# INTERNATIONAL TECHNOLOGY ROADMAP FOR SEMICONDUCTORS

## 1999 EDITION

# DESIGN

# Table of Contents

# List of Figures

# List of Tables

# DESIGN

## SCOPE

The objective of the semiconductor business is to produce chips that create profit for integrated circuit manufacturers, their customers, and their suppliers and strategic partners. The increased difficulty in designing, verifying, and testing these chips has become a larger barrier to achieving this objective than providing the technology for manufacturing them. This chapter describes the challenges in producing correct, performing designs and potential solutions for meeting them. While test equipment and the test of manufactured chips is covered elsewhere, this *Design* chapter addresses design for testability, including built-in self test (BIST). As in the other areas covered, both advances in and roadblocks to technology solutions create problems and opportunities in design.

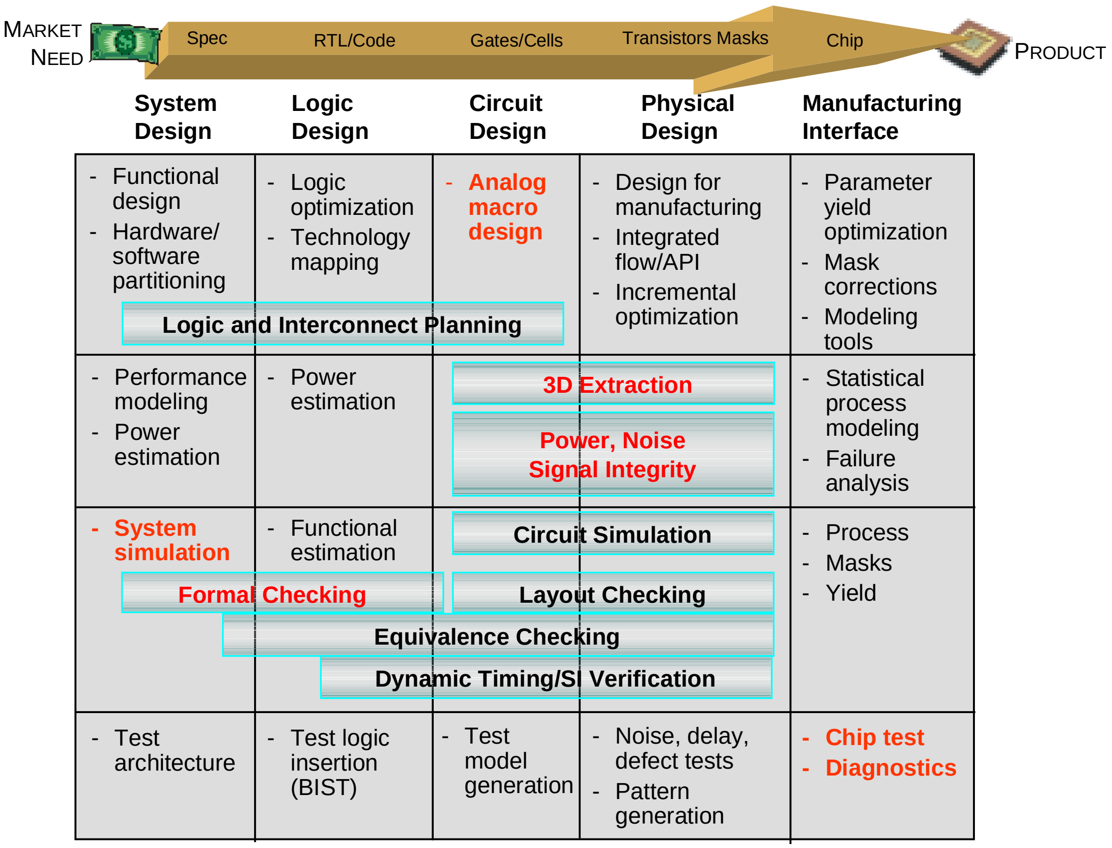


*Figure 4 Issues in IC Design*

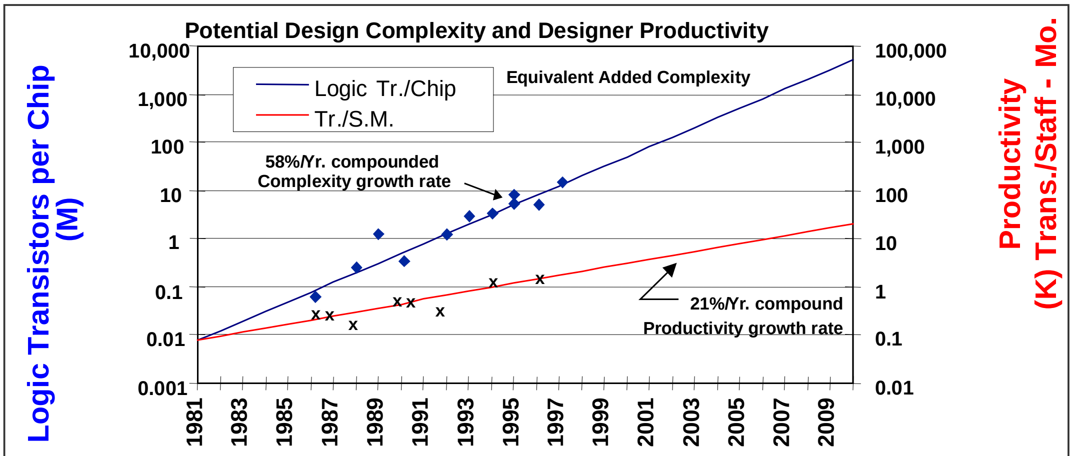
The increasing complexity of design requires a corps of highly skilled and more broadly trained designers, and computer-aided design tools that take into account factors that could be ignored in previous technology generations. Figure 4 illustrates that tools, which earlier could concentrate on discrete parts of the design problem, now must be aware of a broader range of factors. For example, logic synthesis tools now must take interconnect and physical placement into account and increasingly accommodate the transistor-level properties of dynamic circuit families and the noise effects at smaller feature sizes and higher frequencies. The design flow must assure functional correctness and timing, power, reliability, manufacturability, signal integrity, and testability requirements. Thus, design complexity increases superexponentially, but automated

Table of Contents Previous Next Last

tools are handicapped by having to run on the previous generation of computing equipment. This complexity has a similar effect on designers (whose abilities are not tracking the Moore curve), with distinctions being blurred between logic design, layout, and circuit design.

As designs move from digital microprocessors and application-specific integrated circuits (ASICs) to system-on-a-chip (SoC), designers and design tools also encounter complex embedded software, more heterogeneous systems, and the challenges of providing a diverse range of components on a single chip. Analog and mixed-signal, radio frequency (RF), microelectromechanical systems (MEMS), electro-optical, electro-biological, and other nontraditional elements arise from and give rise to changes in technology. With this rapid rise in heterogeneity, technology advances, and new product requirements, reuse of already designed intellectual property (IP), once considered to be a mitigating factor in design productivity, becomes difficult. Moreover, incorporation of several separately-designed components on a chip requires significant integration and verification cost.

The rapidly changing technological environment also shrinks product life cycles, making time to-market one of the most critical issues for semiconductor customers. This demand is driven by growth of personal information flow through wired and wireless voice and data communication, and the internet. There is great pressure to reduce the total time to create a chip for products that are not obsolete before they are brought to market. This time is dominated by the system design and verification phase. Investment in technology improvements has dominated product creation resources, and design productivity has not been able to keep pace with transistor density growth. Figure 5 indicates the growing productivity gap between transistors available and those which can be designed in microprocessors. As noted above, design reuse addresses only part of the productivity gap. Increase in design team size has problems with respect to productivity of large groups and is limited by integration and software issues.



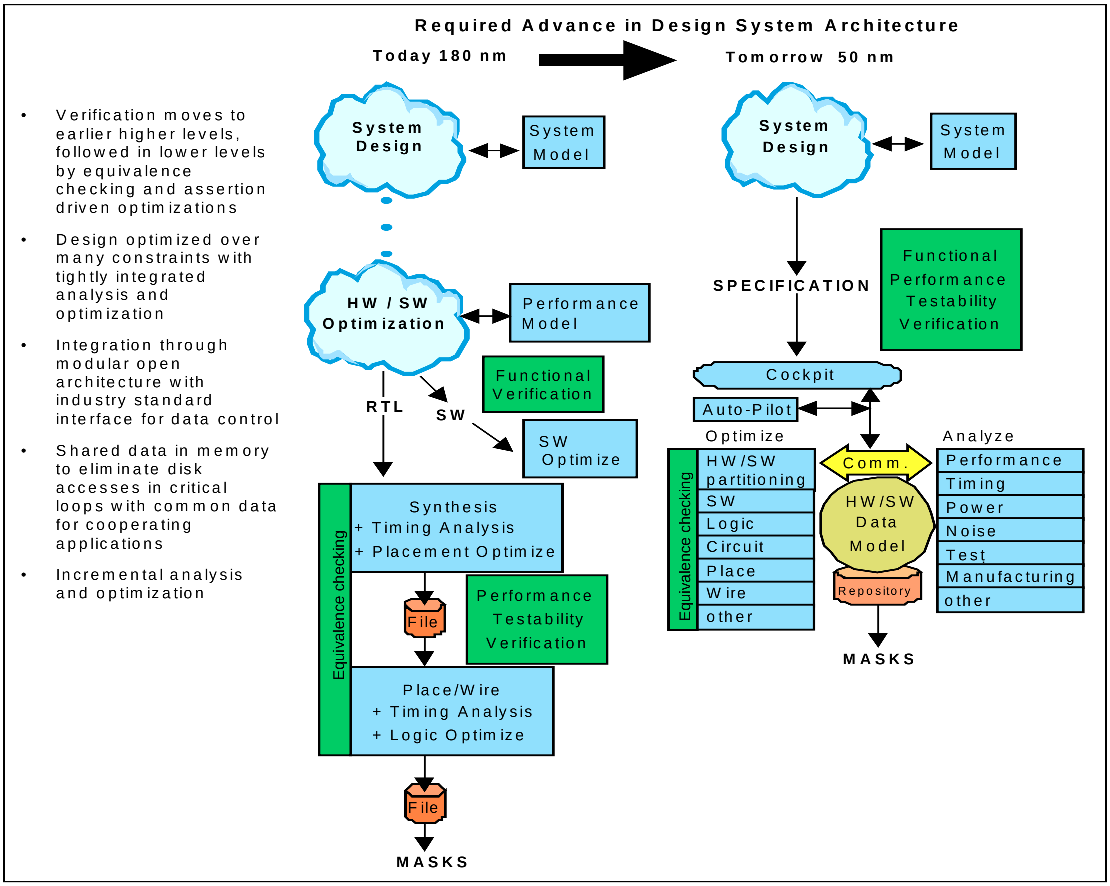
| Year | Technology | Chip Complexity | Frequency | 3 Yr. Design Staff | Staff Cost* |
|---|---|---|---|---|---|
| 1997 | 250 nm | 13 M Tr. | 400 | 210 | 90 M |
| 1998 | 250 nm | 20 M Tr. | 500 | 270 | 120 M |
| 1999 | 180 nm | 32 M Tr. | 600 | 360 | 160 M |
| 2002 | 130 nm | 130 M Tr. | 800 | 800 | 360 M |

* @ $150K / StaffYr. (In 1997 Dollars)

*Figure 5 The Design Productivity Gap*

As time-to-market issues and system-on-a-chip become more common, embedded software emerges as a key design problem. The software content of products and its cost continues to rise. Issues surrounding this include high-level architectural design space exploration; analysis of the tradeoffs (design cycle time, performance, cost) of doing designs in hardware and software; high level design planning and estimation; hardware/software co-design at all design levels; and difficult verification and analysis issues. Software reliability becomes a greater factor, especially in critical applications and in distributed systems. Software, previously designed to run on a dedicated processor now runs in a global, interconnected environment with open systems. High-level design and the accompanying design automation tools are often considered key to design productivity improvement; however, to avoid frequent rework and iteration, decisions made at higher levels of design must be based on accurate estimation of the effects of lower level decisions, and/or must permit later adjustments to logical and physical factors. Figure 6 illustrates the transition of design systems from the traditional chart in which design proceeds independently at discrete levels, to an integrated system in which logical, physical, layout, and other tools can operate together. The designer will manipulate controls invoking automated design tools with “autopilot” aids in a cockpit setting.

Finally, it should be noted that the *Design* chapter of the ITRS, while increasingly important to the overall effort and time-to-market required to produce products and profit, is fundamentally different from the technology roadmap chapters, especially regarding the alignment of required design techniques to technology nodes. Technology advances tend to occur in discrete increments when all needed elements can be put in place in manufacturing. However, improvements in design techniques or the design process can normally be put in place when developed and can lead to improved productivity, or quality, or overall cost reduction immediately.



*HW/SW—hardware/software*

*Figure 6 Required Changes in Design System Architecture*

Table of Contents Previous Next Last

## DIFFICULT CHALLENGES

Design challenges come about because of increases in the complexity of the component being designed; the technology in which it is being designed; and the tools used to design it. The system size, in terms of number of transistors and complexity of embedded software, is increasing exponentially. Additionally, the number of factors that designers and design tools must take into account as system heterogeneity increases and feature size decreases also is going up. Figure 7 indicates a range of factors that must be considered.

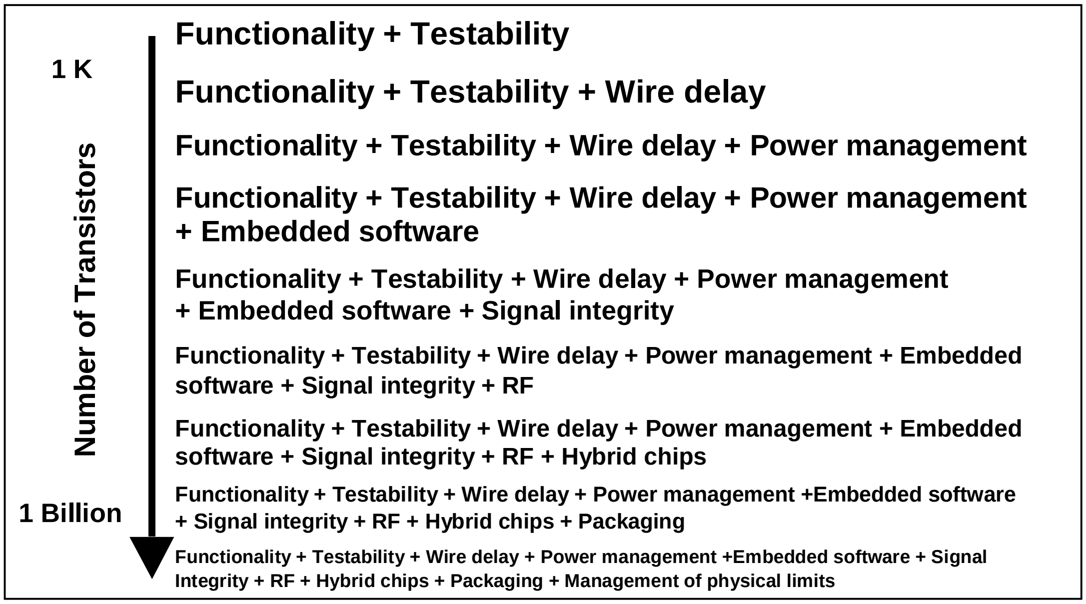


*Figure 7 Superexpoenetially Increasing Design Complexity*

Table 13, describing the difficult challenges in design, is grouped according to the drivers giving rise to five different complexity scales, and is ordered by immediacy of need. *Silicon complexity* is brought about by technology advances—smaller feature sizes not only allow more devices, but bring effects previously able to be ignored to prominence. *System complexity* has enabled this increase in capacity. The driver for system complexity is consumer demand for applications with increasing function leading to shorter time-to-market and lower cost. *Design procedure complexity* increases because of both technology and system factors. And the *verification and analysis*, and *test and testability*, of chips, packages, and entire products also become more complex.

## TECHNOLOGY REQUIREMENTS

Table 14 illustrates the need to greatly increase designer productivity if the advances of technology are to be realized in actual designs. The solutions for achieving these numbers are greatly dependent on the application being designed—high degrees of design reuse, reliance on software and memory on-chip, and improved design tools are all attempts at bridging the productivity gap. The assumptions in the table are for design cycles to decrease slightly, and design teams to remain fixed at about 50 designers for ASICs and 300 designers for microprocessors. Increasing the team sizes beyond this is difficult because of the increased human communication problems that are introduced without methodologies to address them.

*Table 13 Design Difficult Challenges*

| *Five Difficult Challenges*<br>*≥ 100 nm / Through 2005* | *Summary of Issues* |
|---|---|
| Silicon complexity | Large numbers of interacting devices and interconnects<br>Impact of signal integrity, noise, reliability, manufacturability<br>Power and current management; voltage scaling<br>Need for new logic families to meet performance challenges<br>Atomic-scale effects<br>Alternative technologies (such as copper, low κ dielectric, SOI) |
| System complexity | Embedded software as a key design problem<br>System-on-a-chip design with a diversity of design styles (including analog, mixed-signal, RF, MEMS, electro-optical)<br>Increased system and function size<br>Use of open systems and incorporation into global networks<br>Integrated passive components |
| Design procedure complexity | Convergence and predictability of design procedure<br>Core-based, IP-reused designs and standards for integration<br>Large, collaborative, multi-skilled, geographically distributed teams<br>Interacting design levels with multiple, complex design constraints<br>Specification and estimation needed at all levels<br>Technology remapping or migration to maintain productivity |
| Verification and analysis complexity | Formal methods for system-level verification<br>System-on-a-Chip specification<br>Early high-level timing verification<br>Core-based design verification (including analog/mixed-signal)<br>Verification of heterogeneous systems (including mixed-signal, MEMS) |
| Test/testability complexity | Quality and yield impact due to test equipment limits<br>Test of core-based designs from multiple sources (including analog, RF)<br>Difficulty of at-speed test with increased clock frequencies<br>Signal integrity testability |
| *Five Additional Difficult Challenges*<br>*< 100 nm / Beyond 2005* | |
| Silicon complexity | Uncertainty due to manufacturing variability<br>Uncertainty in fundamental chip parameters (such as signal skew)<br>Design with novel devices (multi-threshold, 3D layout, SOI)<br>Soft errors |
| System complexity | Total system integration including new integrated technologies (such as MEMS, electro-optical, electro-chemical, electro-biological)<br>Design techniques for fault tolerance<br>Embedded software and on-chip operating system issues |
| Design procedure complexity | True one-pass design process supporting incremental and partial design specification<br>Integration of design process with manufacturing to address reliability and yield |
| Verification and analysis complexity | Physical verification for novel interconnects (optical, RF, 3D) at high frequency<br>Verification for novel devices (nanotube, molecular, chemical) |
| Test/testability complexity | Dependence on self-test solutions for SoC (RF, analog)<br>System test (including MEMS and electro-optical components) |

*κ—dielectric constant* *SOI—silicon on insulator* *IP—intellectual property*

Table of Contents Previous Next Last

### *Table 14a Design Technology Requirements—Near Term*

| *YEAR*<br>*TECHNOLOGY NODE* | *1999*<br>*180 nm* | *2000* | *2001* | *2002*<br>*130 nm* | *2003* | *2004* | *2005*<br>*100 nm* |
|---|---|---|---|---|---|---|---|
| MPU new design cycle (months) | **36** | **36** | **36** | **32** | **32** | **32** | **30** |
| MPU transistors per designer-month (300-person team) (thousand) | **2** | **3** | **4** | **7** | **10** | **15** | **20** |
| ASIC new design cycle (months) | **12** | **12** | **12** | **12** | **12** | **12** | **12** |
| ASIC transistors per designer-month (50-person team) (million) | **0.3** | **0.4** | **0.5** | **0.7** | **1.0** | **1.3** | **1.8** |
| Portion of verification by formal methods | **15%** | **15%** | **15%** | **20%** | **20%** | **20%** | **30%** |
| Portion of test covered by BIST | **20%** | **20%** | **20%** | **30%** | **30%** | **30%** | **40%** |

### *Table 14b Design Technology Requirements—Long Term*

| *YEAR*<br>*TECHNOLOGY NODE* | *2008*<br>*70 nm* | *2011*<br>*50 nm* | *2014*<br>*35 nm* |
|---|---|---|---|
| MPU new design cycle (months) | **28** | **26** | **24** |
| MPU transistors per designer-month (300-person team) (thousand) | **65** | **200** | **600** |
| ASIC new design cycle (months) | **11** | **10** | **9** |
| ASIC transistors per designer-month (50-person team) (million) | **5** | **13** | **36** |
| Portion of verification by formal methods | **40%** | **50%** | **60%** |
| Portion of test covered by BIST | **50%** | **60%** | **70%** |

*Solutions Exist* ⬜ *Solutions Being Pursued* 🟨 *No Known Solutions* 🟥

## SYSTEM-LEVEL DESIGN

Design is bounded by two endpoints, (1) at the highest level is system design, where architecture must implement the functions needed for the end product, and (2) at the lowest level are primitive building blocks—logic gates, amplifiers—controlled largely by the underlying device technology. Thus, overall system design is the creation of complex architectures using primitive building blocks facilitated by computer aided design (CAD) automation tools. This conceptual picture is significant, because system performance is a strong function of primitive circuit building blocks, and these are rapidly reaching their performance and functionality limits as technology scales into the deep submicron regime. In addition to impending technology limits, system complexity is also growing rapidly as systems become more capable and include more complex algorithms and software as part of large distributed systems. All these factors have serious consequences for system design.

System design has two definitions. It is the development of new architectures and algorithms for complex applications in the context of limited technology capabilities (design). Also, it is the process of creating a system, using the CAD methodology and tools that assist the designer in assuring that all constraints are met in a feasible, near-optimal implementation (design automation).

The “design” portion of system design pertains to the following:

- exploration of the design space at a high level to compose an overall architecture—including *circuit blocks*, *software*, and *communication* that implement the *algorithms* that accomplish the system function.
- exploitation of the opportunities/limitations of available silicon and packaging technology options and decomposition of the system into a manageable hierarchy of smaller subsystems.

The “design automation” portion of system design

- encompasses specification development, requirements and constraint capture, algorithm tradeoff, initial architecture partition and budgeting, including high-level block and wire floorplanning.
- produces an initial specification of partitioned, smaller blocks and their respective interconnect elements. Each block or interconnect element is budgeted and floorplanned on a global basis.

The boundary between pure digital design and circuit design is changing. For example, placement and interconnect delays (which can only come from physical design) must be considered when trying to bring the logic design to timing convergence. “Logic synthesis” must become more focused on cell selection and optimization in light of these physical constraints.

Similarly, the boundary between systems and digital design is blurring. Timing boundaries and discrete logic specification should appear earlier in the design cycle. For example, logic design can not turn a general algorithm into an optimized design where internal bit widths meet a budgeted error or accuracy goal. Instead, global accuracy and error must be traded off. Also, protocols and methods to specify the transference of data both spatially and temporally have a big impact on the algorithm implementation.

### *DIFFICULT CHALLENGES IN SYSTEMS DESIGN ( ≥100 nm, THROUGH 2005 )*

Down to the 100 nm technology node, system design needs the following capabilities and developments:

- The fastest buffered interconnect for a long wire can be over 100× the switching time of an individual gate. Given expected global clock rates approaching 3 GHz, it will be difficult to achieve synchronous, on-chip operation without introducing multi-cycle latencies in the interconnect. System design must comprehend the timing issue more fully.
- The large currents being introduced with increasing power densities and lower voltages all lead to larger supply rail inductive noise. Synchronous systems worsen the problem by scheduling the power switch surges around regular time periods. Thus, supply rail design to reduce effects such as voltage drop or current surges are required early in the design process.
- Estimation techniques will be necessary for supply distribution (such as voltage drop, ground bounce) in light of large current swings (> 500 A) and for clock distribution in light of power surges due to powering up sub-circuits, and power-up / reset condition surges.
- Process technology support for increased switching is not keeping up with the trend in clock rate increase. Thus, design techniques such as reducing the number of logic levels between clocked registers allow the current trend for clock rate increase. Careful consideration of single versus multi-cycle paths and circuits will become more important early in the design process. Specifically, locally synchronous but globally asynchronous design techniques will need to be supported by the tools.
- Floor planning and constraint budget planning will be required especially in light of hard IP blocks and dominance of interconnect over transistor switching time.
- New estimation techniques for interconnect and protocol delays between blocks will need to take into account signal noise (such as coupling), thermal gradients due to power disparities, and other effects. Major interconnect busses and lines will need to be designed early in the floor plan and block partition process.
- Behavioral synthesis has had limited applicability due to poor estimation of the final area or time tradeoff between the various alternatives. This estimation before operator selection and scheduling must be improved to enable this capability. Techniques to aid the estimation are a large selection of pre-defined library (IP) elements and quick synthesis techniques that bypass detailed, final optimization. These same estimation techniques will aid other forms of behavioral synthesis (protocol and datapath synthesis, for example) that in turn aid the system designer in early block partitioning tradeoffs and analysis.
- Embedded software will play a greater role in ASIC and SoC designs, bringing with it the following increased design challenges: hardware/software co-design and design partitioning and tradeoff analysis; software verification and its complexity; hardware/software skills and design team makeup; mixed hardware/software reliability and test; software reuse and software IP.

Table of Contents Previous Next Last

### *DIFFICULT CHALLENGES IN SYSTEMS DESIGN ( <100 nm, BEYOND 2005)*

- An inflection point is reached at around 100 nm design and below. More automated detailed design cannot occur until unbuffered, predictable timing synthesis of small blocks is possible. For 100 nm technology, this implies irregular logic blocks must be no larger than about 100K gates. For an ASIC, this implies that approximately 200 to 400 such blocks must be specified and planned before detailed design can start. This is too much complexity for the system design phase and therefore requires the introduction of a new "middle" layer of design between detailed logic design and system specification/partitioning. Defining the entry into and exit out of this middle layer of the design process will be required.
- Power densities at 100 nm and below are predicted to surpass the ability for all circuits to switch simultaneously. Therefore currents would exceed 1000 A and would be switching within a clock cycle from 0 to 1000 A. Even carefully designed supply rails cannot handle this current without severe impacts on voltage drop and other effects. Techniques to manage the instantaneous power needs and average out the power requirements will need to be put in place. Tools to take this into account during algorithm development will be required to meet optimum performance and power tradeoff points.
- With clock speeds possible exceeding 5 GHz, and across-chip communication taking upwards of 5 to 20 clock cycles, an approach is needed to building a hierarchy of clock speeds with locally synchronous and globally asynchronous interconnect. Tools to handle asynchronous, multi-cycle interconnect as well as locally synchronous, high-performance near-neighbor communication are needed. Abilities to design with two or more clock rates that vary by two orders of magnitude or more within the same block are needed. Communicating asynchronous regions (asynchronously coupled blocks) and multi-cycle interconnect paths will dominate.
- Due to the high cost of "system" communication across the chip, globally accessible register and buffer structures will no longer be possible. Therefore, the cost of communication with various forms of memory will become desperate depending on which block is involved. Design systems that allow for specification and design of distributed memory architectures and the tradeoff with on-chip busses will be required. This makes a major impact on the algorithm tradeoff tool also.
- The prioritization of design tradeoffs must be determined for the new technology beyond 100 nm rules. The biggest impacts of an integrated circuit implementation of such a complex system must be identified and characterized. The impacts can then be built into the tools to allow support in the system design tradeoffs, as early as specification capture if needed. For example, if an algorithm or specification requires random access to twenty 1 Gbit frame buffers, then the cost of implementing a 20 Gbit random access memory must be taken into account early and tradeoffs made to minimize the impact or maximize the specification on a target architecture.
- Multi-tasked, multi-rate systems will be the norm for single chip, encapsulated systems that consume so many transistors. Methods of specifying, verifying, manipulating, and trading-off the various task implementations and their requirements for different response times and data processing rates will be required. Most likely, such multi-tasked systems will need to share resources as seen in multi-media processing of today. Everything from interconnect busses, buffer memory and functional units are shared within complex tasks; each with different data processing requirement rates and response times.
- Additional algebraic or other discrete mathematically based formalisms to cover the various application domains will need to be developed. It is only with more formal specification forms that automated tools can be developed to capture, verify, and then optimize a design. Digital Signal Processing theory, when mapped to the various models of computation such as dynamic data flow, or synchronous data flow, is an example of a new algebra for specifying and manipulating systems in their domain and at a higher abstraction level.
- Even in large electronic systems design today, it is difficult to imagine a completely new design that does not reuse some previous components or capability. Incremental design where less than 25% of the design is new will become more the norm. Tool environments to support such incremental design as opposed to expecting 100% new system specification capture and manipulation need to be developed.

- The ability to insert robustness automatically into the design will become a priority as the systems become too large to test functionally at manufacturing exit. The automatic introduction of techniques such as redundant logic for fault tolerance is needed.

## SYNTHESIS AND LOGIC DESIGN

Logic design is the process of mapping the register transfer level (RTL) hardware-specific outputs from the system design phase into a gate-level representation suitable for input to the physical design stage. At this design stage, the RTL represents only digital hardware functionality, which has previously been “scheduled” to specific clock edges. For reasons of design productivity, most technology mapping and logic optimization improvements in logic design require automation within logic synthesis tools and algorithms.

### DIFFICULT CHALLENGES IN LOGIC DESIGN ( ≥100 nm, THROUGH 2005 )

As the number of transistors per chip increase, the present-day problem of interconnect delay uncertainties will continue to increase. As a result, greater use of block-based and hierarchical design techniques are required at both architectural and logic levels. The physical location of various blocks to be connected will contain greater variability across-chip. This requires automated interconnect exploration and analysis of many candidate placement choices, coupled with advanced synthesis algorithms. Since all system-level operational constraints must also be satisfied, these constraints will be analyzed concurrently as an integral part of the logic design process. Advanced floorplanning tools must evolve to a larger and more central role, eventually becoming the designer's primary “command and control” center for both logical and physical views of the design.

Even at reduced operating voltages, the predicted growth rate of digital functionality per chip will require substantially more power than supported by known thermal packaging options or anticipated battery life enhancements. As a result, architectural and logic-level power management strategies must be integrated—and automated—within the synthesis environment. However, power estimation algorithms will need some means to handle large design scale capacities and heterogeneous digital/analog/software tradeoffs, prior to the existence of simulation vectors. Long simulation run time for power characterization must be replaced by sophisticated conversion to statistical forms that preserve more relative accuracy for earlier estimation and optimization. The algorithms will need to apply lessons learned from prior synthesis and physical design passes to statistically improve accuracy for re-synthesis in subsequent passes.

Signal integrity and reliability concerns will grow to unmanageable levels if not addressed until placement and routing. New techniques must be developed to support design methodologies that avoid obvious or likely violations as part of the synthesis process. This implies new requirements for concurrent logical and physical design, as synthesis becomes more physics-aware and applies context-sensitive rules to avoid downstream reliability concerns for electromigration, voltage drop, channel hot carrier-induced slew rate degradation, and probable antenna violations. Fundamental signal integrity issues such as crosstalk immunity and noise tolerance must become a tradeoff equal to timing, area, and power.

The increasing integration of many different functions and design styles on silicon, warrants the need for better resolution to differentiate among these styles and apply specialized algorithms suited to optimize for a particular style. However, this need for specialization also requires new algorithms that are capable of recognizing these different styles in the input description and then merging the results. This may include specialized synthesis for datapaths and memories that perform localized silicon compilation, where detailed routing is included for those specific design styles.

Logic design trends indicate increased optimization for power and timing using multiple on-chip voltages and scaled voltage operation. It will be necessary for future synthesis environments to optimally consider, tradeoff, and select not only the best library elements, but also which voltage level is to be applied to satisfy design goals. This implies additional enhancements to library characterization flows.

As new materials and manufacturing processes are adopted, subsequent changes will be required to the synthesis environment. Copper-based interconnects offer faster performance, but if mixed with aluminum at different metal levels could create a major challenge requiring more elaborate integration with detailed

Table of Contents Previous Next Last

routing to fully resolve. Similarly, low κ dielectrics between layers can improve circuit speeds, but also present a challenge for estimating interconnect performance and critical paths at the logic design stage. Silicon-on-insulator offers yet another process technology enhancement with very similar challenges. The challenges extend beyond library characterization, reaching into detailed optimization heuristics and algorithms.

While most digital logic functionality will continue to be input at the RTL level, the heterogeneous nature of silicon-based applications of 10–50M gates demand that neighboring blocks of analog or RF be explicitly considered. Synthesis will need to adopt special design mapping and/or optimization rules to help ensure the overall integrity of the physically co-located system. Similarly, reused hard IP blocks placed adjacent to new logic could also need special treatment, especially under an assumption that synthesis, placement, and (some) routing algorithms merge.

Present-day concerns over testability will become insurmountable at future logic densities without more automated and thorough test insertion strategies.

While current approaches add test features as an afterthought, future test requirements (as specified during the system design phase) must be satisfied in a more routinely automated fashion. Synthesis algorithms should be developed that are capable of automatically inserting test structures while satisfying all other timing, power, density, reliability, and signal integrity constraints. Test insertion must include self-test logic, and leverage the use of on-chip processors (possibly implemented with supplemental firmware code).

### *DIFFICULT CHALLENGES IN LOGIC DESIGN (<100 nm / BEYOND 2005)*

As silicon manufacturing technology progresses below 100 nm, additional logic design considerations must be addressed. Due to anticipated low voltage transistor operation, extremely tight thresholds and margins, extreme power management problems, and multi-gigaHertz performance requirements, static CMOS may be complemented or replaced with alternative logic families. Logic synthesis must support technology mapping and optimization algorithms for new logic styles required (for example, partial-swing, multi-threshold, or clock-delayed domino cells).

In the sub-100 nm regime, where hundreds of millions of transistors and their interconnect must operate without fail, synthesis tools will be required to automatically integrate fault-tolerant logic design techniques. Fault-tolerant features include hardware redundancy, re-programmable interconnect, and re-configurable control logic. All of these will be expected to operate under the control of on-chip self-test logic.

Heterogeneous system-on-a-chip designs will integrate neighboring analog, mixed-signal, and RF circuits within a single die. Special attention will be required by the digital logic to minimize the likelihood of problematic signal integrity issues. Robust design requirements must be handled on the digital side, yet design productivity needs prevent this from being feasible without automation. These concerns include crosstalk, switching noise, substrate noise, electromagnetic interference, and voltage drop issues. As MEMS, electro-chemical, and electro-biological technologies find their way into silicon, fault tolerance may require additional avoidance techniques.

Given the enormous amount of logic to be generated under such intimidating system and silicon constraints, design reuse and incremental re-design become mandatory. Incremental synthesis that merges both new design features with existing synthesized logic, while preserving most of the existing design realization, will be a grand challenge. Beyond 2005, existing designs that may include custom designed regions should be capable of participating in incremental synthesis and optimization.

Below 100 nm, soft errors are predicted to be frequent enough to severely impact both semiconductor yields and field-level product reliability. Protection from these soft errors may require advanced and dedicated tools to analyze their statistical likelihood by cell, transistor type, and placement, and performed out of cycle as a characterization step. Then automated methods must be discovered for modifying and adjusting the logical and physical design to prevent these errors, without violating any other design constraint within the chip.

## VERIFICATION AND ANALYSIS

Verification and analysis continue to be serious bottlenecks for the timely design of complex ICs and SoCs. It is estimated that verification requires over one half of the design resources, human and computer, and slows the "time-to-profit" of new products substantially (Figure 8). Research contributions to this area, therefore, represent the greatest potential improvement in development of new products.

Accuracy of verification relies on accuracy of the model used. Speed of the verification simulation is a function of the simulation application software; the hardware platform on which the simulation is run; and the complexity of the model. Since simulation speed is critical to "time-to-profit," good tradeoffs must be made between simulation speed and accuracy. Methodologies need to be developed that include various levels of simulation accuracy as the design progresses.

Typically designs are modeled at many different levels of abstraction including system, architectural, micro-architectural, register transfer, gate, switch, and layout. For models at the register transfer (RT) level and below, many verification tools and methodologies exist, but they are only marginally adequate—due to reduced device feature size, increased clock speeds of synchronous designs and increased dominance of interconnect over device area—not to mention mixed-signal issues. For efficient core reuse, characterization of a block with respect to functionality, timing, and electrical properties becomes increasingly important. System-level design initiatives and hardware/software co-design initiatives are just beginning to produce commercially usable tools that solve the difficult verification and analysis problem. Embedded cores, system-on-a-chip complexity, integration of digital and analog, and the integration of devices such as sensors or actuators reinforce the analysis and verification challenges. For digital designs, the capabilities for "in circuit" emulation, simulation technologies and formal verification are far from optimum even for today's IC complexities. For mixed-signal (digital and analog) chips, there are some tools for analysis and verification, but much improvement is needed.

### SYSTEM LEVEL ARCHITECTURAL VERIFICATION AND ANALYSIS

For designs requiring novel architectures, a top-down design approach is needed with the overall architecture being described by a form of high-level behavioral language. The designer will use this high-level language to exactly describe all the inputs and outputs of the chip and the major blocks within the chip. To do this, the designer must pay close attention to block interface specification. The better the designer does with this level of verification, the smoother the rest of the design will go and the easier subsequent derivative products can be developed with the reusable blocks from the platform design. Better tools are needed for top-level architectural definition that can be used by designers to more quickly and accurately describe complex SoCs by the tools themselves comprehending the language used by various types of systems (networking, telecom, computer, and others). These tools must also interface with the hardware description language (HDL) tools at the next level down to facilitate rapid verification.

Verification at the system architectural level and below requires extensive use of many cooperating abstractions that must be heterogeneous. Currently, the menu of available abstractions is inadequate; methods for defining and manipulating abstract views of a system are almost nonexistent; and the problem of maintaining consistency between several abstract views of the same system, especially as the design evolves, is unsolved.

### DIGITAL VERIFICATION TOOLS

Current industrial practice relies mostly on HDL or gate-level simulation with manually generated test cases or pseudo-random inputs, along with timing analyzers and other simple rule checkers. Over the past decade, advances in formal verification techniques have resulted in verification tools that exceed the capabilities of traditional, simulation-based approaches in their ability to detect and diagnose difficult design errors. Much of this work has been based on exploiting regularity in homogeneous design abstractions, such as via symbolic model checking. Advances in equivalence checking by formal verification have greatly reduced verification time at the gate or netlist logic level. Despite these advances, existing formal verification tools do not scale to handle the size and complexity of gigascale systems.

Model checking is currently employed by a number of companies but has not yet become the mainstream methodology. Simulation-based approaches cover a small percentage of the possible inputs, but current model checkers must either be applied to a small fragment of a design or to a highly abstracted description. Theorem proving has mostly been used in research environments. Symbolic simulation is a different formal verification paradigm that has had some success. Symbolic simulation can be used with various symbolic representations, including binary decision diagrams (BDDs) or logical expressions. Symbolic simulation is promising when dealing with designs that are more data-oriented.

### *HARDWARE/SOFTWARE SYSTEM LEVEL VERIFICATION*

The anticipated heterogeneous components of future gigascale systems require specialized, but cooperating, verification methods. Furthermore, new verification methodologies must be embraced to partition the verification challenge into manageable components. Components may include analog, RF, phase-locked loops, and digital fabrics. The new digital fabrics will include many different design styles for a wide variety of applications, such as digital signal processor (DSP), embedded processors, protocol engines, memory subsystems, and video streams. In addition, sensor and actuator components must interact with the real world and verifying their correctness depends on modeling the physical environment.

New models, notations, and verification techniques for continuous behavior systems must be developed, including analog, RF, new interface fabrics, and embedded controllers. Rather than trying to create a single mathematical model and verification technique that covers the complete range of system designs, heterogeneity presents an opportunity to use domain-specific specification and verification techniques. These can leverage what is known about a particular application or design style to increase the effectiveness of verification by orders of magnitude. In some areas, such as analog, RF, and physical-environment modeling, new theories must be devised to apply any systematic reasoning process. For this to happen, new circuit fabrics must be developed with verification in mind. Simply designing heterogeneous fabrics without consideration for how they will be verified will likely result in large systems that cannot properly be verified. Similarly, formalizing and checking components that act as signal converters between subsystems based on different paradigms, such as analog-to-digital signal levels or asynchronous-to-synchronous timing, is not possible without careful construction of the associated fabrics.

Large-scale system verification is best performed by a heterogeneous collection of tools, allowing tradeoffs between the level of assurance, the tool capacity, and the tool performance. These tools should be integrated as much as possible, using compatible representations for system elements, interfaces, and properties to be verified. The electronic design automation (EDA) industry will continue its development of methods to achieve faster evaluation performance via efficient simulation algorithms and hardware emulation. Well-disciplined techniques for analyzing system correctness are needed. This will include a combination of improved and expanded formal verification techniques, as well as “semi-formal” techniques that occupy intermediate points between simulation and formal verification. These multiple techniques need to be integrated into a common verification framework.

As embedded processors become widespread, the correctness of software is part of the overall system verification. To date, formal methods for software verification have not had widespread industry impact due to limitations of existing tools and methodologies. Formal verification research must also focus on the forms of software found in embedded systems, such as schedulers, device drivers, and protocol converters. Such low-level software can be viewed as extensions of hardware and verified by similar means, but using higher levels of abstraction in modeling the storage and manipulation of data. For other forms of software, semi-formal and testing-based approaches that have been developed in the software engineering community must be leveraged. Methods for software verification will become even more important as the ever-decreasing product life cycle creates the need for more and more software re-configurable circuits.

Hardware emulation via massive PC boards with arrays of configurable gates has proven to be a useful form of verification for complex logic systems. The major difficulty with these methods has been the partitioning problem and getting signals that are normally on-chip to move between chips of gate arrays without disrupting the desired timing. Another difficulty is that the fastest emulators are always a generation or so behind the chip being designed and are not fast enough to represent the new design. Emulation is better suited to pure digital designs such as mixed-signal designs that are not easily verified. Emulators do provide a good means of co-verification of the hardware with the software.

### *TIMING VERIFICATION*

Even for seemingly generic digital components, timing verification faces challenging obstacles. Timing verification is generally performed at the end of the design cycle when the layout has been completed and parasitics can be accurately estimated. Research in timing verification over the last decade has yielded solutions to several long-standing problems, including verification of circuits using multiphase clocking and level-sensitive latching, and the ability to automatically prune away logically unsensitizable, or false, signal paths. However, deep submicron technologies present new problems, such as those due to dominant coupling capacitance, whereby coupling to adjacent wiring can cause delays to vary by as much as 300–400%, depending on the switching state of adjacent wires. This forces functional and timing verification to be done concurrently–at minimum, a quasi-static performance verification. New methodologies for incremental timing verification of digital systems that include dominant interconnect coupling must be developed. Another example of new problems brought about by technology advances is that SOI designs exhibit a “memory” effect due to charge storage in floating wells, such that timing errors arise due to threshold voltage changes. These effects in this case are highly pattern dependent and activity dependent, causing inaccuracies when using static timing verification.

Of further concern for deep submicron are the variations in performance due to manufacturing fluctuations. Statistical verification methodologies that will serve as enabling technologies for reliable design and test must therefore be developed. These methodologies will be incremental in enabling new synthesis and physical design capabilities and must also support abstractions across all levels of design. Therefore, semantic information such as the distinction between data and control signals, or between an opcode, an address, or an operating mode, must be exploited to manage timing verification complexity without compromising its accuracy.

### *POWER ANALYSIS*

Better methods of accurate power prediction and analysis are more important than ever. Power prediction must be done as early and as accurately as possible in the design cycle, if possible at the architectural design phase. More and more applications have power budgets that cannot be exceeded for one reason or another, usually portability or reliability for a given package or system. Inaccurate power estimates often result in rework of the design architecture, logic, timing or sometimes costly rework of an analog block and results in delays in the design.

### *TESTABILITY ANALYSIS*

Testability must be designed in at the earliest possible time. Testability planning must be done during the architectural design phase and any special testability cells inserted in the design early so that the effects on timing as well as test coverage may be analyzed and assessed. The objective is to have high fault coverage with minimal adverse affects on circuit performance, area, or power. Methodologies must ensure that simulations and analysis done earlier in the design do not have to be redone after test insertion.

### *NOISE ANALYSIS*

One of the biggest problems in design of today’s high-performance logic or mixed-signal chips is that of noise analysis and minimization. This is due largely to the complexity of the noise problem that includes parasitic capacitance and inductance of both on-chip and package interconnects, interconnect crosstalk, and noise injection via the substrate. Tools are needed that will quickly and accurately tell the designer when the signal-to-noise limits are exceeded in the digital and the analog domain.

### *ANALOG CIRCUIT VERIFICATION*

Some analog circuits such as an RF amplifier may have as few as three or four transistors, however the verification of such circuits over voltage and temperature may require tremendous simulation times due to precise modeling of all the passive elements, including parasitic elements. As more complexity is added, verification of large analog blocks can take many weeks of simulation time. Ever-faster simulators are the

Table of Contents Previous Next Last

historical solution. New solutions need to include statistical techniques to eliminate many voltage-temperature simulations as well as better compact models that allow faster simulations without sacrificing accuracy.

### *PHYSICAL DESIGN VERIFICATION*

The physical design verification phase is also a critical step in timing verification as it is only during this phase that exact timing information can be extracted. As we move into deep sub-micron (DSM) designs, verification of the physical designs will take longer due to the added complexity of the interconnect parasitics to include R, C, and now L. Currently only a moderate contributor to design cycle time, the verification of the physical design can involve significant resources and is not always a smooth process. Designers often use CAD tools from multiple sources during this phase because one suppliers's tool may do a much better job than another. These different tools do not have standard interfaces to each other, which causes many extra days of program delay. Standard interfaces must be developed to enable tools from different suppliers to communicate smoothly.

### *DESIGN FOR VERIFICATION*

Over the next decade one of the key promises of component-oriented design is its potential to leverage formal verification methods. Disciplined, component-oriented design requires partitioning the system into subsystems of manageable complexity communicating through well-defined and fully characterized interfaces. Verification can exploit this modularity, verifying the interfaces, components, and overall system at appropriate levels of abstraction. To realize this potential, new formal and semi-formal verification techniques must be developed and integrated.

Parameterized modules, where details such as word size, capacities, and detailed functionality can be determined during instantiation will be very important in component-based designs. Methods for verifying the correctness of a parameterized module before instantiation will be highly advantageous, because verification will only need to be done once by the designer, rather than by each customer who instantiates the module. Verification of parameterized components can be done by abstractions that reduce a module of size $n$ to a fixed-size verification problem; by special decision procedures; or by the use of induction in a general-purpose theorem prover. Research results have been published on such techniques, but current capabilities are limited and require great effort and expertise on the part of the user.

Verification of gigascale systems must exploit many levels of hierarchy and abstraction. It is widely recognized that abstraction is the key to verifying large systems, but most uses of abstraction in current verification methods are ad hoc and extremely labor intensive. In applying existing formal verification techniques, the most time-consuming activity and the one requiring the greatest expertise is the construction of abstractions and simplifications that are coarse enough to be checkable yet fine enough to reveal useful information and be logically correct. Methods must be devised to generate suitable abstractions in a more automated way. Generating these abstractions will be greatly aided by use of a more systematic, top-down design methodology. Verification can be coupled into the layered abstractions used during the system design process.

### *INTEGRATED VERIFICATION ENVIRONMENT*

The current practice in industry is to use a diverse collection of verification tools including multiple forms of simulators, timing analyzers, and others. Significant effort is expended translating between incompatible circuit formats and setting up environments that will automatically invoke appropriate tools as a design evolves. Efforts to develop industry standards for tool integration have had limited success. As we add in new forms of tools, including ones that comprehensively analyze different facets of a design and that view the system at different levels of abstraction, tool integration becomes more critical and challenging. In addition to compatible formats, we need methods to ensure consistency of the different representations as well as to evaluate the completeness of the overall verification methodology.

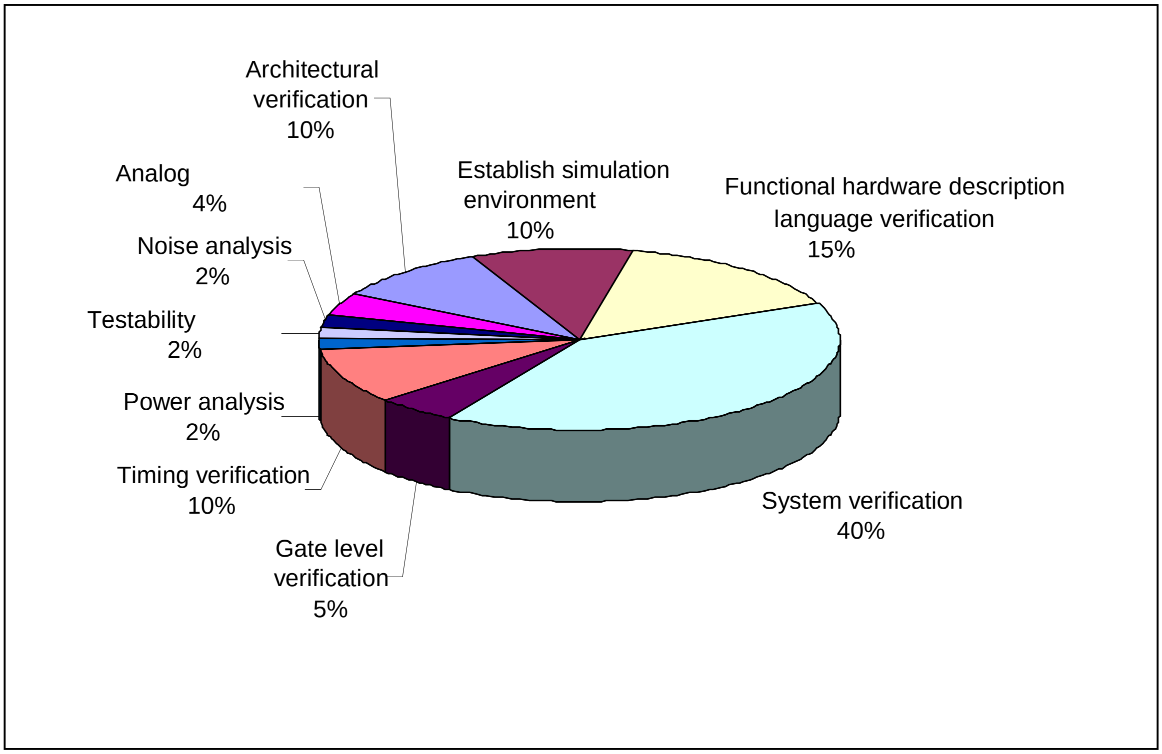


*Figure 8 Allocation of the >50% of Design Effort Devoted to Verification*

### SUMMARY: ISSUES AND NEEDS

The semiconductor industry needs mathematically precise models of system and architectures, automated analysis tools that handle the complexity now routine on a single chip, and automated verification tools that can guarantee compliance of a model with a lower level model of the same design. Increasing complexity demands more automation of the design process. New ideas and more automated tools in formal verification, simulation, emulation, and hybrid approaches are needed now to handle the synthesis, analysis and verification problem.

## PHYSICAL DESIGN

Physical design deals with aspects of the chip implementation related to the spatial embedding of devices and interconnects. With advancing technology, it is harder to estimate and abstract the effects of physics and embedding on achievable design quality, where quality is measured by the key objectives: timing, power, signal integrity, reliability, and manufacturability. Thus, physical design will continue to become more closely linked with system-level design and logic-level design. In addition, physical design requires a new manufacturing interface that enables close ties to circuit implementation and the underlying process technology. The surrounding design methodology, as depicted in Figure 6, must support tight coupling of

1. (incremental) modeling, analysis and abstraction of the key objectives with appropriate accuracy,
2. (incremental) design synthesis and optimization of these abstracted objectives, and
3. (incremental) (re-)specification of the design at multiple levels of representation.

These needs apply equally to the design of digital, mixed-signal, and analog systems.

### DIFFICULT CHALLENGES BEFORE 2005

*Silicon Complexity*—Increasing silicon complexity creates challenges in power management, signal integrity, and layout complexity. New tools will be required for power and current management. This will include both

Table of Contents Previous Next Last

design tools for power delivery as well as analysis tools for IR drop and active power dissipation. Additionally, power density issues will need continuous attention to keep silicon within its operating temperature window. Finally, power dissipation need higher priority while resolving the different cost functions involved at all major junctures in the design flow.

Efforts to ensure acceptable signal integrity will need to address noise as well as coupling effects. Interconnects and their dynamic physical interactions become a major source of uncertainty in achieving signal integrity closure. Tools must comprehend and manage these interactions without compromising the quality of the embedding, such as routability.

Balancing the different cost functions while converging to an acceptable layout solution is another key challenge. Since physical design will increasingly entail large-scale interacting multi-level, multi-objective global optimizations, new paradigms for optimization must be developed, such as constraint-dominated global optimization, incremental optimization, resource-bounded optimization, and optimization with partial or probabilistic design information.

*System complexity*—Increasing system complexity will have its unique set of challenges for physical design due to diverse design styles and the industry trend towards system level integration. This will lend itself to layout tools comprehending critical signal isolation, IP block placement, and interconnect considerations allowing the handling of mixed design styles and types (analog, digital, mixed, RF). Distinctions between high-frequency digital design and analog/mixed-signal design will blur. Controllability and observability will need to be understood at layout as testing and verification responsibilities are dispersed into the physical design flow.

*Design procedure complexity*—Design procedure complexity challenges with IP block insertion and reuse will affect productivity unless addressed. This will require new tools for reuse-driven block characterization and abstraction, design data management, and flexible functional/physical hierarchy management. Process technology mapping to aid reuse across process generations will become necessary to maintain productivity improvement.

Increasing clock frequencies and large switching currents imply a need for tools that unify the synthesis of power, clock and test distribution, along with reliability- and noise-driven layout. As reflected in the new design system architecture of Figure 6, data models, repositories and interfaces must be re-engineered for “unifications” such as concurrent logical-physical design, or tight analysis-synthesis loops. Manufacturing variability implies a need for statistical design and design centering tools, such as timing analysis and physical verification tools that comprehend parasitics, delays and geometries that are parameterized by distributions.

Manufacturing variability (and the proliferation of new materials and processes) also requires a new design-manufacturing interface that supplies design rules and other process abstractions to layout. Key drivers include planarized multilayer interconnect processes and subwavelength optical lithography, both of which affect physical verification and the nature of appropriate handoffs to manufacturing. Physical design may also drive future process development.

*Verification complexity*—Verification complexity increases as system level integration pervades integrated circuits. Additionally, as feature sizes shrink and frequencies increase, the dominant parasitics take on new forms.

One of the challenges that needs addressing is functional and performance verification for both the IP block and it interface connections. Tools must verify both functionality and timing of the individual blocks and their sum across different design styles and design types.

Timing verification must comprehend the physical models and dominant parasitics for both device and interconnect at these small feature sizes and high frequencies. Dynamic timing verification will replace current static timing tools. Additionally, modeling will need to comprehend minimal guard banding to allow a solution space to exist.

*Testing complexity*—As design and manufacturing processes evolve to address challenges of DSM, economics will drive test considerations into all aspects of the design process. Circuit implementations will move away from standard static CMOS and include dynamic logic, asynchronous circuits and will include mixed analog and RF elements. These process and circuit complexity drivers will impact test by increasing effects of noise, crosstalk, and soft error. To address these challenges, increased reliance on Defect Based Testing will require that physical design tools extract fault locations for new types of defects. The drive to rapid production ramps will require that tools be developed to support rapid diagnosis of failures on this wide variety of circuit types.

### DIFFICULT CHALLENGES BEYOND 2005

*Silicon complexity*—Silicon complexity will become compounded with the inclusion of new novel devices to comprehend in the layout activity. Extraction tools will need to identify and model both active (RF/mixed/digital) and passive (inductors) devices. Analysis tools will need to comprehend the operation of these devices as well as on-chip process variation effects.

*System complexity*—System complexity will increase with the integration of new technologies (such as MEMS, electro-optical). Tools will need to recognize the special requirements of these technologies during the physical planning and layout activity.

*Design procedure complexity*—Given the increase of novel devices and technologies, a true incremental one-pass design process, as shown in Figure 6, becomes essential to fully explore the solution space of the design.

*Verification complexity*—Complexity will increase due to introduction of more novel devices and structures, issues that were mentioned prior to 2005 are still applicable.

*Test complexity*—Layout tools will need to support BIST and built-off-chip-self-test (BOST) activities as traditional post silicon testing methodologies become prohibitive in terms of cost and complexity. These tools will recognize and add supporting circuitry automatically.

### PHYSICAL DESIGN POTENTIAL SOLUTIONS

Given these critical challenges, the roadmap of physical design needs includes the integration of the many disparate activities currently found in the design flow. Analyses and syntheses must be unified to erase barriers between today's disparate flow stages to close the analysis-synthesis loop. Constraints will drive synthesis, and estimated parasitics drive analysis, in a "construct by correction" iterative improvement process. An "analysis backplane" will support semantically consistent design views and performance analyses across multiple implementation phases. Such analyses must be increasingly informed by the functional intent of the upstream syntheses, in order to achieve performance convergence.

Layout-level and system-level design must be unified. Modeling capabilities (for both design instances and tool behaviors) must be developed to enable forward estimation-driven syntheses and design exploration. In addition to new support for hierarchy and reuse, the spanning of multiple implementation levels implies greater need for auto-interactive use models and iterative design. Design productivity requirements may also require more "structured" design methodologies and circuit fabrics, such as to simplify verification of crosstalk noise or clock skew at the expense of performance or area.

Technology mapping can be integrated with layout synthesis while timing optimization, clock synthesis and test synthesis can be integrated with placement. Such coexistence of system timing management, logic optimization, placement and routing in a single environment should mature within the next process generation.

Table of Contents Previous Next Last

### SUMMARY AND PRIORITIZATION

While the above elements of context and need are all critical, issues that absolutely must be solved in the near term in priority order include

1. design system reengineering to enable unification of analyses-syntheses and layout/logic-level design,
2. an incremental optimization design flow that will provide substantial productivity over the current cyclic flow now in use, and
3. power management. These lower priority ordered items need an existing infrastructure consisting of the higher prioritized items in place to be useful.

## TEST METHODOLOGY

### THE NEED FOR DESIGN FOR TESTABILITY

In the past, it has been common to view design-for-testability as a design option, something that could be included or not, depending on various economic parameters. Implicit in this view is the assumption that circuit designers would be able to write extensive tests of chip functionality and that automated test equipment (ATE) would be able to apply these complex functional tests in a manner that matched expected chip operation and to accurately distinguish chips that would work from those that would not. It is now apparent that each of these implicit assumptions will fail, either now or in the very near future, and that major changes are underway in test methodology and application.

The cost of designer-developed functional test vectors is increasing exponentially for successive chip generations. It is no longer feasible to commit huge manpower resources to functional test development, even for very high volume products. In addition, as designs become more complex, it becomes increasingly difficult to translate test input and output data from a block boundary, with which a designer is familiar, to the chip boundary, where the ATE interface exists. It is also difficult to extract synchronized, deterministic, chip-level data from a system simulation environment, especially when inherently asynchronous and/or nondeterministic signals such as interrupts and bus transactions are involved.

Even if functional vectors were available, ATE performance is not improving at the same rate as chip performance. Tester overall timing accuracy (OTA) makes up an increasing portion of minimum device period, and is expected to exceed device period within seven years. This means that ATE will not be able to accurately identify failing parts for at-speed functional tests, resulting in either substantial yield loss or inadequate shipped quality. Furthermore, the ATE environment is different from the system environment, usually featuring substantially higher loads per pin. Finally, high-performance ATE is very expensive. Developing the circuitry needed to precisely deliver and measure signals and power in a noise and temperature controlled environment at the speeds predicted by the roadmap is a daunting task. It is therefore likely that functional test cost per manufactured transistor will not change substantially, even as transistor manufacturing cost drops.

As functional test becomes increasingly nonviable, alternatives must be developed. All of these will involve adding test functionality to the design, whether this takes traditional forms such as scan design, and/ or more innovative approaches. If off-chip methods are no longer able to evaluate chip performance, or test data for deeply embedded blocks cannot be brought to the chip boundary, then alternative on-chip methods (built-in self-test) must be developed. These changes are expanded upon in the next section

### ANTICIPATED REQUIREMENTS IN TEST METHODOLOGY

Test continues to be a major expense in the IC development and manufacturing chain, with up to 35% of nonrecurring engineering (NRE) costs attributed to test development and debug, and with ATE cost per transistor expected to remain flat. Changing processes and design methods are pushing testability beyond economic limits. Rapid improvements must be made to improve overall testability and test economics.

- *Built-in-self-test* will be needed as chip performance begins to outpace tester timing accuracy, and as overall test data volume exceeds ATE and chip boundary capability. BIST needs to be made usable in short-design cycle environments by novice designers. Logic BIST methods must be developed that provide high coverage for all fault types required. Power management and test sequencing must be addressed by BIST tools. The concept of BIST needs to be extended to include additional on-chip parametric measurements currently performed by ATE.

- *Divide-and-conquer* techniques such as scan wrappers around hierarchical blocks and localized BIST will be needed to manage test complexity, as the task of developing adequate tests will become impossible for flat designs.
- *Reuse of cores* requires the encapsulation and reuse of test as well. Standardized interfacing and access methods for core testing are needed, as are composition methods to assemble complete chip tests for chips with multiple cores, including analog cores. Methods to test the interconnect between cores must also be developed, and signal integrity standards developed to assure that cores, when embedded, function as expected within the larger chip, even in the presence of noisy power, grounding, and interconnect. In addition, standardized test data formats, such as standard test interface language (STIL), will be needed to ensure portability.
- *New fault types* must be identified, and test methods developed for them, as net timing and signal integrity problems increase dramatically and introduce new modes of chip failure.
- *Design for testability* must be tightly integrated into all steps of the overall design flow, including synthesis, validation, and physical design, and include automatic test generation with high coverage for all relevant fault types (such as stuck-at faults, timing, bridging, signal integrity). These DFT techniques must apply to complex chips with multiple timing domains, and must include methods to simplify probing requirements at wafer test.
- *Signal integrity* and electromagnetic (EM) phenomena will become an increasingly important test issue as chips and test equipment become complex. New fault models (including soft error models) that incorporate the effects of EM fields must be developed. Relationships between design constraints and manufacturability and testability must be developed for different design domains. Test generators must be sensitive to signal integrity issues.
- *Timing tests* are impacted by interconnect delays, slow synthesis system drivers, increased frequencies, multiple timing domains and clock skew. Automatic test generation will be necessary to accommodate the large number of near-critical paths, and BIST systems will have to guarantee high coverage of timing faults.
- *Quiescent current (IDDQ) testing* requires extensions for manufacturing test as background currents increase, although it will remain a valuable technique for failure analysis. Single threshold IDDQ testing is no longer viable for many CMOS processes, and other approaches will be needed to ensure device reliability.
- *Hardware/software co-design* will provide opportunities for system software to be used as an aid for test. All high-level design methodologies must be sensitive to the fact that they may target unknown libraries and processes with special test problems and unknown fault statistics.
- *Analog/RF systems* present continuing challenges for test, primarily because they require test to specifications rather than structural test. As a result, there is inherent difficulty in presenting any simplification to create a meaningful fault model. Embedded analog blocks will likely require the development of BIST approaches.
- *Yield improvement and failure analysis* tools and methods will be needed to provide for rapid yield learning with very complex chips containing embedded cores (on the actual chips, not simplified test devices), and for highly automated failure analysis where faults can no longer be visually detected. Design and synthesis for diagnosis must be included if short failure analysis and yield improvement cycles are to be realized.

Other issues include the insertion of testability circuitry in systems limiting the operating speed of the system. Figure 9 shows the increase in test time relative to ITRS pincount projection. Power consumption during test must be carefully considered so test programs do not put the system into modes that consume excessive power. Application of test programs must also not cause excessive noise with the possibility of soft errors.

Table of Contents Previous Next Last

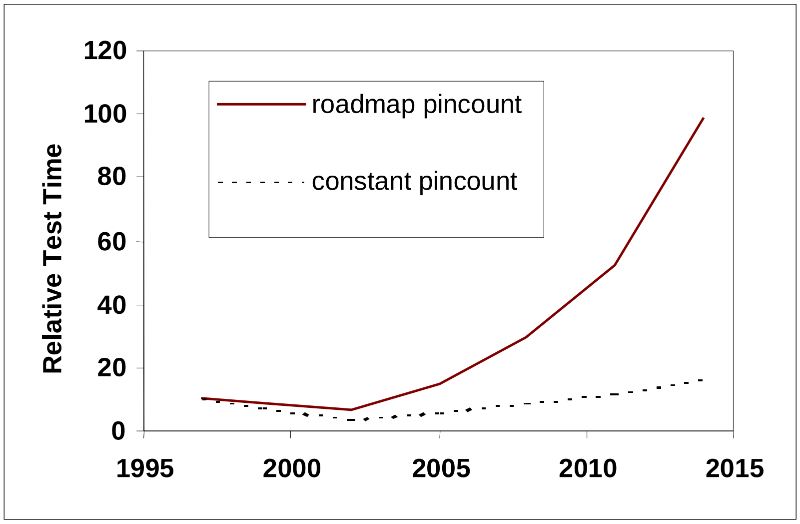


*Figure 9 Increase in Relative Test Time*

## DESIGN TECHNIQUES AND METHODOLOGIES

IC design over the next decade will be dominated by wireless and wired telecom (broadband) as well as signal processing and advanced computing. A looming problem, however, is that IC design complexity increases superexponentially with the expansion of chip function and performance as evidenced by both larger scale digital systems and SoC. On the other hand, design complexity is also growing as the design space becomes more difficult with the onset of physical and technological barriers.

The traditional rules of IC architecture design are being severely strained by these trends, mandating new approaches in the decade ahead. Some of the most widely-held rules include: (1) system-wide synchronicity, (2) use of a switching device—the CMOS transistor—that has no prescribed intrinsic size (that is, it must be scaled for faster performance, and such scaling introduces a statistical variability in transistor behavior), and (3) the capability to fabricate chips with 100% working transistors and interconnects.

### DESIGN TECHNIQUES

Design techniques pertain to the implementation of the steps that comprise a methodology. These address the circuit building blocks (or, primitives) themselves, and these are coupled to the technology infrastructure as well as the end product. Emerging concerns are power management, circuit noise, signal delay and integrity in the interconnect, clock generation and distribution, methods to increase frequency operation of circuit blocks, and faster logic families.

### *High-performance Digital Logic*

Microprocessors, DSPs, and core-based designs (ASIC, SoC) will be the major drivers for high-performance digital logic. Digital logic must evolve to satisfy two disparate demands, as follows:

1. *The MOS transistor is becoming less ideal* as it approaches the practical scaling limit, expected to occur soon after the 70 nm technology node. Deep submicron CMOS circuits need special configurations to accommodate, among others, higher gate and drain leakage currents, the limits of constant-field MOS transistor scaling, higher source/drain resistances, shrinking $V_t$ and $V_{dd}$, and poor scaling of parasitics.
2. *New alternatives to static CMOS required for high-performance designs*—The need for high speed logic now means that “high-performance logic” may be defined to be “not static CMOS.” The lower voltage swing and precharged nodes of some logic breeds facilitate the quest for more speed, though at a higher power, more susceptibility to noise, and a substantial increase in the difficulty of design. For some applications, these may be acceptable tradeoffs. Dynamic logic may find limited use, given the need for

performance, but its design difficulty and susceptibility to noise may limit its usage. New research into fast logic implementations must proceed.

### *Interconnects*

Future design issues with interconnect are: (1) the *interconnect performance bottleneck*, relating to the growing signal transmission delay of global signals relative to the clock rate and gate delay; (2) *signal integrity* concerns stemming from increasing crosstalk and electromagnetic interference, including noise from line termination mismatches; (3) *electro-migration*, which imposes severe restrictions on signal and bus line scaling; (4) need for *CAD capabilities* that address interconnect design early in the design flow to provide maximum degrees of freedom downstream for performance optimization; and (5) *power distribution networks* that produce equipotential planes across the chip.

A key distinction exists between local and global interconnect. Typical wire length distributions for microprocessor architectures, for example, show that well over 90% of wiring instances have lengths that span the physical area of less than 40 logic gates. Estimates of local wiring delay show it to be scaling as rapidly as gate delay scales, and it can be up to 10× smaller than overall gate delay. However, the situation for *global signaling* is quite different. Chip size is increasing, implying an increase in global wire lengths. Clock rates are increasing. The ratio of global wire delay to gate delay is going up rapidly, meaning that wire delay will dominate, and its delay can exceed a clock cycle. In summary, in signal space, the “interconnect problem” refers to the wire delay of global signals.

Technology advances in interconnect—particularly copper and low permittivity dielectrics—have helped postpone the onset of serious global signaling problems for one or two technology generations. By early 1999, copper was nearing production, and low κ dielectrics were regarded to be mid-term research. Optical transmission of global signals may be the next available global interconnect scheme. It seems more probable that industry will exhaust design solutions to solve the interconnect bottleneck before moving on to costly methods. Advances in the following capabilities must occur:

- Interconnect-aware CAD tools to consider wiring constraints early in the overall design cycle, thereby providing maximum degrees of design freedom. This implies both physical and logical design views are considered in parallel from the highest abstraction levels, and integrated into EDA tools.
- Optimal usage of repeaters to facilitate shorter global signal transmission delays
- Technology maturation of low κ dielectrics for use in production alongside the improvements over copper metal systems
- Refinement of on-chip wiring hierarchies that use combinations of available metal systems (Al, Cu, $SiO_2$, low κ, and others) and a formalized *interconnect architecture optimization* method to determine the number and functions of wiring levels
- New circuits and systems architectures that avoid the longer global wires
- Novel transmission line designs that minimize crosstalk, skews, reflections. This will likely involve the use of ground planes.
- Innovative system architectures to maximize local communications and minimize global signaling.
- Better packaging schemes that reduce parasitics. This includes new concepts such as chip-on-board. Power and global signal wiring may be handled in the package. This assumes that efficient chip-to-package signaling and fast trans-package signaling can be done.
- Synchronous and quasi-synchronous architectures.
- Innovative clocking schemes that utilize encoding, extraction, multi-state, and local-phase optimization to compensate for skew and latency concerns.

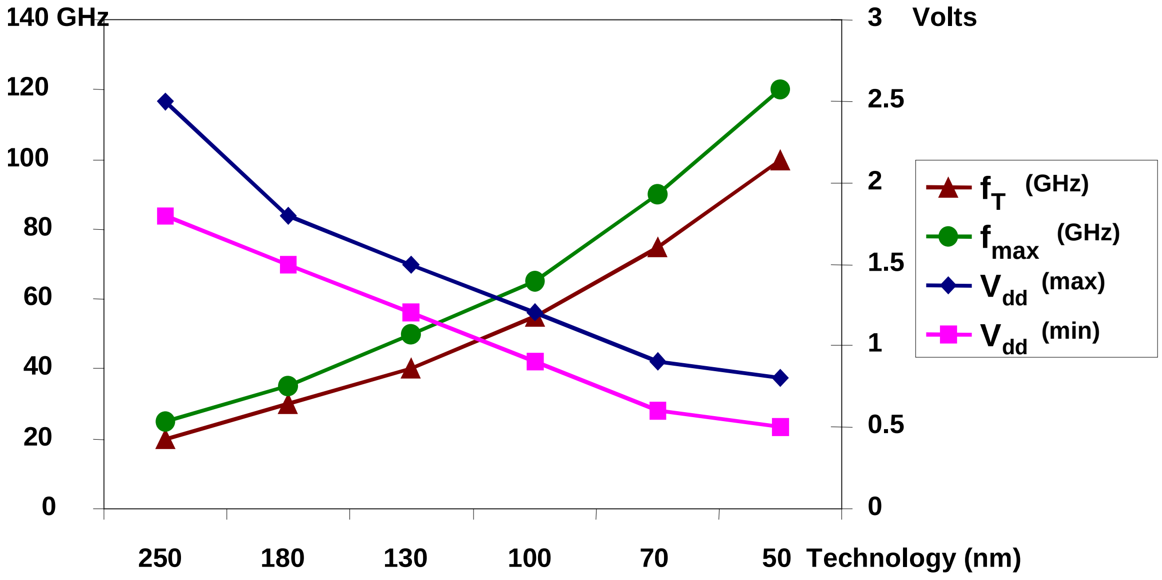
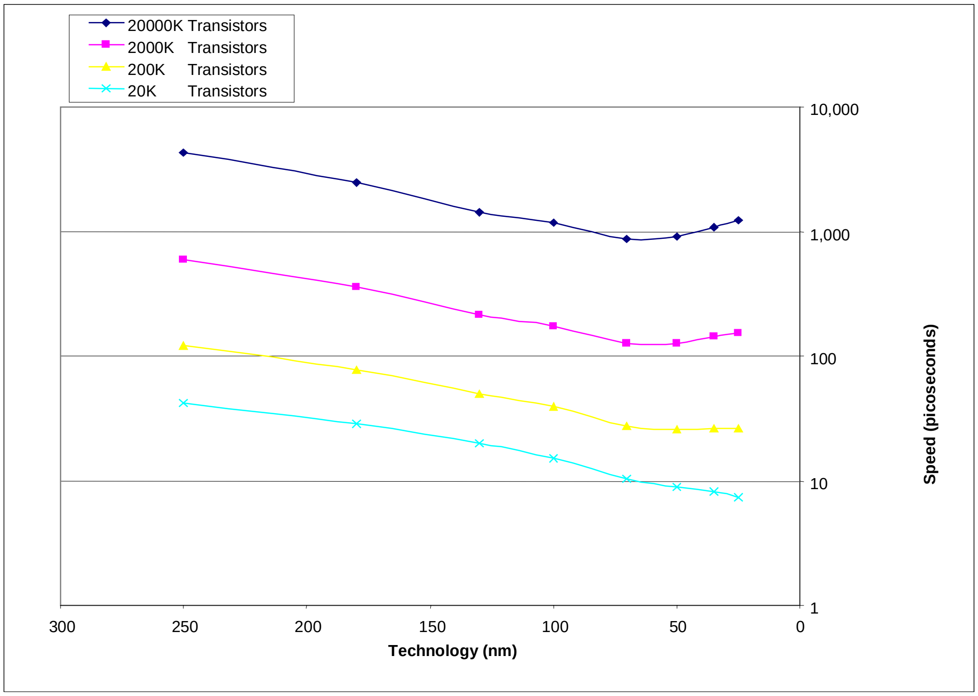


*Figure 10 Analog MOS $f_T$ and $f_{max}$ with $V_{dd}$ versus Technology Node*

With the need for higher performance SoCs, technology scaling does aid in achieving this performance, since device fT increases; *however*, scaling of power supply voltage (shown in Figure 10) does not favor analog circuits. To gain continued performance increases, innovations in the design are required at both the architectural and the circuit-level. Without new design techniques, the incremental improvements in performance will decrease as the integrated circuit processes continue to scale (Figure 11).



**•Includes low κ dielectric**

**•Includes low R conductors**

*Figure 11 Minimum Half Perimeter Delay for Various Size Cores*

Table of Contents Previous

### Clocking and Global Signal Transmission

By 2014, global clock rates are predicted to rise to approximately 3.5 GHz with local clock rates exceeding 10 GHz. Rising and falling edges usually occupy about 25% of the clock pulse width. Given that it takes five or more harmonics to correctly represent a rising/falling edge, the signal frequency content on the interconnect could approach 25 GHz. Moreover, there is the issue of timing skew, a quantity that designers need to control in design. By the 50 nm node, timing skew budgets will likely drop below 40 ps. Superimposed on these performance targets is the need for noise immunity, vis-à-vis, signal integrity. Also, there is the overall issue of delay that is embodied in the previous discussion of interconnect.

There is a need for better methods for clock design and for systems that do not require system wide synchronicity. Clocks of high-performance systems will be locally synchronous. The interfaces between locks will be quasi-synchronous meaning the phase may need to be realigned. The clock frequency in these blocks may be a much higher frequencies than the interfaces.

### DESIGN METHODOLOGY

Design methodology is the shared enterprise of the design and CAD communities. It is defined as the sequence of steps that creates a system. This sequence guides the design flow to produce a design as close as possible to the design target within the feasible design space. Conceptually, this emerges by combining top-down and bottom-up constraint propagation. The top-down path implies planning and enforcement of system specifications. Conversely, the bottom-up path asserts the underlying device infrastructure and its associated physics and technology limits. This dual top-down/bottom-up method has pervaded design all along in an *ad hoc* manner; however, in the VLSI era, the number and difficulty of the constraints require a deeper knowledge of the constraints in higher level CAD tools. Corollary to the top-down/bottom-up method is identification and solution of problems as high in the design hierarchy as possible. This provides the maximum opportunity to find a workable design space.

Future design methodologies and associated CAD tools have their origins in several emerging trends, such as:

*Expansion of top-down and bottom-up design* methodologies and architectures across disciplines to facilitate SoC design. In addition, such capabilities must include software, RF, and analog into traditional digital flows. There are needs for extensions beyond the electrical realm to interface to the human and physical world. These interfaces might include such domains as user interface, electromechanical, sensors, and photonics. Also, comprehensive simulation systems for design at multiple levels of hierarchy in this expanded set of domains are needed.

*Intellectual property (IP) (Core based design and reuse*)—The design community must have available a framework that permits encapsulated IP to be readily incorporated and used. To that end, standard core interfaces should become as pervasive as 7400-series TTL. Block-level "handshaking" protocols must be established, as an example, for timing, and signal levels.

*High-level estimation*—Many low-level problems can be solved at the system level if they are known and representable. Thus, an emerging design methodology is to accommodate low-level needs by using high-level estimations of low-level physical and technological quantities relating to interconnect and transistor/gate performance. Estimations at this level require only relative accuracy, sufficient to choose from among a discrete number of architectural options.

*Correct-by-construction design*—Verification is costly. It is better to minimize iteration between various levels in the design hierarchy. The goal is to move to correct-by-construction design at the behavioral level including electrical characteristics. Functions previously in verification will be included in synthesis.

*Robust, fully-integrated device modeling and simulation*—As design margins shrink, the task of assuring first-pass silicon is becoming very difficult. Adding to this is the problem of many design styles and devices, such as those in SoC designs. Thus, fully integrated and robust systems must exist to provide models and simulation capabilities for analog/mixed-signal and high-performance digital logic. These include:

- Sensitivity analyses to ascertain the criticality of model parameters
- Extensible compact models with variable model complexity
- Comprehensive circuit simulation/analysis capabilities for noise (phase, jitter, linearity), frequency-dependent behaviors
- Models for deep submicron devices, interconnects, parasitics, substrate effects, noise, thermal variations (self-heating), and distributed effects for RF regime

*Signal integrity*—Design methods must materialize to preserve signal integrity through synthesis. Signal integrity considerations must include crosstalk, reflections, EMI, and substrate coupling.