANGE OF VOLTAGE SCALING

	Voltage Range	Frequency Range
IBM PowerPC 405LP [3]	1.0V-1.8V	153MHz-333MHz
TransMeta Crusoe TM5800 [1]	0.8V-1.3V	300MHz-1GHz
Intel XScale 80200 [2]	0.95V-1.55V	333MHz-733MHz

can operate from nominal voltage down to less than 200 mV. In such “subthreshold” operating regimes, the supply voltage lies below the device threshold voltage and the circuit operates using leakage currents. Work has been reported on designs that operate at subthreshold voltages [7], [8], and it was reported that the minimum allowable supply voltage of an ideal functional CMOS inverter is 36 mV [10], [11]. A number of commercial products have also used subthreshold operation for extremely low power applications [12].

With some additional design effort, it is conceivable to significantly extend the operating voltage range of processors. Hence, one issue that needs to be addressed is the determination of a lower limit of the voltage range for optimal energy efficiency. The optimal voltage limit depends on two factors: the power/delay tradeoffs at low operating voltages and the workload characteristics of the specific processor. In this paper, we address both of these issues.

First, we show that the quadratic relationship between energy and V_{dd} deviates as V_{dd} is scaled down into the subthreshold region of MOSFETs. In subthreshold operation, the “on-current” takes the form of subthreshold current, which is exponential with V_{dd} , causing the delay to increase exponentially with voltage scaling. Since leakage energy is linear with the circuit delay, the fraction of leakage energy *increases* with supply voltage reduction in the subthreshold regime. Although dynamic energy reduces quadratically, at very low voltages where dynamic and leakage energy become comparable, the total energy can increase with voltage scaling due to the increased circuit delay. In this paper, we derive an analytical model for the voltage that minimizes energy, and we show that this so-called energy optimal voltage lies well above the previously reported [10] minimal functional operating voltage of 36 mV. We verify our model using SPICE and study its trends as a function of different design and process parameters. As one of the results, our work shows that operation at voltages well below the threshold voltage is never energy-efficient.

A second issue that determines whether to apply such aggressive, or extended, DVS (which we refer to as *Insomniac*) is based on the workload characteristics of the processor. Clearly, it is not necessary to extend the voltage range below that which is needed based on the expected workload of the processor. We show this workload impact with real applications on two commercial processors. Further, we analyze the energy efficiency of different low-power schemes, including multiple-threshold CMOS (MTCMOS), standard DVS, and *Insomniac* for a number of workload traces obtained from a processor running a wide range of applications. Our results show that most applications benefit significantly from *Insomniac* whereas only very

low-activity applications with long idle times benefit greatly from pure MTCMOS.

The remainder of this paper is organized as follows. Section II provides an overview of the voltage limit for functionally correct CMOS logic. Section III presents our analysis of the minimum voltage scaling limit for optimal energy efficiency. Section IV presents our energy modeling for different low-power approaches. Finally, Section V concludes the paper.

II. CIRCUIT BEHAVIOR AT ULTRALOW VOLTAGES

Before we derive the energy optimal operating voltage in Section III, we first briefly review the minimum operating voltage that is required for functional correctness of CMOS logic. The minimum operating voltage was first derived by Swanson and Meindl in [10] and [11] and is given as follows:

$$V_{ddlimit} = 2 \cdot \frac{kT}{q} \cdot \left[1 + \frac{C_{fs}}{C_{ox} + C_d} \right] \cdot \ln \left(2 + \frac{C_d}{C_{ox}} \right) \cong 2V_T \cdot \ln \left(2 + \frac{C_d}{C_{ox}} \right) \quad (1)$$

where C_{fs} is the fast surface state capacitance per unit area [10], C_{ox} is the gate-oxide capacitance per unit area, C_d is the channel depletion region capacitance per unit area, and V_T is the thermal voltage kT/q . For bulk CMOS technology, we know that subthreshold swing (the amount of gate voltage reduction for the drain current to drop by $10\times$) can be expressed as follows:

$$S_S = \ln 10 \cdot V_T \cdot \left(1 + \frac{C_d}{C_{ox}} \right). \quad (2)$$

From this, we can rewrite (1) as follows:

$$V_{ddlimit} = 2V_T \cdot \ln \left(1 + \frac{S_S}{\ln 10 \cdot V_T} \right) \cong 52 \text{ mV} \cdot \ln \left(1 + \frac{S_S}{60 \text{ mV}} \right) \text{ at } 300 \text{ K}. \quad (3)$$

For 0.18- μm bulk CMOS technologies, S_s is typically in the range of 90 mV/decade and therefore

$$V_{dd,limit} = 48 \text{ mV}. \quad (4)$$

Hence, it is theoretically possible to operate circuits deep into the subthreshold regime given that typical threshold voltages are much larger than 48 mV. In fact, SPICE simulation (with BSIM models) confirms that it is possible to construct an inverter chain that works properly at 48 mV, although at this point the internal signal swing is reduced to less than 30 mV. Note that BSIM3 models may be less accurate than other models, such as EKV models [17]. However, BSIM3 models are present industry standard and we will use it for SPICE simulation in the remainder of this paper. In Fig. 2, we also show that it is possible to operate a wide range of standard library gates at similar operating voltages and that their delay tracks are relatively consistently with that of the inverter. It is, however, clear that there are practical reasons why operating circuits at the minimum voltage is not desirable, such as susceptibility to noise, soft errors, and process variations [18]. More importantly, we show in the next section

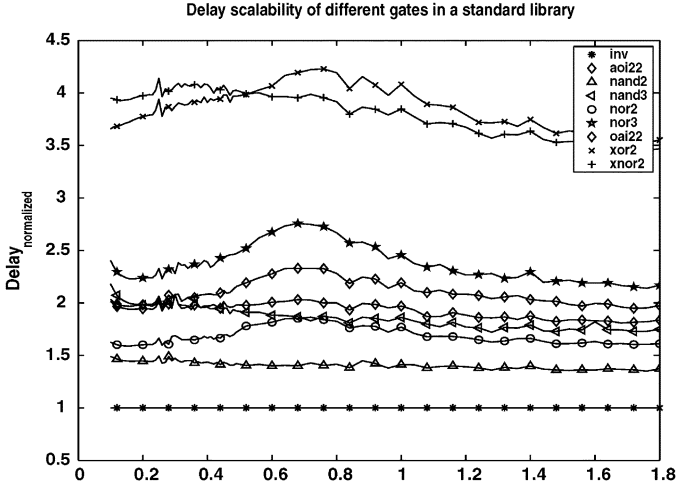


Fig. 2. Delay of typical library gates over a wide voltage range, normalized to inverter delay.

that, from an energy efficiency point of view, the minimum operating voltage for functionally correct operation does not provide the best results.

III. MINIMUM ENERGY ANALYSIS

A. Minimum Energy Point Modeling

We first illustrate the energy dependence on supply voltage using a simple inverter chain consisting of 50 inverters and then extend the analysis for more general circuits. A single transition is used as a stimulus and energy is measured over the time period necessary to propagate the transition through the chain. (Note that simultaneous switching noise is also an important issue in DVS energy efficiency, and we will not address it in this paper for simplicity.) The energy- V_{dd} relation is plotted in Fig. 3. The dynamic energy component E_{active} reduces quadratically while the leakage energy E_{leak} increases with voltage scaling. The reason for the increase in leakage energy in the subthreshold operating regime is that as the voltage is scaled below the threshold voltage, the on-current (and hence the circuit delay) decreases exponentially with voltage scaling, while the off-current is reduced less strongly. Hence, the leakage energy E_{leak} will rise and supersede the dynamic energy E_{active} at about 180 mV. This effect creates a minimum energy point (referred to as V_{min}) in the inverter circuit that lies at 200 mV, as shown in Fig. 3.

In the previous example, we are implicitly assuming that there is always one input transition per clock cycle. However, the switching activity varies in different circuits and therefore we include the input activity factor α , which is the average number of times the node makes a power consuming transition in one clock period. We now derive an analytical expression for the energy of an inverter chain as a function of the supply voltage. Suppose we have an n -stage inverter chain with activity factor of α . The standard expression for subthreshold current is given by²

$$I_{sub} = \mu_{eff} C_{ox} \frac{W}{L_{eff}} (m-1) V_T^2 e^{\frac{V_{gs}-V_{th}}{mV_T}} \left(1 - e^{-\frac{V_{ds}}{V_T}}\right) \quad (5)$$

²BSIM3. [Online]. Available: <http://www-device.eecs.berkeley.edu/~bsim3/get.html>

where V_{th} is the threshold voltage of the MOSFET, μ_{eff} is the effective mobility, W is the transistor width, L_{eff} is the effective channel length, V_{gs} and V_{ds} are the gate to source and drain to source voltages, respectively, and

$$m = \frac{S_s}{\ln 10 \cdot V_T} = \frac{90}{\ln 10 \times 26} = 1.51. \quad (6)$$

In (6), we again assume that S_s is 90 mV/decade, which is a typical value. We now express the total energy E per clock cycle as the sum of dynamic and leakage energy

$$\begin{aligned} E &= E_{active} + E_{leak} \\ &= \alpha \cdot n \cdot E_{switch,inv} + P_{leak} \cdot t_d \\ &= \alpha \cdot n \cdot \left(\frac{1}{2} \cdot C_s \cdot V_{dd}^2 \right) + (n \cdot V_{dd} \cdot I_{leak}) \cdot (n \cdot t_p) \end{aligned} \quad (7)$$

where

α	activity factor;
n	number of stages;
$E_{switch,inv}$	switching energy of a single inverter;
P_{leak}	total leakage power of the entire inverter chain;
t_d	delay of the entire inverter chain;
C_s	total switched capacitance of a single inverter;
I_{leak}	leakage current of a single inverter;
t_p	delay of a single inverter.

Note that we assume that short-circuit power is negligible and can be ignored. This assumption is known to hold for well-designed circuits in normal (super-threshold) operation [14]. Using the method in [15], we measured the short-circuit current for an inverter chain over a wide range of V_{dd} and have found that the short-circuit energy percentage is less than 9% at V_{min} and even lower as V_{dd} is reduced further, which is smaller than that at superthreshold, as shown in Fig. 4. Thus, we ignore the short-circuit component in energy modeling. Although rise and fall time increases almost exponentially with the reduced V_{dd} in subthreshold operation, the average short-circuit current also scales down almost exponentially with V_{dd} . Therefore, short-circuit energy does not increase in subthreshold and, in fact, diminishes because of the leakage energy increase.

First, we focus on finding an accurate estimate of t_p . Let $t_{p,step}$ denote the ideal inverter delay with a step input and $t_{p,actual}$ denote the actual inverter delay with an input rising time of t_r . We can compute $t_{p,step}$ based on a simple charge-based expression

$$t_{p,step} = \frac{\frac{1}{2} \cdot C_s \cdot V_{dd}}{I_{on}} \quad (8)$$

where I_{on} is the average on-current of an inverter. I_{on} can be estimated with alpha-power law [5] in superthreshold and takes the form of (5) (with V_{gs} and V_{ds} substituted by V_{dd}) in the subthreshold regime. Furthermore, for normal operating voltages, the step delay can be extended to the actual delay as follows [21]:

$$t_{pHL,actual} = \sqrt{t_{pHL,step}^2 + \left(\frac{t_r}{2}\right)^2}. \quad (9)$$

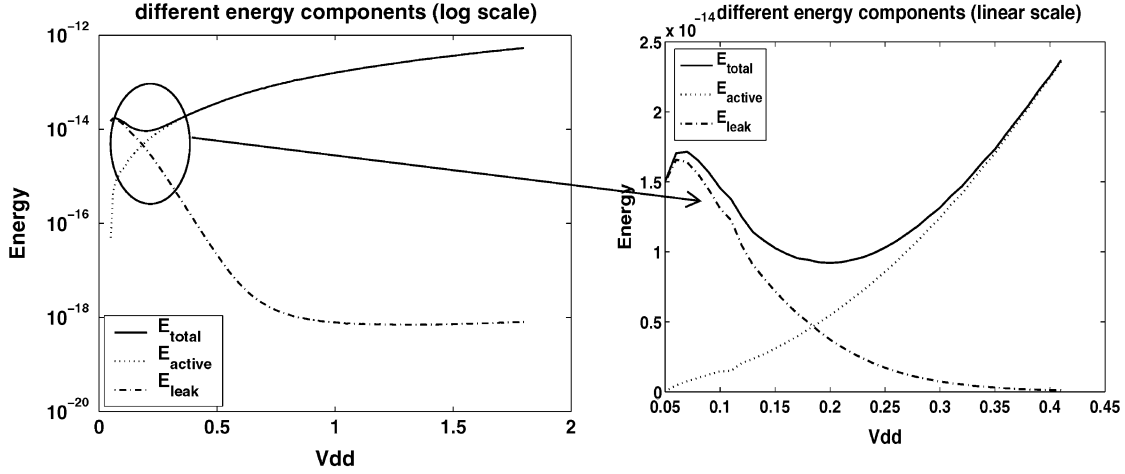


Fig. 3. Energy as a function of supply voltage.

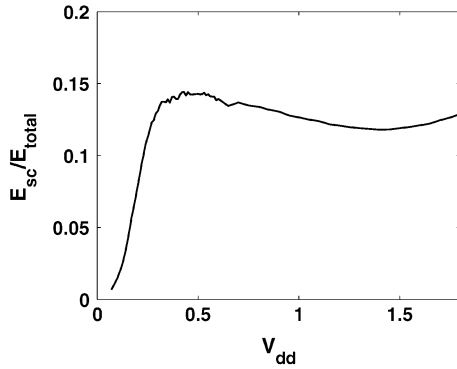


Fig. 4. Short-circuit energy as a function of supply voltage.

It is shown in [14] that, if $t_r > t_{pHL,actual}$ (which is satisfied when an inverter drives another one of the same size, as in our modeling), then

$$t_{pHL,actual} = 0.84 \cdot t_r. \quad (10)$$

Substituting (10) into (9) gives

$$t_{pHL,actual} = 1.2445 \cdot t_{pHL,step}. \quad (11)$$

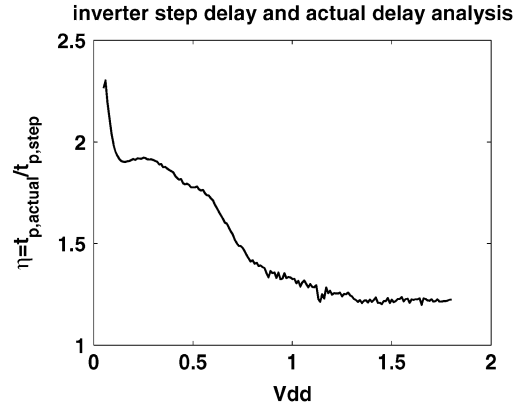
Similar results hold for t_{pLH} [14]. We then can estimate the average $t_{p,actual}$ as

$$\begin{aligned} t_{p,actual} &= 1.2445 \cdot t_{p,step} \\ &= \eta \cdot t_{p,step}. \end{aligned} \quad (12)$$

However, we need to test if this linear model is valid for subthreshold operation. To justify the linear modeling of $t_{p,actual}$ with $t_{p,step}$ at such a wide supply voltage range, we plot the calculated η as a function of V_{dd} , based on SPICE simulation in Fig. 5.

From Fig. 5, it is clear that the coefficient η increases as the supply voltage is reduced to the subthreshold regime. Other factors affecting the accuracy are that (5) does not perfectly model I_{sub} in subthreshold operation³ and that voltage swing degrades

³We find that, over the entire subthreshold region, ($0 < V_{dd} < V_{th}$), I_{sub} deviates from the simple exponential (5) by up to 20% if we treat mobility μ as constant.

Fig. 5. Ratio η in (12) with V_{dd} (SPICE).

at ultralow supply voltages. Taking these factors into account, we found for this technology an effective $\eta = 2.1$ for subthreshold operation.

As the supply voltage reduces, the total energy consumption reaches a minimum at some supply voltage since the delay of the circuit increases dramatically and the circuit now leaks over a larger amount of time. Substituting the equation for circuit delay (12) into (7), we obtain the following expression for total energy:

$$\begin{aligned} E &= \frac{1}{2} \cdot \alpha \cdot n \cdot C_S \cdot V_{dd}^2 + n \cdot V_{dd} \cdot I_{leak} \cdot \eta \cdot \frac{\eta C_S V_{dd}}{2 I_{on}} \\ &= \frac{1}{2} n C_S V_{dd}^2 \cdot \left(\alpha + \eta \cdot n \cdot \frac{I_{leak}}{I_{on}} \right). \end{aligned} \quad (13)$$

Note that I_{on} here is subthreshold “on” current because we are focusing on the subthreshold region where V_{min} occurs. By substituting (5) into (13) (I_{leak} is obtained by substituting V_{gs} and V_{ds} in (5) with 0 and V_{dd} , respectively), we now arrive at our final expression for the total energy as a function of supply voltage for subthreshold operation

$$E = \frac{1}{2} n C_S V_{dd}^2 \cdot \left(\alpha + \eta \cdot n \cdot e^{-\frac{V_{dd}}{m V_T}} \right). \quad (14)$$

Based on this simple expression of total energy, we can find the optimal minimum energy voltage V_{min} by setting

$\partial E / \partial V_{dd} = 0$. Letting $u = \eta \cdot n / \alpha$ and $t = V_{dd} / mV_T$, we obtain

$$e^t = \frac{u}{2} \cdot t - u. \quad (15)$$

Graphically, the solutions of (15) are the interactions of an exponential curve and a straight line, where u is positive. When $u = 2e^3$, there is only one solution $t = 3$ and $V_{min} = 3 mV_T$; when $u > 2e^3$, there are two solutions t_1 and t_2 , with the larger one ($t_1 > 3$) corresponding to V_{min} and the smaller one ($t_2 < 3$) being local optimum; when $u < 2e^3$, there is no solution. Therefore, $3mV_T$ is indeed the lowest possible V_{min} . We will show in the next section that complex gates have a higher V_{min} (if it exists) than inverters.

Since (15) is a nonlinear equation, it is impossible to solve it analytically. Hence, we use curve-fitting to arrive at the following closed-form expression:

$$t = 1.587 \ln u - 2.355. \quad (16)$$

Substituting the original variables gives the following final expression for the energy optimal voltage:

$$V_{min} = \left[1.587 \ln \left(\eta \cdot \frac{n}{\alpha} \right) - 2.355 \right] \cdot mV_T. \quad (17)$$

This closed-form formula is fitted for n and α values $20 < (\eta/\alpha) < 200$ and provides reasonable accuracy ($< 4.2\%$ V_{min} relative error compared to the numerical approach) over the data range.

Note that, in the presented model, the only parameters that are technology-dependent are η and m . As we switch from one technology to another, it is only required to determine these two parameters which can be easily accomplished. Interestingly, the total energy in (14) and the energy optimal voltage V_{min} *does not depend on the threshold voltage V_{th}* , as verified using SPICE. This independence is caused by the fact that, in subthreshold operation, both leakage and delay have similar dependencies on V_{th} , and hence the effect of V_{th} on the total energy cancels out. Also, we find that the energy optimal voltage is strongly dependent on the number of stages in the inverter chain. Active energy is linear with n whereas leakage energy is quadratic with n because, in a longer inverter chain, more gates are leaking, and these gates have more time to leak due to larger total propagation delay. Hence, V_{min} occurs at a higher voltage in a longer chain. Finally, we point out that V_{min} is strongly related to the activity factor α . In a circuit with lower α , V_{min} occurs at a larger voltage than in circuits with higher α , since lower activity gives the circuit more time to leak and effectively increases the stage number, as shown in Fig. 6. We therefore introduce the notation of effective stage number as $n_{eff} = (n/\alpha)$ to be used in the following analysis.

B. Model Verification and Extension to Circuit Blocks

In order to verify the accuracy of the proposed model, we compared the results from (14) with SPICE simulations for inverter chains of different lengths. In Fig. 7, we compare the energy- V_{dd} relationship predicted by the proposed analytical model in the subthreshold region with SPICE simulation results for an industrial $0.18\text{-}\mu\text{m}$ process. The plot shows a range

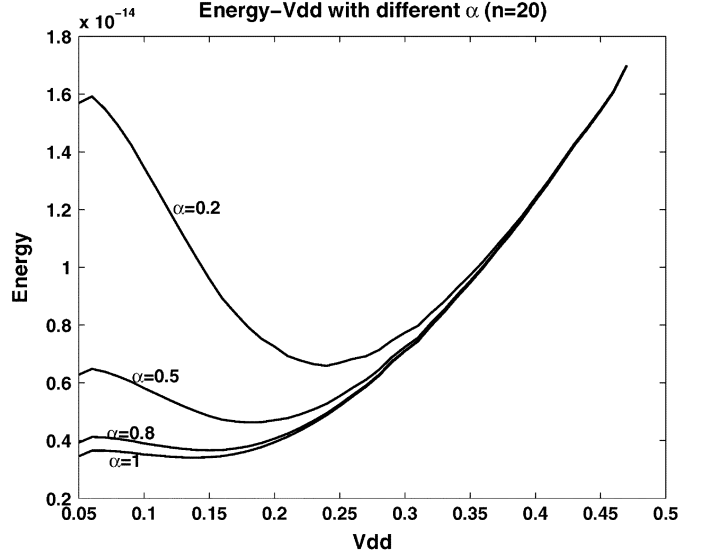


Fig. 6. Energy- V_{dd} for an inverter chain ($n = 20$).

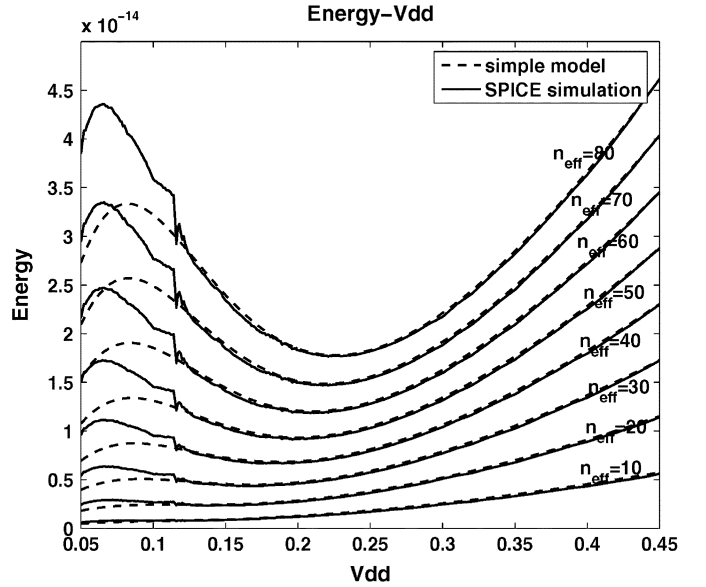
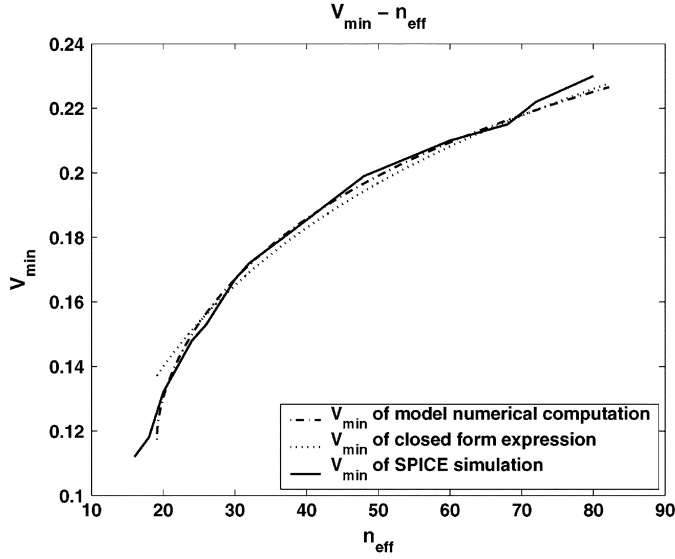
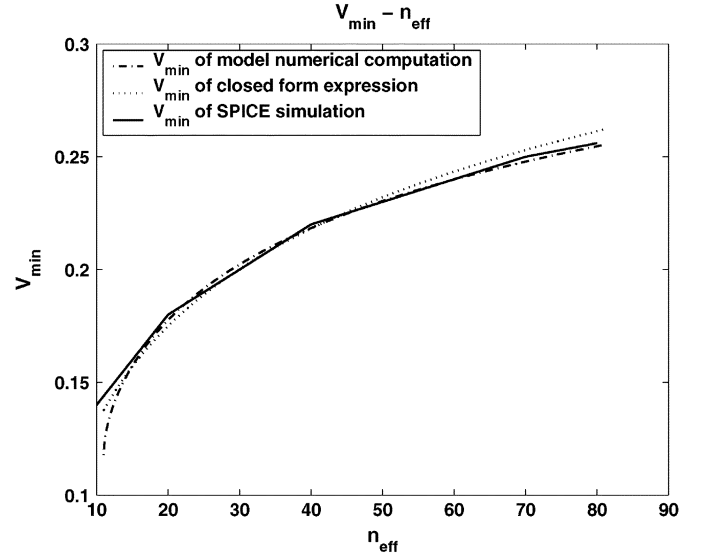
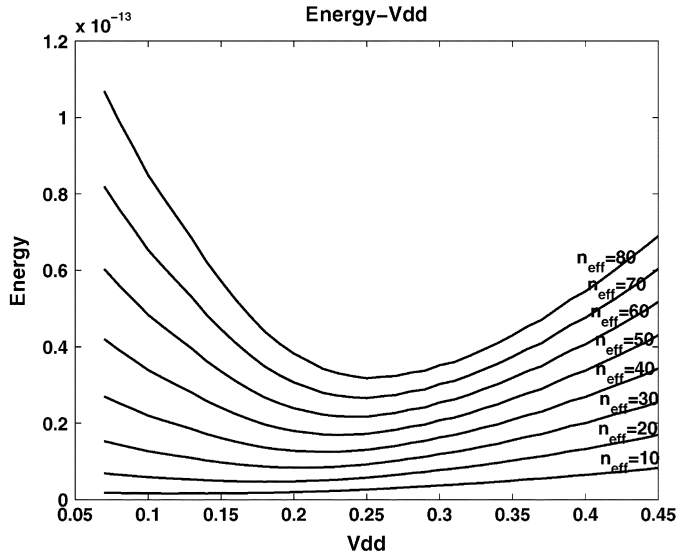


Fig. 7. Inverter chain energy- V_{dd} (analytical model versus SPICE).

of effective inverter chain lengths (n_{eff}). As shown in Fig. 7, the analytical model matches SPICE well, except at voltages less than 100 mV. In this region, the model tends to underestimate the rise in energy consumption due to the dramatic increase of η from Fig. 5, resulting in a delay that is greater than expected. However, this is not a severe problem since the important region of modeling around V_{min} shows good accuracy. When fanout-of-four (FO4) loading is added to each inverter, V_{min} is almost identical because this changes the term C_S in (13), which impacts only the absolute amount of energy consumed but not the minimum energy point (V_{min}).

In Fig. 8, we compare the predicted minimum energy voltage V_{min} based on our model with that measured by SPICE simulation. In the plot, the results using the fitted closed-form expression of (17) are shown, as well as the numerical solution of the nonlinear equation in (15). As can be seen, both match SPICE

Fig. 8. Minimal energy V_{\min} with inverter effective stage number n_{eff} .Fig. 10. Minimal energy V_{\min} with NAND2 effective stage number n_{eff} .Fig. 9. NAND2 chain energy- V_{dd} (SPICE).

with a high degree of accuracy for a wide range of effective inverter chain lengths n_{eff} .

We now consider the energy optimal voltage for more complex gates, such as NAND and NOR, as well as larger circuit blocks. Fig. 9 shows results of SPICE simulations for a NAND2. Similar to the inverter chain case in Section III-A, a single transition is used as a stimulus and energy is measured over the signal propagation through the chain. As can be seen, the minimum voltage V_{\min} shifts right compared to the inverter chain, indicating that the energy optimal voltage occurs at a higher voltage. This is caused by the fact that, for a chain of NAND2 gates, the number of leaking PMOS transistors is doubled in every other gate and NMOS transistors are twice the size to achieve comparable performance. The capacitance increase does not affect V_{\min} because the delay and switching energy are proportional to the loading C_s . This is evident in (13) (if we write out a

similar energy consumption expression for NAND2 case). However, in the NAND2 V_{\min} is different because the term I_{leak} is larger. Therefore, we can model the V_{\min} of a NAND2 chain by lumping the I_{leak} difference into n , as shown in (13). We introduce $n'_{\text{eff,inv}}$ as the equivalent stage number of an inverter chain that gives the same V_{\min} as a NAND2 chain with $n_{\text{eff,nand2}}$. It follows that $n'_{\text{eff,inv}}$ for NAND2 should be $2n_{\text{eff,nand2}}$ due to the doubled leakage current. However, with SPICE measurement we found that the $n'_{\text{eff,inv}}$ is somewhat smaller than $2n_{\text{eff,nand2}}$ due to the stack effect in the NMOS transistors. At the same time, the driving capability of the pull-down NMOS when sized two times is slightly larger than that of the inverter. We therefore empirically compute $n'_{\text{eff,inv}}$ value for the NAND2 chain

$$\begin{aligned} \frac{n_{\text{eff,inv}}}{n_{\text{eff,nand2}}} &= \frac{I_{\text{leak,nand2}}}{n_{\text{leak,inv}}} \cdot \frac{I_{\text{on,inv}}}{I_{\text{on,nand2}}} \\ &\cong 1.91 \cdot \frac{1}{1.1} \\ &\cong 1.74 \end{aligned} \quad (18)$$

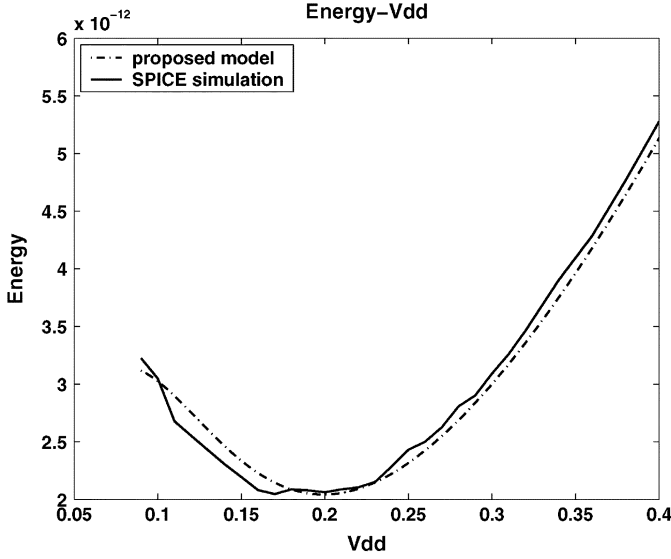
where the average leakage current ratio and on-current ratio are from SPICE simulation.

Using this $n'_{\text{eff,inv}}$, we obtain an accurate match between the modeled V_{\min} and SPICE simulation, as shown in Fig. 10. Other complex gates can be modeled in a similar way by computing an appropriate $n'_{\text{eff,inv}}$ value.

This approach can be extended to larger circuit blocks as well. In Fig. 11, we show the total energy as a function of supply voltage obtained using SPICE for a 16×16 multiplier with activity factor $\alpha = 0.5$, where α is the average switching activity over all of the nodes in the circuits. The multiplier was obtained by synthesizing the ISCA85 benchmark circuit c6288 [22] with an industrial $0.18\text{-}\mu\text{m}$ process using a wireload model for parasitics. We estimate the total power consumption for large circuit blocks such as this by extending the expression in (14) as follows:

$$E_{\text{total}} = E_{\text{active}} + E_{\text{leak}} \quad (19)$$

$$E_{\text{active}} = \alpha \cdot C_{w0} \cdot W_{\text{total}} \cdot V_{\text{dd}}^2 \quad (20)$$

Fig. 11. Energy- V_{dd} for the 16×16 multiplier circuit.

where W_{total} is the total width of all of the transistors in the circuit and C_{w0} is the switched capacitance of a unit width transistor. We compute the total leakage energy as follows:

$$\begin{aligned} E_{\text{leak}} &= I_{\text{leak},\text{total}} \cdot V_{dd} \cdot t_d \\ &= (\gamma_{\text{leak}} \cdot W_{\text{total}} \cdot I_{\text{leak}0}) \cdot V_{dd} \cdot (n_{\text{depth}} \cdot t_{p,\text{FO4}}) \end{aligned} \quad (21)$$

where γ_{leak} is the leaking factor used to model the leakage stack effect and input pattern dependency, $I_{\text{leak}0}$ is the leakage current of a unit width transistor, and n_{depth} is the logic depth in terms of fanout-of-four (FO4) inverter delay $t_{p,\text{FO4}}$, which is expressed as follows:

$$t_{p,\text{FO4}} = \frac{\frac{1}{2} \cdot (4W_{\text{inv}} \cdot C_{w0}) \cdot V_{dd}}{W_{\text{inv}} \cdot I_{\text{on}0}} \quad (22)$$

where $I_{\text{on}0}$ is the on-current of a unit width inverter. Note that α may change with supply voltage as glitches are sensitive to circuit delay, although, for simplicity, we treat α as a constant. Substituting (20) and (21) into (19), we can derive the following expression for total energy of a circuit block as a function of supply voltage in a manner similar to (14):

$$E_{\text{total}} = C_{w0}W_{\text{total}}V_{dd}^2 \left(\alpha + 2\gamma_{\text{leak}}n_{\text{depth}}e^{-\frac{V_{dd}}{mV_T}} \right). \quad (23)$$

For the test circuit in Fig. 11, the following parameters for the model were found using SPICE simulation: $\gamma_{\text{leak}} \cong 0.5$, $n_{\text{depth}} \cong 65$. The total energy predicted by (23) with the above parameters is shown in Fig. 11 for the 16×16 multiplier block together with SPICE simulation results. In order to evaluate the V_{min} for various circuit blocks, logic depth n_{depth} , average activity factor α , and leakage factor γ_{leak} are required. n_{depth} and α can be estimated with a static timing analysis tool and γ_{leak} requires dc simulation at the transistor level, which is significantly faster than transient analysis.

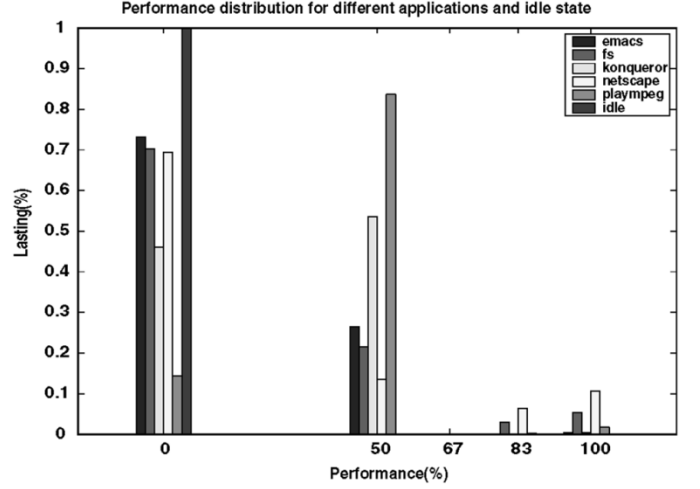


Fig. 12. Recorded performance distribution of different applications.

It is important to note that, for a generic circuit block, n_{eff} is defined as $n_{\text{eff},\text{block}} \equiv (n_{\text{depth}})/(\alpha)$. Therefore, when the activity factor α is very low, based on either the circuit structure or the input data stream, the $n_{\text{eff},\text{block}}$ is much larger than the real logic depth n_{depth} . In a real processor, the activity factor varies across the chip since not all circuit blocks are working intensively at all times. Therefore, in order to achieve energy efficiency, designers must take into account the α difference before estimating the average V_{min} . In other words, for the purposes of optimizing DVS, low activity and large logic depths are interchangeable as they both lead more quickly to leakage dominated designs.

C. Energy Optimality for Different Workloads With DVS

As discussed earlier, the energy optimal voltage depends on both circuit and technology characteristics. At the same time, the best choice for the minimum allowed voltage for a processor depends on its workload distribution. If the workload of a processor is such that low-performance levels are never or rarely required, the minimum operating voltage for energy-efficient operating will be larger than the energy optimal voltage V_{min} computed in Section III-A. Hence, we studied a number of different real applications running on Linux using ARM926 and Transmeta Crusoe TM5600 processors with DVS and recorded traces of the minimum necessary performance levels for each application using real-time monitoring. These real applications are selected based on typical activities of laptop computers and comprise both multimedia and interactive applications:

- *emacs* is a trace of user activity using the editor performing light text-editing tasks;
- *konqueror* and *netscape* are traces of web browsing sessions using the two browsers;
- *fs* contains a record of filesystem-intensive operations;
- *mpeg* is a trace using MPEG2 video playback;
- *idle* traces the activity when the system has no dominant workloads and, as a result, contains little activity except for operating system housekeeping tasks.

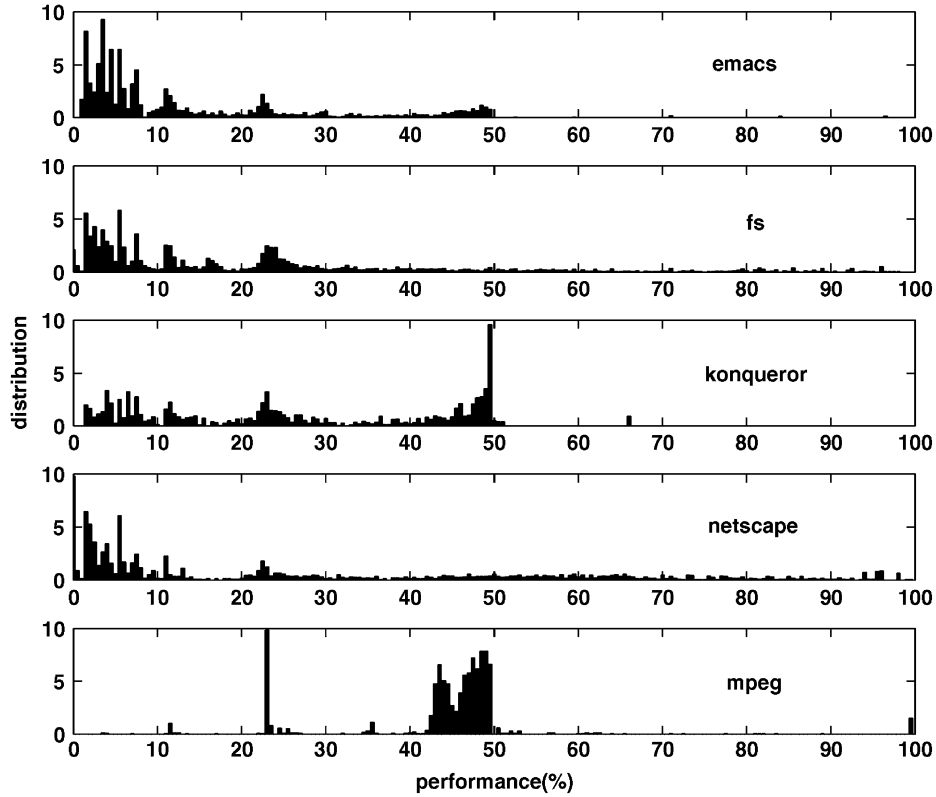


Fig. 13. Histogram of different workloads converted from real traces.

The dynamic performance management policy is based on Vertigo [9] and ARM's Intelligent Energy Manager.⁴ The distribution of the four available performance levels (with a highest frequency of 600 MHz) among the executed tasks is shown in Fig. 12 for each application. This workload distribution is recorded from a real DVS processor running applications. As the bar graph shows, the processor spends significant time in sleep mode, meaning that the processor completes many tasks well ahead of schedule. Most importantly, we observed that, during the execution of all tasks, a run-then-idle pattern was seen 50% of the time. This implies that many tasks could run at a frequency less than the minimum (50%) available on the processor if it was able to do so. Therefore, we convert these traces to ideal continuous distributed performance levels and obtain the histogram shown in Fig. 13. The histograms show that *emacs*, *fs*, and *netscape* spend most of their execution time in the low-performance end.

By extending the lower limit of voltage scaling, the amount of idle time can be reduced, leading to more energy-efficient operation. Based on the previous analysis, energy efficiency can increase until it reaches the energy optimal voltage V_{\min} . In addition, by eliminating the need to enter a sleep state, any energy overhead due to switching to and from sleep mode is also avoided, further increasing the energy efficiency.

We therefore study the total energy consumption of the processor as a function of the lower limit of the performance that the processor provides, denoted by f_{limit} . Assuming that we have an ideal performance scheduler that is able to set the performance

exactly sufficient to just complete every task, we can compute the optimal energy consumption with different f_{limit} values. The total energy is based on the proposed energy model of Section III-A for subthreshold voltage operation, combined with a simple fitted model for energy and delay at super-threshold operating voltages. Note that we do not consider the sleep-wakeup energy overhead now since a more detailed energy model for this will be presented in Section IV. We show the energy/ f_{limit} tradeoff for the first five applications in Fig. 14. As can be seen, the energy efficiency improves as the f_{limit} is reduced and levels off for most applications below 10%, which corresponds to a $V_{\text{dd}}/V_{\text{dd0}}$ of 30.7% (553 mV for a V_{dd0} of 1.8 V). We also analyze the energy/ f_{limit} tradeoff for the idle-mode trace, in which the processor is mostly in sleep mode and wakes up only to do regular "housekeeping" chores for the operating system. Note that this state can be quite common on a processor. The results show that the energy continues to reduce down to a performance level of 0.02%, corresponding to a $V_{\text{dd}}/V_{\text{dd0}}$ of 13% (234 mV for a V_{dd0} of 1.8 V). In such low-activity situations, the *practical* V_{\min} value approaches the theoretical V_{\min} levels of Section III-A. The energy savings of a more voltage scalable processor over the traditional one are summarized in Table II, demonstrating that substantial energy savings can be obtained by extending the voltage range appropriately.

IV. ENERGY EFFICIENCY EVALUATION OF DIFFERENT LOW-POWER APPROACHES

A. Detailed Energy Modeling

In this section, we compare the energy efficiency of different low power design approach. In the analysis we include the over-

⁴[Online]. Available: <http://www.arm.com/products/CPU/cpu-arch-IEM.html>

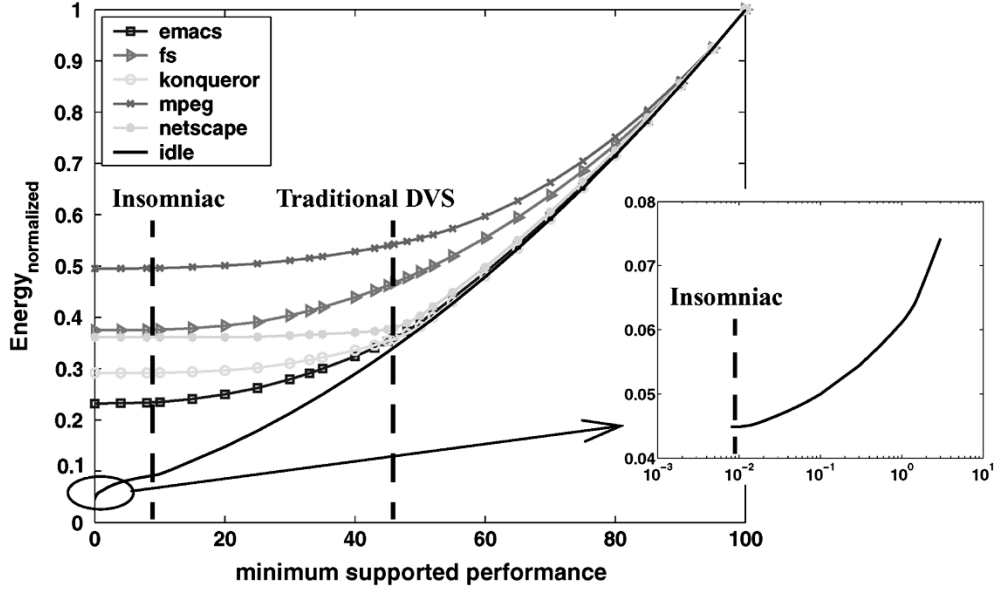
Fig. 14. Energy f_{limit} for different applications.

TABLE II
ENERGY CONSUMPTION COMPARISON BETWEEN INSOMNIAC AND
TRADITIONAL DVS APPROACHES

Application	Normalized Energy		Energy Savings
	Insomniac	Traditional DVS	
emacs	0.235	0.359	34.5%
fs	0.376	0.467	19.5%
konqueror	0.292	0.358	18.4%
netscape	0.361	0.380	5%
mpeg	0.496	0.542	8.49%
idle state	0.0458	0.3324	86.2%

note: In Insomniac, V_{dd}/V_{dd0} is 30.7% for general applications, 13% for idle state; in traditional DVS, V_{dd}/V_{dd0} is assumed as 50%.

head that each specific low power technique incurs, as well as the efficiency of the dc–dc voltage converter. First, we define five different systems:

- S_{basic} , a basic system with clock gating but without power-gating or DVS;
- S_{mtcmos} , a system that employs power-gating during idle mode (clock gating is implied) but no DVS; power-gating has been an effective approach in leakage power reduction [25];
- S_{dvspg} , a partial DVS system with power-gating capability where the minimum scalable voltage is V_{limit} , set to be $V_{dd}/2$;
- S_{dvsonly} , a system similar to S_{dvspg} but without power-gating;
- S_{insom} , an Insomniac system with aggressive voltage scaling capability, down to the energy optimal voltage; note that in this system power gating is not necessary, since processor always runs in “just in time” completion mode.

For S_{basic} , the energy consumption during idle mode is leakage and we can model the total energy E_{basic} for a given workload as

$$E_{\text{basic}} = P_{\text{act,vdd}} \cdot t_{\text{on}} + P_{\text{leak}} \cdot (t_{\text{on}} + t_{\text{off}}) \quad (24)$$

where t_{on} is the time the processor stays busy and t_{off} is the idle time.

For S_{mtcmos} , the energy consumption is modeled as

$$E_{\text{mtcmos}} = P_{\text{act,vdd}} \cdot t_{\text{on}} + P_{\text{leak}} \cdot t_{\text{on}} + E_{\text{overhead}}$$

$$E_{\text{overhead}} = \frac{1}{2} \cdot C_{\text{powerail}} \cdot V_{dd}^2 + \frac{1}{2} \cdot C_{\text{internal}} \cdot V_{dd}^2 + C_{\text{sleep}} \cdot V_{dd}^2 \quad (25)$$

where E_{overhead} is the overhead energy when gating the power supply, C_{powerail} is the virtual supply rail capacitance, C_{internal} is the total internal node capacitance of the circuit, and C_{sleep} is the gate capacitance of the sleep transistors. E_{overhead} arises from three sources: the power rail charge loss, the circuit internal node charge loss, and sleep transistor gate charge needed to conduct power gating. Depending on whether a header or footer transistor is used in S_{mtcmos} , the processor will lose the voltage level on virtual V_{dd} or virtual ground, respectively, shortly after it enters sleep mode [28]. Without loss of generality, we use the average between virtual ground and virtual V_{dd} capacitance in our calculations. To consider the internal node capacitances, we assume that half of the internal nodes are at each logic state 1/0.

According to recent research [28], [29], the virtual power rail can be restored in several cycles if the circuit is carefully designed. Thus, the wakeup process can be treated as instantaneous compared to the thousands/millions of cycles of useful operation.

In order to model S_{dvspg} and S_{dvsonly} , we must know how a DVS system implements the voltage scaling process. There are

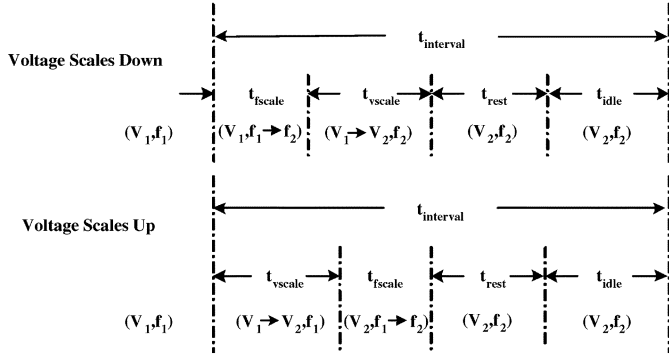
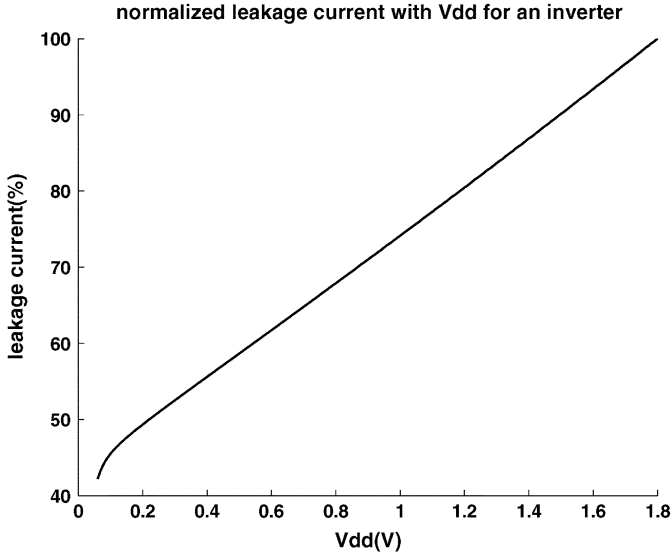


Fig. 15. Illustration of scaling process for a DVS system.

Fig. 16. Inverter leakage current versus V_{dd} .

several ways to design a DVS system [26], [27]. In this paper, we assume a method similar to the IBM405LP [27], which is illustrated in Fig. 15. First, we define some system constants as follows:

- $\tau_f = 1 \mu s$, the constant time it takes to scale the frequency;
- $k_v = 2 mV/\mu s$, the speed of the regulator to scale the voltage;
- $g_{off} = 2.64e-3 siemens$, the equivalent transistor off-state conductance due to DIBL.

The g_{off} parameter is introduced to characterize the leakage current dependency on supply voltage, which comes from V_{th} roll-off due to the DIBL effect

$$V_{th} \cong V_{th0} - (\gamma \cdot V_{BS}) - (\eta_{DIBL} \cdot V_{DS}). \quad (26)$$

According to (5) [33], the “off” state current of an NMOS transistor can be expressed as

$$I_{leak} \propto e^{-\frac{V_{th}}{mV_T}} \propto e^{-\frac{\eta_{DIBL} \cdot V_{dd}}{mV_T}} \propto 1 + \frac{\eta_{DIBL} \cdot V_{dd}}{mV_T}. \quad (27)$$

If η_{DIBL} is small, this linear relationship holds as a good approximation until V_{dd} drops to several mV_T , when the exponential term with V_{ds} in (5) comes into play, as shown in Fig. 16. Based on this notion, we define g_{off} as

$$g_{off} \equiv \frac{\Delta I_{leak}}{\Delta V_{dd}} = \frac{\eta_{DIBL}}{mV_T}. \quad (28)$$

We find that, for the 0.18- μm process, we are considering the average g_{off} for NMOS and PMOS is around $2.64e-3 siemens$, which is found by SPICE simulation.

If we suppose that the system scales from state (V_1, f_1) to (V_2, f_2) , then there are two possible cases.

- 1) Scaling voltage down— $V_2 \leq V_1$ For $S_{dvsonly}$, the energy consumption for a certain application is

$$\begin{aligned} E_{dvsonly, sd} &= (P_{act, V_1} + P_{leak, V_1}) \cdot \tau_f \\ &+ E_{vscale, f_2, V_1 \rightarrow V_2} + (P_{act, V_2} + P_{leak, V_2}) \cdot t_{rest} \\ &+ P_{leak, V_2} \cdot t_{idle} \end{aligned} \quad (29)$$

For S_{dvspg} , the energy consumption is

$$\begin{aligned} E_{dvspg, sd} &= (P_{act, V_1} + P_{leak, V_1}) \cdot \tau_f + E_{vscale, f_2, V_1 \rightarrow V_2} \\ &+ (P_{act, V_2} + P_{leak, V_2}) \cdot t_{rest} \\ &+ \delta \cdot \left(\frac{1}{2} \cdot C_{internal} \cdot V_2^2 + C_{sleep} \cdot V_2^2 \right. \\ &\quad \left. + \frac{1}{2} \cdot C_{powerrail} \cdot V_2^2 \right). \end{aligned} \quad (30)$$

δ is a discrete parameter reflecting whether or not the requested V_{dd} value ($V_{2, request}$) can be achieved, which impacts whether power gating is used. Therefore

- if $(V_{2, request} \geq V_{limit})$: $V_2 = V_{2, request}$, $t_{idle} = 0$, $\delta = 0$
- if $(V_{2, request} < V_{limit})$: $V_2 = V_{limit}$, $t_{idle} > 0$, $\delta = 1$.

- 2) Scaling voltage up— $V_2 > V_1$, for both $S_{dvsonly}$ and S_{dvspg}

$$\begin{aligned} E_{dvs, su} &= (P_{act, V_2} + P_{leak, V_2}) \cdot \tau_f + E_{vscale, f_1, V_1 \rightarrow V_2} \\ &+ (P_{act, V_2} + P_{leak, V_2}) \cdot t_{rest} \\ &+ \frac{1}{2} \cdot C_{powerrail} \cdot (V_2^2 - V_1^2) \\ &+ \frac{1}{2} \cdot C_{internal} \cdot (V_2^2 - V_1^2). \end{aligned} \quad (31)$$

For an optimal voltage up-scaling process, t_{idle} is always zero implying that $S_{dvsonly}$ and S_{dvspg} have the same expression for energy consumption.

A key difference between scaling up and scaling down voltages is that scaling up involves extra energy consumption caused by voltage level change as it draws current from the power supply to charge up this level difference.

In order to make a practical comparison, we extract detailed physical parameters of an existing Alpha processor from its layout as the basis for the following discussion (shown in Table III).

In leading-edge processors, leakage power is much more substantial than that shown in the baseline 0.18- μm Alpha processor. Therefore, we intentionally reduce the V_{th} in this process to set the leakage power to $\sim 10\%$ of active power, which is reasonable for modern processors.

B. Regulator Efficiency Modeling

DVS involves a wide range of operation, and therefore the efficiency of the voltage regulator must also be taken in account.

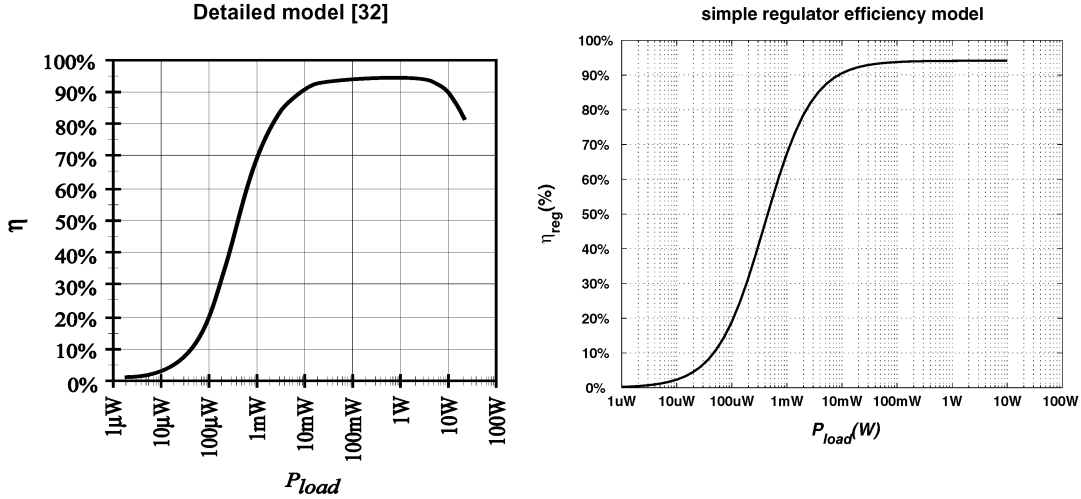


Fig. 17. Verification of the simple regulator efficiency model.

TABLE III
PHYSICAL PARAMETERS FROM AN ALPHA PROCESSOR IN A 0.18- μm
TECHNOLOGY (WITHOUT CACHES CONSIDERED)

nominal supply voltage	1.8 volts
total # of transistors	554,052
total # of gates	39,703
power rail capacitance (V_{dd} & ground)	447.3 pF
internal node capacitance (interconnect included)	3,520 pF
total gate capacitance	948 pF
sleep transistor gate cap. (assumed to be 10% gate cap)	95 pF

The power converting efficiency η_{reg} of an off-chip dc–dc converter⁵ is defined as

$$\eta_{\text{reg}} = \frac{P_{\text{load}}}{P_{\text{load}} + P_{\text{loss}}} \quad (32)$$

where P_{load} is the loading power and P_{loss} is the total loss inside the converter. P_{loss} can be further expressed as [30]

$$\begin{aligned} P_{\text{loss}} &= P_{\text{cond}}(I_{\text{load}}) + P_{\text{switch}} + P_{\text{fixed}} \\ P_{\text{switch}} &= E_{\text{switch}} \cdot f_{\text{switch}} \end{aligned} \quad (33)$$

where P_{cond} is the converter conduction loss, directly dependent on load current and E_{switch} is the energy loss during the transistor switch-on and switch-off transitions, dependent on the switching frequency f_{switch} , and P_{fixed} is the fixed loss, dependent on neither load current nor switching frequency.

With a more efficient regulator [30]–[32], such as a variable-frequency regulator, P_{switch} can be made to scale with load current I_{load} . Given the fact that regulator efficiency is nearly constant over a wide loading range [30], we can assume that the conduction loss P_{cond} and the switching loss P_{switch} scale in the same way as P_{load} . That is,

$$P_{\text{cond}} + P_{\text{switch}} = c_{\text{loss}} P_{\text{load}} \quad (34)$$

⁵In the context of DVS, we usually need a buck dc–dc converter to realize down-converting, therefore all the modeling in this section is based on an off-chip buck dc–dc converter.

where the c_{loss} is defined as the converting loss coefficient for a specific regulator. Since the P_{fixed} is small for nominal converting region, we can get the nominal converting efficiency as

$$\begin{aligned} \eta_{\text{reg,nom}} &= \frac{P_{\text{load}}}{P_{\text{load}} + P_{\text{cond}} + P_{\text{switch}}} \\ &= \frac{1}{1 + c_{\text{loss}}} \end{aligned} \quad (35)$$

Then, a simple regulator efficiency model is developed as

$$\eta_{\text{reg}} = \frac{P_{\text{load}}}{P_{\text{load}} + P_{\text{loss}}} = \frac{P_{\text{load}}}{(1 + c_{\text{loss}})P_{\text{load}} + P_{\text{fixed}}} \quad (36)$$

To verify this modeling approach, we compare it with the more detailed model in [30], as shown in Fig. 17 ($c_{\text{loss}} = 0.0625$, $P_{\text{fixed}} = 420 \mu\text{W}$ are used in the plots).

With the simple power converting model of (36), we can readily take into account the regulator degradation by

$$\begin{aligned} E_{\text{total}} &= \int_0^{t_{\text{total}}} \frac{P_{\text{load}}(t)}{\eta_{\text{reg}}} dt \\ &= \int_0^{t_{\text{total}}} [(1 + c_{\text{loss}})P_{\text{load}} + P_{\text{fixed}}] dt \\ &= (1 + c_{\text{loss}})E_{\text{load}} + P_{\text{fixed}} \cdot t_{\text{total}} \end{aligned} \quad (37)$$

In the following section, we include the regulator loss factor into our computation by means of (37).

C. Energy Efficiency Evaluation Results

With energy models derived for the various low-power schemes, we can now evaluate their energy efficiency under different applications. We studied the same applications as in Section III, i.e., *emacs*, *konqueror*, *netscape*, *fs*, and *mpeg*. To make a fair comparison, we convert these traces to match the Alpha processor that we are using in the analysis. By applying four different low-power schemes to these workloads, we computed the energy savings relative to S_{basic} based on the models in Section IV-A. The results are shown in Fig. 18. As the bar graph shows, if the voltage cannot scale as low as the applications request (S_{dvsonly}), it is helpful to utilize power-gating (S_{dvspg}) to save leakage energy. However, the

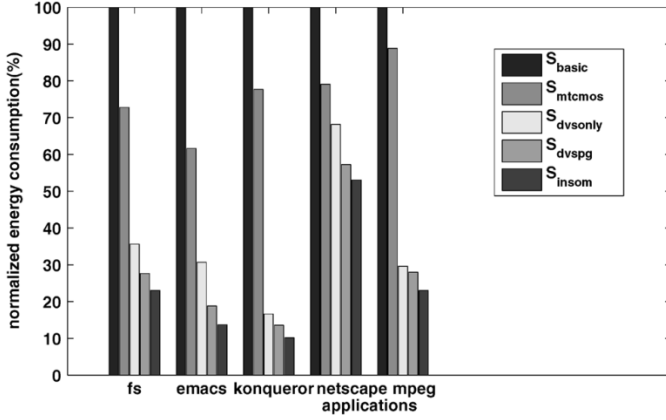


Fig. 18. Comparison of energy consumption under different low-power schemes.

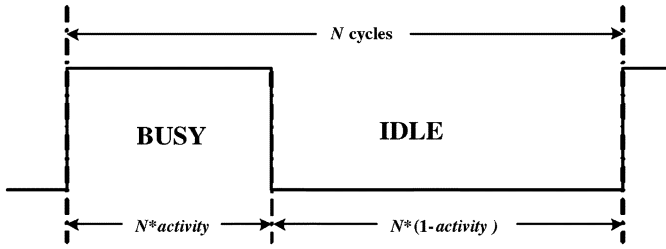


Fig. 19. Illustration of cycle number N and activity.

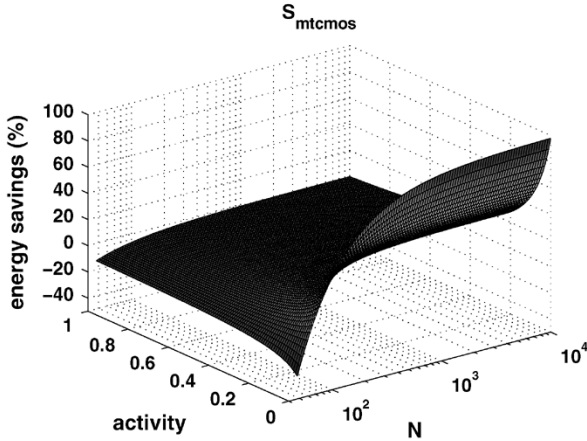


Fig. 20. Energy savings with S_{mtcmos} for general workloads.

largest savings for all five applications is seen with the In-somniac system (S_{insom}). For instance, the energy savings of S_{insom} over S_{dvspg} is 27% for *emacs*, and 25% for *konqueror*.

To obtain a more general evaluation for these approaches, we analyze the energy savings under an artificial workload. We characterize a workload by two parameters: N , *activity*, where the N is the total number of cycles that the processor will run before the deadline at normal operation mode, and *activity* is the ratio of the number of working cycles to N , as shown in Fig. 19. Both of these two factors influence the design choice of low-power systems.

The energy savings for S_{mtcmos} with general workloads are shown in Fig. 20. S_{mtcmos} is useful when *activity* is very low and the number of cycles is large, which gives savings close to

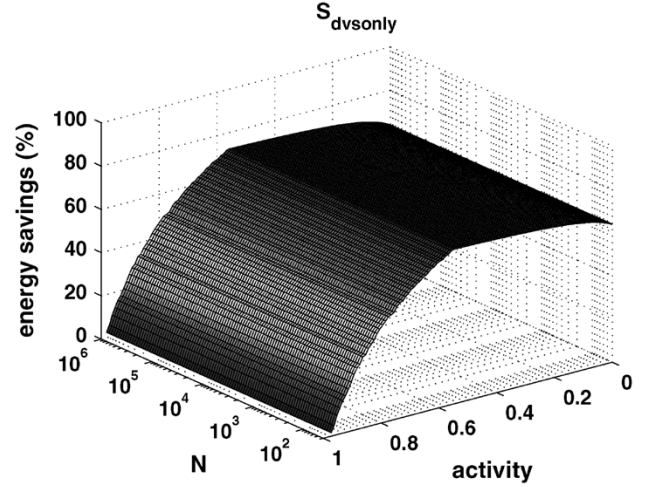


Fig. 21. Energy savings with $S_{dvsonly}$ for general workloads.

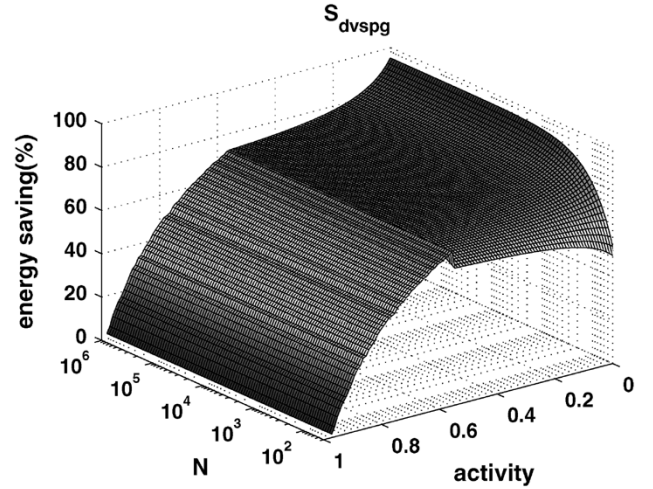


Fig. 22. Energy savings with S_{dvspg} for general workload.

100% and even better savings than S_{dvspg} . This is because energy is almost completely due to leakage in the S_{basic} system. The reason that we find negative savings for S_{mtcmos} in some cases is that the extra power gating energy outweighs the potential leakage energy in S_{basic} .

Figs. 21 and 22 contain the results for $S_{dvsonly}$ and S_{dvspg} , respectively. A system without power gating is independent of the total number of cycles, which can be easily derived from the models in Section IV-A. When activity is high and the application never requests a voltage less than half V_{dd} , there will be no idle period if an optimal scheduler is used. Therefore no difference exists between $S_{dvsonly}$ and S_{dvspg} in this activity range. As activity drops and leads to requested voltages lower than V_{limit} , S_{dvspg} becomes better than $S_{dvsonly}$ because of the leakage savings when power gating is applied. This confirms again that for modern state-of-the-art partial-DVS systems, it is helpful that the system includes power-gating to avoid unwanted leakage at run-time.

For S_{insom} , the energy savings are again independent of N since power gating is not used, but for consistency we still plot the savings in three dimensions as shown in Fig. 23. It is clearly shown that S_{insom} can easily provide energy gains for a very

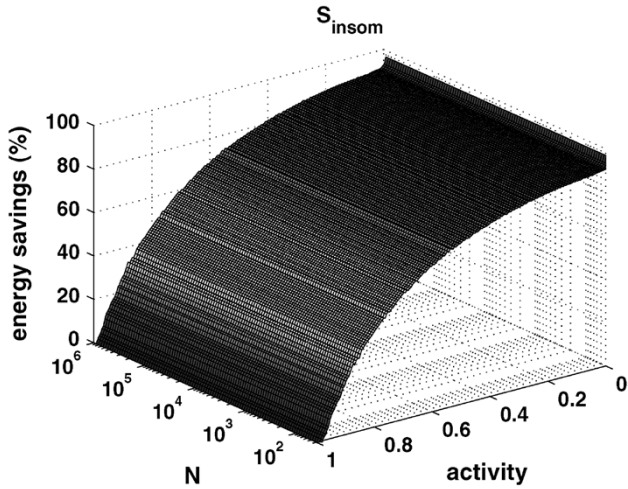


Fig. 23. Energy savings with S_{insom} for general workload.

wide range of activity, and gives significant savings over S_{dvspg} or S_{mtcmos} . Only when activity is very low does the energy savings become saturated, which occurs since the leakage dominates the overall system energy consumption. The system has scaled below V_{min} at this point, as described in Section III.

V. CONCLUSION

In this paper, we developed analytical models for the most energy efficient supply voltage (V_{min}) for CMOS circuits. A number of interesting conclusions are drawn: 1) energy shows a clear minimum in the subthreshold region since the time over which a circuit is leaking (delay) grows exponentially in this region while leakage current itself does not drop as rapidly with reduced V_{dd} ; 2) V_{min} does not depend on V_{th} if V_{min} is smaller than V_{th} ; 3) the circuit logic depth and switching factor impact V_{min} since they relate to the relative contributions of leakage energy and active energy; and 4) the only technology parameters relevant to V_{min} are subthreshold swing and the dependency of delay on input transition time. The proposed analytical models are shown to match very well with SPICE simulations. We then compare the energy savings of different low-power schemes, namely, pure MTCMOS, partial-DVS, partial-DVS with MTCMOS, and Insomniac. The comparison for five applications traces recorded on two commercial processors shows that Insomniac provides the best efficiency. For instance, it can provide 27% energy savings for *emacs* over the traditional DVS-with-MTCMOS design. A comparison for arbitrary workloads shows that for the majority of different application activity ratios Insomniac continues to yield the largest energy improvements.

REFERENCES

- [1] Transmeta Crusoe. [Online]. Available: <http://www.transmeta.com/>
- [2] Intel XScale. [Online]. Available: <http://www.intel.com/design/intelxscale/>
- [3] IBM PowerPC. [Online]. Available: <http://www.chips.ibm.com/products/powerpc/>
- [4] K. Flautner, S. Reinhardt, and T. Mudge, "Automatic performance setting for dynamic voltage scaling," in *Proc. 7th Annu. Int. Conf. Mobile Computing and Networking (MobiCom'01)*, May 2001, pp. 260–271.

- [5] T. Sakurai and A. Newton, "Alpha-power law MOSFET model and its applications to CMOS inverter delay and other formulas," *IEEE J. Solid-State Circuits*, vol. 25, no. 2, pp. 584–594, Apr. 1990.
- [6] B. Zhai, S. Hanson, D. Blaauw, and D. Sylvester, "Analysis and mitigation of variability in subthreshold design," in *Proc. Int. Symp. Low Power Electronics and Design (ISLPED)*, 2005, pp. 20–25.
- [7] M. Miyazaki, J. Kao, and A. Chandrakasan, "A 175 mV multiply-accumulate unit using an adaptive supply voltage and body bias (ASB) architecture," in *Proc. Int. Solid-State Circuits Conf. (ISSCC)*, 2002, pp. 58–59.
- [8] A. Wang and A. Chandrakasan, "A 180 mV FFT processor using subthreshold circuits techniques," in *Proc. Int. Solid-State Circuits Conf. (ISSCC)*, 2004, pp. 292–294.
- [9] K. Flautner and T. Mudge, "Vertigo: Automatic performance-setting for linux," in *Proc. 5th Symp. Operating Systems Design Implementation*, Dec. 2002, pp. 105–116.
- [10] J. D. Meindl and J. A. Davis, "The fundamental limit on binary switching energy for terascale integration (TSI)," *IEEE J. Solid-State Circuits*, vol. 35, no. 10, pp. 1515–1516, Oct. 2000.
- [11] R. M. Swanson and J. D. Meindl, "Ion-implanted complementary MOS transistors in low-voltage circuits," *IEEE J. Solid-State Circuits*, vol. 7, no. 4, pp. 146–153, Apr. 1972.
- [12] F. Miller, "Algorithm and architecture of a 1 V low power hearing instrument DSP," in *Proc. Int. Symp. Low Power Electronics and Design (ISLPED)*, Aug. 1999, pp. 7–11.
- [13] H. Soeleman, K. Roy, and B. Paul, "Robust ultra-low power sub-threshold DTMOS logic," in *Proc. Int. Symp. Low Power Electronics and Design (ISLPED)*, 2000, pp. 25–30.
- [14] J. Rabaey, *Digital Integrated Circuits: A Design Perspective*. Upper Saddle River, NJ: Prentice-Hall, 1996.
- [15] N. H. E. Weste and K. Eshraghian, *Principles of CMOS VLSI Design, a Systems Perspective*, 2nd ed. Reading, MA: Addison-Wesley, 2000.
- [16] H. Soeleman and K. Roy, "Ultra-low power digital subthreshold logic circuits," in *Proc. Int. Symp. Low Power Electronics and Design (ISLPED)*, 1999, pp. 94–96.
- [17] S. C. Terry, J. M. Rochelle, D. M. Binkley, B. J. Blalock, and D. Foty, "Comparison of a BSIM3v3 and EKV MOSFET model for a 0.5 μ m CMOS process and implications for analog circuit design," *IEEE Trans. Nucl. Sci.*, vol. 50, no. 4, pp. 915–920, Aug. 2003.
- [18] H. Soeleman and K. Roy, "Digital CMOS logic operation in the sub-threshold region," in *Proc. Great Lakes Symp. VLSI*, Mar. 2000, pp. 107–112.
- [19] A. Forestier and M. R. Stan, "Limits to voltage scaling from the low power perspective," in *Proc. 13th Symp. Integrated Circuits and Systems Design*, Sept. 2000, pp. 365–370.
- [20] A. Wang, A. P. Chandrakasan, and S. V. Kosonocky, "Optimal supply and threshold scaling for subthreshold CMOS circuits," in *Proc. IEEE Symp. VLSI*, Apr. 2002, pp. 5–9.
- [21] D. Hodges and H. Jackson, *Analysis and Design of Digital Integrated Circuits*. New York: McGraw-Hill, 1988.
- [22] F. Brglez and H. Fujiwara, "A neural netlist of 10 combinational circuits and a target translator in fortran," in *Proc. IEEE ISCAS*, Jun. 1985, pp. 663–698.
- [23] T. D. Burd and R. W. Brodersen, "Design issues for dynamic voltage scaling," in *Proc. Int. Symp. Low Power Electronics and Design (ISLPED)*, 2000, pp. 9–14.
- [24] R. Graybill and R. Melhem, Eds., *Power Aware Computing*. New York: Kluwer Academic/Plenum, 2002.
- [25] W. Liao, J. M. Basile, and L. He, "Leakage power modeling and reduction with data retention," in *Proc. IEEE/ACM ICCAD*, 2002, pp. 714–719.
- [26] T. Burd and R. Brodersen, Eds., *Energy Efficient Microprocessor Design*. Norwell, MA: Kluwer.
- [27] K. Nowka *et al.*, "A 32-bit PowerPC system-on-a-chip with support for dynamic voltage scaling and dynamic frequency scaling," *IEEE J. Solid-State Circuits*, vol. 37, no. 11, pp. 1441–1447, Nov. 2002.
- [28] J. Tschanz *et al.*, "Dynamic sleep transistor and body bias for active leakage power control of microprocessors," *IEEE J. Solid-State Circuits*, vol. 38, no. 11, pp. 1838–1845, Nov. 2003.
- [29] S. Kim, S. Kosonocky, and D. Knebel, "Understanding and minimizing ground bounce during mode transition of power gating structures," in *Proc. Int. Symp. Low Power Electronics and Design (ISLPED)*, Aug. 2003, pp. 22–25.
- [30] R. Erickson and D. Maksimovic, "High efficiency DC-DC converters for battery-operated systems with energy management," *Worldwide Wireless Communications, Annu. Rev. Telecommunications*, 1995.

- [31] A. Dancy and A. Chandrakasan, "Techniques for aggressive supply voltage scaling and efficient regulation," in *Proc. IEEE Custom Integrated Circuits Conf. (CICC)*, 1997, pp. 579–586.
- [32] J. Kim and M. A. Horowitz, "An efficient digital sliding controller for adaptive power-supply regulation," *IEEE J. Solid-State Circuits*, vol. 37, no. 5, pp. 639–647, May 2002.
- [33] A. Chandrakasan, W. J. Bowhill, and F. Fox, Eds., *Design of High-Performance Microprocessor Circuits*. Piscataway, NJ: IEEE Press, 2001.
- [34] R. Hegde and N. Shanbhag, "Toward achieving energy efficiency in presence of deep submicron noise," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 8, no. 4, pp. 379–391, Aug. 2000.



Bo Zhai (S'02) received the B.S. degree in microelectronics from Peking University, Beijing, China, in 2002 and the M.S. degree in electrical engineering from the University of Michigan, Ann Arbor, in 2004, where he is currently working toward the Ph.D. degree in electrical engineering.

He is a Research Assistant with the Advanced Computer Architecture Laboratory, University of Michigan, working with Prof. D. Blaauw. His research focuses on low-power VLSI design.



David Blaauw (M'93) received the B.S. degree in physics and computer science from Duke University, Durham, NC, in 1986, the M.S. degree in computer science from the University of Illinois, Urbana, in 1988, and the Ph.D. degree in computer science from the University of Illinois, Urbana, in 1991.

He was with IBM Corporation as a Development Staff Member until August 1993. From 1993 until August 2001, he was with Motorola, Inc., Austin, TX, where he was the Manager of the High Performance Design Technology Group. Since August 2001, he

has been a member of the faculty at the University of Michigan as an Associate Professor. His work has focused on VLSI design and computer-aided design with particular emphasis on circuit design and optimization for high-performance and low-power designs.

Dr. Blaauw was the Technical Program Chair and General Chair for the International Symposium on Low Power Electronic and Design in 1999 and 2000, respectively, and was the Technical Program Co-Chair and member of the Executive Committee the ACM/IEEE Design Automation Conference in 2000 and 2001.



Dennis Sylvester (S'95–M'00–SM'04) received the B.S. degree (*summa cum laude*) from the University of Michigan, Ann Arbor, in 1995, and the M.S. and Ph.D. degrees from the University of California, Berkeley (UC-Berkeley), in 1997 and 1999, respectively, all in electrical engineering.

He is now an Associate Professor of electrical engineering with the University of Michigan. He previously held research staff positions with the Advanced Technology Group of Synopsys, Mountain View, CA, and with Hewlett-Packard Laboratories, Palo Alto,

CA. He has published numerous papers along with one book and several book chapters in his field of research, which includes low-power circuit design and design automation techniques, design-for-manufacturability, and on-chip interconnect modeling. He also serves as a consultant and technical advisory board member for several electronic design automation firms in these areas.

Dr. Sylvester is a member of the Association for Computing Machinery (ACM), the American Society of Engineering Education, and Eta Kappa Nu. He was the recipient of a National Science Foundation CAREER Award, the 2000 Beatrice Winner Award at ISSCC, the 2004 IBM Faculty Award, and several Best Paper Awards and nominations. He was the recipient of the ACM SIGDA Outstanding New Faculty Award, the 1938E Award from the College of Engineering for teaching and mentoring, and the Henry Russel Award, which is the highest award given to faculty at the University of Michigan. His dissertation research was recognized with the 2000 David J. Sakris Memorial Prize as the most outstanding research in the Electrical Engineering and Computer Science Department of UC-Berkeley. He has served on the technical program committees of numerous design automation and circuit design conferences and was General Chair of the 2003 ACM/IEEE System-Level Interconnect Prediction (SLIP) Workshop and the 2005 ACM/IEEE Workshop on Timing Issues in the Synthesis and Specification of Digital Systems (TAU). He is currently an Associate Editor for the IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS. He also helped define the circuit and physical design roadmap of the International Technology Roadmap for Semiconductors (ITRS) U.S. Design Technology Working Group from 2001 to 2003.



Krisztian Flautner (M'00) received the Ph.D. degree in computer science and engineering from the University of Michigan, Ann Arbor.

He is the Director of Advanced Research at ARM Ltd., Cambridge, U.K., and the architect of ARM's Intelligent Energy Management technology. His research interests include high-performance, low-power processing platforms to support advanced software environments.

Dr. Flautner is a member of the Association of Computing Machinery.