Issues and Directions in Low Power Design Tools: An Industrial Perspective

Jerry Frenkil
Sente, Inc
3 Summer St, Chelmsford, MA 01824
jfrenkil@senteinc.com

Abstract — Designing for low power has become increasingly important in a wide variety of applications, ranging from wireless communications where battery life is the critical factor to high performance computing where reliability and cooling issues have become first order concerns. Numerous tools have emerged in recent years to help designers address these issues, yet some aspects of the power problem are still inadequately addressed. This paper will survey existing commercial tools used to estimate and optimize power, analyze their strengths and weaknesses, and describe a variety of open issues.

I. INTRODUCTION

The issue of power consumption in integrated circuit design continues to grow. Once a concern only to designers of very low power electronics, power consumption now is an issue for designers of all types of integrated circuits. Especially feeling the pain are designers of microprocessors, multi-media and digital signal processors, high-speed networking devices, and of course battery powered and wireless electronics.

Market forces are demanding lower power for many reasons, some obvious and others subtle. Battery life is a well understood advantage of lower consumption, however reliability, performance, cost, and time to market all benefit from lower power. At the same time however, technological forces are pushing power consumption to higher and higher levels.

The initial response to the problem was to lower the power supply voltage, and this trend continues. However, it has become clear that while this is perhaps a necessary part of the solution, it is by no means the full solution as power consumption has continued to rise even while power supply voltages have been dropping. Furthermore, issues which had previously been considered minor design issues have become first order concerns in the face of extremely low voltage or extremely high power devices. Examples of the former are standby leakage currents, while examples of the latter are accurate power bus sizing.

Accordingly, integrated circuit designers and EDA tools developers have in recent years been focusing on design tools and methodologies as a more leveraged approach to reducing power consumption. Section II of this paper will begin by briefly reviewing the sources and types of power consumption in integrated circuits. Section III will continue with a survey of the available EDA tools for low power design, focusing on strengths and weaknesses, while section IV will focus on the issues faced by designers using these tools. Finally, section V will conclude with a prognostication of where tools are methodologies for low power design are headed.

II. POWER SOURCES AND TYPES

A variety of tools are available today for use in low power design. In order to put each tool in the proper perspective it is helpful to briefly review some power fundamentals.

A. Power Sources

An IC's total power consumption is comprised of two different types, static and dynamic. The primary distinguishing factor between the two is that dynamic power is frequency dependent while static is not. Static power is defined to be the product of the power supply voltage and a static current, which itself has two components: leakage current and through current. Leakage currents, which occur in all MOS devices, are due to sub-threshold transistor operation and reverse bias diode leakage. Through currents also occur in normal operation but are due to transistors being continuously operated in their saturation (always on) regions and arise out of design decisions to employ analog techniques or resistive pullups. The magnitude of through currents is usually in the micro-amp to milli-amp range. Leakage currents, by comparison, are parasitic effects which are orders of magnitude smaller and are usually ignored, except in those devices that have extremely long standby times or operate at extremely low voltages.

Dynamic power also has a couple of components, crowbar power and (capacitive) load power. Crowbar power represents the crowbar currents that flow from power to ground when the transistor stacks switch state, while the capacitive load power represents the currents required to charge the fanout load capacitance. In an ASIC, or library based, design methodology these dynamic currents are combined into cell and load currents, where the cell currents include the crowbar currents as well as the currents required to charge the parasitic capacitances internal to the cell, while the load currents represent the current required to charge the load capacitances external to the cell.

B. Power Types

In addition to understanding the sources of power consumption, it is also essential to understand the measurement types of power consumption, as each type is
used for a different purpose. The different types are time-averaged, maximum, RMS, and (instantaneous) peak.

Time-averaged power, also known simply as average power, represents a single value for power consumption, obtained by integrating power on a cycle by cycle basis over a number of clock cycles. Average power is used to calculate battery life and junction temperature, and is thus the most widely used type of power calculation. Maximum power figures are used to size power busses to satisfy IR drop requirements to meet worst case noise margins.

RMS power is also used for sizing, however these values are used to meet electromigration guidelines. Electromigration rules are usually sensitive to both steady state and periodic current flow, so RMS or some form of modified RMS calculations are used to verify that the current density rules are not violated.

Lastly, instantaneous peak currents, sometimes referred to as \( \frac{d}{dt} \) currents, are used to determine the allowable parasitic inductance presented to the device. These values and calculations are essential for minimizing ground bounce effects, both on the chip's IOs as well as on the internal logic.

With this understanding of the physical sources and effects of power, along with a classification of the measurement types, the various design tools for low power design can be evaluated for their efficaciousness in low power design.

### III. LOW POWER DESIGN TOOLS

Numerous EDA tools exist for use in analyzing and minimizing power consumption. Some are targeted specifically for use in the power domain, while others are more general purpose and may typically be used for other tasks, such as timing analysis. In this survey of currently available commercial EDA tools, each tool is classified according to the abstraction layer upon which it operates. Four layers are chosen for classification: transistor (or switch), logic (or gate), architecture (or RTL), and behavior (or system). Note that due to space limitations only the primary examples of tools at each abstraction layer are described.

#### A. Transistor Level Tools

Transistor level analysis tools are generally very accurate and mature, and fall into two general categories: the circuit analysis tools like SPICE and its many variants and the switch analysis tools like PowerMill (Epic), ADM (Avanti/Anagram) and Lsim Power Analyst (Mentor).

The primary advantages of transistor level tools are accuracy, within a few per-cent of silicon, and well-accepted abstraction - most IC designers understand transistor level analysis and rely upon it. However, these tools have significant issues in their applicability to low power design: capacity and run time characteristics limit their use to small circuits, or very limited depths of simulation vectors for larger circuits. For example, circuit analysis tools are usually limited to analyzing circuit primitives or a few hundred transistors due to speed and capacity limitations. The switch analysis tools improve on this dramatically but nonetheless are still limited. Typical execution times for the switch analysis tools average about \( 10^7 \) vector-transistors per minute[1], implying that to run 1,000 vectors on a 1 million transistor circuit would take about a week.

These tools are utilized in two different use models. The first is to characterize circuit elements in order to create timing and power models for use at the higher abstraction layers. The second is to verify, with the highest levels of accuracy, that the completed transistor level design meets the previously targeted power specification.

An optimization tool operating at the transistor level is AMPS (Epic) which is used to automatically size transistors to trade-off timing margin for lower power. It is used in custom design flows that focus on transistor level design.

Other transistor level tools of note are RailMill (Epic) and ThunderandLightning (Simplex) which are used to analyze power bus voltage drop and electromigration effects on transistor level layouts. These tools utilize a transistor network extracted from layout and simulate the current flow in each part of the network to analyze the power grid. While unique in their capabilities to analyze these effects on VLSI/ULSI structures, these tools serve to highlight a problem only after the circuit has been completed through layout.

#### B. Logic Level Tools

Numerous logic level power analysis tools are currently available from a number of vendors: DesignPower and PowerGate (Synopsys), WattWatcher/Gate (Sente), PowerSim (System Sciences), POET (Viewlogic), and QuickPower (Mentor) among others. Each of these tools operates on a gate level netlist, such as Verilog, and assumes the availability of a gate level power library. Each computes power on a cell by cell basis by combining nodal activities with cell specific power data from a power library.

Although less mature than transistor level analysis, gate level power analysis is generally well understood. Compared to the transistor level tools, gate level tools trade off accuracy for significant improvements in run time and capacity. For example, gate level power simulations are generally claimed to be within 10 to 15% of the switch level tools, but run at least an order of magnitude faster. In addition, gate level tools tend to fit into ASIC based design flows much better than transistor level tools, as ASIC vendors generally do not make transistor netlists available. However, gate level tools are still limited in overall capacity, and are often used more in a verification role since the design must be completed, synthesized, and simulated before meaningful power results can be obtained.

Gate level power analysis tools can be differentiated from each other by considering several issues: activity sources, modeling capabilities, handling of static and dynamic power calculations, and scope.

All gate level tools require a source for nodal activities, which are usually obtained from a logical simulation. Some of these tools, such as DesignPower and PowerSim,
operate off of a vcd (value change dump) file, while others such as WattWatcher/Gate collect data activities from specialized PLI routines. DesignPower also has a probabilistic mode in which signal activities are computed probabilistically in lieu of using simulated data.

A gate level power tool requires gate level models, each of which assumes certain power abstractions. Many gate level power tools today utilize an energy-per-pin model, in which a certain amount of energy is consumed when a particular pin transitions. Another approach is to utilize energy arcs, wherein energy is consumed when a particular input-to-output arc occurs. While the arc approach is more amenable to automatic characterization techniques (basically piggy-backing on timing characterizations), pin based approaches can achieve similar results with sufficient state dependencies. Pin based approaches, however, tend to be more amenable to memory modeling than arc based models. DesignPower, WattWatcher/Gate and PowerSim utilize pin based models while POET and QuickPower use arc based models.

The underlying model also affects how the tool calculates total power, where the total power is the sum of static and dynamic power. As static power is state sensitive, the underlying model must be able to reflect state sensitivities in order to maintain good accuracy. Furthermore, the model must also be able to distinguish between leakage currents and through currents. DesignPower claims to handle leakage power, while WattWatcher/Gate and QuickPower claim to handle both types of static power.

Scope, which reflects the portion of the chip which can be modeled for power, is a key feature. Most gate level tools focus only the gate level logic, which will suffice if the design consists solely of gate level elements. However, since most chips today contain memories and a wide variety of IO structures, it is essential for a gate level power tool to handle the entire chip to get an accurate figure for full-chip power. Both QuickPower and WattWatcher/Gate claim to handle a wide variety of circuit structures needed for full chip power analysis.

The only commercially available gate level optimization tool for low power design is PowerCompiler from Synopsys. PowerCompiler uses several optimization techniques such as factoring, technology mapping, and cell sizing to achieve the lowest power under timing constraints. It is claimed to produce an average power reduction of 10% to 15% on synthesizable logic. For best results, a synthesis library is required that contains many different drive strength versions of each cell.

C. Architecture Level Tools

The architectural level, or RT level, is the design entry point for most digital designs today, and recent research has demonstrated that design decisions made at this level can have a dramatic impact on the design's overall power characteristics. Thus the use of tools at this abstraction level is of the utmost importance. [3]

WattWatcher/Architect (Sente) is the first commercially available tool operating at the architectural level. WattWatcher/Architect reads Verilog and VHDL RT level descriptions. It utilizes a conventional gate-level library and simulation data from an RTL simulation to compute a power estimate for the entire chip, including memories, clocks, IOs, datapath and control logic. Power estimates are linked to lines of source code to facilitate an understanding of what constructs result in how much power.[4]

Similar to the comparison between transistor level tools and gate level tools, architectural power estimation trades off accuracy for even larger improvements in run time and capacity. Reasonable accuracy (WattWatcher/Architect claims to be within 25% to 25% of silicon) is maintained while execution speed and capacity are significantly enhanced, thereby enabling design space exploration which would be too slow or tedious to do efficiently at the gate level. WattWatcher/Architect's run times have been measured to be more than an order of magnitude faster than analyses performed at the gate level.

D. Behavior Level Tools

Generally the least explored of the abstraction layers in terms of power, the behavior level is currently unsupported by commercial tools. While an active area of academic research, power estimation at this level is typically accomplished using spread sheets. Designers enter design specific data such as the number of combinational gates, flip-flops, IO cells, and memories along with an estimate of average switching activity into a spread sheet developed specifically for their design, or one provided by their ASIC vendor. Power data for selected cells is usually provided by the library or ASIC vendor. These spread sheets are quick and easy to use, but suffer from a very large variance in accuracy.

E. Tools Comparison by Abstraction Level

Table 1 below summarizes some of the key characteristics of low power design tools, by abstraction level.

<table>
<thead>
<tr>
<th>Level</th>
<th>Accuracy</th>
<th>Speed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transistor - Switch</td>
<td>5-10%</td>
<td>~10⁻³ V-t/min</td>
</tr>
<tr>
<td>Logic - Gate</td>
<td>10-15%</td>
<td>~10⁻³ V-t/min</td>
</tr>
<tr>
<td>Architecture - RTL</td>
<td>20-25%</td>
<td>~10⁻³ V-t/min</td>
</tr>
<tr>
<td>Behavior (spreadsheet)</td>
<td>50-100%</td>
<td>minutes</td>
</tr>
</tbody>
</table>

Table 1: Tools Comparison by Abstraction Level

IV. OPEN ISSUES

Despite numerous tools and methodologies, many issues in the low power design space remain open. Some of these issues are the subject of academic research, while others reflect a discipline that has yet to fully mature. These issues can be categorized into several groups: models, tools, and methodology.

A. Modeling Issues

Power modeling forms the foundation upon which every power tool is built. The type of underlying model directly affects a tool's accuracy, speed, capacity, use model, and versatility. In order to achieve the best accuracy, a rich set of modeling capabilities is required including static and
dynamic power constructs with state dependencies for both. In addition, the model must facilitate efficient processing so as to provide fast execution times. In particular, while the use of state dependencies is required for the best accuracy, it comes at the expense of run time. This is an area where tradeoffs are possible, wherein a limited use of state dependencies can result in adequate accuracy without unduly degrading runtime performance.

The modeling format must be versatile enough to handle large macros, such as memories and arbitrary functions. This is especially important as the use of large macrocells, or cores, becomes more prevalent in producing systems on silicon. State dependencies play an especially important role here, yet the modeling format should not require overly complex constructs to handle the virtually infinite number of states possible in a complex core. Going hand in hand with the modeling format are automatic characterization techniques. Existing techniques relying on SPICE, while adequate for gate level primitives, will not work for cores due to speed and capacity limitations. Thus automatic characterization techniques will need to utilize higher level analysis tools, such as switch or gate level tools, to overcome these limitations. Furthermore, these techniques must be developed in concert with core cell modeling capabilities to ensure easy model generation.

Once developed, these models should be usable across abstraction levels. For example, a power model for an embedded memory should work with RTL, Gate, Switch, and mixed abstraction design representations. The motivation for use of such a core cell power model at the higher levels of abstraction is accuracy, while the motivation at the lower levels is improved capacity and run time.

As a clear sign of the maturation of power issues is the emergence of power library standards. Two standards are currently in development. One is the Advanced Library Format (ALF) being developed by Open Verilog International (OVI) and the other is an enhancement of the Delay Calculation Standard (DCL) to include power constructs. While the existence of two standards seems to be self-contradictory, each standard actually addresses slightly different issues. ALF's primary focus is on the modeling constructs while DCL focuses on providing a secure tool-independent mechanism by which to distribute and access the underlying data.

B. Tools Issues

As discussed above in the survey of currently available tools, the primary issues are accuracy, speed, and capacity.

Accuracy is probably the single most important issue, closely followed by speed. For both of these parameters, conventional wisdom holds that more is usually good. However, since it is generally impossible to increase accuracy without degrading execution time, it is valuable to consider what constitutes the most effective power analysis tool. Consider two analysis results: one that is 5% accurate obtained after two hours and one that is 15% accurate obtained after one hour. If the action taken based on these results is independent of this difference in accuracy, then the less accurate tool is actually more effective since it provides results more quickly than the more accurate tool.

Execution time is also an effect of the type of power analysis. For example, cycle accurate power analysis is much more compute intensive than time-averaged power analysis. Thus to obtain analysis results in the most timely manner, it is essential to match the analysis method and tool to the specifically desired results.

Capacity has very recently become a critical issue as well. Many chips being designed today are either impractical or impossible to run at the transistor level, simply due to memory image size. For the largest designs, gate level analysis may be impractical as well. This limitation forces designers up the abstraction ladder to use RTL tools, which possess not only much greater capacity, but also faster run times, compared to gate and switch level tools.

Scope is also a key parameter, which is indirectly related to capacity. It is common practice to divide a large problem into a set of smaller problems if a tool's capacity is insufficient to deal with the problem all at once. In the power domain, this subdivision can be problematic as it is often difficult to ensure that a consistent set of simulation conditions applies to all subdivisions. A full-chip scope avoids this issue, but hits another one in accurately addressing the wide variety of circuit constructs present in a complete design. A full-chip scope is required in order to obtain meaningful values for total current draw or junction temperature.

While the previous discussion focused on analysis tools, the same issues of accuracy, speed, capacity, and scope apply as well to optimization tools. Scope and capacity are particularly critical. If the scope of the optimization is such that only a portion of the chip can be optimized, then the total amount of the optimization is limited by the size of that portion. For example, consider a logic optimization tool that can reduce power by 25%, but only on control logic. If the control logic only constitutes, say, 20% of the entire power budget of the chip, then the total power savings is only 5%. Capacity is perhaps an even bigger issue. If large portions of the chip cannot be optimized simultaneously, then it is likely that the optimization techniques will reach a local minimum instead of a true minimum. Lastly, applying uniquely to optimization tools are the particular minimization techniques employed. While a number of well known optimization techniques exist at the gate and transistor level, it is generally recognized that techniques available at the higher abstraction levels promise much greater reductions. Unfortunately, no commercial power optimization tools exist today above the gate level.

C. Design Methodology

Most design projects have historically utilized feed-back design methodologies, wherein a target architecture or concept is implemented via gate level design or even transistor layout in order to evaluate its power characteristics. This methodology features a relatively lengthy feedback loop with analysis at the gate or transistor level. Thus, information about the design's power characteristics are not obtained until quite late in the design process, and once this information is available, it is fed-
back to the upper abstraction levels to be used in determining what to do to minimize the power. The farther the power analysis results are in excess of the target specification the higher in the abstraction levels one must return in order to modify the design to meet the target specification.

Design methodology has evolved to this particular point due to the availability of good, reliable lower level tools such as transistor level timing and power analysis tools, and the previous lack of robust tools at the upper abstraction levels. However, this approach has rapidly become inadequate as the feedback loops from the lower level analyses are too long and time consuming. Furthermore, from a schedule perspective, waiting to get accurate power estimates until a gate or transistor level netlist is available dramatically increases the risk of significant, last minute schedule delays. In an era of decreasing time-to-market, this effect is going in the wrong direction. Problems must be avoided, or designed out. It will be too expensive to find them at the end. This will drive the move to early design cycle analysis and optimization.

In addition, research into low power design techniques has shown that the most dramatic power reductions occur from optimizations at the highest levels of abstraction.[5] These observations all point to the need for early analysis and optimization, and a modification of the feed-back design methodology.

V. FUTURE DIRECTIONS

The opportunities for dramatic power savings that exist at the upper abstraction levels will motivate a number of changes and developments. The first will be a modified design methodology that can be viewed as a feed-forward approach, as opposed to the more conventional feed-back methodology described above.

To fully manage and optimize power consumption, a design methodology must address power consumption issues at each stage of the design process and at each level of the design abstraction. Thus, designing for low power will entail starting with a target specification, designing an architecture to meet that specification, performing abstraction level specific optimizations to minimize the power, and then checking to verify that the design is on track to meet the initial specification.

In the feed-forward approach, illustrated in Figure 1, the lengthy, cross synthesis, cross-abstraction loops of the feed-back approach are replaced with more efficient abstraction specific loops. The design that is fed forward to the lower abstraction levels is much less likely to be fed back for reworking, and the analysis performed at the lower levels then becomes a verification effort instead of a design effort. For example, architecture level power analysis is used to ensure that the RTL design passed forward to the logic level is a design that is fundamentally capable of meeting the target power specifications. Other information that could be fed forward includes floor-plans, clock distribution styles, module micro-architecture's, critical net names and power bus topologies. Later in the design flow, gate level and transistor level power analysis is targeted at verifying that the design still meets the previously set power specifications and that any lower level optimizations have achieved the desired results.

Two basic issues will motivate the adoption of this design methodology: first, the aforementioned issues of tool capacity and runtime and second, the opportunities for power reduction at the higher levels of abstraction. On the tools front, this methodology efficiently leverages the speed/capacity/accuracy tradeoff. Early in the design process at the higher levels of abstraction, design exploration requires tools that are fast and capable of easily analyzing the entire design. Later, during verification efforts, which are run much less often, run time becomes less critical and accuracy becomes the driving factor. Thus, this methodology utilizes the best tool for each task, instead of trying to use a one-size-fits-all tool for all tasks.

Figure 1: A Feed-Forward Design Methodology

On the power reduction front, this approach encourages the design efforts to be spent up front, at the architecture level, which is where research has indicated the largest power savings opportunities exist. This will drive the development of optimization capabilities at the RT and behavioral levels. Examples of these tools have already been demonstrated in academia.

Lower level gate and circuit level optimizations, while offering much smaller improvements [6], will continue to
be enhanced to produce larger power savings. Tools to support clock gating, while only a research effort right now [7], will prove especially effective although they will in turn drive the need to understand peak currents, and its derivatives, in much greater detail.

Further emphasis will be placed on RTL analysis by the growth in design complexity. Since complexity, as measured by transistor count, is rising faster than tool capacity, efficient analysis of the largest designs must be done at the RT level. In addition, the anticipated move by ASIC vendors to RTL design signoff will require that accurate power estimates be available at the time of signoff, prior to synthesis or layout.

Optimization of application software will also be pursued, enabled by the emergence of very high level analysis tools that can evaluate the effect on power of different instruction streams running on target architectures. A key enabler here will be the emergence of accurate and efficient models for highly complex devices, such as microprocessors, microcontrollers, and digital signal processors.

Power supply voltages will obviously continue to drop, leading to the increased prevalence of multi-supply voltage and variable supply voltage devices. This will put significant pressure on modeling capabilities and design tools, as fewer tools today are capable of adequately supporting such aggressive low power design techniques. Furthermore, many tools and models will need to be enhanced to support low power circuit design styles, such as limited swing signaling and variable threshold logic.

VI. CONCLUSIONS

Low power design is no longer of interest only to those developing battery powered electronics, but has also become important in a wide variety of applications. Various power analysis tools have emerged to address many of the issues, at several levels of design abstraction. Architectural level design practices and feed forward design methodologies have been developed to utilize these new tools in a comprehensive, top-down methodology for low power design. While the pace of development in low power design has been rapid, much work remains to be done, specifically in the area of modeling, tools, and methodology development.

REFERENCES


