IBM’s 50 Million Gate ASICs

Juergen Koehl
koehl@de.ibm.com

David E. Lackey
delacke@us.ibm.com

IBM Microelectronics
1000 River Road
Essex Junction, VT, 05452, USA

George Doerre
doerrg@us.ibm.com

IBM Microelectronics
Route 52
Hopewell Junction, NY, 12533, USA

Abstract - There is no slowdown in the complexity increase for ASIC and SoC designs. As we write this paper in August, 2002, 40M gate ASICs are nearing tape-out, and 50M gate designs are likely to start before this conference takes place. This paper describes the current tool and methodology development efforts focused on enabling ASIC and SoC designs of these sizes and complexity, centered around the reduction of design turn-around-time, improvement of the quality of results and the modeling and optimization of deep sub-micron electrical effects.

I. Introduction

Designs of this size will not be possible without a considerable amount of reuse, and we expect the percentage of cores and memory to increase over time. This leads to the observation that SoC design complexity will increase more slowly than the raw growth in chip-level gate count.

The 50M gate chip design will replace entire systems and the design teams will in many cases be distributed globally. Different parts of the design will be developed on different schedules. Deep sub-micron effects such as crosstalk and voltage variation will increase in importance and result in more stringent design rules, which in turn will increase the complexity of the timing and layout methodologies. For our customer base, design turn-around-time, the achievable performance, and first time right quality are the most important criteria. Power reduction is an important factor due to both the physical effects of technology scaling, and emerging low power applications.

II. Design Size and Design Style

The increase in design complexity over the years is shown in Table I. The current size of the largest ASICs is around 40M gates and we expect the first 50M gates design in 2003.

Clock frequencies range up to 900 MHz for the core logic and more for high speed interfaces.

An important characteristic of the design style is the degree of hierarchy used. The 50M gate design contains a natural hierarchy of functional, often reused, IP. The development logistics of the multiple design teams (parallel or even different schedules) may also lead to a natural partitioning of the design.

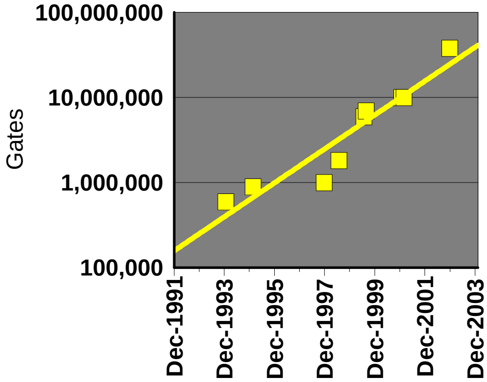

TABLE I
Complexity Increase

On the other hand, a fine-graine partitioning and the necessary floorplanning may limit the opportunities for global design optimization. In many cases, the logic partitioning is not an appropriate physical partitioning. From this point of view, the decision to chose the right level of hierarchy is very design-dependent and requires a detailed understanding of the content, schedule and dataflow of the design. Let us briefly summarize the different design styles.

A. Flat Design

There are no truly flat designs: each design has a certain amount of cores and memory elements that introduce hierarchical boundaries. The remaining part of the logic is placed and routed flat under global timing and congestion optimization criteria. Current synthesis tools are limited in capacity and therefore a partitioning strategy is usually employed in this step. Part of the sub-optimality introduced by synthesis, due to the necessary partitioning, is recovered by the later step of global in-place optimization. The difference in design style typically occurs in the physical implementation. Figure 1 shows the typical relationship between core, memory elements, and standard cells in a 2001 tape out with 1.7 million instances.

The use of trend curves for design complexities hide the fact that this increase as well as the algorithmic and CPU speed improvements are typically step functions and that the resulting overall run time for the largest flat designs varies therefore considerably. With the combined effort of algorithmic improvement and the availability of fast hardware, we were able to deal with the complexity increase in the past.

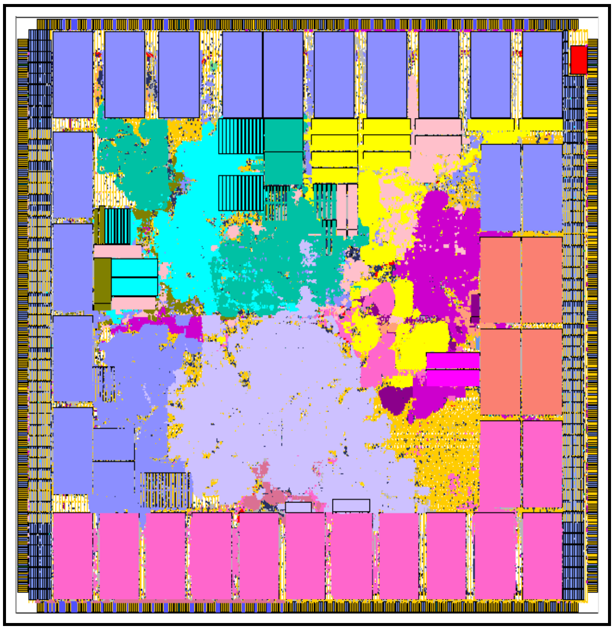

FIGURE 1
NP-1 10-Gigabit 7-Layer Network Processor (see [15])

An interesting constant we have seen, over a period covering the last decade, is the ability to contain a flat placement of the largest ASIC design on the fastest available hardware to a single overnight run. Both placement tools used within IBM [14], [4] can place a 3M instance design in about 10 hours on an IBM pSeries 690 server.

We expect future algorithmic and IT improvements to keep up with the design complexity increases. Parallel algorithms and parallel processing systems are examples of the methods used.

The increase in design complexity is therefore not requiring a fully hierarchical design approach.

B. *Full hierarchical design*

Figure 2 shows a design that is predominantly hierarchical. The majority of the logic in this design is organized into hard blocks called *Random Logic Macros* (RLM's). The hierarchical methods described in this paper allow some logic gates to exist at the top-level of the design (outside of the RLM's), typically consisting of buffers or control logic between RLM's, and I/O cells and their boundary test logic.

For a hierarchical design, the chip's wiring layers above an RLM are assigned either to RLM routes or top-level routes. Metal layers can also be shared between the RLM and the top-level. Area Array I/O technology allows I/O pads and cells to be distributed over the chip's standard cell area. I/O cells belong logically to the top-level, but can be placed inside an RLM's area by reserving the placement area, within an RLM, required for each overlapping I/O cell.

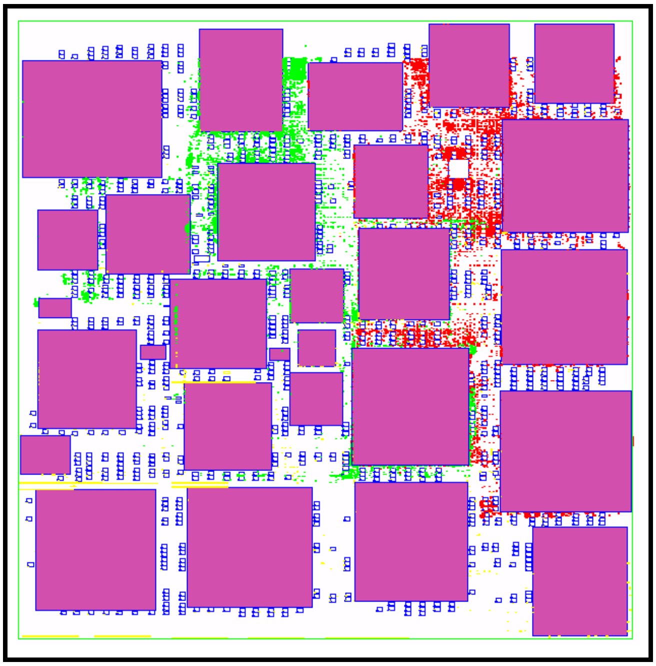

FIGURE 2
Hierarchical Design

Many design teams exploit hierarchical design to expedite timing closure through parallelization of design effort and processing. The success of this approach hinges in three key factors:

- Partitioning: Logical and physical design planning (manual and tool-driven) are used to determine how the design should be subdivided into sub-blocks (RLM's) that are optimum for synthesis and layout.
- Time budgeting: Whereas timing constraints at the chip-level are a critical factor in timing closure for all designs, hierarchical timing closure requires timing constraints that also describe the boundaries of each RLM. For logic paths that cross the boundaries of RLMs, synthesis and timing-driven placement must proceed iteratively. Proportion such paths' delay targets between an RLM, the top-level, and other RLMs, and accordingly budget the timing constraints for each RLM. Use the budgeted constraints towards timing closure of each RLM, then the top-level design. As these timing closure actions affect the proportions of the above paths, updates to the time budgets and re-optimization may be necessary for convergence.
- Top-level timing closure: This is critical on several fronts: developing a reasonable starting point for time budgeting, global optimization techniques for top-level paths, and the ability to *see through* paths that cross into and from RLMs.

For time budgeting, a first-level electrical correction and buffering is necessary to avoid long wires and high fanouts that would create misleading top-level delays and result in unreasonable RLM boundary constraints. Whereas partitioning creates RLMs whose timing closure is localized and thus reduced in complexity, paths comprised of top-level gates require global optimization techniques of the same order of effectiveness as large flat designs. Finally, while closing timing for top-level paths that traverse RLM's, visibility of the entire path is necessary for avoiding TAT-consuming iteration.

C. Hybrid approaches

Often, a mixture of the flat and hierarchical design styles provides the fastest TAT for a design. To achieve timing closure for the logic contained within a small high performance clock domain, for example, an RLM-based approach allows the designer to focus on placement solutions for this logic that meet timing requirements through minimum signal wire lengths. On the other hand, the chip's area may be dominated by a large, complex control function of lesser clock frequency. Flat placement, particularly with partitioning-based timing-driven placement tools, can often achieve the best global solution for timing, placement density, and wiring congestion.

Further, there are advantages in performing some steps in the design flow hierarchically, with later steps flattened. For example, while hierarchical solutions can often provide good placement and timing results (especially for designs of mixed timing criticality and placement densities), routing can often benefit from an unconstrained chip-level solution with all metal layers available to the routing tool. Further, global timing problems that remain after exhausting the designer's hierarchical timing closure approaches, can benefit from relaxing the placement boundaries and performing an incremental global placement and boundary optimization.

D. Hierarchy vs. Flat

The decision to follow a hierarchical or flat approach has to be made for each individual design. Generally speaking the advantages of the hierarchical approach are

1. the ability to start design planning with incomplete information for example the top level netlist and estimated information for the RLMs,
2. the fast turn-around-time for an RLM only EC without a floorplan change.

The advantages of the flat approach are

1. the possibility to use global optimization techniques in cases where the logic partitioning is not an appropriate physical partitioning,
2. the reduced complexity of the design flow and data management and
3. a faster TAT for ECs that would require a floorplan change.

A good guideline for an appropriate use of hierarchy is

- to base the decision on the availability date and the likelihood of ECs for the different partitions of the design,
- to implement reused components as hard macros and
- to limit the maximum number of hierarchical blocks - if any - to a small number.

An important goal for IBM's tools development is the ability to flatten and globally optimize any design if timing or wirability closure cannot be obtained based on the given hierarchy.

III. Design Methodology

A. Overview

An overview of the SoC chip design methodology is shown in Figure 3.

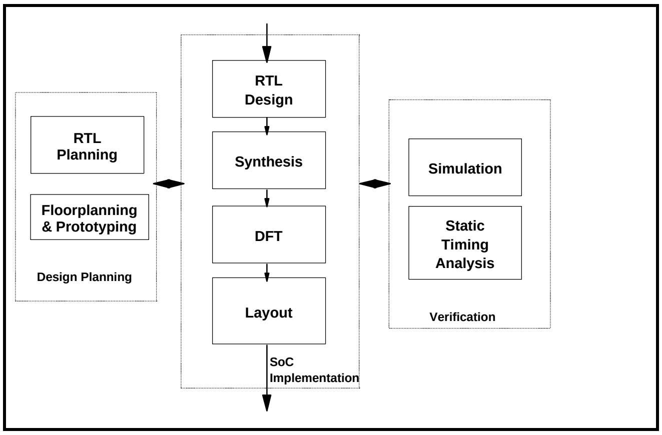

FIGURE 3
Design Methodology

B. RTL Design

The SoC designer describes all functional (logical) behavior at the Register Transfer Level (RTL). This description is a Boolean-accurate representation of the design, and also defines the connection of signals to arrays, phase-locked loops, (PLL's), chip I/O, and other embedded IP.

C. Design Planning

The designer uses various tools for Design Planning [8] to analyze the design for timing behavior, chip area requirements, and power consumption. The designer also determines the optimal functional partitions and the physical locations of partitions and embedded IP on the chip. After an initial logic synthesis of the RTL into a gate-level form, floorplanning (or virtual prototyping) tools [12] are used by the designer to provide the analysis and physical requirements. Further, the designer extracts, using these tools, directions for synthesis (e.g. partition-level timing budgets and wire loads) and layout (e.g. region constraints or hierarchical RLM boundaries).

Additionally, there is a new class of tools arising from the EDA community that performs RTL planning [18], or Virtual Prototyping at the RTL level, so that the designer can begin to assess and physically plan the design even before gate-level synthesis. The value of this approach can be faster correction of the RTL.

D. Synthesis

The RTL or an initial gate-level description of the design, together with information provided by design planning, are optimized using logic synthesis or physical synthesis tools. Synthesis optimizes the logic for timing, area, and power consumption. Physical synthesis tools [17] create an initial placement of the design using the designer's chip utilization plan, and assigns gate placement locations to optimize the logic for timing, while avoiding wiring congestion.

Synthesis tools cannot, at present, optimize an entire 50M gate chip, and thus individual partitions (from design planning) are synthesized. The partitions are integrated into the SoC using design planning tools.

E. Design for Testability (DFT)

IBM's DFT methods [11] allow designers to largely ignore hardware test issues when designing the functional logic. While providing this flexibility for functional design, these DFT techniques also provide extremely high levels of stuck-fault coverage (> 99% on most chips) using automatic test pattern generation (ATPG) software. By using these DFT methods (which include software for DFT design automation and verification) and ATPG, the designer is mostly freed from the need to determine test patterns.

F. Verification

The designer verifies the functional behavior of the SoC using logic simulation, cycle simulation, and formal verification tools. Once the functional behavior is found by the designer to be correct, his focus on verification turns to the correct implementation of the functional design in the implementation of the silicon chip.

Verification of the chip implementation is based upon the following key components:

- *Timing verification.* Assessment of all timing paths accounting for the range of SoC operating conditions (ie. voltage and temperature) and variation in manufacturing process (which can occur even across a single chip).
- *Design for Testability (DFT) structures and ATPG.* Verification of race-free (or timing-independent) and full-scan latch structures [11], that ensures the chip can be tested with patterns from combinationally-based automatic test pattern generation (ATPG) tools, and ensures that automated failure diagnostic patterns can be generated.
- *Technology-based checking.* Checks for logic structures required by the CMOS manufacturing and test equipment, estimations of noise and electromigration effects, placement and wiring checks, DRC and LVS checks to ensure that process rules were not violated during the physical design process.
- *Logic equivalence checking.* Verification throughout the chip implementation process that structural changes introduced (e.g. DFT, clocking, optimization) have not modified the original Boolean function.

G. Layout

The goal of the layout phase is to find a feasible physical implementation that satisfies the designer's timing and die size requirements, or demonstrates the non-feasibility of the design, in the shortest possible time. The approach to layout needs to conform with any proposed hierarchical partitioning, but should not limit the maximum partition size due to complexity limits given by the layout tools and methodology.

The major steps in the layout flow are

1. Congestion- and timing-driven placement [4], [5], including in-place timing optimization [7]. The exit criteria from this step is a positive slack with ideal clocks and wiring based on estimated Steiner routing.
2. Clock construction. The exit criteria of this step is timing closure with real clocks and estimated Steiner wiring.
3. Global routing. The exit criteria of this step is timing closure with real clocks and wiring based on global routing estimates.
4. Detailed routing and electrical analysis.

IV. Design Turn-around-time

It is a trivial observation that the turn-around-time of any process is determined by

1. The number of steps.
2. The number of iterations for each step.
3. The run time of each step.
4. The hold time between each step.

Turn-around-time reduction has to address all of these issues. To identify improvements in TAT, a detailed measurement of the TAT for every part of the design flow, and an analysis of design process contributors to the TAT, are necessary. This also provides an early warning system for TAT detractors due to changes in the design methodology required by new technology requirements. Furthermore, ASIC design is a process with a high degree of uncertainty since each design has it's own challenges, the input quality differs from design to design, and unexpected problems are not uncommon. In a large ASIC design operation with several hundred tape-outs per year, we can see TAT variations between designs of similar complexity of up to 3X from design to design, between customers of up to 2X, and 1.3X between design teams. In each case the tools and methodology are fundamentally the same. Such measurements are the base for a detailed root cause analysis that leads to further TAT improvements.

A. Reducing the Number of Steps:

The 'art' of a methodology definition is to assure the quality of the design process while avoiding any unnecessary process steps. One of the methods is to encapsulate more sophisticated operations in a single tool or step, and hide the interactions between these steps from the user. We were able to reduce the number of individual steps to be executed by the designer from around 200 to about 130 in one year alone.

B. Reducing the Number of Iterations for Each Step:

Iterations are mainly caused by tools that do not address all the design criteria and a non sufficient quality of the input. Timing and congestion driven placement [5], [4] or noise aware routing [13] are examples which incorporate design requirements in the optimization steps to avoid unnecessary iterations. Sophisticated sign-off criteria are necessary to identify problems in the input early enough to avoid difficulties downstream, and allow the fixing of these problems with the least amount of TAT impact. Simple measurements can be very effective. For example, if the static timing analysis of a given netlist (neglecting all the wire loads: e.g. zero wire load analysis) results in a negative slack, it is certainly not possible to find a floorplan that satisfies the timing requirements. This simple observation was incorporated as a filter in our sign-off criteria and has led to a considerable reduction in TAT [10].

C. Reducing the Run Time of Each Step:

This is the domain of tools developers and high-performance servers. Over the years we have seen considerable run time improvements due to both IT and algorithmic research. The 'constant' placement run time for the largest designs mentioned above is one example.

D. Reducing the Hold Time Between Steps:

Automation is the answer to this challenge. The goal is a completely automated flow with zero downtime between steps. We maintained above that every design has it's own challenges, but we have seen that a default or baseline methodology can be personalized for a specific design and can be automated for minimal TAT on the final netlist.

By applying all of these considerations, the IBM ASIC TAT was effectively reduced by 42% in 2001, as measured on the final design version (e.g. the production netlist).

V. Performance, Density and Power

Besides TAT reduction, the quality of the resulting design is another optimization criteria for the design system. As much as the improvements in technology, design automation has to contribute to achieving the performance, density, and power requirements of the most complex ASIC's.

A. Performance

Achieving aggressive clock frequencies in an ASIC environment starts with the timing quality of the technology and the timing accuracy of the circuit libraries.

Timing-driven placement and in-place optimization is mandatory for high-performance ASICs. The most recent enhancement is the automation of clock scheduling, or useful skew. We are currently applying these techniques to the first production parts, and are seeing performance improvements of up to 10% of the desired cycle time [2].

B. Density

Circuit design is again a major contributor to the overall density. Congestion-driven placement [5], [4] has provided an automated alternative to the classical manual tuning of the densities of the various logic blocks. New approaches to global routing could further improve the wirable circuit density [3].

C. Power

As technology scales for increased System-on-Chip (SoC) density and performance, the need to manage power consumption increases in significance [6] as designers strive to utilize the advancing silicon capabilities. The consumer product market further drives the need to minimize chip power consumption.

Due to technology scaling, the capacitance per unit area increases with each process generation. The power increase represented by this capacitance increase is offset by the scaling of the power supply voltage, Vdd. The frequency of operation, however, increases with each generation, leading to an overall increase in active power density from technology generation to technology generation.

Further, as Vdd is reduced, the transistor threshold voltage (Vt) must be reduced in order to maintain or improve circuit performance, despite the drop in Vdd. This decrease in Vt and Tox then drives significant increases in leakage power, which has previously been negligible [1]. As silicon technologies move into the 90nm lithography generations, leakage currents become as important as active power in many applications.

Thus, the need arises to minimize power consumption in 50M gate designs, while maintaining the growth in performance and circuit density now available with 90nm process technologies and beyond. Several approaches will be taken to manage the power problem:

- Leveraging periods of functional inactivity in logic blocks:
  - Clock Gating: reduces active power,
  - Power Sequencing [9] : reduces active and leakage power.
- Operating each functional blocks at a unique voltage level [9]: higher voltage for only timing-critical logic functions, and lower voltage to reduce active power for less-critical functions.
- MultipleVt libraries: By applying higher-leakage yet higher-performance Low-Vt logic libraries for only critical timing paths, the design as a whole can be driven by a lower voltage supply (reduces active power), while applying High Vt libraries (reduces leakage power) for less-critical functions.
- Synthesis- and placement-based methods of reducing power through gate-level optimization choices.

VI. First Time Right

A first-time-right design methodology delivers both design methods, and verification of the chip implementation, that result in the highest likelihood that a functionally-verified ASIC or SoC will work upon initial product integration. As described in Section III, such a methodology is based upon the following key components:

- Timing verification.
- Design for Testability (DFT) structures and ATPG.
- Technology-based checking.
- Logic equivalence checking.

Designing in silicon technologies that enable 50M gate SoC's creates additional considerations in maintaining a First Time Right design methodology into the future.

A. Power and Supply Voltage Distribution

In the past, predefined correct-by-construction chip images could be used to avoid design problems related to power distribution. These images contained robust power grids that were pre-verified to work for ASIC designs having been implemented on these images and having passed technology-based checking.

However, nanometer-level technology advances increasingly create power distribution issues that must be considered on a per-design basis. Voltage or IR drop analysis is required to assess variation in power consumption based upon voltage levels and circuit densities. Signal switching and off-chip power supply noise also needs to be analyzed. Further, these design properties increasingly vary within a single chip, as SoC design sizes increase to 50M gates and beyond. These electrical effects result in variations to logic signal timing, and their accurate measurement is critical in both improving the robustness of the power distribution structures, as well as calculating these variations accurately in static timing analysis, design optimization, and noise avoidance (see next section).

B. Noise and Electrical Analysis

The share of the delay contributed by interconnect increases for every new technology generation. Interconnect loading has also become more complicated, in stages. At first, it could be modeled by a simple lumped capacitance to ground. More recently, the loading has become a distributed RC load, needing to enhance modeling with coupling to adjacent nets, which could theoretically be switching in opposition to a single net. LC load modeling has also become an important consideration for chip I/O signals. Copper metallurgy provided a one-time reduction to resistance, but could not offset the ongoing linear increase in net resistance driven by technology scaling. Even beyond this, "width-dependent" resistive effects for narrow copper lines need to be modeled. Beyond timing closure, the first noise effect, coupling- induced jitter, needed to be considered as effectively additional timing margin. Accuracy is critical, because a guard band that is too large, or too broadly applied, causes timing pessimism that negates the achievable performance of next-generation CMOS technologies.

Microprocessor designs exploit CMOS performance more than typical ASICs for a given process technology generation. Through their counterparts in microprocessor design, IBM ASIC developers have an advanced look at the types of net modeling needed, both for timing closure, and for noise avoidance. ASIC designers have visibility to the fact that coupling effects will continue to increase, and keeping aggressor and victim nets separate will become a larger focus of the physical design process [13]. Also, because of the small size of gates and latches, soft error rates (SER) could become more of a constraint for future bulk CMOS technology generations. Electromigration, though not strictly electrical analysis, will also become more of a constraint for copper metallurgy in the future.

VII.Outlook

Looking ahead, the design challenges for future ASICs will generally be of three types:

A. Continued Performance vs. Capacity Challenges.

100M gate designs will be accomplished in the next-generation nanometer-level CMOS technology, and customers will expect block-level performance to increase by at least 20%, and require circuit-level performance increase by even more. Even at these higher performance levels, customers will expect a fail-safe noise methodology, just as they expect today for timing closure. Power management and reduction will be more critical, and we expect this to be handled at the architectural level as much as at the logical level - the greatest lever may be slowing or stopping of block-level clocks, or completely powering down blocks within the design.

B. Process Technology-related Challenges.

At 100M gates, the nominal primary power supply will be less than one volt. This, along with higher cross-chip feature-size variances and constant devices threshold variances, will mean that a chip's operation needs to be modeled across a wider process window. Being able to model the ASIC logic and memory array libraries, at all corners of this window, will become critical. Designers will need to trade off multiple supply voltages, both the number and level, as part of the chip-level architecture and optimization. Designs will need to be more "manufacturing-aware" in important regards:

- BEOL physical structures to improve manufacturability of copper and low-k interconnect
- FEOL/BEOL lithographic needs, as the lithographic k_1-factor decreases.

C. System Design-related Challenges.

IP reuse and bus-level interconnection will increase substantially in the next generation of products, to effectively increase designer productivity. Designers will need to functionally verify their designs to a more exhaustive extent than today, because substantial parts of their design will have been provided by 3rd parties or other design groups within their own company. Aside from ensuring that their system functions as architected, designers will need to verify the correctness and performance of the 3rd party IP as delivered, as integrated at the SoC-level, and after ECO's.

Designers will also be facing true system-level considerations, such as the architecture of power management, increased hardware/software co-design, and the use of programmable or reconfigurable logic.

Beyond these challenges, designers will face greater mask and prototyping costs, even with the productivity improvements from 300 mm wafer fabrication. With this, single-pass design success will remain the most significant overall measure of a best-of-breed ASIC methodology.

References

[1] Adan, A.O. and Higashi K., "OFF-State Leakage Current Mechanisms in BulkSi and SOI MOSFETs and Their Impact on CMOS ULSIs Standby Current", IEEE Transactions on Electron Devices, Vol. 48, No. 9, Sept. 2001, pp. 2050-2057.

[2] C. Albrecht, B. Korte, J. Schietke, J. Vygen: "Maximum Mean Weight Cycle in a Digraph and Minimizing Cycle Time of a Logic Chip", Discrete Applied Mathematics 123 (2002), 103-127.

[3] C. Albrecht: "Provably good global routing by a new approximation algorithm for multicommodity flow", Proceedings of the International Symposium on Physical Design, April 2000, pp. 19-25.

[4] C. J. Alpert, G.-J. Nam, and P. G. Villarrubia, "Free Space Management for Cut-Based Placement", Proc. IEEE Intl. Conf. on Computer-Aided Design, November, 2002.

[5] U. Brenner, A. Rohe: "An Effective Congestion Driven Placement Framework", ISPD 2002, 6-11.

[6] Enomoto, E., "Low Power Design Technology for Digital LSIs," IEICE Transactions on Electronics vE79-C n 12, Dec. 1996, pp. 1639-1649.

[7] J. Koehl, J. Schietke: : "In-place Timing Optimization", Proceedings of SAME, Sophia Antipolis, 2000.

[8] D.E. Lackey, "Design Planning Methodology for Rapid Chip Deployment", The Eighth IEEE/DATC Electronics Design Processes Workshop, April, 2001.

[9] D.E. Lackey, P.S. Zuchowski, T.R. Bednar, D.W. Stout, S.W. Gould, J.W. Cohn, "Managing Power and Performance for System-on-Chip Designs using Voltage Islands", Proc. IEEE Conference on Computer-Aided Design, November, 2002.

[10] B. Marshall, T.Wagner, J.Koehl: "A New ASIC Timing Signoff Methodology", IBM MicroNews, 2/2002.

[11] S.Oakland, J.Monzel, R.Bassett, P.Gillis, "An ASIC Foundry View of Design for Test", Proceedings of the IEEE International Test Conference, 1994.

[12] J.Y. Sayah *et al.* "Design Planning for High-performance ASICs", IBM Journal of Research and Development, vol.40 no.4, July 1996, pp. 431-452

[13] T.Stoehr et. al. "Analysis, Reduction and Avoidance of Crosstalk on VLSI Chips", Proceedings of the ISPD 1998.

[14] J. Vygen: "Algorithms for Large-Scale Flat Placement", Proceedings of the 34th Design Automation Conference, ACM 1997, 746-751.

[15] Http://www.ezchip.com/

[16] "First Encounter", © 2002 Cadence Design Systems, Inc., http://www.cadence.com/products/first_encounter.html

[17] Synopsys® "Unified Synthesis and Placement", copyright 2002 by Synopsys, Inc., http://www.synopsys.com/products/unified_synthesis/unifies_synthesis.html

[18] "TeraForm® RTL Design Planner for Deep Submicron SOCs", © 2002 Tera Systems, Inc., http://www.terasystems.com/products/datasheet.htm