Low Power CAD: Trends and Challenges

Massoud Pedram
Department of EE - Systems
University of Southern California
Los Angeles, CA 90089
Email: pedram@pollux.usc.edu

Nikkei Microdevices
Nikkei Business Publications Inc.
October 1994

Abstract

Essential elements of a low power design environment include means of analyzing the dissipation of a proposed or an existing design, mechanisms for minimizing the power consumption when needed and techniques to explore the impact of design trade-offs on the power consumption, area and performance of a design. This paper describes the state of the art of CAD tools and methodologies as well as references to find additional more in-depth technical information in specific fields and highlights the research areas.

1 Introduction

Low power, yet high-throughput and computationally intensive, circuits are becoming a critical application domain. One driving factor behind this trend is the growing class of personal computing devices (digital pens, portable desktops, audio- and video-based multimedia products) as well as wireless communications and imaging systems (personal digital assistants, personal communicators, smart cards) which demand high-speed computations, complex functionalities and often real-time processing capabilities with low power consumption. Another crucial driving factor is that excessive power consumption is becoming the limiting factor in integrating more transistors on a single chip or on a multiple-chip module. Unless power consumption is dramatically reduced, the resulting heat will limit the feasible packing and performance of VLSI circuits and systems. Indeed, circuits synthesized for low power are also less susceptible to run time failures.

In rising to the challenge to reduce power the semiconductor industry has adopted a multifaceted approach, attacking the problem on three fronts:

- Reducing chip and packaging capacitance through process scaling and advanced interconnect substrates such as MCM. This approach can be effective but is very expensive.

- Supply voltage scaling. There are some immediate advantages to this approach but they are not generally scalable without incurring the cost of new IC fab processing. Furthermore, supply scaling may be contrary to the industry "pull" for voltage standards and may run into fundamental problems when issues such as signal-to-noise are considered.

- Employing better algorithmic, architectural and 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 two approaches and because it is relatively untapped in potential.

Low power VLSI design can be achieved at various levels. For example, at the algorithmic level, correct data representation and choice of algorithms may significantly reduce power consumption. At the system level, inactive hardware modules may be automatically turned off to save power; modules may be provided with the optimum supply voltage and interfaced by means of level converters; some of the energy that is delivered from the the power supply may be cycled back to the power supply, and module a task may partitioned between various hardware modules and/or programmable processors so as to reduce the system-level power consumption. These observations have been experimentally verified and demonstrated by a number of researchers in various domains such as audio/speech, image/video, telecommunications and networking. However, no automatic procedures for doing this exist.

At the architectural design level, concurrency increasing transformations such as loop unrolling, pipelining and control flow optimization as well as critical path reducing transformations such as height minimization, retiming and pipelining may be used to allow a reduction in supply voltage without degrading system throughput; Algorithm-specific instruction sets may be utilized that boost code density and minimize switching; A Gray code addressing scheme can be used to reduce the number of bit changes on the address bus; Internal busses may be replaced by point-to-point connections to avoid driving a large number of modules on every bus access; Bus architectures with sub 1-volt swing and low standby current can be used; On-chip cache may be added to minimize external memory references; Locality of reference may be exploited to avoid accessing global resources such as memories, busses or ALU's; Control signals that are "don't cares" can be held constant to avoid initiating nonproductive switching.

At the logic level, symbolic states of a FSM can be assigned binary codes to minimize the number of bit changes in the combinational logic for the most likely state transitions; Common subexpressions with low switching probability values can be extracted; Network don't cares can be used to modify the local expression of a node so as to reduce the switching activity in the transitive fanout of the node; High switching activity nodes may be hidden

Level	Example Techniques
Algorithmic	Correct data representation and choice of algorithms
System	Power mode management Application of various energy recovery techniques Use of optimum supply voltage and level converters for modules Task partitioning between hardware modules and programmable processors
Architectural	Concurrency increasing transformations Pipelining, scheduling, module and register assignment Design of application-specific instruction sets Use of on-chip cache, Sub 1-volt swing bus architectures
Logic	Power-sensible retiming and state assignment Two- and multi-level logic optimization targeting low power dissipation Path balancing, technology mapping and pin assignment for low power
Physical	Low power physical partitioning, floorplanning, placement and routing Transistor and/or wire sizing, library design

Table 1: Examples of power reduction mechanisms at various levels of abstraction

inside gates during technology mapping; Gate resizing, signal-to-pin assignment and I/O encoding can further reduce the power consumption. At the physical design level, power may be reduced by appropriate netlist partitioning, placement, global routing, wire sizing and clock tree generation (see Table 1).

The design for low power problem cannot be achieved without good power prediction and optimization tools (see Figure 1). The remainder of this paper describes the CAD tools and methodologies required to effect efficient design for low power during the logic synthesis and layout optimization. As low power design is a relatively new field, the paper is targeted at a wide audience to achieve the following:

- Convey an understanding of the breadth and depth of the problem.
- Explain the state of the art in CAD tools and methodologies.
- Describe the needs and challenges for new CAD tools to support low power design.

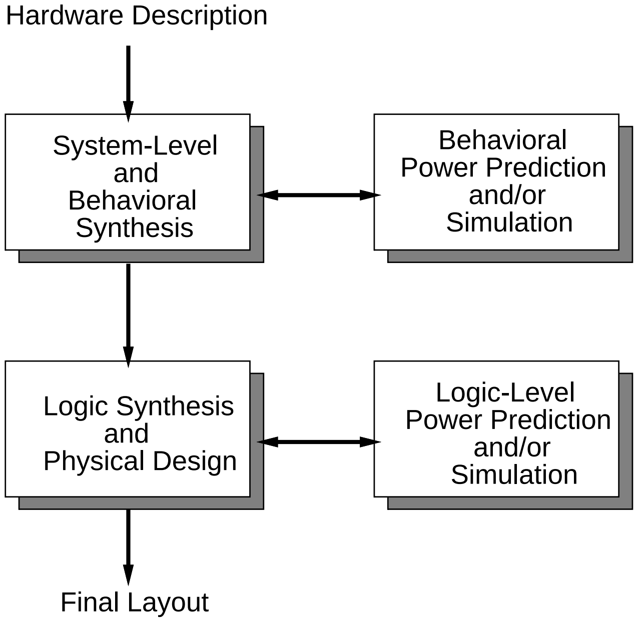
Hardware Description
System-Level and Behavioral Synthesis
Behavioral Power Prediction and/or Simulation
Logic Synthesis and Physical Design
Logic-Level Power Prediction and/or Simulation
Final Layout

Figure 1: The design flow

The remainder of the paper is organized as follows. In Section 2, various power simulation and estimation techniques are reviewed. In Sections 3 and 4, state of the art in combinational and sequential logic synthesis and layout optimization targeting low power consumption will be described. Concluding remarks will be given in Section 5.

2 Power Estimation Techniques

2.1 Sources of Power Dissipation

Power dissipation in CMOS circuits is caused by three sources: the (subthreshold) leakage current which arises from the inversion charge that exists at the gate voltages below the threshold voltage, the short-circuit current which is due to the DC path between the supply rails during output transitions and the charging and discharging of capacitive loads during logic changes.

The subthreshold current for long channel devices increases linearly with the ratio of the channel width over channel length and decreases nearly exponentially with decreasing $V_{GT} = V_{GS} - V_T$ where V_{GS} is the gate bias and V_T is the threshold voltage. Several hundred millivolts of "off bias" (say, 300-400 mV) typically reduces the subthreshold current to negligible values. With reduced power supply and device threshold voltages, the subthreshold current will however become more pronounced. In addition, at short channel lengths, the subthreshold current also becomes exponentially dependent on drain voltage instead of being independent of V_{DS} (see [15] for a recent analysis). The subthreshold current will remain 10^2 - 10^5 times smaller than the "on current" even at submicron device sizes.

The short-circuit power consumption for an inverter gate is proportional to the gain of the inverter, the cubic power of supply voltage minus device threshold, the input rise/fall time, and the operating frequency [41]. The maximum short circuit current flows when there is no load; this current decreases with the load. If gate sizes are selected so that the input and output rise/fall times are about equal, the short-circuit power consumption will be less than 15% of the dynamic power consumption. If, however, design for high performance is taken to the extreme where large gates are used to drive relatively small loads, then there will be a stiff penalty in terms of short-circuit power consumption.

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

$$P = \frac{V_{dd}^2}{2T_{cycle}} \sum_g C_g E_g(sw)$$

where V_{dd} is the supply voltage, T_{cycle} is the clock cycle time, C_g is the capacitance seen by gate g and $E_g(sw)$ is the expected number of transitions at the output of g per clock cycle.

Calculation of $E(sw)$ is difficult as it depends on (1) the input patterns and the sequence in which they are applied, (2) the delay model used and (3) the circuit structure.

Switching activity at the output of a gate depends on not only 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 $E_g(sw) = 3/8$. Now suppose it is known that only patterns 00 and 11 can be applied to the gate inputs and that both patterns are equally likely, then $E_g(sw) = 1/2$. Alternatively, assume that it is known that every 0 applied to input i is immediately followed by a 1 while every 1 applied to input j is immediately followed by a 0, then $E_g(sw) = 4/9$. The first case is an example of spatial correlations between gate inputs while the second case illustrates temporal correlations on gate inputs.

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). It is shown in [1] that although the mean value of the ratio of hazardous component to the total power dissipation varies significantly with the considered circuits (from 9% to 38%), the hazard/glitch power dissipation cannot be neglected in static CMOS circuits. Indeed, an average of 15-20% of the total power is dissipated in glitching. The glitch power problem is likely to become even more important in future scaled technology.

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 strictly required to perform the transition between the initial and final state of each node.

The major difficulty in computing the signal probabilities is the reconvergent nodes. Indeed, if a network consists of simple gates and has no reconvergent fanout stems (or nodes), then the exact signal probabilities can be computed during a single post-order traversal of the network. For networks with reconvergent fanout, the problem is much more difficult.

2.2 Circuit- and Switch-Level Simulation

Circuit simulation based techniques ([20, 39]) simulate the circuit with a representative set of input vectors. They are accurate and capable of handling various device models, different circuit design styles, dynamic / precharged logic tristate drives, latches, flip-flops, etc. Although circuit level simulators are accurate, flexible and easy-to-use, they suffer from memory and execution time constraints and are not suitable for large, cell-based designs. In general, 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 [29].

	Circuit-Level	Intermediate-Level	Switch-Level	Logic-Level
Example	SPICE	PowerMill	IRSIM	SIS
Accuracy	++	+	−	−−
Versatility	++	+	−	−−
Efficiency	−−	−	+	++

Table 2: Comparing various pattern-dependent simulators

A Monte Carlo approach for power estimation which alleviates this problem has been proposed in [6]. The convergence time for this approach is quite good when estimating the total power consumption of the circuit. However, when signal probability (or power consumption) values on individual lines of the circuit are required, the convergence rate is not as good.

Switch-level simulation techniques are in general orders of magnitude faster than circuit-level simulation techniques, but are not as accurate or versatile.

PowerMill [11] 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. PowerMill gives detailed power information (instantaneous, average and rms current values) as well as the total power consumption (due to steady-state transitions, hazards and glitches, transient short circuit currents, and leakage currents). It also tracks the current density and voltage drop in the power net and identifies reliability problems caused by EM failures, ground bounce and excessive voltage drops.

Entice-Aspen [16] is a power analysis system which raises the level of abstraction for power estimation from the transistor level to the gate level. Aspen computes the circuit activity information using the Entice power characterization data as follows. 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.

In summary, accuracy and efficiency are the key requirements for any power analysis prediction tool. PowerMill and Entice-Aspen are steps in the right direction as they provide intermediate level simulation that bridges the gaps between circuit-level and switch-level simulation paradigms (see Table 2).

2.3 Estimation in Combinational Circuits

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, we can write:

$$E_n(sw) = 2prob(n)(1 - prob(n)).$$

Computing signal probabilities has attracted much attention [28, 18, 32, 7, 26]. These works describe various exact and approximate procedures for signal probability calculation. Notable among them is the exact procedure given in [7] which is based on the Ordered Binary-Decision Diagrams (OBDDs) [4]. This procedure which is linear in the size of the corresponding function graph (the size of the graph, of course, may be exponential in the number of circuit inputs). The signal probability at the output of a node is calculated by first building an OBDD corresponding to the global function of the node and then performing a postorder traversal of the OBDD using equation:

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

where f_x and $f_{\bar{x}}$ are the cofactors of f with respect to x and $\bar{x}$, respectively.

The spatial correlation among different signals are modeled in [12] where a procedure is described for propagating signal probabilities from the circuit inputs toward the circuit outputs using only pairwise correlations between signals and ignoring higher order correlation terms.

	With		Without	
	Spatial Correlations			
	With	Without	With	Without
	Temporal Correlations			
Max	0.0463	0.2020	0.2421	0.2478
Mean	0.0115	0.0591	0.0658	0.0969
RMS	0.0185	0.0767	0.0722	0.1103
STD	0.0149	0.0505	0.0960	0.0544

Table 3: Effect of spatio-temporal correlations on switching activity estimation

The temporal correlation between values of some signal x in two successive clock cycles are modeled in [22] 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 $prob^x_{ij}$ that x will go to state j at time $t + 1$ if it is in state i at time t. The transition probability $prob(x_{i \to j})$ is equal to $prob(x = i) prob^x_{ij}$. The activity factor of line x is expressed in terms of these transition probabilities as:

$$E_x(sw) = prob(x_{0 \to 1}) + prob(x_{1 \to 0}).$$

The transition probabilities can be computed exactly using the OBDD representation of the logic function of x in terms of the circuit inputs. An approximate mechanism for propagating the transition probabilities through the circuit is also described in [22] which is more efficient, as the function of each node can be built in terms of the immediate inputs of that node, but is less accurate. The loss is often small while the computational saving is significant. This work is then extended to account for spatio-temporal correlations (i.e., spatial correlations between temporally-dependent events).

Table 3 gives various error measures (compared to exact values obtained from exhaustive binary simulation) for pseudo-random input sequences applied to the *f51m* benchmark circuit. It can be seen that accounting for either spatial or temporal correlations improves the accuracy while the most accurate results are obtained by considering both spatial and temporal correlations.

In summary, the OBDD-based approach is the best choice for signal probability calculation if the OBDD representation of the entire circuit can be constructed. Otherwise, a circuit partitioning scheme which breaks the circuit into blocks for which OBDD representations can be built is recommended. In this case, the correlation coefficients must be calculated and propagated from the circuit inputs toward the circuit outputs in order to improve the accuracy.

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

The concept of a probability waveform is introduced in [5]. This waveform consists of a sequence of transition edges or events over time from the initial steady state (time 0^-) to the final steady state (time ∞) where each event is annotated with an occurrence probability. The probability waveform of a node is a compact representation of the set of all possible logical waveforms at that node. Given these waveforms, it is straight-forward to calculate the switching activity of x which includes the contribution of hazards and glitches. 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 [27].

A tagged probabilistic simulation approach is described in [35] 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. Next, each group is combined into a probability waveform with the appropriate steady-state tag. Given the tagged probability waveforms at the input of a simple gate, it is

	Zero-delay	Real Delay		
		Probabilistic	Tagged Probabilistic	Symbolic
Accuracy	−−	−	+	++
Efficiency	++	+	−	−−

Table 4: Comparing various pattern-independent estimators

then possible to compute the tagged probability waveforms at the output of the gate. The correlation between probability waveforms at the inputs is approximated by the correlation between the steady state values of these lines. This 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 2%.

In summary, symbolic simulation provides the exact switching activity values under a real delay model. It is however very inefficient and impractical, but for small circuits. Probabilistic simulation and its tagged variant constitute the best choice for switching activity estimation at the gate level (see Table 4).

2.4 Estimation in 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 state probability method. Indeed, accurate average switching activity estimation for sequential circuits is considerably more difficult than for combinational circuits, because the probability of the circuit being in each of its possible states has to be calculated.

The exact state probabilities of a sequential machine are calculated in [37] and [24] by solving the Chapman Kolmogorov (C-K) equations for discrete-time, discrete-state Markov process. This method requires the solution of a linear system of equations of size 2^N, where N is the number of flip-flops in the machine. Thus, this method is limited to circuits with < 15 flip-flops, since it requires the explicit consideration of each state in the circuit.

A framework for exact and approximate calculation of switching activities in sequential circuits is also described in [25]. The basic computation step is the solution of a non-linear

	Combinational Techniques	Sequential Techniques	
		Non-linear Equations	C-K Equations
Error	30-60%	5-10%	None
Run Time	1	5-8	Exponential

Table 5: Comparing various power estimation techniques for sequential logic circuits

algebraic system of equations in terms of the signal probabilities of the present state and combinational inputs of the FSM. The fixed point (or zero) of this system of equations can be found using the Picard-Peano or Newton-Raphson iteration. Increasing the number of variables or the number of equations in the above system results in increased accuracy. 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 have significantly greater inaccuracies (see Table 5).

3 Logic Synthesis for Low Power

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 switching activity of the logic (weighted by the capacitive loading) that determines the power consumption of a circuit. In this section, a number of techniques for power estimation and minimization during logic synthesis will be presented. The strategy for synthesizing circuits for low power consumption will be to restructure/ optimize the circuit to obtain low switching activity factors at nodes which drive large capacitive loads.

Both the switching activity and the capacitive loading can be optimized during logic synthesis. It therefore has more potential for reducing the power dissipation than physical design. On the other hand, less information is available during logic synthesis, and hence, factors such as slew rates, short circuit currents, etc. cannot be captured properly. In the following, we present a number of techniques for power reduction during sequential and combinational logic synthesis which essentially target dynamic power dissipation under a zero-delay or a simple library delay model.

Retiming

In [23], it is noted that the flip-flop output may make at most one transition when the clock is asserted. Based on this observation, the authors then describe a circuit retiming technique targeting low power dissipation. The technique does not produce the optimal retiming solution as the retiming of a single node can dramatically change the switching activity in a circuit and it is very difficult to predict what this change will be. 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.

State Assignment

In the past, many researchers have addressed the encoding problem for minimum area of two-level or multi-level logic implementations (e.g., NOVA and JEDI). A state assignment procedure is presented in [30] which minimizes the switching activity on the present state input lines. This formulation however ignores the power consumption in the combinational logic that implements the next state and output logic functions. A state assignment technique that overcomes this shortcoming is presented in [34]. Experimental results on a large number of benchmark circuits show 10% and 17% power reductions for two-level logic and multi-level implementations, respectively.

Multi-Level Network Optimization

Network don't cares can be used for minimization of nodes in a boolean network [31]. Two multi-level network optimization techniques for low power are described in [33] and [19]. One difference between these procedures and the procedure in [31] 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 weighted switching activity. In addition,

[19] considers how changes in the global function of an internal node affects the switching activity (and thus, the power consumption) of nodes in its transitive fanout. Power consumption in a combinational logic circuit has been reduced by some 10% as a result of this optimization.

Common Subexpression Extraction

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 [3]. The kernel extraction procedure is modified in [30] 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. Results show 12% reduction in power compared to a minimum-literal network.

Path Balancing

Balancing path delays reduces hazards/glitches in the circuit which in turn reduces the average power dissipation in the circuit. This 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 can be performed so as to minimize the level difference between the inputs of nodes which are driving high capacitive nodes. 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 appropriate signal to pin assignment. This is possible as the delay characteristics of CMOS gates vary as a function of the input pin which is causing a transition at the output.

Technology Decomposition

It is difficult to come up with a 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.

Given the switching activity value at each input of a complex node, a procedure for AND decomposition of the node is described in [36] which minimizes the total switching activity in the resulting two-input AND tree under a zero-delay model. 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. It is shown that the low power technology decomposition reduces the total switching activity in the networks by 5% over the conventional balanced tree decomposition method.

Technology Mapping

A successful and efficient solution to the minimum area mapping problem was suggested in [21] 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 [36]. This approach consists of two steps. In the first step, power-delay curves (that capture power consumption versus arrival time tradeoffs) 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.

Figures 2 and 3 compare the results of this power-delay mapper with the area-delay mapper of [9] for the $s832$ benchmark circuit. From Figure 2, we can see that the power-delay mapper reduces the number of high switching activity nets at the expense of increasing the number of low switching activity nets. From Figure 3, we can see that for the remaining high switching activity nets, the power-delay mapper 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.

Figure 3: Average load per net vs. switching rate for $s832$

Signal-to-Pin Assignment

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 sizable impact on the propagation delay through the gate.

If we ignore the power dissipation due to charging and discharging of internal capacitances, it becomes obvious that high switching activity inputs should be matched with pins that have low input capacitance. However, the internal power dissipation also varies as a function of the switching activities and the pin assignment of the input signals. To find the minimum power pin assignment for a gate g, one must solve a difficult optimization problem [38]. Alternatively, one can use heuristics, for example, a reasonable heuristic assigns the signal with largest probability of assuming a controlling value (zero for NMOS and one for PMOS) to the transistor near the output terminal of the gate. The rationale is that this transistor will switch off as often as possible, thus blocking the internal nodes from non-productive charge and discharge events.

Table 6 summarizes the reported or predicted (in one-three years) power reduction as a result of various logic synthesis steps. The percentage reduction is given based on our current low power design tools at USC, reported results in the literature, or our preliminary studies. Note that the power savings at the different design phases are, in the best case, additive. For example, a 10% power savings from network optimization together with a 15% power savings from state assignment will yield a total power savings of $(1 - 0.9 \times 0.85) \times 100 = 23.5\%$. However, more often than not, the total power savings is less, say in this example 20%, since the various optimizations may adversely affect each other.

4 Physical Design for Low Power

Physical design fits between the netlist of gates specification and the geometric (mask) representation known as the layout. It provides the automatic layout of circuits minimizing some objective function subject to given constraints. Depending on the target design style (full-custom, standard-cell, gate arrays, FPGAs), the packaging technology (printed circuit boards, multi-chip modules, wafer-scale integration) and the objective function (area, delay,

Optimization	% Power Reduction
Retiming	10-15
State Assignment	15-30
Two-Level Minimization	10-25
Network Optimization	10-20
Subexpression Extraction	10-30
Path Balancing	5-10
Technology Decomposition	5-10
Technology Mapping	20-40
Pin Assignment	10-15

Table 6: Power reduction due to logic synthesis

power, reliability), various optimization techniques are used to partition, place, resize and route gates.

Under a zero-delay model, the switching activity of gates remains unchanged during layout optimization, and hence, the only way to reduce power dissipation is to decrease the load on high switching activity gates by proper netlist partitioning and gate placement, gate and wire sizing, transistor reordering, and routing. At the same time, if a real-delay model is used, various layout optimization operations influence the hazard activity in the circuit. This is however a very difficult analysis and optimization problem and requires further research.

Circuit Partitioning

Netlist partitioning is key in breaking a complex design into 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 FM heuristic [13]) 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

Floorplanning plays an important role during layout optimization as it determines the interface characteristics (shape, size, I/O locations) and positions of custom or semi-custom blocks in a hierarchical design environment. A floorplanner that considers power management is described in [8]. The idea is to generate a set of power-indexed shape functions and then use implementations for each flexible module that satisfies the timing constraints while minimizing the dynamic power dissipation. In addition, this work considers constraints to mitigate power line noises and thermal reliability problems. Results show 18% reduction in power and more smoothly distributed power dissipation over the floorplan area compared to conventional floorplanners with the same delay constraint. There is however a small area penalty.

Placement

A performance driven placement algorithm for minimizing the power consumption is presented in [40]. The problem is formulated as a constrained programming problem and is solved in two phases: global optimization and slot assignment. The objective function used during either phase is the total weighted net length where net weights are calculated as the expected switching activities of gates driving the nets. Constraints on total path delays are also accounted for. On average, this procedure reduces power consumption by about 10% at the expense of 2% increase in circuit delay compared to a placement program minimizing the total interconnection length.

Routing

Routing for low power can be performed by net weighting where again the net weights are derived from the switching activity values of the driver gates. The nets with higher weights are more critical and should be given priority during routing.

Gate Sizing

The treatment of gate sizing problem is closely related to finding a macro-model which captures the main features of a complex gate (area, delay, power consumption) through a small number of parameters (typically, width or transconductance). The first major contribution to transistor sizing problem was the work done in TILOS [14]. This optimization technique is greedy in the sense that they pick a path which fails to meet the timing requirements, and resize some transistor on the path so as to meet the constraint. The procedure is iterated until all timing constraints are satisfied or no further optimization is possible. A novel approach to gate sizing is described in [2]. This approach linearizes the path-based timing constraints and uses a linear programming solver to find the global optimum solution.

Wire Sizing

Wiresizing and/or driver sizing are often needed to reduce the interconnect delay on time-critical nets. Wiresizing however tends to increase the load on the driver and hence increase the power dissipation. [10] presents a combined wiresizing and driver sizing approach which reduces the interconnect delay with only a small increase in the power dissipation. 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. Alternatively, this approach produces delay values which are up to 40% lower when compared to the conventional method (at the cost of increasing the power dissipation by 25%).

Clock Tree Generation

Clock is the fastest and most heavily loaded net in a digital system. Power dissipation of the clock net contributes a large fraction of the total power consumption. A two-level clock distribution scheme based on area pad technology for MCMs is described in [42]. The first level of the tree is routed on the MCM substrate connecting the clock source to the clock area pads while the second level tree lies inside each die with the area pads as the source. The objective is to minimize the load on the clock drivers subjects to meeting a tolerable clock skew. A significant power reduction (70% for one benchmark circuit) over the method with one clock pad per die is reported by using this scheme.

Table 7 summarizes the reported or predicted (in one-three years) power reduction as a result of various layout synthesis steps.

5 Concluding Remarks

The need for lower power systems is being driven by many market segments. There are several approached 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

Optimization	% Power Reduction
Circuit Partitioning	10-30
Floorplanning	15-25
Placement	10-15
Routing	5-10
Transistor / Gate Sizing	10-30
Wire Sizing	10-25
Clock Tree Generation	10-30

Table 7: Power reduction due to layout optimization

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. That is, any power reduction through process scaling or voltage scaling should be above and beyond the 5-10x goals.

References

[1] 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.

[2] 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.

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

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

[5] A. 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.

[6] 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.

[7] 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.

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

[9] K. Chaudhary and M. Pedram. A near-optimal algorithm for technology mapping minimizing area under delay constraints. Proceedings of the 29th Design Automation Conference, June 1992.

[10] J. Cong, C-K. Koh, and K-S. Leung. Wiresizing with driver sizing for performance and power optimization. In *Proceedings of the 1994 International Workshop on Low Power Design*, pages 81–86, April 1994.

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

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

[13] C. M. Fiduccia and R. M. Mattheyses. A linear-time heuristic for improving network partitions. In *Proceedings of the 19th Design Automation Conference*, pages 175–181, June 1982.

[14] J. P. Fishburn and A. E. Dunlop. TILOS: A posynomial programming approach to transistor sizing. In *Proceedings of the* IEEE *International Conference on Computer Aided Design*, pages 326–328, November 1985.

[15] 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.

[16] 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.

[17] A. 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.

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

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

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

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

[22] 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*, November 1994.

[23] 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.

[24] 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*, page , June 1994.

[25] J. Monteiro, S. Devadas, B. Lin, C-Y. Tsui, M. Pedram, and A. M. Despain. Exact and approximate methods of switching activity estimation in sequential logic circuits. In *Proceedings of the 1994 International Workshop on Low Power Design*, pages 117–122, April 1994.

[26] F. N. Najm. Transition density, a stochastic measure of activity in digital circuits. In *Proceedings of the 28th Design Automation Conference*, pages 644–649, June 1991.

[27] 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.

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

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

[30] 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.

[31] 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.

[32] 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.

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

[34] C-Y. Tsui, M. Pedram, C-A. 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*, November 1994.

[35] 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.

[36] 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.

[37] 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*, page , June 1994.

[38] 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.

[39] 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.

[40] 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.

[41] 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.

[42] Q. Zhu, J. G. Xi, W. W-M. Dai, and R. Shukla. Low power clock distribution based on area pad interconnect for multichip modules. In *Proceedings of the 1994 International Workshop on Low Power Design*, pages 87–92, April 1994.