Power Conscious CAD Tools and Methodologies: A Perspective

DEO SINGH, MEMBER, IEEE, JAN M. RABAEY, FELLOW, IEEE,
MASSOUD PEDRAM, MEMBER, IEEE, FRANCKY CATTHOOR, MEMBER, IEEE,
SURESH RAJGOPAL, MEMBER, IEEE, NARESH SEHGAL, MEMEBER, IEEE,
AND THOMAS J. MOZDZEN, MEMBER, IEEE

Invited Paper

Power consumption is rapidly becoming an area of growing concern in IC and system design houses. Issues such as battery life, thermal limits, packaging constraints and cooling options are becoming key factors in the success of a product. As a consequence, IC and system designers are beginning to see the impact of power on design area, design speed, design complexity and manufacturing cost. While process and voltage scaling can achieve significant power reductions, these are expensive strategies that require industry momentum, that only pay off in the long run. Technology independent gains for power come from the area of design for low power which has a much higher return on investment (ROI). But low power design is not only a new area but is also a complex endeavor requiring a broad range of synergistic capabilities from architecture/microarchitecture design to package design. It changes traditional IC design from a two-dimensional problem (Area/Performance) to a three-dimensional one (Area/Performance/Power). This paper describes the CAD tools and methodologies required to effect efficient design for low power. It is targeted to a wide audience and tries to convey an understanding of the breadth of the problem. It explains the state of the art in CAD tools and methodologies. The paper is written in the form of a tutorial, making it easy to read by keeping the technical depth to a minimum while supplying a wealth of technical references. Simultaneously the paper identifies unresolved problems in an attempt to incite research in these areas. Finally an attempt is made to provide commercial CAD tool vendors with an understanding of the needs and time frames for new CAD tools supporting low power design.

I. Introduction

Over the last year there has been increasing concern in system design houses, IC suppliers, and the academic community that power consumption of integrated circuits is beginning to impact design complexity, design speed, design area, and manufacturing costs. The motivation for this paper at this time (since the problem is still at a relatively early stage) is to convey an understanding of the breadth of the problem and the complexity of the solutions that need to be developed (in a relatively short period of time) to hold down cost by driving down power consumption. The paper is written in the form of a tutorial and is targeted at a wide audience, not only low power design experts and researchers in the field, but also system and IC designers that do not have a strong background in low power design and the issues surrounding low power design. It is also targeted at the commercial CAD community in the hope that the CAD companies will reexamine their development plans and develop new value added products to support low power design. As such, the authors have made a concerted effort to make the tutorial easy to read by keeping the technical depth to a minimum while supplying the reader with a wealth of references for further study.

A. Market Forces

Historically power consumption has not been a issue nor a high priority in the semiconductor industry. The industry was driven by the competitive market forces of simultaneously increasing integration and reducing chip area (and cost). In the DRAM market segment, the increase in integration came in the form of larger memories and higher memory density. In the high volume microprocessor market segment, integration was driven by the need to increase the performance. Higher performance was achieved through the use of new architectures (e.g., super scalar and super pipelined); integrating more and more system functions on chip, increasing the processor speed and increasing the processor's ability to perform computation on larger word sizes e.g., 8, 16, and 32 b. The trend in the large, and cost

Manuscript received May 21, 1994; revised August 25, 1994.

J. M. Rabaey is with the University of California, Berkeley, CA 94720 USA.

D. Singh, S. Rajgopal, N. Sehgal, and T. J. Mozdzen are with Intel Corp., Santa Clara, CA 95052–8119 USA.

M. Pedram is with the Department of Electrical Enginering Systems, The University of Southern California, Los Angeles, CA 90089–2562 USA.

F. Catthoor is with IMEC Laboratory, B-3030 Leuven, Belgium.

IEEE Log Number 9409205.

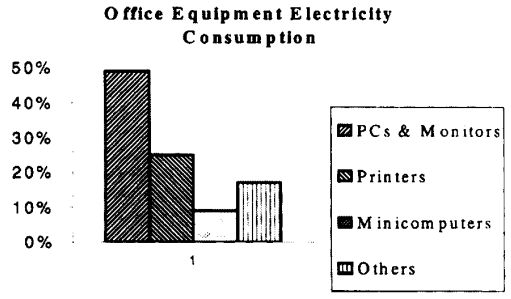


Fig. 1. Electricity consumption by equipment type source: Dataquest.

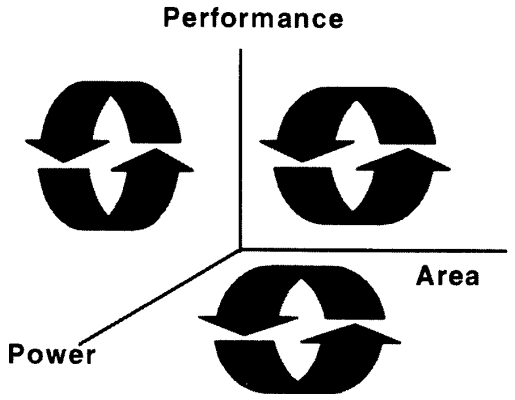


Fig. 2. Optimization changes from two dimensions to three dimensions.

sensitive, embedded-processor market segment is towards larger word sizes and higher frequencies.

The trend in the general purpose DSP markets is also towards higher performance. The embedded processors/microcontroller and DSP chips are targeted at large and significant systems markets such as consumer electronics (e.g., HDTV and video), mobile communications, Broadband ISDN, military, aerospace, surveillance systems, automotive, image processing, medical and office systems. In some of these markets the data throughput requirement is so high, e.g., Digital HDTV and Video that application specific DSP's are needed to deliver billions of operations per second (BOPS). The higher performance is achieved at the cost of greater switching activity.

Higher integration and speeds have resulted in increased power consumption and heat dissipation. The combination of higher speeds and shrinking geometries leads to higher on-chip electric fields which translate into a decrease in reliability. The increased heat dissipation leads to higher packaging costs e.g the addition of a heat sink could easily increase the component cost by \$5–\$10. The additional packaging costs become significant in the low-margin, price-sensitive embedded-processor market segment, as well as in the high volume microprocessor market segment. An additional \$5 cost is not easily absorbed in the embedded market place, and may be prohibitive in entering into this market segment. In the high volume microprocessor market the additional packaging cost could easily (over the life of the product) translate in additional costs of \$100 million.

From the system perspective increased power consumption is reflected in the higher system costs; system designers are now required to effect superior thermal design to rapidly remove unnecessary heat and maintain safe system temperatures, as well as developing more sophisticated power supplies.

Fig. 1 shows the power consumption of office equipment. The mobile computing market segment (notebook, handheld, and pen based computers) is predicted to grow from approximately 8% (in 1992) to approximately 32% (in 1996) of the total PC and personal workstation market [28].

This market segment compounds the foregoing power requirements with the need to operate the mobile product on a finite amount of energy; the equipment is battery powered! Although significant accomplishments had been made to extend battery life from 1–2 h (between charges) in the early notebook computers to about 4 h in state-of-the-art notebook computers, future more powerful processors are predicted to make even greater demands on the battery.

Although the mobile market segment will be a major driving force towards lower power in electronic equipment, it is by no means the only one. The US Environmental Protection Agency [28] estimates that over 80% of Office Equipment Electricity consumption is attributed to PC's, monitors, minicomputers, and printers, and the nonpower-managed PC's consume some 6.5 times the electricity consumption of the power managed systems.

B. Challenge

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

- Reducing chip capacitance through *process scaling*. This approach to reducing the power is very expensive and has a very low return on investment (ROI).
- *Reducing voltage*. While this approach provides the maximum relative reduction in power it is complex, difficult to achieve and requires moving the DRAM and systems industries to a new voltage standard. The ROI is low.
- Employing better *architectural and circuit design techniques*. This approach promises to be the most successful because the investment to reduce power by design is relatively small in comparison to the other two approaches.

The last approach (design for low power) does however, have one drawback, namely the design problem has essentially changed from an optimization of the design in two dimensions (performance and area) to optimizing a three-dimensional problem, i.e., performance, area, and power, see Fig. 2.

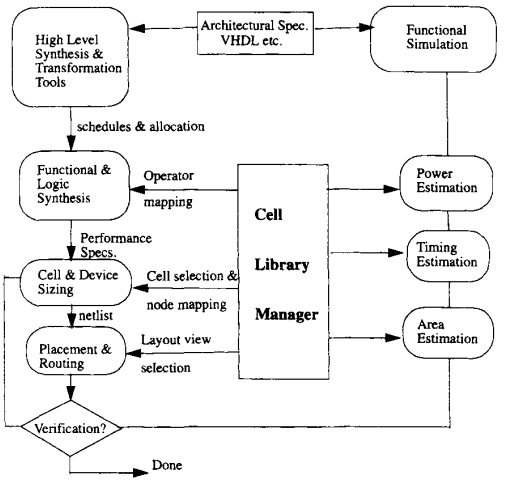


Fig. 3. Design data flow accessing a cell library at all phases

The design for low power problem cannot be achieved without good CAD tools. The remainder of this paper describes the CAD tools and methodologies required to effect efficient design for low power. As stated earlier, low power design is a relatively new problem and the paper is targeted at a wide audience to achieve the following:

- Convey an understanding of the breadth of the problem.
- Explain the state of the art of CAD tools and methodologies and as well as references to find additional more in-depth technical information in specific fields.
- Highlight the areas that need considerably more research.
- Assist the commercial CAD vendors in understanding the needs and time frames for new CAD tools to support low power design.

C. Structure Of Paper

It is well accepted that design is not a purely top down process, however, it is a convenient method of describing the low power activities at different levels of the design-data hierarchy as well as the flow of data and control. The different sections of the paper discuss the power issues in the context of a CAD system to support design for low power. Fig. 3 depicts the flow of data through the CAD system.

Complex systems tend to be specified with a mix of architectural and behavioral constructs, primarily because some parts of the system can be described by an algorithm and others cannot. It is assumed that the design starts off with a behavioral/architectural description of the product specification; DSP systems are more amenable to behavioral descriptions whereas general purpose microprocessor designs are more amenable to microarchitectural specifications. In the case of DSP systems, the behavioral description is transformed through a series of *behavior-preserving transformations* to generate microarchitectural specifications with timing. The microarchitectural description is then transformed and mapped to functional building blocks (FBB's) or templates representing executing units, controllers, and memory elements, etc. The elements of the functional building block are stored in an object-oriented library. Microarchitectural synthesis further refines the FBB's into hardware building blocks (again stored in the library) to meet performance, power, timing, and area constraints. The result of this phase is an RTL description describing datapaths, FSM's, and memories (registers together with necessary busses). The power estimation and optimization tools relating to this phase of the high level design are described in Section II.

Next the RTL description is processed by logic level tools, such as datapath compilers and logic synthesis tools to further optimize the design. Again the design is optimized for performance power, timing and area, and the tools make use of the design entities residing in the object-oriented library. The power estimation and optimization tools relating to this phase of the high level design are described in Section III.

Section IV describes the estimation and optimization techniques required to optimize the physical design. This section discusses partitioning, floorplanning, placement, routing and circuit sizing. It also discusses the role of the object-oriented library. The library is key to the successful implementation and support of the CAD tools and activities at all levels of the design data hierarchy as described in this paper. The library does not naturally fall into any one point of the design flow because it is an enabling technology and supports all phases of the design process; and is often taken for granted. It is described in this section of the paper to emphasize its importance to the layout activities.

Contemporary layout tools have to manipulate standard cell libraries containing a few thousand cells when making tradeoffs of performance and area. The flat library structure associated with mature libraries severely limit large numbers of rapid tradeoffs of power and area. New tools for power estimation and optimization will make the situation a practical impossibility. It is therefore important to apply new library solutions/mechanisms (discussed in Section IV), firstly at the layout level and then at other levels in a bottom up fashion. This approach should lead to a high value added solution.

Sections V and VI discuss reliability and noise-induced packaging issues. There are two factors that make it necessary to address these issues in the context of low power design. To begin with, low power design techniques are a cause for some of these problems, such as low noise margins due to voltage scaling, and increased power supply noise levels due to dynamic on-chip power management. Other noise problems such as simultaneous switching noise and signal crosstalk are due to the continuing demand for higher performance/watt (via higher frequency and increased device integration).

Each section of the paper follows a common theme. The scope of the problem is first established. This is then followed by discussions on the power analysis techniques and power optimization techniques.

The paper concludes with a summary and recommendations for low power tool development together with their priorities.

II. High Level Design

A. The Behavioral Level

While this is the least explored dimension of the power minimization process, it is potentially the one with the most dramatic impact. Selecting the right algorithm can reduce the power consumption of an application by orders of magnitude. Similarly, a single application can be represented in many different ways, one being more amenable than the other for low power implementation. These observations have been experimentally verified and demonstrated by a number of researchers [98], [18], [106] for various domains of such as audio/speech, image/video, telecommunications and networking. Similarly, finding the right system partitioning can have a profound impact on the overall power budget. For instance, when designing a wireless communication system, it makes sense to transfer the brunt of the power intensive functions to the immobile side of the system, relieving the power consumption at the battery powered portable end [98]. A combination of the preceding decisions can result in orders of magnitude of power reduction. This comes, however, at the price of decreased performance, increased latency or area, such that tradeoffs are necessary in practice.

In summary, the problem with power minimization at the behavioral level is that a designer has to tradeoff between multiple contradictory design parameters, the impact of which is mostly unknown in the early phases of the design process. The same holds for the architectural level design as well. As a result, the algorithmic and architectural specifications are often frozen early in the design process, based on "back of the envelope" computations. This reduces the power minimization space to the circuit and logic levels, where generally only limited reductions can be achieved. Providing the designer with a set of tools which can help him to explore, evaluate, compare and optimize the power dissipation alternatives early in the design process is a must to facilitate and enable fast and accurate low power design.

In the remainder of this section, we discuss the existing and ongoing efforts in design methodology and tool development at the behavioral and architectural levels. As is common throughout this tutorial, design tools are partitioned in two categories, being analysis/prediction and the synthesis/optimization.

1) Power Prediction/Estimation: A meaningful minimization process requires a reasonable estimate of the optimization target, in this case power consumption. Trying to predict the power consumption of an application in the early phases of the design process is a nontrivial task. Dissipation is a function of a number of parameters, as demonstrated in (1), the well known expression for dynamic power consumption in CMOS circuits:

$$P_{\text{dyn}} = \alpha C V_{DD} V_{\text{swing}} f \tag{1}$$

where C represents the physical *capacitance* of the design, V_{DD} the supply *voltage*, V_{swing} the logical voltage swing (which most often equals the supply voltage in CMOS design), α the switching *activity*, and f the *periodicity* of the computation, which is most often the sample, instruction or clock rate. Of those factors, especially the capacitance and activity are difficult to predict in the early design phases. The former requires a detailed knowledge of the implementation, while the latter is a strong function of the signal statistics and is, hence, nondeterministic. The two factors are often lumped together in a single parameter, called the effective or switching capacitance of a design [80]:

$$C_{\text{eff}} = \alpha C. \tag{2}$$

a) Determining the Switching Activity: The activity factor is what distinguishes power prediction from, for instance, area or performance analysis and makes it considerably harder. Activity is a function of both the algorithm structure and the data applied. Especially the latter factor is hard to determine deterministically and most of the activity analysis approaches, therefore, require extensive simulation or use empirical data.

As an example of the latter, Masaki [62], [63] determined the global switching activity factor for a variety of computing systems by analyzing their power consumption using both simulation or measurement. For minicomputers, α-factors between 0.01 and 0.005 are obtained, while microcomputers display a higher activity ranging from 0.05 to 0.01.

While the experimental approach is useful when performing a macroanalysis, its accuracy is insufficient to be of any help in algorithm selection and optimization. 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 [20], [64], executed within a given period. The advantage is that this analysis can be executed at a high abstraction level and can therefore be executed on a large data set without incurring a dramatic time penalty. Instruction level simulation or behavioral DSP simulators are easily adapted to produce this information (e.g., [10]). An example of such an approach is documented in [123], where the switching activity related to transitions in data and address signals for off-chip storage is tabulated by applying random excitations to a board level system model.

b) Determining the Capacitance: To translate the activity factors into actual energy measurements, they have to be weighted with the capacitance, switched per access of a given resource. It is important to observe here that, to a first degree, the actual number of instances of a resource does not impact the dissipation. For instance, performing N additions sequentially on a single adder consumes just as much power as performing the additions concurrently on N adders. This observation ignores the architectural overhead involved with either approach, as

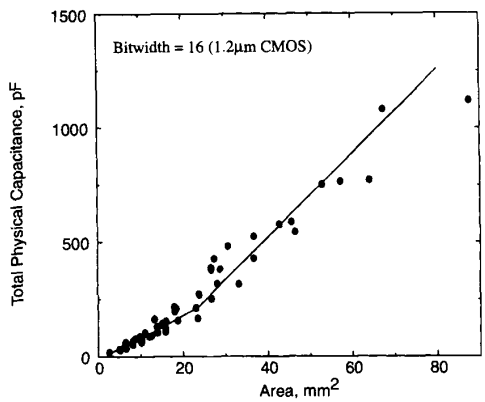


Fig. 4. Statistical estimation of interconnect bus capacitance (from [20]).

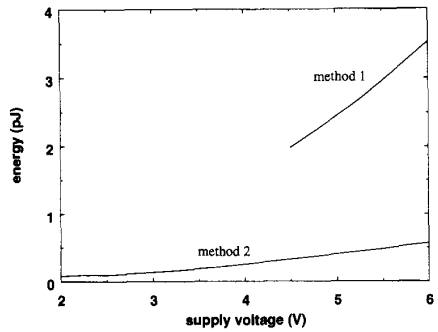


Fig. 5. Comparison of two different algorithms for vector comparison.

well as the impact of signal correlations. The architectural composition, for instance, has an impact on the size of the chip and, consequently, influences the cost of a bus access, an input/output operation or the clock network.

The power cost of accessing a resource is determined by its type. For computational units (adders, multipliers, registers) or for memories, it is possible to get measured or estimated data from the design libraries. At this point in the design process, it is hard to determine the actual switching properties of the data presented to the modules. Consequently, a white noise data model is generally adopted. While presenting a pessimistic view, this simplification is acceptable at this level of abstraction as it does not dramatically distort the correlation with the actual dissipation.

Given these modeling assumptions, the effective capacitance of the computational resources is then estimated by the following expression:

$$C_{\text{comp}} = \sum_{\text{all""Res}} N_{Exu} \cdot C_{Exu} + N_{\text{reg}} C_{\text{reg}} + N_{\text{mem}} C_{\text{mem}} \tag{3}$$

where res is the computational resource—Execution units (Exu), Memories (Mem), Registers (Reg), and C_{res} the capacitance switched per access over a given period.

Predicting the dissipation of other architectural components, such as I/O, busses, clocks and controllers, is more troublesome, as their capacitance is strongly influenced by the subsequent design phases (such as logic synthesis, floorplanning, and place and route). While just ignoring their contribution results in unacceptable results, the best that can be produced at this point is a reasonable estimate. In the high level synthesis community, there have been considerable efforts in estimating the wiring cost, given the active area of a design and other estimated parameters, such as the number of modules, the number of busses and their average bus-width and fanout and the number of control signals. The most popular approach is to generate the layouts for a large number of benchmark examples, collect the routing data and construct a stochastic model [49], [76].

A similar approach can be adapted to predict the average capacitance per bus. This is demonstrated in Fig. 4, which plots the measured bus-capacitance as a function of the chip area. The data is obtained from 50 benchmark examples generated using the HYPER synthesis [82] and the LAGER-IV layout generation [99] tools. The close correlation, which exists between area and bus capacitance, is easily captured in a simple piecewise linear model. The chip area, in turn, can be estimated from the algorithm and the performance constraints [45], [83].

Similar approaches can be used to produce early estimates of control and input/output capacitance [64]. It can be argued that the resulting models are only useful for a given set of design tools and methodologies. While it is obviously true that the model parameters will vary over methodologies and technologies, a more important result of these modeling attempts is the establishment of fundamental dependencies and relationships. The understanding of these dependencies can eventually open the door to more advanced models.

c) Applications of Behavioral Power Modeling: The resulting power model turns out to be particularly useful in the algorithmic optimization process and for design space exploration. This is illustrated with the example of a vector quantization encoder for video compression [64], [56]. A small variation in the formulation of the vector encoding algorithm can result in dramatic power reductions, yet maintaining the same performance. This is demonstrated in Fig. 5, where the energy/iteration is plotted in function of the supply voltage for two formulations of the algorithm. The second variant has the advantage of a reduced critical path (which enables lower voltage operation) and a smaller number of operations (which reduces the overall switching capacitance). Behavioral power estimators can be extremely useful in studying this type of tradeoffs. This is also demonstrated in [122] and [123].

2) Behavioral Level Power Minimization: Prior to a detailed design of the architecture, the only means to affect power is to modify or transform the specification, such that the resulting description is more amenable to low power solutions. 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.

a) Supply Voltage Reduction: Lowering the supply voltage is the most effective means of reducing power consumption as it yields a quadratic improvement in the power-delay product of a logic family. Unfortunately, this simple solution comes at a cost. First of all, finding commodity components that operate at different voltage levels (below 3.3 V) is hard, while multiple power supplies are to be avoided. Some of these issues can be addressed by the introduction of power-efficient dc-dc converters [101]. Reducing the supply voltage degrades the performance of the logic operators [18]. While the performance penalty is initially small (between 3 V and 5 V), a rapid decrease can be observed once V_{DD} approaches two times the transistor threshold. To maintain performance, this loss has to be compensated by other means. At the architectural level, this means using either faster components and operators or exploiting concurrency. Replacing a single fast operator, operated at high voltage, by a multitude of slow operators, running at low V_{DD}, does not impact the effective capacitance (ignoring the overhead of the parallel architecture). But it allows for a lowering of the supply voltage, which results in substantial power savings. This approach is generally known as trading area for power.

Design automation is key to successful exploration of the design space to effect area-power tradeoffs. To make an application amenable for this class of power optimization, it is essential that the algorithm contains sufficient concurrency and that its critical path is smaller than the maximum execution time (under the voltage range of interest). Optimizing transformations to achieve both those goals have been studied extensively in the compiler and high level synthesis communities [1], [78], [40], mostly for speed optimization purposes. The power minimization goal adds another twist to the problem, however: increasing the concurrency or reducing the critical path only makes sense as long as the effective capacitance factor does not increase faster. For instance, at low supply voltages, the amount of concurrency needed to compensate for the increase in propagation delay is so great, that the overhead becomes dominant and power dissipation increases.

A good overview of the use of optimizing transformations for supply voltage reduction is given in [19], [20]. Concurrency increasing transformations include (time) loop unrolling, pipelining, and control flow optimizations. The critical path of an application can be reduced by algebraic transformations [40], retiming and pipelining [54], [31].

b) Reduction in Effective Capacitance: 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). For instance, the computational requirements for the popular Discrete Cosine Transformation (DCT), used extensively in video compression, range over a factor of 10 [98]. Automated techniques have been reported, which allow to globally reduce the number of operations, weighted with an area/power cost factor over a range of algebraic manipulations and other transformations [46]. This has resulted in a near specification invariance for linear applications such as DCT and video filters.

Table 1 Impact of Behavioral Transformations on the Number of Address Line Transitions for a Segment Protocol Processor: Before (SPP1) and After (SPP2). Three Scheduling Simulations—Each Processing About 20k ATM Cells—While Varying an Important System Parameter (SSM), Have Been Listed

	SPP1	SPP2	Reduction
0% SSM	2,561,232	565,086	-39%
50% SSM	2,537,541	1,555,690	-39%
95% SSM	2,301,714	1,523,293	-34%

Another option is to reduce the "*power strength*" of an operation by replacing it with less demanding operations (this term is coined from the traditional strength reduction transformation in optimizing compilers [1]). An example of the latter is the expansion of a constant multiplication into a sequence of add/shift operations [19]. In the degenerated case, where no time multiplexing is performed, these shift operands can even be replaced by simple wiring.

Locality of reference is a prime feature to be pursued when optimizing an algorithm for power consumption [81]. Local operations tend to involve less capacitance and are, consequently, cheaper from a power perspective. Obviously, this locality is only meaningful when exploited properly at the architectural level. For instance, operations in the same subfunction of an algorithm should be allocated to the same processor or the same data path partition. This eliminates expensive data transfers across high capacitance busses. Transformations, improving the locality of reference, are abundant, for instance, in the domain of memory management, although most of the work in that domain has been oriented towards either area or performance optimization [120], [59] until recently.

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 [11]. An extensive study into the impact of memory management on system and architecture-level power consumption has been performed at IMEC [123]. This has shown that the external communication and memory access for e.g. a table-based telecom network subsystem dominates the power budget even when compared to the sum of all the other contributions (clock, data-path, and control). The combined results for this realistic test-vehicle taken from an ATM network, shown in Table 1, illustrate the extent of the improvements which can be obtained.

Finally, selecting the correct *data representation* or encoding can reduce the switching activity [21]. 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 [41], [46] address counters and state machines. In [103], 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. Similar techniques were employed in [123].

B. The Architectural Level

Once the architecture is defined and specified (e.g., using a functional or register–transfer level description), a more refined power profile can be constructed, which opens the way for more detailed optimizations. For instance, the impact of resource multiplexing can be taken into account. The architectural level is the design entry point for the large majority of digital designs and design decisions at this level can have a dramatic impact on the power budget of a design. The availability of efficient, yet accurate tools at this level of abstraction is therefore of utmost importance.

1) Power Analysis: Consider an architecture described at the structural level, consisting of an assembly of (parameterizable) modules and interconnecting buses and wires. Performing an accurate power analysis at this level of abstraction requires the following information:

- power models for the composing modules (library information)
- physical capacitance for the interconnect
- switching statistics for the interconnect signals.

The latter can be obtained from functional or register transfer level simulation. Given a functional model of the architecture, it might even be possible to derive the signal statistics analytically using, for instance, power spectral analysis techniques, long known in signal processing. This approach turns out to be hopelessly complex, however, and is only useful for linear, time-invariant systems (which are rare). Simulation driven by an actual test-bench is therefore the most appropriate technique to gather signal statistics.

As interconnect in general has a significant impact on the overall power budget, getting a meaningful estimate of its capacitance is important. Meaningful estimates can be obtained from stochastic models, as described in the behavioral section or from the initial floorplan. More accurate data can be back-annotated into the model once a more actual floorplan or layout has been produced.

While the abstract modeling of architectural macro-modules with regards to area and power is a well understood craft, providing high level power models has only recently attracted major attention. In a first attempt in that direction, Vermassen [118] derived power models for adders and multipliers, based on complexity theory [113] and a probabilistic analysis of the underlying gate structure.

Another parametric model was described by Svenson and Liu [104], where the power dissipation of the various components of a typical processor architecture were expressed as a function of a set of primary parameters. For instance, the capacitance of the clock wire, distributed in an H-tree structure, can be modeled as 24 $D_t c_{\text{int}}$, with D_t the chip dimension and c_{int} the interconnect capacitance per unit length. Models were developed for memory, logic, interconnect and clocks and were used to predict the global power consumption of a number of processors and ASIC chips. This modeling approach has the advantage of giving a global picture, suitable for driving optimizations and design tradeoffs. On the other hand, when accuracy is an issue, the technique suffers from an abundance of parameters and is sensitive to mismatches in the modeling assumptions. At that point, empirical models become more attractive.

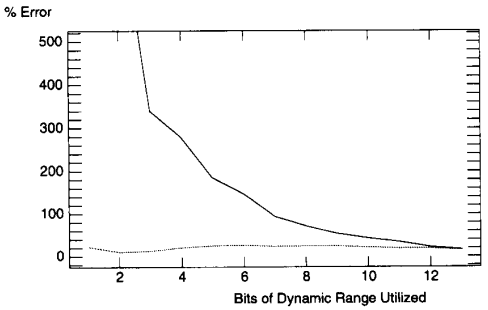


Fig. 6. Error in modeling multiplier power.

A first modeling effort in that direction was performed by Powell and Chau [79], who proposed a parameterized power model for macro-modules based on the following considerations. The power model for an array multiplier might, for instance, be represented as $P = C_u N^2 V^2 f$, where N is the width of the two inputs. Here, the quadratic dependence on N accounts for the N^2 adders of which a typical array multiplier is composed. Given this model, the characterization process consists of simulating or measuring the power dissipation of a multiplier, and fitting the capacitive coefficient, C_U, to these results. The problem with this approach is that the module's power consumption depends on the inputs applied. It being impossible, however, to characterize the module for all possible input statistics, purely random inputs—that is to say, independent uniform white-noise (UWN) inputs—are typically applied when deriving C_U.

This leads to an important source of error as illustrated by Fig. 6, which displays the estimation error (relative to switch-level simulations) for a 16×16 multiplier. Clearly, when the dynamic range of the inputs does not fully occupy the bit width of the multiplier, the UWN model becomes extremely inaccurate. Errors in the range of 60–100% are not uncommon.

A more precise model was proposed by Landman and Rabaey [50]. In the so-called dual bit type (DBT) model, it is projected that typical data in a digital system can be divided into two regions: the LSB's, which act as uncorrelated white noise and the MSB's, which correspond

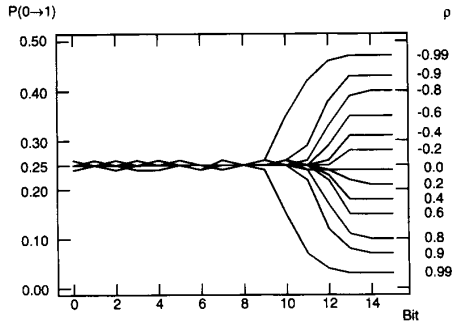


Fig. 7. Transition activity versus bit.

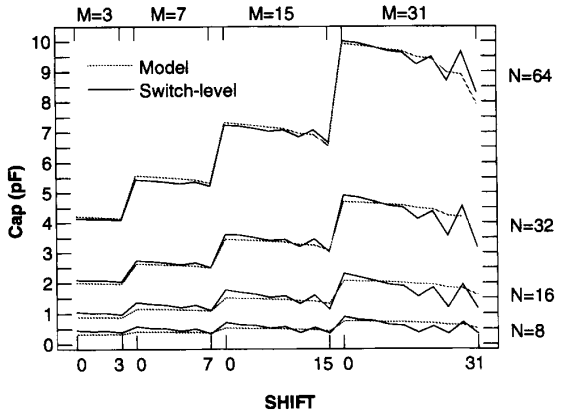


Fig. 8. Architectural model versus switch level simulated power dissipation of logarithmic shifter.

to sign bits. The latter are generally correlated between consequent data values and are far from random. This is illustrated in Fig. 7, which plots the transition probabilities for the different bit-positions in a data word as a function of the correlation factor ρ between consecutive samples. $\rho = 0$ means that the consecutive data samples are totally uncorrelated and, hence, correspond to white noise data. In that case, no difference exists between LSB and MSB bits. When a meaningful correlation is present between samples, an important deviation in transition activity between LSB and MSB's can be observed. The DBT model categorizes a hardware module for both those regions. A power model is automatically generated by applying a predefined set of test patterns (correlated and uncorrelated) to the module using a simulator of choice and extracting the switching capacitance for each of the regions of interest. The resulting power models are accurate to within 15% [51] as illustrated in Fig. 8, which compares the power consumption of a logarithmic shifter, obtained using both switch level simulations and the architectural model.

Once appropriate models for the modules are available, performing power prediction at the architectural or register transfer level becomes feasible by combining signal statistics, interconnect capacitance and macro-module power

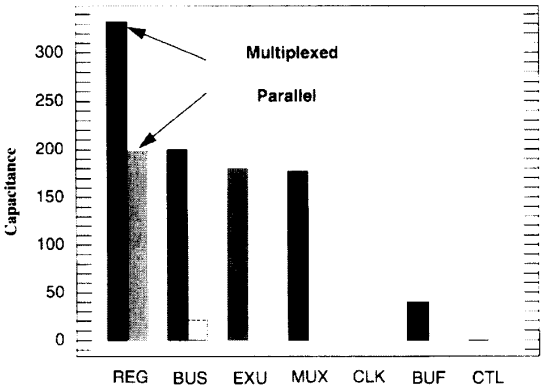


Fig. 9. Effective capacitance for architectural resources in multiplexed and parallel versions of QMF filter, as predicted by the SPA power prediction tool [50].

models. An example of a tool, doing just that, is the SPA architectural power analysis tool from University of California at Berkeley [50].

Fig. 9 illustrates the potential impact of architectural level power estimation and analysis tools. It compares the projected power for the various hardware resources of two architectures, implementing the same quadrature mirror filter (QMF). It is easily observed that the parallel version is far more power effective than the multiplexed one by virtually eliminating the bus and multiplexer effective capacitance [81]. This type of data is instrumental when performing power tradeoff and is hard to obtain from lower design hierarchy levels or "intuition."

2) Power Optimization: Similar to the behavioral level, architectural level optimizations can have a dramatic impact on power dissipations. Researchers have reported orders of magnitude in dissipation reduction by picking the proper architectural choice [106], [81], [21].

Although architectural level synthesis has received considerable attention in the last decade [121], virtually none of the efforts in this domain has been addressing the power dimension. Some initial results have been reported in [19], [64], [31], and [123]. In lieu of analyzing existing approaches, it is worth discussing where and how architectural synthesis can affect dissipation. Similar to the behavioral level optimization, architectural level power minimization targets a dual goal: 1) Minimize the required supply voltage to a minimal allowable level; and 2) minimize the effective capacitance. While the former is obtained at the architecture level by exploiting parallelism and pipelining [19], one of the main architectural means of addressing the latter is the use of locality of reference: whenever possible, accessing global, centralized resources (such as memories, buses or ALU's) should be avoided. How some of these goals can be accomplished at the different levels of the architectural synthesis process is discussed below.

a) Architecture Selection and Partitioning: The choice of the architecture model has a dramatic impact on power dissipation as it determines the amount of concurrency

which can be sustained, the amount of multiplexing of a hardware resource, the required clock frequency, the capacitance per instruction, etc. There is a widespread belief that the fully parallel, nonmultiplexed architecture yields the lowest possible dissipation with the minimum overhead [81], [21]. Unfortunately, such an architecture precludes programming or requires an unwieldy large area. Deciding on the right architecture for a given task or function is therefore a tradeoff between power, area, and flexibility.

Similarly, the partitioning of a task over a number of resources can have a substantial impact on the dissipation. For instance, in [56] it was shown that partitioning of the codebook memory in a vector encoder for video compression reduced the power consumption in the memories (which was the dominant factor in this design) with a factor of 8. In [11], it was demonstrated how the memory hierarchy, the caching approach and the cache sizes impacts the power cost of accessing the main memory in a microprocessor. A similar analysis was performed in [123].

Due to a lack of meaningful tools, the only recourse available to a designer at present is to perform back of the envelope calculations. The emerging architectural level power estimators will help to make this task more accurate and meaningful by identifying the power bottlenecks in an architecture. For instance, the analysis of the signal statistics of these tools can help to determine if and when power-down of a hardware module is desirable.

b) Instruction Set and Hardware Selection: 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). This approach is popular in application specific processors, such as modem and voice coder processors [5].

Another dimension of the power minimization equation is the choice of the hardware module to execute a given instruction. A detailed gate level study of adder and multiplication modules, for instance, revealed that energy structures such as carry-lookahead adders and Wallace multipliers, operate at lower energy levels than the more area effective ripple adder or carry save multiplier [14]. This study was later confirmed by Nagendra *et al.* [70], where the power-delay products of various adder structures were analyzed at the circuit level. The picture becomes more complex when additional optimizations such are operator pipelining are introduced.

Dealing with all these extra factors requires a profound revision of the hardware selection process in high level or architectural synthesis. The availability of functional level power models is a first step in that direction [122]. A vision on how cell libraries can interface and communicate with synthesis tools is presented in [95], which proposes an object oriented cell library manager. Based on extensive area, time and power modeling, the library manager helps to select the best fitting cell for a required functionality.

c) Architecture Synthesis: The architectural synthesis process traditionally consists of three phases: allocation, assignment, and scheduling. In a few words, these processes determine how many instances of each resource are needed (allocation), on what resource will a computational operation be performed (assignment) and when will it be executed (scheduling). Traditionally, architectural synthesis attempts to minimize the number of resources to perform a task in a given time or tries to minimize the execution time for a given set of resources. The underlying concept is to optimize resource utilization, under the assumption that an architecture where all resources are kept busy for most of the time is probably also the smallest solution.

This picture is not valid anymore when power minimization is the primary target. The smallest solution is not necessarily the one with the smallest dissipation. Quite the contrary is true. Indeed, it often makes sense to add extra units, if this contributes to minimizing the effective capacitance.

This difference has an important effect on the architectural synthesis techniques. The prime driving function for assignment now becomes the quest towards regularity and locality. Both help to reduce the overhead of interconnect, while simultaneously keeping the signals more correlated. Scheduling can have a similar impact, as it influences the correlation between the signals on the busses, memories and logic operators. This has been illustrated in [123], where operations were reordered over conditional boundaries. In [103], a "cold scheduling" approach for traditional microprocessor architectures was proposed,

While not specifically targeting power, a number of researchers have already explored some of these issues and some of the proposed approaches could become of particular interest in power sensitive architectural synthesis [87], [15], [37], [52].

The result of this phase of the design yields a timed RTL netlist used to drive logic level tools in the next stage of the design process.

III. Logic Level

Logic design and synthesis fits between the register transfer level and the netlist of gates specification. It provides the automatic synthesis of netlists minimizing some objective function subject to various constraints. Example inputs to a logic synthesis system include two-level logic representation, multilevel Boolean networks, finite state machines and technology mapped circuits. Depending on the input specification (combinational versus sequential, synchronous versus asynchronous), the target implementation (two-level versus multilevel, unmapped versus mapped, ASIC's versus FPGA's), 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 emphasis during power estimation will be on pattern-independent simulation techniques while the strategy for synthesizing circuits for low power consumption will be to restructure/optimize the circuit to obtain low switching activity factors at nodes that drive large capacitive loads.

A. Power Estimation Techniques

1) Sources of Power Dissipation: Power dissipation in CMOS circuits is caused by three sources: the (subthreshold) leakage current that arises from the inversion charge that exists at the gate voltages below the threshold voltage, the short-circuit current that 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 [35] 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 [117]. 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 approximately 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 severe 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:[1]

$$P_{\text{avg}}(g) = \frac{V_{dd}^2}{2 \cdot T_{\text{cycle}}} \cdot C_g \cdot E_g(sw) \qquad (4)$$

where V_{dd} is the supply voltage and T_{cycle} is the clock cycle time.

[1] Observe the similarity between (1) and (4). The difference of a factor between them is a result of deviation in the definitions of the period concept at the architectural and logic/circuit levels.

Calculation of $E(sw)$ is difficult as it depends on:

- input patterns and the sequence in which they are applied,
- delay model used,
- circuit structure.

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 $E_g(sw) =$ 2.1/4.3/4= 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 correlation between gate inputs while the second case illustrates temporal correlation on spatially independent 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 [6] 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 there convergent 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) Circuit- and Switch-Level Simulation: Circuit simulation based techniques [47], [112] 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 [85].

A Monte Carlo approach for power estimation that alleviates this problem has been proposed in [13]. 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. A method for estimating the typical power consumption of synchronous CMOS circuits using a hierarchy of RTL and circuit-level simulators is presented in [116]. In this method, the number of clock cycles used for simulation is incrementally calculated depending on the considered circuit and on the specified accuracy.

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

PowerMill [30] is a transistor-level power simulator and analyzer that 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 provides diagnostic information (by using both static check and dynamic diagnosis) to identify the hot spots (which consume large amount of power) and the trouble spots (which exhibit excessive leakage or transient short-circuit currents). Finally, it 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 [36] is a power analysis system that 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. For each cell in the library, *Entice* requires a transistor level netlist that is generally obtained from the cell layout using a parameter extraction tool. In addition, a stimulus file is to be supplied 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 specifications, 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.

B. Estimation in Combinational Circuits

1) 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 $\mathrm{prob}(n)$, 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:

$$E_n(sw) = 2\mathrm{prob}(n)(1 - \mathrm{prob}(n)). \tag{5}$$

Computing signal probabilities has attracted much attention [75], [39], [96], [17], [72]. These works describe various exact and approximate procedures for signal probability calculation. Notable among them is the exact procedure given in [17] which is based on ordered binary-decision diagrams (OBDD's) [9]. This procedure 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

$$\mathrm{prob}(y) = \mathrm{prob}(x)\mathrm{prob}(f_x) + \mathrm{prob}(\bar{x})\mathrm{prob}(f_{\bar{x}}). \tag{6}$$

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 [32] 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.

The temporal correlation between values of some signal x in two successive clock cycles are modeled in [92] and [61] by a time-homogeneous Markov chain that has two states 0 and 1 and four edges where each edge $ij(i, j = 0, 1)$ is annotated with the conditional probability prob_{ij}^{x} that x will go to state j at time $t + 1$ if it is in state i at time t. The transition probability $\mathrm{prob}(x_{i\to j})$ is equal to $\mathrm{prob}(x = i)prob_{ij}^{x}$. The activity factor of line x can be expressed in terms of these transition probabilities as

$$E_x(sw) = \mathrm{prob}(x_{0\to 1}) + \mathrm{prob}(x_{1\to 0}). \tag{7}$$

The transition probabilities can be computed exactly using the OBDD representation of the logic function of x in terms of the circuit inputs. A approximate mechanism for propagating the transition probabilities through the circuit is described in [61] that is more efficient as there is no need to build the global function of each node in terms of the circuit inputs, but is less accurate. The loss is often

small while the computational saving is significant. This work is then extended in [61] to account for spatiotemporal correlations (i.e., spatial correlations between temporally dependent events).

Table 2 (taken from [61]) gives the various error measures (compared to exact values obtained from exhaustive binary simulation) for pseudo-random input sequences for the *f51m* benchmark circuit [124]. This is a small circuit with 8 inputs, 8 outputs, and 46 CMOS gates. 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 that 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.

2) 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 [38], 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 [12]. 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 that 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[73].

A tagged probabilistic simulation approach is described in [108] 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 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%.

Table 2 Effect of Spatio-Temporal Correlations on Switching Activity Estimation

	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

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.

3) Estimation in Sequential Circuits: Recently developed methods for power estimation have primarily focused on combinational logic. 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 [111] and [66] 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 [67]. The basic computation step is the solution of a nonlinear 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 can have significantly greater inaccuracies.

C. Logic Optimization Techniques

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 that essentially target dynamic power dissipation under a zero-delay or a simple library delay model.

1) Retiming: In [65], 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 technique heuristically selects a set of nodes with the property that if flip-flops are placed at their outputs, switching activity in the network will be reduced. Nodes are selected based on the amount of glitching that is present at their outputs and the probability that this glitching propagates through their transitive fanouts. 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.

2) Pre-Computation Logic: A sequential logic optimization method is described in [2] that is based on selectively precomputing the output logic values of the circuits one clock cycle before they are required, and using the precomputed values to reduced internal switching activity in the succeeding clock cycle. They present an automatic method of synthesizing precomputation logic so as to achieved maximal reductions in power dissipation. Up to 62% reduction in average switching activity and power dissipation are reported with marginal increases in circuit area and delay.

3) 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., [119] and [58]). A state assignment procedure is presented in [89] that minimizes the switching activity on the present state input lines. This problem is equivalent to embedding a weighted graph on a hypercube of minimum (or given) dimensionality which is an NP-hard problem. The authors use simulated annealing to solve this problem.

The shortcomings of the above approach are: (1) It minimizes the switching on the present state bits without any consideration for the loading on the state bits; (2) It does not account for the power consumption in the resulting two- or multi-level logic realization of the next state logic of the FSM. A state assignment technique that not only accounts for the power consumption at the state bit lines, but also the power consumption in the combinational logic that implements the next state logic function is presented in [107]. Experimental results on a large number of benchmark circuits show 10% and 17% power reductions for two-level logic and multilevel implementations, respectively.

4) Multi-Level Network Optimization: Network "don't cares" can be used for minimization of nodes in a boolean network [91]. Two multilevel network optimization techniques for low power are described in [97] and [44]. One difference between these procedure and the procedure in [91] 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, [44] 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. In particular, two types of local don't care conditions are identified: 1) *Function Preserving Don't Care* (FPDC) which consists of all points in the local space of the node that never occur and is simply the complement of the range of the primary input space into the space of the local fanins of the node and 2) *Function Modifying Don't Care* (FMDC) which consists of all points in the local space that do not produce observable outputs. FPDC does not affect the global function of the node while FMDC does. A greedy network optimization procedure based on the above concepts is introduced that reduces the power dissipation in the combinational circuits by about 10%.

5) 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 multilevel Boolean network [8] and [86]. The kernel extraction procedure is modified in [89] to generate multilevel circuits with low power consumption. The main idea is to calculate a power saving factor for each candidate kernel based on how its extraction will effect the loading on the its input lines and the among of logic sharing. Results show 12% reduction in power compared to a minimum-literal network.

6) 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 increased. This is however valid only if the node delay is determined by assuming an AND-OR implementation with the delay of AND and OR gates being a function of the number of inputs to the gate. Logic decomposition can be performed so as to minimize the level difference between the inputs of nodes that 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. This is a difficult task as delay insertion at some node will not only change the spurious activity at the immediate

output of the gate (hopefully, will reduce it), but will affect the spurious activity in the transitive fanout of the node (unfortunately, due to change in output waveforms and sensitizability conditions, the activity may be increased).

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 that is causing a transition at the output.

7) Technology Decomposition: It is difficult to develop a decomposed network that leads 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 [109] that 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 [43] 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. The performance-oriented version of the above problem requires that the increase in the height of the decomposed network (compared to the undecomposed network) be bounded. The solution here is similar to Larmore/Hirschberg's algorithm [53] for solving the tree decomposition problem with minimum average weighted path length subject to a height constraint. 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. This translates to a 3% reduction in power consumption after technology mapping.

8) Technology Mapping: A successful and efficient solution to the minimum area mapping problem was suggested in [48] 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 that can be performed optimally using dynamic programming.

The problem of minimizing the average power consumption during the technology dependent phase of logic synthesis is addressed in [109]. 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.

The algorithm is optimal for trees and has polynomial run time on anode-balanced tree. It is easily extended to mapping a network modeled by a directed acyclic graph.

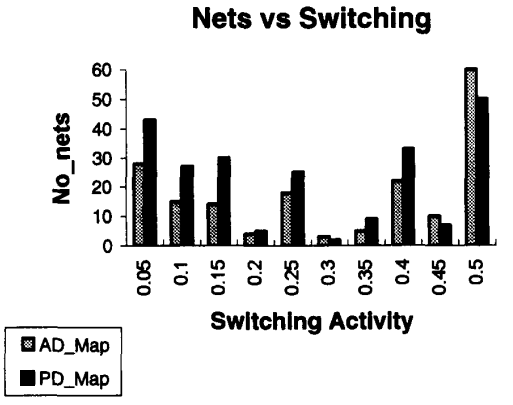


Fig. 10. Number of nets versus switching rate for *s832*.

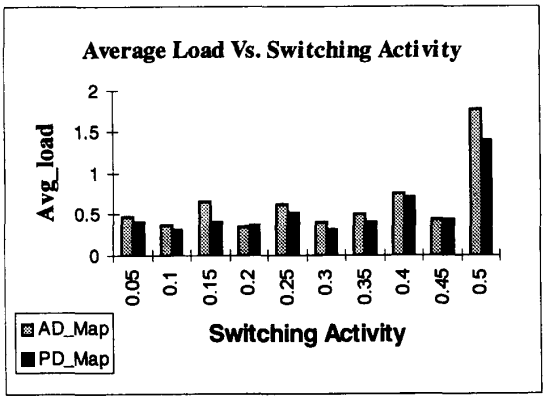


Fig. 11. Average load per net versus switching rate for *s832*.

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.

In Figs. 10 and 11, the results of this power-delay mapper are compared with the area-delay mapper of [23] for the *s832* benchmark circuit [124]. This circuit contains 18 inputs, 19 outputs, 245 product terms, and 25 flip-flops. After the mapping, the circuit contains about 200 CMOS gates (the exact gate counts are different for the area-delay and power-delay mappers). From Fig. 10, 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 Fig. 11, we learn 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.

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 considered in [108]. 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 values of the gate itself. The spatial correlations among the input waveforms are captured using the tagging mechanism described previously.

The concept of power-delay curve has been extended to include the area tradeoff. Instead of generating a set of (power, delay) values, the tradeoff curve consists of a set of (power, delay, area) values [57]. A modification of the SIS mapper for low power is described in [105].

9) 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 [16]. 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 [110]. 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. 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. Alternatively, one could assign the signal with the earliest transition to a controlling value to the transistor near the output terminal. The rationale is that this transistor will switch off as often as (or as early as) possible, thus blocking the internal nodes from nonproductive charge and discharge events.

Another heuristic [57] assigns the signal with highest switching activity to the input pin with the least capacitance. This is not very effective as in the semi-custom libraries, the difference in pin capacitances for logically equivalent pins is small.

The pin permutation for low power should take place on non-critical gates as it is in general different from the pin permutation for minimum delay.

IV. Physical Level

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

A. Layout Optimization Techniques

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 netlist partitioning and gate placement, gate and wire sizing, transistor reordering, and routing. Simultaneously, 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.

1) Circuit Partitioning: Netlist partitioning is key in breaking a complex design into pieces that 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 [33]) 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.

2) 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. [22] describes a floorplanner that considers power management. 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.

3) Placement: A performance driven placement algorithm for minimizing the power consumption is presented in [114]. 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

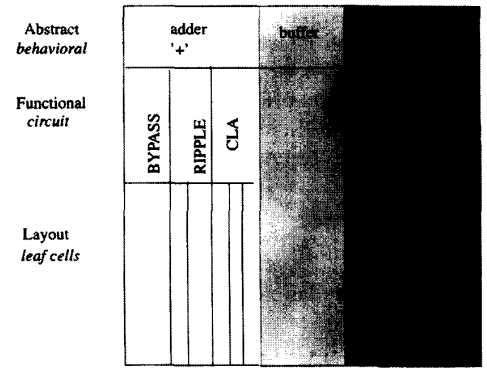


Fig. 12. The datapath library cell model.

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.

4) 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.

5) Gate Sizing: The treatment of gate sizing problem is closely related to finding a macro-model that 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 [34]. This optimization technique is greedy in the sense that they pick a path that 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 [7]. This approach linearizes the path-based timing constraints and uses a linear programming solver to find the global optimum solution. The drawbacks of this approach are the omission of slope factor (input ramp time) of input waveforms from the delay model and use of simple power dissipation model that ignores short-circuit currents.

6) Wire Sizing: Wire sizing 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 combined wire sizing and driver sizing approach that reduces the interconnect delay with only a small increase in the power dissipation is described in [26]. 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 that are up to 40% lower when compared to the conventional method (at the cost of increasing the power dissipation by 25%).

7) 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 MCM's is described in [126]. 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.

B. Libraries

1) Scope of the Problem: As VLSI layout design dimensions continue to shrink, device sizes tend to shrink faster than the routing distances. For future designs, it means that a significant part of the system power will be due to layout parasitics, that are not well modeled at logic and circuit levels. The conflicting needs for high design productivity, low power consumption, high layout density and higher performance goals are difficult to meet in acceptable run times without a pre-characterized library. Existing design methodologies map high level requirements to the cells in a library early in the design cycle. This causes some selection decisions to be made before all the constraints may be fully known, e.g., the pin positions on the neighboring cells. This calls for mismatched cell views during layout phase and consequently longer interconnect wires. Another issue is the growing size of the library, as the search time for a cell, with m constraints among n cells in a relational database using separately sorted tables of attribute values, is $O(n^m)$.

2) Solutions: One possible solution is to overlap the design steps so that the estimates done at the higher levels are more realistic. The tradeoff is in the increased complexity of estimated models and an effort to complete part of the layout design before the logic mapping is fully done. An approach is to use a library of cells with views at every design abstraction level for aiding the decision making process. The cell information is stored in a database which is queried at each design phase. At lowest level of hierarchy, various cells are stored as objects with attributes and at higher levels they are grouped by functionality. The concept of delayed binding is used in selecting the layout view of a library cell identified during technology mapping. Each layout view is designed without any pins on the boundary of the cell and multiple connection flexibility is available after final placement of cell instances during the layout phase A cell in the datapath library may have multiple levels of abstraction (i.e., behavioral, logic, circuit and layout), and multiple views at each level, as shown in Fig. 12 (i.e., an Adder at behavioral level will be the operator "+" but at circuit level the selection can be made between Carry Look-ahead, Ripple Carry style, etc.).

After an individual library cell design is completed, layout parasitics are extracted with simulated routing placed over-the-cell and this information is used to derive charac-

teristic model equations for current and power consumption in terms of actual load being driven, voltage, toggle rate as well the slope of input signal. These electrical equations along with behavioral models and attributes such as area and pin-to-pin delays are stored in an object-oriented database. The database implements search for equivalent functionality cells, and a designer can specify constraints on a cell performance to limit the search space. Since all the performance constraints may not be known at the beginning, i.e., device strengths are not determined until the circuit phase, a group of cells are identified during the logic phase for each component. As more details of a design become known, the size of the mapped group for each design component becomes smaller. If no corresponding cell is available as a new constraint is applied, that constraint is relaxed and a neighboring cell's constraint is tightened.

If more than one cells in the library match the final set of constraints, an objective function is computed over each selected cell (e.g., minimize power/area), and then cells are sorted accordingly to select the most desirable candidates. If no cell meets the given constraints, then a closest cell by relaxing the design constraints is found.

In order to do any kind of design or optimization for low power, a power model for the cells has to be available. In a design, sources of power dissipation in a design are as follows:

$$P_{\text{total}} = P_{\text{dynamic}} + P_{\text{static}} \tag{8}$$

with

$$P_{\text{dynamic}} = \alpha C V_{DD} V_{\text{swing}} f + I_{tsc} V_{DD} \tag{9}$$

the first part of which is identical to (1). I_{tsc} is the transient short circuit current, which is the direct current through the series P and N transistors in CMOS network during logic transition (i.e., transition time = tt)

$$P_{\text{static}} = (I_{\text{bias}} + I_{\text{leakage}}) V_{DD} \tag{10}$$

where I_{bias} is the d.c. current through sense amps, ratioed loads, and I_{leakage} is the circuit leakage current. During search, the value of these equations is computed at run-time and compared with the given constraints. Special algorithms using set theory to optimize the multiple constraints based query time have been developed.

Such a cell library and the database is used during low power design to meet the given budgets on area, performance and power consumption. As compared to a relational database, the object-oriented cell library manager reduces the search time for an appropriate cell, with m constraints among n cells, from $O(n^m)$ to $O(mlogn)$. A prototype implementation with 5000 library cells, with 92 attributes per cell, took only 8.6 s to satisfy power constraints with a matching cell [95].

V. Reliability Issues in Low Power Design

The primary objectives in the earlier sections have been the analysis and optimization of full-chip power across different stages of the design hierarchy. In this section and the next we discuss reliability and noise problems caused by high performance/high frequency design needs as well as low-power design techniques. Reliability verification (RV) addresses electromigration (EM) and the IR voltage drops on power supply lines. Continuing with the theme followed throughout this paper, we first discuss analysis capabilities to do reliability verification and optimization techniques to do design for reliability.

A. Scope of the Problem

Reliability issues are becoming important due to several design constraints—higher performance and frequency, device miniaturization, higher levels of on-chip integration, and lower supply voltages.

Device miniaturization can lead to several reliability problems—hot-electrons that cause threshold voltage shifts, and gate oxide breakdown leading to runaway current and electrostatic discharge (ESD) failures. High current densities from faster circuits, thinner wires and increased device count can cause EM failures resulting in opens in signal and power lines. Another problem caused by high average current (I_{avg}) is the IR voltage drops along the power lines. This along with lower power supply levels can degrade the noise margin and lead to functional failures. In this section we will not address device and material reliability issues, but will deal primarily with signal and power network reliability verification.

B. Reliability Verification

In order to meet EM reliability constraints, the average current in a bidirectional signal line needs to meet the following constraint [3].

$$I_{\text{avg}} = (C_{\text{unit}} l + C_L) V f_{wc} \leq J_{\max} W t \tag{11}$$

where C_{unit} is capacitance per unit length, C_L is the load capacitance, l is the line length, f_{wc} is the worst case switching frequency, $J_{\max}$ is the limit of current density for a specific layer and W and t are the line width and thickness respectively. Different conducting layers (Al, Cu) have different current density limits. The estimated I_{avg} will not only determines which layer to use but also the maximum line length (l) and width (W)

In order to determine IR drop limits on power supply lines, current density and voltage drop information needs to be extracted from the physical layout. The following is needed to accomplish this task.

1) The transistor netlist must be simulated to estimate the current flowing into and out of each power bus via.
2) The physical layout of the power buses must be accurately extracted into an *RC* network with current sources for the transistors.
3) This power network can then be simulated to find the current density and voltage at each node.

Techniques for circuit simulation (to extract I_{avg} and I_{peak}) have been introduced in earlier sections. For large networks obtaining the current waveforms for each transis-

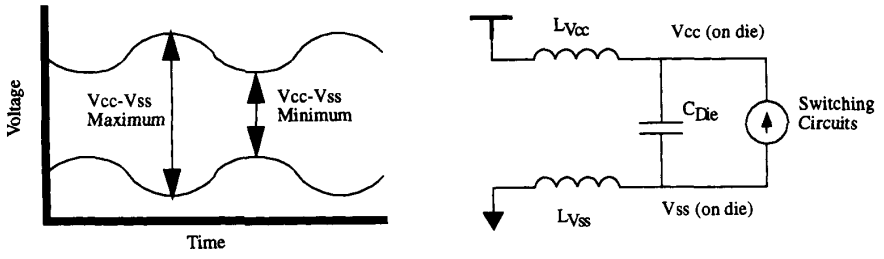


Fig. 13. V_{dd} and V_{ss} oscillations from the intrinsic LC circuit-in the device.

tor can be time-consuming. This drawback can be overcome by using switch-level and logic-level simulators [125], [29] or even pattern-independent simulation techniques (such as [112], [71]). The obvious tradeoff is accuracy.

Resistance extraction algorithms fall into three general categories: numerical solution of Laplace's equation [4], lumped approximation [94] and polygonal reduction [100], [103]. The most accurate method solves Laplace's equation to produce point-to-point resistances. This tends to be too slow for full-chip power supply analysis. The lumped approximation is fast but oversimplifies the network connectivity. The polygonal reduction method provides a speed/accuracy trade-off by producing smaller networks on which point-to-point resistance can be calculated. Capacitance extraction techniques too can vary from the simple [90] to the complex [115], [27].

The simulation of an *RC* network is a classic problem in circuit analysis that requires circuit formulation and matrix computation [25]. It may, however, be the bottleneck in the reliability verification (RV) process due to the size of power network, where millions of parasitic elements are common for today's VLSI chips. The number of equations in the matrix computation can reach 10 000 for a 10K gate VLSI circuit. Moreover, a matrix computation is needed for each timestep in the time-domain transient simulation. Simplified methods can speed up the *RC* network simulation by 1) reducing the massive *RC* network to a smaller size; or 2) limiting the number of power networks analyzed. In the former case parasitic elements are lumped into a small number of elements. For example, a wire modeled by a distributed *RC* is lumped to a simplified p-structure. Another simplification is to reduce the network graph to a tree and then solve for currents and voltages [77]. A second method performs the power network analysis only once for a pre-specified time interval such that average, rms, or peak current of each transistor over this interval is applied to the power network [125]. In the extreme case, only one matrix computation is needed to estimate the average current density and voltage drop. The speedup from these simplifications, however, is achieved at the expense of a degraded accuracy.

The output calculated in power net and signal RV analysis contains a large volume of data making it a tedious process to browse the current density and voltage drop at each location of the power network. For reliability analysis, a graphical display showing violations (and potential violations) where the current density or voltage drop exceeds (or is near) the user-specified threshold, is important for the designer to quickly redesign the power bus layout to meet reliability requirements.

C. Design for Reliability Optimization

For reliability design, signal and power lines need appropriate sizing to meet EM constraints. Adding repeaters to divide long lines into smaller subsections can also help reduce EM [74]. Power distribution can be improved to curb IR voltage drops. This can be done with an interweaved comb-like power distribution network. Using different metal layers for a global and local power distribution also helps [3]. The global network is best laid out in the topmost low-resistivity metal layer which should be thicker than the lower metal layers used for local power distribution. The effective resistance can be further reduced by increasing the number of power connections.

VI. Packaging Induced Electrical Noise Issues in Low-Power Design

Noise management tries to contain the problems caused by fluctuations of the V_{dd}/V_{ss} power supply levels. Many factors have aggravated this noise problem: faster transistors, higher current levels, shorter clock cycles, lower supply voltages, and power savings techniques. One might have the initial impression that low power design techniques must inherently improve the stability of the on die power supply levels, V_{dd} and V_{ss}. As we will see shortly, it actually makes the problem worse. In this section, we begin by discussing the noise problems caused by high performance/high frequency design and low-power design techniques. We then discuss analysis capabilities and optimization techniques to better design for on-chip noise management.

A. The Problem

We will deal primarily with power supply noise issues. However, we will briefly consider signal line noise.

1) Power Supply Noise: Power supply noise levels due to increased di/dt in high performance circuits are also becoming a growing concern. Given an initial arbitrary current pulse stimulus, V_{dd} and V_{ss} will try to oscillate 180° out of phase at their natural ringing frequency, $\omega_0 = 1/(\sqrt{LC})$, where C is the $V_{dd}-V_{ss}$ capacitance and L is the

total power supply loop inductance. See Fig. 13. However, if the stimulus is periodic and strong enough, the oscillator can be forced to oscillate at the frequency of the stimulus.

The peaks and valleys of $V_{dd} - V_{ss}$ can have performance and reliability implications. Timing slowdown may occur when $V_{dd} - V_{ss}$ is at a minimum. Timing skews may arise from some circuits speeding up at high $V_{dd} - V_{ss}$ and others, which switch at a different time in the clock period, may see a low $V_{dd} - V_{ss}$ and slow down. Hot electron operating limits or gate oxide stress limits may be exceeded during the $V_{dd} - V_{ss}$ peaks, leading to reliability failures [42].

The magnitude of the oscillations are a function of the power supply inductance, L_{VCC} and L_{VSS}; the $V_{dd} - V_{ss}$ die capacitance, C_{DIE}; the amount of capacitance which is switching, C_{SW}; power supply resistance; and the magnitude of the current demand from switching circuits, I_{avg}. It is expressed by the following equation (from [68]):

$$\Delta V = \frac{I_{\text{avg}}}{\sqrt{\frac{L_{Vdd-Vss}}{C_{SW}+C_{DIE}}}} \bullet \sin(\omega_o t). \tag{12}$$

In order to reduce the magnitude of the voltage oscillation peaks, the package is supplied with as many V_{dd} and V_{ss} pins as possible to reduce $L_{VCC-VSS}$. Decoupling capacitance is added to the die and on the package so that the highest frequency components of di/dt does not need to be supplied by a highly inductive off package path. Various architectural techniques to limit di/dt can be attempted, but the circuits cannot be slowed down, since performance will be affected.

Low power design introduces its own set of problems. An ideal low power design would result in low values of I_{avg} and di/dt. All units on the die would use small currents when active and very little current when inactive. Low power designs for microprocessors typically results in:

- reducing the maximum current peaks moderately,
- reducing the time spent at peak levels greatly,
- causing very low values of current when the device is carrying out "easy" tasks or is in standby mode.

That is, the current delivered tracks the MIPS required on a real time basis. This means that the di/dt is going to get worse. Power conscious designs in general will go through periods of inactivity, such as, standby mode or sleep mode, followed by intense periods of activity, followed again by periods of inactivity. Fig. 14 shows the differences in current demand between using and not using low power design techniques [93]. Notice how the current differences between peak operation and idle operation are larger in the design using power savings techniques.

One other difficulty caused by low power design is that current activity may be very localized in space. There is naturally occurring decoupling capacitance spread uniformly across the die in the form of N-well to substrate capacitance, nonswitching circuit blocks, and other capacitive parasitics. When the current activity is concentrated in one area of the chip, only the naturally occurring decoupling capacitance close to that circuit will function efficiently, it will not be able to "see" the capacitance at the far end of the chip

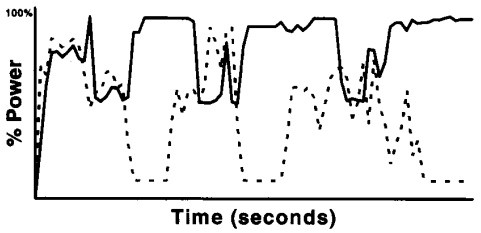


Fig. 14. Plot of power before and after using low power design techniques [93].

[88]. The charge on capacitors far away from the switching circuit will have a transit time determined by its *RC* time constant. Since the switching circuits are switching faster than this *RC* time constant, this far away charge can not arrive in time.

One final area of concern is in the use of low voltage to achieve low power. Although low power supply voltages help lower the power consumed, higher transistor counts and higher frequency rates usually keep I_{dd} relatively high. Lower V_{dd} usually means maintaining a lower absolute value of voltage noise. Considering $I * R$ drop across the die, power supply guardbands and tester guardbands, very little margin is left for the on die power supply oscillations. Since the di/dt usually remains fairly high, large values of decoupling capacitance are needed.

2) Signal Line Noise: With neighboring signal lines lying closer to each other on the die and in the package, the mutual coupling capacitance and inductance between them increases. This along with rapid switching on these lines can cause capacitive and inductive crosstalk leading to timing slowdowns and inadvertent logic transition faults.

B. Noise Analysis

In order to calculate the fluctuations of the power supply voltage levels on the die and its effect on noise, the following is needed:

- *RLC* network of V_{dd}/V_{ss} from the die to the printed circuit board power supply,
- switching activity of the device,
- passive $V_{dd} - V_{ss}$ capacitance of the device and the associated interconnect parasitics, and
- voltage noise specifications for the device under analysis.

Inductive effects caused by fast switching core logic circuits have not been a problem in the past for digital CMOS circuits. The only inductive effects the designer had to deal with were in the periphery (I/O circuits) and they were easy to deal with because they were very localized, had their own power supply, and the designer knew exactly when they switched [69].

Today, high di/dt problems are distributed all over the surface of the die. The designer cannot manually keep track of which circuits in the block switch with every input vector to the block. The on chip inductance is beginning

to become significant, further complicating matters. Clearly sophisticated CAD tools are needed to give the designer a chance at managing di/dt.

1) di/dt Extraction Capabilities: While evaluating di/dt noise, the damping effect of resistance or inductance in the power supply lines is needed to model realistic effects [60]. Existing circuit simulators can provide current waveforms depicting di/dt, but they do not include resistance or inductance in the power supply lines. While resistance extraction tools have been around for a while, parasitic inductance extraction tools are non-existent. Extracting the inductance requires, in addition to knowledge of the circuit block's operation, knowledge of the current flow direction, current flow in neighboring lines, and all of the return paths. To simulate this properly, one usually uses a finite element method [24]. Since each situation is unique, an inductance extraction is needed for each particular case of interest.

Once parasitic values are extracted, simulations can be run. Unfortunately, when a simulation is performed with V_{dd} and V_{ss} as variable circuit nodes, rather than fixed source voltages, it takes up to 25 times longer to run [60]. So now fast simulators are needed that give up some accuracy, but run faster, to handle the cases with parasitic resistance and inductance in the power supply loop.

Information on di/dt helps to estimate the on-chip die and local decoupling capacitance requirements. Determining this estimate at higher levels of abstraction is rapidly gaining importance. A major effort is under way to estimate I_{avg} at the architecture level, the RTL level, schematic level, and even the silicon testing level. Considering that I_{avg} is not easy to calculate, computing di/dt is even more difficult, since a profile over time is needed. One way of doing this at the RTL/schematic level is to generate a switching activity/toggle profile over time.

2) Effective Decoupling Capacitance Calculation: In addition to di/dt, other information is needed, such as the effective capacitance of the block under consideration and the percentage of the gates that are not switching. Basically, we need to know how much capacitance is being charged and discharged and how much is just sitting idle, able to serve as decoupling capacitance. Just as power estimates become more accurate as we move from the architectural level to the circuit level, and eventually the silicon level, we expect our decoupling capacitance estimates to improve in accuracy as well.

Just as the total power of a device is not equal the sum of the worst case power of all of the circuit blocks, so it is with decoupling capacitance. Tools that calculate the effective decoupling capacitance of a circuit block, must also be able to calculate the effective decoupling capacitance of neighboring blocks. A neighboring block may contribute more decoupling capacitance if it is idle than if it is active. The blocks further away from the switching block will have higher parasitic resistance and inductance in the power supplies connecting the blocks. Hence, the effective decoupling capacitance will be lower for these blocks. When all of this is added up, using circuit activity factors to temper the results, one should get a number for the decoupling capacitance which is lower than if neighboring blocks were ignored.

3) Cell Libraries for Decoupling Capacitance Design: Taking this a step further, a set of standard cells or layout design guidelines need to be accessible to circuit designers and the place and route tools. Standard cells for decoupling capacitors will insure that the intrinsic resistance of the capacitor is low enough to function properly. A set of guidelines for interleaving the power supply lines, V_{dd} and V_{ss}, needs to be incorporated into the local and global routing tools. This will insure that resistance, rather than the inductance, will be the dominant component of the impedance of the power supply lines.

The step from dealing with I_{avg} to dealing with $di(t)/dt$ is big and poses a real challenge for CAD tool developers.

4) Crosstalk Analysis Capabilities: Crosstalk induced noise is the voltage that develops upon an idle signal line when another signal line switches. Crosstalk consists of two components: inductive and capacitive. The noise developed is dependant upon $L_m * di/dt$ and $C_m * dv/dt$, where L_m is the mutual inductance and C_m is the mutual capacitance [84].

When crosstalk is dominated by capacitive coupling, as is the case for most CMOS IC's operating below 100 MHz, one just needs to know the various capacitance components of the active line and the passive line. This is basically an exercise in charge sharing and the various capacitance extraction tools available today can be used.

When the signal lines are closer together, and the switching is faster, then both inductive and capacitive effects should be included. This situation applies for package leads and traces, Printed Circuit board lines, and lines in high speed IC's. To analyze these situations properly, a finite element analysis or a similar analysis capability is needed. Many tools are available today. They take inputs such as line width and thickness, spacing to neighboring lines, location of return current planes, microstrip or stripline configuration, etc., and return L_m and C_m as well as other useful information [55].

C. Design for Noise Minimization

Preventing reliability and noise effects from impacting the performance of the chip requires contributions from the architecture, circuit design, physical layout, packaging, and the board design areas. Placing undo burden on any one of them will result in grotesque solutions, if any.

1) Architecture: Architectural techniques can be used to tame the sudden changes incurrent demand. When the device goes into a standby mode, the circuit blocks can be shut down sequentially over two or three clock periods. Similarly, when the device comes out of standby mode, one or two clock periods could be used to started drawing moderate amounts of current again.

When instructions are pipelined, there is advanced warning of when various circuit blocks will switch. A central di/dt bookkeeping unit on the die, could estimate in advance the current activities of the die. When the unit

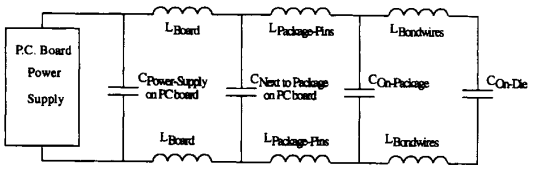


Fig. 15. Decoupling capacitance hierarchy.

predicts a large surge in current, it could partially power up the appropriate circuit blocks in advance to lower di/dt.

2) Logic Design and Physical Layout: To control power supply noise in addition to decoupling capacitance, the addition of power and ground pins to minimize the effect of V_{cc} and V_{ss} inductances has already been discussed. The on-die decoupling capacitance should be distributed across the die, and be placed very close to the circuits needing fast charge. If possible, the capacitors should have a clean V_{dd}/V_{ss} power supply connection. This is to ensure that the capacitor is able to maintain its own nominal value of V_{dd}/V_{ss}. Circuit block placement could also be optimized to share decoupling capacitance, as long as chip routing is not made worse.

Crosstalk problems are most easily solved using physical layout techniques. In Printed Circuit boards and in routable packages, good impedance control will minimize crosstalk. This is achieved by placing power planes on either side of the signal lines. The spacing from the lines to the planes should be closer than the line to line spacing. This can be achieved by either larger line to line spacing or moving the power planes closer to the signal planes.

On the die, this technique is not yet practical because the metal layers are usually used for signal routing and not power planes. One can also minimize the overlap of the signals by routing sensitive signals perpendicular to each other. When this is not possible, one can make the line to line spacing wider in long parallel runs.

One final technique is to route power and ground lines between the signal lines to make sure the EM fields couple to the power and ground lines and not the signal lines.

3) Die, Package, and Printed Circuit Board Decoupling Capacitance: Decoupling capacitance should be added on the die, in the package, and on the printed circuit board.

The decoupling capacitance is a good local source for fast charge because high frequency current can come from the local capacitor and not have to come from a more inductive path. The decoupling capacitors are functioning as high frequency filters [3].

In Fig. 15, we demonstrate how the high di/dt demands on the die can be met despite a relatively high inductive V_{dd}/V_{ss} connection to the printed circuit board power supply. The on die decoupling capacitors have to be high performance capacitors. This means that their own intrinsic inductance and resistance must be small as to not limit the demand for high frequency charge. They will have to supply charge for current demands in the tens of Amps per nanosecond. With this high frequency current shunted through the capacitor, the bondwires now only have to deal with one or two Amps per nanosecond.

Similarly, the on package decoupling capacitors can supply a few amperes per nanosecond, requiring only perhaps tenths of amperes per nanosecond to flow through the package pins. The process continues to the printed circuit board power supply where it is now only required to respond in the amperes per hundreds of microseconds range.

The cost per nanofarad of capacitance has to be managed carefully. Small amounts of capacitance are very cheap to build on the die or on the package, but large amounts can get very expensive. An optimum design uses just the right amount of decoupling at each stage to meet noise target goals. Capacitance on the die and in the package costs money, so we need to design with the right amount and not much more. If enough decoupling capacitance is not used, the on die voltage supply levels will vary too much and there will be yield loss. If design is done with excessive amounts, the die may need to grow or the package may again become so complex that there will be yield loss.

D. Prognosis

Managing di/dt is a complex task which will become more and more critical as technology advances because it is scaling in the wrong direction. Further development in di/dt management tools is needed to prevent di/dt from becoming a major factor limiting further advances in circuit integration.

VII. Summary and Recommendations

The need for lower power systems is being driven by many market segments as outlined in the Section I. There are several approaches to reducing power. The highest ROI 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 power savings at the different design phases are, in the best case, multiplicative. 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)* 0.85)* 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. In addition all the evidence points to the fact that the biggest power savings will be derived from transformations and decisions made early in the design process, i.e., from the high level design phase. However, high level design tools are not as mature as the logic and layout level tools. At all phases of the design process the tools can be classified as either:

- analysis tools,
- optimization tools, or
- libraries and design management tools.

Initially (in the absence of good optimization tools) the analysis tools will have the highest impact, as they will allow the designer to identify areas of the design that need to be optimized for power. In addition the analysis and power estimation tools will enjoy a large market because of their applicability to different design domains such as CPU, DSP, etc. To ensure designer acceptance and minimum learning time (on the tool), the analysis and optimization tools should, when possible support power capabilities in a value added way.

The tools' problem will have to be attacked on two fronts. On one front, existing and mature CAD packages (commercial and non commercial) can be quickly modified to provide basic support for analysis and optimization, e.g. logic simulation and logic synthesis.

On the second front, the research community should embark on longer term, basic and applied, pre-competitive research, to reduce power at all levels of the design, but with a special emphasis on higher level design as described in Sections II and III. Furthermore it is imperative that the research activity necessarily include work on *design and architectures*, as well as CAD tools and methodologies, because good high level CAD tools are domain specific. Domain specific design requires tools targeted at specific architectures such as those used in CPU and DSP designs.

The successful development of new power conscious tools and methodologies requires a clear and measurable goal. One realistic and, yet challenging, target is to reduce power by 3× in three years and 5× in 5 years through design and tool development. That is, any power reduction through process scaling or voltage scaling should be above an beyond the 3× and 5× goals. To achieve these goals, it is necessary that a large number of the tools and techniques, mentioned in this paper, should become widely available (by preference through commercial channels).

To be more specific, in the short term (one year time frame), the following tools are required: logic level power estimation tools (circuit level power estimation is currently available), RTL power estimation, low power logic synthesis, reliability verification and packaging modeling and standard cell libraries characterized for Power, Performance, and Area. In the longer term (three-year frame), a low power design environment should contain the following components: behavioral and architectural level power estimation, low power behavioral/architectural synthesis and tools to optimize the micro-architectural description, based on techniques described in Section II.

Achieving these challenging goals will require an active role of there search community, in both the development and proliferation of low power circuit, logic and architectural techniques.

ACKNOWLEDGMENT

The authors gratefully acknowledge the participation of C. Deng of EPIC Design Technology Inc. for his contribution on PowerMill and circuit level power analysis.

REFERENCES

[1] A. Aho and J. Ullman, *Principles of Compiler Design.* Reading, MA: Addison-Wesley, 1977.
[2] M. Alidina, J. Monteiro, S. Devadas, A. Ghosh, and M. Papaefthymiou, "Precomputation-based sequential logic optimization for low power," in *Proc. 1994 Int. Workshop on Low Power Design,* pp. 5762, Apr. 1994.
[3] H. Bakoglu, *Circuits, Interconnections and Packaging for VLSI.* Reading, MA: Addison-Wesley, 1990.
[4] E. Barke, "Resistance calculation from mask artwork data by finite element method," P. McCormick, Ed. *Proc. 22nd DAC* 1985, pp. 305-311.
[5] Y. Be'ery, S. Berger, and B. Ovadia, "An application specific DSP for portable applications," *Proc. VLSI Signal Proc. Workshop,* Veldhoven, The Netherlands, 1993, pp. 48-56.
[6] L. Benini, M. Favalli, and B. Ricco, "Analysis of hazard contribution to power dissipation in CMOS IC's," in *Proc. 1994 Int. Workshop on Low Power Design*, pp. 27–32, Apr. 1994.
[7] M. Berkelaar and J. Jess, "Gate sizing in MOS digital circuits with linear programming," in *Proc. Europe. Design Autom. Conf.*, pp. 217–221, 1990.
[8] R. K. Brayton and C. McMullen, "The decomposition and facorization of Boolean expressions," in *Proc. Int. Symp. on Circ. and Syst.*, pp. 49–54, Rome, May 1982.
[9] R. Bryant, "Graph-based algorithms for Boolean function manipulation," *IEEE Trans. Compu.*, vol. C-35, pp. 677–691, Aug. 1986.
[10] J. Buck, S. Ha, E. Lee, and D. Messerschmitt, "Ptolemy: A framework for simulating and prototyping heterogeneous systems," *Int. J. Compu. Simulation,* special issue on Simulation and Software Development.
[11] J. Bunda, D. Fussel, and W. Athas, "Evaluating power implications of CMOS microprocessors," *Proc. Int. Workshop on Low Power Design,* Napa Valley, CA, Apr. 1994.
[12] A. R. Burch, F. Najm, P. Yang, and D. Hocevar, "Pattern independent current estimation for reliability analysis of CMOS circuits," in *Proc. 25th Design Autom. Conf.*, pp. 294–299, June 1988.
[13] R. Burch, F. N. Najm, P. Yang, and T. Trick, "A Monte Carlo approach for power estimation," *IEEE Trans. Very Large-Scale Integ. Syst.*, vol. 1, pp. 63–71, Mar. 1993.
[14] T. Callaway and E. Swartzlander, "Optimizing arithmitic elements for signal processing," *Proc. VLSI Signal Proc. Workshop,* IEEE Press, pp. 91–100, 1992.
[15] M. Carazao, M. Khalaf, L. Guerra, M. Potkonjak, and J. Rabaey, "Instruction set mapping for performance optimization," *Proc. ICCAD 1993,* pp. 518–521, Santa Clara, CA, Nov. 1993.
[16] B. S. Carlson and C-Y. R. Chen, "Performance enhancement of CMOS VLSI circuits by transistor reordering," in *Proc. 30th Design Autom. Conf.*, pp. 361–366, June 1993.
[17] S. Chakravarty, "On the complexity of using BDD's for the synthesis and analysis of Boolean circuits," in *Proc. 27th Annu. Allerton Conf. on Commun., Contr. and Computing,* pp. 730-739, 1989.
[18] A. Chandrakasan, S. Sheng, and R. W. Brodersen, "Low-power CMOS design," *J. Solid-State Circ.*, pp. 472-484, Apr. 1992.
[19] A. Chandrakasan, M. Potkonjak, J. Rabaey, and R. W. Brodersen, "HYPER-LP: A system for power minimization using architectural transformations," *Proc. ICCAD* 1992.
[20] A. Chandrakasan, M. Potkonjak, R. Mehru, J. Rabaey, and R. W. Brodersen, "Optimizing power using transformations," to be published.
[21] A. Chandrakasan, R. Allmon, A. Stratakos, and R. Brodersen, "Design of portable systems," *Proc. CICC Conf.*, May 1994.
[22] K-Y. Chao and D. F. Wong, "Low power considerations in floorplan design," in *Proc. 1994 Int. Workshop on Low Power Design,* pp. 45–50, Apr. 1994.
[23] K. Chaudhary and M. Pedram, "A near-optimal algorithm for technology mapping minimizing area under delay constraints," in *Proc. 29th Design Autom. Conf.*, June 1992.
[24] T. Y. Chou, J. Cosentino, and Z. Cendes, "High-speed interconnect modeling and high-accuracy simulation using SPICE and finite element methods," Proc. 30th ACM/IEEE DAC, June 1993.
[25] L. O. Chua and P. M. Lin, *Computer-Aided Analysis Algorithms and Computational Techniques.* Englewood Cliffs, NJ: Prentice Hall.

[26] J. Cong, C-K. Koh, and K-S. Leung, "Wire sizing with driver sizing for performance and power optimization," in *Proc. 1994 Int. Workshop on Low Power Design,* pp. 81–86, Apr. 1994.
[27] R. L. M. Dang and N. Shigyo, "Coupling capacitances for2-dimensional wires," *IEEE Electro* XXX
[28] "Dataquest: The Green PC Revolution," *Focus Rep.*, Oct. 11, 1993.
[29] A. C. Deng, Y-C Shiau, and K-H. Loh, "Time domain current waveform simulation of CMOS circuits," in *IEEE Trans. Comp.-Aided Design,* pp. 208–211, Nov. 1988.
[30] C. Deng, "Power analysis for CMOS/BiCMOS circuits," in *Proc. 1994 Int. Workshop on Low Power Design,* pp. 3–8, Apr. 1994.
[31] P. Duncan, S. Swamy, and R. Jain, "Low-power DSP circuit design using retimed maximally parallel architectures," *Proc. 1st Symp. on Integ. Syst.,* pp. 266–275, Mar. 1993.
[32] S. Ercolani, M. Favalli, M. Damiani, P. Olivo, and B. Ricco, "Estimate of signal probability in combinational logic networks," in *1st Europe. Test Conf.,* pp. 132–138, 1989.
[33] C. M. Fiduccia and R. M. Mattheyses, "A linear-time heuristic for improving network partitions," in *Proc. 19th Design Autom. Conf.,* pp. 175–181, 1982.
[34] J. P. Fishburn and A. E. Dunlop, "TILOS: A posynomial programming approach to transistor sizing," in *Proc. IEEE Int. Conf. on Comp.-Aided Design,* pp. 326–328, Nov. 1985.
[35] T. A. Fjeldly and M. Shur, "Threshold voltage modeling and the subthreshold regime of operation of short-channel MOSFET's," *IEEE Trans. Electron Devices,* vol. 40, pp. 137–145, Jan. 1993.
[36] 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 *Pro. 1994 Int. Workshop on Low Power Design,* pp. 215–218, Apr. 1994.
[37] W. Geurts, F. Catthoor, and H. De Man, "Heuristic techniques for the synthesis of complex functional units," *Proc. 4th ACM/IEEE EDAC Conf.,* Paris, pp. 552–556, Feb. 1993.
[38] A. A. Ghosh, S. Devadas, K. Keutzer, and J. White, "Estimation of average switching activity in combinational and sequential circuits," in *Proc. 29th Design Automation Conf.,* pp. 253-259, June 1992.
[39] L. H. Goldstein, "Controllability/observability of digital circuits," *IEEE Trans. Circ. and Syst.,* vol. 26, pp. 685–693, Sept. 1979.
[40] R. Hartley and A. Casavant, "Tree-height minimization in pipelined architectures," IEEE Trans. Comp. Aided Design, vol. 8, pp. 112–115, 1989.
[41] R. I. Hartley, "Optimization of canonic signed digit multipliers for filter design," *Proc. IEEE Int. Symp. Syst.,* Singapore, June 1991.
[42] C. Hu, S. C. Tam, F. C. Hsu, P. K. Ko, T. Y. Chan, and K. W. Terrill, "Hot-electron induced MOSFET degradation—Model, monitor, and improvement," *IEEE Trans. Electron Devices,* vol. ED-32, p. 375, 1985.
[43] D. A. Huffman, "A method for the construction of minimum redundancy codes," in *Proc. IRE,* vol. 40, pp. 1098–1101, Sept. 1952.
[44] S. Iman and M. Pedram, "Multi-level network optimization for low power," in *Proc. IEEE Int. Conf. on Comp. Aided Design,* Nov. 1994.
[45] R. Jain *et al*, "Module Selection for Pipelined Synthesis," *Proc. Design Automation,* Anaheim, pp. 542-547, 1988.
[46] J. M. Janssen, F. Catthoor, H. De Man, "A specification invariant technique for operation cost minimisation in flow-graphs," *Proc. 7th Int. Workshop on High-Level,* Niagara-on-the-Lake, Canada, May 1994.
[47] S. M. Kang, "Accurate simulation of power dissipation in VLSI circuits," it IEEE J. Solid-State Circ., vol. 21, pp. 889–891, Oct. 1986.
[48] K. Keutzer, "DAGON: Technology mapping and local optimization," in *Proc. Design Automation Conf.,* pp. 341–347, June 1987.
[49] F. J. Kurdahi and A. C. Parker, "Techniques and area estimation of VLSI layouts," *IEEE Trans. Comp.-Aided Design,* vol. 8, pp. xxx, Jan. 19xx.
[50] P. Landman and J. Rabaey, "Power estimation for high level synthesis," *Proc. EDAC-EUROASIC '93,* Paris, France, pp. 361–366, Feb. 1993.
[51] ——, "Black-box capacitance models for architectural power analysis," *Int. Workshop Low Power,* Napa Valley, CA, Apr. 1994.
[52] D. Lanneer, M. Cornero, G. Goossens, and H. De Man, "Data Routing: a paradigm for efficient data path synthesis and codegeneration," *Proc. 7th ACM IEEE Int. Symp. on High Level Synthesis,* Niagara-on-the-Lake, Canada, May 1994.
[53] L. Larmore and D. S. Hirschberg, "A fast algorithm for optimal length-limited Huffman codes," *J. Assoc. for Computing Mach.,* vol. 37, no. 3, pp. 464–473, 1990.
[54] C. Leiserson, F. Rose, and J. Saxe, "Optimizing synchronous circuits by retiming," in *Proc. Third Conf. VLSI,* 1983, pp. 23–26.
[55] J. C. Liao, O. A. Palusinski *et al.*, "University of Arizona Capacitance Calculator," *Users Guide,* June 1986.
[56] D. Lidsky and J. Rabaey, "Low power design of memory intensive applications—Case study: vector quantization," *Proc. 1994 Symp. on Low Power Electron.,* San Diego, CA, 1994.
[57] B. Lin and H. de Man, "Low-power driven technology mapping under timing constraints," in *Int. Workshop on Logic Synthesis,* pp. 9a–19a.16, Apr. 1993.
[58] B. Lin and A. R. Newton, "Synthesis of multiple-level logic from symbolic high-level description languages," in *IFIP Int. Conf. on Very Large Scale Integration,* pages 187-196, Aug. 1989.
[59] P. Lippens, J. van Meerbergen, W. Verhaegh, and A. van der Werf, "Allocation of multiport memories for hierarchical datastreams," *Proc. IEEE Int. Conf. Comp. Aided Design,* Clara CA, Nov. 1993.
[60] F. Maloberti and G. Torelli, "On the design of CMOS digital output drivers with controlled dI/dt," *IEEE Int. Symp. on Circ. and Syst.,* vol. 4, p. 2236, 1991.
[61] R. Marculescu, D. Marculescu, and M. Pedram, "Logic level power estimation considering spatiotemporal correlations," in *Proc. IEEE Int. Conf. on Computer Aided Design,* Nov. 1994.
[62] A. Masaki, "Deep-submicron CMOS warms up to high-speed logic," *Circuits and Devices,* Nov. 1992.
[63] A. Masaki and T. Chiba, "Design aspects of VLSI for computer logic," *IEEE Trans. Electron Devices,* Apr. 1982.
[64] R. Mehra and J. Rabaey, "High level power estimation and exploration," *Proc. Int. Low Power Workshop,* Napa Valley, CA, Apr. 1994.
[65] J. Monteiro, S. Devadas, and A. Ghosh, "Retiming sequential circuits for low power," in *Proc. IEEE Int. Conf. on Comp. Aided Design,* pp. 398–402, Nov. 1993.
[66] ——, "Estimation of switching activity in sequential logic circuits with applications to synthesis for low power," in *Proc. 31st Design Autom. Conf.,* pp. xx, June 1994.
[67] 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 *Proc. 1994 Int. Workshop on Low Power Design,* pp. 117–122, Apr. 1994.
[68] T. Mozdzen and B. Bhattacharyya, "Electrical design rules for using SiO_2 gate oxide as decoupling capacitors," *Intel Internal Rep.,* Nov. 1992.
[69] T. Mozdzen, "The effect of on die $Vcc/VssI * R$ voltage drop upon average and di/dt," *Intel Internal Rep.,* May 1994.
[70] C. Nagendra *et al.*, "A comparison of power-delay characteristics of CMOS adders," *Proc. Int. Workshop on Low Power Design,* Napa Valley, CA, Apr. 1994.
[71] F. Najm and R. Burch, "CREST—A current estimator for CMOS circuits," *Proc. 25th DAC,* June 1988, pp 204–207.
[72] F. Najm, "Transition density, a stochastic measure of activity in digital circuits," in *Proc. 28th Design Autom. Conf.,* pp. 644–649, June 1991.
[73] F. N. Najm, R. Burch, P. Yang, and I. Hajj, "Probabilistic simulation for reliability analysis of CMOS VLSI circuits," in *IEEE Trans. Computer-Aided Design of Integ. Circ. and Syst.,* vol. 9, pp. 439–450, Apr. 1990.
[74] S. Y. Oh, K. J. Chang, N. Chang, and K. Lee, "Interconnect modeling and design in high-speed VLSI/ULSI systems," *Proc. 29th ACM/IEEE DAC,* June 1992, pp. 184-189.
[75] K. P. Parker and J. McCluskey, "Probabilistic treatment of general combinational networks," *IEEE Trans. Computers,* vol. C-24, pp. 668–670, June 1975.
[76] M. Pedram and B. Preas, "Accurate prediction of physical design characteristic for randoml ogic," *1989 IEEE ICCD Conf.,* Boston, pp. 100-108, 1989.

[77] L. T. Pillage and R. A. Rohrer, "Asymptotic waveform evaluation for timing analysis," *IEEE Trans. Comp.-Aided Design*, vol. 9, pp. 352–366, Apr. 1990.
[78] M. Potkonjak and J. Rabaey, "Optimizing resource utilization using transformations," *IEEE Trans. Comp. Aided Design,* vol. 13, pp. 277-292, Mar. 1994.
[79] S. R. Powell and P. M. Chau, "Estimating power dissipation of VLSI signal processing chips: The PFA technique," *VLSI Signal Processing IV,* pp. 250-259, 1990.
[80] J. Rabaey, *Digital Integrated Circuits: A Design Perspective.* Englewood Cliffs, NJ: Prentice Hall, 1995, currently available as EE141 Course Reader, U.C. Berkeley.
[81] ——, "Low power design methodologies," in *Circuits and Systems Tutorials.* Oxford, UK: LTP Electronics Ltd, pp. 373–386, June 1994.
[82] J. Rabaey, C. Chu, P. Hoang, and M. Potkonjak, "Fast prototyping of data path intensive architectures," *IEEE Design and Test* vol. 8, pp. 40-51, 1991.
[83] J. Rabaey and M. Potkonjak, "Complexity estimation for real time application specific circuits," *Proc. IEEE ESSCIRC Conf.,* Milan, Italy, 1991.
[84] S. Rajaram, "Electrical characterization of interconnections," *The Electrical Engineering Handbook,* CRC Press, p. 499, 1993.
[85] S. Rajgopal and G. Mehta, "Expreriences with simulation-based schematic level current estimation," in *Proc. 1994 Int. Workshop on Low Power Design,* pp. 9–14, Apr. 1994.
[86] J. Rajski and J. Vasudevamurthy, "The testability-preserving concurrent decomposition and factorization of Boolean expressions," IEEE Trans. Comp.-Aided Design of Integ. Circ. and Syst., vol. 11, pp. 778–793, June 1993.
[87] D. Rao and F. Kurdahi, "Partitioning by regularity extraction," *29th ACM/DAC'92,* pp. 235–238, 1992.
[88] N. Raver, "Modeling CMOS local drop-point voltage dependencies," *IBM Res. Rep.*, vol. RC 17269, No. 76388, Oct. 1991.
[89] K. Roy and S. C. Prasad, "Circuit activity based logic synthesis for low power reliable operations," *IEEE Trans. Very Large-Scale Integ. Syst.*, vol. 1, pp. 503–513, Dec. 1993.
[90] T. Sakurai and K. Tamaru, "Simple formulas for 2 and 3-dimensional capacitances," *IEEE Trans. Electron Devices,* vol. ED-30, pp. 183-185, Feb. 1983.
[91] H. Savoj, R. K. Brayton, and H. J. Touati, "Extracting local don't cares for network optimization," in *Proc. IEEE Int. Conf. on Comp. Aided Design,* pp. 514–517, Nov. 1991.
[92] P. Schneider and U. Schlichtmann, "Decomposition of Boolean functions for low power based on a new power estimation technique," in *Proc. 1994 Int. Workshop on Low Power Design,* pp. 123–128, 1994.
[93] J. Schutz, "A 3.3V 0.6um BiCMOS Superscaler Microprocessor,"*ISSCC Dig. Tech. Papers,* pp. 202–203, Feb. 1994.
[94] W. S. Scott and J. K. Ousterhout, "Magic's circuit extractor," *Proc. 22nd DAC,* 1985, pp 286-292.
[95] N. Seghal, C. Chen, and J. Acken, "An object-oriented cell library manager," *ICCAD,* Nov. 1994.
[96] S. C. Seth, L. Pan, and V. Agrawal, "PREDICT—Probabilistic Estimation of Digital Circuit Testability," in *Proc. Fault Tolerant Comp. Symp.,* pp. 220–225, June 1985.
[97] A. A. Shen, A. Ghosh, S. Devadas, and K. Keutzer, "On average power dissipation and random pattern testability of CMOS combinational logic networks, in *Proc. IEEE Int. Conf. on Computer Aided Design,*Nov. 1992.
[98] S. Sheng, A. Chandrakasan, and R. Brodersen, "A portable multimedia terminal," *IEEE Commun. Magazine,* pp. 64–75, Dec. 1992.
[99] C. S. Shung *et al.*, "An integrated CAD system for algorithm-specific IC design," *IEEE Trans. CAD Integ. Circ. and Syst.,* Apr. 1991.
[100] D. Stark and M. Horowitz, "REDS: Resistance extractor for digital simulation," *Proc. 24th DAC,* 1987, pp 570-573.
[101] A. Stratakos, R. Brodersen, and S. Sanders, "High low voltage DC-DC conversion for portable applications," *Proc. Int. Low Power Workshop,* Napa Valley, CA, Apr. 1994.
[102] S. L. Su, V. B. Rao and T. N. Trick, "HPEX: A hierarchical parasitic circuit extractor," *Proc. 24th DAC,* 1987, pp 566-569.
[103] C. Su, C. Tsui, and A. Despain, "Low power architecture design and compilation techniques for high performance processors," Proc.
[104] C. Svensson and D. Liu, "A power estimation tool and prospects of power savings in CMOS VLSI chips," *Proc. Int. Workshop on Low Power Design,* Napa Valley, CA, Apr. 1994.
[105] V. Tiwari, P. Ashar, and S. Malik, "Technology mapping for low power," in *Proc. 30th Design Auto. Conf.,* pp. 74–79, June 1993.
[106] E. Tsern, T. Meng, and A. Hung, "Video compression for portable communication using pyramid vector quantization of subband coefficients," *Proc. IEEE Workshop on VLSI Signal Processing,* Veldhoven, pp. 444-452, Oct. 1993.
[107] C-Y. Tsui, M. Pedram, C-A. Chen, and A. M. Despain, "Low power state assignment targeting two- and multi-level logic implementations," in *Proc. IEEE Int. Conf. on Comp. Aided Design,* Nov. 1994.
[108] C. Y. Tsui, M. Pedram, and A. Despain, "Efficient estimation of dynamic power dissipation under a real delay model," in *Proc. IEEE Int. Conf. on Comp. Aided Design,* pp. 224–228, Nov. 1993.
[109] ——, "Technology decomposition and mapping targeting low power dissipation," in *Proc. 30th Design Automation Conf.,* pp. 68–73, June 1993.
[110] ——, "Power efficient technology decomposition and mapping under an extended power consumption model," *IEEE Trans. Comp.-Aided Design of Integrated Circuits and Syst.* vol. 13, pp. xx, Sept. 1994.
[111] ——, "Exact and approximate methods for calculating signal and transition probabilities in FSM's," in *Proc. 31st Design Automation Conf.* June 1994.
[112] A. Tyagi, "Hercules: A power analyzer of MOS VLSI circuits," in *Proc. IEEE Int. Conf. on Computer Aided Design,* pp. 530–533, Nov. 1987.
[113] J. Ullman, *Computational Aspects of VLSI.* Rockville, MD: Computer Science Press, 1984.
[114] H. Vaishnav and M. Pedram, "Pcube: A performance driven placement algorithm for low power designs," in *Proc. Europe. Design Autom. Conf.,* Sept. 1993.
[115] N. P. Van Der Meijs *et al*, "An efficient finite element method for submicron IC capacitance extraction," *Proc. 26th DAC,* pp. 678–681, June 1989
[116] P. Van Oostende, P. Six, J. Vandewalle, and H. DeMan, "Estimation of typical power of synchronous CMOS circuits using a hierarchy of simulators," *IEEE J. Solid-State Circ.*, vol. 28, pp. 26–39, Jan. 1993.
[117] H. J. M. Veendrick, "Short-circuit dissipation of static CMOS circuitry and its impact on the design of buffer circuits," in *IEEE J. Solid-State Circ.,* vol. 19, pp 468–473, Aug. 1984.
[118] H. Vermassen, "Mathematical models for the complexity of VLSI," M.S. thesis, Katholieke Universiteit, Leuven, Belgium, 1986.
[119] T. Villa and A. Sangiovanni-Vincentelli, "NOVA: State assignment of finite state machines for optimal two-level logic implementations," *IEEE Trans. Comp.-Aided Design of Integ. Circ. and Syst.*, vol. 9, pp. 905–924, Sept. 1990.
[120] M. Van Swaaij, F. Franssen, F. Catthoor, and H. De Man, "Automating high-level control flow transformations for DSP memory management," *Proc. IEEE Workshop on VLSI Signal Processing,* Napa Valley, CA, Oct. 1992. Also in *VLSI Signal Processing,* V. K. Yao, R. Jain, W. Przytula, and J. Rabaey, Eds. New York: IEEE Press, pp. 397-406, 1992.
[121] R.A. Walker and R. Camposano, *A Survey of High-Level Synthesis Systems.* Boston: Kluwer, 1992.
[122] S. Wu, "A generic description of hardware libraries," M.S. thesis, Univ. Calif., Berkeley, May 1994.
[123] S. Wuytack, F. Catthoor, F. Franssen, L. Nachtergaele, and H. DeMan, "Global communication and memory optimizing transformations for low power systems," *Int. Workshop Power Design,* Napa Valley, CA, Apr. 1994.
[124] S. Yang, "Logic synthesis and optimization benchmarks user guide," Microelectronics Center of North Carolina, Research Triangle Park, NC, 1991.
[125] T. Yoshitome, "Hierarchical Analyzer for VLSI power supply networks based on a new reduction method," *Proc. ICCAD,* pp. 298–301, 1991.
[126] 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 *Proc. 1994 Int. Workshop on Low Power Design,* pp. 87–92, Apr. 1994.

Deo Singh (Member, IEEE) received Diplomas in electrical engineering from Middlesex Polytechnic and Twickenham Polytechnics, in 1976 and 1977, and the M.Sc.-CAD degree from Kingston Polytechnic in 1981.

In 1977 he joined GEC Semiconductors, Melbourne, FL, where he was responsible for development and support of CAD tools. In 1981 he joined Harris Semiconductors in Melbourne, FL, and worked on logic simulation and cellular/hierarchical design rule checking. From 1983 to 1987 he was Harris' liaison to the MCC CAD program in Austin, TX. From 1988 to 1991 he managed a European commission multinational R&D CAD tools project (SPRITE) in Belgium. The project developed two prototype high-level synthesis systems targeted at application-specific architectural compilation for video, HDTV, and digital communications. He is presently Principal Engineer and Manager of the Low Power Design Technologies (LPDT) group in Intel Corporation, Santa Clara, CA, where he is responsible for reducing power in Intel microprocessors through the development and proliferation of low-power design methodologies (Tools, Methodologies, and Design Techniques).

Jan M. Rabaey (Fellow, IEEE) received the E.E. and Ph.D. degrees in applied sciences from the Katholieke Universiteit, Leuven, Belgium, in 1978 and 1983, respectively.

From 1983 to 1985 he was a Visiting Research Engineer with the University of California at Berkeley. From 1985 to 1987 he directed the Architectural and Algorithmic Strategies Group of the Design Methodologies for the VLSI System Division at IMEC, Belgium. Since 1987 he has been with the University of California at Berkeley's Electrical Engineering and Computer Science Department, where he is now a Professor. He was a Visiting Professor at the University of Pavia, Italy, in 1991. His current research interests include computer-aided analysis and automated design of digital signal processing circuits, and architectural synthesis. He has authored or co-authored numerous technical papers in the area of signal processing and design automation. He was an Associate Editor of the *IEEE Journal of Solid-State Circuits.*

Dr. Rabaey received a number of awards, including the 1985 IEEE Transactions on Computer Aided Design Best Paper Award from the Circuits and Systems Society, the 1989 Presidential Young Investigator Award, and the 1994 Signal Processing Society Senior Award. He is the current Chair of the VLSI Signal Processing Technical Committee of the Signal Processing Society. He is also a member of the Design Automation Conference's Executive Committee.

Massoud Pedram (Member, IEEE) received the B.S. degree in electrical engineering from the California Institute of Technology in 1986 and the M.S. and Ph.D. degrees in electrical engineering and computer sciences from the University of California at Berkeley.

He is Assistant Professor of Electrical Engineering–Systems at the University of Southern California. His research interests and contributions have been in logic synthesis, physical design, and CAS for low power.

Dr. Pedram received the Best Paper Award in the CAD track in the IEEE International Conference on Computer Design in 1989, and the National Science Foundation's Research Initiation and Young Investigator Awards in 1992 and 1994, respectively. He has served on the program committees for several conferences and workshops (including the Design Automation Conference) and is the cofounder of and General Chair of the International Workshop on Low Power Design.

Francky Catthoor (Member, IEEE) received the engineering degree and the Ph.D. degree in electrical engineering from the Katholieke Universiteit Leuven, Belgium, in 1982 and 1987, respectively.

From 1983 to 1987 he was a Researcher in the VLSI Design Methodologies for Signal Processing at IMEC, Heverlee, Belgium. Since 1987 he has headed research domains in the area of architectural and synthesis methodologies, within the VSDM division at IMEC. His current research activities mainly belong to the field of application-specific architecture design methods and system-level transformations intended for real-time signal processing algorithms in image, video, and telecom applications.

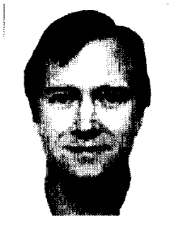
Dr. Catthoor received the Young Scientist Award from the Marconi International Fellowship in 1986.

Suresh Rajgopal (Member, IEEE) received the B.S. degree in computer science and engineering from the Indian Institute of Technology, Karagpur, in 1985. He received the M.S. in computer science from the University of Tennessee, Knoxville, in 1987 and the Ph.D. in computer science from the University of North Carolina, Chapel Hill, in 1992.

He is currently a Senior CAD Engineer in the Low Power Design Technology Group at Intel Corporation, Santa Clara, CA.

Naresh Sehgal (Member, IEEE) received the B.E. degree (hons) from Panjab University, India, in 1984, and the M.S. and Ph.D. degrees in computer engineering from Syracuse University in 1988 and 1994, respectively.

He has been with Intel's Design Technology division since 1988, and currently manages a full-chip assembly place and route tool team. His research interests are in object-oriented databases and layout algorithms.



Thomas J. Mozdzen (Member, IEEE) received the B.S. in physics and the M.S.E.E. from the University of Illinois at Urbana in 1978 and 1980, respectively. He received the M.S. in physics from the University of Texas at Dallas in 1985.

He joined Mostek in 1979 as a Process Characterization Engineer and worked on thin oxide integrity, hot electron effects, and the modeling of failure mechanisms. He joined the DRAM design group in 1984 and codesigned a 256K video DRAM. During 1986–1988 he was with Siemens, Münich, Germany, where he worked on the design of a 4–MEG DRAM and a digital television picture processor. In 1988 he joined the ASIC design group at Intel, where he was the design architect of the Intel ASIC 0.1 μm Standard Cell Library. His current area of focus is on di/dt management across the die, package, and board.

Mr. Mozdzen is a member of the American Physical Society.