Power Minimization in IC Design: Principles and Applications

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 in which power dissipation is as important as performance and area. This article presents an in-depth survey of CAD methodologies and techniques for designing low power digital CMOS circuits and systems and describes the many issues facing designers at architectural, logic and physical levels of design abstraction. It reviews some of the techniques and tools that have been proposed to overcome these difficulties and outlines the future challenges that must be met to design low power, high performance systems.

1. Introduction

In the past, the major concerns of the VLSI designer were area, performance, cost and reliability; power considerations were 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. 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 consumption for a 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 yielding around 65 watt-hours/kilogram [113], 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 [113] 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 a very short battery life or a 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 cock rates [39]. 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 huge. Since core power consumption must be dissipated through the packaging, increasingly expensive packaging and cooling strategies are required. Unless power consumption is dramatically reduced, the resulting heat will limit the feasible packing and performance of VLSI circuits and systems. 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; at the same time, high temperature tends to exacerbate several silicon failure mechanisms. Every 10 °C increase in operating temperature roughly doubles failure rate for the components [133]. In this context, peak power (maximum possible power dissipation) is a critical design factor because 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.

From the environmental viewpoint, the smaller the power dissipation of electronic systems, the lower the heat pumped into the rooms, the lower the elec-

tricity consumed and therefore, the less the impact on global environment, the less the office noise (due to elimination of a fan from the desktop), and the less stringent the environment/office power delivery and cooling 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, non-battery operated systems, such as workstations, set-top computers and multimedia information processing and communication systems, the overall goal of power minimization is to reduce system cost (cooling, packaging and energy bill) and ensure long-term circuit 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.

Our goal in writing this paper is to provide background and outlook for people interested in using or developing low power design methodologies and techniques. Even though we tried to be complete, some research work might have been unintentionally left out. In addition, the description of various techniques may be perceived as uneven at times because of the amount of coverage given to certain topics; this is mainly due to our experience in using these methods for building our power optimization and synthesis system, POSE.

The paper is organized as follows. First, we describe sources of power dissipation in CMOS circuits and degrees of freedom in the low power design space. We then present an in-depth survey (and in many cases analysis) of power estimation and minimization techniques and describe some of the frontiers of the research currently being pursued. We conclude by summarizing the major low power design challenges that lie ahead.

2. Sources of Power Dissipation

Power dissipation in digital CMOS circuits is caused by four sources as follows.

- the *leakage current*, which is primarily determined by the fabrication technology, consists of two components: 1) reverse bias current in the parasitic diodes formed between source and drain diffusions and the bulk region in a MOS transistor, and 2) the subthreshold current that arises from the inversion charge that exists at the gate voltages below the threshold voltage,
- the *standby current* which is the DC current drawn continuously from V_{dd} to ground,
- the *short-circuit (rush-through) current* which is due to the DC path between the supply rails during output transitions,
- the *capacitance current* which flows to charge and discharge capacitive loads during logic changes.

The diode leakage is proportional to the area of the source or drain diffusion and the leakage current density and is typically 1 *picoA* for a 1 *micron* 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 transistor 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 [44] for a recent analysis).

The standby power consumption happens, for example, when both the nMOS and pMOS transistors are continuously on in a pseudo-nMOS inverter, when the drain of an nMOS transistor is driving the gate of another nMOS transistor in a pass-transistor logic, or when the tristated input of a CMOS gate leaks away to a value between V_{dd} and ground. The standby power is equal to the product of V_{dd} and the DC current drawn from the power supply to ground.

The term *static power dissipation* refers to the sum of leakage and standby dissipations. Leakage currents in CMOS circuits can be made small with proper choice of device technology. Standby currents are important in CMOS design styles like pseudo-nMOS and nMOS pass transistor logic and in memory cores. In this article, we assume that the standby dissipation is insignificant, thus limiting ourselves to CMOS technologies, logic styles and circuit structures [63] in which this condition holds.

The short-circuit power consumption for an inverter gate is proportional to the input ramp time, the load and transistor sizes of the gate. The maximum short circuit current flows when there is no load; this current decreases with the load. Depending on the approximations used to model the currents and to estimate the input signal dependency, different formulae [161] [52], with varying accuracy, have been derived for the evaluation of the short circuit power. A useful formula was recently derived in [155] that shows the explicit dependence of the short circuit power dissipation on the design and performance parameters, such as transistor sizes, input and output ramp times and the load. The idea is to adopt an alternative definition of the short circuit power dissipation, through an equivalent (virtual) short circuit capacitance C_{SC}.

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 [161]. If, however, design for high performance is taken to the extreme where large gates are used to drive relatively small loads and if the input ramp time is long, then there will be a stiff penalty in terms of short-circuit power consumption.

The dominant source of power dissipation CMOS circuits is the charging and discharging of the node capacitances (also referred to as the capacitive power dissipation) and is given by:

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

where C_L is the physical capacitance at the output of the node, V_{dd} is the supply voltage, $E(sw)$ (referred to as the *switching activity*) is the average number of output transitions per $1/f_{clk}$ time, and f_{clk} is the clock frequency. The product of $E(sw)$ and f_{clk} which is the number of transitions per second, is referred to as the *transition density* in [101].

The term *dynamic power dissipation* refers to the sum of short circuit and capacitive dissipations. Using the concept of equivalent short-circuit capacitance described above, the dynamic power dissipation can be calculated using equation (1) if we add C_{SC} to C_L. Short-circuit currents in CMOS circuits can be made small with appropriate circuit design techniques. In most of this article, we will thus focus on capacitive power dissipation.

3. Low Power Design Space

The previous section alluded to 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; a factor of two reduction in supply voltage gives a factor of four decrease in power consumption. Furthermore, this power reduction is a global effect that is experienced throughout the entire design. In some cases designers are thus 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 two to three times V_t.

One 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 gain of the CMOS gates at low supply voltage regime and control of the leakage currents. Since the inverse threshold slope (S) of a MOSFET is invariant with scaling [36], for every 80-100 mV (based on the operating temperature) reduction in V_t, the subthreshold current will be increased by one order of magnitude. As a rule, the “off-current” current should remain two to three orders of magnitude smaller than the “on-current”. 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 changes by 3x for a ΔV_{dd} of plus/minus 0.15 V when V_{dd} equals 1 V. Such a large variation in nominal delay values cannot be tolerated. This sets 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 that reduces the V_t fluctuation to plus/minus 0.05 V when V_{dd} equals 1 V [69].

3.2. Physical Capacitance

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 behavioral or logical levels of abstraction is difficult and imprecise because it requires estimation of the load capacitances from structures which are not yet mapped to gates in a cell library. Pre-characterizing the operational modules (such as adders, multipliers, memory arrays and address decoders) is possible. However, for random logic, one can develop analytic models for estimating the physical capacitance as a function of number of inputs and outputs, circuit complexity (e.g., number of states in a finite-state machine or number of cubes in a minimum sum-of-products expression of a Boolean function, and circuit entropy) and technology/library information.

Interconnect complicates the problem even more as it plays an increasingly important role in determining the capacitive loading of gates while its estimation is a very 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 [110] or by using stochastic/procedural interconnect models [108]. Interconnect capacitance estimation after layout is straight-forward and in general accurate.

With this understanding, we can now consider how to reduce physical capacitance. 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

If there is no switching in a circuit, then no dynamic power will be consumed. There are two components to switching activity, f_{clk} which specifies 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 5X as a function of input correlations [88].

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.

3.4. Towards a Useful Guide for Making Design Trade-offs

The data activity *E(sw)* can be combined with the physical capacitance C_L to obtain *switched capacitance*, $C_{sw} = C_L\ E(sw)$, which describes the average capacitance charged during each data period $1/f_{clk}$. It is the switched capacitance that determines the power consumed in a CMOS circuit under fixed supply voltage level and clock frequency. Minimizing the switched capacitance may however adversely affect the maximum clock frequency in the circuit, which may or may not be acceptable depending on the design constraints. The key question is therefore what objective function should be used for low power design. The answer varies from one application domain to next. If extending the battery life is the only concern, then 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 *action* (that is, the energy-delay product) must be minimized. The energy-delay product allows a designer to find optimizations that provide the largest reduction in energy for the smallest change in performance [54].

One can alternatively minimize energy subject to a given delay constraint. In many design scenarios, circuit delay is determined based on system-level considerations, and hence during optimization, one must minimize energy under user-specified timing constraints. Indeed, much of the published literature focuses on this problem, although authors have referred to it as minimizing power (calculated under a fixed clock frequency) under a given delay constraint (see for example [12] [153] [139]). To set the terminology straight, these works are minimizing energy under a delay constraint.

3.5. Calculation of Switching Activity

Calculation of the switching activity in a logic circuit is difficult because 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.

3.5.1 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 $E_g(sw)=1/2$ (see Table 1 a). 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 $E_g(sw)=4/9$ (see Table 1 b). Finally, assume that it is known that i changes exactly if j changes value, then $E_g(sw)=1/4$ (see Table 1 c). 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* correla-

tions.

(a)

i	j	
$0 \to 0$	$0 \to 0$	$0 \to 0$
$0 \to 1$	$0 \to 1$	$0 \to 1$ *
$1 \to 0$	$1 \to 0$	$1 \to 0$ *
$1 \to 1$	$1 \to 1$	$1 \to 1$

(b)

i	j	
$0 \to 1$	$0 \to 0$	$0 \to 0$
$0 \to 1$	$0 \to 1$	$0 \to 1$ *
$0 \to 1$	$1 \to 0$	$0 \to 0$
$1 \to 0$	$0 \to 0$	$0 \to 0$
$1 \to 0$	$0 \to 1$	$0 \to 0$
$1 \to 0$	$1 \to 0$	$1 \to 0$ *
$1 \to 1$	$0 \to 0$	$0 \to 0$
$1 \to 1$	$0 \to 1$	$0 \to 1$ *
$1 \to 1$	$1 \to 0$	$1 \to 0$ *

(c)

i	j	
$0 \to 0$	$0 \to 0$	$0 \to 0$
$0 \to 0$	$1 \to 1$	$0 \to 0$
$0 \to 1$	$0 \to 1$	$0 \to 1$ *
$0 \to 1$	$1 \to 0$	$0 \to 0$
$1 \to 0$	$1 \to 0$	$1 \to 0$ *
$1 \to 0$	$0 \to 1$	$0 \to 0$
$1 \to 1$	$0 \to 0$	$0 \to 0$
$1 \to 1$	$1 \to 1$	$1 \to 1$

Table 1 Effect of the input 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 statistical sampling techniques (e.g., Monte-Carlo simulation) with well-defined stopping criterion for specified relative or absolute error in power estimates for a given confidence level [19].

3.5.2 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 to as the f*unctional activity* while the latter is referred to as the *spurious (hazard) activity.* It is shown in [9] that the ratio of hazardous component to the total power dissipation varies significantly with the considered circuits (from 9% to 38%) and that the mean value of this ratio is 15-20%. The spurious activity is much higher in certain datapath modules (such as adders and multipliers). Indeed, in a 32-bit pipelined multiplier, the power dissipation due to hazard activity is 3 times higher than that due to functional activity [38].

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 (see Figure 1). A real delay model significantly increases the computational requirements of the power estimation techniques while improving the accuracy of the estimates.

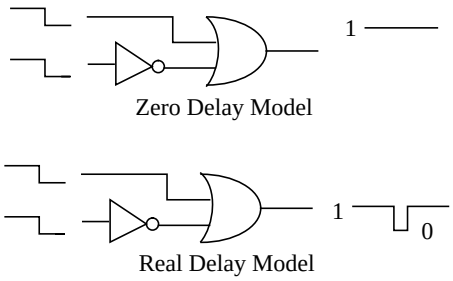


Figure 1 Effect of the delay model.

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. Key problems are to determine when and where in the circuit a hazard is generated and how far and in what form the generated hazard will travel in the circuit before it is possibly suppressed, the latter being a much more difficult problem to solve because hazard propagation using an exact delay model is an analog process that requires detailed circuit-level analysis [43].

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.

3.5.3 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 sig-

nals 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.

3.5.4 Logic Style

Switching activity in CMOS circuits is also a function of the logic style used to implement the circuit. As an example, consider static versus dynamic CMOS logic. The functional activity in dynamic circuits is *always* higher than that in static implementation of the same circuit because 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 activities of these gates in static CMOS which are 3/8, 3/8 and 1/2, respectively. Note however that physical capacitance in dynamic logic tends to be smaller than that in static logic. In addition, dynamic circuits are glitch-free, but consume a lot of clock power. Therefore, the choice between dynamic and static logic implementations is not as clear-cut as one might think. For a detailed comparison of different CMOS logic styles, see [138].

3.5.5 Circuit Structure

The major difficulty in computing the switching activities is the reconvergent fanout 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 because 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.

4. Power Estimation Techniques

The design for low power problem cannot be achieved without accurate power prediction and optimization tools. 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. In this 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 non-simulation based.

4.1. Simulative Techniques

These approaches often have their roots in direct simulation or statistical sampling techniques as detailed below. The main advantage of these techniques is that existing simulators can be used and issues such as hazard generation and propagation and reconvergent fanout-induced correlations in digital circuits are automatically taken into consideration.

4.1.1 Direct simulation

Circuit simulation based techniques [115] [62] [154] 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 [118]. A fast and accurate circuit-level simulator based on the stepwise equivalent conductance and piecewise linear waveform approximation has been described in [17].

PowerMill [56] 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 while maintaining an accuracy of within 10% for a wide range of circuits. PowerMill gives detailed power information (instantaneous, average and RMS current values) as well as the total power consumption (due to capacitance currents, transient short circuit currents, and leakage currents).

Verilog-based gate-level simulation programs (e.g., Verilog-XL Turbo

from Cadence Design) can be adapted to report power dissipation of the circuits under user-specified input sequences. These techniques rely on macromodels built for the gates in the ASIC library as well as detailed gate-level timing analysis to produce power estimates quickly. Their 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 execution time is 3-4 orders of magnitude shorter than SPICE [115]. Similarly, switch-level simulators (such as IRSIM [125]) can be easily modified to report the switched capacitance (and thus dynamic power dissipation) during a simulation run. Switch-level simulation techniques are in general much faster than circuit-level simulation techniques, but are not as accurate or versatile.

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 [25] [72]. Instruction level simulation or behavioral simulators are easily adapted to produce this information.

Estimation of the average *energy consumption per operation* (cycle of activity) in asynchronous (clockless) control circuits that use a two-phase signaling protocol for request/acknowledge handshaking is described in [71]. The proposed method requires pre-calculation of energy consumption per output transition for a small set of predefined macro gates. Estimation of the average *energy consumption per external signal transition* in a *speed-independent* asynchronous control circuit is presented in [7]. The proposed method is simulative in nature, but only requires a small number of input patterns proportional to the size of the high-level specification for the circuit.

4.1.2 Hierarchy of simulation

A simulation method based on a hierarchy of simulators is presented in [160]. 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 trade-off. Another good example is Entice-Aspen [46]. 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 Statistical sampling

A *Monte Carlo simulation (MCS)* approach for power estimation which alleviates the pattern-dependence problem by a proper choice of input vectors has been proposed in [19]. This approach consists of applying randomly generated input patterns at the circuit inputs and monitoring the power dissipation for T clock cycles using a simulator. Each such measurement gives a power sample which is regarded as a random variable. From the *central limit theorem*, as the sample size, T, approaches infinity, the sample density tends to a normal curve.

In practice, a sample size of 30-50 ensures normal sample density for most combinational circuits. For a desired percentage error in the power estimate, ε, a given confidence level, $1-\alpha$, the sample mean, η, and sample standard deviation, σ, the number of required samples, N, can be estimated as follows:

$$N > \left(\frac{t_{\alpha/2}\sigma}{\varepsilon\eta}\right)^2 \tag{2}$$

where $t_{\alpha/2}$ is defined so that the area to its right under the standard normal distribution curve is equal to $\alpha/2$. In estimating the total power consumption of the circuit, the convergence time of the MCS method is short when the error bound is loose or the confidence level is low. Note however that the MCS method may converge to a premature (thus wrong) power estimate if the sample density does not follow a normal distribution (that is, if T was too small). Additionally, this method does not handle spatial correlations at the circuit inputs.

Stopping criteria to obtain a specified switching activity accuracy at all individual nodes in a circuit is proposed in [167]. In this case, the convergence rate, which is determined by the "low-activity" nodes in the circuit, becomes very slow. This problem is addressed by replacing the percentage error bound for these nodes by an absolute error bound, thus allowing possibly large percentage error on these nodes. The overall error however remains small because the contribution of these nodes to the total power dissipation of the circuit is small.

The MCS method has been extended to finite state machines in [104] and

[30] where it is shown that choices of initial states and the length of warm-up periods are critical for generating accurate power estimates. In general, the simulation time for finite state machines is significantly higher than that for combinational circuits of comparable size.

The issue of obtaining run-time and apriori estimates of the number of input patterns for a specified accuracy is discussed in [53]. These estimates are derived through the definition of a set of multinomial random variables and a set of functions based on the parameters of these random variables.

4.2. Non-simulative Approaches

These approaches are based on library models (profile-driven macro-models), stochastic models, and information-theoretic models as detailed below.

4.2.1 Behavioral Level

For functional units (adders, multipliers, and registers) and 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 [112] and [72], 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 [73], 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.

A parametric model is described in [137], where the power dissipation of the various components of a typical processor architecture are expressed as a function of a 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. A power estimation program which combines analytical and stochastic techniques to provide fast and relatively accurate power estimates at the system level is presented in [92].

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 [27][28] 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 [89] and [103] 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. The following summarizes the approach in [89]. Entropy characterizes the uncertainty of a sequence of applied vectors and thus, intuitively, is

related to switching activity. Indeed, it is shown that, under the temporal independence assumption, the average switching activity of a bit is upper-bounded by one half of its entropy. For control circuits and random logic, given the statistics of the input stream and having some information about the structure and functionality of the circuit, the output entropy per bit is calculated as a function of the input entropy per bit and a structure- and function-dependent *information scaling factor*. For dataflow graphs, the output entropy is calculated using a *compositional technique* which has 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 simulative and is thus very fast, yet it produces accurate power estimates.

The above techniques apply to datapaths. Behavioral power prediction models have also been proposed for the controller circuitry in [74][72]. These techniques provide quick estimation of the power dissipation in a control circuit 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 $t-1$), we can write:

$$E_n(sw) = 2\, prob(n)\, (1 - prob(n)). \qquad (3)$$

Computing signal probabilities has attracted much attention. In [107], some of the earliest work in computing the signal probabilities in a combinational network is presented. The authors associate variable names with each of the cir-

cuit inputs representing the signal probabilities of these inputs. Then, for each internal circuit line, they compute algebraic expressions involving these variables. These expressions represent the signal probabilities for these lines. While the algorithm is simple and general, its worse case time complexity is exponential. Approximate signal probability calculation techniques have been proposed in [48] [126] [35] [130] and [70].

In [21], an exact procedure based on Ordered Binary-Decision Diagrams (OBDDs) [16] is described which is linear in the size of the corresponding function graph (the size of the graph, however, may be exponential in the number of circuit inputs). 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:

$$prob(y) = prob(x)\,prob(f_x) + prob(\bar{x})\,prob(f_{\bar{x}}) \tag{4}$$

This leads to a very efficient computational procedure for signal probability estimation. Figure 2 shows an example computation on the OBDD representa-

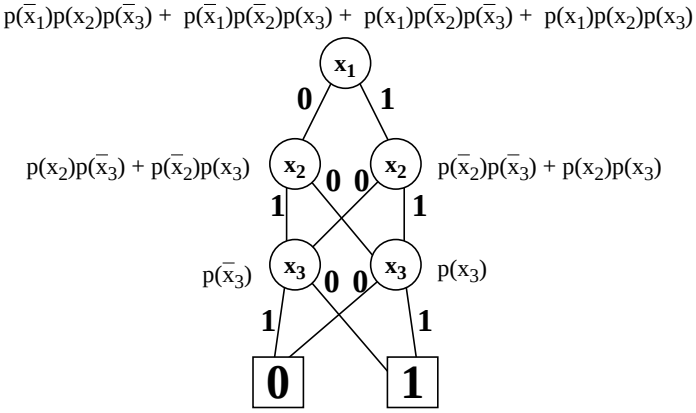


Figure 2 Computing the signal probability using OBDDs.

tion of a three-input **EXOR** gate.

In [41], 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. The correlation coefficient of two signals *i* and *j* is defined as:

$$C(i,j) = \frac{prob(i \wedge j)}{prob(i)\,prob(j)} \tag{5}$$

The correlation coefficients of signal *i* and complement signal $\bar{j}$, complement signal $\bar{i}$ and signal j, etc. are defined similarly. Ignoring higher order correlation coefficients, it is assumed that $C(i,j,k) = C(i,j)\ C(i,k)\ C(j,k)$. The signal probability of *g* is thus approximated by:

NOT gate: $prob(g) = 1 - prob(i)$

AND gate: $prob(g) = \prod_{i \in inputs} prob(i) \cdot \prod_{j>i} C(i,j)$

OR gate: $prob(g) = 1 - \prod_{i \in inputs} (1 - prob(i)) \cdot \prod_{j>i} C(i,j)$

where $C(\bar{i},\bar{j})$ is calculated from *prob(i)*, *prob(j)* and $C(i,j)$.

An incremental technique for switching activity calculation that accounts for spatial correlations is described in [156]. This method takes into account the first-order signal correlations by using the Taylor expansion technique.

In [128] and [87], 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 (*see Figure 3*)*. 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:

$$E_x(sw) = prob(x_{0 \rightarrow 1}) + prob(x_{1 \rightarrow 0})\ . \tag{6}$$

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

In [87], the authors also describe a mechanism for propagating the transition probabilities and correlation coefficients through the circuit which is more efficient because there is no need to build the global function of each node in

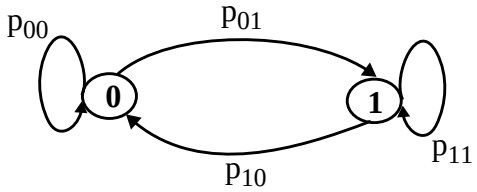


Figure 3 A Markov chain model for representing temporal correlations.

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. The mathematical foundation of this extension is a four state time-homogeneous Markov chain where each state represents some assignment of binary values to two lines *x* and *y* and each edge describes the conditional probability for going from one state to the next. The computational requirement of this extension is however high because it is linear in the product of the number of nodes and number of paths in the OBDD representation of the Boolean function in question. A practical method using local OBDD constructions is described by the authors.

This work has been extended to handle highly correlated input streams using the notions of *conditional independence* and *isotropy of signals* [88]. 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.

The above techniques target average power dissipation. In some applications, peak power dissipation should also be estimated. In [37], a technique for finding the two-vector input sequence that leads to maximum power dissipation in a combinational circuit is described. More recently, a technique is presented that computes the multiple-vector input sequence that leads to maximum average power dissipation in a finite state machine [86].

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 [47], the exact power estimation of a given combinational logic circuit is

carried out by creating a set of symbolic functions that represent Boolean conditions for all values that a node *x* in the circuit can assume at different time instances under a pair of input vectors. The inputs to the created symbolic functions are the circuit input lines at time instances 0^- and ∞. Each symbolic function is the EXOR of the characteristic functions describing the logic values of node *x* at two consecutive time instances (see Figure 4 for an example symbolic network constructed under a unit delay model). The output of the EXOR gate evaluates to one exactly when node *x* makes a transition between the two time instances. Summing the signal probabilities of these symbolic functions gives the average switching activity at *x*. The process, which has to be repeated for all gates in the circuit, is known as the *symbolic simulation*. The major disadvantage of this estimation method is that for medium to large circuits, the symbolic formulae become too large to build. However, for circuits that this method is applicable to and subject to error introduced by the imperfect logic-level glitch propagation scheme, the estimates provided by the method can serve as a basis for comparison among different approximation schemes.

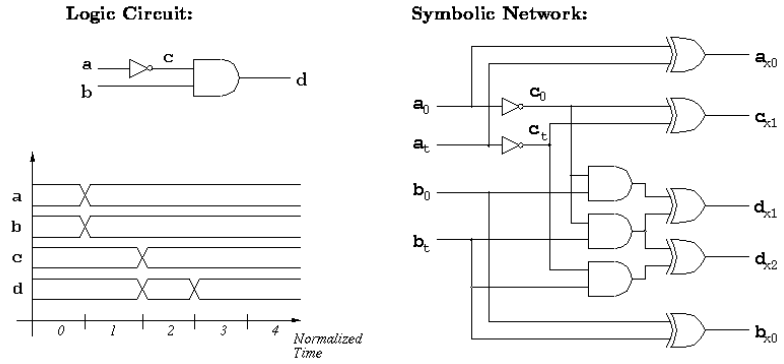


Figure 4 Symbolic simulation under a unit delay model.

The concept of a probability waveform is introduced in [18]. This waveform consists of an *event list*, that is, 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:

$$E_x(sw) = \sum_{t \in eventlist(x)} \left(prob\left(x^t_{0 \to 1} \right) + prob\left(x^t_{1 \to 0} \right) \right). \qquad (7)$$

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 [100], an efficient *probabilistic simulation* technique is described that propagates transition waveforms at the circuit primary inputs throughout 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 [150] 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 ∞).

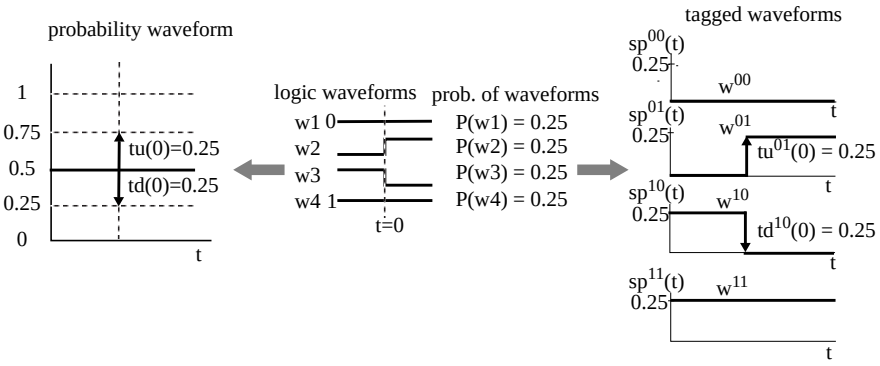


Figure 5 Probability waveforms.

Next, each group is combined into a probability waveform with the appropriate steady-state tag (see Figure 5). Given the tagged probability waveforms at the input of node *n*, it is then possible to compute the tagged probability waveforms at its output. The correlation between probability waveforms at the inputs is approximated by the correlation between the steady state values of these lines, which is in turn calculated efficiently by describing the node function in terms of some set of intermediate variables in the circuit. 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 [38]. The first item refers to the scheme for eliminating from the probability waveforms some of the short glitches that cannot overcome the gate inertial delays. The second item refers to the process of generating accurate and detailed macro-modeling data for the gates in the cell library.

In [101], an efficient algorithm based on the Boolean difference operation is proposed to propagate the transition densities from circuit inputs throughout the circuit. Transition density, *D(y)*, of each node in the circuit is calculated as follows:

$$D(y) = \sum_{i=1}^{n} P\left(\frac{\partial y}{\partial x_i}\right) D(x_i) \tag{8}$$

where y is the output of a node, x_i's are the inputs of the node, and the Boolean difference of function y with respect to x_i gives all combinations for which y depends on x_i. This equation, which can be thought of as a first-order Taylor polynomial approximation of *D(y)*, does not take simultaneous input switching into account. The accuracy of transition density propagation equation can be improved by using higher-order Boolean difference terms as in [29] [93] or by using a conceptual low-pass filter to reduce the hazard count in the above equation as in [102]. A major source of error is the assumption that x_i's are independent. This assumption is however incorrect because x_i's tend to become correlated due to reconvergent fanout structures in the circuit. This problem is solved by describing y in terms of the circuit inputs, which are still assumed to be independent. In this case, the accuracy is improved, but calculation of the Boolean difference terms becomes very expensive. A compromise between accuracy and efficiency can be reached by describing y in terms of some set of intermediate variables in the circuit. One such technique that relies on circuit partitioning and computation caching using OBDDs, is described in [64].

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).

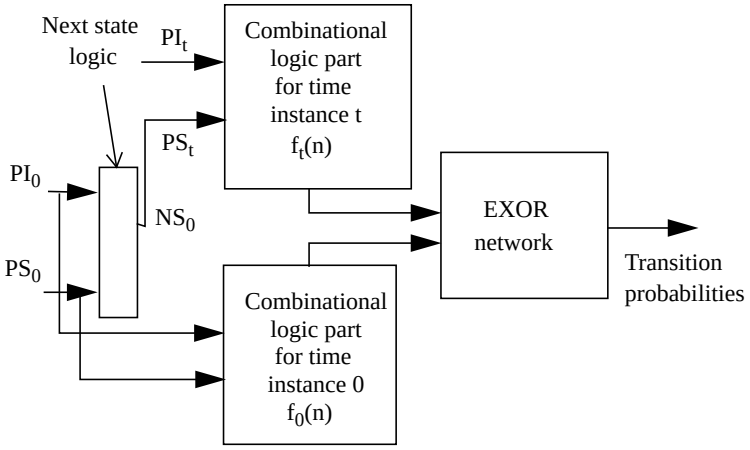


Figure 6 Power estimation for sequential circuits.

A first attempt at estimating switching activity in FSMs has been presented in [47]. 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) as shown in Figure 6. 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 [152] and [97] 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. We describe the method below.

For each state S_i, $1 \le i \le K$ in the STG, we associate a variable $prob(S_i)$ corresponding to the steady-state probability of the machine being in state S_i at $t = \bullet$. For each edge e_{ij} in the STG, we have I_{ij} signifying the input combination corresponding to the edge. Given static probabilities for the primary inputs to the machine, we can compute $prob(S_j \mid S_i)$, the conditional probability of going from

S_i to S_j. For each state S_j, we can write an equation:

$$prob(S_j) = \sum_{S_i \in instates(S_j)} prob(S_i)\, prob\langle S_j | S_i \rangle \tag{9}$$

where $instates(S_i)$ is the set of fanin states of S_i in the STG. Given K states, we obtain K equations out of which any one equation can be derived from the remaining K - 1 equations. We have a final equation:

$$\sum_j prob(S_j) = 1. \tag{10}$$

This linear set of K equations is solved to obtain the different $prob(S_j)$'s.

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. In general, his method cannot handle circuits with large number of flip-flops because it requires explicit consideration of each state in the circuit. On the positive side, state probabilities for some very large FSMs have been calculated using a fully implicit technique described in [50].

The authors of [152] and [97] 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 as follows:

$$\begin{aligned} ps_1 &= f_1(pi, ps_1, ps_2, \ldots, ps_n) \\ &\cdots \\ &\cdots \\ ps_n &= f_n(pi, ps_1, ps_2, \ldots, ps_n) \end{aligned} \tag{11}$$

where ps_i denotes the state bit probabilities of the i^{th} next state bit at the output and the j^{th} present state bit at the input of the FSM, respectively and f_l's are non-linear algebraic functions. The fixed point (or zero) of this system of equations can be found using the Picard-Peano (or Newton-Raphson) iteration [80].

Increasing the number of variables or the number of equations in the above system results in increased accuracy [148]. 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.
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 margins.
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 forces behind this drive are the desire to produce complex, high performance systems on a chip and the projected explosion in demand for low-power portable and wireless systems. It is also expected that various memory and ASICs will also switch to lower supply voltages to maintain manageable power consumption per chip area. 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 that mix and match different supply voltages on board or on the chip.

In [36], 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 reduced 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 7x improvement in speed 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 micron region compared with high performance 0.6 micron 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 non-scalability of the threshold voltage.

The status of silicon-on-insulator (SOI) approach to scaled CMOS is also reviewed in [36], 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. Power Management and CAD Techniques

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

5.2.1 Algorithm and System Design

A number of power minimization strategies have been suggested at this level, including, but not limited to, the following: 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 [24]; some

of the energy that is delivered from the power supply may be cycled back to the power supply [3]; a given task may be partitioned between various hardware modules or programmable processors or both so as to reduce the system-level power consumption; memory optimizing transformations can be used to minimize communications to and from the global memory modules [166]; and software may be compiled so as to minimize the power dissipation when it is executed on a given hardware platform [143].

In many synchronous applications much 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.

In [141], the authors introduce a technique for saving power in the clock tree by stopping the clock fed into idle modules. Sections of the clock tree are turned on or off by gating the clock signals during the active or idle times of the clocked elements as follows. Associated with every node of the clock tree is the activity pattern, which is a binary string of 1's and 0's representing the active/idle status of the node in each time slot. The leaves of the clock tree are sinks and their activities are found from the high level description of the system. The activity patterns of the internal nodes of the clock tree are computed successively by performing bitwise OR operation on the activity patterns of their children. Significant power savings have been reported.

Asynchronous architectures use event-driven handshaking that requests operations to execute only when they are needed, thereby systematically performing what can be considered optimal gated clocking. The disadvantage is that the handshaking control overhead has traditionally limited performance and marginally increased area. For some applications, such as a compact digital cassette error corrector chip set, the performance requirements are easily met and the low-power advantages of completely asynchronous design have yielded an energy savings of up to a factor of five compared with synchronous counterparts [10]. In addition, the ongoing project to implement a fully compatible low-power asyn-

chronous ARM microprocessor has had promising results [45].

Memory power is an important part of the power budget in today's systems. The authors of [2] examine circuit techniques that can be used to reduce the power requirements of a wide memory while having minimum effect on its access time. The technique sets the swings on high capacitance bitlines and I/O lines to 10% of the supply voltage by controlling the time the lines are driven by a replica feedback. A good overview of low power read/write memory design techniques is given in [61].

Memory segmentation/partitioning problem for exploiting the sleep mode operation in low power digital circuits is addressed in [42]. They show that the problem is equivalent to partitioning a set of circuit elements such that each partition as a whole can be put into sleep and the partitioning solution results in the minimum power consumption.

Power saving 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 [3] (see [4] for a recent overview). Additionally, 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 [105], partial transfer of the energy stored on a capacitance to some charge sharing capacitance and then reusing this energy at a later time [51], and electronic compensation for variations in V_T thus making it possible to scale power supply voltages down to very low levels [20], show good signs. Design of energy efficient level-converters and DC/DC converters is also essential to the success of adaptive supply voltage strategies [135].

An integrated approach to the design of a low-power video compression/decompression system which focuses on both the algorithm and architectural design techniques at power levels which are two orders of magnitude below existing solutions is described in [95]. Algorithmic trade-offs include the use of on-chip computation to eliminate off-chip memory accesses, the use of channel-optimized data representations to avoid the error control hardware that would otherwise be necessary, and the encoding of internal data representations to further reduce the energy consumed in data exchanges. Architectural and circuit design techniques include the selection of a filter bank structure that minimizes the energy consumed in the datapath, the adoption of a data shuffle strategy that

results in reduced internal memory size, and the design of digital and analog circuits optimized for low supply voltages.

A number of other power saving techniques have been applied at the algorithm and system level. Interested reader is referred to [91] for a recent review of power optimization techniques at this level.

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 datapath and a logical view of the control unit of a circuit. The datapath 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 datapath 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 [24] 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 more than compensated for by the quadratic power reduction due to reducing the voltage, resulting 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.

A good overview of the use of optimizing transformations for supply voltage reduction is given in [25]. 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. In [90], the authors present a software tool for optimizing the average and peak power dissipation in ASICs using a combination of techniques, including shut-down of modules, lowering supply voltages, using mixed voltages and making architectural trade-offs.

Another power saving strategy is to use multiple power supplies in a circuit. This gives rise to an interesting problem. As the voltage supplied to a functional unit is reduced, the unit slows down, but also consumes lower power. It is therefore desirable to establish a voltage supply value for each functional unit, thereby fixing the latency through the unit, such that the timing constraint for the overall system is met and power dissipation is minimized. In [117], an exact graph-theoretic algorithm for minimizing the system power through variable-voltage scheduling is presented. The area overhead of this approach - in terms of number of required functional units and other support circuitry - however is high.

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

Consider a module *M* in a behaviorally-described circuit that performs two operations A and B. Switching activity at the inputs of *M* is determined by the number of bit flips between the values taken on by variables that are input to the two operations. Module binding determines the mapping from operations to physical modules and, hence, influences the switching characteristics at the inputs of the modules. Similarly, consider a register *R* that is shared between two data values X and Y. Switching activity of *R* depends on the correlation between these two variables. Register binding determines the mapping from date values to registers and, hence, influences the switching characteristics at the inputs of the registers. It thus becomes clear that the circuit activity and power dissipation can vary greatly as a function of the module and register binding performed in the circuit.

In [28], the power optimization problem during module allocation and binding in a functionally pipelined datapath is formulated as a multi-commodity flow problem and solved optimally. The proposed algorithm considers the capacitance switched due to transitions occurring between values of one iteration and the next iteration executed in a loop. Experimental results indicate that significant power savings can be obtained using this method without increasing the area or delay of the datapath and the controller complexity. In [27], the register allocation and binding problem for minimum power consumption is formulated as a minimum cost clique covering of an appropriately defined compatibility graph and

solved optimally in polynomial time using a max-cost flow algorithm. This algorithm accounts for the switched capacitance in a hardware-shared design due to transitions between values of signals in the same iteration of a loop.

In [120] and [121], simulation and profiling are used to construct *switched capacitance matrices* for each type of library module. Entry *(i,j)* of this matrix represents the switched capacitance for the instance *i* of the module when its input *j* changes. The proposed module and register binding algorithms are based on heuristic or integer linear programming techniques for solving the same problems. In [122], an *iterative improvement* algorithm for performing concurrent scheduling, clock selection, resource allocation and binding with the aim of reducing power consumption in synthesized datapath circuits is presented. Results show that a sizeable reduction in power is possible.

Capacitances for the I/O and the global busses are significantly larger than those for the internal circuitry. It is therefore essential to develop techniques for reducing the activity on the I/O pins and the busses. An instruction encoding and scheduling scheme based on Gray coding is presented in [136] which minimizes switching activity in the instruction unit (and the address bus) of a high performance micro-processor. A *Bus-Invert* method for minimizing the activity on I/O pins is proposed in [134]. The idea is to add an extra line to the bus which indicates if the value being transmitted is the true or complement of the intended value. Depending on the value transmitted in the previous cycle, a decision is made to either transmit the true or the complemented value on the bus so as to minimize the bit activity on the bus.

Another low power I/O encoding method based on transition signalling (instead of the usual level signalling) and limited-weight codes, is also described in the same reference. These methods resulted in average of 25% reduction in average power dissipation under a binomial distribution of the distance between consecutive patterns. Methods to implement low-activity arithmetic units based on the one-hot residue coding of the input operands are presented in [31]. CMOS implementation of a direct digital frequency synthesizer for a frequency-hopped spread spectrum communication systems using this technique resulted in almost 2X reduction in the power-delay product compared to a conventional, fully-encoded design.

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 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.

Precomputation Logic

The basic idea is to selectively precompute the output logic values of the circuits one clock cycle before they are required, and then use the precomputed values to reduce internal switching activity in the succeeding clock cycle [1]. A precomputation architecture is shown in Figure 7. The inputs to block A have been partitioned into two sets, corresponding to registers R_1 and R_2. The output of the logic block A feeds register R_3. Two Boolean functions g_1 and g_2 are the *predictor functions*. It is required that:

$$g_1 = 1 \Rightarrow f = 1 \tag{12}$$

$$g_2 = 1 \Rightarrow f = 0 \tag{13}$$

Therefore, during clock cycle t if either g_1 or g_2 evaluates to 1, we set the load enable signal of register R_2 to 0. This implies that the outputs of R_2 during clock cycle $t+1$ do not change. However, since the outputs of R_1, which are determining the function value, are updated, function f will be calculated correctly.

Power reduction is achieved because only a subset of the inputs to block A change, implying reduced switching activity in block A. An example that illustrates the precomputation logic is the *n*-bit comparator that compares two *n*-bit numbers *C* and *D* and computes the function *C* > *D*. Assuming that each *C*<*i*> and *D*<*i*> has a 0.5 signal probability, the probability of correctly predicting the output result using the most significant bit is 0.5 regardless of *n*. Thus, one can achieve a power reduction of 50% (ignoring the overhead of implementing the control logic, g1 and g2). A key design concern in this architecture is to ensure that the control logic does not become too complex.

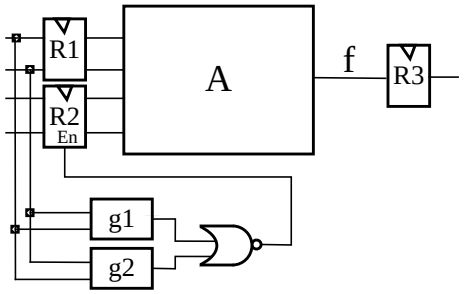


Figure 7 A precomputation architecture for sequential circuits.

In a combinational circuit, it is also possible to identify subsets of gates which do not contribute to the computation initiated with some input stimulus. Power can thus be reduced by turning off these subsets of gates. The overhead of detecting and disabling these sub-circuits may however be large. Different approaches for performing these tasks are described in [98] or [144]. For many circuits, this approach does not produce an appreciable decrease in power dissipation, however, power savings of up to 40% have been reported for some circuits [144].

Retiming

Retiming is the process of re-positioning the flip-flops in a pipelined circuit so as to either minimize the number of flip-flops or minimize the delay through the longest pipeline stage. In [96], it is noted that a flip-flop output makes at most one transition when the clock is asserted (see Figure 8). Based on this observation, the authors then describe a circuit retiming technique targeting low power dissipation. The idea is to identify circuit nodes with high hazard activity and high load capacitance as candidates for adding a flip-flop. The technique does not

produce the optimal retiming solution because the retiming of a single node can dramatically change the switching activity of many other nodes in the circuit.

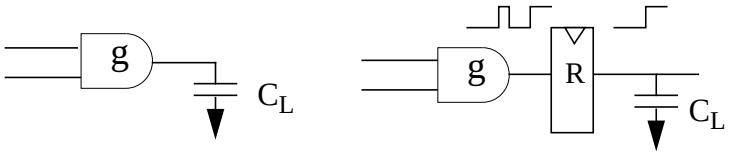


Figure 8 Flip-flop insertion to minimize hazard activity.

The authors report that the power dissipated by the 3-stage pipelined circuits obtained by retiming for low power with a delay constraint is about 8% less than that obtained by retiming for minimum number of flip-flops given a delay constraint.

Synthesis of FSMs with Gated Clock

A technique for automatic synthesis of FSMs with gated clocks that reduces the power dissipation is presented in [8]. The idea is to modify the flip-flop based FSM architecture by adding a new *activation* signal whose purpose is to selectively stop the local clock for the FSM when the machine is idle and does not perform state transitions. The activation function is implemented in the form of a combinational logic block that uses as its inputs the primary inputs and the state lines of the FSM. Applying this technique to some FSM circuits has resulted in large power savings.

State Assignment

State assignment of a finite state machine (which is the process of assigning binary codes to the states) has a significant impact on the area of its final logic implementation. In the past, many researchers have addressed the encoding problem for minimum area of two-level or multi-level logic implementations. These techniques can be modified to minimize the power dissipation. One approach is to minimize the switching activity on the present state lines of the machine by giving minimum-distance (ideally uni-distance) codes to states with high transition frequencies to one another [124]. In [49], a fully implicit encoding algorithm for reducing the average number of bit changes per state transition is presented.

The above formulation however ignores the power consumption in the combinational logic that implements the next state and output logic functions. In an attempt to account for power consumption in the combinational part of the

FSM, the authors of [106] minimize a linear combination of the number of state bits that change every cycle and the number of literals in a multi-level logic implementation of the FSM using a genetic local search algorithm. A more effective approach is presented in [149] where the complexity of the combinational logic resulting from the state assignment is considered by modifying the objective functions used in the conventional encoding schemes such as NOVA [162] and JEDI [84] to achieve lower power dissipation. Experimental results on a large number of benchmark circuits show 10% and 17% power reductions for two-level logic and multi-level implementations compared to NOVA and JEDI, respectively.

Multi-Level Network Optimization

Network don't cares can be used for minimizing the intermediate nodes in a boolean network [127]. Two multi-level network optimization techniques for low power are described in [131] and [57]. The main difference between the procedure in [127] and the low power procedures is in the cost function used during the two-level logic minimization. The new cost function minimizes a linear combination of the number of product terms and the switched capacitance. In addition, the authors of [57] consider how changes in the global function of an internal node affect the switching activity (and thus, the power consumption) of nodes in its transitive fanout. The paper presents a greedy, yet effective, network optimization procedure as summarized below.

The procedure presented in [57] proceeds in a reverse topological fashion from the circuit outputs to the circuit inputs simplifying fanouts of a node before reaching that node. Once a node n is simplified, the procedure propagates those don't care conditions which could only increase (or decrease) the signal probability of that node if its current signal probability is greater than (less than or equal to) 0.5. This will ensure that as nodes in the transitive fanin of n are being simplified, the switching activity of n will not increase beyond its value when node n was optimized. Power consumption in a combinational logic circuit has been reduced by some 10% as a result of this optimization.

The above restriction on the construction of ODC may be overly constraining for the resynthesis process. In [79], a node simplification procedure is presented that identifies *good candidates* for resynthesis, that is, nodes where a local change in their activity plus the change in activity throughout their transitive

fanout nodes, reduces the power consumption in the circuit. Both (delay-independent) functional activity and (delay-dependent) spurious activity are considered.

Node simplification process itself consists of using the appropriate don't care to minimize the area cost of the node. In [165] and [59], this procedure is modified to minimize the power cost of the node. First, consider an example that illustrates the difference between minimizing area and power cost of the node. Assume *a*, *b* and *c* are uncorrelated signals with $p(a) = 0.9$, $p(b) = p(c) = 0.5$ and the following two-level implementations of node *f*:

$$F_1 = a.b + b.c$$

$$F_2 = a.b + \bar{a}.b.c$$

Under the temporal independence assumption, we obtain:

$$P(F_1) = E(a) + 2E(b) + E(c) + E(a.b) + E(b.c) + E(F_1) = 3.04$$

$$P(F_2) = 2E(a) + 2E(b) + E(c) + E(a.b) + E(\bar{a}.b.c) + E(F_2) = 2.89$$

where $P(f)$ denotes the power cost (that is, switched capacitance) of function *f* and all its inputs. This example shows that implementation F_2 provides a better power solution in spite of including a non-prime implicant. Even though the implementation for a non-prime implicant requires more literals and more transistors, overall, less power is consumed.

This observation motivates the definition for *power prime implicants* (PPIs) in [147] and [59]. A PPIs is an implicant whose power cost is strictly less than the power cost of all implicants that contain it. PPIs thus define the set of all implicants that are sufficient and necessary for obtaining a minimum power solution. Given a function *f* and its don't care set, an algorithm for generating the set of all PPIs of *f* is presented in [59]. Using this set, the minimum power solution for a two-level function is then generated by solving a minimum covering problem. The main difficulty in generating a minimum power solution is that compared to a minimum area solution which only requires prime implicants, more implicants need to be considered while solving the covering problem. An upper bound on the expected number of PPIs that will be generated, is derived in [59]. This average-case analysis shows that assuming uniformly distributed values for the input signal probabilities, the number of power prime implicants of a function is linearly proportional to the number of prime implicants of the function where the proportionality constant is $< 4/3$ times the number of inputs to the function.

In [5], the authors describe extensions to the algorithms used in the ESPRESSO two-level logic minimization program by adding heuristics that bias the minimization toward lowering the power dissipation in the circuit. The new two-level minimizer is used in the context of multi-level network optimization. To achieve better results, gate clustering is applied before node simplification. Results indicate about 10% reduction in power.

Common Sub-expression Extraction

The major issue in decomposition is the identification of common sub-expressions. Sharing of such expressions across the design reduces the complexity of the synthesized network and its implementation cost. Extraction based on algebraic division (using cube-free primary divisors or kernels) has proven to be very successful in creating an area-optimized multi-level Boolean network [13] and [119]. The kernel extraction procedure is modified in [124] to generate multi-level circuits with low power consumption. The main idea is to calculate the power savings factor for each candidate kernel based on how its extraction will affect the loading on its input lines and the amount of logic sharing. In [58], an alternative power saving factor has been proposed which assumes that nodes in the multi-level network are in two-level logic form. This is consistent with the assumption made for calculating the literal saving cost of a candidate divisors during algebraic operations. This work also presents power conscious techniques for node elimination, factorization and logic decomposition and presents *scripts* for effective minimization of power dissipation by combining various logic transformations.

An example decomposition is shown in Figure 9 where two network structures that compute the same function $f=ab+ac+bc$ are depicted. Note that the two configurations have the same number of literals in the factored form of their intermediate nodes. It can be also seen that if $E_a(sw) + E_g(sw) > E_c(sw) + E_h(sw)$, then $P^A > P^B$. For example, if $prob(a)=0.5$ and $prob(b)=prob(c)=0.25$, then $P^A - P^B = 13/128$. In general, power savings of about 10% (compared to a minimum-literal network) are expected.

As the active area (e.g., the number of literals) in a circuit strongly influences the power consumption, one must minimize a lexicographic cost (D_a, D_p) where D_a is the literal saving factor and D_p is the power saving factor. At the same time, the above power saving factor is expensive to compute, therefore, it is desirable to calculate it only for a subset of candidate divisors (say, the top 10%

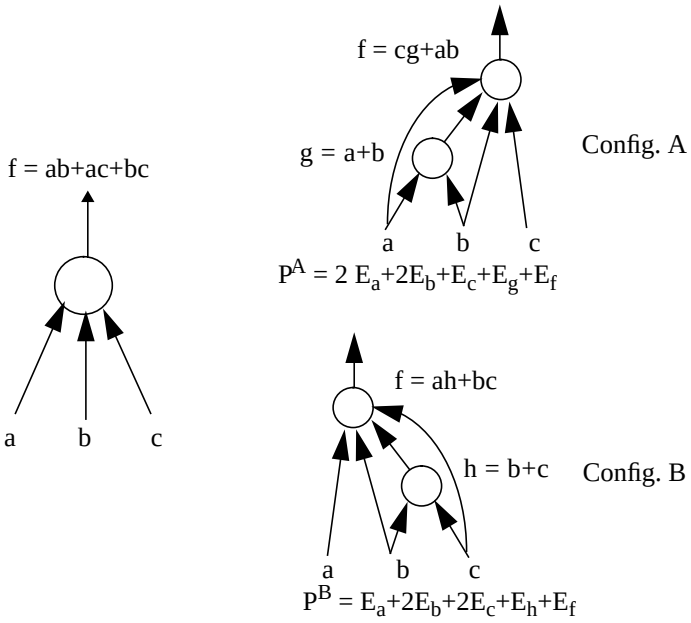


Figure 9 Two decompositions with equal literal counts but different power.

of divisors in terms of their literal saving factors).

Path Balancing

To reduce spurious activity in a circuit, delay of all true paths that converge at each gate, must be roughly balanced. This is because balancing path delays leads to nearly simultaneous switching on input signals to a gate, and thus eliminates possible hazards at the output of the gate (see Figure 9). This in turn reduces the average power dissipation in the circuit. Path balancing can be achieved before technology mapping by selective collapsing and logic decomposition or after technology mapping by delay insertion and pin reordering.

The rationale behind selective collapsing is that by collapsing the fanins of a node into that node, the arrival time at the output of the node can be changed. Logic decomposition and extraction can be performed so as to minimize the level difference between the inputs of nodes which are driving high capacitive nodes. Additionally by inserting variable-delay buffers in a circuit, the delays of all paths in the circuit can be made equal. The key issue in delay insertion is to use the minimum number of delay elements to achieve the maximum reduction in spurious switching activity. Path delays may sometimes be balanced by appropri-

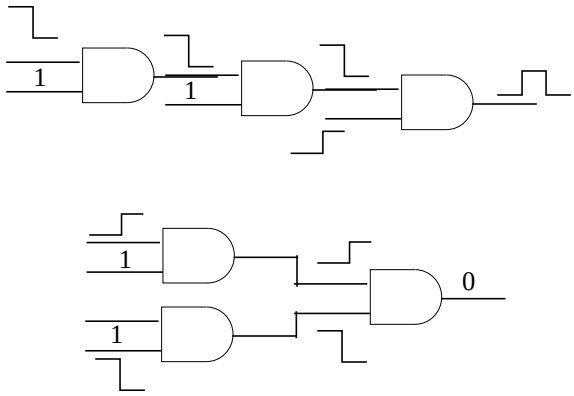


Figure 10 Effect of path balancing on hazard generation.

ate signal to pin assignment. This is possible because the delay characteristics of CMOS gates vary as a function of the input pin which is causing a transition at the output.

Reducing the Circuit Depth

As a result of kernel extraction, it is possible to increase the circuit depth to such an extent that the circuit delay becomes unacceptably large. This problem is often mitigated by a *reduce_depth* operation that implements a depth optimal node clustering algorithm based on [76]. This algorithm however makes no attempt to explore alternative clustering solutions that result in the same logic depth, but have lower power dissipation.

In [158] a formal mechanism is described that implicitly enumerates all non-inferior power-delay clustering solutions and selects the one which has minimum logic depth, but lower power dissipation. This is achieved by enumerating, in postorder, all candidate clusters of up to a maximum cluster size and selecting the power-optimal cluster solution for each delay value at every gate in the circuit. The algorithm which is linear in circuit size but exponential in the maximum cluster size, is provably power- and delay-optimum for trees. The algorithm produces optimum delay solutions for general directed acyclic graphs, but the results are not power-optimum because of the possible logic duplication at the multiple fanout nodes in the circuit. Thus, it is often necessary to perform a delay-constrained power-recovery step as a postprocess.

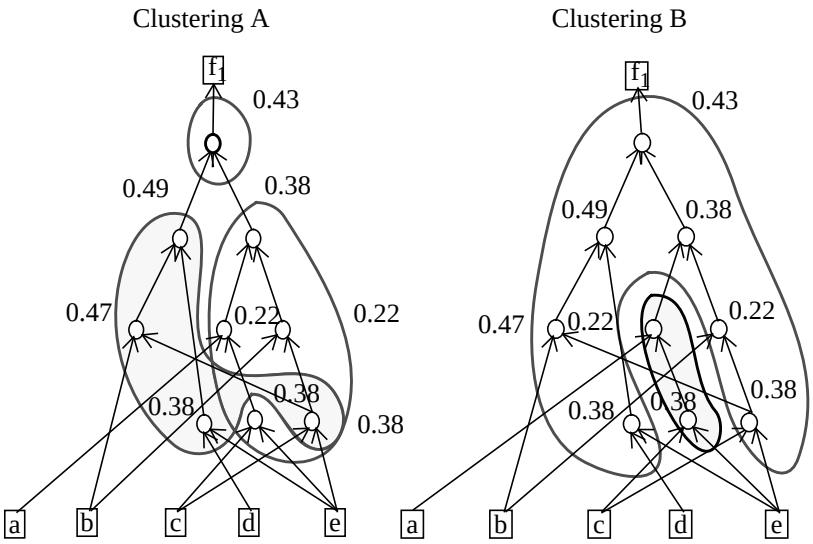


Figure 11 Clustering solutions with equal logic depth but different power.

Two example clustering solutions are shown in Figure 9 where the solution on the left is obtained by Lawler's algorithm while the solution on the right corresponds to power and delay optimal clustering solution (the maximum cluster size is seven). In this example, all input activities are set to 0.5 and the numbers shown beside the nodes represent their switching activities obtained by symbolic simulation of the Boolean network. Both solutions have a depth of two. However, the power cost (switched capacitance) of *inter-cluster* lines in Clustering A is 1.3 while that in Clustering B is 0.65. Experimental results indicate that, on average, 25% improvement in power dissipation of multi-level Boolean circuits is obtained without any increase in circuit delay (assuming that the physical capacitance on inter-cluster lines is much higher than the capacitance on intra-cluster lines).

Technology Decomposition

This is the problem of converting a set of Boolean equations (or a Boolean network) to another set (or another network) consisting of only two-input NAND and inverter gates. It is difficult to come up with a NAND decomposed network which will lead to a minimum power implementation after technology mapping since gate loading and mapping information are unknown at this stage. Nevertheless, it has been observed that a decomposition scheme which minimizes the sum

of the switching activities at the internal nodes of the network, is a good starting point for power-efficient technology mapping.

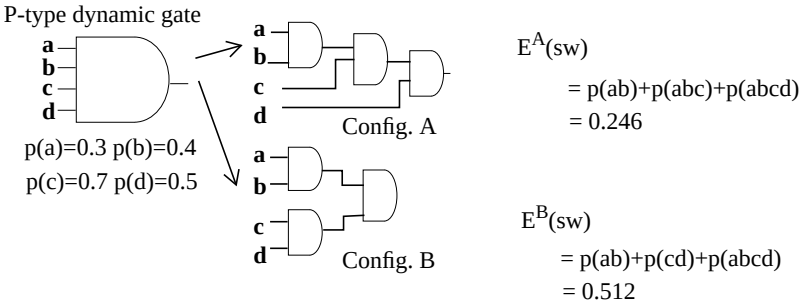


Figure 12 Technology decomposition for minimizing switching activity.

Given the switching activity value at each input of a complex node, a procedure for AND decomposition of the node is described in [151] which minimizes the total switching activity in the resulting two-input AND tree under a zero-delay model. The principle is to inject high switching activity inputs into the decomposition tree as late as possible. The decomposition procedure (which is similar to Huffman's algorithm for constructing a binary tree with minimum average weighted path length) is optimal for dynamic CMOS circuits and produces very good results for static CMOS circuits. An example is shown in Figure 12 where the input signal with the highest switching activity (that is, signal *d*) is injected last in the decomposition tree in configuration A, thus yielding lower power dissipation for this configuration.

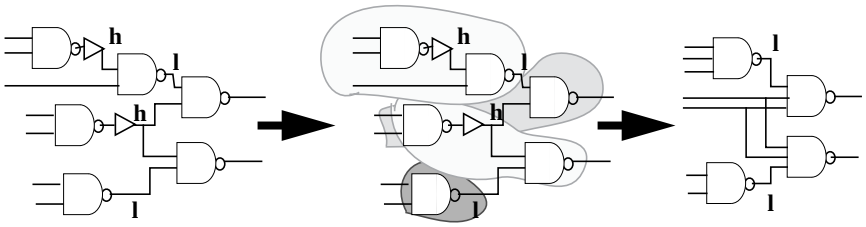
In general, the low power technology decomposition procedure reduces the total switching activity in the networks by 5% over the conventional balanced tree decomposition method.

A different technology decomposition technique is described in [99]. This technique, which again exploits Huffman's algorithm, aims at minimizing the total number of transitions in the binary decomposed tree (including glitches). Under a non-zero delay model and with certain assumptions about spacing of the input arrival times and lack of buffers, the paper presents an optimal algorithm for achieving a minimum transition count decomposition. The paper however ignores the probabilistic nature of logic transitions at the inputs.

Technology Mapping

This is the problem of binding a set of logic equations (or a boolean network) to the gates in some target cell library. A successful and efficient solution to the minimum area mapping problem was suggested in [66] and implemented in programs such as DAGON and MIS. The idea is to reduce technology mapping to DAG covering and to approximate DAG covering by a sequence of tree coverings which can be performed optimally using dynamic programming.

The problem of minimizing the average power consumption during technology mapping is addressed in [151],[142] and [81]. The general principle is to hide nodes with high switching activity inside the gates where they drive smaller load capacitances (see Figure 13).



h: node with high switching activity
l: node with low switching activity

Figure 13 Technology mapping for minimizing switched capacitance.

The approach presented in [151] consists of two steps. In the first step, power-delay curves (that capture power consumption versus arrival time trade-off) at all nodes in the network are computed. In the second step, the mapping solution is generated based on the computed power-delay curves and the required times at the primary outputs. For a NAND-decomposed tree, subject to load calculation errors, this two step approach finds the minimum area mapping satisfying any delay constraint if such a solution exists. Compared to a technology mapper that minimizes the circuit delay, this procedure leads to an average of 18% reduction in power consumption at the expense of 16% increase in area without any degradation in performance.

Generally speaking, the power-delay mapper reduces the number of high switching activity nets at the expense of increasing the number of low switching activity nets. In addition, it reduces the average load on the nets. By taking these two steps, this mapper minimizes the total weighted switching activity and hence

the total power consumption in the circuit.

Under a real delay model, the dynamic programming based tree mapping algorithm does not guarantee to find an optimum solution even for a tree. The dynamic programming approach was adopted based on the assumption that the current best solution is derived from the best solutions stored at the fanin nodes of the matching gate. This is true for power estimation under a zero delay model, but not for that under a real delay model.

The extension to a real delay model is also considered in [151]. Every point on the power-delay curve of a given node uniquely defines a mapped subnetwork from the circuit inputs up to the node. Again, the idea is to annotate each such point with the probability waveform for the node in the corresponding mapped subnetwork. Using this information, the total power cost (due to steady-state transitions and hazards) of a candidate match can be calculated from the annotated power-delay curves at the inputs of the gate and the power-delay characteristics of the gate itself.

Synthesis of Shannon Circuits

A method of synthesizing low-power combinational circuits as *timed Shannon circuits* is proposed in [75]. This method, which exploits the property of Shannon circuits whereby only one path in the circuit is active during an input evaluation, aims to minimize the switched capacitance in the circuit under a bounded fanout model. Experimental results on some benchmark circuits are promising. However, the area overhead of this circuit design style is high while the performance penalty is yet to be determined.

PLA Minimization

High speed PLAs are built by transforming the SOP representation of a two level logic to the NOR-NOR structure with inverting inputs and outputs and implementing it with two NOR arrays. Two common types of implementing the NOR arrays are pseudo-NMOS NOR gates and dynamic CMOS NOR gate.

The primary source of power consumption for a pseudo-NMOS NOR gate is the static power dissipation (see Figure 14). When the NOR gate evaluates to zero, both the PMOS and NMOS parts of the gate are on and there exists a direct current path. The charging and discharging energy is negligible compared with that dissipated by the direct current. Furthermore, the direct current I_{dc} is rela-

tively constant irrespective of the number of NMOS transistors that are on. Therefore the power cost for a product (AND) term is given by:

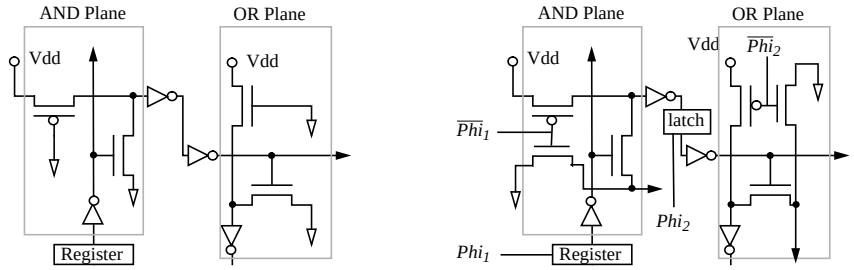
$$V_{dd} \cdot I_{dc} \cdot prob^{0}_{AND} \tag{14}$$

where $prob^{0}_{AND}$ is the probability that the AND term evaluates to 0.

In a dynamic PLA circuit, dynamic power consumption is the major source of power dissipation (see Figure 14). The output of the product term is precharged to 1 and switches when it is evaluated to 0. Therefore the power cost for a product (AND) term is given by:

$$\frac{V_{dd}^2 \cdot f}{2} \cdot \left(\sum_{i=1}^{k} C_i E_i(sw) + C_{AND} prob^{0}_{AND} + 2C_{clock} \right) \tag{15}$$

where C_i is the gate capacitance seen by the i^{th} input of the AND term, C_{AND} is the load capacitance that the AND term is driving, and C_{clock} is the load capacitance of the precharge and evaluate transistors that the clock drives and f is the clock frequency.



Pseudo-NMOS PLA Dynamic PLA

Figure 14 NOR-NOR PLAs.

Fortunately, in both cases it has been shown that the optimum two-level cover will consist of only prime implicants [147] [60]. The resulting minimization problems can be solved exactly by changing the cost function used in the Quine-McClusky procedure or the Espresso heuristic minimizer [14]. In general, optimization for power resulted in a 5% increase in the number of cubes of the function while reducing the power by an average of 11%.

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 (General Cells, Standard Cells, 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, placement, gate and wire sizing, transistor reordering, and routing. If however a real-delay model is used, various layout optimization operations will influence the hazard activity (and thus switching activity) of gates in the circuit, thus greatly complicating various layout optimization steps. It should be also noted that by applying post-layout logic restructuring techniques (such as collapsing logic, local restructuring, re-mapping, etc.), power can be further reduced.

Circuit Partitioning

Netlist partitioning is key in breaking a complex and large design into smaller pieces which are subsequently optimized and implemented as separate blocks. In general, the off-block capacitances are much higher than the on-block capacitances (one to two orders of magnitude). It is therefore essential to develop partitioning schemes that keep the high switching activity nets entirely within the same block as much as possible. Techniques based on local neighborhood search (e.g., the Kernighan-Lin algorithm [65]) and simulated annealing [67] can be easily adapted to do this. In particular, it is adequate to assign net weights based on the switching activity values of the driver gates and then find a minimum cost partitioning solution.

Floorplanning and Placement

Floorplanning is the process of assigning shapes, pin positions and locations to a set of macro-cells or modules so as to minimize the area of the floorplan. One successful floorplanning approach is based on computing the shape functions (height versus width trade-off curves) during a postorder traversal of a

cluster tree that captures the connectivity among modules. The optimal floorplan topology, block shapes and room assignments, and pin positions (or block orientations) are determined during a preorder traversal of this tree [169] [109]. The two dimensional shape function curves can be indexed by the power cost, that is, for each distinct power dissipation value, one shape function is built. These indexed shape functions can then be used during the preorder traversal to compute the optimal power solution which also leads to minimum chip area (see [26] for details).

Placement refers to the process of assigning locations to gates in a circuit netlist. Placement algorithms can be easily modified to minimize the power dissipation. For example, a common placement algorithm for small-cell ICs is to formulate the problem as a constrained mathematical programming problem and then solve it in two phases: global optimization and slot assignment [145] [68]. The objective function is the sum of squares of net lengths while the constraints are center-of-mass and/or path-based timing constraints. The only change needed in the low power formulation is to use the sum of squares of switched capacitances as the objective function during each phase [158]. With this modification, an average power reduction of 8% has been obtained compared to the minimum net length solution without any increase in circuit delay.

Global and Detailed Routing

Global routing produces routing trees for all nets in the circuit so as to minimize the interconnect length and/or chip area. The routing trees for multi-terminal nets are often constructed as Rectilinear Spanning or Steiner trees. In routing a single net to achieve lower power dissipation, the goal is to minimize the physical capacitance which coincides with the minimum length objective used in conventional routing. When routing a collection of nets in fixed-size routing channels (e.g., Gate Array or FPGA layouts), in variable-width routing channels (e.g., Standard Cell layout) or in general area (e.g., General Cell layouts), the difference between minimizing the total physical capacitance and the total switched capacitance comes to surface. In the following, Standard Cell layout will be used as an example.

Both sequential [123] and parallel [32] [77] routing algorithms for routing in Standard Cell layouts have been proposed. Sequential routing algorithms can be modified to produce minimum-power routing solution by simple net weighting where the net weights are derived from the switching activity values of the driver

gates. Nets with higher weights are given priority during routing and thus tend to assume their lowest possible routes. In contrast, low activity nets may encounter blockages, congestion, etc. and thus tend to assume longer lengths than is ideally possible. Alternatively, one can modify the feedthrough insertion and net segment assignment steps in the parallel global routers to generate tree connections with smaller lengths for nets that are driven by gates with higher switching rates [159]. Experimental results have produced only marginal improvements in power dissipation. This is because global routing is a complex process where the net lengths and channel congestion are dictating the routing solution for each net; an extra weighting factor for the nets can only produce a sizeable difference in the final result if net activities (especially on large nets where global routers have many options to route them) are drastically different. This condition was not met in the examples attempted in [159].

Detailed routing produces the wiring geometries and layer assignments within a routing channel, switchbox or general area. To reduce power dissipation during detailed routing, one can give high priority to active nets in using the available routing resources (e.g., tracks, layers). Power dissipation due to cross-talk can be minimized by ensuring that wires carrying high activity signals are placed sufficiently far from the other wires.

Transistor and Gate Sizing

If performance was not a design constraint, design for low (capacitive) power would be achieved by using minimum-sized gate versions everywhere. The gate sizing problem is thus to find a minimum power solution subject to meeting a given delay constraint.

An efficient approach to *continuous* (generator-based) gate sizing for low power is to linearize the path-based timing constraints and use a linear programming solver to find the global optimum solution [11]. This work has been extended to handle setup and hold time constraints in [139]. The drawbacks of this approach are the omission of slope factor (input ramp time) for input waveforms from the delay model and use of a simple power dissipation model that ignores short-circuit current. The LP-based cell selection algorithm can be easily extended to account for the short-circuit power dissipation as described in [111].

A heuristic technique for *discrete* (library-based) gate sizing for minimum power subject to a given delay constraint is described in [140]. The idea is to start with minimum-sized gate versions, and then size up gates along the paths with

negative slacks (that is, critical paths) so as to satisfy the constraints while increasing the switched capacitance of the circuit minimally. Alternatively, one may start with the fastest possible design and then size down the gates along the paths with positive slack (compared to the given delay constraint) so as to maximize the reduction in switched capacitance. Another technique presented in [82], starts with a circuit that satisfies the timing constraint and sizes down certain gates (which are not necessarily on the non-critical paths) to reduce the power dissipation. The shortcoming of these approaches is their greedy nature which leads to sizing one gate a time.

Discrete gate sizing problem is a special case of technology mapping problem and thus the dynamic programming technique can be applied to build the power-delay trade-off curves during a postorder traversal of the circuit and then perform the gate selection during a preorder traversal so as to satisfy the delay constraints while minimizing the switched capacitance.

In [12], the problem of transistor sizing in a static CMOS layout to minimize the capacitive plus short circuit power dissipation. It is shown that the power-optimal size for the transistors in a gate that is driving a given load, can be larger than minimum size. The authors next derive the power-delay optimal sizes for these transistors and present a greedy algorithm for calculating the optimal power sizing subject to a given delay constraint for all gates in a circuit. This algorithm starts by doing an initial power-optimal transistor sizing on each gate. If the power-minimal layout satisfies the delay constraint, the process is terminated; otherwise, the power-delay optimal sizing is applied to gates on the critical paths until the timing target is met.

These researchers have reported about 15-20% reduction in total power dissipation as a result of cell selection or transistor sizing.

Transistor Reordering

In general, library gates have pins that are functionally equivalent which means that inputs can be permuted on those pins without changing function of the gate output. These equivalent pins may have different input pin loads and pin dependent delays. It is well known that the signal to pin assignment in a CMOS logic gate has a sizeable impact on the propagation delay through the gate [63].

If we ignore the *parasitic* (internal) power dissipation due to charging and discharging of source/drain to bulk diffusion capacitances inside a CMOS logic gate, it becomes self-evident that high switching activity inputs should be

matched with pins that have low input capacitance [81]. This scheme is however not very effective as in the semi-custom libraries, the difference in pin capacitances for logically equivalent pins is small. The parasitic power dissipation varies in turn as a function of the switching activities and pin assignment of the input signals (see [153] and [83] for details of the parasitic power calculation model). To find the minimum power pin assignment for a gate that accounts for this internal power dissipation, one must solve a difficult optimization problem as formulated in [153]. As the number of functionally equivalent pins in a typical semi-custom library is not greater that six, it is feasible to exhaustively enumerate all pin permutations to find the minimum power pin assignment.

One can also use heuristics, for example, one such rule assigns input signal with the largest probability of assuming a controlling value (zero for NMOS and one for PMOS) to the transistor near the output terminal of the gate (for series-connected transistors in the pull-up or pull-down blocks of a logic gate) [111]. The rationale is that this transistor will switch off more frequently, thus blocking the internal nodes from non-productive charging and discharging. Another rule is presented in [114] where the input that has the highest switching activity when all other inputs are set to their non-controlling values (one for NMOS and zero for PMOS in series-connected transistors) is directed to the input closest to the output terminal. The rationale is that assigning such a signal closest to the V_{dd} and ground terminals would lead to large power dissipation. The authors of [132] derive similar rules to those mentioned above and point out that if there is a conflict between the two rules, then the transistor ordering should be determined by the ratio of the probability of assuming controlling value over probability of making transitions, that is input with the highest ratio will be placed closest to the output terminal. Experimental results show that about 5% power reduction can be achieved by transistor ordering.

In general, pin permutation for minimum delay produces results that are very different from those obtained for minimum power. Therefore, pin permutation for low power should take place on non-critical gates.

Wire and Driver Sizing

Wire and/or driver sizing are often needed to reduce the interconnect delay on time-critical nets. Wire sizing however tends to increase the load on the driver and hence increase the power dissipation. A simultaneous wire and driver sizing approach can reduce the interconnect delay with only a small increase in the

power dissipation. The approach in [33] uses the properties of monotonicity, separability and dominance (which apply to Elmore delay) to determine the optimal wire sizing solution. The delay is measured using the distributed Elmore delay model and power estimations include both capacitive and short circuit power components. Experimental results show that for the same delay constraint, this approach reduces the power by about 10% when compared to the conventional method of driver sizing only. Another optimal buffer and wire sizing approach based on convex programming techniques which avoids the monotonicity and separability assumptions of the delay model is presented in [94]. This method can be easily extended to determine the optimal gate size and wire widths so as to minimize the power dissipation instead of the area required for the circuit layout.

Super Buffer Design

Super buffer design is a chain of inverters designed to derive a large capacitive load with minimal signal propagation time [63]. A power-optimal buffer sizing technique applicable to the design of super buffers at high speed is presented in [170]. This work is based on an analytic relationship among signal delay, power dissipation, driver size and interconnect load which is in turn derived from the *I-V* characteristics of CMOS transistors. This work shows that optimal-power sizing requires a variable tapering (scaling) factor for the inverter chain.

Clock Tree Generation

Clock is the fastest and most heavily loaded net in a digital system. Ideally, clock signals should have minimum rise/fall times, specified duty cycles and zero skew. Power dissipation of the clock net contributes a large fraction of the total power consumption in a digital circuit [39], thus, it is also desirable to minimize the total capacitive load seen by the clock source.

Many zero-skew clock routing algorithm have been proposed. In one approach, a chain of drivers is introduced at the source and zero-skew is achieved by wire extending or sizing [146] [171]. In another approach, buffers are inserted at internal points in the clock tree [168] for satisfying rise/fall time constraints and for minimizing the area of the clock net. The rationale is that instead of increasing wire widths and lengths to reduce the skew which will result in increased power dissipation, one can use a balanced buffer insertion scheme to partition a large clock tree into a small number of subtrees with minimum wire widths. In [163] a technique for low power clock synthesis that simultaneously inserts buffers and generates the clock tree topology is presented. The main

advantage of this approach is that by judicious buffer insertion, one can reduce the total wire length needed to achieve zero-skew clock tree. Experimental results show improvements in terms of area, rise/fall times and power dissipation compared to the case where buffers are inserted into clock tree as a postprocessing step. The paper also demonstrates that inserting buffers at internal nodes of the clock tree leads to better results compared to inserting buffers at the root of the clock tree only.

Zero-skew is imposed to ensure correct circuit operation. In practice, circuits function correctly within a tolerable clock skew. The objective of low power clock routing is thus to minimize the load on the clock drivers (and hence the clock tree length) subject to meeting a tolerable clock skew. Algorithms for minimum cost bounded skew clock and Steiner tree routing are described in [34] and [55].

Power Distribution

As the supply voltage is reduced, the noise margins are diminished, thus, small voltage drop in the power distribution may have a relatively big impact on the circuit speed. Careful power distribution is thus becoming more important at lower supply voltages. In [164], a technique for concurrent topology design and wire sizing in power distribution networks is presented. The objective is to minimize the layout area while limiting the average current density to avoid electromigration-induced reliability problems and large resistive voltage drops. This technique is based on the observation that when two sinks do not draw currents at the same time, narrow wires can be used for power distribution to those sinks, thus reducing the layout area. The authors report up to 30% area saving compared with the *star connection* scheme.

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.

It is worthwhile to enumerate the major challenges that, to our belief [116], have to be addressed if we want to keep power dissipation within bounds in the future 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.
- Introduction of low-power system synchronization approaches, using either self-timed or locally synchronous 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.
- 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, datapath 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 perform energy recovery are promising in applications where speed can be traded for lower power.

In summary, low power design requires a rethinking of the conventional design process, where power concerns are often overridden by performance and area considerations. This article presented a detailed coverage of low power design methodologies and techniques ranging from technology and devices to circuits and systems. In addition to offering a broad introduction to low power electronics, the article offers an extensive set of references that can be used by researchers.

7. Acknowledgment

This paper could have not been written without the help of students in the CAD group at USC whose research works and papers provided good summaries of some sections of this survey. I am also indebted to professors P. Beerel and C-Y. Tsui and my students C-S. Ding and S. Iman for their careful readings of the manuscript and suggestions to improve parts of it. I also like to thank the anonymous reviewer whose comments made the paper more balanced in the treatment of some ideas. This work was performed in part under ARPA contract No. F33615-95-C1627, NSF NYI award No. MIP-9457392 and SRC contract No. 94-DJ-559.

8. 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] B. S. Amrutur and M. Horowitz . " Techniques to reduce power in fast wide memories . " In *Proceedings of the Symposium on Low Power Electronics*, pages 92-93, October 1994.

[3] 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.

[4] W. Athas. " Energy-recovery CMOS. " In *Low Power Design Methodologies*. J. Rabaey

and M. Pedram (Editors). Kluwer Academic Publishers, pages 63-97, 1996.

[5] I. Bahar and F. Somenzi. " Boolean techniques for low power driven re-synthesis. " In *Proceedings of the IEEE International Conference on Computer Aided Design*, pages 428-432, November 1995.

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

[7] P. A. Beerel, C-T. Hsieh and S. Wadekar. " Estimation of energy consumption in speed-independent control circuits. " In *Proceedings of the 1995 International Symposium on Low Power Design*, pages 39-44, April 1995.

[8] L. Benini and G. De Micheli. " Transformation and synthesis of FSMs for low power gated clock implementation. " In *Proceedings of the 1995 International Symposium on Low Power Design*, pages 21-26, April 1995.

[9] 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.

[10] C.H. (Kees) van Berkel, R. Burgess, J. Kessels, A. Peeters, M. Roncken and F. Saeijs. " A fully-asynchronous low-power error corrector for the digital compact cassette player. " In *Proceedings of the International Solid-State Circuits Conference*, 1994.

[11] 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.

[12] M. Borah, R. M. Owens and M. J. Irwin. " Transistor sizing for minimizing power consumption of CMOS circuits under delay constraint. " In *Proceedings of the 1995 International Symposium on Low Power Design*, pages 167-172, April 1995.

[13] R. K. Brayton, G. D. Hachtel, and A. L. Sangiovanni-Vincentelli. " Multilevel logic synthesis. " *Proceedings of the* IEEE, volume 78, pages 264–300, February 1990.

[14] R. K. Brayton, G. D. Hachtel, C. McMullen and A. L. Sangiovanni-Vincentelli. " Logic minimization algorithms for VLSI synthesis. " Kluwer Academic Publishers, Boston, Massachusetts, 1984.

[15] R. W. Brodersen et al, "Technologies for personal communications. "*Proceedings of the VLSI Symposium*, Japan, pages 5-9, 1991.

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

[17] P. Buch, S. Lin, V. Nagasamy and E. S. Kuh. " Techniques for fast circuit simulation applied to power estimation of CMOS circuits. " In *Proceedings of the 1995 International Symposium on Low Power Design*, pages 135-138, April 1995.

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] A. Chandrakasan, R. Allmon, A. Stratakos, and R. W. Brodersen. " Design of portable systems. " *Proceedings of CICC*, San Diego, May 1994.

[23] A. Chandrakasan, S. Sheng, and R. W. Brodersen. " Low-power techniques for portable real-time DSP applications. " *Proceedings of VLSI Design*, India, 1992.

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

[25] 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*, pages 300-303, November 1992.

[26] K. Y. Chao and D. F. Wong. " Low-power consideration in floorplan design. " In *Proceedings of the 1994 International Workshop on Low Power Design*, pages 45-50, April 1994.

[27] 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.

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

[29] T-L. Chou , K. Roy and S. Prasad. " Estimation of circuit activity considering signal correlation and simultaneous switching. " In *Proceedings of the* IEEE *International Conference on Computer Aided Design*, pages 300-303, November 1994.

[30] T-L. Chou and K. Roy. " Statistical estimation of sequential circuit activity. " In *Proceedings of the* IEEE *International Conference on Computer Aided Design*, pages 34-37, November 1995.

[31] W. A. Chren. " Low delay-power product CMOS design using one-hot residue coding. " In *Proceedings of the 1995 International Symposium on Low Power Design*, pages 145-150, April 1995.

[32] J. Cong and B. T. Preas. " A new algorithm for standard cell global routing. " In *Proceedings of the* IEEE *International Conference on Computer Aided Design*, pages 176-180, November 1988.

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

[34] 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.

[35] M. A. Critic. " Estimating dynamic power consumption of CMOS circuits. " In *Proceedings of the* IEEE *International Conference on Computer Aided Design*, pages 534-537, November 1987.

[36] 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.

[37] S. Devadas, K. Keutzer and J. White. " Estimation of power dissipation in CMOS combinational circuits using Boolean function manipulation. " IEEE *Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 11(3):373-383, March 1992.

[38] 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*, pages 42-43, September 1995.

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

[40] J. Eager, " Advances in rechargeable batteries spark product innovation. " *Proceedings of the 1992 Silicon Valley Computer Conference*, Santa Clara, pages 243-253, Aug. 1992.

[41] 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.

[42] A. H. Farrahi, G. E. Tellez and M. Sarrafzadeh. " Memory segmentation to exploit sleep mode operation. " In *Proceedings of the 32nd Design Automation Conference*, pages 36-41, June 1995.

[43] M. Favalli and L. Benini. " Analysis of glitch power dissipation in CMOS ICs. " In *Proceedings of the 1995 International Symposium on Low Power Design*, pages 123-128, April 1995.

[44] 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.

[45] S. Furber. " Computing without clocks: micropipelining the ARM processor. " In *Asynchronous Digital Circuit Design.* G. Birtwistle and A.Davis (Editors), Springer, pages 211-262, 1995.

[46] 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.

[47] 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.

[48] H. Goldstein. " Controllability/observability of digital circuits. " IEEE *Transactions on Circuits and Systems*, 26(9):685–693, September 1979.

[49] G. D. Hachtel, M. Hermida, A. Pardo, M. Poncino and F. Somenzi." Re-encoding sequential circuits to reduce power dissipation. " In *Proceedings of the* IEEE *International Conference on Computer Aided Design*, pages 70-73, November 1994.

[50] G. D. Hachtel, E. Macii, A. Pardo and F. Somenzi. " Probabilistic analysis of large finite state machines. " In *Proceedings of the 31st Design Automation Conference*, pages 270-275, June 1994.

[51] 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*, pages 60-61, October 1995.

[52] N. Hedenstierna and K. Jeppson. " CMOS circuit speed and buffer optimization. " IEEE *Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 6(3):270-281, March 1987.

[53] A. M. Hill and S-M. Kang. " Determining accuracy bounds for simulation-based switching activity estimation. " In *Proceedings of the 1995 International Symposium on Low Power Design*, pages 215-220, April 1995.

[54] M. Horowitz, T. Indermaur and R. Gonzalez. " Low-power digital design. " In *Proceedings of the 1995 IEEE Symposium on Low Power Electronics*, pages 8-11, October 1995.

[55] 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.

[56] C. X. Huang, B. Zhang, A-C. Deng and B. Swirski. " The design and implementation of PowerMill. " In *Proceedings of the 1995 International Symposium on Low Power Design*, pages 105-110, April 1995.

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

[58] S. Iman and M. Pedram. " Logic extraction and decomposition for low power. " In *Proceedings of the 32nd Design Automation Conference*, pages 433-438, June 1995.

[59] 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.

[60] 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.

[61] K. Itoh, K. Sasaki and Y. Nakagome. " Trends in low-power RAM circuit technologies. " *Proceedings of* IEEE, 83(4):524-543, April 1995.

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

[63] S. M. Kang and Y. Leblebici. *CMOS Digital Integrated Circuits: Analysis and Design.* McGraw-Hill Companies, Inc. 1996.

[64] B. Kapoor. " Improving the accuracy of circuit activity measurement. " In *Proceedings of the 1994 International Workshop on Low Power Design*, pages 111-116, April 1994.

[65] B. W. Kernighan and S. Lin. " An efficient heuristic procedure for partitioning graphs. " *Bell System Technical Journal*, 49(2):291-307, February 1970.

[66] K. Keutzer. " DAGON: Technology mapping and local optimization. " In *Proceedings of the 24th Design Automation Conference*, pages 341–347, June 1987.

[67] S. Kirkpatrick and C. D. Gelatt and M. P.Vecchi. " Optimization by simulated annealing. " *Science*, 220(4598):671-680, May 1983.

[68] J. M. Kleinhans, G. Sigl, F. M. Johannes and K. J. Antreich. " GORDIAN: VLSI placement by quadratic programming and slicing optimization. " IEEE *Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 10(3):356-365, March 1991.

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

[70] B. Krishnamurthy and I. G. Tollis. " Improved techniques for estimating signal probabilities. " IEEE *Transactions on Computers*, 38(7):1245–1251, July 1989.

[71] P. Kudva and V. Akella. " A technique for estimating power in asynchronous circuits. " In *Proceedings of the International Symposium on Advanced Research in Asynchronous circuits and systems*, pages 166-175, 1994.

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

[73] 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.

[74] 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*, pages 93-98, April 1995.

[75] L. Lavagno, P. M. McGeer, A. Saldanha, A. L. Sangiovanni-Vincentelli. " Timed Shannon Circuits: a power-efficient design style and synthesis tool. " In *Proceedings of the 32nd Design Automation Conference*, pages 4254-260, June 1995.

[76] E. L. Lawler and K. N. Levitt and J. Turner. " Module clustering to minimize delay in digital networks. " IEEE *Transactions on Computers*, pages 45-57, January 1969.

[77] K. W. Lee and C. Sechen. " A new global router for row-based layout. " In *Proceedings of the* I EEE *International Conference on Computer Aided Design*, pages 180-183, November 1988.

[78] C. E. Leiserson, F. M. Rose and J. B. Saxe. " Optimizing synchronous circuitry by retiming. " In *Proceedings of the Third Caltech Conference on VLSI*, pages 23-36, March 1983.

[79] 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.

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

[81] 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.

[82] H-R. Lin and T-T. Hwang. " Power reduction by gate sizing with path-oriented slack calculation. " In *Proceedings of the 1st Asia-Pacific Design Automation Conference*, pages 7-12, August 1995.

[83] J. Y. Lin, T. C. Liu and W. Z. Shen. " A cell-based power estimation in CMOS combina-

tional circuits. " In *Proceedings of the* I EEE *International Conference on Computer Aided Design*, pages 304-309, November 1994.

[84] B. Lin and A. R. Newton. " Synthesis of multiple-level logic from symbolic high-level description languages. " In *Proceedings of IFIP International Conference on Very Large Scale Integration*, pages 187-196, August 1989.

[85] D. Maliniak, " Better batteries for low-power jobs. " *Electronic Design*, vol. 40, no. 15, pages 18, July 23, 1992.

[86] S. Manne, A. Pardo, R. Bahar, G. Hachtel, F. Somenzi, E. Macii and M. Poncino. " Computing the maximum power cycles of a sequential circuit. " In *Proceedings of the 32nd Design Automation Conference*, pages 23-28, June 1995.

[87] 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.

[88] 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.

[89] 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.

[90] R. S. Martin and J. P. Knight. " Power-profiler: optimizing ASICs power consumption at the behavioral level. " In *Proceedings of the 32nd Design Automation Conference*, pages 42-47, June 1995.

[91] R. Mehra, D. B. Lidsky, A. Abnous, P. E. Landman and J. M. Rabaey. " Algorithm and architectural level methodologies for low power. " In *Low Power Design Methodologies*. J. Rabaey and M. Pedram (Editors). Kluwer Academic Publishers, pages 333-362, 1996.

[92] R. Mehra and J. Rabaey. " Behavioral level power estimation and exploration. " In *Proceedings of the 1994 International Workshop on Low Power Design*, pages 197-202, April 1994.

[93] H. Mehta, M. Borah, R. M. Owens and M. J. Irwin. " Accurate estimation of combinational circuit activity. " In *Proceedings of the 32nd Design Automation Conference*, pages 618-622, June 1995.

[94] N. Menezes, R. Baldick and L. T. Pileggi. " A sequential quadratic programming approach to concurrent gate and wire sizing. " In *Proceedings of the* IEEE *International Conference on Computer Aided Design*, pages 144-151, November 1995.

[95] T. H. Meng, B. M. Gordon, E. K. Tsern and C. Hung. " Portable video-on-demand in wireless communication. " *Proceedings of* IEEE, 83(4):659-680, April 1995.

[96] 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.

[97] 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.

[98] J. Monteiro, J. Rinderknecht, S. Devadas and A. Ghosh. " Optimization of combinational and sequential logic circuits for low power using precomputation. " In *Proceedings of the 1995 Chapel Hill Conference on Advanced Research on VLSI*, pages 430-444, March 1995.

[99] R. Murgai, R. K. Brayton and A. Sangiovanni-Vincentelli. " Decomposition of logic functions for minimum transition activity. " In *Proceedings of the 1994 International Workshop on Low Power Design*, pages 33-38, April 1994.

[100] 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.

[101] F. N. Najm. " Transition density: A new measure of activity in digital circuits. " IEEE *Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 12(2):310-323, February 1993.

[102] F. N. Najm. " Low pass filter for computing transition density in digital circuits. " IEEE *Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 13(9):1123-1131, September 1994.

[103] 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.

[104] F. N. Najm, S. Goel and I. Hajj. " Power estimation in sequential circuits. " In *Proceedings of the 32nd Design Automation Conference*, pages 635-640, June 1995.

[105] L.S. Nielsen, C. Niessen, J. Sparso and C.H. van Berkel. " 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.

[106] E. Olson and S. Kang. " Low-power state assignment for finite state machines. " In *Proceedings of the 1994 International Workshop on Low Power Design*, pages 63-68, April 1994.

[107] K. P. Parker and J. McCluskey. " Probabilistic treatment of general combinational networks. " IEEE *Transactions on Computers*, 24(6):668–670, June 1975.

[108] 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.

[109] M. Pedram, M. Marek-Sadowska and E. S. Kuh. " Floorplanning with pin assignment. " In *Proceedings of the* IEEE *International Conference on Computer Aided Design*, pages 98-101, November 1990.

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

[111] M. Pedram. " Power estimation and optimization at the logic level. " Int'l Journal of High Speed Electronics and Systems, 5(2):179-202, June 1994.

[112] 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.

[113] R. A. Powers. " Batteries for low power electronics. " *Proceedings of* IEEE,

38(4):687-693, April 1995.

[114] S. C. Prasad and K. Roy. " Circuit optimization for minimization of power consumption under delay constraint. " In *Proceedings of the 1994 International Workshop on Low Power Design*, pages 15-20, April 1994.

[115] T. Quarles. " The SPICE3 Implementation Guide. " Technical Report No. M89-44, Electronics Research Laboratory, University of California, Berkeley, California, April 1989.

[116] J. Rabaey and M. Pedram (Editors). *Low Power Design Methodologies*. Kluwer Academic Publishers, 1996.

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

[118] 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.

[119] 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.

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

[121] A. Raghunathan and N. K. Jha. " An ILP formulation for low power based on minimizing switched capacitance during datapath allocation. " In *Proceedings of the Int'l Symposium on Circuits and Systems*, pages --, May 1995.

[122] A. Raghunathan and N. K. Jha. " An iterative improvement algorithm for low power datapath synthesis. " In *Proceedings of the* IEEE *International Conference on Computer Design*, pages 597-602, November 1995.

[123] K. A. Roberts. " Automatic layout in the Highland system. " In *Proceedings of the* IEEE *International Conference on Computer Aided Design*, pages 224-226, November 1984.

[124] 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.

[125] 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.

[126] J. Savir, G. Ditlow, and P. Bardell. " Random pattern testability. " IEEE *Transactions on Computers*, 33(1):1041–1045, January 1984.

[127] 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.

[128] 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.

[129] S. C. Seth and V. D. Agrawal. *A new model for calculation of probabilistic testability in combinational circuits*. Elsevier Science Publishers, 1989.

[130] S.C. Seth, L. Pan, and V.D. Agrawal. " PREDICT - Probabilistic estimation of digital circuit testability. " In *Proceedings of the Fault Tolerant Computing Symposium*, pages 220–225, June 1985.

[131] 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*, pages 402-407, November 1992.

[132] W-Z. Shen, J-Y. Lin and F-W. Wang. " Transistor reordering rules for power reduction in CMOS gates. " In *Proceedings of the 1st Asia-Pacific Design Automation Conference*, pages 1-5, August 1995.

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

[134] M. Stan and W. Burleson. " Limited-weight codes for low power I/O. " In *Proceedings of the 1994 International Workshop on Low Power Design*, pages 209-214, April 1994.

[135] 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*, pages 105-110, April 1994.

[136] C-L. Su, C-Y. Tsui and A. M. Despain. " Low power architecture design and compilation techniques for high-performance processors. " In *CompCon'94 Digest of Technical Papers*, pages 489-498, February 1994.

[137] 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.

[138] C. Svensson and D. Liu. " Low power circuit techniques. " In *Low Power Design Methodologies*. J. Rabaey and M. Pedram (Editors). Kluwer Academic Publishers, pages 38-64, 1996.

[139] Y. Tamiya, Y. Matsunaga and M. Fujita." LP based cell selection with constraints of timing, area and power consumption. " In *Proceedings of the* IEEE *International Conference on Computer Aided Design*, pages 4378-381, November 1994.

[140] C-H. Tan and J. Allen. " Minimization of power in VLSI circuits using transistor sizing, input ordering and statistical power estimation. " In *Proceedings of the 1994 International Workshop on Low Power Design*, pages 75-80, April 1994.

[141] G. E. Tellez, A. Farrahi and M. Sarrafzadeh. " Activity-driven clock design for low power circuits. " In *Proceedings of the* IEEE *International Conference on Computer Aided Design*, pages 62-65, November 1995.

[142] 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.

[143] V. Tiwari, S. Malik and W. Wolfe. " Power analysis of embedded software: a first step towards software minimization. " IEEE *Transactions on VLSI Systems*, 2(4):437-445, December 1994.

[144] 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.

[145] R. S. Tsay, E. S. Kuh and C. P. Hsu. " PROUD: A sea-of-gates placement algorithm. " In *Proceedings of the* IEEE *International Conference on Computer Aided Design*, pages 318-323, November 1988.

[146] R. S. Tsay. " An exact zero-skew clock routing algorithm. " IEEE *Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 12(3):242-249, March 1993.

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

[148] 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, 3(3):404-416, September 1995.*

[149] 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.

[150] 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.

[151] 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.

[152] 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.

[153] 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.

[154] 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.

[155] S. Turgis, N. Azemard and D. Auvergne. " Explicit evaluation of short circuit power dissipation for CMOS logic structures. " In *Proceedings of the 1995 International Symposium on Low Power Design*, pages 129-134, April 1995.

[156] T. Uchino, F. Minami, T. Mitsuhashi and N. Goto. " Switching activity analysis using Boolean approximation method. " In *Proceedings of the* IEEE *International Conference on Computer Aided Design*, pages 20-25, November 1987.

[157] 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.

[158] 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.

[159] H. Vaishnav. *Optimization of Post-Layout Area, Delay and Power Dissipation.* Ph.D. Dissertation, Computer Engineering, University of Southern California, August 1995.

[160] 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.

[161] 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*, 19:468–473, August 1984.

[162] T. Villa and A. Sangiovanni-Vincentelli. " NOVA: State assignment of finite state machines for optimal two-level logic implementations. " IEEE T*ransactions on Computer-Aided Design of Integrated Circuits and Systems*, 9: 905-924, September 1990.

[163] A. Vittal and M. Marek-Sadowska. " Power optimal buffered clock tree design. " In *Proceedings of the 32nd Design Automation Conference*, pages 497-502, June 1995.

[164] A. Vittal and M. Marek-Sadowska. " Power distribution topology design. " In *Proceedings of the 32nd Design Automation Conference*, pages 503-507, June 1995.

[165] S. B. K. Vrudhula and H-Y. Xie. " Techniques for CMOS power estimation and logic synthesis for low power. " In *Proceedings of the 1994 International Workshop on Low Power Design*, pages 21-26, April 1994.

[166] S. Wuytack, F. Catthoor, F. Franssen, L. Nachtergaele and H. De Man. " Global communication and memory optimizing transformations for low power systems. " In *Proceedings of the 1994 International Workshop on Low Power Design*, pages 203-208, April 1994.

[167] 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.

[168] J. G. Xi and W-M. Dai. " Buffer insertion and sizing under process variations for low power. " In *Proceedings of the 32nd Design Automation Conference*, pages 491-496, June 1995.

[169] G. Zimmermann. " A new area and shape function estimation technique for VLSI layout. " In *Proceedings of the 25th Design Automation Conference*, pages 60-65, June 1988.

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

[171] Q. Zhu, W. M. Dai and J. G. Xi. " Optimal sizing of high speed clock network based on distributed and transmission line models. " In *Proceedings of the* IEEE *International Conference on Computer Aided Design*, pages 628-633, November 1993.