Final Report: NSF/NSC International Workshop on
Challenges and Opportunities In Giga-Scale Integration For
System-On-A-Chip

1 Introduction

This is the final report of the International Workshop on Challenges and Opportunities in Giga-Scale Integration for System-On-A-Chip, jointly sponsored by the U.S. National Science Foundation (NSF) and Taiwan National Science Council (NSC). The workshop was organized by Professor Jason Cong from the University of California, Los Angeles, and Professors Youn-Long Lin and C. L. Liu from the National Tsinghua University, Taiwan. It was held in Hsin-Chu, Taiwan, during August 24 — 26, 1999. The attendees included:

- **From U.S.:** Chi-Foon Chan (Synopsys), Tim Cheng (UC Santa Barbara), Jason Cong (UCLA), Wayne Dai (UC Santa Cruz), Al Dunlop (Lucent Technologies), Scott Hauck (Univ. of Washington), Steve Kang (UIUC), Cheng-Kok Koh (Purdue Univ.), Massoud Pedram (USC), Miodrag Potkonjak (UCLA), Bryan Preas (Xerox PARC), Steve Trimberger (Xilinx), and Albert Wang (Tensilica).

- **From Taiwan:** Liang-Gee Chen (National Taiwan University), Youn-Long Lin (National Tsing Hua University), C. L. Liu (National Tsing Hua University), Sheng-Chun Lo (Avant!), Wen-Zen Shen (National Chiao Tung University), Jyuo-Min Shyu (Electronics Research & Service Organization), M. K. Tsai (Faraday Technology, a UMC-affiliated design service), Kuen-Jong Lee (National Cheng Kung University, David C. Chen (Macronix), and Ping Yang (TSMC).

- **Observers:** William Chang (NSF) and Robert Grafton (NSF).

The goal of this international workshop was to understand the challenges and critical research needs of design and test technologies for giga-scale system-on-a-chip (SOC) integration in nanometer technologies and identify promising opportunities for innovation. This international workshop was held in Taiwan with the active participation of researchers and engineers of Taiwan, because the dominant portion of the world’s largest and most advanced integrated circuit fabrication facilities are now located in Taiwan. In the system-on-a-chip design era, system integration happens at fabrication foundries during IC manufacturing, as opposed to in system houses through PCB board-level assembly as in the past.

The two-day workshop included a keynote speech by Dr. Chi-Foon Chan, President of Synopsys, presentations by a number of workshop attendees, a field trip to TSMC, the largest IC foundry worldwide, and a number of discussions and brainstorming sessions, which form the basis of this report. A copy of the workshop agenda can be found at the

---

1 This report is compiled and edited by Jason Cong. He can be reached at the Computer Science Department, University of California, Los Angeles, CA 90095 (Email: cong@cs.ucla.edu, Tel.: 310-206-2775).
end of this report. Related information about the workshop, including the biography of each attendee, viewgraphs of the keynote speech and workshop summary, as well as this report, are available at the website
http://cadlab.cs.ucla.edu/~cong/nsf_workshop99/

The remainder of this report will begin with the motivation for this workshop and then will discuss the challenges and critical research needs in the areas of physical design, synthesis, system-level design, programmable components and systems, test and verification as identified by the workshop.

2  Motivation

The driving force behind the spectacular advancement of the integrated circuit technology in the past thirty years has been the exponential scaling of the feature size, i.e., the minimum dimension of a transistor. It has been following Moore’s Law at the rate of 0.7X reduction every three years. It is expected that such exponential scaling will continue for at least another 10 to 12 years as projected in the 1997 National Technology Roadmap for Semiconductors (NTRS’97). This will lead to over half a billion transistors integrated on a single chip with an operating frequency of 2 to 3 GHz in the 70nm technology by Year 2009. In order to sustain such an exponential growth and to achieve giga-scale integration, however, a great deal of innovation is required in the design and test technologies, in terms of both steady incremental extension of the existing design and test capabilities, and the revolutionary development of new design and test paradigm and methodologies. For example, the study by SEMATECH showed that although the level of on-chip integration, expressed in terms of the number of transistors per chip, increases at an approximately 58% per year compound growth rate, the design productivity, measured in terms of the number of transistors per person-month, grows only at a 21% per year compound rate. Such a mismatch of silicon capacity and design productivity, if not resolved timely, will seriously limit the potential of achieving high-degree on-chip integration. Moreover, there are many new challenges associated with giga-scale integration in nanometer technologies such as: lack of high-level design abstraction due to the increasing importance of interconnect performance and reliability, lack of adequate methodology for design reuse, difficulties in integrating and interfacing heterogeneous systems on a single IC, rapidly growing design and test complexity, and so on. These will prevent us from improving design productivity, unless significant progress and innovation are made in design and test technologies.

The objective of this workshop was to understand the challenges and critical research needs of design and test technologies for giga-scale SOC integration in nanometer technologies, and identify promising opportunities for innovation. As the result of two-day intensive discussions at the workshop, a set of important challenges and critical research needs was identified, and divided into the areas of physical design, synthesis, system-level design, programmable components and systems, test and verification. Each will be discussed in detail in the following sections.
3 Physical Design and DSM Effects

With rapid feature size scaling, circuit performance is increasingly determined by the interconnects instead of devices in deep submicron (DSM) designs. Rapid scaling has introduced many challenges on interconnect performance, modeling, and reliability. Since interconnects will not be finalized until physical design is carried out, there is a strong need for more research in physical design with consideration of DSM effects. In particular, the following areas related to physical designs are identified as crucial for system-on-a-chip integration.

- **Timing closure** is the key problem facing today's designers of high-speed VLSI circuits and systems. Timing closure can be obtained not only through development of parasitic-aware, constraint-driven design tools, but also through introduction and adoption of integrated design methodologies and flows. Layout-driven synthesis and synthesis-driven layout are two common ways to approach this problem. Integrated CAD databases, well-characterized cells, and more powerful techniques for concurrent placement/routing and logic optimization are required to achieve timing closure in a cell-based ASIC design methodology.

- **Power closure** is becoming an important design concern in some applications, including battery-powered microelectronics where maximizing the battery service life is one primary driver, and high-end systems where excessive power dissipation limits the number of devices that can be integrated on the same chip. Minimization of switched capacitances during physical design and development of tools and methodologies that support multiple Vdd and multiple Vt or even dynamically varying Vdd and clock frequency in response to varying computational workloads are important steps in achieving power closure. Thermal management is a closely related problem that also needs attention.

- **Power/ground network design, clock distribution, and signal integrity-related problems** are of primary concern in DSM designs. The former includes the IR drop and the bounce problem, whereas the latter includes signal cross talk, noise coupling through substrate, and mixed-signal interference. These problems should be considered, especially in view of the process technology and temperature variations. There is, therefore, a need to develop statistical modeling and analysis tools as well as physical synthesis techniques that are aware of such variations and optimize the circuit accordingly.

- **Integrated package-chip modeling, co-analysis and co-design** are required to provide the accuracy needed for DSM designs. Package design should be considered during IC chip planning.

- **Process migration** is required in many instances where we must retarget an existing design cost-effectively and painlessly to a new process technology or a new cell library.

- **Reliability** must be designed from the early design phase and not as a backend process. This includes, for example, considerations for electromigration, I/O protection circuitry, latchup, hot carrier effects, and so on.

- **Better characterization of small cells and IP blocks** must be provided for power, timing and P/G noise so as to allow more effective design tradeoffs.

- **Performance and reliability-aware design rules** are to be developed to help manage and/or overcome potential signal integrity, power, substrate noise, and reliability problems without having to rely solely on simulation.
On-chip optical interconnects look promising in view of their performance, noise immunity and area comparison with respect to the conventional metallic interconnects. Wave division multiplexing has become a mature technology in opto-electronics and may be applied for on-chip interconnects.

Mixed-signal design issues are critical in many applications. Improved techniques are needed for noise shielding and guard-banding.

On-the-fly extraction and order reduction of parasitic RLC and parasitic device is required to handle the complexities of the DSM designs.

New component models and CAD tools for MEMS and MOEMS are needed to allow integration of mechanical and optical components on chips.

Computationally efficient, highly scalable algorithms for physical design optimization are needed so that they can far extend the capabilities of the current algorithms in handling complex designs and in achieving global optimization.

Integrated and flexible physical design tools that consider realistic operating conditions (temperature variations, process variations, etc.) are needed. Uniform temperature assumption may hide functional failures during simulation.

4 Synthesis

Significant progress has been made in the past two decades in the logic-level and RT-level synthesis, which successfully raised the design abstraction from schematic diagrams to HDLs (hardware description languages). In order to maintain such a high level of design abstractions in the presence of increasing interconnect delays and design complexity in nanometer designs, much research is needed in the following areas related to the synthesis technologies.

Synthesis to support IP creation: Synthesis tools are needed to generate easy-to-use IP blocks for SOC designs. In SOC design, parameterized modules such as application-specific instruction-set processors (ASIPs), signal processing functional macros, application-specific memory modules, built-in self-test (BIST) logic, etc., will be widely used. Advanced synthesis tools should be able to generate IP modules that are not only reusable but also customizable by the users. Accompanying these IPs, the synthesis tools should be able to generate complete test vectors and interface information.

Hierarchical synthesis: Synthesis for the whole chip is a computational intensive process. An effective hierarchical approach that can handle area, timing, and power constraints is the key to making SOC synthesis feasible. At the same time, incorporation of IPs into the synthesis process is also necessary as more and more IPs become easily accessible. The use of hierarchical synthesis makes incremental change more efficient and controllable because it can be done on local modules.

Low power synthesis: Design tools are required to achieve low power dissipation in VLSI circuits. Although some commercial tools address power optimization during technology mapping and gate sizing, they do not address power optimization during such steps as technology-independent logic synthesis, state encoding, scheduling and resource binding, bus architecture design, and so on. Furthermore, the quality of commercially
available automated clock gating tools is poor since they do not use sophisticated data flow graph analysis and make use of statistics of the applied data input to improve the expected power efficiency. Tools and design methodologies that support multiple Vdd and continuously varying Vdd are also lacking.

- **Layout-based synthesis:** It is more and more apparent that interconnect delay and signal integrity will dominate the performance of the design. In order to obtain high performance and accurate design, such physical design effects must be considered during synthesis.

- **Analog synthesis:** Designing or incorporating an analog block plays a critical role in heterogeneous SOC designs. The designer needs an interactive synthesis environment to explore possible architectures for the desired performance. The synthesizer should employ reduced models for efficient optimization runs during synthesis, and provide relevant models (such as timing, power, and layout) for subsequent design steps to use. Note that since analog circuits are very sensitive to noises generated from other parts of the chip, the synthesizer also needs to provide second-order effect information of the synthesized layout for the designer to consider the trade-off of the whole chip performance and the manufacturing yield.

- **Bus and interface synthesis:** It has been long recognized that interface issues are of special importance for complex design, and in particular for SOC. Bus and interconnect often dominate delay, area, power, and other metrics of SOC designs. There is a wide spectrum of important issues, and related challenges and opportunities in bus and interface related problems, including bus protocol specification, bus architecture and implementation, physical interconnect-based bus design, bus performance estimation and evaluation, flexible and fast dedicated interface design, bus and interface wrappers, time-segmented scheduled bus design, and support for bus-related OS and compiler issues.

## 5  System-Level Designs

We have been building large, electronic systems for years, but we have only recently faced the challenge of integrating large systems on a single chip. The giga-scale SOC era will place unprecedented demands on our system design capabilities. A SOC has been defined as the integration of computation engine, memory and logic on a single chip. While this definition may be accurate, it understates the problems facing system designers in the giga-scale SOC era. The challenges facing a SOC designer are more readily conveyed by a thought experiment: Consider the motherboard of a modern workstation, complete with memory, including the operating system and the applications software. Now add analog sensors and actuators and wireless communications. Future systems designers will face this level of complexity. Furthermore, the couplings and interactions among system components will increase as we put more of the system on a silicon die.

The EDA industry is just now facing up to system design issues of this magnitude. This lack of maturity compared to more established EDA fields means that our descriptions of the challenges will be more nebulous and more open ended. We have divided the challenges facing giga-scale SOC designers into the following areas:

- **Design methodology:** We need to design at much higher levels of abstraction.

- **Core-based design:** Much of our design work will be in the design and utilization of reusable cores.
• **Intellectual property issues:** Once the cores have been designed, there are many challenges in importing, exporting, documenting and reusing these cores.

• **Architectural design issues:** We need much better tools and methodologies to support architectural design.

• **Multi-disciplinary design:** Future system designers need to make tradeoffs over a broader range of issues because of the closer interactions required by SOC.

• **Increased productivity:** The need for increased productivity underlies all of the other areas, but we have devoted a separate section to productivity.

• **Miscellaneous challenges:** Some important issues did not fit into broader categories. However, we feel that they are important, so we have grouped them under miscellaneous.

### 5.1 Challenges in Design Methodology

Most people who have thought about design methodology for SOC agree that we need to design at much higher levels of abstraction. However, much research and development is needed before we can make this a reality. In order to benefit from higher-level design, the lower levels must either be previously designed cores or they must be synthesized or composed automatically (or nearly so).

Automatic synthesis is the most recent example where the electronics industry, as a whole, began designing at a higher level of abstraction. This associated design methodology allowed designers to think about the behavior of their integrated circuit rather than the structure of the circuitry. This methodology was supported by a large number of floor planning, placement, routing, modeling, simulation, verification and testing tools that effectively automated design and composition of the lower abstraction levels.

In order for SOC designers to design at higher levels of abstraction, they will need tools and methodologies that allow them to design, compose and integrate system-level components.

As SOC designers integrate more complex functions, it will be increasingly difficult to combine designs from different sources that were designed by different methodologies. We need new languages that express the behavior and structure of systems. We need powerful, automated methods to compose system-level components with widely varying interface requirements. We expect that the wrappers and bridges between bridges will adversely affect performance. We need more powerful inter-block communication protocols and tools to synthesize the chip-level communications.

We are pragmatic enough to expect that this automatic composition of interfaces will not work in all cases. Therefore, we envision the need for reconfigurable logic on SOC. This reconfigurable logic will serve the function of today’s FPGA and CPLD glue logic. Section 6 will discuss the issues related to programmable systems in greater detail.

### 5.2 Issues in Core-Based Design

Today we see a large number of core modules available for designers to incorporate into designs. Examples are embedded processors, media processing functions and signal processors. In the SOC era, because of productivity and time-to-market constraints, the use of core modules will be almost universal.
The design environment will change radically. As a result, we expect fundamental changes in the way designers think about their designs. The time and expense required to develop such sophisticated cores means that these designs must be reused many times. As designers search farther and wider for cores to integrate into their SOCs, it will be increasingly difficult to combine cores from different vendors that were designed using different design methodologies. The various cores to be combined may have varying interface requirements and communications protocols.

Currently, we think of integrated circuit design as a group process. Team members can influence and be influenced by others. In the SOC era, design teams will be distributed around the world. Because of geographical or organizational boundaries, designers may have little or no access to the designers of other components that comprise their system. As a result, designers will engage in defensive design in the same way that system and software designers of Internet protocols engage in defensive programming in the extreme.

We also expect to see a stratification of designers. Currently, we see a stratification of electronics designers into system designers and integrated circuit designers. These two groups use different tools and approach their designs in fundamentally different ways. In the SOC era, we expect a comparable stratification of designers into those who author core modules and those who compose the core modules to make systems.

These issues will force designers to approach their designs in a more structured and systematic way. Specification, characterization and standardization will be much more important than they are today. The heavy investment in core design will force designs through several process migrations.

5.3 IP Related Issues

Intellectual property issues are closely related to those of core-based designs. In fact, some people argue that there is no difference. We have chosen to separate the design and reuse issues (core-based design) from those of importing and exporting system-level components (IP related issues).

IP is crossing organizational boundaries at an ever-increasing rate. This leads to a tension between the creators and the consumers. The creators are interested in protecting the value of their property while consumers are interested in ease of use. We need better methods to protect IP (perhaps through watermarking, fingerprinting, and encryption) while allowing consumers to access the information required to incorporate the IP into their SOC designs.

Improvements are needed in the areas of exporting (implementation, documentation creation, tool support, integrated development environments, test and customer support) and importing IP (evaluation, qualification, and computability).

The cost of design for reuse is two to seven times the cost of design when reuse is not considered. We need to make reuse significantly cheaper for both the producer and the consumer.

5.4 Architectural Design Issues

When designing at higher levels of abstraction, designers will need to explore and analyze alternative system architectures. We need tools to support this architectural design process. We need system-level partitions and synthesis aids. These aids will need to operate on more abstract, higher-level views of the system components. New, higher-level views of the design (for example, IO, behavioral, software, cost, timing and power) will need to be incorporated into the design process. We will need to manage and control power at the architectural level.
5.5 Multi-Disciplinary Design

In today’s system design environment, the design of the system components proceeds relatively independently. For example, the memory, RF, analog, sensor, and embedded processor functions may be designed as separate integrated circuits and composed into a system on a multi-chip module or a printed circuit board. When these functions are integrated onto an SOC, the design will become much more interdependent. The system designer will need to be much more familiar with the details of these functions. Multi-disciplinary optimizations will include packaging, mechanical, sensors, actuators, and electrical properties.

In the SOC era, hardware / software mapping and optimization will be even more important than today. When coupled with the high cost and productivity pressures, we expect hardware / software co-design to expand into co-specification, co-analysis, co-design and co-verification. High-level power management and control will also assume added importance.

5.6 Productivity Issues

Increasing design is probably the single most important issue facing the electronics industry. We must find approaches to design SOCs using fewer engineers and shorter time cycles while being concerned with more complex electrical and physical details. Design productivity will be increased by increasing the efficiency of the team producing the design and by reducing the number of design iterations.

Moving to higher levels of design abstraction and design reuse will provide part, but not all, of the increased productivity.

Verification remains the biggest bottleneck to increased productivity. Design methodologies and processes must avoid problems rather than finding and correcting errors. Designers will need to thoroughly evaluate each component and encapsulate that entity as a known good design. The logical conclusion of this process is to verify once and use the results forever. The issues related to verification will be discussed in greater detail in Section 7.

Details will dominate design in the SOC era, and the simplifications that we use today will no longer be valid. Deep sub-micron effects will force us to adopt an interconnect-centric or communication-centric design, even at the system level.

We will need to effectively utilize loosely coupled design teams from different geographical locations and cultures.

5.7 Miscellaneous Topics in System Designs

As we thought about the issues that will be important in the giga-scale SOC era, we categorized these issues into topics. In the end, we were left with issues that we felt were important, but which did not fit into our other categories. Thus, we have a miscellaneous section.

- **Packaging**: As designers integrate more and more components onto a SOC, packaging will assume more importance. System-in-a-package will be considered as an alternative to system-on-a-chip for performance, power, cost, and reliability trade-off.
• **Error detection/recovery:** As systems become more complex and more integrated, designers will need to include more error detection and recovery. Fault diagnosis and repair may well take on a different meaning when the entire system is a single integrated circuit.

## 6 Programmable Components and Systems

Today, logic designers have a wide range of implementation technologies from which to choose, including processor, programmable logic and ASIC. Each of these methods of constructing arbitrary functionality has a particular point in the system design space for logic speed, density, power, flexibility, and design methodology. Full custom ASICs provide the highest performance and capacity possible, but with high NREs and long time-to-market. CPUs and DSPs trade some of this performance for programmability, allowing a single commodity device to support numerous application domains. These chips also provide very efficient sequential control flow. FPGAs provide a somewhat middle ground, providing some of the performance of ASICs, yet with the programmability of processors. Designers and design tools will be called on to evaluate the requirements of the design and to select the implementation method that best suits it.

Systems-on-a-chip (SOC) design will require the integration of many different types of components, including ASIC logic, programmable processors and regions of uncommitted or application-specific reconfigurable logic. Handling issues of programmability and heterogeneity in SOC designs is a complex, multi-dimension optimization problem. Integrating these resources into future systems requires more than just the inclusion of CPU or FPGA macros into an IP design suite. The custom fabrication of these resources within each SOC design opens the door to radically new optimization techniques. Also, current challenges in CPU and FPGA design will become magnified in future design methodologies, giving added urgency to their solutions.

### 6.1 Styles of Programmable SOCs

It was once possible to categorize chips into separate classes of functionality: ASIC, programmable processor, DSP, FPGA. However, chips today are becoming more hybrid devices, integrating many different types of resources into a single piece of silicon. Just as microprocessor systems became many and varied, so will programmable logic systems. While we will likely still see purely CPU and purely FPGA chips with ever growing capacity and performance, hybrid devices will become common. Applications will drive architectures, as complete systems demand many different types of resources on a single device. These hybrid devices bring challenges and opportunities.

Current FPGA offerings can be classified as *prefabricated programmable* devices where the manufacturer selects the amount of programmable logic and the way that programmable logic is combined with other logic in the device. As applications migrate away from ASIC designs (due to the huge NREs, long time-to-market and design detail required for deep sub-micron implementation), these applications push programmable logic systems to include application-specific components. Existing programmable logic devices include microprocessors and bus interfaces. A future *field-programmable system-on-a-chip* (FPSOC) for wireless applications might include programmable processors and reconfigurable logic together with domain-specific RF and analog components. Designers of these programmable
systems on a chip have a difficult task, both in selection of the collection of components and in the integration of those components into a single device.

For some applications, performance, power or other concerns will require an ASIC implementation mask programmed specifically for that application. However, portions of the design that require flexibility and can tolerate the programmable logic overhead can be mapped to programmable processors or programmable logic inside a mask-programmed device. Since these chips will be custom fabricated for an application, the programmable resources should be optimized to better suit the application’s requirements. In such programmable-in-ASIC designs, major challenges will include the optimization of programmable resources to the application’s needs, integration of programmable logic with more traditional ASIC, and partitioning of designs among integration of these resources.

Both FPSOC and programmable-in-ASIC seek to merge the power, performance and capacity benefits of custom logic with the flexibility and time-to-market of programmable solutions, providing much of the benefits of both technologies. These two techniques differ less in their result than in their architectural model: FPSOC consists of a general-purpose programmable logic device that includes fixed function units. A programmable-in-ASIC device is a design for a specific customer application that includes areas of programmable logic. There are research challenges not only in developing a programmable SOC, but also unique challenges in each of these domains.

6.2 Challenges and Opportunities in Programmable Systems Methodology

Many of the issues in SOC design apply to field-programmable devices as well as mask-programmed ASICs. Programmable logic designs must deal with multi-million gate designs, tight performance constraints and power consumption limits. As programmable devices include fixed logic blocks, designs and tools must deal with them. Also, the potential integration of analog circuitry or MEMS onto the same silicon will require very careful attention to inter-resource interactions in all areas, from process technology to overall chip architecture.

Intellectual property (IP) cores, complete functions to be dropped into a design, will also target programmable logic, so issues of interface and system verification with IP apply to programmable logic as well as mask-programmed logic. In many cases these IP cores could have many different implementations, potentially with mappings to ASIC, FPGA, and CPU resources all possible. We have no design specification that permits efficient porting of designs between any two of these. We would like to port among all three, designing without regard to the implementation.

Pre-designed, programmable components can address some problems of deep sub-micron design, allowing IP cores that are more portable. Because the programmable base array is pre-designed, it can be laid out in advance to minimize crosstalk and other effects. Interconnect performance degradation can be characterized in advance. As a result, designs implemented in programmable logic need not address these issues individually, and can be designed with a design abstraction that does not need to take into account the subtleties of deep sub-micron design.

Integration of FPGA, ASIC and processor on a single die significantly increases the bandwidth available when compared to a multi-chip solution, but inter-resource communication is still likely to be a major potential bottleneck in the system design. Determining an efficient interconnection scheme to minimize these restrictions will be an important consideration. The programmability of programmable logic adds flexibility, but opens new questions. For example, does a microprocessor view programmable logic as generic logic or as a programmable execution unit?
The integration of different resource types gives rise to interesting new interactions. For example, a complete system-on-a-chip requires even greater consideration of overall testability, since the increasing logic resources increase the amount of testing required. Programmable resources (both CPU and FPGA) provide new mechanisms for increasing testability and potentially new methods for self-testing. Programmable resources become obvious targets for mapping test circuitry, because test circuitry in programmable logic can be overwritten with application logic after testing is complete, so it incurs no area or performance overhead. It can identify problems in both the programmable and non-programmable parts of the device. Programmable logic can perform parametric as well as functional test. Judicious application of these technologies, as well as an integration of these techniques with more established hardware testing methodologies, will be required to provide efficient debugging and fault detection for future SOC designs.

One can also consider using the reprogrammable resources to provide more graceful system degradation in the face of design, fabrication, or run-time faults. For example, if a given fixed hardware resource was found to be faulty (either by incorrect design or by chip faults) its computations could potentially be migrated to programmable resources, either by reusing programmable resources (though potentially degrading chip performance) or by using spare resources provided explicitly for this purpose. The portion of a design implemented in programmable logic can be re-mapped to avoid defects. Techniques need to be developed to detect such faults during system operation and to automatically or manually invoke this self-repair process. Also, efficiently providing programmable spares on the chip will require complex design decisions, both because enough resources must be provided to meet expected demand, and because these spares must be provided with adequate interconnections to the rest of the system to allow them to actually replace that faulty functionality.

Integrated power management in future systems will also be a critical concern. While numerous techniques have been considered for ASIC designs, low-power architectures for reconfigurable resources (especially FPGAs) are still in infancy. Power management in a heterogeneous system-on-a-chip design will require much more complex tradeoffs than in a single-functionality design, and all components must be optimized if the overall power consumption is to be improved. Here too, the integration of heterogeneous resources also provides new opportunities. For example, there is the notion of reducing power consumption by reducing the amount of computation resources employed during low demand portions of the device’s operation, when the resulting slowdowns will not be a concern. With multiple potential implementations of a given functionality, either in fixed logic, reprogrammable logic, or reprogrammable processors, one can consider migrating complete computations to more power-efficient resource classes during reduced demand periods. In general, an integrated power management strategy that optimizes each resource type, while taking advantage of the potential of each resource, will be critical to best optimizing overall system power consumption.

Programmable logic components can take advantage of reconfigurable logic techniques that have been the target of significant research throughout the 1990s. However, reconfigurable logic continues to suffer from lack of tools and design methodology. There is no efficient computational model to use to divide logic in time to use reconfiguration generically. There is no model for integrating time-varying logic in design flows or systems, so each use is a custom design and integration task.
6.3 Challenges in Prefabricated Programmable Devices

Many current systems are built out of prefabricated components: general-purpose processors, DSPs, FPGAs, and fixed-function chips for various needs. As chip capacity increases, we will see the integration of many of these separate components into single prefabricated programmable devices. Different applications and domains will require different amounts of RAM, CPU, FPGA, fixed functional units and other resources. For example, wireless applications, prototyping, and telecommunications/networking each place different demands on chip resources. Since these domains are large enough to support custom chips and have requirements for which such optimizations can provide significant benefits, we can expect separate product offerings tuned to each of these application domains. The base array, as well as the collection of IP blocks, can be optimized to the target application domain. The number of possible combinations of different components is staggering.

Software techniques can be developed to help reduce the number of different combinations of required resources. For example, techniques have already been developed to harness unused resources of one type as extra resources of another style. Some FPGAs currently allow unused logic blocks to be used as memories, and unused memories as large logic functions. Developing new techniques that blur the boundaries between resource types will increase the utility of these devices by helping designers squeeze more logic into the devices and improve performance.

Programmable logic devices have the possibility to significantly change the hardware development cycle. Users of programmable processors employ a much different design style than ASIC designers. In software development, code is developed, mapped to the programmable system, and tested in actual operation. Flaws are discovered through experimentation and then rectified in the system design. This incremental, iterative design methodology allows a fast development cycle, with less concern for first-time success. In contrast, ASIC designers must strive for first-pass success, since the first instantiation of a design involves huge NREs and significant lead times. Programmable logic permits an incremental methodology more like programming. Existing ASIC design methodology has few tools to exploit this style of design. Although tools exist for incremental physical design, we are missing tools to minimize changes through synthesis. Tools for in-situ debugging these self-emulated designs are also missing in the design flow, though logic analyzers are often part of the post-design characterization flow.

Field programmable systems allow post-deployment upgrades of the hardware resources, such as currently found in the download of software patches over the Internet. Whether such a complete change in design methodology will be appropriate for hardware development is unclear. However, what is clear is that it offers new opportunities in complete system design, with differing strengths from current design strategies. Integrating such technologies into more traditional ASIC flows will be a major challenge to the community.

Prefabricated programmable devices are pre-tested to guarantee functionality regardless of the design implemented in them. If programmable logic eliminates the per-design test considerations, we now have another design tradeoff -- zero test cost in programmable logic against the speed and density of ASIC. How do we take advantage of this?
6.4 Challenges in Programmable-in-ASIC

Although power, performance or area constraints may require a custom ASIC, it is likely that some part of every system can be best supported on a programmable substrate. Since the chip will be custom fabricated, one can consider optimizing the programmable components, both processor and FPGA, to the target application. Not only can we choose the relative amounts of programmable and fixed logic, but we can also optimize the programmable logic structures to the specific application. This can be as simple as the embedding of fixed logic structures into programmable units, all the way to the large-scale reoptimization of logic resources and interconnect structures to the chip’s application. This will require methods for optimizing the architecture, as well as customization of tools for automatically implementing the desired functionality on these optimized resources.

There is also a need to estimate the chip resource requirements in advance of full system implementation. One of the major advantages of programmable resources is the ability to adapt to changes in circuit requirements, even allowing some portions of the design to be left undefined until after chip fabrication. However, techniques for ensuring that adequate resources are present for the system is a complex estimation problem, but one that is critical to solve in order to take full advantage of this capability.

With the inclusion of programmable subsystems into a system-on-a-chip design, the efficient mapping of tasks to these components becomes a major concern. Given an overall computation that must be achieved, it is critical to determine what type of resource to use for each portion of the operation. The tradeoffs are many and varied. Programmable resources can provide post-fabrication customization and flexibility for different computations, yet may require greater area, power, and delay than ASIC logic. Custom hardware normally provides the fastest and densest implementation, but requires significant development time up-front and limits post-fabrication flexibility. Microprocessors provide a well-defined computation model and effective handling of complex decisions and data structures. Currently, the partitioning of the system into different components is done manually with little or no automation. Here is an important area of research.

6.5 Summary of Research Needs for Future Programmable Systems

The inclusion of reprogrammable resources provides significant opportunities in future chip designs. Programmable logic provides another implementation target for logic, so tools are needed for estimating the amount of different types of logic, for integrating them and for trading them off against one another.

The flexibility of programmable logic admits new a design methodology more similar to software development than to traditional ASIC design. The incremental, iterative design style potentially affects every tool in the current tool chain, and needs much research. Programmable logic permits new ways of looking at test and built-in-self-test.

By allowing post-fabrication customization, new paradigms of circuit generation and execution become available. Designs may be partitioned in time and run in pieces serially on programmable logic. Systems including programmable logic may be repairable in the field. These unique capabilities need to be considered in the design tools and design methodologies for future SOCs.
If programmable logic is included in application-specific devices, the architecture of that logic may be optimized to the application domain. Tools and methodologies for systematically customizing and optimizing a programmable architecture for a given application domain need to be researched and developed.

7 Test and Verification

Verification and test are increasingly becoming bottlenecks in designing highly complex integrated systems. This section discusses future challenges in verification and test, and identifies opportunities for research and innovation.

7.1 Challenges and Opportunities in Functional Verification

Verification is increasingly a bottleneck in the design process due to the following reasons:

- **Limited capacity**: Formal verification has very limited capacity and the current techniques are not quite scalable to larger designs.
- **Demands of SOC**: Simulation/emulation cannot keep up with the demand of SOC, as simulation provides low coverage and remains expensive and requires significant effort in generating test benches.
- **Lacking general simulation framework**: An efficient mixed-mode/mixed-level simulation framework is lacking for heterogeneous SOCs consisting of both digital and analog components described at several different levels of abstraction.

Given these challenges, we have identified the following areas that present needs and opportunities for research and innovation.

- **Hybrid, semi-formal verification**
  
  - There is a need for an integrated framework for simulation, emulation and model checking which verifies properties with hybrid techniques.
  
  - We need solutions to automatic test bench generation from behavior/RTL code to minimize the manual effort in test bench generation and to improve the quality of simulation-based verification.
  
  - New techniques are needed to improve the efficiency and capacity of formal verification engines for the applications of model checking and symbolic simulation.

- **Multi-level/mixed-mode/mixed-technology simulation framework**
  
  - We need solutions that support modeling of vastly different functional blocks, predict interference between the blocks and simulate the whole chip accurately within reasonable time.
  
  - Such a framework should also include optimization capability to trade-off among design quality factors such as area, power, signal integrity, and manufacturing yield.
7.2 Challenges and Opportunities in Test

Deep sub-micron technologies and high integration of system-on-chips present challenges to test in a number of areas. The research community must cope with an enormous spectrum of difficult problems ranging from, for instance, high-level test synthesis for core-based design, to noise and power dissipation problems in extremely high performance (in reality analog) pin electronics. The following list highlights some key challenges in test.

- **Automatic test equipment is reaching limitations in testing high-performance devices.** On chip clock speed increases dramatically while the tester overall timing accuracy (OTA) does not. This trend implies increasing yield loss (estimated to be up to 48% by 2012) as guardbanding to cover tester error results in the loss of more and more good chips. Limited capacity and bandwidth long test application time and excessive cost of ATE also contribute to the increasing overall test cost.

- **Integration of complex, heterogeneous components in SOC posts serious test problems.** Cost-effective methods to test embedded analog, mixed-signal and RF components and devices using multiple technologies, multiple logic families and even multiple voltage levels are needed for such heterogeneous chips. Volume of test data per gate is expected to remain constant, and therefore much more data needs to get across the chip boundary, resulting in bandwidth problems.

- **Deeper submicron technology results in new failure modes.** Process variations are now more likely to cause devices to marginally violate the performance specifications. Testing has to target not only spot defects but also such parametric performance failures. A new class of noise faults caused by excessive noise such as power bus noise, substrate noise, crosstalk-induced delay faults and distributed delay defects needs to be properly modeled for the applications in fault simulation, test generation and design for testability.

- **Paradigm shifts.** Innovative test research is needed to address the challenges outlined above. Without new test methods, testing may become the key showstopper to limit future design and manufacturing technologies. Test technology must be able to support higher-level design and test handoff. Testability must be analyzed/inserted early in the design process of chips and systems. To overcome the ATE limitations, the test systems must become simpler and ICs must be capable of more self-test. Even though ATE will always be required, its role will change in the future. Finally, many of today’s deep submicron design validation problems will become tomorrow’s test problems, and test research should start addressing them.

Given these challenges, we identified the following areas of needs and opportunities for research and innovation in test in the SOC era.

- **Testing and diagnosis for deep sub-micron SOC**
  - Defect characterization, fault modeling, testability analysis, vector generation and self-test techniques for emerging DSM defects such as noise- and coupling-induced faults and parametric faults caused by local process variations.
  - Diagnostic methodologies/tools for defects in DSM devices for shortening process/design debugging time and for monitoring manufacturing defects
• **Test solutions for analog/mixed signal/RF circuits**
  - BIST solutions for analog and analog-digital interface blocks in mixed-signal designs: For most mixed-signal
    SOCs, the analog circuitry accounts for only a small fraction of the total silicon area, while a major fraction of
    the test equipment investment and test time is devoted to the analog parts. Therefore, analog BIST, which
    could significantly reduce the need for expensive external analog testers, shorten the test time and minimize
    the measurement noise, offers great promises.
  - Modeling, design for testability and BIST for high frequency (to GHZ range) RF circuits.
  - On-chip measurement techniques for high-speed serial communication links. Testing devices for applications
    such as Gigabit Ethernet and Firewire require very expensive test systems capable of generating and receiving
    differential signals up to several GHz with voltage swings as low as 100 mV. The excessive test time and cost
    makes existing solutions impossible for volume production.

• **Performance/delay testing**
  - Self-testing techniques for delay faults that apply delay tests on chips with greater accuracy than available at
    the tester.
  - Delay test techniques for multi-clock designs (synchronized or not), including on-chip generated clocks.
  - Delay test solutions incorporating interconnect, including coupling capacitance, supply line effects and
    resistive bridges for many levels of metal.
  - High speed/frequency testing, paying attention to energy consumption during switching, careful consideration
    during ATPG of race conditions, multi-cycle delay paths, etc.
  - Delay fault models and test strategies for new technologies such as silicon-on-insulator.

• **New current-measurement-based test strategies** for deep sub-micron, low voltage, low threshold CMOS
  circuits: Due to the scaling down of supply voltage and transistor threshold, background IDDQ increases
  inexorably and the spread of IDDQ distribution is also increasing. IDDQ testing must be adapted to exist in an
  environment of decreasing signal-to-noise ratio, or be replaced with a better-suited method that maintains its
  effectiveness in defect screening and reliability prediction.

• **System testing for SOCs**
  - Testing strategy/standard for embedded IP blocks.
  - Full-chip self-test techniques/methodologies for SOC.
  - Built-in self-test, built-in self-diagnosis and built-in self-repair techniques for deep sub-micron, embedded
    memories.
  - Test scheduling and power/thermal management for SOC: power consumption during test must be carefully
    considered so that testing does not put the system into modes that consume excessive power/current or cause
    excessive noise resulting in soft errors.
  - SOC test synthesis tool development.
8  Concluding Remarks
The exponential scaling of the integrated circuit technology results in the integration of hundreds of millions of transistors on a single chip, which provides the possibility of integration to a highly complex, heterogeneous electronic system on a single chip. To achieve such giga-scale integration, however, requires a great deal of innovation in the design and test technologies, in terms of both steady incremental extension of the existing design and test capabilities and the revolutionary development of new design and test paradigm and methodologies. Significant investment is needed in these areas in order to extend the Moore’s Law into the next two decades. This workshop identified a set of key challenges and opportunities in physical design, synthesis, system-level design, programmable components and systems, verification and test for giga-scale system-on-a-chip integration. We hope that this workshop and this report can stimulate an increasing level of research activities in these areas to propel the advance of design, verification, and test technologies for future system-on-a-chip designs.

9  Acknowledgements
This workshop was made possible under NSF Award 9908422. The workshop organizers would like to thank the U.S. National Science Foundation and the Taiwan National Science Council for their support of the workshop. They would also like to thank Dr. Chi-Foon Chan from Synopsys for delivering a stimulating keynote speech despite his busy schedule, Patty Huang from the National Tsinghua University and Jie Fang from UCLA for assisting in the organization of the workshop. Each area covered in Sections 3 to 7 corresponds to a discussion group at the workshop. These discussion groups were coordinated by:

- Physical design: Steve Kang
- Synthesis: Jyuo-Min Shyu
- System-level designs: Bryan Preas
- Programmable components and systems: Steve Trimberger
- Testing and verification: Tim Cheng

These individuals spent much effort in moderating the discussions, summarizing the results, and writing the corresponding sections in this report. Scott Hauck also contributed greatly to Section 6. These efforts are greatly appreciated.

Appendix: Workshop Schedule

24 August (Tuesday)
- 6 to 8 p.m.: Reception

25 August (Wednesday)
- **Morning**: Invited talks and position statements from selected workshop attendees (open to the public)
- **Afternoon**: Visit to TSMC and presentations from TSMC on their view, approaches, and needs for SOC and IP integration
- **Evening**: Working dinner -- identify top ten problems for design and test technologies for SOC designs

26 August (Thursday)
- **Morning**: Partition into 3 to 4 working groups and group discussions to identify challenges and opportunities, and give recommendations
• Afternoon:
  - Discussion on implication of graduate education to meet giga-scale integration for system-on-a-chip and possible future US/Taiwan research/education collaboration in this area
  - Produce workshop report draft
• Evening: Banquet, closing statements