To appear in Encyclopedia of Computer Science and Technology, 1995

Design Technologies for Low Power VLSI

Massoud Pedram

Department of EE-Systems
University of Southern California
Los Angeles , CA 90089

Abstract

Low power has emerged as a principal theme in today's electronics industry. The need for low power has caused a major paradigm shift where power dissipation has become as important a consideration as performance and area. This article reviews various strategies and methodologies for designing low power circuits and systems. It describes the many issues facing designers at architectural, logic, circuit and device levels and presents some of the techniques that have been proposed to overcome these difficulties. The article concludes with the future challenges that must be met to design low power, high performance systems.

1. Motivation

In the past, the major concerns of the 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 watt-hours/kilogram [52], this terminal would require an unacceptable 6 kilograms of batteries for 10 hours 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 watt-hours/kilogram over the next 5 years [52] which still leads to an unacceptable 3.6-4.4 kilograms of battery cells. In the absence of low-power design techniques then, current and future portable devices will suffer from either very short battery life or very heavy battery pack.

There also exists a 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 [20]! 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. Since 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 [63]. 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 volage 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 micro-powered 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, set-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 next. If extending the battery life is the only concern, then the energy (that is, 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 [26]. In this case one can alternatively minimize the energy/delay ratio (that is, 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.

2. Sources of Power Dissipation

Power dissipation in 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 picoA for a 1 micro-meter 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 instead of being independent of V_{DS} (see [22] 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 [79]. 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, 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:

$$P = 0.5CV_{dd}^{2}E(sw)f_{clk} \tag{1}$$

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.

3. 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.

3.1. 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 two reduction in supply voltage yields a factor of four decrease in power consumption. Furthermore, this power reduction is a global effect, experienced not only in one sub-circuit 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 [11], 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 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. Since the inverse threshold slope (S) of a 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. Anther 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 plus/minus 0.15 Vat 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 rduce the V_t fluctuation to plus/minus 0.05 V at V_{dd} of 1 V [33].

3.2. 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 [49] or by using stochastic / procedural interconnect models [50]. Interconnect capacitance estimation after layout is straight-forward and in general accurate.

With this understanding, we can now consider how to reduce physical capacitance. From the previous discussion, we recognize that capacitances can be kept at a minimum by using less logic, smaller devices, 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 sub-function 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.

3.3. 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(sw)* which determines how many transitions each arrival will generate. For circuits that do not experience glitching, *E(sw)* can be interpreted as the probability that a power consuming transition will occur during a single data period. Even for these circuits, calculation of *E(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 [41].

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. Since glitching can cause a node to make several power consuming transitions, it should be avoided whenever possible.

The data activity *E(sw)* can be combined with the physical capacitance *C* to obtain *switched capacitance*, $C_{sw}=C.E(sw)$, 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.

3.4. 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.

3.4.1 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 $E_g(sw)$=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 $E_g(sw)$=1/2. Alternatively, assume that it is known that every 0 applied to input i is immediately followed by a 1 while every 1 applied to input j is immediately followed by a 0, then $E_g(sw)$=4/9. Finally, assume that it is known that i changes exactly if j changes value, then $E_g(sw)$=1/4. The first case is an example of *spatial* correlations between gate inputs, the second case illustrates *temporal* correlations on gate inputs while the third case describes an instance of *spatiotemporal* correlations.

The straight-forward 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 criterion for specified relative or absolute error in power estimates for a given confidence level [8].

3.4.2 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 as the f*unctional activity* while the latter is referred as the *spurious activity.* It is shown in [4] 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 3-4 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 (non-glitch) 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 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.

3.4.3 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 while 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.

3.4.4 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 pseudo-random 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 activites 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!

3.4.5 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 (that is, 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 mehods (i.e., symbolic simulation) have also been proposed, but are impractical due to excessive time and memory requirements.

3.4.6 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 IC parameter fluctuations than the power required to perform the transition between the initial and final state of each node.

4. Power Estimation Techniques

The design for 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 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.

4.1. Simulative Approaches

4.1.1 Brute-force simulation

Circuit simulation based techniques ([32],[75]) 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 and multi-phase clocking methodologies, tristate drives, etc. 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 [54].

PowerMill [18] 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 [59]) can be easily modified to report the switched capacitance (and thus dynamic power dissipation) during a simulation run.

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 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 3-4 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 [12] [34]. Instruction level simulation or behavioral simulators are easily (and have indeed been) adapted to produce this information.

4.1.2 Hierarchical simulation

A simulation method based on a hierarchy of simulators is presented in [78]. The idea is to use a hierarchy of power simulators (for example, at architectural, gate-level and circuit-level) to achieve a reasonable accuracy and efficiency tradeoff. Another good example is Entice-Aspen [23]. 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.

4.1.3 Monte Carlo simulation

A *Monte Carlo simulation* approach for power estimation which alleviates the input pattern dependence problem has been proposed in [8]. 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 [80]. The method does not handle spatial correlations at the circuit inputs.

4.2. Non-simulative Approaches

4.2.1 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 pseudo-random 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 pseudo-random 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. Multi-parameter modules are characterized with respect to all the parameters, yielding a multi-parameter equation or table. Multi-function modules (e.g., 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 [51] and [34], 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 [35], 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 [65], 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 [13][14] 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 [42] and [46] which relies on information theoretic measures of activity (for example, entropy) to devise fast, yet accurate, power estimation at the algorithmic and structural behavioral levels. In the following, the approach presented in [42] 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 the 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 that it 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 [36][34]. These techniques provide quick estimation of the power dissipation in a controller based on the knowledge of its target implementation style (that is, precharged pseudo-NMOS or dynamic 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 squared fit error analysis on real data.

4.2.2 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 $E_n(sw)$ at a circuit node n involve estimation of signal probability $prob(n)$, which is the probability that the signal value at the node is one. Under the assumption that the values applied to each circuit input are temporally independent (that is, value of any input signal at time t is independent of its value at time $t\text{-}1$), we can write:

$$E_n(sw) = 2\, prob(n)\, (1 - prob(n)). \tag{2}$$

Computing signal probabilities has attracted much attention [48], [10]. In the recent years, a computational procedure based on Ordered Binary-Decision Diagrams (OBDDs) [6] 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:

$$prob(y) = prob(x)\, prob(f_x) + prob(\bar{x})\, prob(f_{\bar{x}}) \tag{3}$$

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

In [21], 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 [61] and [40], 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 edge *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 $prob\ (x_{i \rightarrow j})$ is equal to $prob\ (x = i)\ prob_{ij}^{x}$. Obviously, $prob_{00}^{x} + prob_{01}^{x} = prob_{10}^{x} + prob_{11}^{x} = 1$ while $prob\ (x) = prob\ (x_{0 \rightarrow 1}) + prob\ (x_{1 \rightarrow 1})$ and $prob\ (\bar{x}) = prob\ (x_{0 \rightarrow 0}) + prob\ (x_{1 \rightarrow 0})$. The activity factor of line *x* can be expressed in terms of these transition probabilities as follows:

$$E_x(sw) = prob\,(x_{0 \rightarrow 1}) + prob\,(x_{1 \rightarrow 0}) \ . \tag{4}$$

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 [40] 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 spatio-temporal correlations. This work has been extended to handle highly correlated input streams using the notions of *conditional independence* and *isotropy of signals* [41]. 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 [24], 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 that this method is applicable to, 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 [7]. 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 straight-forward to calculate the switching activity of *x* which includes the contribution of hazards and glitches, that is:

$$E_x(sw) = \sum_{t \in eventlist(x)} \left(prob\left(x^t_{0 \to 1} \right) + prob\left(x^t_{1 \to 0} \right) \right). \tag{5}$$

Given such waveforms at the circuit inputs and with some convenient partitioning of the circuit, the authors examine every sub-circuit and derive the corresponding waveforms at the internal circuit nodes. In [45], 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 [71] 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 [19]. 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 macro-modeling data for the gates in the cell library.

4.2.3 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 (that is, 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 [24]. 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 [73] and [44] 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, since it requires the explicit consideration of each state in the circuit.

The authors of [73] and [44] also describe a method for approximate switching activity estimation of sequential circuits. The basic computation step is the solution of a non-linear system of equations in terms of the present state bit probabilities and signal probabilites 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 [38]. Increasing the number of variables or the number of equations in the above system results in increased accuracy [69]. 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.

5. 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 SOI with partially or fully depleted wells, CMOS scaling to submicron device sizes, and advanced interconnect substrates such as Multi-Chip 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.

5.1. 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 [17], two CMOS device and voltage scaling scenarios are described, one optimized for the highest speed and one trading off high performance 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 7x 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 micrometer region as compared with today's high performance 0.6 micrometer devices at 5 volts. This paper also presents a discussion of how high the electric field in a transistor channel can go without impacting the long term device reliability, while at the same time 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 3x 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.

5.2. 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.

5.2.1 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.

5.2.2 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 [12]. 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; 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 [14], [13], [56] and [57]

In [53], an exact (graph-theoretic) algorithm for minimizing the system power through variable-voltage scheduling is presented. The idea is 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 excecuted using minimum possible supply voltage).

5.2.3 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, multi-level 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 multi-level, unmapped versus mapped, ASICs versus 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 [70]; Latches in a pipelined design can be repositioned to eliminate hazardous activity in the circuit [43]; 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 [1]; Common sub-expressions with low transition probability values can be extracted [29]; 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 [28]; Nodes with high switching activity may be hidden inside CMOS gates where they drive smaller physical capacitances [74]; 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; Circuit depth and power dissipation may be simultaneously minimized using a node clustering appraoch ; PLAs can be implemented to reduce static or dynamic power dissipation in pseudo-NMOS or dynamic NOR-NOR implementations [31]. Power dissipation may be further reduced by gate resizing [5], signal-to-pin assignment and I/O encoding.

5.2.4 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, multi-chip 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, 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. This is however a very difficult analysis and optimization problem and requires further research.

It should be noted that by applying post-layout optimization techniques (such as buffer and wire sizing, local restructuring and re-mapping, 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, floorplanning, placement [76] and routing; Individual transistors may be sized down to reduce the power dissipation along the non-critical 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 [81]; Wire and driver sizing may be combined to reduce the interconnect delay with only a small increase in the power dissipation [15]; Clock trees may be constructed that minimize the load on the clock drivers subject to meeting a tolerable clock skew [16] [27].

5.2.5 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 [2]. 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 [47], 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 [25], 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 [9], show good signs . Design of energy efficient level-converters and DC/DC converters is also essential to the success of adaptive supply voltage strategies.

5.3. 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.

6. Challenges Ahead

The need for lower 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 5-10x in three 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.

- A low voltage/low threshold technology and circuit design approach, targeting supply voltages around 1 Volt 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 sub-circuits whose energy levels can be independently controlled and by powering down sub-circuits 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 processor with instruction set, data path width and functional units best suited to algorithm, mapping functions to hardware so that inter-chip 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.

7. References

[1] 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*, pages 57-62, April 1994.

[2] W.C.Athas, L. J. Svensson, J.G.Koller, N.Thartzanis and E. Chou. " Low-Power Digital Systems Based on Adiabatic-Switching Principles. " IEEE *Transactions on VLSI Systems*, 2(4)398-407:, December 1994

[3] H. Bakoglu, *Circuits, Interconnections, and Packaging for VLSI*, Addison-Wesley, Menlo Park, CA, 1990.

[4] 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*, pages 27–32, April 1994.

[5] M. Berkelaar and J. Jess. " Gate sizing in MOS digital circuits with linear programming. " In *Proceedings of the European Design Automation Conference*, pages 217-221, 1990.

[6] R. Bryant. " Graph-based algorithms for Boolean function manipulation. " IEEE *Transactions on Computers*, volume C-35, pages 677–691, August 1986.

[7] 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*, pages 294–299, June 1988.

[8] R. Burch, F. N. Najm, P. Yang, and T. Trick. " A Monte Carlo approach for power estimation. " IEEE *Transactions on VLSI Systems*, 1(1):63–71, March 1993.

[9] L. Richard Carley and Ihor Lys. " QuadRail: A design methodology for low power ICs. " IEEE *Transactions on VLSI Systems*, 2(4):383-390, December 1994.

[10] 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*, pages 730–739, 1989.

[11] A. Chandrakasan, S. Sheng, and R. W. Brodersen, "Low-power CMOS design. " *IEEE Journal of Solid-State Circuits*, pages 472-484, April 1992.

[12] A. Chandrakasan, M. Potkonjak, J. Rabaey and R. W. Brodersen, " HYPER-LP: A System for Power Minimization Using Architectural Transformation. "In *Proceedings of the I EEE International Conference on Computer Aided Design*, pages 300-303, November 1992.

[13] J-M. Chang and M. Pedram. " Low power register allocation and binding. " In *Proceedings of the 32nd Design Automation Conference*, pages 29-35, June 1995.

[14] J-M. Chang and M. Pedram. " Power efficient module allocation and binding. " *CENG Technical Report 95-16*, University of Southern California, June 1995.

[15] J. Cong, C-K. Koh and K-S. Leung. " Simultanous driver and wire sizing for performance and power optimization. " IEEE *Transactions on VLSI Systems*, 2(4):408-425, December 1994.

[16] J. Cong and C-K. Koh. " Minimum-cost bounded-skew clock routing. " In *Proceedings of the International Symposium on Circuits and Systems*, pages 215-218, 1995.

[17] B. Davari, R. H. Dennard and G. G. Shahidi. " CMOS scaling for high performance and low power.. " *Proceedings of* IEEE, 83(4):408-425, April 1995.

[18] C. Deng. " Power analysis for CMOS/BiCMOS circuits. " In *Proceedings of the 1994 International Workshop on Low Power Design*, pages 3–8, April 1994.

[19] 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*, September 1995.

[20] D. Dobberpuhl et al, " A 200MHz, 64b, dual issue CMOS microprocessor. "*Digest of Technical Papers, ISSC ‘92*, pages 106-107, 1992.

[21] S. Ercolani, M. Favalli, M. Damiani, P. Olivo, and B. Ricco. "Estimate of signal probability in combinational logic networks. " In *First European Test Conference*, pages 132–138, 1989.

[22] T. A. Fjeldly and M. Shur. " Threshold voltage modeling and the subthreshold regime of operation of short-channel MOSFET’s. " IEEE *Transactions on Electron Devices*, 40(1):137–145, Jan. 1993.

[23] 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*, pages 215–218, April 1994.

[24] 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*, pages 253–259, June 1992.

[25] M. Hahm. " Modest power savings for applications dominated by switchning of large capacitive loads. " In *Proceedings of the 1995 IEEE Symposium on Low Power Electronics*, pages 60-61, October 1995

[26] M. Horowitz, T. Indermaur and R. Gonzalez. " Low-power digital design. " In *Proceed-*

ings of the 1995 IEEE Symposium on Low Power Electronics, pages 8-11, October 1995.

[27] 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*, pages 508-513, June 1995.

[28] S. Iman and M. Pedram. " Multi-level network optimization for low power. " In *Proceedings of the* I EEE *International Conference on Computer Aided Design*, pages 372–377, November 1994.

[29] S. Iman and M. Pedram. " Logic extraction and decomposition for low power. " In *Proceedings of the 32nd Design Automation Conference*, June 1995.

[30] S. Iman and M. Pedram. " Two level logic minimization for low power. " In *Proceedings of the* I EEE *International Conference on Computer Aided Design*, pages 372–377, November 1994.

[31] 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.

[32] S. M. Kang. " Accurate simulation of power dissipation in VLSI circuits. " IEEE *Journal of Solid State Circuits*, 21(5):889–891, Oct. 1986.

[33] T. Kobayashi and T. Sakurai. " Self-adjusting threshold-voltage scheme for low voltage high speed operation. " Proceedings of CICC, pages 271-274, May 1994.

[34] N. Kumar, S. Katkoori, L. Rader and R. Vemuri. " Profile-driven behavioral synthesis for low power VLSI systems. "*To appear*, 1995.

[35] P.E. Landman and J. Rabaey. " Power estimation for high level synthesis. " In *Proceedings of the European Conference on Design Automation*, pages 361-366, February 1993.

[36] P.E. Landman and J. Rabaey. " Activity sensitive architectural power analaysis for control path. " In *Proceedings of the 1995 International Symposium on Low Power Design*, pages 93-98, April 1995.

[37] C. Lennard and A. R. Newton. " An estimation technique to guide low power resynthesis algorithms. " In *Proceedings of the 1995 International Symposium on Low Power Design*, pages 227-232, April 1995.

[38] H. M. Lieberstein. " *A Course in Numerical Analysis*. " Harper & Row Publishers, 1968.

[39] B. Lin and H. De Man. " Low-power driven technology mapping under timing constraints " In *Proceedings of the International Conference on Computer Design*, pages 421-427, October 1993.

[40] 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*, pages 294–299, November 1994.

[41] R. Marculescu, D. Marculescu, and M. Pedram. " Efficient power estimation for highly correlated input streams. " In *Proceedings of the 32nd Design Automation Conference*, pages 628-634, June 1995.

[42] 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*, pages 81-86, April 1995.

[43] J. Monteiro, S. Devadas, and A. Ghosh. " Retiming sequential circuits for low power. " In *Proceedings of the* IEEE *International Conference on Computer Aided Design*, pages 398–402, November 1993.

[44] 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*, pages 12-17 , June 1994.

[45] F. N. Najm, R. Burch, P. Yang, and I. Hajj. " Probabilistic simulation for reliability analysis of CMOS VLSI circuits. " IEEE *Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 9(4):439–450, April 1990.

[46] F. N. Najm. " Towards a high-level power estimation capability. " In *Proceedings of the 1995 International Symposium on Low Power Design*, pages 87-92, April 1995.

[47] 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 *Transactions on VLSI Systems*, 2(4):391-397, December 1994.

[48] K. P. Parker and J. McCluskey. " Probabilistic treatment of general combinational networks. " IEEE *Transactions on Computers*, C-24:668–670, Jun. 1975.

[49] M. Pedram and N. Bhat. " Layout driven technology mapping. " In *Proceedings of the 28th Design Automation Conference,* pages 99-105, June 1991.

[50] M. Pedram, B. T. Preas. " Interconnection length estimation for optimized standard cell layouts. " In *Proceedings of the* IEEE *International Conference on Computer Aided Design*, pages 390-393, November 1989.

[51] S. R. Powell and P. M. Chau. " A model for estimating power dissipation in a class of DSP VLSI chips. " IEEE *Transactions on Circuits and Systems*, 36(6):646-650, June 1995.

[52] R. A. Powers. " Batteries for low power electronics. " *Proceedings of* IEEE, 38(4):687-693, April 1995.

[53] S. Raje and M. Sarrafzadeh. " Variable voltage scheduling. " In *Proceedings of the 1995 International Symposium on Low Power Design*, pages 9-13, April 1995.

[54] S. Rajgopal and G. Mehta. " Experiences with simulation-based schematic level current estimation. " In *Proceedings of the 1994 International Workshop on Low Power Design*, pages 9–14, April 1994.

[55] J. Rajski and J. Vasudevamurthy. " The testability-preserving concurrent decomposition and factorization of Boolean expressions. " IEEE *Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 11(6):778–793, June 1993.

[56] A. Raghunathan and N. K. Jha. " Behavioral synthesis for low power. " In *Proceedings of the* IEEE *International Conference on Computer Design,* pages 318-322, October 1994.

[57] A. Raghunathan and N. K. Jha. " An ILP formulation for low power based on minimizing switched capacitance during data path allocation. " *To appear*, 1995.

[58] K. Roy and S. C. Prasad. " Circuit activity based logic synthesis for low power reliable operations. " IEEE *Transactions on VLSI Systems*, 1(4):503–513, December 1993.

[59] A. Salz and M. A. Horowitz. " IRSIM: An incremental MOS switch-level simulator. " In *Proceedings of the 26th Design Automation Conference*, pages 173-178, June 1989.

[60] H. Savoj, R. K. Brayton, and H. J. Touati. " Extracting local don't cares for network optimization. " In *Proceedings of the* IEEE *International Conference on Computer Aided Design*, pages 514–517, November 1991.

[61] 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*, pages 123–128, April 1994.

[62] A. A. Shen, A. Ghosh, S. Devadas, and K. Keutzer. " On average power dissipation and random pattern testability of CMOS combinational logic networks. " In *Proceedings of the* IEEE *International Conference on Computer Aided Design*, November 1992.

[63] C. Small, " Shrinking devices put the squeeze on system packaging. "*EDN*, vol. 39, no. 4, pages 41-46, Feb. 17, 1994.

[64] A. Stratakos, R. W. Brodersen, and S. R.Sanders, " High-efficiency low-voltage DC-DC conversion for portable applications. "*1994 International Workshop on Low-Power Design*, Napa Valley, CA, April 1994.

[65] 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*, pages 171-176, April 1994.

[66] V. Tiwari, P. Ashar, and S. Malik. " Technology mapping for low power. " In *Proceedings of the 30th Design Automation Conference*, pages 74-79, June 1993.

[67] V. Tiwari, S. Malik and P. Ashar. " Guarded evaluation: Pushing power management to logic synthesis/design. " In *Proceedings of the 1995 International Symposium on Low Power Design*, pages 221-226, April 1995.

[68] C-Y.Tsui. *Power analysis and optimization for CMOS circuits.* PhD Dissertation, Computer Engineering, University of Southern California, 1994.

[69] C-Y. Tsui, J. Monteiro, M. Pedram, S. Devadas, A. M. Despain and B. Lin. " Power estimation in sequential logic circuits. " IEEE *Transactions on VLSI Systems, September 1995.*

[70] 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*, pages 82–87, November 1994.

[71] 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*, pages 224–228, November 1993.

[72] C-Y. Tsui, M. Pedram, and A. M. Despain. " Technology decomposition and mapping targeting low power dissipation. " In *Proceedings of the 30th Design Automation Conference*, pages 68–73, June 1993.

[73] C-Y. Tsui, M. Pedram, and A. M. Despain. " Exact and approximate methods for calculating signal and transition probabilities in fsms. " In *Proceedings of the 31st Design*

Automation Conference, pages 18–23, June 1994.

[74] C-Y. Tsui, M. Pedram, and A. M. Despain. " Power efficient technology decomposition and mapping under an extended power consumption model. " IEEE *Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 13(9), September 1994.

[75] A. Tyagi. " Hercules: A power analyzer of MOS VLSI circuits. " In *Proceedings of the* IEEE *International Conference on Computer Aided Design*, pages 530–533, November 1987.

[76] H. Vaishnav and M. Pedram. " PCUBE: A performance driven placement algorithm for low power designs. " In *Proceedings of the European Design Automation Conference*, pages 72-77, September 1993.

[77] H. Vaishnav and M. Pedram. "Delay optimal partitioning targeting low power VLSI circuits. " In *Proceedings of the* IEEE *International Conference on Computer Aided Design*, November 1995.

[78] P. Van Oostende, P. Six and J. Vandewalle and H. De Man. " Estimation of typical power of synchronous {CMOS} circuits using a hierarchy of simulators. " IEEE *Journal of Solid State Circuits*, 28(1):26-39, January 1993.

[79] H. J. M. Veendrick. " Short-circuit dissipation of static CMOS circuitry and its impact on the design of buffer circuits. " IEEE *Journal of Solid State Circuits*, 19(8):468–473, August 1984.

[80] M. Xakellis and F. Najm. " Statistical estimation of switching activity in digital circuits. " In *Proceedings of the 31st Design Automation Conference*, pages 728-733, June 1994.

[81] D. Zhou and X. Y. Liu. " Optimal drivers for high speed low power ICs. " *To appear*, 1995..