ENCYCLOPEDIA OF COMPUTER SCIENCE AND TECHNOLOGY

EXECUTIVE EDITORS

Allen Kent *James G. Williams*

UNIVERSITY OF PITTSBURGH
PITTSBURGH, PENNSYLVANIA

ADMINISTRATIVE EDITOR

Carolyn M. Hall

ARLINGTON, TEXAS

VOLUME 36

SUPPLEMENT 21

MARCEL DEKKER, INC. NEW YORK · BASEL · HONG KONG

MARCEL DEKKER, INC.
270 Madison Avenue, New York, New York 10016

LIBRARY OF CONGRESS CATALOG CARD NUMBER: 74-29436
ISBN: 0-8247-2289-2

Current Printing (last digit)
10 9 8 7 6 5 4 3 2 1

PRINTED IN UNITED STATES OF AMERICA

DESIGN TECHNOLOGIES FOR LOW-POWER VLSI

MOTIVATION

In the past, the major concerns of the very-large-scale integration (VLSI) designer were area, performance, cost, and reliability; power consideration was mostly of only secondary importance. In recent years, however, this has begun to change and, increasingly, power is being given comparable weight to area and speed considerations. Several factors have contributed to this trend. Perhaps the primary driving factor has been the remarkable success and growth of the class of personal computing devices (portable desktops, audio- and video-based multimedia products) and wireless communications systems (personal digital assistants and personal communicators) which demand high-speed computation and complex functionality with low-power consumption.

In these applications, average power consumption is a critical design concern. The projected power budget for a battery-powered, A4 format, portable multimedia terminal, when implemented using off-the-shelf components not optimized for low-power operation, is about 40 W. With advanced nickel–metal–hydride (secondary) battery technologies offering around 65 W hr/kg (1), this terminal would require an unacceptable 6 kg of batteries for 10 hr of operation between recharges. Even with new battery technologies such as rechargeable lithium ion or lithium polymer cells, it is anticipated that the expected battery lifetime will increase to about 90–110 W hr/kg over the next 5 years (1) which still leads to an unacceptable 3.6–4.4 kg of battery cells. In the absence of low-power design techniques, current and future portable devices will suffer from either very short battery life or very heavy battery pack.

There also exists strong pressure for producers of high-end products to reduce their power consumption. Contemporary performance optimized microprocessors dissipate as much as 15–30 W at 100–200 MHz clock rates (2)! In the future, it can be extrapolated that a 10-cm^2 microprocessor, clocked at 500 MHz (which is a not too aggressive estimate for the next decade) would consume about 300 W. The cost associated with packaging and cooling such devices is prohibitive. Because core power consumption must be dissipated through the packaging, increasingly expensive packaging and cooling strategies are required as chip power consumption increases. Consequently, there is a clear financial advantage to reducing the power consumed in high-performance systems.

In addition to cost, there is the issue of reliability. High-power systems often run hot, and high temperature tends to exacerbate several silicon failure mechanisms. Every 10°C increase in operating temperature roughly doubles a component's failure rate (3). In this context, peak power (maximum possible power dissipation) is a critical design factor, as it determines the thermal and electrical limits of

designs, impacts the system cost, size and weight, dictates specific battery type, component and system packaging, and heat sinks, and aggravates the resistive and inductive voltage-drop problems. It is therefore essential to have the peak power under control.

Another crucial driving factor is that excessive power consumption is becoming the limiting factor in integrating more transistors on a single chip or on a multiple-chip module. Unless power consumption is dramatically reduced, the resulting heat will limit the feasible packing and performance of VLSI circuits and systems.

From the environmental viewpoint, the smaller the power dissipation of electronic systems, the lower the heat pumped into the rooms, the lower the electricity consumed and, hence, the lower the impact on global environment, the less the office noise (e.g., due to elimination of a fan from the desktop), and the less stringent the environment/office power delivery or heat-removal requirements.

The motivations for reducing power consumption differ from application to application. In the class of micropowered battery-operated, portable applications, such as cellular phones and personal digital assistants, the goal is to keep the battery lifetime and weight reasonable and the packaging cost low. Power levels below 1–2 W, for instance, enable the use of inexpensive plastic packages. For high-performance, portable computers, such as laptop and notebook computers, the goal is to reduce the power dissipation of the electronics portion of the system to a point which is about half of the total power dissipation (including that of display and hard disk). Finally, for high-performance, nonbattery operated systems, such as workstations, desk-top computers, and multimedia digital signal processors, the overall goal of power minimization is to reduce system cost (cooling, packaging, and energy bill) while ensuring long-term device reliability. These different requirements impact how power optimization is addressed and how much the designer is willing to sacrifice in cost or performance to obtain lower power dissipation.

The next question is to determine the objective function to minimize during low-power design. The answer varies from one application domain to the next. If extending the battery life is the only concern, then the energy (i.e., the power-delay product) should be minimized. In this case, the battery consumption is minimized even though an operation may take a very long time. On the other hand, if both the battery life and the circuit delay are important, then the energy-delay product must be minimized (4). In this case, one can alternatively minimize the energy/delay ratio (i.e., the power) subject to a delay constraint. In most design scenarios, the circuit delay is set based on system-level considerations, and hence during circuit optimization, one minimizes power under user-specified timing constraints.

SOURCES OF POWER DISSIPATION

Power dissipation in Complementary Metal-Oxide-Silicon (CMOS) circuits is caused by three sources: (1) The leakage current which is primarily determined by the fabrication technology, consists of reverse bias current in the parasitic diodes formed between source and drain diffusions and the bulk region in a MOS transistor as well as the subthreshold current that arises from the inversion charge that exists at the gate voltages below the threshold voltage; (2) the short-circuit (rush-through)

current which is due to the DC path between the supply rails during output transitions; and (3) the charging and discharging of capacitive loads during logic changes.

The diode leakage occurs when a transistor is turned off and another active transistor charges up or down the drain with respect to the first transistor's bulk potential. The resulting current is proportional to the area of the drain diffusion and the leakage current density. The diode leakage is typically 1 pA for a 1-μm minimum feature size. The subthreshold leakage current for long-channel devices increases linearly with the ratio of the channel width over channel length and decreases exponentially with V_{GS}-V_t, where V_{GS} is the gate bias and V_t is the threshold voltage. Several hundred millivolts of "off bias" (say, 300–400 mV) typically reduces the subthreshold current to negligible values. With reduced power supply and device threshold voltages, the subthreshold current will, however, become more pronounced. In addition, at short channel lengths, the subthreshold current also becomes exponentially dependent on drain voltage V_{DS} instead of being independent of V_{DS}, (see Ref. 5 for a recent analysis). The subthreshold current will remain 10^2–10^5 times smaller than the "on current," even at submicron device sizes.

The short-circuit (crowbar current) power consumption for an inverter gate is proportional to the gain of the inverter, the cubic power of supply voltage minus device threshold, the input rise/fall time, and the operating frequency (6). The maximum short-circuit current flows when there is no load; this current decreases with the load. If gate sizes are selected so that the input and output rise/fall times are about equal, the short-circuit power consumption will be less than 15% of the dynamic power consumption. If, however, the design for high performance is taken to the extreme where large gates are used to drive relatively small loads, then there will be a stiff penalty in terms of short-circuit power consumption.

The short-circuit and leakage currents in CMOS circuits can be made small with proper circuit and device design techniques. The dominant source of power dissipation is thus the charging and discharging of the node capacitances (also referred to as the dynamic power dissipation) and is given by

$$P = 0.5CV_{dd}^2E(\text{sw})f_{clk} \qquad [1]$$

where C is the physical capacitance of the circuit, V_{dd} is the supply voltage, E(sw) (referred as the *switching activity*) is the average number of transitions in the circuit per $1/f_{clk}$ time, and f_{clk} is the clock frequency.

LOW-POWER DESIGN SPACE

The previous section revealed the three degrees of freedom inherent in the low-power design space: voltage, physical capacitance, and data activity. Optimizing for power entails an attempt to reduce one or more of these factors. This section briefly discusses each of these factors, describing their relative importance, as well as the interactions that complicate the power optimization process.

Voltage

Because of its quadratic relationship to power, voltage reduction offers the most effective means of minimizing power consumption. Without requiring any special circuits or technologies, a factor of 2 reduction in supply voltage yields a factor of 4

decrease in power consumption. Furthermore, this power reduction is a global effect, experienced not only in one subcircuit or block of the chip but throughout the entire design. Because of these factors, designers are often willing to sacrifice increased physical capacitance or circuit activity for reduced voltage. Unfortunately, we pay a speed penalty for supply voltage reduction, with delays drastically increasing as V_{dd} approaches the threshold voltage V_t of the devices. This tends to limit the useful range of V_{dd} to a minimum of about $(2–3)V_t$.

In Ref. 7, an architecture-driven voltage scaling strategy is presented in which parallel and pipelined architectures are used to compensate for the increased gate delays at reduced supply voltages and meet throughput constraints. Another approach to reduce the supply voltage without loss in throughput is to modify the V_t of the devices. Reducing the V_t allows the supply voltage to be scaled down without loss in speed. The limit of how low the V_t can go is set by the requirement to set adequate noise margins and to control the increase in subthreshold leakage currents. The optimum V_t must be determined based on the current drives at low-supply-voltage operation and control of the leakage currents. Because the inverse threshold slope (S) of a MOS Field-Effect Transistor (MOSFET) is invariant with scaling, for every 80–100 mV (based on the operating temperature) reduction in V_t, the standby current will be increased by one order of magnitude. This tends to limit V_t to about 0.3 V for room-temperature operation of CMOS circuits. Another important concern in the low V_{dd}–low V_t regime is the fluctuation in V_t. Basically, delay increases by 3x for a delta V_{dd} of ± 0.15 V at V_{dd} of 1 V. This is a major limitation on how low V_{dd} can go unless the V_t fluctuation is cancelled by circuit techniques such as the self-adjusting threshold scheme which will reduce the V_t fluctuation to ± 0.05 V at V_{dd} of 1 V (8).

Physical Capacitance

Dynamic power consumption depends linearly on the physical capacitance being switched. So, in addition to operating at low voltages, minimizing capacitances offers another technique for minimizing power consumption. In order to consider this possibility we must first understand what factors contribute to the physical capacitance of a circuit.

Power dissipation is dependent on the physical capacitances seen by individual gates in the circuit. Estimating this capacitance at the behavioral or logical levels of abstraction is difficult and imprecise, as it requires estimation of the load capacitances from structures which are not yet mapped to gates in a cell library; this calculation can, however, be done easily after technology mapping by using the logic and delay information from the library.

Interconnect plays an increasing role in determining the total chip area, delay, and power dissipation, and, hence, must be accounted for as early as possible during the design process. The interconnect capacitance estimation is, however, a difficult task even after technology mapping, due to lack of detailed place and route information. Approximate estimates can be obtained by using information derived from a companion placement solution (9) or by using stochastic/procedural interconnect models (10). Interconnect capacitance estimation after layout is straightforward and, in general, accurate.

With this understanding, we can now consider how to reduce physical capaci-

tance. From the previous discussion, we recognize that capacitances can be kept at a minimum by using less logic, smaller devices, and fewer and shorter wires. Example techniques for reducing the active area include resource sharing, logic minimization, and gate sizing. Example techniques for reducing the interconnect include register sharing, common subfunction extraction, placement, and routing. As with voltage, however, we are not free to optimize capacitance independently. For example, reducing device sizes reduces physical capacitance, but it also reduces the current drive of the transistors, making the circuit operate more slowly. This loss in performance might prevent us from lowering V_{dd} as much as we might otherwise be able to do.

Switching Activity

In addition to voltage and physical capacitance, switching activity also influences dynamic power consumption. A chip may contain an enormous amount of physical capacitance, but if there is no switching in the circuit, then no dynamic power will be consumed. The data activity determines how often this switching occurs. There are two components to switching activity: f_{clk} which determines the average periodicity of data arrivals and $E(\text{sw})$ which determines how many transitions each arrival will generate. For circuits that do not experience glitching, $E(\text{sw})$ can be interpreted as the probability that a power consuming transition will occur during a single data period. Even for these circuits, the calculation of $E(\text{sw})$ is difficult, as it depends not only on the switching activities of the circuit inputs and the logic function computed by the circuit but also on the spatial and temporal correlations among the circuit inputs. The data activity inside a 16-bit multiplier may change by as much as one order of magnitude as a function of input correlations (11).

For certain logic styles, however, glitching can be an important source of signal activity and, therefore, deserves some mention here. Glitching refers to spurious and unwanted transitions that occur before a node settles down to its final steady-state value. Glitching often arises when paths with unbalanced propagation delays converge at the same point in the circuit. Because glitching can cause a node to make several power-consuming transitions, it should be avoided whenever possible.

The data activity $E(\text{sw})$ can be combined with physical capacitance C to obtain *switched capacitance*, $C_{sw} = CE(\text{sw})$, which describes the average capacitance charged during each data period $1/f_{clk}$. It should be noted that it is the switched capacitance that determines the power consumed by a CMOS circuit.

Calculation of Switching Activity

Calculation of the switching activity in a logic circuit is difficult, as it depends on a number of circuit parameters and technology-dependent factors which are not readily available or precisely characterized. Some of these factors are described next.

Input Pattern Dependence

Switching activity at the output of a gate depends not only on the switching activities at the inputs and the logic function of the gate but also on the spatial and temporal dependencies among the gate inputs. For example, consider a two-input AND gate g with independent inputs i and j whose signal probabilities are 1/2, then $E_g(\text{sw}) =$

3/8. This holds because in 6 out of 16 possible input transitions, the output of the two-input and gate makes a transition. Now suppose it is known that only patterns 00 and 11 can be applied to the gate inputs and that both patterns are equally likely, then $E_g(\text{sw}) = 1/2$. Alternatively, assume that it is known that every 0 applied to input i is immediately followed by a 1, whereas every 1 applied to input j is immediately followed by a 0; then $E_g(\text{sw}) = 4/9$. Finally, assume that it is known that i changes exactly if j changes value; then $E_g(\text{sw}) = 1/4$. The first case is an example of *spatial* correlations between gate inputs, the second case illustrates *temporal* correlations on gate inputs, and the third case describes an instance of *spatiotemporal* correlations.

The straightforward approach of estimating power by using a simulator is greatly complicated by this pattern-dependence problem.

It is clearly infeasible to estimate the power by exhaustive simulation of the circuit. Recent techniques overcome this difficulty by using probabilities that describe the set of possible logic values at the circuit inputs and developing mechanisms to calculate these probabilities for gates inside the circuit. Alternatively, exhaustive simulation may be replaced by Monte Carlo simulation with well-defined stopping criteria for specified relative or absolute error in power estimates for a given confidence level (12).

Delay Model

Based on the delay model used, the power estimation techniques could account for steady-state transitions (which consume power but are necessary to perform a computational task) and/or hazards and glitches (which dissipate power without doing any useful computation). Sometimes, the first component of power consumption is referred to as the *functional activity*, whereas the latter is referred to as the *spurious activity*. It is shown in Ref. 13 that the mean value of the ratio of hazardous component to the total power dissipation varies significantly with the considered circuits (from 9% to 38% in random logic circuits) and that the spurious power dissipation cannot be neglected in CMOS circuits. The spurious activity is much higher in certain data path modules (such as adders and multipliers). Indeed, in a 32-bit pipelined multiplier, the power dissipation due to hazard activity is three to four times higher than that due to functional activity! The spurious power dissipation is likely to become even more important in the future scaled technologies.

Current power estimation techniques often handle both zero-delay (nonglitch) and real-delay models. In the first model, it is assumed that all changes at the circuit inputs propagate through the internal gates of the circuits instantaneously. The latter model assigns to each gate in the circuit a finite delay and can thus account for the hazards in the circuit. A real-delay model significantly increases the computational requirements of the power estimation techniques while improving the accuracy of the estimates.

Calculation of the spurious activity in a circuit is, in general, very difficult and requires careful logic- and/or circuit-level characterization of the gates in a library as well as detailed knowledge of the circuit structure.

Logic Function

Switching activity at the output of a logic gate is also strongly dependent on the Boolean function of the gate itself. This is because the logic function of a gate

determines the probability that the present value of the gate output is different from its previous value. For example, under the assumption that the input signals are uncorrelated, switching activity at the output of a (static) two-input NAND or NOR gate is 3/8, whereas that at the output of a two-input XOR gate is 1/2. Indeed, switching activity at the output of a K-input NAND or NOR gate approaches $1/2^{K-1}$ for large K, whereas that for a K-input XOR gate remains at 1/2.

Logic Style

Switching activity of the circuits is also a function of the logic style used to implement the circuit. The functional activity in dynamic circuits is *always* higher than that in static implementation of the same circuit, as all nodes are precharged to some value (one in N-type dynamic and zero in P-type dynamic) before the new input data arrives. This effectively increases the number of power-consuming transitions. For example, under pseudorandom input signals, switching activities of two-input N-type dynamic NAND, NOR and XOR gates are 3/2, 1/2, and 1, respectively, and those of the P-type version of these same gates are 1/2, 3/2, and 1, respectively. These values should be compared to the switching activities of these gates in static CMOS which are 3/8, 3/8, and 1/2, respectively. Note, however, that the physical capacitance in dynamic logic tends to be smaller than that in static logic, so the choice between dynamic and static logic implementations is not as clear-cut as it would be otherwise. Dynamic circuits are also glitch-free!

Circuit Structure

The major difficulty in computing the switching activities is the reconvergent nodes. Indeed, if a network consists of simple gates and has no reconvergent fanout nodes (i.e., circuit nodes that receive inputs from two paths that fanout from some other circuit node), then the exact switching activities can be computed during a single post-order traversal of the network. For networks with reconvergent fanout, the problem is much more challenging, as internal signals may become strongly correlated and exact consideration of these correlations cannot be performed with reasonable computational effort or memory usage. Current power estimation techniques either ignore these correlations or approximate them, thereby improving the accuracy at the expense of longer run times. Exact methods (i.e., symbolic simulation) have also been proposed but are impractical due to excessive time and memory requirements.

Statistical Variation of Circuit Parameters

In real networks, statistical perturbations of circuit parameters may change the propagation delays and produce changes in the number of transitions because of the appearance or disappearance of hazards. It is therefore useful to determine the change in the signal transition count as a function of this statistical perturbations. Variation of gate-delay parameters may change the number of hazards occurring during a transition as well as their duration. For this reason, it is expected that the hazardous component of power dissipation is more sensitive to integrated-circuit (IC) parameter fluctuations than the power required to perform the transition between the initial and final states of each node.

POWER ESTIMATION TECHNIQUES

The design for the low-power problem cannot be achieved without accurate power prediction and optimization tools or without power-efficient gate and module libraries. Therefore, there is a critical need for computer-aided design (CAD) tools to estimate power dissipation during the design process to meet the power budget without having to go through a costly redesign effort and enable efficient design and characterization of the design libraries.

In the following section, various techniques for power estimation at the circuit, logic, and behavioral levels will be reviewed. These techniques are divided into two general categories: simulation based and nonsimulation based.

Simulative Approaches

Brute-Force Simulation

Circuit simulation-based techniques (14,15) simulate the circuit with a representative set of input vectors. They are accurate and capable of handling various device models, different circuit design styles, single-phase and multiphase clocking methodologies, tristate drives, and so forth. However, they suffer from memory and execution time constraints and are not suitable for large, cell-based designs. In addition, it is difficult to generate a compact stimulus vector set to calculate accurate activity factors at the circuit nodes. The size of such a vector set is dependent on the application and the system environment (16).

PowerMill (17) is a *transistor-level power simulator* and analyzer which applies an event-driven timing simulation algorithm (based on simplified table-driven device models, circuit partitioning, and single-step nonlinear iteration) to increase the speed by two to three orders of magnitude over SPICE.

Switch-level simulation techniques are, in general, much faster than circuit-level simulation techniques but are not as accurate or versatile. Standard switch-level simulators [such as IRSIM (18)] can be easily modified to report the switched capacitance (and thus dynamic power dissipation) during a simulation run.

The Verilog-XL logic simulator is a Verilog-based gate-level simulation program that relies on the accuracy of the macromodels built for the gates in the application-specific IC (ASIC) library as well as gate-level timing analysis to produce fast and accurate power estimates. The accuracy depends heavily on the quality of the macromodels, the glitch filtering scheme used, and the accuracy of physical capacitances provided at the gate level. The speed is three to four orders of magnitude faster than SPICE.

Most of the high-level power-prediction tools use profiling and simulation to address data dependencies. Important statistics include the number of operations of a given type, the number of bus, register, and memory accesses, and the number of I/O operations executed within a given period (19,20). Instruction-level simulation or behavioral simulators are easily (and have indeed been) adapted to produce this information.

Hierarchical Simulation

A simulation method based on a hierarchy of simulators is presented in Ref. 21. The idea is to use a hierarchy of power simulators (e.g., at architectural, gate level,

and circuit level) to achieve a reasonable accuracy and efficiency tradeoff. Another good example is Entice–Aspen (22). This power analysis system consists of two components: Aspen which computes the circuit activity information and Entice which computes the power characterization data. A stimulus file is to be supplied to Entice, where power and timing delay vectors are specified. The set of power vectors discretizes all possible events in which power can be dissipated by the cell. With the relevant parameters set according to the user's specs, a SPICE circuit simulation is invoked to accurately obtain the power dissipation of each vector. During logic simulation, Aspen monitors the transition count of each cell and computes the total power consumption as the sum of the power dissipation for all cells in the power vector path.

Monte Carlo Simulation

A *Monte Carlo simulation* approach for power estimation which alleviates the input pattern dependence problem has been proposed in Ref. 12. This approach consists of applying randomly generated input patterns at the circuit inputs and monitoring the power dissipation per time interval T using a simulator. Based on the assumption that the power consumed by the circuit over any period T has a normal distribution, and for a desired percentage error in the power estimate and a given confidence level, the number of required power samples is estimated. The designer can use an existing simulator (circuit level, gate level, or behavioral) in the inner loop of the Monte Carlo program, thus trading accuracy for higher efficiency. The convergence time for this approach is fast when estimating the total power consumption of the circuit. However, when signal probability (or power consumption) values on individual lines of the circuit are required, the convergence rate is very slow (23). The method does not handle spatial correlations at the circuit inputs.

Nonsimulative Approaches

Behavioral Level

For functional units (adders, multipliers, and registers) or for memories, power estimates are directly obtained from the design library whereby each functional unit has been simulated using pseudorandom white noise data and the average switched capacitance per clock cycle has been calculated and stored in the library.

The power model for a functional unit may be parametrized in terms of its input bit width. For example, the power dissipation of an adder (or a multiplier) is linearly (or quadratically) dependent on its input bit width. The library thus contains interface descriptions of each module, description of its parameters, its area, delay, and internal power dissipation (assuming pseudorandom white noise data inputs). The latter is determined by extracting a circuit- or logic-level model from the layout or logic-level descriptions of the module, simulating it using a long stream of randomly generated input patterns and calculating the average power dissipation per pattern. These characteristics are available in terms of the parameter values (i.e., equations) or in the form of tables. Multiparameter modules are characterized with respect to all the parameters, yielding a multiparameter equation or table. Multifunction modules (e.g., Arithmetic Logic Unit (ALU)) are characterized for each function separately.

The power model thus generated and stored for each module in the library has

to be "conditioned" or "modulated" by the *actual* input switching activities in order to provide power estimates which are sensitive to the input activities. In Refs. 20 and 24b, the model consists of a single physical capacitance value and a single switching activity value which represents the average switching activity on each input bit. In Ref. 24a, a more detailed model is presented, where it is projected that data in the datapath of a digital system can be divided into two regions: the least significant bits (LSB), which act as uncorrelated white noise, and the most significant bits (MSB), which correspond to sign bits and exhibit strong temporal dependence. The power model thus uses two capacitance values and requires two input switching activity values corresponding to the LSB and MSB regions. Both models ignore the spatial correlations among bits of the same input or across bits of different inputs.

Another parametric model is described in Ref. 25, where the power dissipation of the various components of a typical processor architecture are expressed as a function of set of primary parameters. The technique suffers from an abundance of parameters, requires a lot of fine-tuning for specific architectures, and is sensitive to mismatches in the modeling assumptions.

Word-level behavior of a data input can be properly captured by its probability density function (pdf). Similarly, spatial correlation between two data inputs can be captured by their joint pdf. This observation is used in Refs. 26 and 27 to develop a probabilistic technique for behavioral-level power prediction which consists of four steps: (1) building the joint pdf of the input variables of a data flow graph (DFG) based on the given input vectors, (2) computing the joint pdf for any combination of internal arcs in the DFG, (3) calculating the switching activity at the inputs of each functional block or register in the DFG using the joint pdf of the inputs and the data representation format which determines the (bit-level) Hamming distances of (word-level) data values, (4) estimating the power dissipation of each functional block using the input statistics obtained in step 3 and the library characterization data that gives the physical capacitance information for each module in the library. This method is very robust, but suffers from the worst-case complexity of joint pdf computation and inaccuracies associated with the library characterization data.

An information-theoretic approach is described in Refs. 28 and 29 which relies on information-theoretic measures of activity (e.g., entropy) to devise fast, yet accurate, power estimation at the algorithmic and structural behavioral levels. In the following, the approach presented in Ref. 28 will be summarized. Entropy characterizes the uncertainty of a sequence of applied vectors and thus, intuitively, is related to switching activity. Indeed, it is shown that an upper bound on the average switching activity of a bit is half of its entropy. Knowing the statistics of the input stream and having some information about the structure (or functionality) of the circuit, the input and output entropies per bit are calculated using a closed-form expression that gives the output entropy per bit as a function of the input entropy per bit, a structure-dependent *information scaling factor*, and the distribution of gates as a function of logic depth in the circuit (or using a *compositional technique* which has a linear complexity in terms of the circuit size). Next, the average entropy per circuit line is calculated and used as an estimate of the average switching activity per signal line. This is then used to estimate the power dissipation of the module. A major advantage of this technique is not simulation based and is thus very fast, yet

it produces accurate power estimates. In general, using structural information can provide more accurate estimates based on the entropy measure. On the other hand, evaluations based on functional information need less information about the circuit and, therefore, may be more appealing in practice as it provides an estimate of power consumption earlier in the design cycle.

The above techniques apply to data paths. Behavioral power prediction models have also been proposed for the controller circuitry in Refs. 20 and 30. These techniques provide quick estimation of the power dissipation in a controller based on the knowledge of its target implementation style (i.e., precharged pseudo-nMOS or dynamic Programmable Logic Array (PLA)), the number of inputs, outputs, states, and so on. The estimates can be made more accurate by introducing empirical parameters that are determined by curve fitting and least-squares fit error analysis on real data.

Logic Level

Estimation Under a Zero-Delay Model: Most of the power in CMOS circuits is consumed during charging and discharging of the load capacitance. To estimate the power consumption, one has to calculate the (switching) activity factors of the internal nodes of the circuit. Methods of estimating the activity factor $E_n(\text{sw})$ at a circuit node n involve estimation of signal probability $\text{prob}(n)$, which is the probability that the signal value at the node is 1. Under the assumption that the values applied to each circuit input are temporally independent (i.e., value of any input signal at time t is independent of its value at time $t - 1$), we can write

$$E_n(\text{sw}) = 2\,\text{prob}(n)[1 - \text{prob}(n)]. \qquad [2]$$

Computing signal probabilities has attracted much attention (31,32). In recent years, a computational procedure based on ordered binary-decision diagrams (OBDDs) (33) has become widespread. In this method, which is known as the *OBDD-based* method, the signal probability at the output of a node is calculated by first building an OBDD corresponding to the *global function* of the node (i.e., function of the node in terms of the circuit inputs) and then performing a postorder traversal of the OBDD using equation:

$$\text{prob}(y) = \text{prob}(x)\,\text{prob}(f_x) + \text{prob}(\bar{x})\,\text{prob}(f_{\bar{x}}). \qquad [3]$$

This leads to a very efficient computational procedure for signal probability estimation.

In Ref. 34, a procedure for propagating signal probabilities from the circuit inputs toward the circuit outputs using only *pairwise correlations* between circuit lines and ignoring higher-order correlation terms is described. In Refs. 35 and 36, the temporal correlation between values of some signal x in two successive clock cycles is modeled by a time-homogeneous Markov chain which has two states 0 and 1 and four edges where each $ij(i, j = 0, 1)$ is annotated with the conditional probability prob_{ij}^x that x will go to state j at time $t + 1$ if it is in state i at time t. The transition probability $\text{prob}(x_{i\to j})$ is equal to $\text{prob}(x = i)\,\text{prob}_{ij}^x$. Obviously, $\text{prob}_{00}^x + \text{prob}_{01}^x = \text{prob}_{10}^x + \text{prob}_{11}^x = 1$, whereas $\text{prob}(x) = \text{prob}(x_{0\to 1}) + \text{prob}(x_{1\to 1})$ and $\text{prob}(\bar{x}) = \text{prob}(x_{0\to 0}) + \text{prob}(x_{1\to 0})$. The activity factor of line x can be expressed in terms of these transition probabilities as follows:

$$E_x(\text{sw}) = \text{prob}(x_{0\to 1}) + \text{prob}(x_{1\to 0}). \quad [4]$$

The various transition probabilities can be computed exactly using the OBDD representation of the logic function of x in terms of the circuit inputs.

The authors of Ref. 36 also describe a mechanism for propagating the transition probabilities through the circuit which is more efficient as there is no need to build the global function of each node in terms of the circuit inputs. The loss in accuracy is often small while the computational saving is significant. They then extend the model to account for spatiotemporal correlations. This work has been extended to handle highly correlated input streams using the notions of *conditional independence* and *isotropy of signals* (11). Based on these notions, it is shown that the relative error in calculating the signal probability of a logic gate using pairwise correlation coefficients can be bounded from above.

Estimation Under a Real-Delay Model: The above methods only account for steady-state behavior of the circuit and thus ignore hazards and glitches. This section reviews some techniques that examine the dynamic behavior of the circuit and thus estimate the power dissipation due to hazards and glitches.

In Ref. 37, the exact power estimation of a given combinational logic circuit is carried out by creating a set of symbolic functions such that summing the signal probabilities of the functions corresponds to the average switching activity at a circuit line x in the original combinational circuit (this method is known as the *symbolic simulation* method). The inputs to the created symbolic functions are the circuit input lines at time instances 0^- and ∞. Each function is the **exclusive or** of the characteristic functions describing the logic values of x at two consecutive instances. The major disadvantage of this estimation method is its exponential complexity. However, for the circuits to which this method is applicable, the estimates provided by the method can serve as a basis for comparison among different approximation schemes.

The concept of a probability waveform is introduced in Ref. 38. This waveform consists of a sequence of transition edges or events over time from the initial steady state (time 0^-) to the final steady state (time ∞) where each event is annotated with an occurrence probability. The probability waveform of a node is a compact representation of the set of all possible logical waveforms at that node. Given these waveforms, it is straightforward to calculate the switching activity of x which includes the contribution of hazards and glitches; that is,

$$E_x(\text{sw}) = \sum_{t^i \in \text{eventlist}(x)} [\text{prob}(x^i_{0\to 1}) + \text{prob}(x^i_{1\to 0})]. \quad [5]$$

where eventlist(x) denotes one list of events in the probability waveform of x.

Given such waveforms at the circuit inputs and with some convenient partitioning of the circuit, the authors examine every subcircuit and derive the corresponding waveforms at the internal circuit nodes. In Ref. 39, an efficient *probabilistic simulation* technique is described that propagates transition waveforms at the circuit primary inputs up in the circuit and thus estimates the total power consumption (ignoring signal correlations due to the reconvergent fanout nodes).

A *tagged probabilistic simulation* approach is described in Ref. 40 that correctly accounts for reconvergent fanout and glitches. The key idea is to break the set of possible logical waveforms at a node n into four groups, each group being characterized by its steady-state values (i.e., values at time instance 0^- and ∞).

Next, each group is combined into a probability waveform with the appropriate steady-state tag. Given the tagged probability waveforms at the input of a simple gate, it is then possible to compute the tagged probability waveforms at the output of the gate. The correlation between probability waveforms at the inputs is approximated by the correlation between the steady-state values of these lines. This is much more efficient than trying to estimate the dynamic correlations between each pair of events. This approach requires significantly less memory and runs much faster than symbolic simulation, yet achieves very high accuracy (e.g., the average error in aggregate power consumption is about 10%). In order to achieve this level of accuracy, detailed timing simulation along with careful *glitch filtering*, and *library characterization* are needed (41). The first item refers to the scheme for eliminating some of the short glitches that cannot overcome the gate inertias from the probability waveforms. The second item refers to the process of generating accurate and detailed macromodeling data for the gates in the cell library.

Sequential Circuits

Recently developed methods for power estimation have primarily focused on combinational logic circuits. The estimates produced by purely combinational methods can greatly differ from those produced by the exact method. Indeed, accurate average switching activity estimation for finite-state machines (FSMs) is considerably more difficult than that for combinational circuits for two reasons: (1) The probability of the circuit being in each of its possible states has to be calculated; (2) The present state line inputs of the FSM are strongly correlated (i.e., they are temporally correlated due to the machine behavior as represented in its state transition graph description and they are spatially correlated because of the given state encoding).

A first attempt at estimating switching activity in FSMs has been presented in Ref. 37. The idea is to *unroll* the next-state logic once (thus capturing the temporal correlations of present-state lines) and then perform symbolic simulation on the resulting circuit (which is hence treated as a combinational circuit). This method does not, however, capture the spatial correlations among present-state lines and makes the simplistic assumption that the state probabilities are uniform.

The above work is improved upon in Refs. 42 and 43, where results obtained by using the Chapman–Kolmogorov equations for discrete-time Markov chains to compute the exact state probabilities of the machine are presented. The Chapman–Kolmogorov method requires the solution of a linear system of equations of size 2^N, where N is the number of flip-flops in the machine. Thus, this method is limited to circuits with a small number of flip-flops, as it requires the explicit consideration of each state in the circuit.

The authors of Refs. 42 and 43 also describe a method for approximate switching activity estimation of sequential circuits. The basic computation step is the solution of a nonlinear system of equations in terms of the present-state bit probabilities and signal probabilities for the combinational inputs of the FSM. The fixed point (or zero) of this system of equations can be found using the Picard–Peano (or Newton–Raphson) iteration (44). Increasing the number of variables or the number of equations in the above system results in increased accuracy (45). For a wide variety of examples, it is shown that the approximation scheme is within 1–3% of the exact method but is orders of magnitude faster for large circuits. Previous

sequential switching activity estimation methods exhibit significantly greater inaccuracies.

POWER MINIMIZATION TECHNIQUES

To address the challenge to reduce power, the semiconductor industry has adopted a multifaceted approach, attacking the problem on four fronts:

1. **Reducing chip and package capacitance**: This can be achieved through process development such as silicon-on-insulator with partially or fully depleted wells, CMOS scaling to submicron device sizes, and advanced interconnect substrates such as multichip modules (MCM). This approach can be very effective but is also very expensive and has its own pace of development and introduction to the market.
2. **Scaling the supply voltage**: This approach can be very effective in reducing the power dissipation, but often requires new IC fabrication processing. Supply voltage scaling also requires support circuitry for low-voltage operation including level converters and DC/DC converters, as well as detailed consideration of issues such as signal-to-noise.
3. **Employing better design techniques**: This approach promises to be very successful because the investment to reduce power by design is relatively small in comparison to the other three approaches and because it is relatively untapped in potential.
4. **Using power management strategies**: The power savings that can be achieved by various static and dynamic power management techniques are very application dependent but can be significant.

In the following we will discuss these strategies in some depth. The various approaches interact with one another; for example, CMOS device scaling, supply voltage scaling, and choice of circuit architecture must be done judiciously and carefully in order to find an optimum power-area-delay trade-off.

CMOS Device and Voltage Scaling

In the future, the scaling of voltage levels will become a crucial issue. The main force behind this drive is the ability to produce complex, high-performance systems on a chip. This is further exacerbated by the projected explosion in demand for portable and wireless systems with very low power consumption. It is also expected that various memory and ASIC's will also switch to lower supply voltages to maintain manageable power densities. A key concern is the availability of the complete chip set to make up systems at reduced supply voltages. However, most of the difficulties can be circumvented by techniques to mix and match different supply voltages on-board or on the chip.

In Ref. 46, two CMOS device and voltage scaling scenarios are described, one optimized for the highest speed and one trading off high speed for significantly lower power (the speed of the low-power case in one generation is about the same as the speed of the high-performance case of the previous generation, with greatly reduce power consumption). It is shown that the low-power scenario is very close to

the constant electric-field (ideal) scaling theory. It is shown that a speed improvement of 7× and over two orders of magnitude improvement in power-delay product (mW/MIPS) are expected by scaling of bulk CMOS down to sub-0.1-μm region as compared with today's high-performance 0.6-μm devices at 5 V. This article also presents a discussion of how high the electric field in a transistor channel can go without impacting the long-term device reliability while achieving high performance and low power. Next the speed/standby current trade-off is addressed, dealing with the issue of nonscalibity of the threshold voltage.

The status of silicon-on-insulator (SOI) approach to scaled CMOS is also reviewed, showing that the potential for 3× savings in power compared to the bulk case at the same speed. The performance improvement of SOI compared to bulk CMOS is mainly due to the reduction of parasitic capacitances and body effect. Also, in partially depleted device designs, the floating-body effect can give rise to a sharper subthreshold slope (<60 mV/dec) at high drain bias, which effectively reduces the threshold voltage and can actually improve the performance at a given standby current. In addition, CMOS on SOI offers significant reduction in soft-error rate, latch-up elimination, and simpler isolation which results in reduced wafer fabrication steps. The main challenges are the availability of low-cost wafers with low defect density at high volumes, floating-body effects on the device and circuit operation, and heat dissipation through the buried oxide.

CAD Methodologies and Techniques

Low-power VLSI design can be achieved at various levels of the design abstraction from algorithmic and system levels down to layout and circuit levels. In the following, some of these optimization techniques will be briefly mentioned.

System Design

At the system level, inactive hardware modules may be automatically turned off to save power; modules may be provided with the optimum supply voltage and interfaced by means of level converters; some of the energy that is delivered from the power supply may be cycled back to the power supply; a given task may be partitioned between various hardware modules or programmable processors or both so as to reduce the system-level power consumption.

Behavioral Synthesis

Behavioral synthesis is the process of generating a register-transfer-level (RTL) design from an algorithmic behavioral specification. In particular, it constructs a structural view of the data path and a logical view of the control unit of a circuit. The data path consists of a set of interconnected functional units (arithmetic, logic, memory, and registers) and steering units (multiplexers and busses) while the control unit sends signals to the data path to schedule the appropriate sequence of operations in time. The behavioral synthesis process consists of three steps: allocation, assignment, and scheduling. These steps determine how many instances of each resource are needed, on what resource each operation is performed, and when each operation is executed.

A wide class of transformations can be done at the behavioral level and most of them are typically aimed at either reducing the number of cycles in a computation

or reducing the number of resources used in the computation. One interesting approach is to introduce more concurrency in a circuit to speed it up and then to reduce the voltage until it realizes its originally required speed. The linear increase in capacitance due to parallelism is compensated for by the quadratic power reduction due to reducing the voltage. This can result in circuits that use several times less power. Although this transformation is not directly changing the supply voltage, it allows a design to operate with a lower supply voltage by increasing the concurrency. Another interesting approach is to reduce the supply voltage of each functional unit (thus reducing the power consumption but increasing the delay of the unit) in the data path as much as possible while satisfying the timing requirements in terms of the cycle time or throughput (in the case of pipelined circuits). This approach requires various support circuitry including level converters and DC/DC converters. A good overview of the use of optimizing transformations for supply voltage reduction is given in Ref. 19. These transformations include concurrency increasing transformations such as (time) loop unrolling and control-flow optimizations and critical path reducing transformations such as retiming and pipelining.

At the early stages of the behavioral design process, concurrency increasing transformations such as loop unrolling, pipelining, and control-flow optimization as well as critical-path-reducing transformations such as height minimization, retiming, and pipelining may be used to allow a reduction in supply voltage without degrading system throughput; algorithm-specific instruction sets may be utilized that boost code density and minimize switching; a Gray code addressing scheme can be used to reduce the number of bit changes on the address bus; an on-chip cache may be added to minimize external memory references; locality of reference may be exploited to avoid accessing global resources such as memories, busses, or ALUs; control signals that are "don't cares" can be held constant to avoid initiating nonproductive switching.

Other transformations at this level do not differ fundamentally from the classical behavioral transformations, but now the cost function used to steer the transformations is different. A key challenge, however, is to exploit the input signal statistics (i.e., switching activity on individual inputs and correlations among a set of inputs) to minimize the power consumption during register and module allocation and binding while maintaining the same cycle time or throughput.

Consider a module M in an RTL circuit that performs two operations, A and B. The switching activity at the inputs of M is determined by the number of bit flips between the values taken on by the variables that are inputs to the two operations, which, in turn, depend on the bit-level statistical characteristics of the variables. Hence, the power dissipation depends on the module binding. Similarly, consider a register R that is shared between two data values X and Y. The switching activity of R depends on the correlations between these two variables X and Y. Hence, the power dissipation depends on the register binding as well. These observations form the basis for power optimization during module and register allocation and binding in Refs. 26, 27, 47, 48.

In Ref. 49, an exact (graph-theoretic) algorithm for minimizing the system power through variable-voltage scheduling is presented. The idea is to establish a supply voltage level for each of the operations in a data flow graph, thereby fixing the latency of that operation, such that the system timing constraint is met while power is minimized (because each operation will be executed using minimum possible supply voltage).

Logic Synthesis

Logic synthesis fits between the register-transfer level and the netlist of gates specification. It provides the automatic synthesis of netlists minimizing some objective function subject to various constraints. Example inputs to a logic synthesis system include two-level logic representation, multilevel Boolean networks, finite-state machines, and technology-mapped circuits. Depending on the input specification (combinational versus sequential, synchronous versus asynchronous), the target implementation (two-level versus multilevel, unmapped versus mapped, ASICs versus Field-Programmable Grate Arrays (FPGAs)), the objective function (area, delay, power, testability) and the delay models used (zero-delay, unit-delay, unit-fanout delay, or library delay models), different techniques are applied to transform and optimize the original RTL description.

Once various system-level architectural and technological choices are made, it is the switched capacitance of the logic that determines the power consumption of a circuit. In this section, a number of techniques for power estimation and minimization during logic synthesis will be presented. The strategy for synthesizing circuits for low power consumption will be to restructure or optimize the circuit to obtain low switching activity factors at nodes which drive large capacitive loads.

At the register-transfer (RT) and logic levels, symbolic states of a finite-state machine (FSM) can be assigned binary codes to minimize the number of bit changes in the combinational logic for the most likely state transitions (50); latches in a pipelined design can be repositioned to eliminate hazardous activity in the circuit (51); parts of the circuit that do not contribute to the present computation may be shut off completely; output logic values of a circuit may be precomputed one cycle before they are required and then used to reduce the internal switching activity of the circuit in the succeeding clock cycle (52); common subexpressions with low transition probability values can be extracted (53); network "don't cares" can be used to modify the input variable support and thus the local expression of a node so as to reduce the bit switching in the transitive fanout of the node (54); nodes with high switching activity may be hidden inside CMOS gates where they drive smaller physical capacitances (15); hazards/glitches in the circuit can be reduced by appropriate use of selective collapse, logic decomposition, or delay insertion which lead to path-balanced circuit structures; the circuit depth and power dissipation may be simultaneously minimized using a node clustering approach; PLAs can be implemented to reduce static or dynamic power dissipation in pseudo-NMOS or dynamic NOR–NOR implementations (55). Power dissipation may be further reduced by gate resizing (56), signal-to-pin assignment, and I/O encoding.

Physical Design

Physical design fits between the netlist of gates specification and the geometric (mask) representation known as the layout. It provides the automatic layout of circuits minimizing some objective function subject to given constraints. Depending on the target design style (full custom, standard cell, gate arrays, FPGAs), the packaging technology (printed-circuit boards, multichip modules, wafer-scale integration) and the objective function (area, delay, power, reliability), various optimization techniques are used to partition, place, resize, and route gates.

Under a zero-delay model, the switching activity of gates remains unchanged during layout optimization; hence, the only way to reduce power dissipation is to decrease the load on high-switching-activity gates by proper netlist partitioning and

gate placement, gate and wire sizing, transistor reordering, and routing. At the same time, if a real-delay model is used, various layout optimization operations influence the hazard activity in the circuit. This is, however, a very difficult analysis and optimization problem and requires further research.

It should be noted that by applying postlayout optimization techniques (such as buffer and wire sizing, local restructuring and remapping, etc.), power can be further reduced. Under a zero-delay model, the switching activity of gates remains unchanged during layout optimization, and hence, the only way to reduce power dissipation is to decrease the load on high-switching-activity gates by proper netlist partitioning and gate placement, gate and wire sizing, transistor reordering, and routing. At the same time, if a real-delay model is used, various layout optimization operations influence the hazard activity in the circuit.

At the physical design level, power may be reduced by using appropriate net weights during netlist partitioning, floor-planning, placement (6), and routing; individual transistors may be sized down to reduce the power dissipation along the noncritical paths in a circuit; large capacitive loads can be buffered using optimally sized inverter chains so as to minimize the power dissipation subject to a given delay constraint (57); wire and driver sizing may be combined to reduce the interconnect delay with only a small increase in the power dissipation (58); clock trees may be constructed that minimize the load on the clock drivers subject to meeting a tolerable clock skew (59,60).

Circuit Design

At the circuit level, power-savings techniques that recycle the signal energies using the adiabatic switching principles rather than dissipating them as heat are promising in certain applications where speed can be traded for lower power (61). Similarly, techniques based on combining self-timed circuits with a mechanism for selective adjustment of the supply voltage that minimizes the power while satisfying the performance constraints (62), those based on partial transfer of the energy stored on a capacitance to some charge-sharing capacitance and then reusing this energy at a later time (63), and those based on electronic compensation for variations in V_T, thus making it possible to scale power-supply voltages down to very low levels (64) show good signs. Design of energy-efficient level converters and DC/DC converters is also essential to the success of adaptive supply voltage strategies.

Power Management Strategies

In many synchronous applications, a lot of power is dissipated by the clock. The clock is the only signal that switches all the time and it usually has to drive a very large clock tree. Moreover, in many cases, the switching of the clock causes a lot of additional unnecessary gate activity. For that reason, circuits are being developed with controllable clocks. This means that from the master clock, other clocks are derived that can be slowed down or stopped completely with respect to the master clock, based on certain conditions. The circuit itself is partitioned in different blocks and each block is clocked with its own (derived) clock. The power savings that can be achieved this way are very application dependent, but can be significant.

Power savings techniques that recycle the signal energies using the adiabatic switching principles rather than dissipating them as heat are promising in certain

applications where speed can be traded for lower power. Similarly, techniques based on combining self-timed circuits with a mechanism for selective adjustment of the supply voltage that minimizes the power while satisfying the performance constraints show good signs.

CHALLENGES AHEAD

The need for low-power systems is being driven by many market segments. There are several approaches to reducing power; however, the highest return-on-investment approach is through designing for low power. Unfortunately designing for low power adds another dimension to the already complex design problem; the design has to be optimized for power as well as performance and area.

Optimizing the three axes necessitates a new class of power-conscious CAD tools. The problem is further complicated by the need to optimize the design for power at all design phases. The successful development of new power conscious tools and methodologies requires a clear and measurable goal. In this context the research work should strive to reduce power by 5–10× in 3 years through design and tool development.

To conclude this introduction, it is worthwhile to summarize the major challenges that, to our belief, have to be addressed if we want to keep power dissipation within bounds in the next generations of digital integrated circuits (65).

- A low-voltage/low-threshold technology and circuit design approach, targeting supply voltages around 1 V and operating with reduced thresholds.
- Low-power interconnect, using advanced technology, reduced swing, or reduced activity approaches.
- Dynamic power management techniques, varying supply voltage and execution speed according to activity measurements. This can be achieved by partitioning the design into subcircuits whose energy levels can be independently controlled and by powering down subcircuits which are not in use.
- System performance can be improved by moving the work to less-energy-constrained parts of the system; for example, by performing the task on fixed stations rather than mobile sites, by using asymmetric communication protocols, or unbalanced data compression schemes.
- Application-specific processing. This might rely on the increased use of application-specific circuits or application- or domain-specific processors. Examples include implementing the most energy consumptive operations in hardware, choosing a processor with instruction set, data path width and functional units best suited to the algorithm, mapping functions to hardware so that interchip communication is reduced, and using suitable memory hierarchy.
- Move toward self-adjusting and adaptive circuit architectures that can quickly and efficiently respond to the environmental change as well as varying data statistics.
- An integrated design methodology—including synthesis and compilation tools. This might require the progression to higher-level programming and specification paradigms (e.g., data flow or object-oriented programming).

- Development of power conscious techniques and tools for behavioral synthesis, logic synthesis, and layout optimization. The key requirements for these techniques are accurate and efficient estimation of the power cost of alternative organizations and/or implementations and the ability to minimize the power dissipation subject to given performance (or throughput in case of pipelined designs) constraints and supply-voltage levels.
- Power-savings techniques that recycle the signal energies using the adiabatic switching principles rather than dissipating them as heat are promising in certain applications where speed can be traded for lower power.

REFERENCES

1. R. A. Powers, "Batteries for Low Power Electronics," *Proc. IEEE, 38*(4), 687–693 (1995).
2. D. Dobberpuhl et al., "A 200 MHz, 64b, Dual Issue CMOS Microprocessor," *Digest of Technical Papers, ISSC '92*, 1992, pp. 106–107.
3. C. Small, "Shrinking Devices Put the Squeeze on System Packaging," *Electric Design Newsletter, 39*(4), 41–46 (1994).
4. M. Horowitz, T. Indermaur, and R. Gonzalez, "Low-Power Digital Design," in *Proceedings of the 1995 IEEE Symposium on Low Power Electronics*, 1995, pp. 8–11.
5. T. A. Fjeldly and M. Shur, "Threshold Voltage Modeling and the Subthreshold Regime of Operation of Short-Channel MOSFET's," *IEEE Trans. Electron. Devices, ED-40*(1), 137–145 (1993).
6. H. Vaishnav and M. Pedram, "PCUBE: A Performance Driven Placement Algorithm for Low Power Designs," in *Proceedings of the European Design Automation Conference*, 1993, pp. 72–77.
7. A. Chandrakasan, S. Sheng, and R. W. Brodersen, "Low-Power CMOS Design," *J. Solid-State Circuits, 27*(4), 472–484 (1992).
8. T. Kobayashi and T. Sakurai, "Self-Adjusting Threshold-Voltage Scheme for Low Voltage High Speed Operation," *Proceedings of CICC*, 1994, pp. 271–274.
9. M. Pedram and N. Bhat, "Layout Driven Technology Mapping," in *Proceedings of the 28th Design Automation Conference*, 1991, pp. 95–105.
10. M. Pedram and B. T. Preas, "Interconnection Length Estimation for Optimized Standard Cell Layouts," in *Proceedings of the IEEE International Conference on Computer Aided Design*, 1989, pp. 390–393.
11. R. Marculescu, D. Marculescu, and M. Pedram, "Efficient Power Estimation for Highly Correlated Input Streams," in *Proceedings of the 32nd Design Automation Conference*, 1995, pp. 628–634.
12. R. Burch, F. N. Najm, P. Yang, and T. Trick, "A Monte Carlo Approach for Power Estimation," *IEEE Trans. VLSI Syst., 1*(1), 63–71 (1993).
13. L. Benini, M. Favalli, and B. Ricco, "Analysis of Hazard Contribution to Power Dissipation in CMOS IC's," in *Proceedings of the 1994 International Workshop on Low Power Design*, 1994, pp. 27–32.
14. S. M. Kang, "Accurate Simulation of Power Dissipation in VLSI Circuits," *J. Solid State Circuits, 21*(5), 889–891 (1986).
15. C-Y. Tsui, M. Pedram, and A. M. Despain, "Power Efficient Technology Decomposition and Mapping Under an Extended Power Consumption Model," *IEEE Trans. Computer-Aided Design Integrated Circuits Systs, 13*(9), 1110–1122 (1994).
16. S. Rajgopal and G. Mehta, "Experiences with Simulation-Based Schematic Level Cur-

rent Estimation," in *Proceedings of the 1994 International Workshop on Low Power Design*, 1994, pp. 9–14.

17. C. Deng, "Power Analysis for CMOS/BiCMOS Circuits," in *Proceedings of the 1994 International Workshop on Low Power Design*, 1994, pp. 3–8.
18. A. Salz and M. A. Horowitz, "IRSIM: An Incremental MOS Switch-Level Simulator," in *Proceedings of the 26th Design Automation Conference*, 1989, pp. 173–178.
19. A. Chandrakasan, M. Potkonjak, J. Rabaey, and R. W. Brodersen, "HYPER-LP: A System for Power Minimization Using Architectural Transformation," in *Proceedings of the IEEE International Conference on Computer Aided Design*, 1992, pp. 300–303.
20. N. Kumar, S. Katkoori, L. Rader, and R. Vemuri, "Profile-Driven Behavioral Synthesis for Low Power VLSI Systems," unpublished.
21. P. Van Oostende, P. Six, J. Vandewalle, and H. De Man, "Estimation of Typical Power of Synchronous {CMOS} Circuits Using a Hierarchy of Simulators," *J. Solid State Circuits, 28*(1), 26–39 (1993).
22. B. J. George, D. Gossain, S. C. Tyler, M. G. Wloka, and G. K. H. Yeap, "Power Analysis and Characterization for Semi-Custom Design," in *Proceedings of the 1994 International Workshop on Low Power Design*, 1994, pp. 215–218.
23. M. Xakellis and F. Najm, "Statistical Estimation of Switching Activity in Digital Circuits," in *Proceedings of the 31st Design Automation Conference*, 1994, pp. 728–733.

24a. P. E. Landman and J. Rabaey, "Power Estimation for High Level Synthesis," in *Proceedings of the European Conference on Design Automation*, 1993, pp. 361–366.

24b. S. R. Powell and P. M. Chau, "A Model for Estimating Power Dissipation in a Class of DSP VLSI Chips," *IEEE Trans. Circuits Syst., CS-36*(6), 646–650 (1995).

25. C. Svensson and D. Liu, "A Power Estimation Tool and Prospects of Power Savings in CMOS VLSI Chips," in *Proceedings of the 1994 International Workshop on Low Power Design*, 1994, pp. 171–176.
26. J-M. Chang and M. Pedram, "Low Power Register Allocation and Binding," in *Proceedings of the 32nd Design Automation Conference*, 1995, pp. 29–35.
27. J-M. Chang and M. Pedram, "Power Efficient Module Allocation and Binding," CENG Technical Report 95-16, University of Southern California (June 1995).
28. D. Marculescu, R. Marculescu, and M. Pedram, "Information Theoretic Measures for Energy Consumption at Register Transfer Level," in *Proceedings of the 1995 International Symposium on Low Power Design*, 1995, pp. 81–86.
29. F. N. Najm, "Towards a High-Level Power Estimation Capability," in *Proceedings of the 1995 International Symposium on Low Power Design*, 1995, pp. 87–92.
30. P. E. Landman and J. Rabaey, "Activity Sensitive Architectural Power Analysis for Control Path," in *Proceedings of the 1995 International Symposium on Low Power Design*, 1995, pp. 93–98.
31. K. P. Parker and J. McCluskey, "Probabilistic Treatment of General Combinational Networks," *IEEE Trans. Computers, C-24*, 668–670 (1975).
32. S. Chakravarty, "On the Complexity of Using BDDs for the Synthesis and Analysis of Boolean Circuits," in *Proceedings of the 27th Annual Allerton Conference on Communication, Control and Computing*, 1989, pp. 730–739.
33. R. Bryant, "Graph-Based Algorithms for Boolean Function Manipulation," *IEEE Trans. Computers, C-35*, 677–691 (1986).
34. S. Ercolani, M. Favalli, M. Damiani, P. Olivo, and B. Ricco, "Estimate of Signal Probability in Combinational Logic Networks," in *First European Test Conference*, 1989, pp. 132–138.
35. P. Schneider and U. Schlichtmann, "Decomposition of Boolean Functions for Low Power Based on a New Power Estimation Technique," in *Proceedings of the 1994 International Workshop on Low Power Design*, 1994, pp. 123–128.

36. R. Marculescu, D. Marculescu, and M. Pedram, "Logic Level Power Estimation Considering Spatiotemporal Correlations," in *Proceedings of the IEEE International Conference on Computer Aided Design*, 1994, pp. 294–299.
37. A. Ghosh, S. Devadas, K. Keutzer, and J. White, "Estimation of Average Switching Activity in Combinational and Sequential Circuits," in *Proceedings of the 29th Design Automation Conference*, 1992, pp. 253–259.
38. R. Burch, F. Najm, P. Yang, and D. Hocevar, "Pattern Independent Current Estimation for Reliability Analysis of CMOS Circuits," in *Proceedings of the 25th Design Automation Conference*, 1988, pp. 294–299.
39. F. N. Najm, R. Burch, P. Yang, and I. Hajj, "Probabilistic Simulation for Reliability Analysis of CMOS VLSI Circuits," *IEEE Trans. Computer-Aided Design Integrated Circuits Syst., 9*(4), 439–450 (1990).
40. C-Y. Tsui, M. Pedram, and A. M. Despain, "Efficient Estimation of Dynamic Power Dissipation Under a Real Delay Model," in *Proceedings of the IEEE International Conference on Computer Aided Design*, 1993, pp. 224–228.
41. C-S. Ding and M. Pedram, "Tagged Probabilistic Simulation Provides Accurate and Efficient Power Estimates at the Gate Level," in *Proceedings of the Symposium on Low Power Electronics*, 1995.
42. C-Y. Tsui, M. Pedram, and A. M. Despain, "Exact and Approximate Methods for Calculating Signal and Transition Probabilities in Finite State Machines," in *Proceedings of the 31st Design Automation Conference*, 1994, pp. 18–23.
43. J. Monteiro, S. Devadas, and A. Ghosh, "Estimation of Switching Activity in Sequential Logic Circuits with Applications to Synthesis for Low Power," in *Proceedings of the 31st Design Automation Conference*, 1994, pp. 12–17.
44. H. M. Lieberstein, *A Course in Numerical Analysis,* Harper & Row, New York, 1968.
45. C-Y. Tsui, J. Monteiro, M. Pedram, S. Devadas, A. M. Despain, and B. Lin, "Power Estimation in Sequential Logic Circuits," *IEEE Trans. VLSI Syst., 3*(3), 404–416 (1995).
46. B. Davari, R. H. Dennard, and G. G. Shahidi, "CMOS Scaling for High Performance and Low Power," *Proc. IEEE, 83*(4), 408–425 (1995).
47. A. Raghunathan and N. K. Jha, "Behavioral Synthesis for Low Power," in *Proceedings of the IEEE International Conference on Computer Design*, 1994, pp. 318–322.
48. A. Raghunathan and N. K. Jha, "An ILP Formulation for Low Power Based on Minimizing Switched Capacitance During Data Path Allocation," *Proceedings of the IEEE International Symposium on Circuits and Systems*, 1995.
49. S. Raje and M. Sarrafzadeh, "Variable Voltage Scheduling," in *Proceedings of the 1995 International Symposium on Low Power Design*, 1995, pp. 9–13.
50. C-Y. Tsui, M. Pedram, C-H. Chen, and A. M. Despain, "Low Power State Assignment Targeting Two- and Multi-Level Logic Implementations," in *Proceedings of the IEEE International Conference on Computer Aided Design*, 1994, pp. 82–87.
51. J. Monteiro, S. Devadas, and A. Ghosh, "Retiming Sequential Circuits for Low Power," in *Proceedings of the IEEE International Conference on Computer Aided Design*, 1993, pp. 398–402.
52. M. Alidina, J. Monteiro, S. Devadas, A. Ghosh, and M. Papaefthymiou, "Precomputation-Based Sequential Logic Optimization for Low Power," in *Proceedings of the 1994 International Workshop on Low Power Design*, 1994, pp. 57–62.
53. S. Iman and M. Pedram, "Logic Extraction and Decomposition for Low Power," in *Proceedings of the 32nd Design Automation Conference*, 1995, pp. 248–253.
54. S. Iman and M. Pedram, "Multi-Level Network Optimization for Low Power," in *Proceedings of the IEEE International Conference on Computer Aided Design*, 1994, pp. 372–377.
55. S. Iman, C. Y. Tsui, and M. Pedram, "PLA Minimization for Low Power VLSI

Designs," CENG Technical Report, Dept. of EE-Systems, University of Southern California (April 1995).
56. M. Berkelaar and J. Jess, "Gate Sizing in MOS Digital Circuits with Linear Programming," in *Proceedings of the European Design Automation Conference*, 1990, pp. 217–221.
57. D. Zhou and X. Y. Liu, "Optimal Drivers for High Speed Low Power ICs," *Int. J. of High Speed Elec., 7*, 1996.
58. J. Cong, C-K. Koh, and K-S. Leung, "Simultaneous Driver and Wire Sizing for Performance and Power Optimization," *IEEE Trans. VLSI Syst., 2*(4), 408–425 (1994).
59. J. Cong and C-K. Koh, "Minimum-Cost Bounded-Skew Clock Routing," in *Proceedings of the International Symposium on Circuits and Systems*, 1995, pp. 215–218.
60. D. J. Huang, A. B. Kahng, and C.W. Tsao, "On the Bounded-Skew Clock and Steiner Tree Problems," in *Proceedings of the 32nd Design Automation Conference*, 1995, pp. 508–513.
61. W. C. Athas, L. J. Svensson, J. G. Koller, N. Thartzanis, and E. Chou, "Low-Power Digital Systems Based on Adiabatic-Switching Principles," *IEEE Trans. VLSI Syst., 2*(4), 398–407 (1994).
62. L. S. Nielsen, C. Niessen, J. Sparso, and C. H. van Berke, "Low-Power Operation Using Self-Timed Circuits and Adaptive Scaling of the Supply Voltage," *IEEE Trans. VLSI Syst., 2*(4), 391–397 (1994).
63. M. Hahm, "Modest Power Savings for Applications Dominated by Switching of Large Capacitive Loads," in *Proceedings of the 1995 IEEE Symposium on Low Power Electronics*, 1995, pp. 60–61.
64. L. R. Carley and I. Lys, "QuadRail: A Design Methodology for Low Power ICs," *IEEE Trans. VLSI Syst., 2*(4), 383–390 (1994).
65. J. Rabaey and M. Pedram (eds.), *Low Power Design Methodologies*, Kluwer Academic Publishers, New York, 1996.

MASSOUD PEDRAM