Keynote Paper

An Outlook on Design Technologies for Future Integrated Systems

Giovanni De Micheli, *Fellow, IEEE*

***Abstract*—The economic and social demand for ubiquitous and multifaceted electronic systems—in combination with the unprecedented opportunities provided by the integration of various manufacturing technologies—is paving the way to a new class of heterogeneous integrated systems, with increased performance and connectedness and providing us with gateways to the living world. This paper surveys design requirements and solutions for heterogeneous systems and addresses design technologies for realizing them.**

***Index Terms*—Computer-aided design, cooperative engineering, electronic design automation (EDA), gene regulatory networks, Lab on Chip (LoC), microarray, nanoarchitectures, nanoelectronic, nanotechnology, networks on chips, System on Chip (SoC), VLSI, 3-D integration.**

I. Introduction

ABOUT 60 years after the invention of the transistor, solid-state electronics has revolutionized our lives. Indeed, our hands and eyes interact on a daily basis mainly with objects that have an electronic dimension. We need electronic technology to move (e.g., vehicles), to interact (e.g., communicators), to learn (e.g., computers and databases), and to relax (e.g., broadcast, games), just to mention a few activities. The electronic technology has deeply permeated the society, and its impact is largely positive in many metrics.

It is therefore interesting to project the future of electronic technology, with specific reference to its objectives and impact on society. This analysis will help us to understand which technologies will be needed to progress further. When referring to "technologies," this word is used in the broad sense. It encompasses manufacturing and design technologies, the latter being the enabling methods to carry complex projects to completion. From a practical standpoint, software means are ubiquitously used for the conception, design, and run-time operation of most electronic products. Software design tools, also called *computer-aided design* tools and methods, have been instrumental in the growth and success of electronic products. They realize algorithms and procedures that are rooted both in formal methods (e.g., mathematics) and in design experience (e.g., heuristics). Broadly speaking, I will refer to this body of knowledge as *design technologies* (DTs). Such "soft" technologies, together with the "hard" *manufacturing technologies* (MTs) are tightly interrelated.

It is the purpose of this survey to address the future evolution of electronic design and its technologies, with specific attention to DTs. For this reason, I will consider first the evolutionary path and forecast the growth and needs of electronic systems. Evolutionary technologies involve mainly silicon-based electronic design, possibly while considering the addition of new materials and devices to enhance the system capabilities. Mainstream electronic designs are realized by *Systems-on-Chips* (SoCs) and *Systems-in-Packages* (SiPs) that push the limits of integration. Most SoCs and SiPs are multiprocessor-based systems (i.e., MPSoCs), and thus require careful architectural considerations in hardware and in software to deliver the desired performance to the applications.

Revolutionary technologies for electronic design can take several incarnations, some of which are related to how novel *nanotechnologies* can be exploited to realize systems. While this is yet to be proven, new nonsilicon-based MTs will require correspondingly new DTs. Moreover, there is a strong trend and interest to interface electronic technology to living matter, for purposes that span a wide range of applications, including bioanalysis, medical implants, and neural interfaces. *Laboratories-on-Chips* (LoCs) represent the future evolution of SoCs, and require a heterogeneous MT as well as new DTs. With this perspective in mind, we can speculate on the growth of DTs in the coming years, on the research challenges and on the commercial opportunities.

I think it is extremely stimulating and exciting to foster the growth of DT and of the *electronic design automation* (EDA) industry beyond the current domain limited to chip design and production. Indeed, the knowledge accumulated through decades of R&D in DTs is deep and broad, and provides us with a solid framework to tackle products in new domains.

This survey is organized as follows. Section II presents the broad objectives for new electronic products, the state of the art of new technologies and their limitations, the requirements for SoCs in terms of performance, power consumption and reliability, and the broad trends in DTs. Section III describes some architectural solutions that are applicable to current and evolutionary technologies, and that can be embodied by crossbar

Manuscript received August 7, 2008; revised January 9, 2009. Current version published May 20, 2009. This paper was recommended by Associate Editor L. Scheffer.

The author is with the Institute of Electrical Engineering and the Integrated Systems Centre, INF 314-Station 14, Ecole Polytecnique Federale de Lausanne, 1015 Lausanne, Switzerland (e-mail: giovanni.demicheli@epfl.ch).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCAD.2009.2021008

structures for computation and storage and by network-oriented communication schemes, which are eventually applicable to 3-D integrated systems. Section IV focuses on heterogeneous integration and shows both hardware and data management challenges for LoCs by means of a few illustrative examples. Finally, Sections V and VI address distributed systems embedding SoCs and LoCs as components, and the challenge of cooperative heterogeneous engineering.

II. Challenge

A. Social and Economic Pull

We have in front of us some audacious goals for electronic systems. I will mention just a few, as examples of the drivers that pull the marketplace. The global economy and society demands that each human is reachable: Language barriers are still an impediment for the largest part of the world population. Real-time natural language translation, achieved by portable devices, is a major objective for the engineering community. Many years of research in this domain [2], [28], [45] have shown that the problem has solutions, but the required computational effort is high. This objective represents an important driver for multiprocessor architecture and related technologies.

Connectivity, anywhere and anyhow, is an important problem. We would like to connect each human on the planet, and thus solve the last-mile problem. Cellular technology has been instrumental in addressing this problem in developed and developing countries; still, the world has significant blackout areas. Moreover, data communication requires addressing issues of privacy and security, particularly in the case of personal (e.g., medical) information.

We are using an increasingly larger number of untethered devices and energy supply is a major issue for various reasons. First, we pay a price—in terms of nonrenewable energy consumption—for each service provided to us from electronic products, and we need to reduce the ecological cost of using technology. Next, we need to charge, replace, and dispose of batteries. Last but not least, it is inconvenient, and sometimes impossible, to refurnish energy (e.g., change batteries) in some circumstances. Energy-efficient designs, drawing a minimum energy consumption, have been a goal of researchers for over two decades [99]. Nevertheless, we are still far from being able to design systems that can *harvest* energy [96] from the environment enough to be independent from energy sources. This requires a specific perspective on how hardware and software are conceived, and this issue has been addressed by several researchers, e.g., [22], [100], [113].

The use of electronic systems for biodiscovery, health monitoring, and improvement as well as environmental monitoring [94] is also an important goal with large social-value added. The integration of sensing in SoCs and the use of suitable materials and nanotechnology have opened the door to an unprecedented wave of innovation, also addressing the social need for new smart electronic products.

In summary, the economic and social pull of electronics is strong: We will need increasingly larger processing power and efficiency as well as new means to communicate and to interact with the environment. There is no reason to believe that we can be satisfied with computing and communication systems as they are now; indeed, we are just in the infancy of information age.

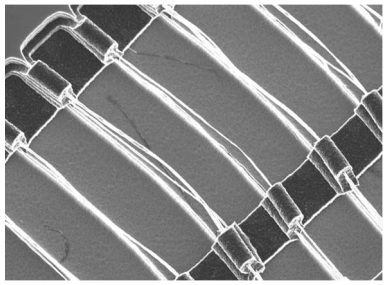

Fig. 1. SINW with Gate-All-Around transistors (Courtesy of D. Sacchetto, EPFL).

B. Technology Push

At the time of this writing, CMOS circuits designed with the 45-nm node are in production and use. There is strong reason to believe that the 32/22-nm CMOS nodes will be realized within five years [60]. Still, technical and economic difficulties plague the technology growth. Since the setup of manufacturing in each new technology node requires large capital expenses, few companies—mainly nowadays linked by technology alliances—can afford advanced design. This reduces unfortunately the pool of suppliers of chips in advanced technologies, and consequently the market for EDA tools.

Much has been said about the potentials of nanotechnologies to improve electronic chips. Before delving into this issue, one has to realize that current 45-nm CMOS processes are considered as nanotechnologies, even though these processes are an evolution of current MTs. There is a continuum spectrum of solutions between evolutionary and revolutionary technologies. The fact that a technology becomes "disruptive" is related to extrinsic factors such as the ability of lowering significantly the cost of fabrication, power consumption, and/or raising performance. Most advanced technology studies on CMOS beyond the 45-nm node already use a plethora of materials and plan on using tridimensional transistors, thus departing from the principle of planarity that has characterized silicon technologies.

Recent research on *silicon nanowires* (SINW) has shown the possibility of achieving transistors with interesting characteristics, such as abrupt transitions in the I–V plane which is important for low-power dissipation [87]. Moreover, nanowires can be arranged to create integrated computation and wiring structures, thus supporting regular and predictable design methodologies (Fig. 1). It is also important to remember that SINW mix and match well with CMOS [38], [44], and thus support the realization of special-purpose macros.

The future of *graphene* and *carbon nanotubes* (CNTs) is promising but still hard to predict. Metallic CNTs perform well as interconnect, due to the high thermal and electrical conductivity. Semiconductor CNTs can be used to create transistors, even though an important question is the ability to design large-scale circuits with them. Hybrid technologies that use silicon and CNTs are subject of current investigation, as CNTs provide us with switching devices with higher carrier mobility (as compared to silicon). The current difficulty in realizing long straight CNTs has prompted design methods and tools to achieve correct and robust design. An example is the choice of specific layout styles and rules to avoid spurious connections [16], [97].

Molecular switches provide another interesting technology. Some molecules, like *rotaxane*, have two stable states: one

conducting and one nonconducting. When these molecules are placed in specific positions, like at the cross-point of a wire array, their state can provide a means for storing information or for performing computation. Indeed, architectures reminiscent of nonvolatile memories [20] or *programmable-logic arrays* (PLAs) can be designed efficiently with molecular electronics [73].

The confinement of semiconductor carriers along the three spatial dimensions gives rise to *quantum dots* that can operate as *single-electron transistors* and show the Coulomb blockade effect. Quantum dots can implement *qubits* for *quantum information processing*. Whereas the potentials of this technology is large and disruptive, its practical applications (except for quantum cryptography) are still far on the horizon. Nevertheless, the interest in the quantum computing paradigm has already spurred research on DT for circuits implementing qubits, e.g., in physical design [78], clocking [133], and synthesis [116].

When facing the prospect of using new MTs for future SoCs, a few important questions come to mind. First of all, are these new technologies apt (and ready) for system design? We have seen some circuit demonstrators, but designing robust large-scale systems—as done in CMOS—requires a set of specific characteristics. The large investment in capital and expertise in CMOS leads us to think that it will be difficult for a new technology to replace CMOS. On the other hand, CMOS enhancements with the help of new nanotechnologies are likely to happen. There exist already several examples of hybridization of technologies, such as using nanowires together with CMOS cells [39] and CNTs to provide interconnection on chips [26].

An important issue is the DT support for these new technologies. From a superficial look at the issue, it would seem just a question of changing the back-end of physical design tools. However, when looking at the integration of logic and physical synthesis, as well as variability and dependability issues, it seems appropriate that we should rethink all the way in which systems are conceived and synthesized.

Heterogeneous integration is another direction of growth of silicon and postsilicon technologies to support the combination of electrical and mechanical components, called *micro/nanoelectrical mechanical systems* (M/NEMS). Accelerometers, such as those used within the Nintendo WII console and in vehicle airbag control are the most common MEMS examples. MEMS are also used as energy harvesting devices, by providing a moving system connected to an electrical generator stimulated by environmental vibration [83]. A notable example of harvester-powered systems are automatic (batteryless) quartz watches, where the electronic quartz-controlled circuit is energized by the movement of the human arm. *Micro/nanofluidic systems* provide the means to transport and process chemical (biological) samples on a chip (see Section IV for details). As in the case of M/NEMS, these systems can be combined on the same substrate hosting electronic circuits or not, depending on the objectives. Heterogeneous integration poses several design challenges that range from design methods and tools to advanced integration and packaging technologies.

C. Requirements for Micro/Nanoarchitectures

The electronic market is driven by two conflicting goals: achieving *high performance*, as required by multimedia and gaming systems, and achieving *low-power* consumption, as needed by all portable systems. In general, both objectives need to be met simultaneously, as high-performance systems cannot afford high-power consumption and high temperatures (for reliability reasons), and mobile devices need to support complex software applications requiring thus high-performance processing.

In general, low-power consumption is achieved by operating SoCs at low voltage, in combination with voltage (and frequency) scaling and gating. Ultralow power consumption systems will operate at low voltage (few tenths of a volt) and possibly CMOS transistors will be in weak inversion. The operation at low voltage will impose a limit on the maximum operating frequency, and thus on the performance delivered by a processor. Therefore, multiprocessing is needed to achieve high performance. At the same time, scaling allows us to pack many processor cores on a single chip.

The trend toward multiprocessing SoCs (MPSoCs) is also due to addressing *reliability* problems that may arise from applying deeply scaled technologies to *life-critical* applications. As scaling may lead us to transistors and interconnect with higher *failure rates*, system-level reliability can be insured by *redundancy*, such as having spare processing cores and reconfigurable interconnect means [82]. Nevertheless, reliability enhancement by parallel computation is a complex task [125], and DTs can be very instrumental in designing and coordinating software for this objective.

As a result, there is a strong tendency in compensating the limitations on clock frequency with multiprocessing [127], [136], as witnessed also by the personal computer market. Whereas MT and DT for multiprocessing are well developed, their efficient use is still limited by the technology of parallelizing software applications (e.g., parallelizing compilers), which is still in the infancy, and by the limited experience with parallel languages and programming environments. Moreover, software applications and operating systems need to be rethought for multiprocessing platforms, and thus multiprocessing does not deliver yet the expected gain on standard applications.

D. DTs Trends

DTs for integrated circuits flourished in the 1970s, and led us to both a solid understanding of the theory and practice of *modeling, analysis*, and *synthesis* of circuits and systems as well as to the EDA industry that supplies tools and flows [108].

The technical base of DT comprises algorithms and software systems. Some algorithms for EDA evolved from classic algorithms in computer science (e.g., shortest path) and specialized for the particular problems of interest, while some others (e.g., layout tools) were invented to address circuit design issues. When scientists realized in the 1970s that exact solutions to most DT problems could not be achieved because of intrinsic computational complexity [51] and large scale, heuristic solutions flourished to provide engineers with practical (e.g., fast) approximate solution methods to design problems. Thus, the wealth and particular flavor of EDA is the ability to tackle complex large-scale problems and to provide effective solutions. Other fields, like computational biology, have much to learn from EDA in this perspective.

It is interesting to notice that in the last 20 years, some problems were solved exactly for most instances (e.g., two-level logic minimization [107]), due to the use of smart algorithms and/or data structures (e.g., *binary decision diagrams* (BDDs)

[17]) and to the availability of larger memory spaces and faster processors. Still nowadays, there are a small number of problems for which an exact solution is attainable for a significantly large number of instances. Therefore, from a practical standpoint, the use of heuristics and approximation algorithms is widespread.

It is interesting to note some recent trends in EDA. An avenue of research is to develop robust exact solvers for fundamental problems, such as *satisfiability* (SAT) [3], [52], [88], [119], and then map other problems to SAT. This approach has been very successful for problems that are intrinsically difficult but that have structure and a limited number of constraints. Examples are in various fields, including verification and model checking [62], test generation [123], and physical design (e.g., routing) [92].

Another interesting trend is motivated by variability of MT. Due to the lack of precise knowledge of some quantities (e.g., gate and wire delays), statistical design methods have become mainstream. The most well-known example is *statistical timing analysis* [15], where critical path delays are computed based on statistical models of gate delays. A further example is *statistical logic synthesis* [66], where a gate-level interconnection is constructed while taking into consideration the statistical variations.

Variability problems can be mitigated by using *self-calibrating* circuits and related DTs for their design and insertion. These approaches exploit run-time adaptation and address both variations from die to die as well as variations due to environmental factors (e.g., temperature) and/or aging, and thus remove the conservative assumption of using worst-case delays. An example is the use of variable voltage swings on buses to minimize power consumption, and the application of error detecting codes to insure correct communication as well as a means to regulate the voltage to keep the error frequency within bounds [59]. Another example is given by the "razor" methodology [41], which was conceived to yield correct processor operation with overaggressively low-voltage supply and later found many other important embodiments.

A generalization of these problems is captured by the generic name of *dependable design*, where dependability is a catch-all term encompassing reliability, availability, and safety [68]. As an example, *design under uncertainty* addresses the problems of MT variability, changes in environmental conditions, as well as nondeterminism in the design specifications. Designing "reliable systems with unreliable components" is a specific problem that we are facing nowadays in view of the possibility that components (including processor cores) deteriorate and fail at run time [137].

Many DT problems are computationally intractable because of their discrete nature. When dealing with statistical models, the problems often relate to optimizing expected values of continuous variables. Thus, optimization (or decision) problems become easier to be solved numerically in an exact fashion. An example is the use of *stochastic optimum control* techniques for *dynamic power management* [9], [104], [120], where the expected value of power consumption (or latency penalty) can be minimized subject to latency (or power consumption) constraints. Indeed, these problems can be mapped to linear programs and solved exactly and effectively. I expect that the paradigm of optimizing expected values of system observables to be generalized to various problems and various types of observables.

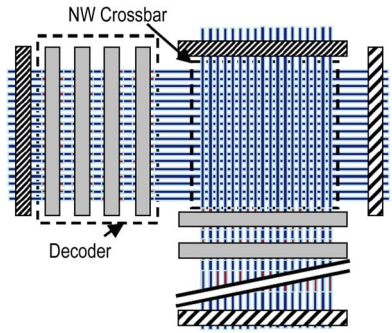

Fig. 2. General structure of a crossbar computational array and decoder.

III. Architectural Solutions and Related DTs

Current and future design of circuits and systems has to cope with two major problems: 1) predictable design in terms of timing and 2) fast design closure. The cause of the first problem is the variability of delays and the unpredictability of the wiring structure after physical design. Indeed, a large portion of delays is in the interconnection wires, whose pattern is unpredictable when using standard-cell or gate-array design styles and their derivatives. Fast design closure is a prerogative of the synthesis tool flow that should allow a designer to complete a design with a limited number of changes to meet timing requirements. Clearly, the two problems are intimately related.

Architectural support for predictable timing and fast design closure is applicable to both computational and communication structures in SoCs. Storage structures, such as memory arrays, have since ever been designed with rectangular, regular, and predictable structures. Within current CMOS technologies, the use of regular circuit fabrics has been subject of research in the last two decades [101].

A. Crossbar Array-Based Design

There is a renewed interest in crossbar array-based design for realizing computational units. A crossbar is a rectangular mesh of wires designed in two orthogonal directions (Fig. 2). The crossbar is personalized by switching devices positioned at the crosspoints; their pattern is often abstracted as the *personality matrix* of the array. The persistence and completeness of the personality distinguishes an array dedicated to computing from a read-write memory array. A computing array is dedicated to implement a specific logic function, its personality is fixed and its dimensions are related to the function being implemented. Reconfigurable arrays add the twist of being able to change the personality during operation. Crossbar arrays are reminiscent of *PLAs*, often used in the 1970s because their regularity eased physical design (in a time when physical design was much less developed as compared to now). PLAs lost ground to standard cells and other styles because of flexibility and the need of either dynamic or power-hungry pseudo-NMOS operation in CMOS technology. Interestingly enough, Mo and Brayton [86] revisited recently the use of PLAs with the objective of exploiting their regularity to achieve predictable timing. DeHon [31] proposed the use of PLAs as computational structures for molecular electronics and nanowires. His work, deeply inspired by self-assembly, is based on the conjecture that computational structures can be created with a function unknown *a priori*, and then programmed to match the functional requirements.

When looking at future opportunities in nanoelectronics, it would be desirable to be able to use crossbars with connections (e.g., nanowires) that can have dimensions beyond the optical lithography limit. Nevertheless, a reasonable assumption is that the overall wiring structure in a SoC is still limited by lithography. Thus, an embedded nanoarray would have the appeal of being compatible with standard MT, while having a smaller size and, therefore, higher computational density. Based on this conjecture, there are two important problems to be solved: 1) dealing with manufacturing defects and failure rates in the nanoarray and 2) interfacing the nanoarray to the external circuit. The former problem can be addressed by storing redundant information in various ways, including the use of spare rows and columns. Examples of these techniques are reminiscent of those used to test PLAs offline or online [128].

The latter interfacing problem is a new and critical problem presented by nanotechnologies. It encompasses the issues of connecting wires (called *meso* and *nano* wires) of mismatching sizes and electrical driving strengths. Moreover, the external decoding and interconnection has to be such that no area is wasted around the nanoarray. In other words, if the interconnection structure is conservatively designed, then the area advantages of using a nanoarray vanishes out. Likharev and Strukov [71] proposed various interconnection schemes for nanoarrays based on a rotation of the nanoarray axes against the mesowire frame of reference to provide efficient interconnection.

The architecture of decoders for interfacing to nanoarrays is an important issue [12], [105]. All proposed solutions, either suggested as concepts [32], [56], [109], or implemented on real chips [8], rely on the principle of a linear array of transistors that can be aligned with the nanowires. Beckman *et al.* [8] demonstrated a robust, pitch-, and technology-independent technique, which, in turn, needs a larger decoder than the theoretically achievable size. High-density decoding can be achieved by using multivalued logic to address the nanoarray (which can store binary or multivalued information). Moreover, the addressing scheme can be made robust against threshold voltage variations by using specific encoding scheme that exploit redundant information [12].

B. Networks on Chips

Also in the case of *on-chip* communication, timing predictability and design closure are extremely important. The advancement of MT in terms of integration leads us to SoCs with many (e.g., 10–1000) digital units (e.g., processor cores, controllers, storage, application-specific units) that need to be interconnected in an efficient and reliable way. Moreover, SoC architectures are often heterogeneous, i.e., units have different sizes and the communication requirements differ radically from point to point. The *network-on-chip* (NoC) technology developed rapidly in the first years of the millennium [11], [30], [54] and addresses three major design requirements: 1) realizing a modular and structured interconnect scheme, thus addressing predictability and timing closure issues; 2) overcoming the limitations of standard buses that do not scale up in terms of connected components as far as performance and power consumption are concerned; and 3) addressing reliability issues in the interconnect by providing path diversity as well as a layered approach to error detection and correction. With technology scaling (e.g., below the 65-nm node), the use of NoCs becomes increasingly more compelling [102]. Whereas NoCs address current design needs in CMOS, they will be essential to connect nanoarrays and 3-D systems as mentioned in the next section.

There are different flavors of NoCs according to the functionality and market of the related SoC [33]:

General-purpose on-chip multiprocessors are high-performance chips that benefit from spatial locality to achieve high performance. They are the evolution of on-board multiprocessors, and they are typified by having a homogeneous set of processing and storage arrays. For these reasons, NoCs are typically structured as meshes or regular network architectures, reminiscent of those used for on-board multiprocessors, with the appropriate adjustments to operate on a silicon substrate. The main purpose of the NoC is to sustain high-performance computing. As an example, Intel realized in 2007 a large chip with 80 cores that are interconnected by a NoC [131], [132].

Application-specific SoCs and *platforms* are hardware chips dedicated to an application or to a family of applications, such as GSM/UMTS telephony. In most cases, as for all mobile applications, energy consumption is a major concern, and a major objective of the NoC is to support low-energy communication. Often, these systems contain fairly heterogeneous computing elements, such as processors, controllers, DSPs, and a number of domain-specific hardware accelerators. This heterogeneity may lead to specific traffic patterns and requirements, thus requiring NoCs with specialized architectures and protocols. Examples of these type of SoCs are the Aethereal architecture [53] and the BONE series of chips [67].

Field-programmable gate arrays (FPGAs) are hardware systems where the functionality is determined after manufacturing by connecting and configuring components. Components vary in size and in functionality (e.g., Xilinx's Virtex family) and are connected by reprogrammable networks. These networks are simple and provide bit-level connectivity with little or no control, thus having only few prerogatives of NoCs. Nevertheless, we expect FPGAs to grow substantially over the coming years, include complex cores, and thus require effective structured communication as provided by fully fledged NoCs.

In general, a distinguishing characteristic of NoCs is low latency of communication and corresponding streamlined protocols. NoCs can be made modular and be built out of a library of few programmable elements. For example, the *xPipes* library [5] consists of network interfaces, switches, and links. The network interfaces encapsulate the digital units and act as protocol converters, thus transforming the processor core interface standard protocols (e.g., OCP [95]) into an *ad hoc* internal NoC protocol. The switches route the information and can be embodied in various ways according to the overall design objectives. The physical links realize the interconnect and can be pipelined to operate across significant on-chip distance. Differently from general networks, a specific feature of NoCs is their ability to be tailored to the architecture at hand, and their parameters be optimized to satisfy the given design constraints.

As a result, an important problem is how to design NoCs from high-level specifications, while incorporating network specialization and optimization of hardware components and protocols. NoC synthesis is the most recent evolution of

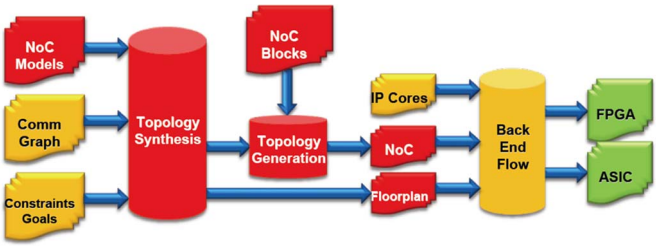

Fig. 3. Flow for NoC Design (Courtesy iNOCs, Inc.).

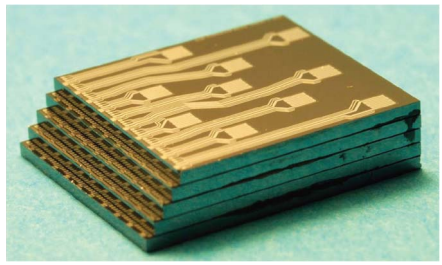

Fig. 4. Three-dimensional test structure to investigate heat transfer (Courtesy of Y. Leblebici).

physical design, with the perspective of placing and linking hardware units with the complexity of processor cores. Nowadays, there are new design flows and tools that allow us to implement NoCs starting from high-level specification, as well as emerging start-up companies. As an example, a simplified version of a NoC tool flow is shown in Fig. 3.

There are several challenging problems related to DTs for NoCs. Some are connected to modeling, i.e., how to capture the communication structure to be realized by a NoC and the related constraints (e.g., traffic, speed, jitter, etc.). High-level NoC synthesis problems relate to the choice of topology, and to the corresponding selection of the routes for the data. Topology and floorplanning determine the macroscopic figures of merit of the design. Detailed NoC synthesis include the choices of routers, links and buffering scheme/size, the tuning of the transmission parameters, and the insertion of mechanisms to insure fault tolerance. The automation of these steps [14], [90] is key to design closure as well as to achieving low-power and high-performance implementations. A large amount of research on NoC architectures and DTs is carried out at several research sites. The interested reader is referred to [6], [61], and [77] for a survey of the recent activities, as well as for an extended list of references.

C. 3-D Integration

The trend of moving from planar to *3-D integration* is fueled by several reasons, including complexity, packaging constraints, and heterogeneity [79]. Even when considering electronic-only SoCs, just combining analog and digital parts makes the monolithic integration difficult, because of differences in supply voltages and noise issues. Moreover, memories (volatile and nonvolatile) are best realized with MTs different from those used in processing, and experiences in embedded DRAM design have shown that process compromises are not satisfactory. Radio frequency circuits and micro/nanointegrated antennas require their own manufacturing steps. Thus, it makes more sense to realize different functions in different chips, and then enclose them together in a package. 3-D integration differs from SiP solutions because the various chips are stacked upon each other and interconnected mainly by *through-silicon vias* (TSVs) [115]. This technology provides an efficient way to realize physical routing in three dimensions, and enables the packing of complex and diverse functionalities in a minimal space and with shorter interconnections as compared to planar chips.

There are several DT challenges in designing 3-D integrated systems. The first one is related to thermal management. Indeed, 3-D integration exacerbates the heat distribution and extraction problem of planar chips. A simple computation can show that while a planar chip can easily produce 100 W/cm^2, a 10-chip 1-mm stack can produce 10 kW/cm^3, which is much harder to dissipate. Cooling can be done in various manners, including the use of microfluidic flow in silicon to extract heat [58]. Proactive cooling can be achieved by limiting the heat generation by dynamic thermal management, which can also be used to determine the heat profile inside the structure. Thermal management can operate on hardware by applying *dynamic voltage/frequency scaling* (DVS/DFS) [10] as well as on software by allocating active jobs to processor cores in various parts of the system, thus effecting temperature-aware load balancing [89] (Fig. 4).

Thermal control is extremely important in 3-D systems, as component failure rates grows exponentially with temperature. Whereas typical failures are due to electromigration and dielectric breakdown, thermal stress can induce also failures in chips. Thus, the control of the temperature profile in time is essential to avoid stress cycles on materials [82], [121]. In this respect, thermal management differs from load balancing as well as from power management. Indeed, the choice of a dependability-oriented objective function, such as maximizing the *system mean time to failure*, makes the problem and its solutions different from standard DVS/DFS approaches.

Physical routing in 3-D systems is complex, because of the added vertical dimension, and of the constraints on the positions of the TSVs and their timing models. Moreover, it is plausible that chips are designed and manufactured to be used in different 3-D systems to reduce their manufacturing cost. This flexibility on the mixing and matching chips poses constraints on the position of the TSVs. Needless to say, reconfiguration means for the vertical interconnection channels are crucial for achieving system-level flexibility. In this perspective, the concept of NoCs for 3-D systems is extremely powerful, because it embeds the notion of modularity and (run-time) reconfigurability.

Recent research efforts have been addressing 3-D NoCs [27], [72], [112], [144]. Relevant problems are the modeling and management of the anisotropic delays, the physical design of the NoC with specific reference to floorplanning and global routing of the links through TSVs (with possible limitation on count, size, and positions), and the corresponding impact on NoC architectures. Moreover, testing 3-D chips poses interesting problems because of the alignment and yield of the TSVs as well as controllability and observability in three dimensions. Thus, the overall design of a 3-D NoC requires solving a set of high-level design issues simultaneously. While this DT problem is computationally challenging, the payoff of finding a NoC that fits a 3-D system is a significant reduction of energy dissipation and satisfaction of aggressive timing goals.

IV. From SoCs to LoCs

An LoCs can be seen as the integration of chemical and biological manipulation on an intelligent substrate [76]. In general, LoCs support microchemistry, and thus can be used for analysis and for synthesis [7] of compounds. In the former domain, LoCs can be used for biodiscovery, environmental monitoring, and medical diagnosis. As for other electronic products, volume production and corresponding competitive cost will be key for acceptance at points of care, to enable faster, cheaper, and more precise diagnosis, as well as at other locations, such as mass transport facilities, for effective health control and pandemics prevention. Moreover, LoCs can support local and/or distributed computation and access to databases, thus enhancing the effectiveness of diagnosis. This technology can be multifaceted and serve various objectives: It is important for advanced countries where the cost of health care is skyrocketing as well as for developing countries where it is very important to bring medicine to an affordable cost to everyone.

LoCs have many interesting technical features. They show the ultimate hybridization of technologies. Their range of complexity varies. Components that can be integrated in a LoC include, but are not limited to, *microfluidics* parts for sample transport, sensors to detect proteins/DNA, low-noise electronics, and on-chip data processing algorithms and software to elaborate the biological information. LoCs can be programmed (at various levels) to do specific tests. Thus, we can envision *field-programmable LoCs* that can be set to do a specific experiment, such as looking for specific compounds in water, according to the circumstances. As in the case of FPGAs, flexibility, programmability, and volume production reduce the *nonrecurring engineering costs* per unit and are enablers for the broad use of this technology.

A. LoC Design

A complete description of the MTs for LoCs goes beyond the scope of this survey (see [129] for details). I will present here just a simplified view of some examples to motivate the use of DT in this domain.

Sample transport can be achieved in different ways. Biological samples can be moved on chip by fluidic convection, by electric [35] and/or magnetic [69] means. Micro/nanopumps can be realized on chip as well as channels on layers above functionalized silicon or amorphous material. Magnetic fields can move samples that are attached to *microbeads*. As an example, Fig. 5 shows samples that are moved by means of a magnetic field generated by spirals that are designed on the top metal level of the chip [70]. With this technology, it is possible to achieve transport, split, and merge of droplets over a 2-D array. This involves to *schedule* and *route* the transport of multiple samples at the same time, while avoiding collisions [25], [141].

DTs for capturing, realizing, and optimizing microfluidics systems have been studied by various authors. In general, you can view the *fluidic path* as the counterpart of a data *path*: Both require a corresponding control unit. The design of such a *control unit* can borrow DT from standard circuit design. As an example related to microdroplet processing [37], the fundamental operations to be performed by the fluidic path are the following: MIX (mix two droplets), SPLIT (split a droplet), INPUT (define a reference port), MOVE (transport a droplet), and PATH (define a path). These operations are related to a digital (i.e., quantized and not continuous) view of microfluidics and operations can take different time to execute, even though these "execution times" can be integer multiples. The design and optimization of a fluidic path consists then in implementing a behavior, defined by a fluidic flow (represented by a partial order as in a data flow) and by resource constraints, where resources are placed in the array where droplets are processed. Ding *et al.* [37] researched the optimal flow in a microfluidic circuit by mapping it into a scheduling problem and modeling it using *integer linear programming*. He then applied the method to realizing a *polymerase chain reaction* fluidic circuit. Beyond this example, design frameworks starting from high-level models in languages like SystemC [142] have been defined for microfluidics and the corresponding codesign problems formulated and solved.

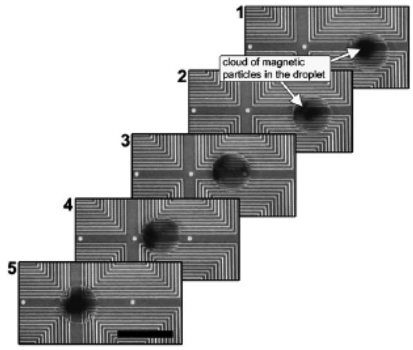

Fig. 5. Droplet moved on a chip surface by magnetic fields [70].

Overall, we can see operations in microfluidic systems as related to processing, storage, and transport, such as in electronic circuits. Let us focus now on how processing is done in the specific case of sample recognition. Fig. 6(a) shows how a DNA strand can bind (i.e., hybridize) to a complementary probe. A conceptually similar, but more complex mechanism, can be used to trap proteins, e.g., by using antibodies as probes. With these working principles, *microarrays* can capture in parallel biological samples, thus becoming key instruments for *high-throughput biological* experiments [1], [36], [98] as well as for medical diagnosis. Whereas this technology has reached some maturity, most commercial products use optical techniques to read microarrays. Namely, samples are tagged with fluorophores, and the hybridized array is scanned optically, yielding a set of colored pixels to analyze. These techniques are bulky and hard to integrate in a monolithic chip. For this reason, *nonlabeled sensing* techniques have been proposed, where the matching of a sample to a probe creates a reaction (e.g., redox) yielding a variation of an electrical quantity (e.g., impedance, capacitance, current) that then can be measured by placing the sensor under the probe itself [13], [19], [85]. With this technology, it is possible to integrate sensing with readout electronics and signal processing on the same substrate [55], [117], [122]. Moreover, probes are organized as arrays, thus enabling parallel sampling.

A key problem for probe arrays is avoiding false positive and negative readouts. Thus, the design of probe arrays may involve the accurate choice of the probes and of redundancy mechanisms. Indeed, it is convenient that the presence of a target is

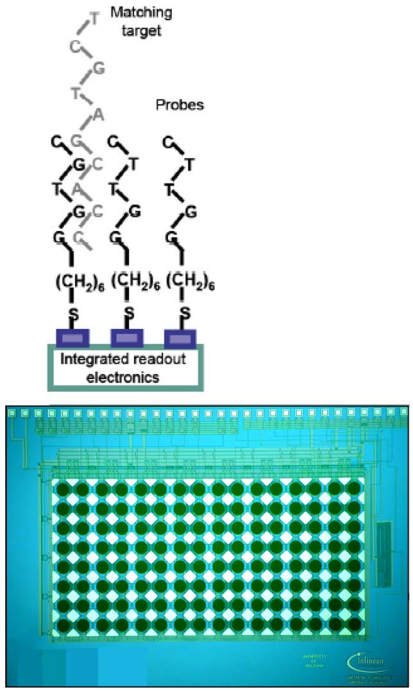

Fig. 6. (a) Nonlabeled sensing principle. (b) DNA probe array [122].

detected by multiple matching occurrences at various probes. Similarly, to avoid spurious matching, probes for various targets may be designed to differ significantly to avoid readout errors. From a DT standpoint, the problem may be formalized by a *0–1 incidence matrix*, where rows are *targets* and where columns represent *probes*. Whereas such a matrix represents all possible matching conditions, an optimal design is represented by a subset probes and targets, such that each target is recognized by at least n probes and each target matches uniquely at least m probes. These constraints embed a notion of safety margin, to make false positives/negatives unlikely to happen. The corresponding array optimization problems relate to maximizing the rows (targets) of the implementation submatrix and/or minimizing the required columns (probes) [110]. Both problems can be related to optimization problems in logic synthesis and/or graph theory and solved exactly or heuristically [50].

B. Data Analysis

Since probes are usually organized into arrays [Fig. 6(b)], the result of an experiment is a matrix of real numbers giving the expression levels of the quantities being measured. Typically, a matrix is organized as a set of rows and columns, representing genes (or proteins) and samples, respectively (or vice versa). A graphic rendering of this matrix, called *heat map*, is shown in Fig. 7, where colors encode the expression levels. The interpretation of microarray data is important, as it provides the means of extracting a biologically significant answer from a data set. Much research has been done in the last decade, and it is summarized next.

The task of analyzing microarray data can be done by using some techniques based on *clustering* which are reminiscent of some methods used within electronic DT (e.g., for partitioning and placement) [75]. Clustering is an *unsupervised learning* technique that groups subsets whose elements are closer (in some metric) among each other as compared to elements

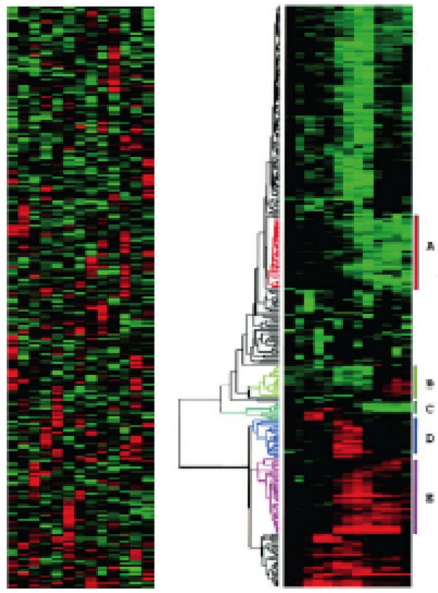

Fig. 7. (a) Graphic rendering of gene expression levels. (b) Clusters of expression levels [40].

across the subsets. Several clustering techniques have been applied to microarray data, including *hierarchical clustering, self-organizing maps,* and *principal component analysis* [103]. From an intuitive standpoint, clustering can be visualized by performing a matrix row and column permutation that clusters together areas with similar colors [Fig. 7(b)]. Nevertheless, by carefully analyzing the problem, the issue is to find subsets of rows and columns, displaying genes (or proteins) that have consistent behavior under a set of varying conditions. The corresponding mathematical problem, called *biclustering* [134], [138], consists of finding submatrices of the original data matrix displaying closeness according to a metric. For example, δ*-biclusters* are submatrices so that the difference in variation in expression levels across two genes and two conditions is bounded from above by a constant δ. Alternatively, biclusters can be defined in terms of *coherence* [23], by requiring the *mean square residual* of a submatrix to be lower than δ. *Maximal* δ-biclusters are matrices that are not contained in any δ-bicluster matrix. The search of the maximal biclusters is key to understanding the underlying biological regulation problem. It entails covering the microarray data space by (possibly overlapping) biclusters that represent coregulated trends. Yoon *et al.* [138], [140] pinned down formally the definitions and properties of biclusters, and developed an algorithm that can compute and rank all maximal biclusters. The algorithm relies on a compact data representation based on *zero-suppressed decision diagrams* [84] and on symbolic set manipulation.

In general, clustering can be applied to data in different forms, and in particular to genetic measurements of sequences of experiments done at different time points. In this case, the *time series* of genetic expression values can shed light on the evolution of biomedical experiments and the underlying gene regulatory mechanisms. In *clinical genomics*, the *supervised* analysis of microarray data combines genetic information with the information gathered by the long practical experience coded into clinical traits. Early attempts were based on statistical correlation methods, such as using Spearman's coefficient. Examples of recent work have established correlation between human

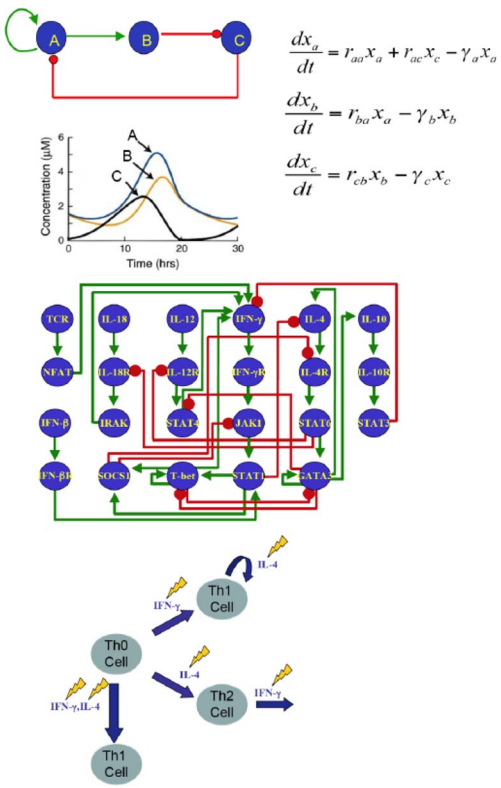

Fig. 8. Abstraction levels in biology. (a) Biochemical model. (b) Zero-delay model [80]. (c) Functional model.

genomic data and radiological traits [93]. In these studies, a set of clinical traits were manually extracted from medical images and then correlated to the data extracted from microarrays. In this context, various types of algorithms have been devised, such as *significance analysis of microarray* (SAM), [130] and *gene expression enrichment analysis*, [124]. These algorithms extract a ranked set of gene candidates to be significantly related to a given external clinical trait. An interesting approach is coclustering [139], which operates as follows. First, a correlation matrix is constructed from the genetic and clinical data in matrix form using the statistics defined in SAM. Next, δ-biclusters are searched for in this correlation matrix. This method was tested on data from *Acute Myelogenous Leukemia* (AML) [93], yielding 43 clusters, some of which with strong biological significance. As an example coclustering showed that the trait "survival" is correlated with genes that play a central role in the control of growth, differentiation and morphogenesis of normal and malignant cells.

C. Modeling, Abstraction and Analysis

Gene and protein arrays are some of the means to extract information about biological processes. Some open databases (e.g., KEGG [64]) are large repositories for biological information. A distinguishing characteristics of biological processes is the very large amount of data to be manipulated in the attempt (not always successful) to understand the biological functions. As in other domains, abstraction and modeling are crucial for attaining the desired results. Within biological processes, several abstractions can be established. In the sequel, I will give examples of three major abstraction layers, as shown in Fig. 8.

The *biochemical abstraction* layer models the dynamics of the chemical reactions. Timing is an essential ingredient of this modeling layer, and differential equations are the natural mathematical formalism [106]. Bioanalysis at the biochemical layer entails solving large sets of differential equations. The *zero-delay abstraction* [63], [126] is reminiscent of logic modeling of circuits. In this abstraction layer, the interaction among biological compounds is reduced to causes/effects and transition timing is abstracted away. In other words, a condition (or state) can be activated or inhibited by one (or more) other conditions. Thus, zero-delay (i.e., logic) models of biological systems can be represented by directed graphs, where often edges have an annotation denoting that the edge has an *activation* or *inhibiting effect*. These models are reminiscent of *finite-state machines*. Moreover, they can have a synchronous or asynchronous semantics [49]. In the former case, all transitions are assumed to take place simultaneously, thus replacing the detailed timing information by a common timing-quantum for all transitions. Asynchronous models are more accurate: Despite the fact that exact timing information is not used, asynchronous models assume different timing for different transitions, or equivalently that at each "equivalent synchronoustic" only one transition can occur [46]. In the *functional abstraction*, we represent the input–output relation of biological process. Often, we are only interested in the final state (states) produced by a biological process under specific stimuli.

As an example, Fig. 8(b) and (c) relate to the evolution of *T-helper cells*, which play an important role within the immune system. T-helper cells can be grouped into precursor Th0 cells and effector Th1 and Th2 cells [91]. From a molecular standpoint, Th1 and Th2 differ in their patterns of cytokine secretion and the evolution of Th0 into either type is important. From a functional standpoint [Fig. 8(c)], the evolution is enabled/inhibited by the presence of compounds (e.g., Interleukin4, Interferon γ). Nevertheless, it is important to understand the steps (i.e., transitions) that correspond to the overall evolution: This is shown by the zero-delay model shown in Fig. 8(b). Here, each state corresponds to the presence of a compound. In this model, states can abstract the *expression level* of a compound by a binary value (i.e., 1 or 0, expressed or not expressed) or more generally by multivalued discrete [47] or continuous variables. There is of course a tradeoff between accuracy of the representation and effectiveness of the computation.

Zero-delay models for biological process are receiving an increasingly larger attention, particularly in view of the search of systemic properties [4], [42]. Indeed, performing zero-delay simulation to understand system behavior and separately computing the detailed timing models of the biochemical reactions corresponds to achieving the *orthogonalization of concerns* as used nowadays to design and verify complex SoCs [65].

Since logic-level models of biosystems are represented by logic equations (similar to logic networks [34]), their simulation with untimed or timed models is straightforward. Nevertheless, the sheer size of these networks makes simulation runs very long. Moreover, biologists are often interested in the final outcome of the network evolution, possibly under some specific stimuli and/or network modifications. *Network traversal* by implicit methods applies well to biological networks, when an appropriate model [46] of the transition relation of the finite state system is provided. Such models are typically provided by

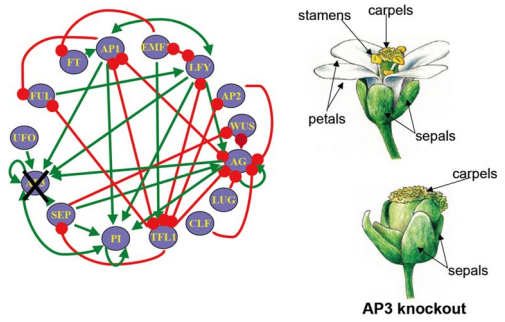

Fig. 9. GRN for *Arabidopsis Thaliana* and result of knock out of gene AP3 [42].

biologists, even though a promise of microarray-based methods is to extract *gene regulatory networks* (GRNs) automatically from biological samples.

In essence, a GRN can be modeled by a set of states (corresponding to the gene expression) that are represented by binary or multiple-valued variables, and a set of logic equations (expressing activation and inhibition) that yield the state values at the next time-point of interest. Traversal is useful to determine the final steady states of a network. Formally, these states are defined to be the set closed under the forward image operation, and such that once one of its elements is reached, the probability of revisiting it is one. Steady states can have different topologies, ranging from simple self-loops, to loops of simple states and to nested loops [49]. The complexity of natural systems (much higher as compared to finite-state controls of engineering systems) gives rise to these topologies. Yet, traversal [24] is a very efficient method to determine the steady states of these networks, by using forward and backward image computation based on BDDs.

Network traversal provides also an effective framework to study the behavior of differentiation and mutants. Indeed, it is quite straightforward to emulate *in silico* the effects of constraining a gene (or protein) to a constant value. This is reminiscent of the *stuck-at* model used in testing of integrated circuits. For example, a *knock-out* experiment is the result of silencing a gene, i.e., setting its expressed value to zero. Fig. 9 shows the GRN for the flower *Arabidopsis Thaliana* and the effects of knocking out gene AP3 [42]. In general, traversal tools have shown to be very effective to measure the differential behavior originating by constraining the GRN state set.

A fundamental and difficult problem is the acquisition of good models for GRNs with high confidence levels and assess their robustness, i.e., their correctness in view of possible changes of the working hypotheses. Various models have been proposed, ranging from *Bayesian networks* [43] to *probabilistic Boolean networks* (PBNs) [118]. As an example, PBNs can express nondeterminism, i.e., multiple behaviors. In this case, the state behavior (e.g., gene) can be expressed by multiple Boolean functions, each function with an associated probability. PBNs can be represented in an implicit (or explicit) way, and it is possible to compute effectively the steady states under the given probability distribution [48].

Overall, the objective of computer-aided bioanalysis are many and multifaceted. They range from the rational design of drugs (e.g., *pharmacogenomics*: a drug therapy which is cognizant of the patient genotype) to the deep understanding of biological mechanisms. Bioanalysis is also a key supporting discipline for *synthetic biology* [7] which is designing and engineering systems with biological components. Not surprisingly, as in the case of electronic design, the understanding of the biological mechanism unleashes the way to develop biological components for computation. This area is important, complex and growing, and the interested reader is referred to [7] for further details.

D. Networking SoCs and LoCs

There is a strong trend in designing and deploying distributed systems that use SoCs and LoCs as components. The application fields can vary, ranging from environmental monitoring systems [57], [94] to wearable health systems [94], [114] and to computer-assisted driving and navigation, just to mention a few examples. Distributed systems involve networks of nodes, which can communicate via standard (e.g., wired, cellular) or *ad hoc* networks. In particular, *wireless sensors networks* (WSNs) have been a subject of extensive research [18], [81], and their advantages and limitations have been reported. Currently, one of the most pressing issue for WSNs is to provide very large data bandwidth (e.g., visual information) with limited energy consumption.

Within distributed systems, an important issue is how to localize data processing, storage and communication. Therefore, two factors are extremely important: data *abstraction*, because data needs to be condensed before transmission, and data *integration*, i.e., data interpolation or extrapolation to fill in the voids for missing data samples. For most systems, the likely paradigm of choice is *distributed intelligence*, i.e., reasoning and acting locally with some global information. In this perspective, DT challenges relate to distributed-system modeling, including constraints and their verification.

The quest for *energy efficiency* is extremely important [21], because of direct and environmental costs. Along this line of thought, WSN need eventually to be autonomous. Energy has to be *harvested* from the environment, in the case of both mobile and fixed applications [83]. Whereas several researchers have designed and realized (integrated) energy-harvesting devices, there is still a significant gap between the available and required energy levels (e.g., from one to two orders of magnitude).

Interestingly enough, energy harvesting can be seen as converting unused (or degraded) energy into information. Conversely, as energy distribution must be efficient (e.g., in the smart home, building, factory, electrical grid), it is necessary to use local information to optimize energy distribution. Therefore, in this case, information is converted into energy saving. Indeed, electronic systems, whether distributed or not, are machines whose efficiency is ruled by the laws of thermodynamics. The mutual relation between energy and information can be managed by *policies* that control the run-time system execution. Policies for run-time *energy and information management* are an extremely important subject, and they represent the evolution of policies for power management [10].

Finally, system-level dependability is extremely important for networked embedded systems. This problem is multifaceted, as malfunctions can stem from hardware, software, and communication problems. Nevertheless, the distributed nature of processing and storage, the network topology, and the related communication protocols offer the technical means to deliver reliable system-level services.

V. Human Factors: Cooperative Engineering

Cooperative engineering is a key factor in achieving the vision of distributed embedded systems, particularly those that monitor biological information and/or interface with living beings. Indeed, the required technical skills to design and run such systems are shared among engineers, computer scientists, chemists, physicists, biologists, and medical doctors. It is very important to find ways of translating specific technical idioms and to provide means for researchers with different backgrounds to communicate. For this reasons, abstraction and modularity of information about system design and operation is extremely important. In the past, DT for integrated systems has encompassed hardware/software codesign techniques. This notion has to be generalized to the concurrent design of complex multifaceted systems. The ability of determining models and interfaces among various systems aspects is a key aspect of DT of the future.

Currently, a few multidisciplinary research programs are tackling the design of distributed systems (embedding SoCs and LoCs) and the related DT [18], [57], [94], [135]. Among these, the *nano-tera.ch* [94] program addresses bettering human health and monitoring the environment, by developing micro/nano/info technologies that enable to design and manage distributed embedded systems. Another noteworthy program is the *humanitarian technology challenge*, which is a new partnership between the IEEE and the United Nations, with the objectives of identifying the technologies in the health/environment domain that can benefit developing countries. Examples include food, water, and health monitoring. Both programs have ethical objectives that can raise enthusiasm as well as broaden the perspective of engineers.

VI. Conclusion

The growth of integrated circuits into SoCs and LoCs, and their use as embedded components in distributed systems has opened unprecedented opportunities for research and development. The challenges to design and operate successfully such systems are huge, because of the system heterogeneity and of the wide body of competences required.

This survey started by elaborating on the numerous possibilities that stem from silicon and postsilicon technologies in the nanometer range of feature sizes. It deals mainly with evolutionary technologies that may add to and transform the current CMOS processes into versatile platforms that combine computation with micro/nanomechanical components, sensing, and fluidic transport. The merger and hybridization of technologies will support the design and manufacturing of new families of integrated systems, with a much higher level of complexity as compared to current SoCs.

The successful design of advanced SoCs and LoCs requires bold steps in architectural organization as well as in DTs. Along the former avenue, structured organization of computational elements (by regular fabrics and/or array-oriented computational units) as well as structured communication (by means of NoCs) are key to manage possible showstoppers such as complexity and variability. Modular 3-D stacks can be the answer to the combination of heterogeneous technologies, such as those needed to realize LoCs with local processing.

Most SoCs and LoCs will embed wireless interfaces to increase their autonomy and portability. The creation of networks of SoCs and LoCs will be a key enabling technology to address global problems such as health and environmental management. To be successful, a few hurdles must be overcome, such as achieving ultralow energy computation and energy harvesting as well as providing dependable computation and communication.

DT support is crucial for realizing such distributed embedded systems. New DTs can be built upon the current ones by extending the guiding principles of modeling, analysis, and synthesis to complex engineering systems. Modularity and abstraction will play a key role in supporting the concurrent design (codesign) of various facets of SoCs and LoCs and in leveraging the expertise of designers/operators with different backgrounds. The problems that stem from these DTs provide a challenging playground in research, as well as a commercial opportunity.

It is easy to conjecture that the social and commercial value of distributed embedded systems will mainly come from the services that they can provide to the users, and the revenue of these services will out shadow the one coming from hardware components, unless in special niche markets. Nevertheless, the component design and the related DT are necessary steps to provide humanity with growth and progress in global information systems. The strategic importance of this area should not be underestimated and, on the contrary, the enabling capabilities of DT should be highly rewarded.

References

[1] [Online]. Available: http://www.affymetrix.com/index.affx

[2] J. Allen, *Natural Language Understanding*. Redwood City, CA: Benjamin Cummings, 1995.

[3] F. A. Aloul, A. Ramani, I. L. Markov, and K. A. Sakallah, "Solving difficult instances of Boolean satisfiability in the presence of symmetry," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 22, no. 9, pp. 1117–1137, Sep. 2003.

[4] E. R. Alvarez-Buylla, A. Chaos, M. Aldana, M. Benitez, Y. Cortes-Poza, C. Espinosa-Soto, D. A. Hartasánchez, R. B. Lotto, D. Malkin, G. J. Escalera Santos, and P. Padilla-Longoria, "Floral morphogenesis: Stochastic explorations of a gene network epigenetic landscape," *PLoS ONE*, vol. 3, no. 11, p. e3626, 2008.

[5] F. Angiolini, P. Meloni, S. Carta, L. Benini, and L. Raffo, "Contrasting a NoC and a traditional interconnect fabric with layout awareness," in *Des., Autom. Test Eur. Conf.*, Mar. 6–10, 2006, pp. 124–129.

[6] D. Atienza, F. Angiolini, S. Murali, A. Pullini, L. Benini, and G. De Micheli, "Network-on-chip design and synthesis outlook," *Integration—VLSI J.*, vol. 41, no. 3, pp. 340–359, May 2008.

[7] D. Bake, G. Church, J. Collins, D. Endy, J. Jacobson, J. Keasling, P. Modrich, C. Smolke, and R. Weiss, "Engineering life: Building a fab for biology," *Sci. Amer.*, vol. 294, no. 6, pp. 44–51, Jun. 2006.

[8] R. Beckman, E. Johnston-Halperin, Y. Luo, J. Green, and J. Heath, "Bridging dimensions: Demultiplexing ultrahigh-density nanowire circuits," *Science*, vol. 310, no. 5747, pp. 465–468, Oct. 21, 2005.

[9] L. Benini, A. Bogliolo, G. Paleologo, and G. De Micheli, "Policy optimization for dynamic power management," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 18, no. 6, pp. 813–833, Jun. 1999.

[10] L. Benini, A. Bogliolo, and G. De Micheli, "A survey of design techniques for system-level dynamic power management," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 8, no. 3, pp. 299–316, Jun. 2000.

[11] L. Benini and G. De Micheli, "Networks on chip; A new design paradigm," *Computer*, vol. 35, no. 1, pp. 70–78, Jan. 2002.

[12] H. Ben Jamaa, K. Moselund, D. Atienza, D. Bouvet, A. Ionescu, Y. Leblebici, and G. De Micheli, "Fault-tolerant multi-level logic decoder for nanoscale crossbar memory arrays," in *ICCAD*, 2007, pp. 765–772.

[13] C. Berggren, P. Stalhandske, J. Brundell, and G. Johansson, "A feasibility study of a capacitive biosensor for direct detection of DNA hybridization," *Electroanalysis*, vol. 11, no. 3, p. 11, Mar. 1999.

[14] D. Bertozzi, A. Jalabert, S. Murali, R. Tamhankar, S. Stergiou, L. Benini, and G. De Micheli, "NoC synthesis flow for customized domain specific multiprocessor systems-on-chip," *IEEE Trans. Parallel Distrib. Syst.*, vol. 16, no. 2, pp. 113–129, Feb. 2005.

[15] D. Blaauw, K. Chopra, A. Srivastava, and L. Scheffer, "Statistical timing analysis: From basic principles to state of the art," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 27, no. 4, pp. 589–607, Apr. 2008.

[16] S. Bobba, J. Zhang, A. Pullini, D. Atienza, S. Mitra, and G. De Micheli, "Design of compact imperfection-immune CNFET layouts for standard-cell-based logic synthesis," in *DATE*, 2009, pp. 616–621.

[17] R. Bryant, "Graph-based algorithms for Boolean function manipulation," *IEEE Trans. Comput.*, vol. C-35, no. 8, pp. 677–691, Aug. 1986.

[18] [Online]. Available: http://bwrc.eecs.berkeley.edu/

[19] S. Carrara, V. Bhalla, C. Stagni, L. Benini, A. Ferretti, F. Valle, A. Gallotta, B. Riccò, and B. Samorì, "New insights for using self-assembly materials to improve the detection stability in label-free DNA-chip and immunosensors," in *Biosens. Bioelectron.*, 2008, to be published.

[20] G. Cerofolini, "Realistic limits to computation. II. The technological side," *Appl. Phys. A, Mater. Sci. Process.*, vol. 86, no. 1, pp. 31–42, Jan. 2007.

[21] A. Chandrakasan, R. Amirtharajah, S. Cho, J. Goodman, G. Konduri, J. Kulik, W. Rabiner, and A. Wang, "Design considerations for distributed microsensor systems," in *Proc. Custom Integr. Circuit Conf.*, 1999, pp. 279–286.

[22] F. Catthoor, E. de Greef, and S. Suytack, *Custom Memory Management Methodology*. Norwell, MA: Kluwer, 1998.

[23] Y. Cheng and G. Church, "Biclustering of expression data," in *Proc. ISMB*, 2000, pp. 93–103.

[24] H. Cho, G. Hachtel, E. Macii, B. Plessier, and F. Somenzi, "Algorithms for approximate FSM traversal based on space decomposition," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 15, no. 12, pp. 1465–1478, Dec. 1996.

[25] M. Cho and D. Pan, "A high-performance droplet routing algorithm for digital microfluidic biochips," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 27, no. 10, pp. 1714–1724, Oct. 2008.

[26] G. F. Close and H.-S. P. Wong, "Fabrication and characterization of carbon nanotube interconnects," in *IEDM Tech. Dig.*, Washington, DC, Dec. 10–12, 2007, pp. 203–206.

[27] J. Cong, J. Wei, and Y. Zhang, "A thermal-driven floorplanning algorithm for 3D ICs," in *Proc. ICCAD*, 2004, pp. 306–313.

[28] R. V. Cox, C. A. Kamm, L. R. Rabiner, J. Schroeter, and J. G. Wilpon, "Speech and language processing for next-millennium communications services," *Proc. IEEE*, vol. 88, no. 8, pp. 1314–1337, Aug. 2000.

[29] M. Dall'Osso, G. Biccari, L. Giovannini, D. Bertozzi, and L. Benini, "Xpipes: A latency insensitive parameterized network-on-chip architecture for multi-processor SoCs," in *Int. Conf. Comput. Des.*, 2003, pp. 536–539.

[30] W. Dally and B. Towles, "Route packets, not wires: On-chip interconnection networks," in *Proc. DAC*, 2001, pp. 684–689.

[31] A. DeHon, "Array-based architecture for FET-based nanoscale electronics," *IEEE Trans. Nanotechnol.*, vol. 2, no. 1, pp. 23–32, Mar. 2003.

[32] A. DeHon, P. Lincoln, and J. E. Savage, "Stochastic assembly of sublithographic nanoscale interfaces," *IEEE Trans. Nanotechnol.*, vol. 2, no. 3, pp. 165–174, Sep. 2003.

[33] G. De Micheli and L. Benini, *Networks on Chips*. San Mateo, CA: Morgan Kaufmann, 2006.

[34] G. De Micheli, *Synthesis and Optimization of Digital Circuits*. New York: McGraw-Hill, 2004.

[35] N. Demierre, "Continuous-flow separation of cells in a lab-on-a-chip using liquid electrodes and multiple-frequency dielectrophoresis," Ph.D. dissertation, EPFL, Lausanne, Switzerland, 2008.

[36] J. De Risi, L. Penland, P. Brown, M. Bittner, P. Meltler, M. Ray, Y. Chen, Y. Su, and M. Trent, "Use of a cDNA microarray to analyze gene expression patterns in human cancer," *Nat. Genet.*, vol. 14, no. 4, pp. 457–460, Dec. 1996.

[37] J. Ding, K. Chakrabarty, and R. Fair, "Scheduling of microfluidic operations for reconfigurable two-dimensional electrowetting arrays," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 20, no. 12, pp. 1463–1468, Dec. 2001.

[38] S. Ecoffey, M. Mazza, V. Pott, D. Bouvet, A. Schmid, Y. Leblebici, M. J. Declercq, and A. M. Ionescu, "A new logic family based on hybrid MOSFET-polysilicon nanowires," in *IEDM Tech. Dig.*, Washington, DC, Dec. 2005, pp. 269–272.

[39] S. Ecoffey, V. Pott, D. Bouvet, M. Mazza, S. Mahapatra, A. Schmid, Y. Leblebici, M. J. Declercq, and A. M. Ionescu, "Nano-wires for room temperature operated hybrid CMOS-NANO integrated circuits," in *IEEE Int. Solid-State Circuits Conf.*, Feb. 6–10, 2005, pp. 260–262.

[40] M. Eisen, P. Spellman, P. Brown, and D. Bostein, "Cluster analysis and display of genome-wide expression patterns," *Proc. Nat. Acad. Sci.*, vol. 95, no. 25, pp. 14 863–14 868, Dec. 1998.

[41] D. Ernst, N. Kim, S. Das, S. Pant, R. Rao, P. Toam, C. Ziesler, D. Blaauw, T. Austin, and T. Mudge, "Razor: A low-power pipeline based on circuit-level timing speculation," in *Proc. MICRO-36*, 2003, pp. 7–18.

[42] C. Espinosa-Soto, P. Padilla-Longoria, and E. Alvarez-Buyilla, "A gene regulatory network model for cell fate determination during Arabidopsis thaliana flower development that is robust and recovers experimental gene expression profiles," *Plant. Cell*, vol. 16, no. 11, pp. 2923–2939, Nov. 2004.

[43] N. Friedman, M. Linial, I. Nachman, and D. Pe'er, "Using Bayesian networks to analyze expression data," *J. Comput. Biol.*, vol. 7, no. 3/4, pp. 601–620, Aug. 2000.

[44] W. W. Fang, N. Singh, L. K. Bera, H. S. Nguyen, S. C. Rustagi, G. Q. Lo, N. Balasubramanian, and D.-L. Kwong, "Vertically stacked SiGe nanowire array channel CMOS transistors," *IEEE Electron Device Lett.*, vol. 28, no. 3, pp. 211–213, Mar. 2007.

[45] Y. Gao, L. Gu, B. Hou, R. Sarikaya, M. Afify, H.-K. Kuo, W.-Z. Zhu, Y. Deng, C. Prosser, W. Zhang, and L. Besacier, "IBM MASTOR SYSTEM: Multilingual automatic speech-to-speech translator," in *Proc. Workshop Medical Speech Translation HLT-NAACL*, 2006, pp. 57–60.

[46] A. Garg, I. Xenarios, L. Mendoza, and G. De Micheli, "An efficient method for dynamic analysis of gene regulatory networks and in silico gene perturbation experiments," in *Proc. RECOMB*, 2007, pp. 62–67.

[47] A. Garg, L. Mendoza, I. Xenarios, and G. De Micheli, "Modeling of multiple valued gene regulatory networks," in *Proc. EMBC*, 2007, pp. 1398–1403.

[48] A. Garg, D. Banerjee, and G. De Micheli, "Implicit methods for probabilistic modeling of gene regulatory networks," in *Proc. EMBC*, 2008, pp. 1398–1404.

[49] A. Garg, A. DiCara, I. Xenarios, L. Mendoza, and G. De Micheli, "Synchronous vs. asynchronous modeling of gene regulatory networks," *Bioinformatics*, vol. 24, no. 17, pp. 1917–1925, Sep. 2008.

[50] A. Garg, Personal Communication.

[51] M. Garey and D. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. San Francisco, CA: Freeman, 1979.

[52] E. Goldberg and Y. Novikov, "BerkMin: A fast and robust SAT solver," *Discrete Appl. Math.*, vol. 155, no. 12, pp. 1549–1561, Jun. 2007.

[53] K. Goossens, J. Dielissens, and A. Radulescu, "AEthrereal network on chip: Concepts, architectures and implementations," *IEEE Des. Test Comput.*, vol. 22, no. 5, pp. 414–421, Sep./Oct. 2001.

[54] P. Guerrier and A. Greiner, "A generic architecture for on-chip packet-switched interconnections," in *Des. Autom. Test Eur. Conf.*, 2000, pp. 250–256.

[55] C. Guiducci, C. Stagni, G. Zuccheri, A. Bogliolo, L. Benini, A. Samorì, and B. Riccò, "DNA detection by integrable electronics," *Biosens. Bioelectron.*, vol. 19, no. 8, pp. 781–787, Mar. 2004.

[56] T. Hogg, Y. Chen, and P. J. Kuekes, "Assembling nanoscale circuits with randomized connections," *IEEE Trans. Nanotechnol.*, vol. 5, no. 2, pp. 110–122, Mar. 2006.

[57] [Online]. Available: http://hpwren.ucsd.edu/

[58] [Online]. Available: http://www.zurich.ibm.com/news/08/3D_cooling.html

[59] P. Ienne, P. Thiran, G. De Micheli, and F. Worm, "An adaptive low-power transmission scheme for on-chip networks," in *Proc. 15th Int. Symp. Syst. Synthesis*, 2002, pp. 92–100.

[60] [Online]. Available: http://www.zurich.ibm.com/news/08/3D_cooling.html

[61] A. Jantsch and H. Tenhunen, *Networks on Chip*. Norwell, MA: Kluwer, 2003.

[62] H.-J. Kang and I.-C. Park, "SAT-based unbounded symbolic model checking," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 24, no. 2, pp. 129–140, Feb. 2005.

[63] S. A. Kauffman, "Metabolic stability and epigenesis in randomly constructed genetic nets," *J. Theor. Biol.*, vol. 22, no. 3, pp. 437–467, Mar. 1969.

[64] [Online]. Available: http://www.genome.jp/kegg/

[65] K. Keutzer, R. Newton, J. Rabaey, and A. Sangiovanni-Vincentelli, "System-level design: Orthogonalization of concerns and platform-based design," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 19, no. 12, pp. 1523–1543, Dec. 2000.

[66] L. N. Chakrapani, P. Korkmaz, B. E. S. Akgul, and K. V. Palem, "Probabilistic system on chip architectures," *ACM Trans. Des. Autom. Electron. Syst.*, vol. 12, no. 3, Aug. 2007.

[67] S.-J. Lee, S.-J. Song, K. Lee, J.-H. Woo, S.-E. Kim, B.-G. Nam, and H.-J. Yoo, "An 800 MHz star-connected on-chip network for application to systems on a chip," in *IEEE Int. Solid-State Circuits Conf.*, Feb. 2003, pp. 468–469.
[68] E. Lewis, *Introduction to Reliability Engineering*. Hoboken, NJ: Wiley, 1996.
[69] U. Lehmann, M. Sergio, S. Pietrocola, C. Niclass, E. Charbon, and M. A. M. Gijs, "Microparticle photometry in a CMOS microsystem combining magnetic actuation and in situ optical detection," *Sens. Actuators B, Chem.*, vol. 132, no. 2, pp. 411–417, Jun. 16, 2008.
[70] U. Lehmann, "Manipulation of magnetic microparticles in liquid phases for on-chip biomedical analysis methods," Ph.D. dissertation, EPFL, Lausanne, Switzerland, 2008.
[71] K. K. Likharev and D. B. Strukov, *Introducing Molecular Electronics*. New York: Springer-Verlag, 2004.
[72] S. Lim, "Physical design for 3D systems on package," *IEEE Des. Test Comput.*, vol. 22, no. 6, pp. 532–539, Nov./Dec. 2005.
[73] Y. Luoer, P. C. Collier, J. O. Jeppesen, K. A. Nielsen, E. Delonno, G. Ho, J. Perkins, H. R. Tseng, T. Yamamoto, J. F. Stoddart, and J. R. Heath, "Two-dimensional molecular electronic circuits," *ChemPhysChem*, vol. 3, no. 6, pp. 519–525, Jun. 2002.
[74] J. Macdonald, D. Stefanovic, and M. Stojanovic, "DNA computing for work and play," *Sci. Amer.*, vol. 299, no. 5, pp. 60–67, Nov. 2008.
[75] S. Madeira and A. Oliveira, "Biclustering algorithms for biological data analysis: A survey," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 1, no. 1, pp. 24–45, Jan.–Mar. 2004.
[76] A. Manz, N. Graber, and H. M. Widmer, "Miniaturized total chemical analysis systems: A novel concept for chemical sensing," *Sens. Actuators B, Chem.*, vol. 1, no. 1–6, pp. 244–248, Jan. 1990.
[77] R. Marculescu, U. Ogras, K. Peh, N. Jerger, and Y. Hoskote, "Outstanding research problems in NoC design: System, microarchitecture, and circuit perspectives," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 28, no. 1, pp. 3–21, Jan. 2009.
[78] D. Maslov, S. M. Falconer, and M. Mosca, "Quantum circuit placement," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 27, no. 4, pp. 752–763, Apr. 2008.
[79] J. D. Meindl, "Interconnect opportunities for gigascale integration," *IEEE Micro*, vol. 23, no. 3, pp. 28–35, May/Jun. 2003.
[80] L. Mendoza and I. Xenarios, "A method for the generation of standardized qualitative dynamical systems of regulatory networks," *Theor. Biol. Med. Model.*, vol. 3, p. 13, 2006.
[81] [Online]. Available: http://www.mics.org/
[82] C. Mihic, T. Simunic, and G. De Micheli, "Power and reliability management of SoCs," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 15, no. 4, pp. 391–403, Apr. 2007.
[83] R. Min, M. Bhardwaj, S.-H. Cho, N. Ickes, E. Shih, A. Sinha, A. Wang, and A. Chandrakasan, "Energy-centric enabling technologies for wireless sensor networks," *Wireless Commun.*, vol. 9, no. 4, pp. 28–39, Aug. 2002.
[84] S. Minato, *Binary Decision Diagrams for VLSI/CAD*. Norwell, MA: Kluwer, 1996.
[85] V. Mirsky, M. Riepl, and O. Wolfbeis, "Capacitive monitoring of protein immobilization and antigen-antibody reactions on monomolecular alkylthiol films on gold electrodes," *Biosens. Biolectron.*, vol. 12, no. 9/10, pp. 977–989, 1997.
[86] F. Mo and R. Brayton, "Whirlpool PLAs: A regular logic structure and their synthesis," in *Proc. ICCAD*, 2002, pp. 543–550.
[87] K. E. Moselund, V. Pott, D. Bouvet, and A. M. Ionescu, "Hysteretic inverter-on-a-body-tied-wire based on less-than-10 mV/decade abrupt punch-through impact ionization MOS PIMOS switch," in *Proc. Int. Symp. VLSI-TSA*, Hsinchu, Taiwan, Apr. 21–23, 2008.
[88] M. Moskewick, C. Madigan, Y. Zhao, and S. Malik, "CHAFF: Engineering an efficient SAT solver," in *Proc. Des. Autom. Conf.*, 2001, pp. 530–535.
[89] F. Mulas, M. Buttu, M. Pittau, S. Carta, D. Atienza, A. Acquaviva, L. Benini, and G. De Micheli, "Thermal balancing policy for streaming computing on multiprocessor architectures," in *Proc. DATE*, 2008, pp. 734–739.
[90] S. Murali, P. Meloni, D. Atienza, S. Carta, L. Benini, G. De Micheli, and L. Raffo, "Synthesis of predictable networks-on-chip based interconnect architectures for chip multi-processors," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 15, no. 8, pp. 869–880, Aug. 2007.
[91] K. M. Murphy and S. L. Reiner, "The lineage decisions on helper T cells," *Nat. Rev., Immunol.*, vol. 2, no. 12, pp. 933–944, Dec. 2002.
[92] N. Gi-Joon, K. A. Sakallah, and R. A. Rutenbar, "A new FPGA detailed routing approach via search-based Boolean satisfiability," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 21, no. 6, pp. 674–684, Jun. 2002.
[93] C. Nardini, L. Benini, and G. De Micheli, "Circuits and systems for high-throughput biology," *IEEE Circuits Syst. Mag.*, vol. 6, no. 3, pp. 10–20, 2006.
[94] [Online]. Available: www.nano-tera.ch
[95] [Online]. Available: http://www.ocpip.org/home
[96] J. Paradiso and T. Starner, "Energy scavenging for mobile and wireless electronics," *Pervasive Comput.*, vol. 4, no. 1, pp. 18–27, Jan.–Mar. 2005.
[97] N. Patil, D. Jie, H.-S. P. Wong, and S. Mitra, "Automated design of misaligned-carbon-nanotube-immune circuits," in *Des. Autom. Conf.*, Jun. 2007, pp. 958–961.
[98] A. Pease, D. Solas, E. J. Sullivan, M. T. Cronin, C. P. Holmes, and S. P. Fodor, "Light-generated oligonucleotide arrays for rapid DNA sequencing analysis," *Proc. Nat. Acad. Sci.*, vol. 91, no. 11, pp. 5022–5026, May 1994.
[99] M. Pedram and J. Rabaey, *Power Aware Design Methodologies*. New York: Springer-Verlag, 2002.
[100] A. Peymandoust, T. Simunic, and G. De Micheli, "Complex instruction and software library mapping for embedded software using symbolic algebra," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 22, no. 8, pp. 964–975, Aug. 2003.
[101] L. Pileggi, H. Schmit, A. J. Strojwas, P. Gopalakrishnan, V. Kheterpal, A. Koorapaty, C. Patel, V. Rovner, and K. Y. Tong, "Exploring regular fabrics to optimize the performance-cost trade-off," in *Des. Autom. Conf.*, 2003, pp. 782–787.
[102] A. Pullini, F. Angiolini, S. Murali, D. Atienza, G. De Micheli, and L. Benini, "Bringing NoCs to 65 nm," *IEEE Micro*, vol. 27, no. 5, pp. 75–85, Sep./Oct. 2007.
[103] J. Quackenbush, "Computational analysis of microrarray data," *Nat. Rev. Genet.*, vol. 2, no. 6, pp. 418–427, Jun. 2001.
[104] Q. Qiu and M. Pedram, "Dynamic power management based on continuos-time Markov decision processes," in *Des. Autom. Conf.*, 1999, pp. 555–561.
[105] E. Rachlin and J. Savage, "Nanowire addressing with randomized-contact decoders," in *Proc. ICCAD*, 2006, pp. 735–742.
[106] C. V. Rao, D. M. Wolf, and A. P. Arkin, "Control, exploitation and tolerance of intracellular noise," *Nature*, vol. 420, no. 6912, pp. 231–237, Nov. 2002.
[107] R. Rudell and A. Sangiovanni-Vincetelli, "Multiple-valued minimization for PLA optimization," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. CAD-6, no. 5, pp. 727–750, Sep. 1987.
[108] A. Sangiovanni-Vincentelli, "The tides of EDA," *IEEE Des. Test Comput.*, vol. 20, no. 6, pp. 59–75, Nov./Dec. 2003.
[109] J. E. Savage, E. Rachlin, A. DeHon, C. M. Lieber, and Y. Wu, "Radial addressing of nanowires," *ACM J. Emerg. Technol. Comput. Syst.*, vol. 2, no. 2, pp. 129–154, Apr. 2006.
[110] A. Schliep, D. C. Torney, and S. Rahmann, "Group testing with DNA chips: Generating designs and decoding experiments," in *Proc. IEEE CSB*, 2003, pp. 84–91.
[111] U. Seger, S. Gawad, R. Johann, A. Bertsch, and P. Renaud, "Cell immersion and cell dipping in microfluidic devices," *Lab Chip*, vol. 4, no. 2, pp. 148–151, Apr. 2004.
[112] C. Seiculescu, S. Murali, L. Benini, and G. De Micheli, "SunFloor 3D: A tool for networks on chip topology synthesis for 3D systems on chip," in *Proc. DATE*, 2009, pp. 9–14.
[113] D. Singh, J. Rabaey, M. Pedram, F. Catthoor, S. Rajgopal, N. Seghal, and T. Mozdzen, "Power conscious CAD tools and methodologies: A perspective," *Proc. IEEE*, vol. 83, no. 4, pp. 570–594, Apr. 1995.
[114] [Online]. Available: http://www.csem.ch/sfit/
[115] L. Schaper, S. Spiesshoefer, G. Vangara, Z. Rahman, and S. Polamreddy, "Architectural implications and process development of 3-D VLSI Z -Axis interconnects using through silicon vias," *IEEE Trans. Adv. Packag.*, vol. 28, no. 3, pp. 356–366, Aug. 2005.
[116] V. V. Shende, A. K. Prasad, I. L. Markov, and J. P. Hayes, "Synthesis of reversible logic circuits," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 22, no. 6, pp. 710–722, Jun. 2003.
[117] M. Schienle, C. Paulus, A. Frey, F. Hoffmann, B. Holzapfl, P. Schindelr-Bauer, and R. Thewes, "A fully electronic DNA sensor with 128 positions and in-pixel A/D conversion," *IEEE J. Solid-State Circuits*, vol. 39, no. 12, pp. 2438–2445, Dec. 2004.
[118] I. Shmulevich, E. R. Dougherty, S. Kim, and W. Zhang, "Probabilistic Boolean networks: A rule-based uncertainty model for gene regulatory network," *Bioinformatics*, vol. 18, no. 2, pp. 261–274, Feb. 2002.

[119] J. Silva and K. Sakallah, "GRASP: A search algorithm for propositional satisfiability," *IEEE Trans. Comput.*, vol. 48, no. 5, pp. 506–521, May 1999.
[120] T. Simunic, L. Benini, P. Glynn, and G. De Micheli, "Event-driven power management," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 20, no. 7, pp. 840–857, Jul. 2001.
[121] J. Srinivasan, S. V. Adve, P. Bose, J. Rivers, and C. K. Hu, *RAMP: A Model for Reliability Aware Microprocessor Design*. Poughkeepsie, NY: IBM, 2003.
[122] C. Stagni, D. Esposti, C. Guiducci, C. Paulus, M. Schienle, Maugustyniak, G. Zuccheri, B. Samori, L. Benini, B. Ricco, and R. Thewes, "Fully electronic CMOS DNA detection array based on capacitance measurement with on-chip analog-to-digital conversion," in *Proc. ISSC*, San Francisco, CA, 2006, pp. 69–78.
[123] P. Stephan, R. K. Brayton, and A. L. Sangiovanni-Vincentelli, "Combinational test generation using satisfiability," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 15, no. 9, pp. 1167–1176, Sep. 1996.
[124] A. Subramanian, P. Tamayo, V. K. Mootha, S. Mukherjee, B. L. Ebert, M. A. Gillette, A. Paulovich, S. L. Pomeroy, T. R. Golub, E. S. Lander, and J. P. Mesirov, "Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles," *Proc. Nat. Acad. Sci. USA*, vol. 102, no. 43, pp. 15 545–15 550, Oct. 2005.
[125] D. Tang and R. Iyer, "Dependability measurement and modeling of a multicomputer system," *IEEE Trans. Comput.*, vol. 42, no. 1, pp. 62–75, Jan. 1993.
[126] R. Thomas, "Regulatory networks seen as asynchronous automata: A logical description," *J. Theor. Biol.*, vol. 153, pp. 1–23, 1991.
[127] [Online]. Available: www.tilera.com
[128] R. Treuer, H. Fujiwara, and V. Agarwal, "Implementing a built-in self-test PLA design," *IEEE Des. Test Comput.*, vol. 2, no. 2, pp. 37–48, Apr. 1985.
[129] A. Tüdos, G. Besselink, and Schasfoor, "Trends in miniaturized total analysis systems for point-of-care testing in clinical chemistry," *Lab Chip*, vol. 1, no. 2, pp. 83–95, Dec. 2001.
[130] V. Tusher, R. Tibsharani, and G. Chu, "Significance analysis of microarrays applied to the ionizing radiation response," *Proc. Nat. Acad. Sci. USA*, vol. 98, no. 9, pp. 5116–5121, Apr. 2001.
[131] S. Vangal, J. Howard, G. Ruhl, S. Dighe, H. Wilson, J. Tschanz, D. Finan, P. Iyer, A. Singh, T. Jacob, S. Jain, S. Venkataraman, Y. Hoskote, and N. Borkar, "An 80-tile 1.28TFLOPS network-on-chip in 65 nm CMOS," in *Proc. Int. Solid-State Circuits Conf.*, 2007, pp. 98–99.
[132] S. R. Vangal, J. Howard, G. Ruhl, S. Dighe, H. Wilson, J. Tschanz, D. Finan, A. Singh, T. Jacob, S. Jain, V. Erraguntla, C. Roberts, Y. Hoskote, N. Borkar, and S. Borkar, "An 80-tile sub-100-W TeraFLOPS processor in 65-nm CMOS," *IEEE J. Solid-State Circuits*, vol. 43, no. 1, pp. 29–41, Jan. 2008.
[133] V. Vankamamidi, M. Ottavi, and F. Lombardi, "Two-dimensional schemes for clocking/timing of QCA circuits," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 27, no. 1, pp. 34–44, Jan. 2008.
[134] H. Wang, W. Wang, J. Yang, and P. Wu, "Clustering by pattern similarity in large data sets," in *Proc. ACM Conf. Manage. Data*, 2002, pp. 394–405.
[135] [Online]. Available: http://www.cs.waseda.ac.jp/gcoe/eng/members/
[136] W. Wolf, "The future of microprocessor systems on chips," in *Proc. DAC*, 2004, pp. 681–685.
[137] W. Wolf, *High Performance Embedded Computing: Architectures, Applications and Methodologies*. San Mateo, CA: Morgan Kaufmann, 2006.
[138] S. Yoon, C. Nardini, L. Benini, and G. De Micheli, "Discovering coherent biclusters from gene expression data using zero-suppressed binary decision diagrams," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 2, no. 4, pp. 339–354, Oct.–Dec. 2005.
[139] S. Yoon, L. Benini, and G. De Micheli, "Finding co-clusters of genes and clinical parameters," in *Proc. EMBC*, 2005, pp. 906–912. No. 11.3.1.4.
[140] S. Yoon, L. Benini, and G. De Micheli, "A pattern mining method for high-throughput lab-on-chip data analysis," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 25, no. 2, pp. 358–377, Feb. 2006.
[141] P.-H. Yuh, C.-L. Yang, and Y.-W. Chang, "BioRoute: A network-flow-based routing algorithm for the synthesis of digital microfluidic biochips," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 27, no. 11, pp. 1928–1941, Nov. 2008.
[142] T. Zhang, K. Chakrabarty, and R. Fair, "Integrated hierarchical design of microelectrofluidic systems using SystemC," *Microelectron. J.*, vol. 33, no. 5/6, pp. 459–470, May 2002.
[143] T. Zhang, K. Chakrabarty, and R. Fair, "Design of reconfigurable composite microsystems based on hardware/software codesign principles," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 31, no. 8, pp. 987–995, Aug. 2002.
[144] P. Zhou, Y. Ma, Z. Li, R. Dick, L. Shang, H. Zhou, X. Hong, and Q. Zhou, "3D-STAF: Scalable temperature and leakage aware floorplanning for 3-dimensional integrated circuits," in *Proc. ICCAD*, 2007, pp. 590–597.

Giovanni De Micheli (S'79–M'83–SM'89–F'94) is a Professor and the Director of the Institute of Electrical Engineering and of the Integrated Systems Centre, Ecole Polytecnique Federale de Lausanne (EPFL), Lausanne, Switzerland. He also chairs the Scientific Committee of Centre Suisse d'Electronique et de Microtechnique, Neuchâtel, Switzerland. His research interests include several aspects of design technologies for integrated circuits and systems (CAS), such as synthesis, hw/sw codesign, and low-power design, as well as systems on heterogeneous platforms including electrical, micromechanical, and biological components. He is author of: *Synthesis and Optimization of Digital Circuits* (McGraw-Hill, 1994), coauthor and/or coeditor of six other books and of over 400 technical articles.

Prof. De Micheli was the recipient of the 2003 IEEE Emanuel Piore Award for contributions to computer-aided synthesis of digital systems. He is a Fellow of Association for Computing Machinery. He received the Golden Jubilee Medal for outstanding contributions to the IEEE CAS Society in 2000. He received the 1987 D. Pederson Award for the best paper on the IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS (CAD/ICAS), two Best Paper Awards at the Design Automation Conference (DAC) in 1983 and 1993, and a Best Paper Award at the DATE Conference in 2005. He has been serving IEEE in several capacities, namely: Division 1 Director (2008–2009), Cofounder and President Elect of the IEEE Council on Electronic Design Automation (2005–2007), President of the IEEE CAS Society (2003), and Editor in Chief of the IEEE TRANSACTIONS ON CAD/ICAS (1987–2001). He was Program Chair of the pHealth and Very Large Scale Integration System-on-Chip conferences in 2006. He was the Program and General Chair of the DAC in 1996–1997 and 2000, respectively. He is a founding member of the Advanced Learning and Research Institute, Universita' della Svizzera Italiana, Lugano, Switzerland, where he is currently a Scientific Counselor.