INTERNATIONAL TECHNOLOGY ROADMAP FOR SEMICONDUCTORS 2007 EDITION

DESIGN

THE ITRS IS DEVISED AND INTENDED FOR TECHNOLOGY ASSESSMENT ONLY AND IS WITHOUT REGARD TO ANY COMMERCIAL CONSIDERATIONS PERTAINING TO INDIVIDUAL PRODUCTS OR EQUIPMENT.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

DESIGN

SCOPE

Design technology (DT) enables the *conception*, *implementation*, and *validation* of microelectronics-based systems. Elements of DT include *tools*, *libraries, manufacturing process characterizations,* and *methodologies*. DT transforms ideas and objectives of the electronic systems designer into manufacturable and testable representations. The role of DT is to enable profits and growth of the semiconductor industry via cost-effective production of designs that fully exploit manufacturing capability. In the 2007 ITRS, the Design International Technology Working Group (ITWG) is responsible for the Design and *System Drivers* chapters, along with models for clock frequency, layout density, and power dissipation in support of the Overall Roadmap Technology Characteristics. Specific DT challenges and needs are mapped, as appropriate, to System Drivers. Readers of this chapter are encouraged to also review previous editions of the ITRS Design Chapter, which provide excellent and still-relevant summaries of DT needs.

The main message in 2007 remains—*Cost (of design) is the greatest threat to continuation of the semiconductor roadmap.* Cost determines whether differentiating value is best achieved in software or in hardware, on a programmable commodity platform, or on a new IC. Manufacturing non-recurring engineering (NRE) costs are on the order of millions of dollars (mask set + probe card); design NRE costs routinely reach tens of millions of dollars, with design shortfalls being responsible for silicon re-spins that multiply manufacturing NRE. Rapid technology change shortens product life cycles and makes time-to-market a critical issue for semiconductor customers. Manufacturing cycle times are measured in weeks, with low uncertainty. Design and verification cycle times are measured in months or years, with high uncertainty.

Recent editions of the ITRS have noted a *design productivity gap*—the number of available transistors growing faster than the ability to meaningfully design them. This gap impacts IC product value, placing at risk foundry amortization, return-on-investment (ROI) for supplier industries, and indeed the entire semiconductor investment cycle. Yet, investment in process technology continues to dominate investment in design technology. The DT roadmap enables control of design costs, as shown in Figure DESN1.

- *Hardware* aspects of design continue to witness soaring verification team sizes and test costs; Design for Manufacturability (DFM) issues now permeate the design flow. DT innovations keep hardware design costs in check, at an estimated $15M in 2007 for the consumer portable system-on-chip (SOC-CP) defined in the *System Drivers chapter*, versus around $900M had DT innovations between 1993 and 2007 not occurred.
- *Software* aspects of IC design can now account for 80% or more of embedded systems development cost, especially as we have fully entered the era of multi-core designs, with both homogeneous and heterogeneous architectures. In 2007, with the addition of hardware-related software development cost, overall design cost reaches almost $40M. Many DT innovations required in the next 15 years address software aspects of design.

Failure to effectively develop and deploy the roadmap of DT innovations will break the long-standing trend of progress in the semiconductor industry. Hence, we view these DT gaps as important crises to be addressed in the next 15 years.

This chapter first presents *silicon complexity* and *system complexity* challenges, followed by five *crosscutting challenges* (productivity, power, manufacturing integration, interference, and error tolerance) that permeate all DT areas. The bulk of the chapter then sets out detailed challenges in the form of design technology requirements and solutions tables that comprise a *quantitative* design technology roadmap. The organization follows a traditional landscape of DT areas: design process; system-level design; logical, circuit and physical design; design verification; design test, and Design for Manufacturability.[1] These challenges are discussed at a level of detail that is actionable by management, R&D, and academia in the target supplier community, such as the electronic design automation (EDA) industry. When appropriate, the detailed challenges are mapped to the microprocesser (MPU), system on chip (SOC, consumer or networking), analog/mixed-signal (AMS), and memory system drivers. Most challenges map to MPU and SOC, reflecting today's EDA technology and market segmentation. A brief unified overview of AMS-specific DT is given. The overall approach reflects the rise of application- and driver-specific DT.

[1] *Additional discussion of analog/mixed-signal circuits issues is contained in the* System Drivers chapter *(AMS Driver). Test equipment and the test of manufactured chips are discussed in the* Test chapter*, while this chapter addresses design for testability, including built-in self test (BIST).*

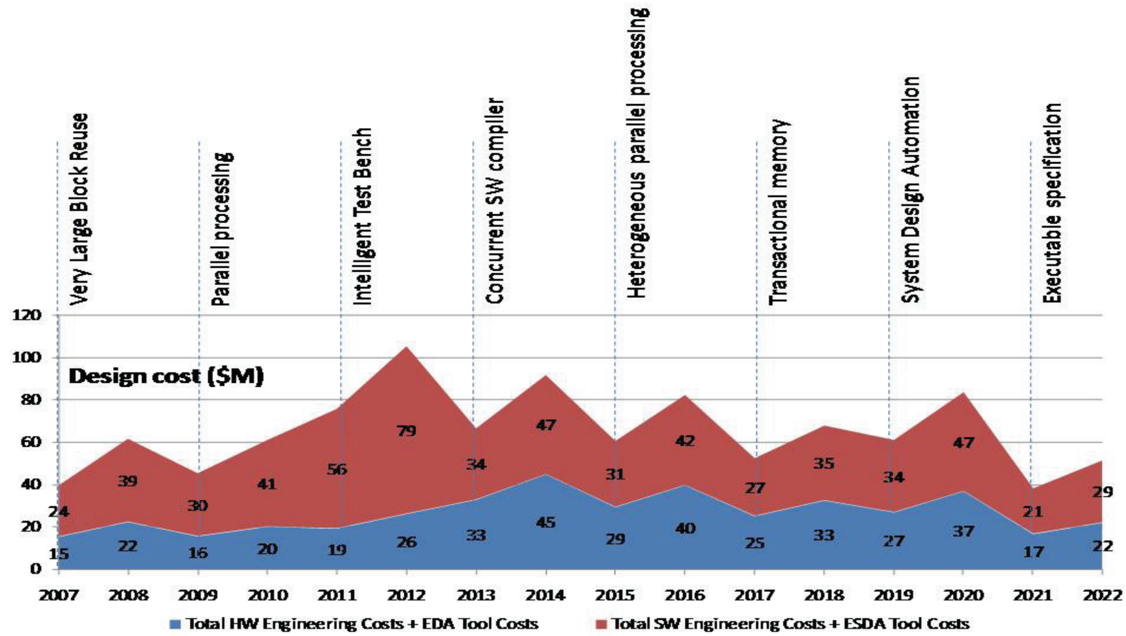

Figure DESN1 Impact of Design Technology on SOC Consumer Portable Implementation Cost

Roadmapping of DT is different from roadmapping of manufacturing technology. Manufacturing technology seeks to implement a set of requirements, and faces limits imposed by physical laws and material properties. In contrast, DT seeks to optimize designs so that they will meet their market requirements, and faces limitations imposed by computational intractability, the unknown scope of potential applications, and the multi-objective nature of design optimization. Because underlying optimizations are intractable, heuristics are inherent to DT, as are practical tradeoffs among multiple criteria such as density, speed, power, testability, or turnaround time. Evaluation of DT quality is thus context-sensitive, and dependent on particular methodologies or design instances. Furthermore, while ITRS technology generations occur discretely when all needed technology elements are in place, DT improvements can improve productivity or quality even "in isolation," and are thus deployable when developed.

OVERALL CHALLENGES

DT faces two basic types of complexity—*silicon complexity* and *system complexity*—that follow from roadmaps for ITRS manufacturing technologies.

Silicon complexity refers to the impact of process scaling and the introduction of new materials or device/interconnect architectures. Many previously ignorable phenomena now have great impact on design correctness and value:

- *Non-ideal scaling of device parasitics and supply/threshold voltages* (leakage, power management, circuit/device innovation, current delivery)
- *Coupled high-frequency devices and interconnects* (noise/interference, signal integrity analysis and management, substrate coupling, delay variation due to cross-coupling)
- *Manufacturing variability* (statistical process modeling and characterization, yield, leakage power)
- *Complexity of manufacturing handoff* (reticle enhancement and mask writing/inspection flow, NRE cost)
- Process variability (library characterization, analog and digital circuit performance, error-tolerant design, layout reuse, reliable and predictable implementation platforms)
- Scaling of global interconnect performance relative to device performance (communication, synchronization)

- *Decreased reliability* (gate insulator tunneling and breakdown integrity, joule heating and electromigration, single-event upset, general fault-tolerance)

Silicon complexity places long-standing paradigms at risk, as follows: 1) system-wide synchronization becomes infeasible due to power limits and the cost of robustness under manufacturing variability; 2) the CMOS transistor becomes subject to ever-larger statistical variabilities in its behavior; and 3) fabrication of chips with 100% working transistors and interconnects becomes prohibitively expensive. Available implementation fabrics (from direct-mapped custom through general-purpose software-programmable) easily span four orders of magnitude in performance metrics (e.g., GOps/mW), and there is added opportunity to leave value on the table via ill-advised guardbands, abstractions, or other methodological choices. These challenges demand more broadly trained designers and design technologists, as well as continued mergers between traditionally separated areas of DT (synthesis-analysis, logical-physical, etc.).

System complexity refers to exponentially increasing transistor counts enabled by smaller feature sizes and spurred by consumer demand for increased functionality, lower cost, and shorter time-to-market.[2] Many challenges are facets of the nearly synonymous *productivity* challenge. Additional complexities (system environment or component heterogeneity) are forms of *diversity* that arise with respect to system-level SOC integration. Design specification and validation become extremely challenging, particularly with respect to complex operating contexts. Tradeoffs must be made between all aspects of value or quality, and all aspects of cost. (A simplistic example: "Moore's Law" for performance might suggest a tradeoff of design time (= time-to-market) for speed at roughly 1% per week.) Implied challenges include:

- *Reuse*—support for hierarchical design, heterogeneous SOC integration (modeling, simulation, verification, test of component blocks) especially for analog/mixed-signal
- *Verification and test*—specification capture, design for verifiability, verification reuse for heterogeneous SOC, system-level and software verification, verification of analog/mixed-signal and novel devices, self-test, intelligent noise/delay fault testing, tester timing limits, test reuse
- *Cost-driven design optimization*—manufacturing cost modeling and analysis, quality metrics, co-optimization at die-package-system levels, optimization with respect to multiple system objectives such as fault tolerance, testability, etc.
- *Embedded software design*—predictable platform-based electronic system design methodologies, codesign with hardware and for networked system environments, software verification/analysis
- *Reliable implementation platforms*—predictable chip implementation onto multiple circuit fabrics, higher-level handoff to implementation
- *Design process management*—design team size and geographic distribution, data management, collaborative design support, "design through system" supply chain management, metrics and continuous process improvement

Together, the silicon and system complexity challenges imply *superexponentially increasing complexity* of the design process. To deal with this complexity, DT must provide concurrent optimization and analysis of more complex objectives and constraints, acknowledge added considerations such as design reuse and manufactured system cost in the design optimization, and encompass added scope such as embedded software design and interfaces to manufacturing. The sheer breadth of silicon and system complexities is also a challenge to roadmapping of DT and the electronic design automation (EDA) industry.

Five *crosscutting challenges*—1) design productivity, 2) power management, 3) design for manufacturability, 4) interference, and 5) reliability—underlie the design cost "meta-challenge" and have potential solutions that span all areas of DT. Three are highlighted as difficult challenges in the ITRS Executive Summary. *Design productivity* (a "cost-effective manufacturing" challenge) is closely linked to system and design process complexity, and is the most massive and critical DT, challenge both near and long term. *Power management* (an "enhancing performance" challenge) oscillates between a performance-driven active power crisis and a variability-driven leakage power crisis in the near term. *Design for Manufacturing* (a "cost-effective manufacturing" challenge) is required for the industry to produce chips in large quantities at acceptable cost and on economically feasible schedules: the past focus on lithography hardware

[2] *A "Law of Observed Functionality," notorious in consumer electronics, states that transistor count increases exponentially while the system value (utility) increases linearly (see T. Claasen, "The Logarithmic Law of Usefulness," Semiconductor International, July 1998). Similarly diminishing returns in the MPU space (Pollack's Rule) are described in the System Drivers chapter.*

limitations will broaden as variability in its many forms becomes a crisis, and deep integrations of DFM with yield management and design for test are needed. Table DESN1 summarizes key aspects of the crosscutting DT challenges.

Table DESN1 Overall Design Technology Challenges

Challenges ≥ 32 nm	*Summary of Issues*
Design productivity	System level: high level of abstraction (HW/SW) functionality spec, platform based design, multi-processor programmability, system integration, AMS co-design and automation Verification: executable specification, ESL formal verification, intelligent test bench, coverage-based verification Logic/circuit/layout: analog circuit synthesis, multi-objective optimization
Power consumption	Logic/circuit/layout: dynamic and static (leakage), system and circuit, power optimization
Manufacturability	Performance/power variability, device parameter variability, lithography limitations impact on design, mask cost, quality of (process) models ATE interface test (multi-Gb/s), mixed-signal test, delay BIST, test-volume-reducing DFT
Reliability	Logic/circuit/layout: MTTF-aware design, BISR, soft-error correction
Interference	Logic/circuit/layout: signal integrity analysis, EMI analysis, thermal analysis
Challenges <32 nm	*Summary of Issues*
Design productivity	Complete formal verification of designs, complete verification code reuse, complete deployment of functional coverage Tools specific for SOI and non-static logic, and emerging devices Cost-driven design flow Heterogeneous component integration (optical, mechanical, chemical, bio, etc.)
Power consumption	SOI power management
Manufacturability	Uncontrollable threshold voltage variability Advanced analog/mixed signal DFT (digital, structural, radio), "statistical" and yield-improvement DFT Thermal BIST, system-level BIST
Reliability	Autonomic computing, robust design, SW reliability
Interference	Interactions between heterogeneous components (optical, mechanical, chemical, bio, etc.)

ATE—automatic test equipment BISR—built-in self repair BIST—built-in self test DFT—design for test EMI—electromagnetic interference ESL—Electronic System-Level HW/SW—hardware/software MTTF—mean time to failure SOI—silicon on insulator

Crosscutting Challenge 1: Design Productivity. To avoid exponentially increasing design cost, overall productivity of designed functions on chip must scale at > 2× per technology generation. Reuse productivity (including migration and analog, mixed-signal, and RF (AMSRF) core reuse) of design, verification and test must also scale at > 2× per technology generation. Implied needs are in: 1) verification, which is a bottleneck that has now reached crisis proportions; 2) reliable and predictable silicon implementation fabrics that support ever-high level electronic system design handoff; 3) embedded software design, which has emerged as the most critical challenge to SOC productivity; 4) particularly for the MPU context, improved productivity of large, distributed design organizations that work with tools from a variety of sources; and 5) automated methods for AMS design and test, which are required by the SOC and AMS system drivers. These improvements require metrics of normalized design quality as a function of design quality, design NRE cost, manufacturing NRE cost, manufacturing variable cost, and semiconductor product value. Metrics of design technology quality such as stability, predictability, and interoperability must be improved as well. Time-to-market of new design technology must be reduced, e.g., via standards and platforms for interoperability and DT reuse.

Crosscutting Challenge 2: Power Management. Non-ideal scaling of planar CMOS devices, together with the roadmap for interconnect materials and package technologies, presents a variety of challenges related to power management and current delivery. 1) Both the MPU and Consumer Portable drivers in the *System Drivers chapter* require flat active and standby power, even as logic content and throughput continue to grow exponentially. DT must address the resulting *power management gap*. 2) Increasing power densities worsen thermal impact on reliability and performance, while decreasing supply voltages worsen switching currents and noise. These trends stress on-chip interconnect resources (such as to control V = IR power supply drop in light of the Assembly and Packaging roadmap for bump count and passivation opening size), ATE equipment limits, and burn-in paradigms. 3) Integration of distinct high-performance, low operating power (LOP), and low standby power (LSTP) devices demands power optimizations that simultaneously exploit many degrees of freedom, including multi-V_t, multi-T_{ox}, multi-V_{dd} coexisting in a single core—while guiding additional power optimizations at the architecture, operating system, and application software levels. 4) Leakage power varies exponentially with key process parameters such as gate length, oxide thickness and threshold voltage; this presents severe challenges to both analysis and optimization in light of both scaling and variability.

Crosscutting Challenge 3: Design for Manufacturing. "Red bricks," that is, technology requirements for which no known solutions exist, are increasingly common throughout the ITRS. At the same time, challenges that are impossible to solve within a single technology area of the ITRS may be solvable (more cost-effectively) via appropriate synergies with DT. New interconnections between Design and all other Manufacturing-related disciplines lead to the rise of Design for Manufacturability (DFM), to which this DT roadmap devotes an entire section. Indeed, the feasibility of future technology nodes will come to depend on this communication. Several examples are as follows. 1) Tester equipment cost and speed limitations may be addressed by more rapid adoption of new fault models (for example, crosstalk, path delay), along with corresponding automatic test pattern generation (ATPG) and BIST techniques. 2) System implementation cost, performance verification, and overall design turnaround time (TAT) may be improved through die-package-board co-optimization and analysis, as well as DT for system-in-package design. 3) CD control requirements in the Lithography, Process Integration, Devices, and Structures (PIDS), Front-End Processing (FEP), and Interconnect technology areas may be relaxed by new DT for correctness under manufacturing variability (e.g., variability-aware circuit design, regularity in layout, timing structure optimization, and static performance verification). 4) Manufacturing non-recurring costs can be reduced by more intelligent interfaces to mask production and inspection flows.

Crosscutting Challenge 4: Interference. Resource-efficient communication and synchronization, already challenged by global interconnect scaling trends, are increasingly hampered by noise and interference. Prevailing signal integrity methodologies in logical, circuit and physical design are reaching their limits of practicality. These methodologies include repeater insertion rules for long interconnects, slew rate control rules, power/ground distribution design for inductance management, etc. Scaling and SOC integration of mixed-signal and (radio frequency) RF components will require more flexible and powerful methodologies. Issues include noise headroom (especially in low-power devices and dynamic circuits); large numbers of capacitively and inductively coupled interconnects; supply voltage IR drop and ground bounce; thermal impact on device off-currents and interconnect resistivities; and substrate coupling. A basic DT challenge is to improve characterization, modeling, analysis, and estimation of noise and interference at all levels of design.

Crosscutting Challenge 5: Reliability. Relaxing the requirement of 100% correctness for devices and interconnects may dramatically reduce costs of manufacturing, verification, and test. Such a paradigm shift is likely forced in any case by technology scaling, which leads to more transient and permanent failures of signals, logic values, devices, and interconnects. Several example issues are as follows. 1) Below 65nm, single-event upsets (soft errors) impact field-level product reliability, not only for embedded memories, but for logic and latches as well. 2) Methods for accelerated lifetime testing (burn-in) become infeasible as supply voltages decrease (resulting in exponentially longer burn-in times); even power demands of burn-in ovens become overwhelming. 3) Atomic-scale effects can demand new "soft" defect criteria, such as for non-catastrophic gate oxide breakdown. In general, automatic insertion of robustness into the design will become a priority as systems become too large to be functionally tested at manufacturing exit. Potential solutions include automatic introduction of redundant logic and on-chip reconfigurability for fault tolerance, development of adaptive and self-correcting or self-healing circuits, and software-based fault-tolerance.

DETAILED DESIGN TECHNOLOGY CHALLENGES

The remainder of this chapter gives an overview of design methodology, followed by quantified challenges and potential solutions in five main areas of DT. As noted above, most challenges map to SOC, reflecting today's EDA technology and market segmentation.

DESIGN METHODOLOGY

The process of designing and implementing a chip requires a large collection of *techniques*, or *tools*, and an effective *methodology* by which a designer's input predictably leads to a manufacturable product.[3] While considerable attention is given to the tools needed, the equally important subject of design methodology is often neglected. Each technology generation requires designers to consider more issues; hence, new analysis methods and tools must be developed to evaluate new phenomena and aid the designer in making critical design decisions. An even greater challenge is determining the most effective sequence in which issues should be considered, and design decisions made, in order to minimize iterations.

With the transition from microelectronics to nanoelectronics along "More Moore", "More than Moore", and "Beyond CMOS" trajectories come inevitable paradigm shifts in the design of silicon systems. These affect all levels of the design process, and require enormous effort toward new methodologies and tools. DT must enable the creation of highly complex yet cost-efficient silicon system solutions, while exploiting all available opportunities afforded by nanoelectronics. For innovative applications (see the *System Drivers chapter* for more details) to become affordable, daunting challenges with respect to EDA must be overcome. Shrinking of silicon devices and related fabrication processes has been the foundation for ever more powerful integrated silicon solutions that permeate daily life. This has depended on the availability of EDA and design technologies that smoothly transform specification data into manufacturing-ready layout data, and perform necessary verifications at different levels of abstraction (see Figure DESN2). However, the availability of such design technologies cannot be taken for granted anymore.

Cost and time-to-market of new SOCs requires DT that spans all parts of a complex design process (Figure DESN2) which consists of two main elements: the implementation path (left) and the verification path (right). The figure shows the so-called V-Cycle of a design system architecture integrating both "More Moore" and "More than Moore" design aspects. The arrows indicate available (**green**) and partially available (yellow) elements of a state of the art design system environment. Future requirements for EDA are indicated in **red**.

[3] *Design methodology is developed jointly by designers and design technologists; it is the sequence of steps by which a design process will reliably produce a design "as close as possible" to the design target while maintaining feasibility with respect to constraints. Design methodology is distinct from design techniques, which pertain to implementation of the steps that comprise a methodology and are discussed in the context of their respective DT areas. All known design methodologies combine 1) enforcement of system specifications and constraints via top-down planning and search, with 2) bottom-up propagation of constraints that stem from physical laws, limits of design and manufacturing technology, and system cost limits.*

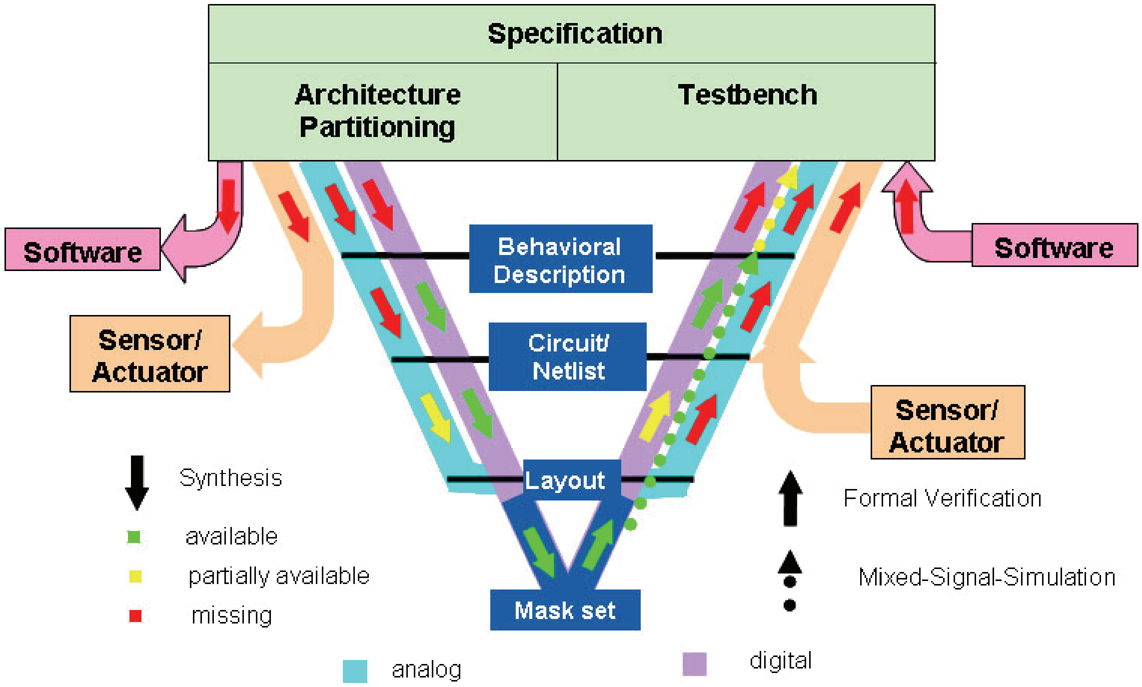

Figure DESN2 The V-Cycle for Design System Architecture[4]

A system specification undergoes stepwise refinement from architecture down to mask set. The idea of SOC design is to explore the system in initial stages; then, while transforming the system-level description to more detailed levels (behavioral, circuit/netlist, layout and mask), a verification methodology must "guard", i.e., prove correctness of, each incremental refinement step. Beyond the SOC's components and behavior, system software and sensors/actuators must also be handled by future implementation and verification procedures (**red**). The digital design path (**purple**) of the design system will require tools and methodologies above the behavioral abstraction level (**red**). The analog design path is even more challenging (indicated by several **red** arrows) because complete solutions are rarely found at all level of abstractions. As a result, an automated design process of analog design (**light blue**) is still an open question in today's design environments. To summarize, future requirements are twofold. 1) More than Moore scaling requires the integration of aspects outside the traditional SOC flow, notably software on one hand sensors and actuators on the other, all marked in red. 2) The future holds new requirements for the SOC design flow, notably tool support for higher abstraction levels in both digital and analog flows. This will entail behavioral synthesis from higher-level specifications on the left side, as well as the corresponding abstract verification steps on the right hand side, all marked in red. The analog and mixed-signal flow additionally requires tools for analog circuit synthesis and automated analog verification on all design levels.

The transition to nanoscale technologies causes all design steps from specification to fabrication to not only become seriously interdependent, but also closely linked to the IC's eventual yield and reliability. Consequently, SOC design productivity cannot follow the pace of the nanoelectronics technology innovation which is characterized by Moore's law (see Figure DESN3). To drastically increase design productivity, an additional level of abstraction – the so-called System Level – has been introduced. New system-level challenges and solutions are described in the next section.

---

[4] *Elements of this discussion were initially developed as part of a recent update of the Strategic Research Agenda (SRA) within the European Nanoelectronics Initiative Advisory Council (ENIAC) chapter on Design Automation. Figure DESN2 shows status of the EDA design flow from an automotive point of view, but is general enough to demonstrate the future EDA requirements. The figure was developed by Peter van Staa (Bosch).*

SYSTEM-LEVEL DESIGN

For decades, designers have reasoned about systems at various levels of abstraction (block diagrams, incomplete state charts, program models, etc.) with little support from design automation tools. This situation must change in the near future if necessary advances in productivity are to be achieved. To simplify the specification, verification and implementation of systems including hardware and software, and to enable more efficient design space exploration, a new level of abstraction is needed above the familiar register-transfer level.

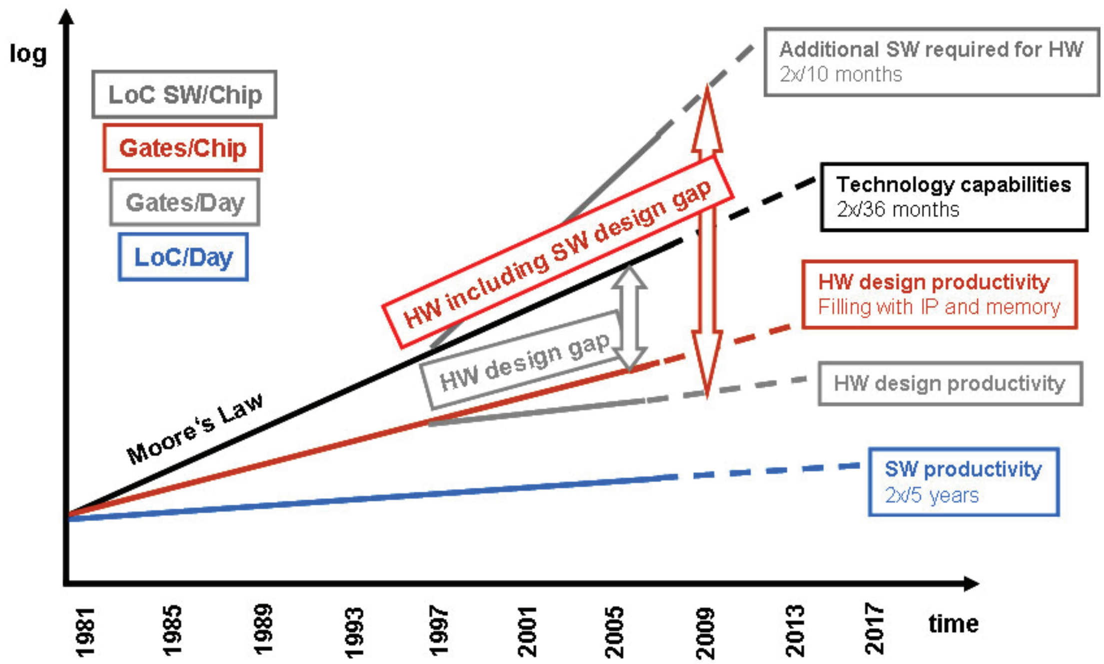

Figure DESN3 Hardware and Software Design Gaps Versus Time[5]

In system-level design[6], methodological aspects are rapidly becoming much harder than tools aspects: enormous system complexity can be realized on a single die, but exploiting this potential reliably and cost-effectively will require a roughly 50× increase in design productivity over what is possible today. Silicon complexities imply that reliable and available systems must be built out of heterogeneous, unreliable device and interconnect components. Global synchronization becomes prohibitively costly due to process variability and power dissipation, and cross-chip signaling can no longer be achieved in a single clock cycle. Thus, system design must comprehend networking and distributed computation metaphors (for example, with communication structures designed first, and functional blocks then integrated into the communication backbone), as well as interactions between functional and interconnect pipelining. System complexities dramatically increase with the amount of software in embedded systems and the rapid adoption of multi-core SOC architectures. Not only is software dominating overall design effort as shown in Figure DESN3, but hardware dependent

[5] *This figure shows the demand for software which is currently doubling every 10 months, the capability of technology which is currently doubling every 36 months all well as the hardware and software design productivity. Whereas the hardware design productivity improved over the last couple of years by filling the silicon with multi-core components and memory providing functionality only with additional software, the productivity especially for hardware-dependent software is far behind and only doubling every 5 years. This ITRS version includes therefore for the first time additional figures for illustrating these new software requirements. This material is based on ITRS data on one hand and on additional input from Infineon and STMicroelectronics and has been presented by Wolfgang Ecker from Infineon at DATE 2007. The red arrow summarizes the new design gap including both hardware and software.*

[6] *At the system-level, silicon resources are defined in terms of abstract functions and blocks; design targets include software (embedded code in high level and assembly language, configuration data, etc.) and hardware (cores, hardwired circuits, busses, reconfigurable cells). "Hardware" (HW) corresponds to implemented circuit elements, and "software" (SW) corresponds to logical abstractions (instructions) of functions performed by hardware. Behavior and architecture are independent degrees of design freedom, with software and hardware being two components of architecture. The aggregate of behaviors defines the system function, while the aggregate of architecture blocks defines a system platform. Platform mapping from system functionality onto system architecture is at the heart of system-level design, and becomes more difficult with increased system complexity and heterogeneity (whether architectural or functional).*

software that is tightly coupled to hardware and required functionality must be eventually handled by an SOC integration and verification process that is still hardware-centric today.

In this 2007 edition of the ITRS roadmap, the challenges in system-level design remain largely the same as in previous editions, this fact itself being evidence of the enormous complexity of these challenges. For instance, although behavioral synthesis is essential to system-level design, efficient behavioral synthesis is not yet realized today, despite having been a research topic for more than a decade, and despite recent advances driven by C and SystemC-based synthesis and transaction level modeling (TLM) technologies. Also, where the current SOC design process is still largely "best effort" driven, an inevitable new direction is that of self-healing, self-configuration and error correction. DT will play an enabling role in linking application requirements derived from societal and market needs to their eventual cost-effective system implementation in the "More Moore" and "More than Moore" domains. Table DESN2 gives quantitative requirements for system-level design in future technology generations.

Table DESN2a System-Level Design Requirements—Near-term Years

Year of Production	*2007*	*2008*	*2009*	*2010*	*2011*	*2012*	*2013*	*2014*	*2015*
DRAM ½ Pitch (nm) (contacted)	*65*	*57*	*50*	*45*	*40*	*36*	*32*	*28*	*25*
Design Reuse									
Design block reuse [1] % of all logic	35%	36%	38%	40%	41%	42%	44%	46%	48%
Platform Based Design									
Available platforms [2] Normalized to 100% in the start year [3]	87%	83%	75%	70%	60%	55%	52%	48%	45%
Platforms supported [4] % of platforms fully supported by tools [5]	10%	25%	35%	50%	57%	64%	75%	80%	85%
High Level Synthesis									
Accuracy of high level estimates (performance, area, power, costs) [6] % versus measurements	60%	63%	66%	70%	73%	76%	80%	83%	86%
Reconfigurability									
SOC reconfigurability [7] % of SOC functionality that is reconfigurable	28%	28%	30%	35%	38%	40%	42%	45%	48%
Analog/Mixed Signal									
Analog automation [8] % versus digital automation [9]	17%	17%	24%	24%	27%	30%	32%	35%	38%
Modeling methodology, description languages, simulation environments [10] % vs. digital methodology	55%	58%	60%	62%	65%	67%	70%	76%	78%

Manufacturable solutions exist, and are being optimized
Manufacturable solutions are known
Interim solutions are known
Manufacturable solutions are NOT known

Table DESN2b System-Level Design Requirements—Long-term Years

Year of Production	*2016*	*2017*	*2018*	*2019*	*2020*	*2021*	*2022*
DRAM ½ Pitch (nm) (contacted)	*22*	*20*	*18*	*16*	*14*	*12*	*10*
Design Reuse							
Design block reuse [1] % of all logic	49%	51%	52%	54%	55%	57%	58%
Platform Based Design							
Available platforms [2] Normalized to 100% in the start year [3]	43%	40%	37%	35%	32%	29%	27%
Platforms supported [4] % of platforms fully supported by tools [5]	90%	92%	94%	95%	97%	99%	100%
High Level Synthesis							
Accuracy of high level estimates (performance, area, power, costs) [6] % versus measurements	90%	92%	94%	95%	97%	99%	100%
Reconfigurability							
SOC reconfigurability [7] % of SOC functionality that is reconfigurable	50%	53%	56%	60%	62%	65%	68%
Analog/Mixed Signal							
Analog automation [8] % versus digital automation [9]	40%	43%	46%	50%	52%	55%	58%
Modeling methodology, description languages, and simulation environments [10] % versus digital methodology	80%	83%	86%	90%	92%	95%	98%

Notes for Tables DESN2a and b

[1] This requirement is not unique to system-level design, but is also a requirement for design in general (see the SOC Consumer Design Productivity Trends in the System Drivers chapter). Definition: The portion of a design which is not newly developed but which is composed of pre-existing components. Rationale: Reuse is one of the main factors which drive design productivity, and is one of the key concepts behind system-level design.

The reuse in year n from a particular reference year can be calculated as reuse(n) = 1 - (1 – reuse) • ((1 + pgrowth)^n / (1 + cgrowth)^n) where reuse = reuse in reference year; pgrowth = (expected) mean annual productivity growth rate, excluding the effect of reuse; and cgrowth= (expected) mean annual growth rate of design complexity. The calculation further assumes that the size of the design staff as well as the design cycle time stay constant over the given period of time. The rationale for the formula is that the gap between the productivity growth (without effect of reuse) and the complexity growth must be filled by reuse if the silicon process technology is to be fully exploited in SOC design.

[2] Definition: A platform is a specific combination of system components that supports a specific application area (e.g. wireless, automotive, consumer electronics/multimedia, Small Office Home Office (SOHO) networks, etc.). System components comprise one or more processors, (real time) operating system, communication infrastructure, memory, customizable analog and digital logic, and virtual sockets for new logic. Basic functionality for the application area is provided by a number of already integrated components. System differentiation is achieved by integration of few new components either in hardware or software. Rationale: Platform based design is an important driver for design productivity, since it highly promotes reuse. In addition, system-level specifications require platforms to which they can be mapped.

[3] Different platforms are expected to converge in the future, owing to advances in manufacturing technology and higher integration densities; hence, the total number of platforms is expected to decrease.

[4] Definition: (Full) Support for a particular platform means an integrated development environment that supports and automates architectural exploration, HW/SW partitioning, architectural/platform mapping, HW/SW co-verification, performance/area/power/cost tradeoffs, HW and SW synthesis, and HW/SW interface synthesis for that platform. Rationale: A high degree of automation is a key to the success of system-level design.

[5] Although there already exist some solutions today for some aspects of platform based modeling, full integration has not yet been achieved.

[6] Definition: The degree to which the estimated results match measurements taken on the fabricated IC. Rationale: For high-level synthesis techniques a high accuracy of estimations is essential to delivery of high-quality synthesis results that meet user constraints, such as minimum performance, maximum area, etc.

[7] Definition: The portion of an SOC or of a design, in terms of functionality, that is implemented in SW or HW that is reconfigurable. Rationale: Growing system complexity will make it impossible to ship designs without errors in the future. Hence, it is essential to be able to fix errors after fabrication. In addition, reconfigurability increases reuse, since existing devices can be reprogrammed to fulfill new tasks.

[8] Definition: Degree of automation in analog design. Rationale: Analog components are to be found in most electronic systems today and analog/mixed-signal design is an essential and important part of electronic design. Hence, as for digital design, a high degree of automation in analog design across as many levels as possible is required in the future, if design productivity growth is to be maintained or increased.

[9] The degree to which the degree of automation in analog design matches the degree of automation in digital design.

[10] Definition: The degree to which analog methodology, description languages, and simulation environments match the maturity of their digital counterparts. Rationale: As digital and analog design become nearly equally important at a system level, analog design and modeling methodologies must reach a similar maturity as their digital counterparts in order to be able to support required productivity growth in system-level design.

Figure DESN4 specifies solutions for system-level design in their corresponding time frames, and Table DESN3 explains how requirements correspond to solutions.

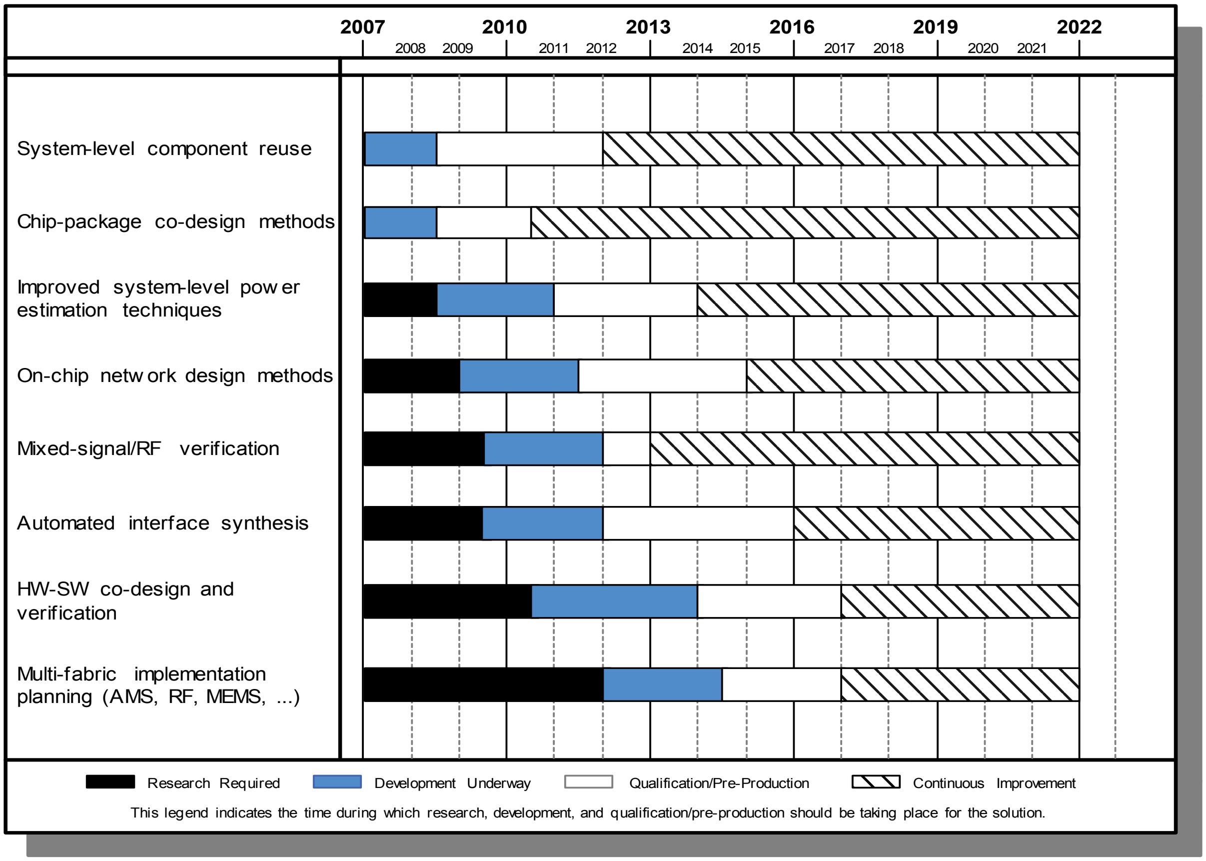

Figure DESN4 System-Level Design Potential Solutions

Table DESN3 Correspondence Between System-Level Design Requirements and Solutions

Requirement	*Solution*	*Explanation of the Correspondence*
Design block reuse	System-level component reuse	The larger and more complex the components that can be reused, the greater the expected overall design reuse
	On-chip network design methods	Standardized communication structures and interfaces support reuse: IPs with standardized interfaces can be easily integrated and exchanged, and communication structures reused
Available platforms	Multi-fabric implementation planning (AMS, RF, MEMS, …)	Enables integration of different fabrics on same die or in same package (SIP); hence, enables reduced number of platforms
Platforms supported	Automated interface synthesis	Automated interface synthesis is one building block to an integrated synthesis flow for whole platforms.
	Automated HW-SW co-design and verification	Required for integrated, platform-based system development
Accuracy of high level estimates	Improved system-level power estimation techniques	System-level power estimation needs to match progress in high-level area and performance estimation
	Chip-package co-design methods	Packaging effects, e.g., on timing, must be accounted for in higher-level estimations
SOC reconfigurability	On-chip network design methods	To provide flexible, reconfigurable communication structures
Analog automation	Multi-fabric implementation planning (AMS, RF, MEMS, …)	Multi-fabric implementation planning for AMS and RF components are a building block to analog automation
Modeling methodology, description languages, simulation environments	Mixed-Signal/RF verification	As in digital design, verification is an increasingly critical and time-consuming activity in the design flow

LOGICAL, CIRCUIT AND PHYSICAL DESIGN

In the traditional view of IC implementation, *logical design* is the process of mapping from the system-level design handoff (currently at the register-transfer level) to a gate-level representation that is suitable for input to physical design. *Circuit design* addresses creation of device and interconnect topologies (standard cells, full-custom analog, etc.) that achieve prescribed electrical and physical properties while remaining feasible with respect to process- and manufacturability-induced constraints. *Physical design* addresses aspects of chip implementation (floor planning, placement, routing, extraction, performance analysis) related to the correct spatial embedding of devices and interconnects. The output of physical design is the handoff ("tapeout") to manufacturing (currently centered around a generalized data stream (GDSII) file), along with verifications of correctness (design rules, layout versus schematic, etc.) and constraints (timing, power, reliability, etc.). Together, logical, circuit and physical design comprise the *implementation* layer of DT that supports system-level design.

Design productivity requires system-level signoff into reliable, predictable implementation fabrics. However, silicon complexity makes it difficult to estimate and abstract the effects of physics and embedding on eventual design quality (timing, power, signal integrity, reliability, manufacturability, etc.). To avoid excessive guardbanding due to poor estimates, logical design and eventually system-level design must become more closely linked with physical design. Thus, the recent paradigm of hierarchical, top-down, layout-based implementation planning supported by a tightly integrated, incremental static (power, timing, noise) analysis "backplane" is likely to persist. Future successful implementation DT will depend heavily on methodology choices to juggle process/device abstraction, constraint manipulation, analyses, and optimizations in the face of exploding complexities and emerging concerns such as error-tolerance, variability and cost.

Current hardware design automation practices must evolve to handle the challenges and opportunities of finer-featured fabrication processes. The methodologies premised on the principle of separation of concerns in which a complex design flow is serialized into a sequence of manageable steps that are loosely coupled are becoming increasingly outdated. In those scenarios, decisions made in the early stages of design flow become binding constraints on later stages. Such serialization potentially yields less optimal designs than a methodology that simultaneously considers all design aspects. This situation is unavoidable, however, due to the practical difficulty of concurrent optimization of all design parameters, and is deemed acceptable as long as the constraints that are fed forward can be met. The methodology breaks down, however, when these constraints become infeasible; the typical action in such cases is an iteration that revisits earlier design stages to identify and change problematic decisions. Such iteration has become particularly necessary between the logical and physical synthesis steps due to the inability of layout synthesis to satisfy timing requirements, that is, achieve timing closure. Ideally, the time-wasting iteration between logic and layout synthesis in today's design methodologies could be eliminated by fusing these stages to simultaneously optimize the logical structure as well as layout of a circuit.

LOGICAL, CIRCUIT, AND PHYSICAL DESIGN REQUIREMENTS

Each technology requirement for logical, circuit and physical design is either a workaround that addresses effects of a given technology (i.e., a *control requirement*), or a desired capability for a given time frame and technology (i.e., an *objective requirement*). Table DESN4 gives numerical targets for each requirement.

Table DESN4a Logical/Circuit/Physical Design Technology Requirements—Near-term Years

Year of Production	*2007*	*2008*	*2009*	*2010*	*2011*	*2012*	*2013*	*2014*	*2015*
Asynchronous global signaling: % of a design driven by handshake clocking	7%	11%	15%	17%	19%	20%	22%	23%	25%
Parameter uncertainty:%-effect (on signoff delay)	6%	8%	10%	11%	11%	12%	14%	15%	18%
Simultaneous analysis objectives: # of objectives during optimization	4	5	6	6	6	6	7	8	8
Circuit families: # of families in a single design	3	3	4	4	4	4	4	4	4
Synthesized analog content: % of total design analog content	15%	16%	17%	18%	19%	20%	23%	25%	28%
Full-chip leakage (normalized to full-chip leakage power dissipation in 2007)	1	1.5	2	2.5	2.75	3	3.5	4	6

Table DESN4b Logical/Circuit/Physical Design Technology Requirements—Long-term Years

Year of Production	*2016*	*2017*	*2018*	*2019*	*2020*	*2021*	*2022*
Asynchronous global signaling: % of a design driven by handshake clocking	30%	30%	30%	35%	40%	43%	45%
Parameter uncertainty: %-effect (on signoff delay)	20%	20%	20%	22%	25%	26%	28%
Simultaneous analysis objectives: # of objectives during optimization	8	8	8	8	8	8	8
Circuit families: # of families in a single design	4	4	4	4	4	4	4
Synthesized analog content: % of total design analog content	30%	35%	40%	45%	50%	55%	60%
Full-chip leakage (normalized to full-chip leakage power dissipation in 2007)	8	8	8	8	8	8	8

Manufacturable solutions exist, and are being optimized
Manufacturable solutions are known
Interim solutions are known
Manufacturable solutions are NOT known

Explanations of the technology requirements are as follows.

Asynchronous global signaling—(Objective Requirement) A key challenge in modern IC design is distribution of a centralized clock signal throughout the chip with acceptably low skew. Combining pre-designed modules on the chip, and providing reliable communication between independent modules, are also challenging. The asynchronous global signaling requirement refers to a design style where synchronization is performed through request and acknowledge lines, rather than through a global clock. The requirement grows with the power, congestion and area costs of traditional repeater insertion in long global lines. Globally asynchronous, locally synchronous (GALS) design develops coarse-grained functional modules using conventional design techniques, then adds local clock generators and self-timed wrappers so that modules can communicate using asynchronous handshake protocols. The number of handshake components determines the system complexity. The projected number of independent components is not projected to increase dramatically due to the associated coordination overheads, such as clock distribution. By 2012, further progress in asynchronous clocking will depend on commercial tool support. Emerging arbitration schemes (e.g., trees, the crossbar of Fulcrum Microsystems, etc.) are likely to evolve through 2014, improving latency within a design.

1. *Parameter uncertainty—(Control Requirement)* EDA tools manage the uncertainty of interconnect delay by creating tighter links between the logical and physical domains. While this works for interconnect, substantial electrical uncertainty in transistors remains unaccounted for. As a result, static timing analysis tools needlessly overestimate operational delay in the circuits, and introduce significant pessimism into the operational frequency of a design. This requirement highlights the need for accurate consideration of process variations and resulting parametric uncertainty in logic and circuit design. In the table, %-effect is predicted as a function of parametric variation in shrinking devices.
2. *Simultaneous analysis objectives—(Objective Requirement)* In modern technologies, many objectives and physical effects interact with each other, and must be simultaneously analyzed and optimized. In order, these objectives are: delay, power (active or dynamic), area, signal integrity, delta delay, delta power, printability, and reliability. Currently developed statistical methods will integrate yield optimization into existing design flows by 2008, and become mainstream in 2010 to recover up to 25% of development time and yield. Statistical methods are likely to involve simultaneous analysis of delta delay and delta power. Optimizations for reliable computing will extend existing techniques for robust computation (e.g., the Razor technique from the University of Michigan), and may be integrated into the design flow by as early as 2009. Productivity and cost will be part of the equation by 2013. Since productivity measures are only loosely defined today, and no mature optimization techniques are known, the requirement becomes red in 2013.

Number of circuit families in a single design—(Objective Requirement) High-performance integrated circuits often mix circuit families to achieve better speed, at the possible expense of area, power consumption, design effort, etc. The typical order in which circuit families are integrated is static CMOS, followed by multi-Vt, multi-Vdd, and dynamic CMOS. Other circuit types such as asynchronous, or retention FFs, are also making their way into product chips. It is difficult to predict the exact mix of circuit families that will prevail in the future. However, the ability to handle multiple circuit families on a single design will remain as an important requirement. At the same time, design complexity and fabrication difficulties that stem from mixing logic families, along with the low dimensionality of design constraints, will likely stabilize the number of different families on a single chip at a relatively small number.

Analog content synthesized—(Objective Requirement) Table DESN4 projects the amount of synthesized analog circuitry, as a percentage of a design's total analog content. See the AMS section of the *Systems Driver chapter* for more details. The trajectory of analog synthesis may recall that of logic synthesis 20 years ago: 1) today, it is on the brink of replacing designers; 2) by 2013, the significance of analog will compare to that of digital synthesis for microprocessors in the 1990s, and will apply to 25% of typical analog content; and 3) by 2020, current analog synthesis techniques will start saturating returns, calling for automation of new techniques just as digital synthesis now looks to automated datapath implementation and other extensions.

Adaptive/self-repairing circuits—(Objective Requirement) Nearly all circuits today cease to function properly when they incur any sort of damage. Manufacturing process variations also dramatically reduce reliability and yield of fabricated chips. In many cases, the tradeoffs between system size and complexity will still dictate that it is cheaper to create a compact, non-robust implementation of an electronic component and then replace it when it fails. However, in more and more contexts it is too expensive to access the system, or too expensive to communicate with the system to remotely diagnose its failure state and repair it. Hence, demand will increase for circuits that consume non-traditionally large areas, but are highly adaptive to internal failures. Such designs can help control costs of design verification, manufacturing and testing. Circuits that remodel and repair themselves are already are already manufactured (e.g., IBM eFuse technology to regulate voltage and frequency of failing parts) and will be more common in new technologies. Autonomic or self-managing computer systems will be vital in the face of continually rising operational and maintenance costs.

Full-chip leakage power—(Control Requirement) Power consumption is now the major technical problem facing the semiconductor industry. Leakage power (subthreshold, gate) increases exponentially as process moves to finer technologies. Techniques such as dynamic Vt, clock gating, power domains/voltage islands, dynamic voltage and frequency scaling, multiple Vt transistors, and body bias solutions will mitigate leakage until 2012. While current technologies still see dominance of gate leakage over subthreshold leakage, the latter is likely to become performance-limiting as high-k dielectrics bring gate leakage under control. The leakage requirement is normalized to 2007 values, and is based on scaling of inverter leakage across technology nodes.

LOGICAL, CIRCUIT, AND PHYSICAL DESIGN POTENTIAL SOLUTIONS

Logical, circuit, and physical design potential solutions are given in Figure DESN5. Explanatory comments are as follows.

3. *Automated handshake logic/circuit tools.* Needed for asynchronous and latency-insensitive global communication on chip.
4. *Synthesis and timing accounting for variability*. Synthesis that accounts for parametric variation of a technology; avoids overly pessimistic clock frequencies.
5. *Circuit/layout enhancement accounting for variability.* Power, interference, and error tolerance optimizations which account for parametric variations on a chip.
6. *Macro block/chip leakage analysis.* Accurate full-chip leakage estimation considering environmental variations.
7. *Power management analysis & logic insertion SOI SOC tools.* Techniques for power management at the logic level, e.g., power gating and clock gating – particularly in support of SOI technology.
8. *Analog synthesis (circuit/layout).* ASIC analog physical synthesis requires improved modeling methodology that is suited to structured synthesis flows. Multi-fabric synthesis tools also needed.
9. *Non-static logic implementation.* Logical and physical synthesis for dynamic circuit families, mixed static-dynamic logic.
10. *Cost-driven implementation flow.* Tighter integration of physical domain costs (parametric yield and unit costs, leakage and total power) into analysis and design space exploration flows. Analysis tools to decide between different implementation fabrics. Tools to extrapolate costs into different technology nodes. (Pre-layout) manufacturability analysis.

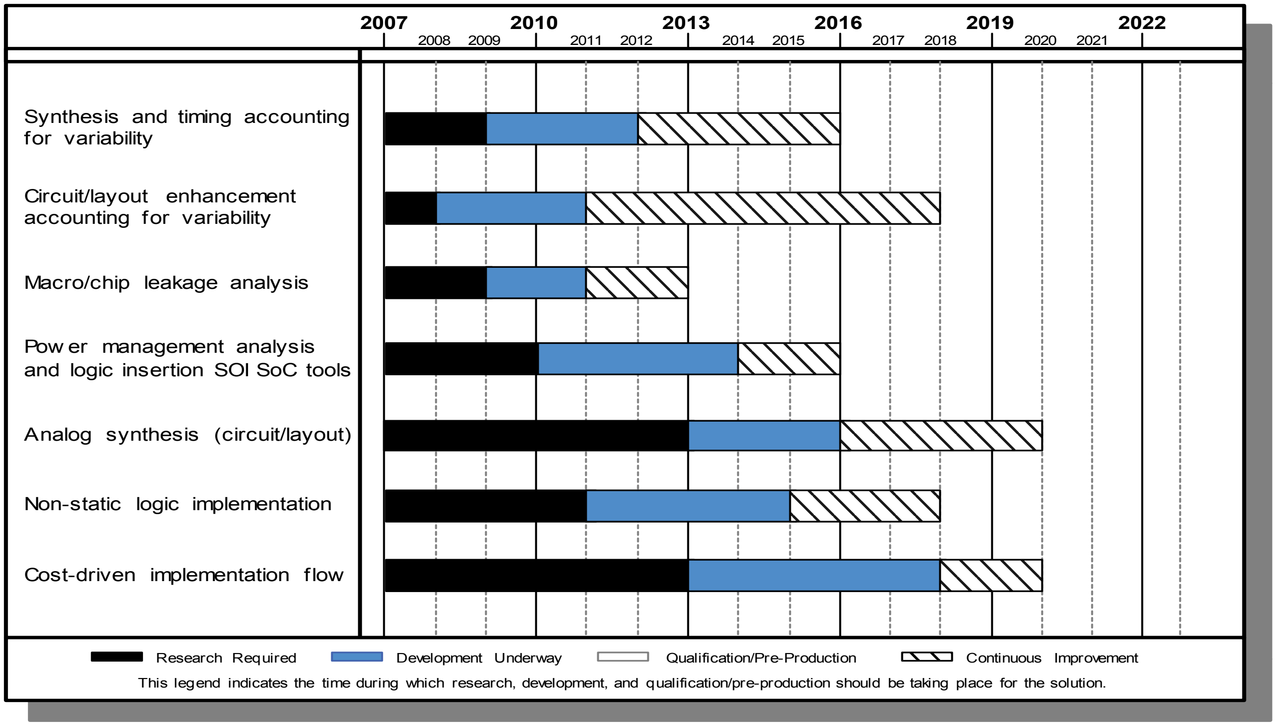

Figure DESN5 Logical/Circuit/Physical Design Potential Solutions

CORRESPONDENCE BETWEEN REQUIREMENTS AND SOLUTIONS

Correspondences between logical/circuit/physical design requirements and potential solutions are given in Table DESN5. For each requirement, potential solutions comprise the tools or methods that enable numerical targets to be met.

Table DESN5 Correspondence Between Logical/Circuit/Physical Requirements and Solutions

Requirement	*Solution*	*Explanation of the Correspondence*
Asynchronous global signaling *% of a design (SOC)*	Automated handshake logic/circuit tools	Departure from fully synchronous design paradigm needed for power reduction, latency insensitivity, variation-tolerance
Parameter uncertainty *%-effect (on signoff delay)*	Synthesis and timing analysis accounting for variability	Tools that account for process uncertainty, and resulting parametric uncertainty, will reduce guardbanding and increase chip yields
Simultaneous analysis objectives	Circuit/layout enhancement accounting for variability	Optimizations which consider parametric uncertainty
Simultaneous analysis objectives	Power management analysis and logic insertion SOI SOC tools	Requires budgeting of area/power/timing constraints
Simultaneous analysis objectives	Cost-driven implementation flow	Cost is an engineering parameter that affects turnaround times. Silicon cost no longer dominant; test and manufacturing costs increase emphasis on adaptive, self-repairing circuits
Circuit families *# of circuit families in a single design*	Non-static logic implementation	Non-static implementations help improving different chip parameters
Synthesized analog content	Analog synthesis (circuit/layout)	Allows for larger portions of a chip to be analog
Full-chip leakage	Macro block/chip leakage analysis	Enables accurate leakage predictions

DESIGN VERIFICATION

The overarching goal of functional verification is to ensure that a system's implementation fully matches its specification, that is, the planned intended behavior of the device. Unfortunately, due to the growing complexity of silicon designs, functional verification is still an unresolved challenge, defeating the enormous effort put forth by armies of verification engineers and academic research efforts. The current practice in industry is a partial verification process, developed under tight time and cost constraints, where only a few aspects of a design are addressed, checked, and verified. These aspects constitute only a vanishingly small fraction of the entire universe of possible design behaviors. The underlying reasons for this unmanageable complexity lie in the inability of verification to keep pace with highly integrated system-on-a-chip (SOC) designs and parallel chip-multiprocessor systems, paired with highly interconnected communication protocols implementing distributed computation strategies. Because of the extremely high costs associated with flawed microchips, industrial development efforts devote a significant fraction of engineering time and resources to achieve functional correctness. Multiple sources report that in current development projects verification engineers outnumber designers, with this ratio reaching two to one for the most complex designs. Part of this high resource allocation is due to verification methodologies that are still largely ad hoc and experimental, and to a lack of robust solutions.

Without major breakthroughs, design verification will be a non-scalable, show-stopping barrier to further progress in the semiconductor industry. There is hope for breakthroughs to emerge via a shift from ad hoc verification methods to more structured, formal processes. The mainstream methodology used in industry today attempts to verify the functionality of a system design by repeatedly building models, simulating them on a few selected vectors, and then patching any bugs that are exposed. To this end, logic simulation techniques predominate, because they can generate simulation vectors at a very high rate. However, the coverage of these tests is usually very low, and when a design error is found, the debugging phase entails manual analysis of very long and complex bug traces to narrow down the source of the problem. Additionally, these methods require substantial engineering effort to direct the verification activity toward specific design areas of critical quality, low coverage, etc. Traditional verification techniques are entrenched because the cost to transition to alternative methodologies is high. Hence, adoption of new and alternative formal and semi-formal techniques, which have started to become available, is progressing slowly.

More structured verification approaches organize the verification effort by generating a "golden model" of the system's intended behavior and comparing simulation outcomes with this model. Coverage metrics are collected to support some degree of confidence in the correctness of the design-under-verification. It is common to organize the verification activities hierarchically, by addressing first individual components (generally developed by a single designer), then the entire chip, and eventually the full system, so that bugs that are contained within a single unit can be addressed and solved earlier and more easily. (Due to the complexity of full system simulation, hardware emulation is sometimes used instead of simulation, particularly for large-market designs, where the additional costs can be easily absorbed. Hardware emulation buys several orders of magnitude of performance improvement in "simulation" speed, and enables an early start to system integration and software development. At the same time, emulation represents only a constant-factor improvement.) The most complex aspect of this methodology is the verification of the communication interface between components; late or escaped bugs are often found in complex unverified interactions between units. Finally, there is a growing interest for formal and semi-formal verification techniques: today, IC development teams are exploring new methodologies where mainstream validation is complemented by semi-formal methods, using a handful of commercially available tools. This is a rapidly evolving arena: several new solutions, often integrating formal methods with simulation-based techniques in new and creative ways, appear each year both in academic environments and in new industrial developments.

Post-silicon validation focuses on the validation of silicon devices after tapeout and before customer shipping. This is an area of verification which has grown rapidly in the past decade, mostly due to the large complexity of silicon-based systems, which, in turn, leads to many potential flaws that can only be exposed after system integration. The goals of post-silicon validation range from detecting logic and design bugs, which should manifest in each manufactured part, to expose electrical and process-related issues, which usually manifest only in a fraction of the components and, finally, to identify and exclude rare random manufacturing defects. The process involves building full prototype systems with the manufactured components, and run software test programs, ranging anywhere from embedded software to high level applications. Two key characteristics of post-silicon validations are 1) the high execution speed which enables the verification team to evaluate a system over several order-of-magnitudes more cycles than it is possible in pre-silicon simulation (e.g., a validation suite that requires months of runtime on a simulator or emulator could finish in minutes), and 2) the limited visibility inside the manufactured silicon parts. In fact, internal signals cannot be probed any longer as they could in a pre-silicon simulation environment, making test validation and debugging much more challenging operations. These inspection tasks are usually performed by accessing and probing the device's external signals and register interfaces using specialized tools, such as logic analyzers and oscilloscopes.

DESIGN VERIFICATION REQUIREMENTS

Table DESN6 quantifies design verification technology requirements needed to support design complexities in future technology nodes. The technology metrics evaluate both the ability to guarantee the correctness of a design and the amount of effort spent in verification within a development project. Table values are for a new development design project, as opposed to a proliferation within an existing family of designs. If verification productivity is to scale in proportion to design complexity, the classification of verification software into formal verification versus simulation will over time evolve into a dichotomy of hybrid versus semi-formal verification solutions. Table DESN6 estimates, within integrated hybrid solutions, the fraction that will be simulation-based as opposed to formal verification-based. The increasing impact of software and mixed-signal components, and the importance of electrical effects due to the reduction in feature sizes, together imply that verification of the interaction between heterogeneous components and the system as a whole will require an increasing fraction of overall verification effort. Finally, the shift toward a more structured approach to verification demands efforts toward the formalization of a design specification, which in turns leads to improved automation of the entire verification process.

We divide verification development into three segments: 1) code newly developed specifically for the design being verified, 2) code acquired from third parties (i.e., verification IP), and 3) code reused from previous designs within the same company. Table DESN6 reports the first two components under "Reuse"; the third can be calculated by subtracting the other two rows from 100%. The last two rows in the table estimate the growing importance of functional coverage. While traditional coverage techniques focus on code and state coverage, functional coverage captures more directly the relevant aspects of a design's functionality that need to be verified, but it requires additional engineering effort because of its domain-specific character. The need to quantify the progress of verification in the near- and long-term future demands improved deployment of this technique. Although expertise within engineering teams in developing and reusing functional coverage is growing, the table indicates that a more pervasive deployment is needed in the future.

Table DESN6a Design Verification Requirements—Near-term

Year of Production	*2007*	*2008*	*2009*	*2010*	*2011*	*2012*	*2013*	*2014*	*2015*
Productivity									
Design size verifiable by 1 engineer-year (in millions of transistors - based on an SOC design and a 10-person engineering team) [1]	7.9	10.3	13.5	17.6	23.1	30.3	39.8	52.3	69.6
Methodology									
Design errors exposed using formal or semi-formal verification (%, versus simulation)	4.7	7.1	9.4	11.8	14.1	16.5	18.8	21.2	23.5
Effort spent on system-level verification: software, hardware and electrical effects (%)	11.6	13.1	14.7	16.3	17.8	19.4	20.9	22.5	24.1
Portion of the design specification formalized for verifiability (%)	13.8	17.5	21.3	25.0	28.8	32.5	36.3	40.0	43.8
Bugs									
Escape rate: bugs found after first tapeout (per each 100K lines of design code)	8	7	7	7	6	6	6	6	5
Bugs found after system integration until tapeout (per each 100K lines of design code)	62	68	74	79	85	91	97	103	109
Reuse									
Portion of the verification infrastructure (e.g., test beds, coverage, checkers) which is newly developed (versus reused components and acquired IP) (%) [2]	73.9	70.8	67.8	64.7	61.6	58.6	55.5	52.5	49.4
Portion of the verification infrastructure which is acquired from third parties (i.e., verification IP) (%) [2]	15.5	18.3	21.1	23.8	26.6	29.4	32.1	34.9	37.6
Functional coverage									
Portion of design for which verification quality is evaluated through functional coverage (%)	46.5	49.7	52.9	56.2	59.4	62.6	65.9	69.1	72.4
Coverage goal density (expressed as number of coverage goals for each million transistors of the design) [3]	1294	1608	1922	2235	2549	2863	3176	3490	3804

Table DESN6b Design Verification Requirements—Long-term

Year of Production	*2016*	*2017*	*2018*	*2019*	*2020*	*2021*	*2022*
Productivity							
Design size which can be verified by 1 engineer-year (in millions of transistors - based on an SOC design and a 10-person engineering team) [1]	91.8	121.0	159.7	210.9	278.6	368.5	487.6
Methodology							
Design errors exposed using formal or semi-formal verification (%, versus simulation)	25.9	28.2	30.6	32.9	35.3	37.6	40.0
Effort spent on system-level verification: software, hardware and electrical effects (%)	25.6	27.2	28.8	30.3	31.9	33.4	35.0
Portion of the design specification formalized for verifiability (%)	47.5	51.3	55.0	58.8	62.5	66.25	70
Bugs							
Escape rate: bugs found after first tapeout (per each 100K lines of design code)	5	5	4	4	4	3	3
Bugs found after system integration until tapeout (per each 100K lines of design code)	115	121	126	132	138	144	150
Reuse							
Portion of the verification infrastructure (e.g. test beds, coverage, checkers) which is newly developed (versus reused components and acquired IP) (%) [2]	46.4	43.3	40.2	37.2	34.1	31.1	28.0
Portion of the verification infrastructure which is acquired from third parties (i.e., verification IP) (%) [2]	40.4	43.2	45.9	48.7	51.5	54.2	57.0
Functional coverage							
Portion of design for which verification quality is evaluated through functional coverage (%)	75.6	78.8	82.1	85.3	88.5	91.8	95.0
Coverage goal density (expressed as number of coverage goals for each million transistors of the design) [3]	4118	4431	4745	5059	5373	5686	6000

Manufacturable solutions exist, and are being optimized
Manufacturable solutions are known
Interim solutions are known
Manufacturable solutions are NOT known

Notes for Tables DESN6a and b

All values, except for the first row (productivity) are linearly increasing or decreasing trends, where the initial values in 2007 are derived from industry survey data, and the final values are estimated.

[1] The productivity requirement is derived from the SOC design productivity requirement of the System Drivers chapter, and divides it by the percentage of design effort spent on verification, so as to obtain the amount of logic that can be verified in a year. The percentage of the design effort spent on verification has a linear trend starting at 70% in 2007 and ending at 50% in 2022.

[2] Verification reuse is modeled as: new verification infrastructure + reused + 3rd party verification IP = 100%. The table provides values for the new verification infrastructures and acquired 3rd party IP; these two values determine the proportion of reused verification components.

[3] In other words, the number of assertions or checkers inserted in a design or referring to an aspect of the design, for each corresponding million of transistors of logic generated when synthesizing that design.

DESIGN VERIFICATION CHALLENGES

Many of the key challenges for verification are relevant to most or all system drivers. In the near term, the primary issues are centered on making formal and semi-formal verification techniques more reliable and controllable. In particular, major advances in the capacity and robustness of formal verification tools are needed, as well as meaningful metrics for the quality of verification. In the longer term, issues focus mainly on raising the level of abstraction and broadening the scope of formal verification. These longer-term issues are actually relevant now, although they have not reached the same level of crisis as other near-term challenges. In general, all of the verification challenges apply to SOC. MPUs present a distinct set of issues, both because of their leading-edge complexity, and because of the unique economics of an incredibly complex design family that is produced in incomparably high volumes. As a result, different domain-specific verification challenges and opportunities exist, both in the near- and long-term.

Capacity—Today, verification techniques can be grouped in two main families: formal verification and simulation-based. Both have crucial downsides: while formal tools can only handle small to medium size designs, simulation-based tools can simulate designs of almost arbitrary complexity, but they provide a vanishingly small coverage even with extremely long simulation times. Emulation and rapid hardware prototyping perform several orders of magnitude faster than software logic simulators, thus providing the ability to achieve higher coverage. However, the improvement is only a constant factor and does not scale at the same rate as design complexity. Post-silicon validation is concerned with the overall correct functionality of the system composed of silicon hardware and application software, and as such it operates at the top level of design complexity integrating all components together. The challenge in this space is in creating new solutions, probably integrating the positive aspects of both families of approaches, that can provide high coverage at all hierarchical levels of a design.

Robustness—A crucial aspect of current verification solutions is their reliability. On one hand, simulation-based methods are fairly predictable, because their execution time per simulation vector scales linearly with design complexity. Their performance, thus, will linearly decrease as we move towards the most complex design levels. Emulation techniques present a similar trend, while silicon prototypes are consistently reliable. On the other hand, formal verification techniques depend on highly temperamental heuristics, in order to cope with the complexity of the verification problem. For any given pair of design and verification algorithms, even an expert can be hard-pressed to determine whether the verification algorithm will complete. Common measures of problem size, such as transistor, gate, or latch counts, correlate only vaguely with formal verification complexity; it is fairly easy to find designs with less than one hundred latches that defy all known verification methods, as well as designs with thousands of latches that can be verified easily. Such unpredictability is not acceptable in a production environment. A crucial verification challenge is to make the verification process more robust. This can take the form of either improved heuristics for the verification algorithms, or improved characterization of the difficulty of verifying a given design, leading to methodologies for easy-to-verify designs.

Verification metrics—An important near-term verification challenge is the need to quantify the quality of the verification effort. In particular, a meaningful notion of coverage is needed. Multiple types of coverage metrics are available, each with their own benefits and limitations. *Code coverage* is concerned with measuring which fraction of the design source code has been stimulated during simulation (for example, lines of code, expression coverage). Its main limitation is in its projecting a dynamic simulation execution to a static source code, thus missing many potential problems that could be still present because of all the possible paths of execution available during simulation. *Structural coverage*, such as state or pair arc coverage, focuses on the finite state machines of the design and measures which states have been covered by simulation. However, it is generally not possible to consider the whole design as a single finite state machine, rather only a few machines are considered by coverage. The downside is that while the coverage provides good results for single machines, combinations of states resulting from products of multiple machines are not considered. *Functional coverage* targets each of the design's functionalities. Since these differ for each distinct design, the specific definition needs to be provided by the verification team based on the design's specification. The quality of the result is thus dependent also on the quality of the coverage definition. In the current landscape, there is a lack and a need at the same time for a common foundation metric for functional coverage in a unified environment, so that achieving a certain level of coverage in a design could have meaning beyond the specific coverage used in that particular design. Moreover, there is no general methodology in designing functional coverage units; an abstract fundamental model to use as a guideline during the development is needed and could support the development methodology. Ultimately, there is a need for a defect model for functional bugs, much like there is a defect model for testing, to support the development of coverage and a unified metric for verification.

Software—The verification of complex systems, such as systems-on-a-chip (SOC) entails the verification of the hardware components, of the hardware/software interface and of the application software running on the system. The software components of an SOC can be divided into 1) application software, 2) hardware-independent layer, such as an operating system, which controls the execution of the application, and 3) lower hardware-dependent software, such as drivers, etc. Because in these systems the software layer can provide much of the functionality, a major challenge in SOC verification is how to verify the software and the hardware/software interface. Presently, software development is not as rigorous as hardware development in terms of design reviews, tools for analysis and testing. Software is intrinsically harder to verify: it has more complex, dynamic data and a much larger state space. The most common software verification technique in use today is "on-chip-verification", which entails running the software on a production version of the hardware components. While it allows very fast simulation, as it is required by the intrinsic complexity of software, its downside is that software verification can only start very late in the design cycle. Classical formal techniques for software verification are still too labor-intensive to be widely applicable for SOC, that is, systems with such large state spaces, and require very aggressively abstracted models of the software application. The verification of the hardware/software interface is a challenge on its own, since it requires verifying the two domains together. To make this task more manageable, there is a need for techniques which provide proper abstractions of the interface activity, solutions to check the correctness of the abstracted driver layers, and assertion checking tools for drivers' invariants at the non-abstract level. The near-term challenge will be to develop techniques that allow verification of even elementary and low-level pieces of software. The longer-term challenge will be to develop robust verification methods for software, and hardware/software interfaces, as well as an understanding of design-for-verifiability as applied to software.

Reuse—Pre-designed IP blocks promise to allow assembling SOCs of unprecedented complexity in a very short time. The major challenge is in developing the corresponding verification methodology to allow rapid verification of a system assembled from pre-designed (and pre-verified) blocks. Key issues are how to rigorously and completely describe the abstract behavior of an IP block, how to describe the environmental constraints assumed by the IP block, and how to exploit the hierarchy to simplify verification. Some IP components have started to ship with associated verification IPs, from standard protocols verification IPs, to abstract models for general IP blocks, protocol checkers to check environmental constraints surrounding the block, to transaction generators. However, these are still preliminary attempts, while the need is for consistent availability of verification IPs associated with any IP block in order to develop a reuse methodology. Besides verification components associated with IP blocks, there is some availability of independent verification IPs, such as environment generators for specific protocols. Near-term progress will most likely occur for standardized IP interconnects, such as on-chip buses, but the general problem for arbitrary IP block interfaces must be eventually solved.

Specialized verification methodology—The biggest issue in design verification is that all currently known algorithmic solutions are running out of capacity with respect to the designs being developed today. The only foreseeable way to overcome this issue in the short term is with an adequate verification methodology. Current trends in this direction include coverage-driven verification, both for simulation-based verification and semi-formal verification, more in use today than in the past; specification documents in formal notation, that make easily available the set of formal properties to be verified in the implementation; coverage model templates. Many challenges are still to be solved to obtain a sufficiently robust and complete methodology: There is a need for ways to obtain consistent abstraction techniques of design components, interfaces, etc. that do not miss key aspects of the design in the abstraction. Formal specifications are starting to be developed; however, a major limitation is the completeness of such specifications, a challenge that becomes even more compelling in a world where different IP components in the same system are developed by completely unrelated design teams. The acceptance of new verification methodologies is progressing slowly; even the basic deployment of formal properties in a design is a challenge since it often requires a global understanding of some specific aspect of a design, which involves additional effort on the development team. Finally, the verification methodology is ultimately an evolving target that can only provide a risk reduction to the development, not a guarantee of correctness through a well-defined recipe.

Soft and permanent failures—The need for verification techniques which can provide information on a system's reliability in presence of soft errors and permanent transistor failures is become increasingly acute. Industry experts are warning repeatedly of the decaying reliability of the silicon substrate due to extreme transistor scaling. In this space, initial solutions are already available which can harden the storage elements of a design to be resilient to SEU faults. In addition a few research-level solutions have proposed protection against permanent transistor failures (whether in the field or at birth) at very low additional area cost. These solutions are also viable to protect manufacturer against lowering yield, due to the high fraction of defective manufactured parts.

Design for verifiability—A few specialized activities in the area of design for verifiability are already foreseeable, enabling more effective verification. For instance, facilities for software and/or hardware debug can be put in silicon. In this direction there is preliminary work being done on self-checking processors, in which a small watchdog processor or a network of distributed hardware checkers verifies the correct execution of the main processor. For mixed-signal designs, the insertion of loop-back modes to bypass the analog portion of the design allows to verify the system as a fully digital design. The use of synchronizers in multi-threaded systems forces the tasks to not proceed independently past the synchronization points, creating checkpoints for verification and reducing the searchable state space. In MPU designs synchronizers would reduce the complexity of verifying speculative execution systems. The challenge in this area is the development and the adoption of design-for-verifiability techniques for a few major domains. Efforts are also made in developing design methodologies by incremental refinement to produce systems that are correct-by-construction; however, it is not clear how automatic the refinement process can be made, while, on the other hand, manual intervention is a potential source of design errors. In the longer term, major changes in methodology may be required, and some performance degradation is likely.

Specification for verifiability—How to specify the desired behavior of a design is a continuing challenge in design verification. Current available notations for specification are not powerful enough to approach this problem in a generalized way. A deeper understanding of what makes a specification clear or opaque, modifiable or intractable will be needed to guide development of languages that are used to specify ever more complex designs. For instance, there is a need for automatic ways to check the self-consistency of a specification document, so that different specifications don't state conflicting requirements. In addition, specific training is needed for designers to use these notations and to be able to develop formal specifications consistently.

New kinds of concurrency—As MPU designs become more complex, new kinds of concurrency become important. Already, many of the bugs that elude verification relate to cache coherence and other concurrency issues. New designs greatly complicate the verification process by increasing the level of concurrency via techniques such as chip-level multiprocessing and on-chip cache coherence protocols, and simultaneous multithreading. In the future, new kinds of concurrency both at the intra-processor level and in multi-processor systems, or in other hardware context will present difficult challenges to verification. There is a need for new models of failure to grasp this additional level of complexity. The solution will probably require a mix of hardware and software techniques to reduce the complexity of the interactions and make the concurrent protocols verifiable.

Higher levels of abstraction—As design moves to a level of abstraction above RTL, verification will have to keep up. The challenges will be to adapt and develop verification methods for the higher-levels of abstraction, to cope with the increased system complexity made possible by higher-level design, and to develop means to check the equivalence between the higher-level and lower-level models. This longer-term challenge will be made much more difficult if decisions about the higher-level of abstraction are made without regard for verification (e.g., languages with ill-defined or needlessly complex semantics, or a methodology relying on simulation-only models that have no formal relationship to the RTL model).

Verification in presence of non-digital effects—To date, design verification has mainly focused on the discrete behavior of digital systems. The dual challenges of silicon complexity and system complexity will force future verification efforts to analyze a broader class of aspects. The complexity of silicon integrated circuit systems is making the clean, digital abstraction of a VLSI system increasingly precarious. Analog electrical effects will impact performance and, eventually, functionality. The existing simulation methodology (SPICE) for analyzing these effects is too slow, and may become unreliable as smaller devices become increasingly sensitive to process variations. In this direction there is preliminary work being done on architectural simulation of microprocessors, in which the high-level architectural simulator, interacts with a low-level context-specific simulator to acquire metrics such as timing and voltage, which are then fed back for overall system evaluation with minimal performance impact. In the long term, formal techniques will be needed to verify these issues at the boundary of analog and digital, treating them as hybrid systems. (N.B.: Hybrid systems have both complex discrete behavior (e.g., finite-state machines) as well as complex continuous behavior (e.g., differential equation models). The discipline borrows techniques from both discrete formal verification as well as classical control theory.) Similarly, at the highest-levels of design, system complexity dictates that future verification tasks will likely require specifying and verifying analog and probabilistic behaviors (such as quality of service guarantees in a network processor). Thus, there will be the challenges of hybrid systems and probabilistic verification.

Heterogeneous systems—The development of new technologies that are placed side-by-side with a digital design in a silicon die presents a whole set of new challenges. Examples are MEMS, electro-optical devices, and electro-biological devices. These new components will require modeling of both the interface between the digital portion and the non-digital components and a proper abstraction of the non-digital system behavior in order to still be able to verify the digital portion of the system.

Analog-Mixed signal—Today, analog systems are mostly verified through classic systems analysis tools for continuous systems, through system modeling and analysis in the frequency domain. The bulk of verification happens in the post-design phase, by verifying test dies with analog lab equipment. Mixed-signal designs undergo separate verification activities for the digital and analog portions. In the future, mixed-signal systems will become a more relevant fraction of all silicon developments, bringing the development of proper verification methodologies in this arena to a critical level. The challenge in this area is in tying the verification of the analog portion of a system with its digital counterpart. One of the requirements in achieving this goal is in bridging the current performance gap between digital and analog simulation.

DESIGN VERIFICATION SOLUTIONS

We now present some of the solutions that are available or are being developed to address the above challenges. Figure DESN6 summarizes key directions in which the verification crisis may be attacked, along with expected time frames of availability to development teams.

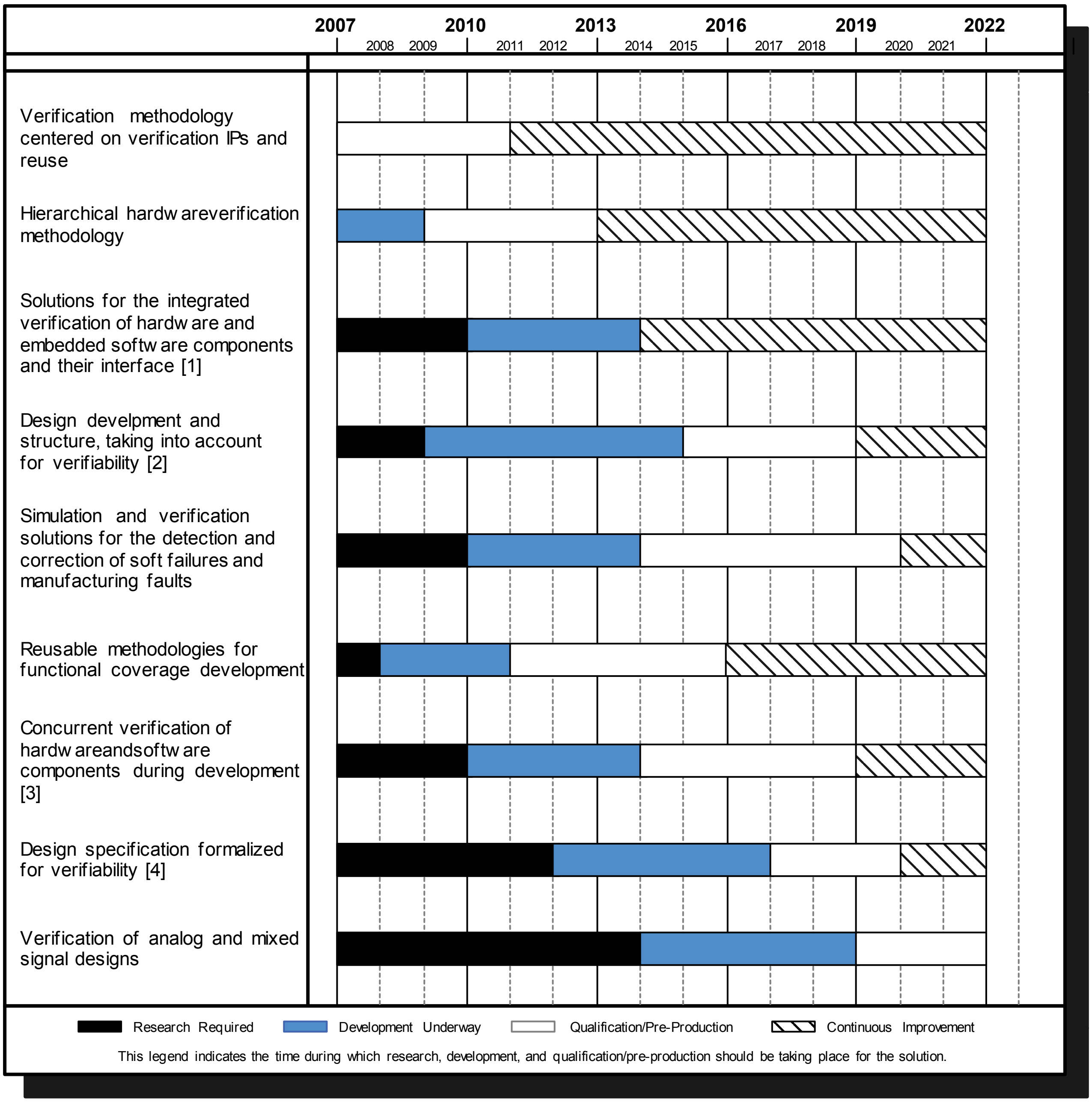

Figure DESN6 Design Verification Potential Solutions

Notes for Figure DESN6.

[1] This entails verification techniques for high-level integrated HW/SW systems. Current design technologies are starting to automate the process of partitioning the functionalities of a system that should be implemented in hardware and those that should be implemented in software. Typically, the result is a high-level description of the system's components and whether they should be implemented in embedded software or in hardware. The complexity of such high-level descriptions, however, is growing: there is a need for solutions which provide validation of a system at this stage of development and, in particular, of the correctness of the high-level interaction protocols.

[2] In other words, techniques for developing a hardware design so that it is well suited for verification.

[3] In other words, verification solutions which address the correct interaction between the hardware layer and the embedded software layer. Currently available solutions verify these two layers separately: 1) the hardware layer is verified mostly through classic validation techniques, and 2) embedded software is verified by running the software on a simulation model of the hardware (which is usually still under development). Due to the rapidly growing complexity of embedded software applications, this is becoming an increasingly pressing problem that requires specialized solutions.

[4] (Possibly formal) techniques to develop a specification for a design in a way that facilitates verification of the specification itself.

Verification IPs are already available today, and they are part of a growing methodology trends to boost the productivity of the process of verification. In the future this approach will be much more widespread, in parallel with the aggressive deployment of third party IPs in SOC designs. As discussed above, solutions structuring a hierarchical and integrated hardware/software methodology are being researched today and will start to become available in the next few years. The other solutions outlined in the table are still at a conceptual stage today. Formal methodologies to develop easy-to-verify designs and to correct soft errors and permanent transistor defects have seen an initial exploration in the MPU domain, as mentioned in the previous section. Techniques for structural functional coverage and specification are even further from becoming mainstream solutions, however, they are crucial in solving the design-verification productivity gap. Table DESN7 establishes correspondences between design verification requirements and solutions.

Table DESN7 Correspondence Between Design Verification Requirements and Solutions

Requirement	*Solution*	*Explanation of the Correspondence*
Productivity of verification tasks	Verification methodology centered on verification IPs and reuse	Verification IPs and reuse reduce the amount of new verification development required in a project
	Hierarchical hardware verification	Structured methodologies improve design team productivity
	Reusable methodologies for functional coverage development	Functional coverage is time-consuming, and specific for each distinct design; development of reusability techniques is critical to boosting productivity
	Concurrent verification of hardware and software components during development	Advancing the verification of hardware in parallel with that of software components can significantly shorten time-to-market of a product, in contrast to methodologies that begin software verification only after the first hardware prototype
Formal and semi-formal verification centered methodology	Hierarchical hardware verification methodology	Enables the decomposition of the system into smaller blocks which are suitable for formal verification
	Design development and structure taking into account verifiability	Design for verifiability organizes a design so as to simplify verification; additional verification-specific hardware structures further simplify design-time verification tasks
Methodologies for system-level verification	Verification methodology centered on verification IPs and reuse	Verification IP components enable an early start on system-level verification
	Integrated verification of hardware and embedded software and their interface	Directly provides solutions for effective system-level verification
Portion of design specification formalized for verifiability	Design specification formalized for verifiability	Formal languages and methodologies to support the formal specification of a design
Escape rate after tapeout	Design structure taking into account verifiability	Development of hardware structures (checker-like) which can be used to detect and correct a system entering an escaped erroneous configuration after customer shipment
System integration bug rate	Analog and mixed-signal verification	Limits the bug rate due to analog effects
	Simulation and verification solutions for the detection and correction of soft failures and manufacturing faults	Manufacturing faults occurring in post-silicon are detected at system-level integration; techniques to detect and correct electrical and transient defects reduce the effort required to expose and correct these problems.
	Hierarchical verification methodology	Supports management of complexity through decomposition
Functional coverage	Reusable methodologies for functional coverage development	Reusable functional coverage solutions leverage the coverage development effort and boost quality of results

DESIGN FOR TEST

Nanometer process technology, increasing clock rate, mixed analog-digital-RF circuits, and highly integrated SOC and SIP present severe challenges to DFT. The test industry must cope with an enormous spectrum of problems ranging from high-level test synthesis for component-based design to noise/interference and power dissipation in extremely high-performance analog and RF subsystems. Many problems can be solved only if proper testability and DFT are considered and incorporated early in the design phase. Furthermore, the methodology precepts above note a natural evolution of analyses into verifications into tests, and the need for deeper unification of design with test. Effort and results (flows, vectors, fault models, sensitivities) from analyses in logical-circuit-physical implementation, and from design verification, must be reused during design and test. Across industry segments as varied as memory, digital signal processing (DSP), PE, SOC, analog/mixed-signal/RF, and MPU, high-level test challenges demand significant expansion of on-chip DFT, BIST, and testability features, and better integration to a pre-planned manufacturing test equipment set during the chip planning phase of development.

The DFT technology requirements and potential solutions vary widely based on the specific subsystems and yet they all have to converge at the system level so that SOC and SIP are testable with lower manufacturing cost and little performance degradations. Using system drivers as the common framework, DFT requirements and solutions are described in this section. Table DESN8 summarizes DFT technology requirements and Figure DESN7 summarizes DFT potential solutions for four major classes of system drivers based on specific design technologies: analog/mixed-signal/RF system driver, MPU/PE/DSP system driver, memory system driver, and generic SOC/SIP system driver.

Analog/Mixed-signal/RF system drivers—Analog, mixed-signal, and RF subsystems, together with component I/O speed, have become as important to system performance as core clock frequency or transistor and architectural performance. Currently, the industry is facing serious problems regarding RF measurements beyond 5 GHz. It is very expensive to test parts in excess of 5 GHz at wafer level or package tests—special testers or added hardware modules are needed to perform such tests, with added cost and test time. Designs at 10 GHz and beyond are being generated but there are no production testers capable of testing at those frequencies. This problem is expected to worsen in the future. It is imperative that new techniques be developed that allow high-frequency mixed-signal circuitry (10–30 GHz) to be fully tested and characterized for complex test specifications using low-cost testers. At the same time, new I/O protocols are being introduced and extended to the multi-GHz range. These I/O schemes are not only faster but also more complex, with source synchronous, differential, and even simultaneous bidirectional schemes operating at Gbit/s rates with differential voltage swings one-tenth of the supply V_{dd} range. By contrast, ATE and component test legacies include common clock-based testing and I/O measurements in the MHz range. Hence, I/O speeds and protocols drive significant instrumentation, materials, and cost challenges to the ATE equipment, interface hardware, and test sockets used for both design verification and manufacturing test. This inflection point demands broad industry development and application of on-die testability capabilities specifically for I/Os.

DFT methods for these analog/mixed-signal/RF subsystems are under intense research and development efforts, but to be successful, these DFT circuits must use digital components as much as possible to reduce the design effort and improve robustness as well as noise immunity. The results from these DFT methods, whether they are Pass/Fail indicators or parametric measurements, must also be correlated to standard specification-based test methods to gain credibility and designers' acceptance. This correlation is even more critical before the complexity and performance of analog/mixed-signal/RF subsystems increase to the point where they will no longer be testable using any external ATE or instrumentation.

Analog/mixed-signal/RF reliability issues become more important due to parametric degradation induced over time or by the operating environment. A comprehensive understanding of parametric issues requires well-characterized fault models for these circuits, especially to account for soft faults, noise-induced performance issues (crosstalk, substrate noise), process variations, thermal effects, etc. Fault models, whenever possible, must be physically correlated to defect models so that process quality and control can be improved to reduce and eliminate defects and improve yield. It is crucial to develop meaningful figures of merit for test quality of analog test techniques, including analog self-test. Many analog failures are due to parameters out of specification ranges, measured in continuous variables (time, voltage, phase, etc.), and arising from manufacturing variations or mismatch. Fault models for efficient and effective analog fault grading and test generation will be required. For analog/mixed-signal/RF designs, tools are needed that can minimize the computation complexity of fault simulation while maintaining high simulation accuracy. The requirements of analog fault models and process defect models are quite difficult to meet but they are paramount in future products integrating analog/mixed-signal/RF subsystems into SOC and SIP.

Table DESN8a Design for Test Technology Requirements—Near-term Years

Year of Production	*2007*	*2008*	*2009*	*2010*	*2011*	*2012*	*2013*	*2014*	*2015*
System Driver: Analog/Mixed-signal/RF									
All-digital DFT for analog/mixed-signal/RF circuits and systems. % digital circuits in DFT implementations	40	45	50	55	60	60	60	60	80
Correlation of DFT results with existing specification-based test methods. % results correlated	40	45	50	55	60	60	60	60	80
Availability of fault/defect models for DFT-oriented test methods. % AMS/RF blocks with accepted fault models	25	30	35	40	45	50	55	60	65
System Drivers: MPU/PE/DSP									
DFT coverage of digital blocks or subsystems. % blocks with DFT	70	70	70	75	75	75	80	80	85
DFT for delay test of critical paths. % paths covered	55	55	60	60	60	60	70	70	70
DFT for fault tolerance in logic blocks. % blocks with fault tolerance	40	40	45	45	50	50	55	55	60
System Drivers: Memories									
DFT for yield improvement.	85	90	90	90	90	95	95	95	95
General SOC/SIP requirements									
DFT support for logic and other non-memory circuit repair. % blocks with repair	50	60	60	60	70	70	70	80	80
DFT reuse for performance calibration, and measurement purposes. % DFT circuits reused	35	35	40	40	40	45	45	50	50
DFT impact on system performance (noise, power, sensitivity, bandwidth, etc.). % performance impact (aggregate figure of merit)	15	15	10	10	10	10	10	10	10
DFT efficacy in test volume reduction. Reduction factor	5×	5×	5×	10×	10×	10×	20×	20×	20×
DFT / ATE interface standard, including DFT control via standard test access protocols. % of test interface standardized	45	45	50	50	60	60	70	70	75

Table DESN8b Design for Test Technology Requirements—Long-term Years

Year of Production	*2016*	*2017*	*2018*	*2019*	*2020*	*2021*	*2022*
System Driver: Analog/Mixed-signal/RF							
All-digital DFT for analog/mixed-signal/RF circuits and systems. % digital circuits in DFT implementations	85	90	90	100	100	100	100
Correlation of DFT results with existing specification-based test methods. % results correlated	85	90	90	100	100	100	100
Availability of fault/defect models for DFT-oriented test methods. % AMS/RF blocks with accepted fault models	70	75	80	85	90	95	100
System Drivers: MPU/PE/DSP							
DFT coverage of digital blocks or subsystems. % blocks with DFT	85	90	90	95	95	97.5	100
DFT for delay test of critical paths. % paths covered	80	80	90	90	100	100	100
DFT for fault tolerance in logic blocks. % blocks with fault tolerance	65	70	80	90	100	100	100
System Drivers: Memories							
DFT for yield improvement.	98	98	98	100	100	100	100
General SOC/SIP requirements							
DFT support for logic and other non-memory circuit repair. % blocks with repair	80	90	90	100	100	105	110
DFT reuse for performance calibration, and measurement purposes. % DFT circuits reused	60	60	70	70	70	72.5	75
DFT impact on system performance (noise, power, sensitivity, bandwidth, etc.). % performance impact (aggregate figure of merit)	5	5	5	5	5	5	5
DFT efficacy in test volume reduction. Reduction factor	20×	50×	50×	50×	50×	50×	50×
DFT / ATE interface standard, including DFT control via standard test access protocols. % of test interface standardized	90	80	90	100	100	100	100

Manufacturable solutions exist, and are being optimized
Manufacturable solutions are known
Interim solutions are known
Manufacturable solutions are NOT known

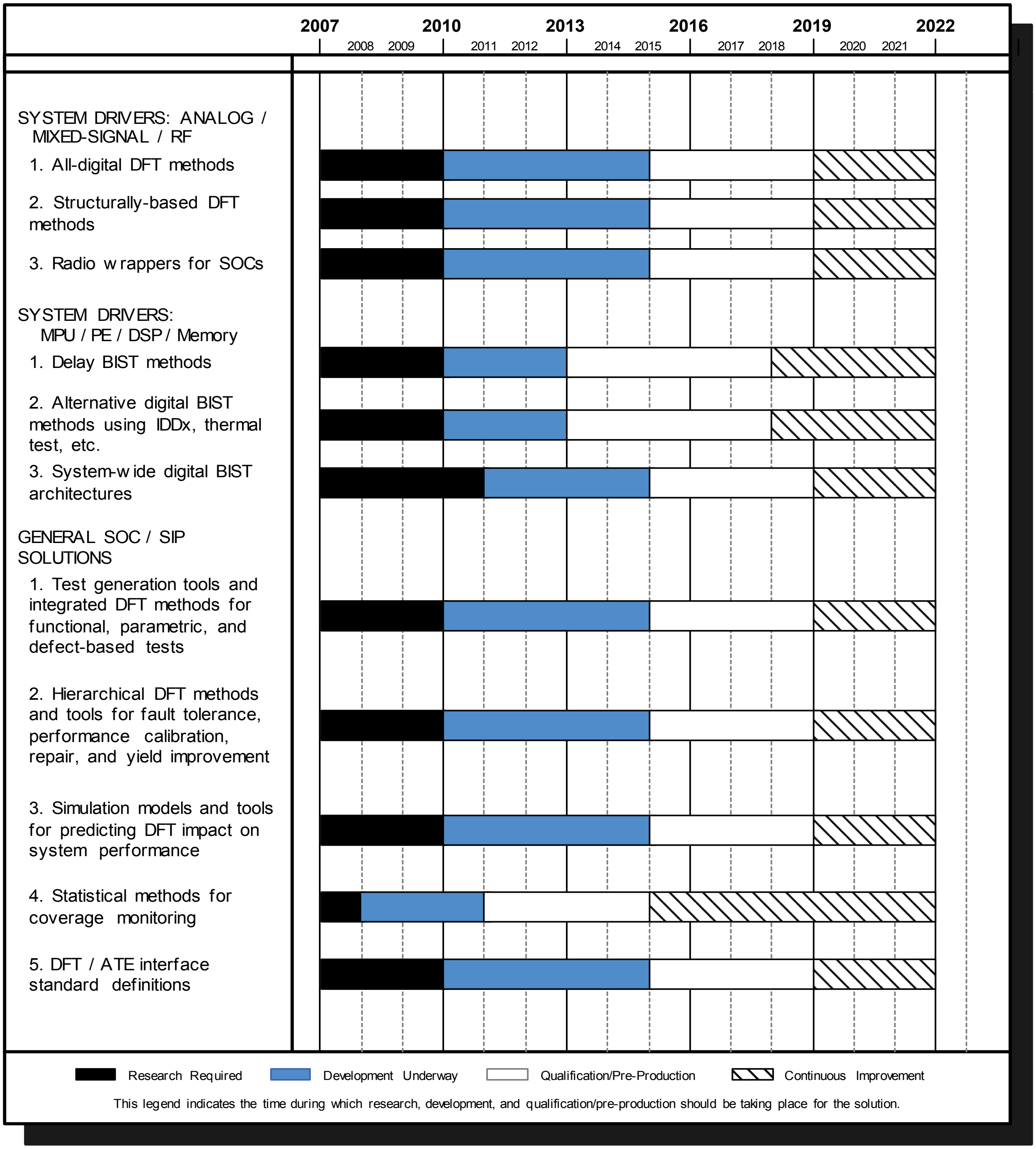

Figure DESN7 Design for Test Potential Solutions

The potential solutions to these requirements are just as numerous as the subsystems. Diverse approaches have been developed by industry and by academia to solve the analog/mixed-signal/RF DFT problems, many of which focus on specific product requirements (such as phase-locked loop (PLL), jitter BIST, converter BIST, transceiver DFT). While convergence of analog/mixed-signal/RF DFT methods is not expected or desirable since a DFT method must be selected to optimize the overall electronic system design and test for each product class, several essential features common to all

DFT methods are critical to the overall system-level solution. All-digital DFT methods will be the preferred solution due to the characteristics mentioned above: robustness, ease of design, ease of integration, and ease of developing supporting computer-aided design (CAD) tools. While functional test and parametric test still dominate current DFT approaches, structurally-based DFT methods provide a long-term solution, especially if the results of these methods are demonstrated to correlate well with standard functional and parametric tests. A fundamental advantage of structurally based DFT methods is the deeper understanding of defects and faults, which leads to quality improvement and cost reduction, two critical figures of merit in any system development effort. One particular DFT solution, which may be seamlessly merged with current design technologies integrating computing and communication, is the use of radio wrappers for wireless control and communication of on-chip DFT subsystems.

The test access problem is reduced for internal subsystems; online monitoring is possible; and communication with external test equipment is standardized, but there are inherent tradeoffs in other design parameters such as chip area, power, noise, and extra loadings on circuit generations. The complex challenges in analog/mixed-signal/RF DFT demand creative solutions and unconventional thinking from both system designers and test architects. These solutions must be developed in a timely manner to avoid the analog/mixed-signal/RF test bottleneck in highly complex system integration efforts.

MPU/PE/DSP system drivers—These system drivers are more mature with respect to DFT, BIST, and alternative test methods such as IDDx testing, even though recent advances in GHz clock speeds have also led to additional test issues not covered by digital DFT and BIST. These additional issues, such as testing high-speed clock for jitter and phase noise, may be considered as part of the overall mixed-signal or RF test requirements discussed above. From the logic design perspectives, the requirements for the MPU/PE/DSP/Memory system drivers are much better understood and many challenges were successfully resolved in the past decade by the DFT and ATE community. Looking ahead, the major requirements include better digital DFT coverage in the sense that more digital blocks must have built-in DFT or be a part of an overall DFT scheme. These built-in DFT methods can vary widely, depending on specific block functions, design styles, and particular goals in fault detection (such as hard fault, soft fault, parametric fault). Delay and power are two examples of digital parameters where coverage needs to be improved significantly, with more DFT and BIST efforts to be devoted to enhancing the test methods for these parameters. Coverage monitoring, given a wide range of faults, defects, and parameters to be tested, must be incorporated into the overall DFT and BIST schemes, especially to target parametric degradation and reliability requirements. Online or offline coverage monitoring in operating environments is essential to anticipate and isolate possible subsystem failures so that repair can be performed in a timely manner. DFT methods with fault-tolerance and repair capabilities are required for these highly complex digital subsystems to enhance yield in the presence of parametric faults and manufacturing defects. Quality must be an integrated goal in the DFT and BIST frameworks for logic systems.

Mature solutions exist for many logic systems in this classification. The remaining challenges demand system-level solutions rather than block-level solutions. As pointed out above, several critical parameters in digital system performance still need effective DFT solutions—delay, power, jitter, local operating temperature, etc. Cost-effective DFT and BIST methods are highly desirable to determine these parameters both in characterization test and high-volume manufacturing test. Solutions that provide integrated calibration and repair capabilities will emerge as crucial to overall electronic system design and test architectures. With both current fault-oriented DFT/BIST methods and these new digital parametric-oriented DFT/BIST methods, system test designers must create cohesive system-wide DFT/BIST architectures to make effective use of the various methods at their disposal. A patchwork test architecture or a haphazard test integration likely create problems in logistics and in test execution planning, which in terms of test time and test data volume, would reduce the advantages of DFT/BIST. Given the diversity of digital DFT/BIST methods, a system test designer should be able to create a unified test architecture for a particular digital system. Unity in diversity is a critical feature in any system-wide digital test architecture solution.

Memory system drivers—The memory system driver is more mature with respect to DFT, BIST, and alternative test methods, even though recent advances in ultra-large memory blocks have also led to additional test issues. These additional issues, e.g. new failure mechanisms, timing uncertainties in read/write access, etc., may be considered as part of the overall mixed-signal test requirements discussed above but given the memory size, they pose separate problems to be solved in the integrated SOC context. From the memory design perspectives, the essential requirement is to improve yield. Redundant design methods have helped immensely in this area but as memory blocks increase in size, yield issues encompass both the requirement for good memory cells but also for memory timing performance within specifications. DFT for yield improvement must provide methods to satisfy both of these goals: physical quality and timing performance of memory blocks.

Mature solutions exist for memory systems in this classification with respect to physical quality enhancement. As mentioned above, redundant design methods and better layout styles, coupled with better understanding of memory fault models, have gone a long way toward improving yield. The remaining challenges in timing performance demand DFT/BIST methodologies that optimize read/write access time and improve reliability, especially for large and/or programmable memory blocks.

Highly integrated general SOC and SIP designs—Integration of pre-existing design blocks into larger integrated devices produces non-linear complexity growth for design tools, DFT, and manufacturing test, even when the blocks are homogeneous (such as all logic). Increasingly, a wider variety of circuits are being integrated. Logic, static RAM (SRAM) and DRAM have been commonly integrated in recent years, but now analog, mixed-signal, RF, and non-volatile flash circuits are being combined with logic and RAM. Silicon complexity and costs are relatively predictable for integrated devices. However, embedded blocks and mixed device types drive highly nonlinear and unpredictable increases in the cost of testability, design verification, and manufacturing test. ASIC or MPU macros wholly embedded within larger logic devices are already seeing this impact; manufacturing test costs are exceeding silicon costs. Even with DFT, these costs may be non-linear. It is emphasized that the DFT/BIST requirements in this section are not just a collection of block-level DFT methods already covered in the three system drivers described above. The system-wide DFT requirements and solutions target the entire system, provide the overall DFT/BIST logistics and scheduling paradigms, and include test capabilities missed or not covered at the lower levels.

Within the system context, DFT must provide methods for on-chip or in-system test generation and test application to reduce the burden in test time and test volume using external ATE. The test generation hardware and embedded algorithms reduce the need for test access while providing an infrastructure for system-wide built-in self-test (BIST), which would be the ultimate system DFT requirements. As DFT technologies for analog, mixed-signal, and RF subsystems improve, their test generation and application resources must continuously be integrated into this system test infrastructure in a seamless manner. Faults (some classes of hard faults and many classes of parametric faults) detected by DFT should be repairable to improve yield and reduce time to market, thus the DFT and electronic system design methodologies must provide this repair capability to work in concert with DFT. A more global perspective and also an appealing aspect of DFT are the DFT capabilities to perform calibration via *in situ* measurement and feedback. Many calibration methods employed by designers to tune the system performance in the presence of process variations and device imperfections are identical to DFT methods. A consistent usage of DFT as a system calibration methodology would promote designers' acceptance and facilitate tighter integration between design and test. In-service or online measurement, field repair, correction of performance degradation over time, etc. are fundamental performance requirements that can be met by a comprehensive DFT methodology.

Testing of embedded blocks may also entail orders of magnitude longer test time than testing of the non-embedded versions. The test methods, ATEs, and manufacturing integrations for SRAM/DRAM, logic, flash, and analog/mixed-signal/RF silicon come from radically different legacies with unique optimizations that are typically broken in testing on an integrated logic device. Extremely long test times for standalone analog or RAM devices are offset by different ATE or test equipment costs and the high degree of parallelism (SRAM/DRAM) that is unavailable to SOC or integrated devices. Again, the embedded nature of the integration may preclude or hamper access to block I/Os that would be available on standalone devices. Not only expanded DFT techniques and protocols, but significant advances and use of BIST and/or embedded software-based self-testing for larger portions of test, are required. DFT and BIST for embedded mixed-signal/analog/RF blocks—long a research area—will become an industrial reality, driven by the dominant use of these types of circuits in integrated devices and SOCs.

An impediment to DFT incorporation has been the perception that DFT is costly in terms of system impact: chip area, test-specific I/O allocation, power, bandwidth, signal sensitivity, etc. The integration of analog/mixed-signal/RF subsystems and very high speed digital subsystems into SOC and SIP has led to a stricter examination of additional leakage currents, noise, and loads on sensitive nodes being monitored by DFT circuits. The impact of DFT on system performance, such as bandwidth loss or additional noise, must be quantified and estimated well enough to permit an early benefit versus cost study of possible DFT methods to be chosen for incorporation. DFT methods should also focus on monitoring less sensitive nodes and indirectly estimate test results (such as fault detection, parametric measurement).

The greatest advantage of DFT is test volume and test cost reduction. This advantage must be well demonstrated at the SOC/SIP level, where test time and test cost are most apparent and more easily calculated. The test volume reduction target in the table covers the entire system test volume and test time, not the test volume and test time of a specific block within the system. For example, a 2× reduction requirement for 2006 seems to be quite conservative for the digital subsystems but considering the entire system (where analog/RF tests consume most of the test time), the 2× reduction is a reasonable requirement. Test time and cost are also strongly correlated to ATE since, even with DFT and BIST, ATE still plays an important role in test. The interface between various DFT methods and any ATE must be well defined to permit flexibility in system implementation of different DFT paradigms and in ATE selection to meet specific product test needs. While test access port and other test standards, such as IEEE 1149.1 (*IEEE Standard Test Access Port and Boundary-Scan Architecture*), 1149.4, 1450 (*IEEE P1450.6, Draft Standard for Standard Test Interface Language for Digital Test Vector Data – Core Test Language*), and 1500 (*IEEE 1500-2005, IEEE Standard Testability Method for Embedded Core-Based Integrated Circuits*) are available, comprehensive DFT/ATE interface protocols spanning various system layers (behavioral layer, physical layer, data communication layer, etc.) must be established to fully exploit the advantages of the DFT methods and the ATE capabilities.

Solutions to the overall SOC/SIP DFT/BIST problems have been developed in the past five years and continue to be created at a very fast pace. The diversity of technologies and subsystem design methods within a SOC or SIP demand solutions for test generation tools and DFT/BIST tools that integrate digital test methods (fault-based, defect-based, and some parametric-based) with analog/mixed-signal/RF test methods (functional-based, parametric-based, and future fault-based and defect-based).

Solutions must be accompanied by design and test planning tools to reduce system complexity, design effort, time-to-market, and overall test cost. The various levels of integration within a SOC or SIP (from highly integrated digital blocks to macro-based RF or analog blocks) lead to solutions that can span the entire system hierarchy and provide appropriate capabilities for the specific subsystem under test: fault tolerance, calibration, tuning, repair, with the overall goal of yield and quality improvement. At any hierarchy level, an effective system solution must provide a collection of compatible DFT/BIST methods available to the designers and must integrate the various choices made by the designers at different hierarchy levels as the design proceeds. The total solution at the end must be a cohesive integrated solution reflecting both design and test optimization.

Efficient simulation models and tools are inherent in all design and test planning efforts, and must be a part of the test solution, especially in estimating the impact of DFT/BIST on system performance. Simulation tools must provide this impact estimate as the design proceeds across various levels—behavioral, circuit, and physical—to permit the selection of the most suitable DFT/BIST methods. An overall figure of merit is system test coverage (fault or parametric or functional or a combination thereof) that must be computed as part of the DFT/BIST incorporation and monitored both during test and during operations. Capabilities for online coverage monitoring are highly desirable and algorithms must be developed to relate coverage statistics to fault and defect statistics to enable manufacturing quality improvements.

Certainly the most controversial aspect of system-wide DFT/BIST solutions is the interface between on-chip DFT/BIST capabilities and ATE. Many test protocols exist, at least for digital systems and in recent years for mixed-signal systems, but an overall interface standard for overall SOC/SIP is still lacking. Since test resource partitioning practices vary widely in industry, the interface between DFT/BIST and ATE has not been well-defined but future solutions must contain this definition with at least two critical characteristics: rigorous clear interface definitions to permit the interchangeable and effective use of various ATEs in system testing, and flexible interface definitions to avoid constraining the electronic system design and test architectures.

DESIGN FOR MANUFACTURABILITY (DFM)

Increasing variability, mask cost and data explosion, and lithography hardware limitations are posing significant design challenges for the manufacturability of integrated circuits:

Architecture challenges—architectural redundancy will be required due to the difficulty in making circuits yield. It will be hard to do much more at this level of abstraction.

Logic and circuit challenges—digital and mixed-signal adaptive circuits will be increasingly necessary. Statistical design, including power and timing convergence will be fundamental. But it will be hard to make designs work without addressing two primary challenges: 1) characterization and modeling inputs to statistical design tools, and 2) the evolution from statistical analysis to optimization with the subsequent increase in computation complexity. Finally, statistical analysis tools and statistical optimization methods must model actual manufacturing and design-induced variations rather than crude and possibly inaccurate abstractions. The composition of variations during statistical analyses and optimizations should match the methods used for initial decomposition via statistical metrology techniques during process characterization. A mismatch between composition and decomposition will introduce unnecessary error and impart dubious value to the computationally costly results.

Layout and physical design challenges—First, the complexity of design rules checking is increasing (include graph). Rules have evolved to a two-tier system (required versus recommended rules) and may need to evolve either to a three-tier system or to a no-tier system (where rules are not pass-fail but provide a Pareto curve of yield benefit versus area cost that the designer can apply as criteria for tapeout). This checking will need to be done without increasing designer-perceived complexity. Second, the limitation in lithography hardware resolution will require design flows to more explicitly account for the impact of reticle enhancement techniques (RET). RET tools, such as OPC and chemical-mechanical planarization (CMP) fill, must become explicitly aware of circuit metrics such as timing and power. Such awareness aligns the tools with overall product goals and enables yield enhancements, manufacturing cost reductions, and improvement in mask data preparation time. As an example, OPC would be applied only to features in critical timing paths. This approach entails a tighter flow integration to communicate circuit intent downstream, and to avoid independent modifications by several tools leading to incorrect results. As a consequence, the tools in the register transfer level to general data stream (RTL to GDS) flow will have to properly plan for RET and OPC modifications downstream. For example, global and local layer densities should be taken into account at the global placement level to determine possible locations for dummy fill insertion, thus allowing pre-placement of CMP fills and accounting for their capacitance while at the same time propagating information about critical nets down to layout finishing, mask data preparation (MDP), and OPC steps.

Yield prediction and optimization as design challenge—Layout ground rules as specified are no longer “hard numbers” since such an interpretation tends to drives design *compliance* rather than design *improvement*. To achieve reasonable mature yields and a steep yield ramp capability a strategy for meaningful design rule relaxation has to be used. These “recommended” rules have been derived from the interaction of the design layout and wafer process requirements, such as alignment tolerances, optical proximity corrections, RET enhancements and many other constraints. DFM measures act differently for each design because they impact power, area, and speed as the design is tuned for yield. During the design process the mutual interaction of yield, area, power and speed must be analyzed and a commercially useful tradeoff must be found. DFM measures that effect the functional and parametric yield must be integrated as a new optimization feature into the design flow and tools rather than as post processing that allows only limited results and is typically time consuming. That means that yield prediction has to be integrated into tools for design planning synthesis and place and route to consider all design targets to optimize simultaneously for yield, performance, power, signal integrity, and area (with yield as the explicit new design target). Finally yield is a function of both product-specific design attributes and process-specific failure probabilities. Therefore, a design that has been optimized to tolerate specific fail/marginality patterns of a particular process may actually yield poorly in different process conditions. To enable the correct evaluation of the yield cost of different design implementations in the place and route (P&R) tools it is therefore necessary to pre-characterize accurate yield models of the logic library that are based on the actual target process data. Library yield models need to be frequently updated to track process evolution from pre-ramp to a mature stage.

Table DESN9 quantifies the key challenges above as a set of DFM technology requirements. Figure DESN8 then defines a set of DFM potential solutions that will be needed to address the technology requirements over time.

Table DESN9a Design for Manufacturability Technology Requirements—Near-term Years

Year of Production	*2007*	*2008*	*2009*	*2010*	*2011*	*2012*	*2013*	*2014*	*2015*
Normalized mask cost from public and IDM data	1.0	1.3	1.7	2.3	3.0	3.9	5.1	6.6	8.7
% V_{dd} variability: % variability seen in on-chip circuits	10%	10%	10%	10%	10%	10%	10%	10%	10%
% V_{th} variability: doping variability impact on V_{th}, (minimum size devices, memory)	31%	35%	40%	40%	40%	58%	58%	81%	81%
% V_{th} variability: includes all sources	33%	37%	42%	42%	42%	58%	58%	81%	81%
% V_{th} variability: typical size logic devices, all sources	16%	18%	20%	20%	20%	26%	26%	36%	36%
% CD variability	12%	12%	12%	12%	12%	12%	12%	12%	12%
% circuit performance variability circuit comprising gates and wires	46%	48%	49%	51%	60%	63%	63%	63%	63%
% circuit total power variability circuit comprising gates and wires	56%	57%	63%	68%	72%	76%	80%	84%	88%
% circuit leakage power variability circuit comprising gates and wires	124%	143%	186%	229%	255%	281%	287%	294%	331%

Table DESN9b Design for Manufacturability Technology Requirements—Long-term Years

Year of Production	*2016*	*2017*	*2018*	*2019*	*2020*	*2021*	*2022*
Normalized mask cost from public and IDM data	11.4	14.9	19.6	25.6	33.6	44.2	57.7
% V_{dd} variability: % variability seen in on-chip circuits	10%	10%	10%	10%	10%	10%	10%
% V_{th} variability: doping variability impact on V_{th}, (minimum size devices, memory)	81%	81%	112%	112%	112%	112%	112%
% V_{th} variability: includes all sources	81%	81%	112%	112%	112%	112%	112%
% V_{th} variability: typical size logic devices, all sources	36%	36%	50%	50%	50%	50%	50%
% CD variability	12%	12%	12%	12%	12%	12%	12%
% circuit performance variability circuit comprising gates and wires	63%	65%	66%	69%	69%	71%	73%
% circuit total power variability circuit comprising gates and wires	92%	96%	102%	110%	121%	130%	140%
% circuit leakage power variability circuit comprising gates and wires	368%	381%	395%	360%	325%	477%	628%

Manufacturable solutions exist, and are being optimized
Manufacturable solutions are known
Interim solutions are known
Manufacturable solutions are NOT known

DFM requirements fall into the following categories:

- *Requirements due to fundamental economic limitations*—This category includes mask cost, which is reaching multi-million dollar levels, thereby jeopardizing System-On-Chip innovation coming from small companies and emerging-market institutions.
- *Requirements due to variability and lithography limitations*—This category includes limitations at low abstractions levels; those that are directly associated with devices and wires; and at higher levels of abstraction, those associated with the overall circuits being designed. At the lower level, quantified requirements include voltage supply variability (as delivered to on-chip circuits), voltage threshold variability for both minimum size memory devices and typically sized logic devices (in terms of both the impact of doping concentration variability and the overall trends), and percentage CD (critical dimension, specifically channel length) variability. At the higher level, requirements include percentage of circuit performance variability (the percentage uncertainty in the speed of the circuit that determines overall chip performance, such as its critical path), and percentage of circuit power variability (the percentage uncertainty for power consumption, including both active and standby power).

Economic parameters such as mask cost will be difficult to control directly and thus increases are expected. Instead, workarounds will be found, such as multi-project wafers, configurable logic, and structured ASICs. With respect to variability, some parameters, or at least their targets, will be controlled by design, including "process-level" parameters such as CD control, and "circuit-level" parameters, such as voltage supply. Other parameters, such as threshold voltage variability (including the impact of channel doping, which is known to scale with the inverse of device area) will unavoidably grow, leading to potentially critical red bricks next decade or earlier. Due to upward parameter propagation from process level to device level to circuit level, very large overall circuit performance and power consumption variability will need to be addressed, unless radical new solutions are found.

To address DFM technology requirements, the DFM solutions described in Figure DESN8 will be needed:

- *Solutions that address fundamental economic limitations*—Future DFM tools will be required that directly account for economic factors, including but not limited to mask cost, in their main algorithms and interfaces. Teams and their managers will be able to more directly assess the economic value of difficult DFM tradeoffs.
- *Solutions that address the impact of variability*—Future DFM will have to directly address and tolerate various dimension of variability. First, they will need to address variability in both performance and power consumption—thereby leading to a surge in the importance of statistical performance analysis, active power, and leakage power analysis tools. Second, they will need to address and distinguish between two distinctive types of statistical yield-loss behavior: systematic and random. Third, they will need to optimize across the various sources of environmental and process-induced variability, including voltage supply, temperature, and threshold voltage variability. Finally, tool-based solutions will not be sufficient—design techniques that help compensate for variability will be needed, including 1) sophisticated adaptive circuits that can sense and differentiate between sources of variability and adjust circuit activity, supply, clocking and/or inputs, and 2) architectures that are fundamentally resistant to variability, including locally asynchronous designs, redundancy, and error correcting signal encodings (ECC).
- *Solutions that address the impact of lithography limitations*—due to the prominent role played by lithography, the ITRS addresses lithography-related DFM problems and areas requiring solutions. Future design flows will need to include a radically increased effort specifically addressing lithography limitations. These techniques will likely include both rule-based (which does not change tools and/or flows) and model-based (directly changing tools and/or flows) layout correction. First, RET, which are applied post-layout today, will need to increasingly interact with traditional design steps, such as synthesis, timing, placement and routing, and they will need to more explicitly include performance and power consumption metrics in their functionality. This interaction may be direct or indirect, in the form of "model-based" design steps, such as physical verification and synthesis. Second, conventional rules and designs will increasingly become "manufacturing friendly," such as fundamentally manufacturable by design. Rules will be a key piece of these solutions, including manufacturing-friendly rules (strict "hard" rules that follow basic yet effective manufacturability principles) radically restricted rules (RDRs, simple rules that ensure manufacturability at little area or performance cost, such as grid-like layouts with no diagonals), and litho-router-friendly standard cells and cores.

Early solutions that directly handle variability (such as timing analysis) are emerging. In less than five years (between 2009 and 2011), statistical methods will be completely embedded in the design flow. In the meantime, they will be selectively applied as they mature, or will be part of premium design technologies used in high value or high volume designs.

DFM techniques that directly account for lithography have become very popular but will take even longer to become qualified in production flows, primarily due to their even tighter link to manufacturing models and data. The next decade will feature chips that will be swarmed with manufacturability issues—fortunately, however, the design flow will have been completely overhauled by then with production-level DFM techniques that will be transparent to the designer except for the information that can effectively be used to address manufacturability directly on the design.

Table DESN10 establishes correspondences between DFM requirements and DFM potential solutions.

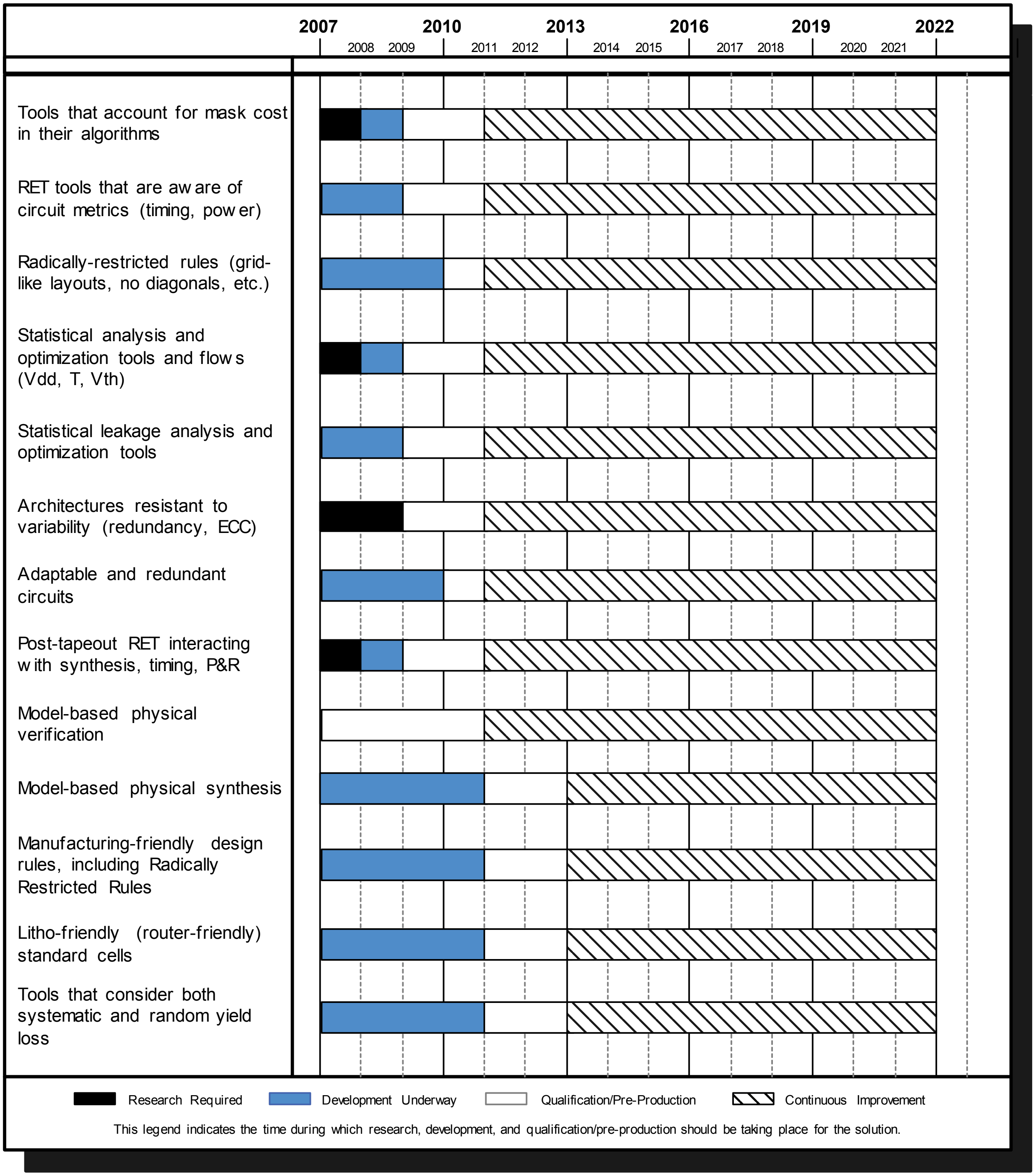

Figure DESN8 Design for Manufacturability Potential Solutions

Table DESN10 Correspondence Between Design for Manufacturability Requirements and Solutions

Requirement	*Solution*	*Explanation of the Correspondence*
Mask cost	Tools that account for mask cost in their algorithms	Obvious
	RDRs (grid-like layouts, no diagonals, etc.)	Better manufacturability and yield, less mask complexity
	RET tools aware of circuit metrics (timing, power)	More effective optimization, fewer design iterations
	Statistical leakage analysis and optimization tools	Estimation and control of soaring leakage variability
	Post-tapeout RET interacting with synthesis, timing, P&R	By interacting with earlier-in-the-flow EDA tools, can more effectively address litho issues
	Model-based physical verification	Can address litho issues with precision
	Model-based physical synthesis	Explicit litho model-based approach moves into the physical synthesis toolset
	Manufacturing-friendly design rules (hard rules)	Reduces mask, manufacturing cost; addresses printability
% V_{dd} variability seen at on-chip circuits	Tools that account for mask cost in their algorithms	Obvious
	RDRs (grid-like layouts, no diagonals, etc.)	Better manufacturability and yield, less mask complexity
	RET tools aware of circuit metrics (timing, power)	More effective optimization, fewer design iterations
	Statistical leakage analysis and optimization tools	Estimation and control of soaring leakage variability
	Post-tapeout RET interacting with synthesis, timing, P&R	By interacting with earlier-in-the-flow EDA tools, can more effectively address litho issues
	Model-based physical verification	Can address litho issues with precision
	Model-based physical synthesis	Explicit litho model-based approach moves into the physical synthesis toolset
	Manufacturing-friendly design rules (hard rules)	Reduces mask, manufacturing cost; addresses printability
% V_{th} variability (doping variability impact)	Statistical analysis, opt tools and flows (Vdd, T, Vth)	Better estimate of variability impact reduces overdesign
% V_{th} variability Includes all sources	Statistical analysis, opt tools and flows (Vdd, T, Vth)	Better estimate of variability impact reduces overdesign
	Adaptable and redundant circuits	Inherent circuit robustness to variability
	Statistical leakage analysis and optimization tools	Estimation and control of soaring leakage variability.
% CD variability	RET tools aware of circuit metrics (timing, power)	More effective optimization, fewer design iterations
	RDRs (grid-like layouts, no diagonals, etc.)	Better manufacturability and yield, less mask complexity
	Adaptable and redundant circuits	Inherent circuit robustness to variability
	Statistical leakage analysis and optimization tools	Leakage power variability will soar. Statistical leakage tools are critical to estimate and control it.
	Post-tapeout RET interacting with synthesis, timing, P&R	By interacting with earlier-in-the-flow EDA tools, can more effectively address litho issues
	Model-based physical verification	Can address litho issues with precision
	Model-based physical synthesis	Explicit litho model-based approach moves into the physical synthesis toolset
	Manufacturing-friendly design rules (hard rules)	Reduces mask, manufacturing cost; addresses printability
Circuit performance variability (gates and wires)	Router-friendly standard cells	Routing-friendly rules reduce design, mask, and manufacturing complexity
	Adaptable and redundant circuits	Inherent circuit robustness to variability
Circuit power variability (gates and wires)	Adaptable and redundant circuits	Inherent circuit robustness to variability
	Statistical leakage analysis and optimization tools	Estimation and control of soaring leakage variability.
	Post-tapeout RET interacting with synthesis, timing, P&R	By interacting with earlier-in-the-flow EDA tools, can more effectively address litho issues
	Model-based physical verification	Can address litho issues with precision
	Model-based physical synthesis	Explicit litho model-based approach moves into the physical synthesis toolset
	Manufacturing-friendly design rules (hard rules)	Reduces mask, manufacturing cost; addresses printability
	Router-friendly standard cells	Routing-friendly rules reduce design, mask, and manufacturing complexity

MORE THAN MOORE ANALYSIS

To understand the relative impact of Moore-based and non-Moore sources of design technology improvement, we have classified all solutions in the five main sections in this chapter into the categories of 'geometric scaling', 'equivalent scaling', and 'functional diversification'. The results are shown in Figure DESN9.

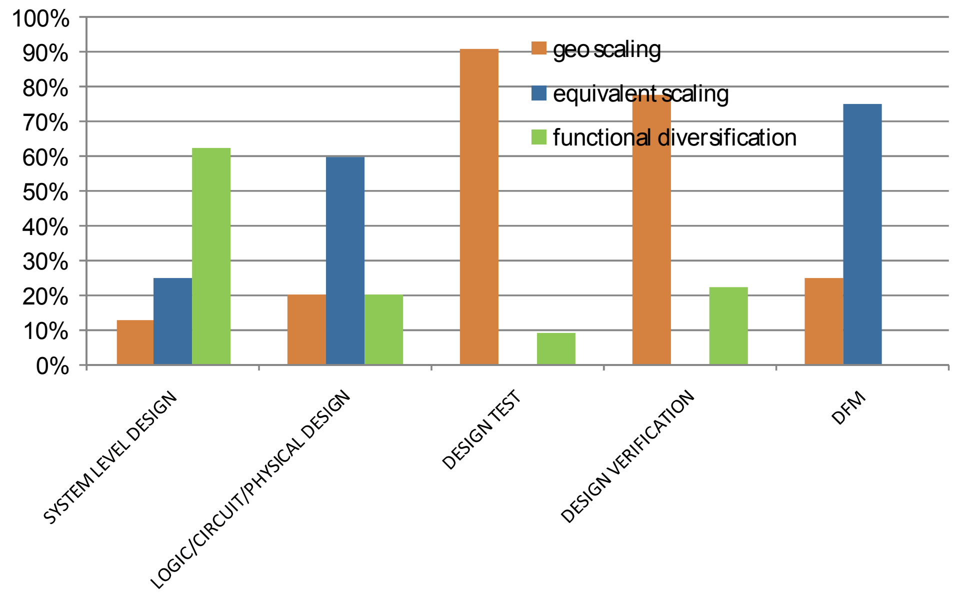

Figure DESN9 Moore and Non-Moore Design Technology Improvements

As the figure shows, it is early in the design process, during system-level design tasks, that functional diversification has greatest impact in terms of solutions to the industry's product requirements. As the design process progresses to lower levels of abstraction, equivalent scaling and especially geometrical scaling techniques dominate. This is consistent with the way modern design works: 1) initially, the system level of abstraction mixes multiple cores, implementation fabrics, and circuit types, while 2) at physical realization levels of abstraction every block is synthesized into a type of layout and circuit for which well-defined design tools and methodology can be applied.

ANALOG, MIXED-SIGNAL AND RF SPECIFIC DT TRENDS AND CHALLENGES

Analog/mixed-signal and RF circuits are different compared to digital circuits. They do not contain quantized information that is represented in a defined range of voltage levels for the two states of a bit ("high" (V_{dd} – noise tolerance) and "low" (V_{ss} + noise tolerance)) and during a defined discrete time interval (such as clock signal high). Rather the signals processed in analog and RF circuits are continuous in time and amplitude up to a much higher degree of precision (or to a smaller tolerance both in time and amplitude). Therefore non-idealities like linearity, noise, parasitic elements, and electrical non-uniformity of the devices used in a circuit directly cause distortion and noise in the analog or RF signals processed in the circuit. Digital circuits have the built-in ability to suppress a high level of these noise sources due to a significant gain in the transition point of each logic gate. This simple signal-recovery principle cannot be used in analog and RF signal processing due to the much higher dynamic range of the involved signals. Speed issues, or simply the fact that a signal-recovery circuit produces more noise and distortion than it prevents the signal from being susceptible to, make these issues much more challenging and less straightforward in the analog domain.

Analog and RF designs have driven the needs for high precision compact modeling as well as for characterization and extraction of all kinds of device non-idealities. However, the above-mentioned problems make extraction of simple rules for higher levels of abstraction in analog and RF circuit design (or even layout) very difficult. Tools used for digital circuits have been in a wrong style but also too inaccurate to be used in analog design. Since the number of analog or even RF transistors per chip increases at a much slower rate compared to digital transistors per chip, historically all these boundary conditions have kept analog and RF designers away from system-level design tools. Circuit and electronic system design has remained mainly at the level covered by the section on Logical, Circuit, and Physical Design in this chapter. Today similar problems to those commonly known for analog and RF designs start to arise in digital designs as well (IR-drop, cross-talk, etc.). SOC digital designs share chip area with analog and RF circuits. The ever-shortened time to market and the need for higher productivity have changed historic paradigms in recent years, and new analog and RF specific challenges for the EDA industry are arising today.

As noted in the *System Drivers chapter*, there are many challenges to scaling and migration of AMS designs. These challenges include decreasing supply voltages, increasing relative parametric variations, increasing numbers of analog transistors per chip, increasing signal, clock and intrinsic device speeds, increasing leakage and crosstalk in SOC integration, and a shortage of design skills and automation. Particular challenges include 1) deep submicron effects and higher signal and clock frequencies: even "digital goes to analog," increasing role of parasitics, transmission line effects, and signal integrity (SI); 2) analog synthesis tools to reduce effort spent on analog circuit design by a limited number of analog designers; 3) yield enhancement that requires "design for manufacturing," 4) close integration of signal processing systems (mostly digital) and analog RF front ends in almost all mobile communication devices; 5) common consideration of chip, bond wires and package with respect to electrical parasitics modeling, EMI/SI as well as thermal effects and 6) tight integration of electrical and non-electrical components (such as MEMS). These challenges may be elaborated in the context of traditional DT areas, as follows.

Accurate calculation of distributed analog components (inductors and coupled inductors, interconnects and transmission lines, electronically tunable capacitors) is necessary for higher frequencies (> 1 GHz). These calculations have to take into account skin effect and dispersion (that is, the frequency-dependent phase velocity that leads to waveform distortion) and to introduce these effects into macromodels. The impact of such calculations and some interesting results are reported in the literature (2005) in the context of the design of a 43 GHz voltage-controlled oscillator with today's technology (130 nm CMOS standard process). An inductor has been realized as a single turn of the top copper metal layer of length 2×100 µm, width of the wire 5 µm, and distance between the centers of the two wires of 15 µm. This very simple layout resulted in an effective inductance of 83 pH and a Q of 35. 2.5D electromagnetic simulation has been used for accurate modeling of magnetic fields and the skin effect. Capacitors have been realized with an area of about 100×100 µm each.

Antenna effects are important for frequencies > 10 GHz (the wavelength λ in vacuum is 3 mm at f = 100 GHz!). It is clear that antennas-on-chip have to be designed with sophisticated numerical methods, for example, finite difference time domain (FDTD), but the same effects have to be captured by simpler methods (perhaps with reduces accuracy) if they should be considered as parasitics in the context of interconnection networks. Similar problems exist and were solved if advanced analog circuits are used for digital clock generation (>1GHz) with standing-wave and traveling-wave oscillators.

Almost all of the above mentioned effects have influence on the construction of macromodels. Electrical capacitive and magnetic coupling in interconnection networks, thermal properties of the chip embedded in the package, far-field and near-field effects of antennas as well as the behavior of MEMS can be analyzed with 2D/3D field calculation methods and lead to systems of between 10^4 and 10^8 equations. Without serious loss of accuracy, the number of equations can be reduced to between 10^2 and 10^4 by sophisticated model order reduction methods. Rapid progress in this area should enhance the speed and reach of CAD tools over the next few years. This new kind of automatically generated macromodel may be used in analog circuit design as well as in the investigation of digital timing behavior and signal integrity.

SYSTEM-LEVEL DESIGN FOR ANALOG, MIXED-SIGNAL AND RF

In System-Level Design, the critical AMS challenges are non-scalability of analog circuits and analog behavioral modeling and synthesis. Automated analog circuit synthesis and optimization is needed, along with language-level modeling methodologies that permit analysis of overall system function and interfaces with other integrated technologies (digital, software). Issues include coping with vastly varying time scales (analog frequencies up to 100s of GHz versus digital frequencies up to 1s of GHz) in simulation; creating heterogeneous test benches and ensuring coverage; achieving systematic top-down constraint propagation; and mixing of functional and structural representations.

LOGICAL, PHYSICAL, AND CIRCUIT DESIGN FOR ANALOG, MIXED-SIGNAL AND RF

In Logical, Physical, and Circuit Design, the key challenge is analog synthesis. Scalable SOC design requires elimination of the analog design bottleneck. A technology need shared with system-level design is for reusable, retargetable analog IP generators. Today's specialized automatic circuit syntheses for particular classes of circuits (PLL, op-amp, power amplifier, etc.) must be augmented by more general techniques. Automatic layout syntheses must be able to handle the needs for high-performance analog designs (for example, cross-coupled layout for mismatch sensitive transistors). Analog post-layout simulation must handle increased opportunities for distortion and nonlinearity due to impact ionization, thermal nonlinearity, body contacts acting as low-pass filters. Syntheses must also handle future regimes of increased manufacturing variability, such as by hybrid analog-digital compensation for device mismatch. In the near term, new synthesis tools for optical interface circuits and high-Q CMOS on-chip inductors and tunable resonators are needed. Long-term circuit types of interest include low-power sensing and sensor interface circuits, and micro-optical devices.

DESIGN VERIFICATION FOR ANALOG, MIXED-SIGNAL AND RF

In Design Verification, AMS circuits require checking "to specification" rather than with respect to structure. While ever-faster simulation has been the historical solution, new verification solutions must include statistical techniques, better compact models that speed up simulation even increasing accuracy, and new acceptance criteria. AMS designs also force the long-term issue of hybrid-systems verification—a field still in its infancy—into the near term. Thus, an immediate challenge is to provide any support possible to improve the current ad hoc approaches and find a path toward more powerful techniques. As MEMS, electro-optic, and electro-biological devices become more than simple transducers, a further challenge is to model, analyze and verify integrated systems having such heterogeneous parts. The same is true for pure electronic circuits with new device types (such as carbon nanotube transistors, single electron transistors, and resonant tunneling diodes) that use sophisticated physical effects and lead to non-classical device models. Table DESN11 summarizes near-term AMS design technology breakthroughs expected through 2011. The reader is also referred to the excellent discussion of AMS DT requirements in the *MEDEA+ Design Automation Roadmap*.

Table DESN11 Near-term Breakthroughs in Design Technology for AMS

Field of Breakthrough	*2007 State-of-the-Art*	*2008/09*	*2010/11*
Specification, validation, verification	Established AMS Hardware Description Languages	Multi-language support, AMS extension of HW/SW description languages for full system simulation	Complete specification-driven design flow; some specialized formal verification methods
Architectural design	Algorithm-oriented design (e.g., with Matlab/Simulink)	Language-based performance evaluation; closer coupling of architectural, block, and circuit level	Synthesizeable AMS description; power-aware HW/SW partitioning extended to AMS systems
Physical mixed A/D and RF design	Procedural layout generation, module generators for a few block types	Module generators for often re-used blocks, design centering, performance estimation	Synthesis: behavior to layout (at least for the most important building blocks)
Parasitics extraction, automated modeling, accelerated simulation	Electromagnetic immunity simulation works but is too complicated for broad usage	2D/3D model-based order reduction for interconnect systems and substrate effects on chip, thermal package modeling	New fault-tolerant circuit architectures, robustness against technology parameter variations; order reduction for all kinds of parasitics and antennas

With respect to Design Test, analog circuitry dominates production test cost despite being a small fraction of the total area of mixed-signal SOCs. The ratio of analog testing cost to total mixed-signal product cost will continue to increase unless changes to analog testing are made. The near-term requirement is for analog/mixed-signal DFT/BIST, especially at higher resolution and/or higher frequencies beyond baseband. Test techniques for on-chip, high-resolution (>14–16 bits) ADC and high-speed (>1–5 GHz) RF components must be not only cost-effective, but also nonintrusive—they must not degrade performance of embedded analog blocks. Since high-resolution ADCs are usually constructed with multiple stages connected in serial or mash configuration, one possible direction is to utilize this structural knowledge in developing a DFT or self-test strategy.

Although the *Process Integration, Devices, and Structures chapter* specifies an analog CMOS transistor that has a higher analog supply voltage and is not scaled across 2–3 technology generations, this does not solve critical cost issues of power, process compatibility, area efficiency, design complexity, or verification and test. Furthermore, AMS design productivity remains a key challenge to development of new mixed-signal parts. A near-term roadmap for AMS DT includes new description languages as well as tools for:

- System exploration, considering the chip and the package
- Circuit synthesis and sizing
- Schematic validation
- Design for manufacturing
- Analog/RF layout synthesis
- Parasitics extraction (capacitive, inductive, thermal), automated modeling and fast simulation
- Analog IP encapsulation and reuse

CROSS-CUT TWG ISSUES

MODELING AND SIMULATION

One of the key problems that challenges design in connection with further shrinking feature sizes is the increasing variability of design-related parameters, resulting either from fluctuations of fabrication parameters or from the intrinsic atomistic nature affecting, for example, channel doping. Modeling and simulation can and must help to ease this problem by assessing the quantitative impact of such variabilities on the relevant design parameters. Statistical variations as well as drifts of fabrication parameters must be translated via appropriate equipment, process and device simulation as well as parameter extraction into the resulting distribution of design parameters, such as sizes and spacings of active and passive devices, transistor characteristics, and coupling of interconnects leading to signal delay and distortion. Increasingly important is the atomistic nature of dopants which in some cases results in just one or a few dopant atoms being present, on average, in the channel region, giving rise to enormous relative fluctuations of doping and, in turn, electrical device parameters. Also important are the interactions between different subsequent process steps, such as lithography and etching, which may either amplify or smooth such process fluctuations. Simulation should further contribute to the assessment of the impact of parasitics, delay variations, noise and reliability issues, including thermal effects during operation. The overall target is to link design parameters more closely to the technology and device architectures used, especially including their process-induced variations, in order to help designers to select appropriate safety margins, which may vary within the layout. The added value that only simulation can provide is that a wide set of variations may be investigated largely automatically, within relatively small time and at relatively small costs.

APPENDIX I: VARIABILITY MODELING AND ROADMAP

Since variability is expected to be the source of multiple critical DFM challenges, a systematic method to roadmap required and/or desired variability trends will become a crucial component of the overall Design roadmap. It may also allow for design-manufacturing "co-roadmapping," which may help gather indications of whether our industry should invest in variability reduction or in design productivity improvements. The requirement for such a framework is underlined by the sensitivity of variability-related information for industry participants. Since the design community needs to view variability from their parameter-based perspective, such a variability framework needs to be multi-level, i.e., it needs to cover multiple levels of design abstraction.

A variability roadmap framework (VRF) is being developed by the Design TWG as illustrated in Figure DESN10. In this Framework the following three levels of abstraction have initially been considered:

- *Circuit/chip level*—this is the abstraction level most relevant to designers. Ideally, timing and power consumption variability for a given circuit would be roadmapped at this level, based on lower-level parametric variations.
- *Device level*—since circuits are composed of devices, at this level device-related parameters are roadmapped, such as the threshold voltage V_t, or the off-current I_{off}.

- *Physical level*—this is the level closest to the design-manufacturing interface, where parameters such as CD or effective device length (L_e), and actual doping level (N_A) are roadmapped. These parameters' variability stems from some of challenging issues enumerated above, including lithography hardware resolution limitations, and the inability to control the exact number of dopants in a channel.

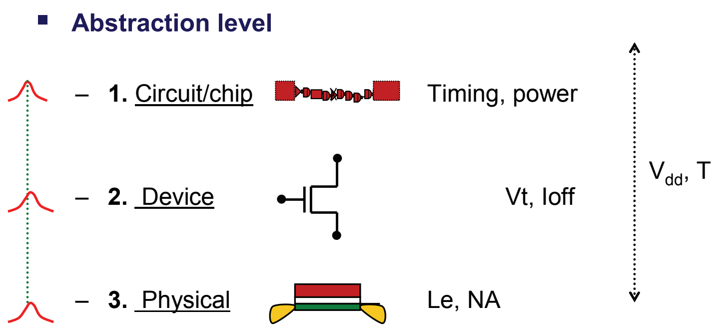

Figure DESN10 Possible Variability Abstraction Levels

In this framework, the modeling process can be symbolically described as

$$\Delta \text{ outputs} = \text{model } (\Delta \text{ inputs}) \qquad [1]$$

Based on published approximated models, this model is based on a simplified gate + wire circuit slice where parameters are decomposed into gate-related and wire-related parameters. Performance is analytically modeled in the current model as total delay, which in turn is decomposed into gate delay and wire delay. Delay variability is modeled as a delta that is statistically computed from a statistical distribution of individual delay values that stems from "simulating" distributions of the input parameters in a Monte Carlo style. Model inputs include gate channel length and width, and oxide thickness; wire width, length, and thickness; ILD thickness, and wire sheet resistance.

Among other applications, this framework will be used for circuit performance variability trending. For example, estimated delay variability will be estimated over time for two possible scenarios: 10% required CD (actual channel feature length) variation, and 20% required CD variation. In this particular illustration, the model could indicate that circuit performance variability does not seem to vary very significantly with the CD tolerance requirement, which might suggest the possibility of a relaxation of the requirement.

APPENDIX II: DT COST AND VALUE

As Figure DESN11 shows, the cost of developing and marketing a complex integrated circuit is affected by many different factors. Each factor represents either a fixed or a variable cost component. While fixed costs do not depend on the number of units being sold, variable costs grow with the number of units sold. Product development is a fundamental part of the electronic product value chain, and can generally be seen as a fixed cost factor that is spread across the number of units sold. For the purpose of this discussion, design cost is defined as the direct product development R&D cost plus its associated overhead. Unfortunately, the ever-increasing complexity of systems-on-a-chip makes design costs, and thus unit costs, difficult to control. Rising costs, combined with an increasingly competitive environment, can make profitability ever more difficult to achieve. This is exacerbated by the fact that product development costs come upfront in the life cycle, whereas substantial revenue often comes years later (discounted-cash-flow effect). The following analysis suggests that without a continued design technology innovation pipeline, design cost (and thus product development cost) would quickly become prohibitive, or else designs will be forced to have less valuable content.

In Figure DESN11, items in bold can be seen as design costs (opportunity and lost-revenue costs are not included). The figure shows that product development cost can be roughly decomposed into direct labor costs and infrastructure costs. Labor costs include chip, circuit, and layout/physical design; chip integration; verification and test; software development; EDA integration; and software and technology support. Infrastructure costs include design software licenses (including software development environments), test chip infrastructure, and depreciation. These costs are often expressed as direct costs plus an allocated "overhead" component, including general and administrative expenses. The vital contribution of DT to semiconductor product profitability can be understood by enumerating and analyzing the impact of DT innovations on each of these cost components.

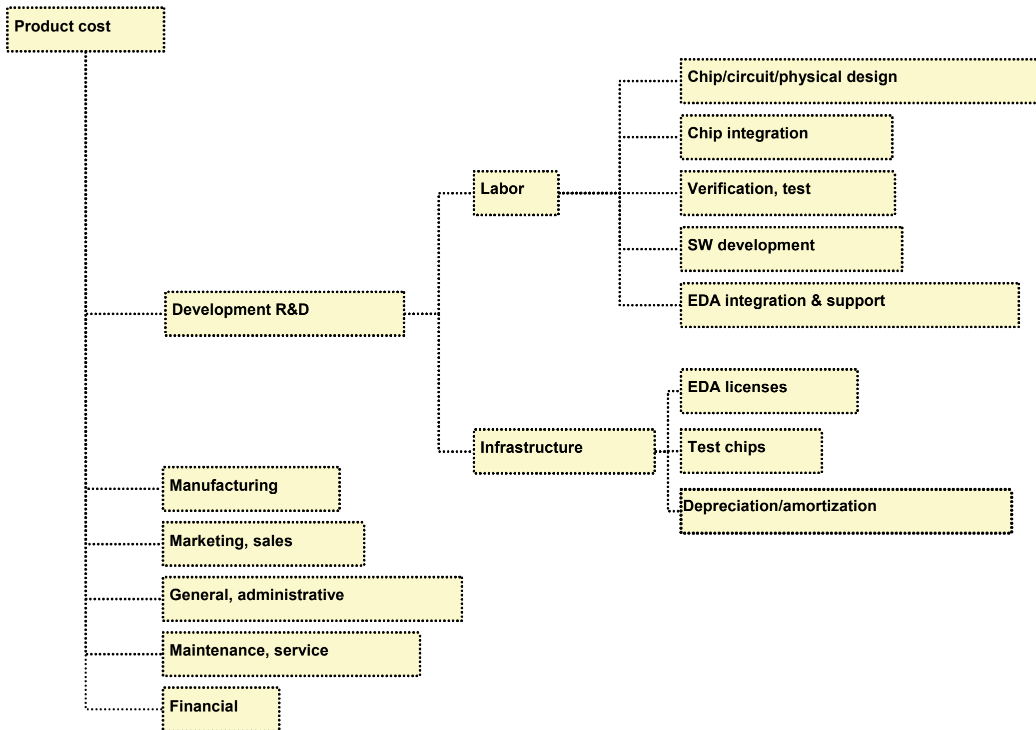

Figure DESN11 Simplified Electronic Product Development Cost Model

LABOR COSTS

The labor cost component is approximately proportional to *labor unit costs* (in terms of compensation dollars per engineer-year), *design complexity* (in terms of the amount of behavior or devices in a chip) and *designer productivity* (in terms of the design complexity that an average engineer can fully design in a year):

$$DesignLaborCost = \frac{LaborUnitCost \times DesignComplexity}{Designer\,Pr\,oductivity}$$

Since DT innovations have increased designer productivity, their strongest effect is on the labor cost component. To measure the labor cost impact of DT innovation, Gartner/Dataquest was requested by the ITRS Design ITWG to measure designer productivity and to calibrate productivity improvements from major DT innovations. Designer productivity was measured at 4K gates (=16K transistors) per year in 1990, the year in which the so-called "RTL methodology" originated. Succeeding improvements are described in Table DESN12, where the gray items denote ongoing or future DT innovations. The table shows that designer productivity (measured as number of logic gates per designer-year) has grown an average of 39.6% per year from 1990 to 2007. Specifically, the number of designers per million gates (inverse of productivity) went from 250 in the year 1990 to 8 in 2005. Labor unit costs, however, have not remained constant since 1990. According to the *ITRS design cost* model, the historical rate of increase in engineer labor cost is 5% per year (assuming salary and overheads starting at $181,568 in 1990). This assumption may be reviewed in upcoming revisions in light of design globalization trends. While growth slowed to 2% during 2002 thru 2005, it is now back to the typical 5% per year with IC design engineering shortages even being reported from India and China.

INFRASTRUCTURE COSTS

The rate of increase in EDA tool cost *per engineer* is estimated at 3.9% per year (starting at $99,301 per engineer in 1990). But again in 2002 that growth stopped and has been below historical norms for the last six years. The industry is now recovering from this slump, and tool average selling price growth is expected to return to its typical rate of 3.9% in 2008. The total implied infrastructure cost from this item is given by the product of EDA tool cost times the number of engineer-years:

$$EDAInfrastructureCost = \frac{EDAUnitCost \times DesignComplexity}{Designer\ Pr\ oductivity}$$

which relates this cost to labor cost. Other infrastructure costs are assumed to be included as overhead in the current model. Since average labor unit costs have grown faster than EDA infrastructure costs, the proportion of labor in the product development costs is increasing.

TOTAL DESIGN COST

To bring this chapter full circle, we refer again to Figure DESN1, which quantifies the impact of the DT innovations on design cost for the SOC-CP (Consumer Portable) driver defined in the *System Drivers chapter*. The SOC-CP has 17M logic gates in 2007, implying a typical consumer/portable SOC hardware design cost (designers + tools) of almost $15.5M, plus $24.0M in software design cost: in 2007, software design cost exceeds hardware design cost, for the first time in the history of SOC design. Without the six major DT innovations that occurred between 1993 and 2005, the hardware design cost alone for this SOC would be approximately $900M.

Table DESN12 Design Technology Improvements and Impact on Designer Productivity

DT Improvement	*Year*	*Productivity Delta*	*Productivity (Gates/Design-Year)*	*Cost of Component Affected*	*Description of Improvement*
None	1990		4K		
In-house place and route	1993	+38.9%	5.55K	PD Integration	Automated block placement and routing
Engineer	1995	+63.6%	9.09K	Chip/circuit/PD Verification	Engineer can pursue all required tasks to complete a design block, from RTL to GDSII
Reuse—small blocks	1997	+340%	40K	Circuit/PD Verification	Blocks from 2,500–74,999 gates
Reuse—large blocks	1999	+38.9%	56K	Chip/circuit/PD Integration Verification	Blocks from 75,000–1M gates
IC implementation suite	2001	+63.6%	91K	Chip/circuit/PD Integration EDA support	Tightly integrated tool set that goes from RTL synthesis to GDSII through IC place and route
RTL functional verification tool suite	2003	+37.5%	125K	SW development Verification	RTL verification tool (“cockpit”) that takes an ES-level description and partitions it into verifiable blocks, then executes verification tools on the blocks, while tracking and reporting code coverage
Transactional Modeling	2005	+60%	200K	SW development Verification	Level above RTL, including both HW and SW design; it consists of a behavioral (where the system function has not been partitioned) and an architectural level (where HW and SW are identified and handed off to design teams)
Very large block reuse	2007	+200%	600K	Chip/circuit/PD Verification	Blocks >1M gates; intellectual-property cores
Homogeneous parallel processing	2009	+100% HW +100% SW	1200K	Chip/circuit/PD Design and Verification	Many identical cores provide specialized processing around a main processor, which allows for performance, power efficiency, and high reuse
Intelligent test bench	2011	37.5%	1650K	Chip/circuit/PD Verification	Like RTL verification tool suite, but also with automation of the Verification Partitioning step
Concurrent software compiler	2013	200% SW	1650K	Chip and Electronic System Design and Verification	Enables compilation and SW development in highly parallel processing SOCs
Heterogeneous massive parallel processing	2015	+100% HW +100% SW	3300K	System Electronic Design and Verification	Each of the specialized cores around the main processor is not identical from the programming and implementation standpoint
Transactional Memory	2017	+100% HW +100% SW	6600K	System Electronic Design and Verification	Automates true electronic system design on- and off-chip for the first time, including heterogeneous technologies (Phase 1)
System-level DA	2019	60% HW 38% SW	10557K	System Electronic Design and Verification	Automates true electronic system design on- and off-chip for the first time, including heterogeneous technologies (Phase 2)
Executable specification	2021	200% HW +200% SW	31671K	System Electronic Design and Verification	Automates true electronic system design on- and off-chip for the first time, including heterogeneous technologies (Phase 3)
Total		+264,000%			