

# CAD for Low Power: Status and Promising Directions \*

Massoud Pedram  
University of Southern California  
Department of Electrical Engineering - Systems  
Los Angeles, CA 90089

## Abstract

Low power design is gaining increasing attention as the market for battery powered portable products expands and as power consumption becomes the stumbling block for further system integration. This paper examines strategies to minimize power consumption of digital circuits by reducing the supply voltage, by using power-conscious design methodologies and tools at the behavioral, logic and circuit levels, and by dynamic power management. The paper highlights some of the more effective and promising approaches for achieving ultra low power VLSI circuits and systems.

## 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 desk-tops, audio- and video-based multimedia products) as well as wireless communications and imaging systems (personal digital assistants, personal communicators, smart cards) that 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.

Synthesis and design tools equipped with power estimation capabilities could be used to observe the effects of various transformations and optimizations on key design space parameters: area, delay and power. Unfortunately, existing CAD systems do not have any power estimation tools at the behavioral synthesis levels. At the same time, they lack robust, accurate and efficient techniques for power estimation at the logic level. Ideally, the behavioral power prediction tools should interact with the logic level tools to improve their accuracy. Similarly, logic level tools should interact with the circuit-level power simulation techniques. Combining all of these capabilities in a single CAD framework, providing various mechanisms for back annotation of detailed power estimates into the higher levels, and technology and implementation style calibration of the high-level prediction tools are important open problems that must be addressed.

Most of the high-level power prediction tools use a combination of deterministic algorithm analysis, combined with

profiling and simulation to address data dependencies. The analytic models make simplifying assumptions about the word-level statistics (e.g., uniform white noise or dual bit-type data) and the spatial and/or temporal correlations (e.g., lack thereof). New methods are required that address these issues by permitting arbitrary word-level statistics and capturing these correlations. At the same time, the empirical models are significantly less accurate than their circuit and gate-level counterparts due to lack of detailed information about the switching activities and the physical capacitances.

A major shortcoming of the existing gate-level power analyzers is that they do not consider correlations among the circuit inputs (i.e., temporal interdependence of each input and concurrent transitions of multiple inputs). Another shortcoming is that the existing tools do not calculate the power dissipation due to hazards and short glitches accurately. Yet the hazardous component of dynamic power dissipation is the dominant component in certain circuits and in any case is not negligible. Finally, short circuit dissipation which is a strong function of input slew rates is currently not handled. The short circuit dissipation is however expected to rise in significance for low voltage submicron CMOS technologies.

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

It is necessary to develop behavioral synthesis techniques that also account for power dissipation in the circuit. This extends the two-dimensional optimization problem to a third dimension. The three phases of the behavioral synthesis process must be thus modified to produce low power circuits. Unfortunately, power dissipation is a strong function of signal statistics and correlations, and hence is non-deterministic. Automatic techniques that (1) minimize the switching activity on globally shared busses and register files, (2) combine architecture optimization with voltage scaling to allow trade-off between area and low-power, (3) exploit the input signal statistics to perform register and module allocation and binding for low power, (4) schedule operations to minimize the switching activity from one cycle step to next, (5) do loop transformations to minimize the switching activity, (6) generate code that has less memory operands and whenever possible use registers instead of memory, (7) minimize accessing distant or globally shared resources, etc. are needed.

\*This work was supported in part by ARPA under contract no. F33615-95-C-1627 and by SRC under contract no. 94-DJ-559.

Logic synthesis determines the gate-level representation of a circuit. 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 may be applied to transform and optimize the initial description.

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. Research in this area is focusing on the following: (1) Developing low power version of various logic minimization and restructuring techniques and doing this in such a way that the timing and/or area constraints are met, (2) Developing more accurate, yet simple, computational models for the slew rates, short circuit currents, parasitic wiring capacitances, signal statistics, etc.

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

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

## 2 Status

In the past, there was a very low effort on power estimation techniques and virtually no interest from industrial companies. It is only recently that more research is being done in this area and that companies are interested, pushed by a need for portable products and cheap packaging. In the following, we will discuss the existing and on-going efforts in design methodology and tool development targeting low power dissipation at various levels of the design hierarchy.

Most of the high level power prediction tools use a combination of deterministic algorithm analysis, combined with profiling and simulation to address data dependencies. Important statistics include the number of instructions of a given type, the number of bus, register and memory accesses and the number of I/O operations ([6, 21]), executed within a given period. Instruction level simulation or behavioral DSP

simulators are easily adapted to produce this information. A parametric model is described in [36], 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 and is sensitive to mismatches in the modeling assumptions. Analytic modeling efforts have been described in [29, 16] where a parameterized power model is developed for macro-modules. These models however ignore the correlations among various data busses, and hence, tend to produce inaccurate power estimates.

Assuming that the repetition frequency (sample, execution or instruction rate) is a given design constraint, the only means to reduce the projected dissipation of an application is by either reducing the supply voltage or the effective capacitance (i.e., the product of physical capacitance and the switching activity). A good overview of the use of optimizing transformations for supply voltage reduction is given in [6]. Concurrency increasing transformations include (time) loop unrolling, pipelining and control flow optimizations. The critical path of an application can be reduced by algebraic transformations, retiming and pipelining. There are many means of reducing the potential effective capacitance of an algorithm. The most obvious way is to reduce the number of operations by choosing either the right algorithm for a given function or by eliminating redundant operators (for instance, using dead code and common sub-expression elimination) [33]. Memory accesses often contribute a substantial part of the dissipation in both computational and signal processing applications. Replacing expensive accesses to background (secondary) memory by foreground memory references, or using distributed memory instead of a single centralized memory can cause a substantial power reduction [46]. Finally, selecting the correct data representation or encoding can reduce the switching activity [5].

Optimizing the instruction set is another means of reducing the power consumption in a processor. As an example, providing a special datapath for often executed instructions reduces the capacitance switched for each execution of that instruction (compared to executing that instruction on a general purpose ALU). In [35], it was demonstrated that choosing a Gray-coded instruction addressing scheme results in an average reduction in switching activity, equal to 37% over a range of benchmarks. In that same reference, a "cold scheduling" approach for traditional microprocessor architectures was proposed. In [39], a CISC-type instruction level power model for analyzing and minimizing the power dissipation of embedded software is presented.

At the behavioral synthesis level, there is a great need to develop automatic techniques for low power allocation, assignment and scheduling techniques, for concurrency increasing and critical path reducing transformations, and for dynamic power management. At the instruction level, a power model for RISC-type architectures must be developed and software compilation techniques for low power explored.

At the RT level, power estimation techniques based on information theoretic measures (entropy and informational energy) have been proposed in [25, 18]. Power prediction techniques at the logic level can be divided into two categories: simulation-based techniques (using SPICE, iRSIM-cap, PowerMill [10], Monte Carlo approach [4]) and probabilistic techniques [28, 26, 11, 32]. Probabilistic techniques for power estimation in combinational circuits have been recently extended to account for spatial and temporal correlations, sequential circuit behavior and parasitic capacitances

at the internal nodes of CMOS gates [42, 41, 43, 23, 19, 20]. These works must be extended to account for power dissipation due to wiring capacitances, slew rate, perturbation of gate delay parameters due to process or temperature variations. The impact of additional sources of power consumption (i.e., short-circuit and DC leakage currents) should be studied and conditions and design styles under which these sources of power consumption become important should be identified.

A number of techniques for reducing power consumption during behavioral and logic synthesis and physical design have been proposed in the recent past including, among others, techniques for using self-timed circuits and selective adjustment of the supply voltage [27], module allocation and binding and scheduling [30, 12], register allocation and binding [7], exploiting gated clocks during FSM synthesis [3], generating pre-computation logic [1], retiming [22], state assignment and re-encoding [31, 40, 13], kernel extraction [31, 15], multi-level network optimization [32, 14], technology decomposition and mapping [38, 17, 44], floorplanning [8], placement [45], transistor sizing and ordering [37], wire sizing [9] and clock tree generation [47]. Most of these techniques only consider power dissipation due to steady-state transitions and ignore the effect of hazards/glitches, interconnect capacitances, short-circuit currents and even leakage current (DC-leakage paths and subthreshold currents). The SOI technology, submicron device sizes, and lower voltage supply tends to exacerbate some of these second-order effects to a point where they cannot be ignored.

Providing a good library with a lot of different instances of the same cell (with different drive strengths) is important to give the technology mapping and sizing algorithms enough flexibility to optimize the circuit for power dissipation and to obtain solutions that come close to semi-custom designs. Studies such as that reported in [24] will be useful in developing a macro-cell library for low power applications.

Techniques which trade-off switching time for power dissipation during signal transition have not been incorporated into the optimization process. It is worthwhile to integrate these newly developed techniques into logic synthesis and physical design. For example, timing analysis can determine which signals can be “softly” switched without impacting overall performance. From this information, optimization and synthesis algorithms can be applied to evaluate and automatically insert logic for recovering signal energy [2].

A more detailed overview of the state-of-art in low power digital design, the impact of CAD and the challenges ahead is given in [34].

## 3 Promising Directions

### 3.1 Behavioral Level

A wide class of transformations can be done at the behavioral level and most of them are typically aimed at either reducing the number of cycles in a computation or reducing the number of resources used in the computation. One interesting approach is to introduce more concurrency in a circuit to speed it up and then to reduce the voltage until it realizes its originally required speed. The linear increase in capacitance due to parallelism is compensated for by the quadratic power reduction due to reducing the voltage. This can result in circuits that use several times less power. Although this transformation is not directly changing the supply voltage, it allows a design to operate with a lower supply voltage by increasing the concurrency. Another interesting approach is to reduce the supply voltage of each functional unit (thus reduc-

ing the power consumption, but increasing the delay of the unit) in the data path as much as possible while satisfying the timing requirements in terms of the cycle-time or throughput (in the case of pipelined circuits). This approach requires various support circuitry including level-converters and DC/DC converters.

It is desirable to keep the system power consumption close to a minimum level necessary to perform the required task. This can be achieved by partitioning the design into sub-circuits whose power dissipation levels can be independently controlled and by powering down sub-circuits which are not in use. We can also move the work to less power constrained parts of the system, for example, by performing the task on fixed stations rather than mobile sites.

It has been shown in both computational and signal processing applications that memory accesses often contribute a major part of the power dissipation. So it is attractive to concentrate on transformations that particularly aim at minimizing the power dissipation during memory accesses. Replacing expensive accesses to background memory (which switches larger capacitance per access) by foreground memory references (which switches smaller capacitance per access), or using distributed memory instead of a single centralized memory can cause a substantial power reduction.

Finally, selecting the correct data representation or encoding can reduce the switching activity. For instance, the almost universally used two’s complement notation has the disadvantage that all bits of the representation are toggled for a transition from 0 to  $-1$ , which occurs rather often. This is not the case in the sign magnitude representation, where only the sign bit is toggled. Choosing the correct data encoding can impact the dissipation in data signals with distinct properties, such as signal processing data paths, address counters and state machines.

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

### 3.2 Logic Level

Once the various system level, architectural and technological choices are made, it is the switched capacitance (that is, the product of the switching activity and the capacitive loading) that determines the power consumption of a circuit. In the remainder of this section, some techniques for reducing the switched capacitance subject to meeting the performance specifications will be enumerated. In general, the strategy for synthesizing circuits for low power consumption is to restructure the circuit to obtain low switching probability values at nodes that drive large capacitive loads.

State assignment of a finite state machine 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 uni-distance codes to states with high transition frequencies to one another. A more effective approach is to consider the complexity of the combinational logic resulting from the state assignment and to modify the objective

functions used in the conventional encoding schemes.

Network don't cares can be used to simplify each node so as to minimize the switching activity of the node. One should however consider 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. The impacts of this optimization on post-mapping area and delay must however be carefully considered.

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. The kernel extraction procedure can be modified 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.

Minimizing the average power consumption during technology mapping is proven to be simple and effective. The approach consists of two steps. In the first step, an optimized Boolean network is decomposed into two-input NAND and inverter gates such that the sum of average switching rates for all nodes in the network is minimum. In the second step, power consumption versus delay tradeoff curves are constructed and used during technology mapping to find a minimal power mapping for given timing constraints (subject to the error arising from unknown loads). Gate and/or transistor sizing for low power dissipation also promises to be quite effective.

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. 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.

### 3.3 Physical Level

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.

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) 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.

Placement algorithms can be easily modified to minimize the power dissipation. For example, a popular 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. The only change needed here is to use the total weighted net length as the objective function during each phase (net weights are calculated as the expected switching activities of gates driving the nets).

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. Alternatively, one can modify a hierarchical global routing procedure based on recursive construction of cut lines and linear assignment, to generate tree connections with smaller lengths for nets that are driven by gates with higher switching rates.

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 however reduce the interconnect delay with only a small increase in the power dissipation.

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. The objective of low power clock routing is to minimize the load on the clock drivers (and hence the clock tree length) subject to meeting a tolerable clock skew. Algorithms are needed for minimum Steiner tree routing with bounded difference between the shortest and the longest source to sink path length in the resulting tree and/or with bounded skew.

### 3.4 Power Management Strategies

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

Pre-computation logic may reduce the power dissipation in a data-path by a significant amount with marginal increases in circuit area and delay. The basic idea is to selectively pre-compute 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. In a combinational circuit, it is 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.

Power savings techniques that recycle the signal energies

using the adiabatic switching principles rather than dissipating them as heat are promising in certain applications where speed can be traded for lower power. Similarly, techniques based on combining self-timed circuits with a mechanism for selective adjustment of the supply voltage that minimizes the power while satisfying the performance constraints show good signs.

#### 4 Conclusion

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.

A number of researchers are investigating modeling and estimation of power consumption as well as techniques for minimizing power at the various levels of design abstraction (layout, logic, register-transfer, behavioral, system and algorithmic levels). The primary goal is to achieve a 10X reduction in power without sacrificing functionality and performance. To this end, they are developing general principles and novel techniques to guide the design of power-efficient electronic systems and explore how the availability of low-power design techniques impacts chip, module, and system level design decisions.

Power management strategies such as gated clocks, stoppable clocks, adaptive supply voltages, precomputation logic, energy recovery techniques, various power management modes, dynamic switching between power modes, etc. are also being researched and employed.

#### References

- [1] M. Alidina, J. Monteiro, S. Devadas, A. Ghosh, and M. Papaefthymiou. Precomputation-based sequential logic optimization for low power. *IEEE Transactions on VLSI Systems*, 2(4):426–436, December 1994.
- [2] W. C. Athas, L. J. Svensson, J. G. Koller, N. Thartzanis, and E. Chou. Low-power digital systems based on adiabatic-switching principles. *IEEE Transactions on VLSI Systems*, 2(4):398–407, December 1994.
- [3] 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*, April 1995.
- [4] 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.
- [5] A. P. Chandrakasan, R. Allmon, A. Stratakos, and R. W. Brodersen. Design of portable systems. In *Proceedings of the IEEE Custom Integrated Circuits Conference*, May 1994.
- [6] A. P. 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.
- [7] J-M. Chang and M. Pedram. Low power register allocation and binding. In *Proceedings of the 32nd Design Automation Conference*, June 1995.
- [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] 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.
- [10] C. Deng. Power analysis for CMOS/BiCMOS circuits. In *Proceedings of the 1994 International Workshop on Low Power Design*, pages 3–8, April 1994.
- [11] 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.
- [12] L. Goodby, A. Orailoglu, and P. M. Chau. Microarchitectural synthesis of performance-constrained, low power VLSI designs. In *Proceedings of the International Conference on Computer Design*, pages 322–326, October 1994.
- [13] G. D. Hachtel, M. Hermida, A. Pedro, 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.
- [14] 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.
- [15] S. Iman and M. Pedram. Logic extraction and decomposition for low power. In *Proceedings of the 32nd Design Automation Conference*, June 1995.
- [16] P. E. Landman and J. M. Rabaey. Power estimation for high level synthesis. In *Proceedings of the European Conference on Design Automation*, pages 361–366, February 1993.
- [17] B. Lin and H. de Man. Low-power driven technology mapping under timing constraints. In *International Workshop on Logic Synthesis*, pages 9a.1–9a.16, April 1993.
- [18] 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*, April 1995.
- [19] 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.
- [20] R. Marculescu, D. Marculescu, and M. Pedram. Efficient power estimation for highly correlated input streams. In *Proceedings of the 32nd Design Automation Conference*, June 1995.
- [21] R. Mehra and J. M. Rabaey. High level estimation and exploration. In *Proceedings of the 1994 International Workshop on Low Power Design*, pages 197–202, April 1994.

- [22] 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.
- [23] 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.
- [24] C. Nagendra, R. M. Owens, and M. J. Irwin. Power-delay characteristics of CMOS adders. *IEEE Transactions on VLSI Systems*, 2(3):377–381, September 1994.
- [25] F. Najm. Towards a high-level power estimation capability. In *Proceedings of the 1995 International Symposium on Low Power Design*, April 1995.
- [26] 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.
- [27] L. S. Nielsen, C. Niessen, J. Sparso, and K. 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.
- [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. R. Powell and P. M. Chau. Estimating power dissipation of vlsi signal processing chips: The pfa technique. In *VLSI Signal Processing IV*, pages 250–259, 1990.
- [30] A. Raghunathan and N. Jha. Behavioral synthesis for low power. In *Proceedings of the International Conference on Computer Design*, pages 318–322, October 1994.
- [31] 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.
- [32] 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.
- [33] S. Sheng, A. Chandrakasan, and R. Brodersen. A portable multimedia terminal. In *IEEE Communications Magazine*, pages 64–75, December 1992.
- [34] D. Singh, J. Rabaey, M. Pedram, F. Catthoor, S. Rajgopal, N. Sehgal, and T. Mozdzen. Power-conscious cad tools and methodologies: a perspective. In *Proc. of the IEEE*, April 1995.
- [35] 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.
- [36] 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.
- [37] 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.
- [38] 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.
- [39] V. Tiwari, S. Malik, and A. Wolfe. Power analysis of embedded software: A first step towards software power minimization. *IEEE Transactions on VLSI Systems*, 2(4):437–445, December 1994.
- [40] 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.
- [41] 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.
- [42] C-Y. Tsui, M. Pedram, and A. M. Despain. Power estimation considering charging and discharging of internal nodes of cmos gates. In *Proc. the Synthesis and Simulation Meeting and Int'l Interchange*, pages 345–354, October 1993.
- [43] 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.
- [44] 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.
- [45] 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.
- [46] 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.
- [47] 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.