Power, Thermal, and Reliability Modeling in Nanometer-Scale Microprocessors

Power is the source of the greatest problems facing microprocessor designers. Rapid power variation brings transient errors. High power densities bring high temperatures, harming reliability and increasing leakage power. The wages of power are bulky, short-lived batteries, huge heat sinks, large on-die capacitors, high server electric bills, and unreliable microprocessors. Optimizing power depends on accurate and efficient modeling that spans different disciplines and levels, from device physics, to numerical methods, to microarchitectural design.

David Brooks
Harvard University

Robert P. Dick
Russ Joseph
Northwestern University

Li Shang
Queen's University

System integration and performance requirements are dramatically increasing the power consumptions and power densities of high-performance microprocessors; some now consume 100 watts. High power consumption introduces challenges to various aspects of microprocessor and computer system design. It increases the cost of cooling and packaging design, reduces system reliability, complicates power supply circuitry design, and reduces battery life. Researchers have recently dedicated intensive effort to power-related design problems. Modeling is the essential first step toward design optimization. In this article, we explain the power, thermal, and reliability modeling problems and survey recent advances in their accurate and efficient analysis.

Implications of power consumption

Figure 1 illustrates the relationships among power consumption, temperature, reliability, and process variation. Shaded boxes indicate attributes that do not significantly influence others. Dashed boxes indicate the processes by which one attribute affects others. Dynamic and leakage power consumption are at the roots of many problems.

Rapid changes in dynamic power consumption (indicated by a solid box in Figure 1) can result in transient voltage fluctuations as a result of the distributed inductance and resistance in off-chip and on-chip microprocessor power delivery networks. These dI/dt effects, indicated by dashed boxes in Figure 1, can change logic combinational path delays, resulting in

0272-1732/07/$20.00 © 2007 IEEE Published by the IEEE Computer Society.

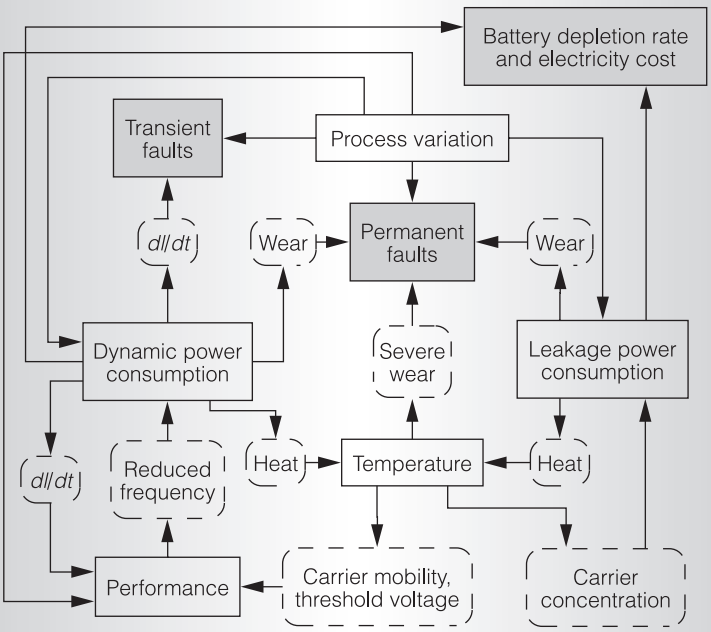

Figure 1. Power and its implications.

timing violations; this results in transient faults or requires decreasing the processor frequency. High dynamic power consumption implies high current density, resulting in accelerated wear and permanent faults due to processes such as electromigration. Dynamic power consumption produces heat, which increases microprocessor temperature. The effects of leakage power consumption are similar, with the exception of *dI*/*dt*-related problems. In general, leakage power variation is smaller than dynamic power variation. Both dynamic and leakage power consumption decrease battery life spans in portable devices and increase server electric bills.

Both dynamic and leakage power increase temperature. Temperature profiles depend on the temporal and spatial distribution of power as well as the microprocessor's cooling and packaging solutions. High temperature increases charge-carrier concentrations, resulting in increased subthreshold leakage power consumption. In addition, it decreases charge-carrier mobility, decreasing transistor and interconnect performance, and decreases threshold voltage, increasing transistor performance. Finally, temperature has a tremendous impact on the fault processes that lead to the majority of microprocessor permanent faults. Modeling and optimizing microprocessor thermal properties is thus essential for reliability, power consumption, and performance.

Process variation greatly influences the other power-related integrated circuit (IC) characteristics explained in this article. It influences transient fault rate via changes to critical timing paths; permanent fault rate via changes to numerous parameters such as wire and oxide dimensions; leakage power consumption via changes in dopant concentration; and dynamic power consumption.

As Figure 1 illustrates, the relationships among power, temperature, and reliability are complex. Controlling power, temperature, and reliability requires modeling and optimization at multiple design stages as well as runtime support.

Power modeling

Architectural power modeling before the register-transfer level (RTL) has gained prominence with the rising importance of energy-constrained portable devices and thermally constrained high-end machines. Understanding the power characteristics and ramifications of early-stage design decisions is essential, because architectural decisions can have a huge impact on overall power efficiency, and serious missteps made at the architecture level are often too difficult to correct at later stages in a design project.

Architectural power simulators are often tightly coupled with existing cycle-level architectural performance models. Microarchitectural utilization information, including bit-level activity in some cases, is combined with energy models for the power consumption of microarchitectural blocks under various usage scenarios—for example, power consumption with various amounts of activity after applying power- or clock-gating techniques. Collecting utilization information is not particularly challenging, so we focus on the more difficult task of developing power models for microarchitectural blocks. Most existing efforts in this area can be broadly characterized as either

analytical or empirical techniques for both dynamic and leakage power consumption.

Analytical power models

We can construct analytical models for dynamic power dissipation by identifying a microarchitectural block's key internal nodes and deriving analytical formulas for the capacitance of these nodes. Microarchitects have applied this technique to RAM and CAM structures such as caches, register files, buffers, and queues,[1–3] and to regular combinatorial logic such as decoders, select trees, and arbiters. (In contrast, empirical models are generally used to estimate the power of irregular combinatorial logic such as integer and floating-point ALUs.)

For example, most popular analytical models use the following equations to calculate SRAM array bitline energy:

$$E = C_{\text{bitline}} \times V_{\text{DD}} \times V_{\text{swing}} \quad (1)$$

$$C_{\text{bitline}} = (C_{\text{diff-access}} \times N) + C_{\text{diff-precharge}} + C_{\text{diff-column}} + C_{\text{wire}} \quad (2)$$

Here, C_{bitline} is the total stacked bitline capacitance and V_{swing} is the voltage swing of the bitline. The total bitline capacitance includes N stacked access transistor diffusion capacitance, precharge circuit capacitance, bitline wire capacitance, and column select circuit capacitance.

Analytical power models provide a fast, flexible method of estimating dynamic power consumption. However, the accuracy of these models suffers for several reasons:

- *Inaccurate capacitance estimates*. Node capacitance depends heavily on the actual design layout (including overlap and node-to-node coupling capacitances) and the supply voltage. Thus, it can be difficult to accurately estimate using simple analytical formulas.
- *Direct-path current*. Analytical models neglect direct-path current, which is difficult to estimate because it depends heavily on the circuit design. However, direct-path current can be appreciable in the precharge circuitry of certain memory designs.
- *Memory cell toggle power*. Many analytical models neglect the power to toggle memory cell bits when writing to memory. Although this power is usually small, it can be a fairly significant error source when there are periods of intensive writes and toggles to the memory bits.

Analytical power models for leakage power are also available, based on simple analytical formulas for subthreshold leakage and gate leakage.[4] For example, a common way to estimate leakage current is through the following equation:

$$I_{\text{leak}} = \beta \times e^{b(v_{\text{DD}} V_{\text{DD0}})} \times V_t^2 \times \left(1 - e^{\frac{-V_{\text{DD}}}{V_t}}\right) \times e^{\frac{-|V_{\text{TH}}| - V_{\text{off}}}{nV_t}} \quad (3)$$

Here, V_t is the thermal voltage; V_{TH} is the threshold voltage. This leakage model is an approximation; the leakage of a circuit is highly dependent on circuit topology and circuit state. Leakage power is also closely related to temperature, requiring tight coupling between leakage-power and temperature models.

Empirical power models

Empirical modeling approaches are based on circuit-level power analysis performed on structures from recently designed processors. SimplePower and PowerTimer use this approach.[5,6] PowerTimer's energy models are based on circuit-level power analysis of all microarchitectural structures in the Power4 microprocessor. Macro-level circuit power analysis, where several large macros combine to form a microarchitectural structure, determines power for microarchitectural blocks as a function of the activities of individual blocks. Empirical modeling approaches then combine these power estimates with utilization information from performance simulators.

Empirical approaches are accurate in formulating models for design macros that

tend to be reused in newer designs with relatively small changes. As such, they are quite useful for exploration of the design space of existing microarchitectures that are similar to the original analyzed design. However, the models rely on complete circuit simulation, which is generally slow. In addition, flexibility is a major problem, especially if a new process technology or a new microarchitecture is under investigation. Simple linear scaling of architectural parameters might not be accurate, and the use of newer technologies often brings even larger errors.

Microarchitects have widely used both analytical and empirical models to explore power-efficient architectures, and these models are a critical building block for temperature and reliability modeling and research. However, architects must understand the limitations of the modeling infrastructure before attempting a detailed quantitative evaluation. Later, we'll highlight some future directions in the area of power modeling that have the potential to address the accuracy and scalability concerns of existing modeling approaches.

Thermal modeling and analysis

IC thermal analysis requires the simulation of thermal conduction from an IC's power sources—that is, transistors and interconnects—through cooling package layers to the ambient environment. Modeling thermal conduction is analogous to modeling electrical conduction—with thermal conductance corresponding to electrical conductance, power dissipation corresponding to electrical current, heat capacity corresponding to electrical capacitance, and temperature corresponding to voltage.

The following equation governs thermal conduction in chips and cooling packages:

$$\rho c \frac{\partial T(\vec{r},t)}{\partial t} = \nabla \times [k(\vec{r}) \nabla T(\vec{r},t)] + p(\vec{r},t) \quad (4)$$

where ρ is the material density, c is the mass heat capacity, $T(\vec{r},t)$ and $k(\vec{r})$ are the temperature and thermal conductivity of the material at position $\vec{r}$ and time t, and $p(\vec{r},t)$ is the power density of the heat source. To solve this differential equation using numerical methods, we apply a finite-difference discretization method to decompose the chip and package into numerous discrete thermal elements, which may be of nonuniform sizes and shapes. Adjacent thermal elements interact via heat diffusion. Each element has a power dissipation, temperature, thermal capacitance, and thermal resistance to adjacent elements. Thus,

$$\mathbf{C}dT(t)/dt + \mathbf{A}T(t) = Pu(t) \quad (5)$$

where thermal capacitance matrix $\mathbf{C}$ is an $N \times N$ diagonal matrix, thermal conductivity matrix $\mathbf{A}$ is an $N \times N$ sparse matrix, $T(t)$ and P are $N \times 1$ temperature and power vectors, and $u(t)$ is the unit step function. In this tutorial, we assume that P is constant within an analysis interval but may vary between analysis intervals.

Steady-state thermal analysis

Steady-state thermal analysis characterizes temperature distribution when heat flow does not vary with time. For microprocessors, steady-state thermal analysis is sufficient for applications with stable power profiles or periodically changing power profiles that cycle quickly.

For steady-state thermal analysis, we drop the left term in Equation 5, expressing temperature variation as a function of time t. Thus,

$$\mathbf{A}T = P \rightarrow T = \mathbf{A}^{-1}P \quad (6)$$

Therefore, given thermal resistance matrix $\mathbf{A}$ and power vector P, the main task of steady-state thermal analysis is to invert matrix $\mathbf{A}$. The computational complexity of steady-state thermal analysis is thus determined by the size of matrix $\mathbf{A}$, which is in turn determined by the discretization granularity of the chip and the cooling package. Accurate IC thermal analysis requires fine-grained discretization. The size of matrix $\mathbf{A}$ is typically large. Matrices may be inverted using direct or iterative methods. Matrix inversion using direct methods, such as LU decomposition, is sufficient for small matrices, such as those associated with models

that model functional units with single thermal elements.[7] However, it is intractable for large matrices. Iterative methods permit the solution of larger problems. Recently, researchers have used multigrid relaxation for steady-state IC thermal analysis.[8,9] Each relaxation stage is responsible for eliminating errors in the frequency band corresponding to the stage's discretization granularity. Although multigrid methods often permit rapid convergence for linear systems, our recent analysis shows no significant performance improvement from using multigrid relaxation for steady-state IC thermal analysis, compared to an otherwise identical iterative solver.

Dynamic thermal analysis

Dynamic thermal analysis is used to characterize the runtime IC thermal profile when transient variations are significant. Researchers have applied time-domain and frequency-domain methods to this problem.

Time-domain methods use numerical integration to estimate the runtime IC thermal profile. The targeted time interval is partitioned into several time steps. If the size of each time step is small enough, we can accurately estimate IC thermal transition within each time step using the finite-difference temperature approximation function

$$T(T + h) = \frac{h[Pu(t) - AT(t)]}{\boldsymbol{C}} + T(t) \quad (7)$$

where h is the time step size. Numerical integration comprises a broad family of methods, which are typically classified on the basis of the approximation error bound. Well-known numerical integration methods include Euler's method, the trapezoidal method, Runge-Kutta methods, and Runge-Kutta-Fehlberg methods. Fourth-order Runge-Kutta methods are popular because of their accuracy and speed.

Frequency-domain methods estimate a runtime IC thermal profile using approximate analytical solutions, thereby avoiding repeated numerical integration. As Equation 8 shows, we use the Laplace transform to translate the dynamic thermal analysis problem into the frequency domain

$$\begin{aligned} \mathbf{C}[s\mathbf{T}(s) - \mathbf{T}(0^-)] &= \mathbf{AT}(s) + \mathbf{P}/s \\ \mathbf{T}(s) &= -\mathbf{A}^{-1}\left(\mathbf{I} - s\mathbf{CA}^{-1}\right)^{-1} \\ &\quad \times [\mathbf{P}/s + \mathbf{CT}(0^-)] \end{aligned} \quad (8)$$

Expanding $(\mathbf{I} - s\mathbf{CA}^{-1})^{-1}$ at $s = 0$:

$$\mathbf{T}(s) = -\mathbf{A}^{-1}\left\{\sum_{k=0}^{\infty}\left(s\mathbf{CA}^{-1}\right)^k \times [\mathbf{P}/s + \mathbf{CT}(0^-)]\right\} \quad (9)$$

Thus, IC runtime temperature can be accurately represented as an infinite series in the frequency domain. Because the long-time-scale impact of high-frequency components is negligible, truncating Equation 9 and ignoring high-frequency components yields an analytical approximation.

Time-domain and frequency-domain methods are each best suited to different modeling scenarios. Time-domain methods rely on numerical integration. Their running times increase with increasing time scale. In addition, because approximation error accumulates, the accuracy of time-domain methods generally degrades as time scales increase. Therefore, time-domain methods are most suitable for short-time-scale thermal analysis—that is, up to a few tens of milliseconds. Frequency-domain methods, on the other hand, approximate an IC runtime thermal profile using analytical solutions, avoiding the need for numerical integration. However, the one-time computational cost of deriving the analytical approximation is high. Moreover, frequency-domain methods ignore high-frequency components, introducing short-time-scale errors. Therefore, frequency-domain methods are more appropriate for long-time-scale thermal analysis—that is, more than a few milliseconds.

Adaptive thermal analysis

The major challenges of numerical IC thermal analysis are high computational complexity and memory usage. High

modeling accuracy requires fine-grained discretization, resulting in numerous grid elements. For dynamic thermal analysis using time-domain methods, higher modeling accuracy requires fine spatial and temporal discretization granularities, increasing the computational overhead and memory usage. For dynamic thermal analysis using frequency-domain methods, deriving analytical approximations involves computation- and memory-intensive numerical operations, such as the inversion and multiplication of large matrices. High thermal element count may hinder or prevent the application of frequency-domain methods to model complicated IC thermal setups.

Recent progress on adaptive numerical modeling methods tackles the high computation complexity and memory usage of steady-state, time-domain, and frequency-domain thermal analysis.[9]

Spatial adaptation. Chip and package thermal gradients exhibit significant spatial variation because of the heterogeneity of thermal conductivity and heat capacity in different chip and cooling package materials, as well as variation in power profiles. The wide distribution of temperature differences across the chip and the cooling package motivates the design of a thermal-gradient-sensitive, adaptive spatial-discretization refinement technique that automatically adjusts the spatial-partitioning resolution to maximize thermal modeling efficiency and guarantee modeling accuracy. In this technique, temperature difference constraints govern the spatial discretization process. Iterative refinement takes place in a hierarchical fashion. Regions with high-magnitude gradients are recursively partitioned until elements are isothermal and the temperature profile converges. This method uses fine-grained discretization only when necessary for accuracy, thereby improving the efficiency of steady-state and time-domain dynamic thermal analysis, and enabling the use of frequency-domain methods in modeling complicated chips and packages.

Temporal adaptation. There are two categories of time-domain methods: nonadaptive and adaptive. Nonadaptive methods use the same time-step size throughout analysis. Therefore, their performance is bounded by the smallest time step required by any thermal element at any time. Temporally adaptive methods, in contrast, can improve performance substantially without loss of accuracy by adapting time-step sizes during dynamic analysis.

Temporally adaptive methods, in turn, fall into two categories: synchronous and asynchronous time-marching methods. In synchronous time-marching methods, such as adaptive fourth-order Runge-Kutta methods, all thermal elements advance in time in lockstep. However, before each step, these methods adjust the step size to the minimal size required for accuracy by any thermal element. In contrast, asynchronous time-marching methods permit different elements to advance forward in time at different rates.[9] This method exploits temporal and spatial differences in temperature variation within chip and cooling packages. Each thermal element uses the largest safe step size for its position and time, instead of being forced to use the same step size as all other elements in the chip and package. By allowing elements to progress in time asynchronously, these methods can dramatically accelerate thermal analysis without losing accuracy.

Temperature-dependent leakage power analysis

Technology scaling is increasing the proportion of power consumption due to leakage; accurate leakage analysis is now important. IC leakage-power consumption is a strong function of temperature. Many researchers have used iterative temperature-dependent leakage analysis for accurate leakage estimation. This approach was developed based on the following observations. First, subthreshold leakage power consumption increases superlinearly with temperature. This has led to the incorrect assumption that highly accurate thermal modeling embedded within the power analysis flow is necessary to accurately determine temperature-dependent leakage-power consumption. Second, leakage variation per unit temperature change is less than 1—that is, $dP_{\text{leakage}}/dT < 1$. An iterative-based thermal-leakage

flow can thus guarantee convergence and correctness. The major challenge of existing thermal-leakage analysis flow is high computational complexity, which significantly increases the running times of both IC thermal analysis and power analysis.

A recent study shows that highly efficient and accurate temperature-dependent leakage power analysis is possible using coarse-grained thermal modeling.[10] Leakage power depends mainly on IC thermal profile and circuit design style. Despite the nonlinear dependence of leakage power on temperature, within the operating temperature ranges of real ICs, using a linear leakage model for individual functional units results in less than a 1 percent error in leakage estimation and permits more than four orders of magnitude reduction in analysis time relative to an approach relying on detailed accurate thermal analysis.

Power- and temperature-related microprocessor reliability

As Figure 1 shows, power and its by-products are now a great threat to the reliability of microprocessors. High power consumption translates to high temperatures: a recipe for permanent faults due to electromigration and other failure processes. Rapidly changing power consumption of components supplied by resistive and inductive power delivery networks results in transient voltage fluctuations and therefore transient faults. Soft errors are also becoming an increasingly important source of IC reliability problems.[11] However, the focus of this article is power consumption, and most soft errors are orthogonal to power consumption.

Modeling permanent power- and temperature-related faults

For IC permanent faults, major failure mechanisms include electromigration, thermal cycling, time-dependent dielectric breakdown, and stress migration.[12,13]

Electromigration refers to the gradual displacement of the atoms in metal wires caused by electrical current. It leads to voids and hillocks within metal wires that result in open- and short-circuit failures. The following equation gives the mean time to failure (MTTF) due to electromigration:

$$MTTF_{\mathrm{EM}} = \frac{A_{\mathrm{EM}}}{J^n} e^{\frac{E_{\mathrm{aEM}}}{\kappa T}} \quad (10)$$

where A_{EM} is a constant determined by the physical characteristics of the metal interconnect, J is the current density, E_{aEM} is the activation energy of electromigration, n is an empirically determined constant, and T is the temperature.

Thermal cycling refers to IC fatigue failures caused by thermal mismatch deformation. In the chip and package, adjacent material layers such as copper and low-κ dielectric have different coefficients of thermal expansion. As a result, runtime thermal variation causes fatigue deformation, leading to failures. The following equation gives the MTTF due to thermal cycling:

$$MTTF_{\mathrm{TC}} = \frac{A_{\mathrm{TC}}}{\left(T_{\mathrm{average}} - T_{\mathrm{ambient}}\right)^{\mathrm{q}}} \quad (11)$$

where A_{TC} is a constant coefficient, T_{average} is the chip average runtime temperature, T_{ambient} is the ambient temperature, and q is the Coffin-Manson exponent constant.

Time-dependent dielectric breakdown refers to deterioration of the gate dielectric layer. This effect is a strong function of temperature and is becoming increasingly prominent with the reduction of gate-oxide dielectric thickness and nonideal supply voltage reduction. The following equation gives the MTTF due to time-dependent dielectric breakdown:

$$MTTF_{\mathrm{TDDB}} = A_{\mathrm{TDDB}} \left(\frac{1}{V}\right)^{(a-bT)} \times e^{\frac{A + B/C + CT}{\kappa T}} \quad (12)$$

where A_{TDDB} is a constant; V is the supply voltage; and a, b, A, B, and C are fitting parameters.

Stress migration refers to the mass transportation of metal atoms in metal wires due to mechanical stress caused by thermal mismatch among metal and dielectric materials. The following equation gives the

MTTF due to stress migration:

$$MTTF_{SM} = A_{SM}|T_0 - T|^{-n} \times e^{\frac{E_{aSM}}{\kappa T}} \quad (13)$$

where A_{SM} is a constant, T_0 is the metal deposition temperature during fabrication, T is the runtime temperature of the metal layer, n is an empirically determined constant, and E_{aSM} is the activation energy for stress migration.

Equations 10 to 13 indicate that IC fault processes are strongly influenced by temperature, and in many cases are exponentially dependent on it. As a result, it appears that detailed and accurate thermal modeling is necessary for reliability estimation. Some of these fault processes are also accelerated by increases in other parameters related to power, such as current density and voltage.

The most dangerous fault processes in microprocessors accelerate as a result of wear, and this complicates reliability modeling and analysis. Several researchers have assumed that microprocessor fault processes are Poisson processes and have used exponential distributions to model them. Exponential distributions are mathematically convenient because they permit the addition of the rates of different fault processes operating on different components to determine the failure rate of the entire microprocessor. However, they do not model wear, which is generally required for accurate reliability modeling.

The log-normal distribution better models prominent microprocessor fault processes because it models the increase in failure rate with increasing wear. However, this property complicates system-level modeling. There is no straightforward method of deriving a closed-form expression for the failure rate of a microprocessor composed of numerous components for which log-normal fault process models are used. Therefore, some microprocessor reliability estimation work assumes exponential fault processes, which may be inaccurate; other work uses Monte Carlo simulation[14] or techniques based on statistical curve fitting, each of which may be slow. At present, efficient and accurate system-level lifetime reliability estimation is a goal that remains just slightly out of grasp, but toward which research is rapidly progressing.

Modeling transient power-related faults

In recent years, there has been increased interest in microarchitectural support for an increasingly important reliability concern, power supply noise. The circuits on high-performance chips place stringent demands on the power delivery infrastructure responsible for satisfying current demands and maintaining reference voltages. Power supply integrity is essential because even minor deviations in power or ground reference levels can introduce noise or delay into critical signal transitions, leading to an unrecoverable error. Noise reduction is difficult due to parasitic inductance in the power delivery system. Whenever the processor current demands vary, the transient load causes voltage fluctuations given by the well known equation $V = L(dI/dt)$, where L is the value of the inductor and dI/dt is the rate of change of the current. This relationship gives inductive noise its informal name, the dI/dt problem.

Interest in microarchitectural support has grown because of the increasing difficulty of mitigating noise through conventional avenues.[15–17] Traditionally, microarchitects have addressed inductive noise by reducing effective inductance or by adding capacitance on die or in the package to dampen out the noise. However, the voltage noise tolerance of future processors will decrease at a rate that outstrips our ability to remove the parasitics or add capacitance. In particular, current processor designs use considerable on-die capacitance; they might devote as much as 15 to 20 percent of die area to decoupling capacitors. Although on-die capacitors are very effective at dampening inductive noise, they consume leakage power and die area because they are implemented as nonswitching transistors with large gate capacitances. Furthermore, in future designs, absolute voltage tolerance will decrease as the supply voltage scales, and load transients will increase as total power increases. Consequently, the rate of change for load current (dI/dt) will increase,

and the allowable variation in supply voltage will shrink.

Microarchitectural techniques seek to limit inductive noise by controlling the rate at which load current changes. They do this essentially by smoothing out or altering the processor's current profile to eliminate problematic load transients. These techniques can reduce the burden on traditional circuit and package solutions for inductive noise.

Impact of frequency. The severity of inductive noise depends not only on the magnitude of transient current, but also on the frequency range over which it changes. Conventional power supply systems consist of a complex network of die, package, and motherboard capacitances, inductances, and resistances. Specific current variation patterns excite these elements to different degrees. Mid-frequency noise in the range of 50 to 200 MHz and high-frequency noise near the processor clock rate have achieved the most attention in the literature.

Mid-frequency noise is associated with package inductance that reacts with die and package capacitors whenever the processor current varies within the problematic frequency range. Variations at these critical frequencies are problematic because they allow resonance to build in the power delivery network, producing violent swings in supply voltage. High-frequency noise produces sharp, localized fluctuations in on-chip supply voltage. These high-frequency noise patterns arise when processor execution resources have abrupt changes in current demand that cannot be adequately serviced by neighboring on-chip or in-package capacitors and solder bump connections to the off-chip network. We focus here on mid-frequency noise because it provides the greatest opportunity for architectural solutions.

Mid-frequency models. Researchers in the electronic packaging community have characterized the overall response of the power delivery system as a second-order linear system,[18] which we can essentially represent as a single lumped resistance-inductance-capacitance (RLC) underdamped network.[19]

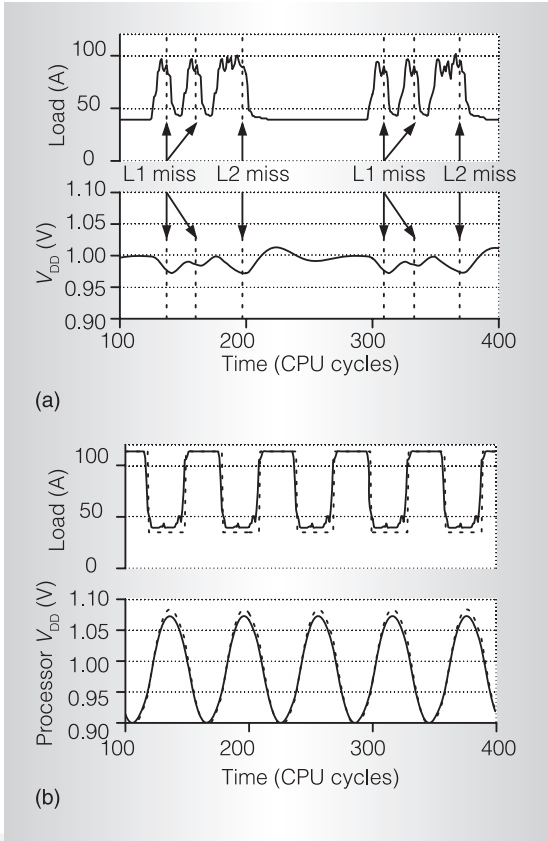

Figure 2. Noise impact of different current consumption patterns: an execution sequence from 164.gzip, which does not hit a resonant frequency (a), and a trace from a *dI/dt* microbenchmark, which features severe ILP variation and mimics a resonant pulse (b).

In this simple model, the R, L, and C values do not correspond to any specific physical elements in the real system, rather they are effective values that consider the composite effects of all elements in the network.

The dominant characteristic of mid-frequency models is resonance. The most problematic load profile is a pulse pattern at the natural frequency of the RLC network, namely $f_c = 1/2\pi(LC)^{1/2}$. In current processors this translates to alternating patterns of on/off activity on the order of many tens to about one hundred processor cycles. Figure 2 illustrates this phenomenon, showing the impact of a nonresonant current load and a resonant current pulse train, which has a far more significant noise impact. Architecture-level phenomena such as cache misses and variations in instruc-

tion-level parallelism (ILP) can produce bursty current behavior at this frequency range if they are sequenced in an unfortunate manner.

Microarchitectural simulators with support for cycle-by-cycle power estimates can model the impact of mid-frequency inductive noise. In essence, the processor's total current load can be assumed to be its instantaneous power divided by the ideal supply voltage. In an actual circuit, a real supply voltage droop would cause current draw to decrease, making division by a constant value a conservative estimate. However, dividing by varying voltage would be unnecessarily pessimistic. This instantaneous current can be related to varying supply voltage through filter techniques that use convolution to relate a history of current consumption to voltage,[15,16] or through circuit simulation that directly models the RLC network.[20] In either case, models can be extended to capture higher-order power delivery system effects through use of a more complex convolution filter or a detailed circuit model.

Modeling process variation

Parameter variation is an unavoidable consequence of continued technology scaling. The impact of random and systematic variation on physical factors such as gate length and interconnect spacing will have a profound impact on performance and power consumption. In current designs, foundry-induced physical deviations already produce significant die-to-die variation. In particular, industry data for a high-performance processor in 180-nm technology shows that individual dies produced with the same fabrication equipment can have as much as a 30 percent die-to-die frequency variation and a 20× leakage power variation.[21] The *International Technology Roadmap for Semiconductors* (http://www.itrs.net) predicts that manufacturing variability will have an increasing prominence in future designs.

Current architecture-level models for process variation have focused on deviations in effective threshold voltage (v_{th}) and attempt to relate physical uncertainty in this highly influential parameter to performance and power. This involves combining analytical and empirical transistor models, circuit structure characteristics, and statistical principles that relate physical uncertainty to architecturally visible characteristics.

Variation under statistical distributions

Physical variation at the transistor level can be modeled according to statistical relationships. Researchers and microarchitects often use a combination of analytical models and Monte Carlo simulations to give modeled transistor parameters the appropriate statistical distributions. Dopant ion concentration is normally modeled as a Gaussian distribution. Under this model, there is no correlation between any pair of transistors in the design. On the other hand, gate length has strong spatial-correlation properties.[22] Because of localized imperfections in lithography, neighboring transistors are likely to have similar gate-length deviations. As the distance between transistors increases, correlation decreases linearly.[22] This is most frequently analyzed using Monte Carlo simulations. The die is modeled as a collection of grid sections. All transistors within a section are assumed to have identical parameter variation. Researchers have used several methods, including hierarchical methods[23] and convolution kernels,[24] to assign parameter deviations to neighboring blocks. An alternative to using Monte Carlo simulations is to model systematic components of leakage variation through an empirically derived gate-length deviation model.[25] This approach is useful for average case studies but cannot capture probabilistic aspects of leakage variation.

Relating V_{TH} variation to microarchitecture

Performance and power variations at the transistor level are dominated by effective threshold voltage, which is in turn determined by drawn gate length and dopant concentration.[26] Process variations in either of these physical factors will cause deviations in V_{TH} and lead to conflicting impacts on performance and leakage power.[21] Lower effective threshold voltages caused by short gate lengths or over-doping decrease delay in a roughly linear fashion but lead to an exponential increase in subthreshold leakage

current. Deviations that increase v_t reduce leakage with the penalty of increased delay. Although v_t variation can also affect dynamic power, its impact on leakage power dominates and has been the primary focus of power models for parameter variation.

At the gate level, we can model the effects of varying threshold voltage analytically or empirically. Analytical models use device models to relate physical variation in critical parameters, namely gate length and concentration of dopant ions to calculate effective threshold voltage. Based on an effective threshold voltage, we can compute the leakage current using Equation 3.

Likewise, we can use the well-known Alpha power law to model the effect of threshold voltage on transistor delay:

$$td \propto V_{DD}/\left(V_{DD} - V_{TH}{}^{\propto}\right) \qquad (14)$$

Together, these equations govern the relative impact of process variation at the transistor level and form the basis for analytical models. For leakage, existing architectural models provide baseline leakage power on a component basis. The analytical models project how leakage scales. For performance, the impact on V_{TH} can also be related to architectural structure. Within each pipeline stage, the number of critical paths determines the minimum frequency of the stage.[27] In essence, a large number of critical paths increases the probability that one of the paths will be slow and hence decreases the clock frequency. Architectural models for critical path count can be derived from circuit structure.[25] In particular, we can use hardware description language (HDL) descriptions of processor designs to count critical paths in each stage.[28] Spice-based empirical models offer another alternative. Detailed simulation on either complete circuits or representative sections can determine the impact on power and performance.

Continued technology scaling and emerging directions in design will change the problems of power, thermal, reliability, and process-variation modeling. In the near future, several problems are likely to arise in these modeling domains.

In the *power-modeling* realm, the move to nanoscale technologies will break down traditional circuit-scaling theory, exacerbating the difficulties associated with analytical dynamic- and leakage-power modeling. Furthermore, asymmetric and heterogeneous chip-level multiprocessors (CMPs) will lead to an increased diversity in microarchitectures and accelerator cores, limiting the utility of empirical modeling approaches. As more designers explore many-core and heterogeneous core systems, power-modeling tools will need to incorporate better estimates for interconnect and combinatorial logic structures. To address these challenges, future power-modeling approaches might rely on a mixture of analytical and empirical modeling techniques, to leverage the advantages of both.

Increasing IC integration and power density will bring new *thermal-modeling* challenges. Moreover, the changes that new fabrication processes, materials, and cooling solutions bring to thermal models will require advances in analysis to permit efficiency and speed. For instance, the thermal properties of stacked 3D ICs will differ greatly from conventional 2D ICs, and high power density remains one of the major 3D integration challenges. Increasing microprocessor power density will require novel cooling solutions such as microchannel cooling or nanotube thermal vias, requiring multiresolution modeling down to the micrometer or nanometer scale.

Most system-level *reliability models* are derived using statistical techniques based on empirical models. These statistical approximations can benefit greatly from calibration and validation. Technology scaling further increases the importance and challenges of reliability modeling. Reduced feature size will increase the vulnerability of individual devices and interconnects. This will require models that capture the system-level impact of nanoscale components while permitting efficient analysis. However, increasing integration density complicates the development of accurate microarchitectural reliability models. Future CMP architectures also present additional

opportunities for per-core power-down and dynamic voltage and frequency scaling, leading to additional *dI/dt* noise challenges.

To effectively model *process variations*, architects will need to improve the accessibility of their models and adjust them to keep pace with manufacturing trends. In addition, architects will have to develop unified models that better capture couplings between power, performance, and reliability aspects of variation. Although Monte Carlo variation analysis is relatively easy to apply for power-performance studies, it does not offer the same intuition and insights that analytic models can. Mathematical models that are easily parameterizable but can capture probabilistic characteristics such as mean, variance, and skew could be extremely beneficial.

Power-related design challenges have become a critically important topic for computer architects and system designers. Microarchitectural design requires detailed models of power-related phenomena. As advanced design techniques and fabrication process changes reveal new power-related phenomena, power, thermal, reliability, and process-variation models will require ongoing improvement. MICRO

Acknowledgments

This work was supported in part by the US National Science Foundation under awards CCF-0048313 and CNS-0347941, in part by Intel, in part by IBM, and in part by the Natural Sciences and Engineering Research Council of Canada under Discovery Grant 388694-01.

References

1. P. Shivakumar and N.P. Jouppi, *CACTI 3.0: An Integrated Cache Timing, Power, and Area Model,* tech. report, Western Research Lab, 2001.
2. D. Brooks, V. Tiwari, and M. Martonosi, ''Wattch: A Framework for Architectural-Level Power Analysis and Optimizations,'' *Proc. Int'l Symp. Computer Architecture* (ISCA 00), IEEE CS Press, 2000, pp. 83-94.
3. N.S. Kim et al., ''Microarchitectural Power Modeling Techniques for Deep Sub-Micron Microprocessors,'' *Proc. Int'l Symp. Low-Power Electronics and Design* (ISLPED 04), IEEE CS Press, 2004, pp. 212-217.
4. J.A. Butts and G.S. Sohi, ''A Static Power Model for Architects,'' *Proc. Int'l Symp. Microarchitecture* (MICRO 00), IEEE CS Press, 2000, pp. 191-201.
5. W. Ye et al., ''The Design and Use of SimplePower: A Cycle-Accurate Energy Estimation Tool,'' *Proc. 37th Design Automation Conf.* (DAC 00), ACM Press, 2000, pp. 340-345.
6. D. Brooks et al., ''New Methodology for Early-Stage, Microarchitecture-Level Power-Performance Analysis of Microprocessors,'' *IBM J. Research and Development*, vol. 47, no. 5-6, 2003, pp. 653-670.
7. K. Skadron et al., ''Temperature-Aware Microarchitecture,'' *Proc. Int'l Symp. Computer Architecture* (ISCA 03), IEEE CS Press, 2003, pp. 2-13.
8. P. Li et al., ''Efficient Full-Chip Thermal Modeling and Analysis,'' *Proc. Int'l Conf. Computer-Aided Design* (ICCAD 04), IEEE CS Press, 2004, pp. 319-326.
9. Y. Yang et al., ''Adaptive Multi-Domain Thermal Modeling and Analysis for Integrated Circuit Synthesis and Design,'' *Proc. Int'l Conf. Computer-Aided Design* (ICCAD 06), IEEE CS Press, 2006, pp. 575-582.
10. Y. Liu et al., ''Accurate Temperature-Dependent Integrated Circuit Leakage Power Estimation Is Easy,'' *Proc. Design Automation and Test in Europe Conf.* (DATE 07), IEEE CS Press, 2007, pp. 204-209.
11. S.S. Mukherjee et al., ''A Systematic Methodology to Compute the Architectural Vulnerability Factors for a High-Performance Microprocessor,'' *Proc. Int'l Symp. Microarchitecture* (MICRO 03), IEEE CS Press, 2003, pp. 29-40.
12. *Failure Mechanisms and Models for Semiconductor Devices,* publication JEP 122-B, JEDEC, 2003.
13. J. Srinivasan et al., ''Exploiting Structural Duplication for Lifetime Reliability Enhancement,'' *Proc. Int'l Symp. Computer Architecture* (ISCA 05), IEEE CS Press, 2005, pp. 520-531.
14. J. Srinivasan et al., ''The Impact of Technology Scaling on Lifetime Reliability,'' *Proc. Int'l Conf. Dependable Systems and Networks* (DSN 04), IEEE Press, 2004, pp. 177-186.

15. W. El-Essawy and D. Albonesi, ''Mitigating Inductive Noise in SMT Processors,'' *Int'l Symp. Low-Power Electronics and Design* (ISLPED 04), IEEE CS Press, 2004, pp. 332-337.
16. R. Joseph, D. Brooks, and M. Martonosi, ''Control Techniques to Eliminate Voltage Emergencies in High Performance Processors,'' *Proc. 9th Int'l Symp. High-Performance Computer Architecture* (HPCA 03), IEEE CS Press, 2003, pp. 79-90.
17. M.D. Powell and T.N. Vijaykumar, ''Exploiting Resonant Behavior to Reduce Inductive Noise,'' *Proc. 31st Int'l Symp. Computer Architecture* (ISCA 04), IEEE CS Press, 2004, pp. 288-299.
18. D.J. Herrell and B. Beker, ''Modeling of Power Distribution Systems for High-Performance Microprocessors,'' *IEEE Trans. Advanced Packaging*, vol. 22, no. 3, Aug. 1999, pp. 240-248.
19. I. Kantorovich et al., ''Measurement of Low Impedance On Chip Power Supply Loop,'' *IEEE Trans. Advanced Packaging*, vol. 27, no. 1, Feb. 2004, pp. 10-14.
20. M.D. Powell and T.N. Vijaykumar, ''Pipeline Muffling and A Priori Current Ramping: Architectural Techniques to Reduce High-Frequency Inductive Noise,'' *Proc. Int'l Symp. Low-Power Electronics and Design* (ISLPED 03), IEEE Press, 2003, pp. 223-228.
21. S. Borkar et al., ''Parameter Variations and Impact on Circuits and Microarchitecture,'' *Proc. 40th Design Automation Conf.* (DAC 03), ACM Press, 2003, p. 338.
22. P. Friedberg et al., ''Modeling Within-Die Spatial Correlation Effects for Process-Design Co-optimization,'' *Proc. 6th Int'l Symp. Quality Electronic Design* (ISQED 05), IEEE Press, 2005, pp. 516-521.
23. A. Agarwal et al., ''Path-Based Statistical Timing Analysis Using Bounds and Selective Enumeration,'' *Proc. Int'l Workshop Timing Issues in the Specification and Synthesis of Digital Systems* (TAU 02), ACM Press, 2002, pp. 16-21.
24. K. Meng et al., ''Modeling and Characterizing Power Variability in Multicore Architectures,'' *IEEE Symp. Analysis of Software and Systems* (ISPASS 07), IEEE Press, 2007, pp. 146-153.
25. E. Humenay, D. Tarjan, and K. Skadron, ''Impact of Parameter Variations on Multicore Chips,'' *Proc. Workshop Architectural Support for Gigascale Integration* (ASGI 06), 2006, http://www.cs.virginia.edu/~skadron/Papers/PV_asgi06.pdf.
26. S.R. Nassif, ''Modeling and Forecasting of Manufacturing Variations,'' *Proc. 5th Int'l Workshop Statistical Metrology*, IEEE Press, 2000, pp. 2-10.
27. K.A. Bowman, S.G. Duvall, and J.D. Meindl, ''Impact of Die-to-Die and Within-Die Parameter Fluctuations on the Maximum Clock Frequency Distribution for Gigascale Integration,'' *IEEE J. Solid State Circuits*, vol. 37, no. 2, Feb. 2002, pp. 183-190.
28. B.F. Romanescu, S. Ozev, and D.J. Sorin, ''Quantifying the Impact of Process Variability on Microprocessor Behavior,'' *Proc. Workshop Architectural Reliability* (WAR 06), 2006, http://www.ee.duke.edu/~sorin/papers/war06_variability.pdf.

David Brooks is an associate professor of computer science at Harvard University. His research interests include power and thermal-efficient hardware-software system design, and approaches to cope with reliability and variability in next-generation computing systems. He has a BS from the University of Southern California, and an MA and a PhD from Princeton University, all in electrical engineering.

Robert P. Dick is an assistant professor of electrical engineering and computer science at Northwestern University. His research interests span a broad area in computer engineering, with a recent focus on embedded systems and temperature-aware and low-power design. He has a BS computer engineering from Clarkson University and a PhD in electrical engineering from Princeton University.

Russ Joseph is an assistant professor of electrical engineering and computer science at Northwestern University. His research interests focus primarily on power- and reliability-aware computer architecture. He has a BS in electrical and computer

engineering, with an additional major in computer science, from Carnegie Mellon University and a PhD in electrical engineering from Princeton University.

Li Shang is an assistant professor in the Department of Electrical and Computer Engineering at Queen's University, Canada. His research interests include computer architecture, interconnection networks, design automation for embedded systems, and design for nanotechnologies. He has a BE from Tsinghua University and a PhD from Princeton University, both in electrical engineering.

Direct questions and comments about this article to Robert Dick, L477 Technological Institute, EECS Department, Northwestern University, 2145 Sheridan Rd., Evanston, IL 60208; dickrp@northwestern.edu.