17 Architectures and Simulations for Nanoprocessor Systems Integrated on the Molecular Scale

Shamik Das[1], Garrett S. Rose[1], Matthew M. Ziegler[2], Carl A. Picconatto[1], and James C. Ellenbogen[1]

[1] The MITRE Corporation, 7525 Colshire Drive, McLean, VA 22102
{sdas,grose,picconatto,ellenbgn}@mitre.org
[2] IBM T. J. Watson Research Center, 1101 Kitchawan Road, P. O. Box 218, Yorktown Heights
zieglerm@us.ibm.com

Abstract. This chapter concerns the design, development, and simulation of nanoprocessor systems integrated on the molecular scale. It surveys ongoing research and development on nanoprocessor architectures and discusses challenges in the implementation of such systems. System simulation is used to identify some advantages, issues, and trade-offs in potential implementations. Previously, the authors and their collaborators considered in detail the requirements and likely performance of nanomemory systems. This chapter recapitulates the essential aspects of that earlier work and builds upon those efforts to examine the likely architectures and requirements of nanoprocessors. For nanoprocessor systems, simulation, as well as design and fabrication, embodies unique problems beyond those introduced by the large number of densely-packed, novel nanodevices. For example, unlike the largely homogeneous structure of circuitry in nanomemory arrays, a high degree of variety and inhomogeneity must be present in nanoprocessors. Also, issues of clocking, signal restoration, and power become much more significant. Thus, building and operating nanoprocessor systems will present significant new challenges and require additional innovations in the application of molecular-scale devices and circuits, beyond those already achieved for nanomemories. New nanoelectronic devices, circuits, and architectures will be necessary to perform the more complex and specialized functions inherent in processing systems at the nanometer scale. This chapter highlights the fundamental design requirements of such nanoprocessor systems, presents various device and design options, and discusses their potential implications for system performance.

17.1 Introduction

The excitement that surrounds the field of molecular electronics is premised, to a great extent, upon the prospect that soon we may be able to design, fabricate, and demonstrate an entire, ultra-dense electronic computer that is integrated on the molecular scale. In fact, development of such nanoelectronic

S. Das et al.: *Architectures and Simulations for Nanoprocessor Systems Integrated on the Molecular Scale*, Lect. Notes Phys. **680**, 479–513 (2005)
www.springerlink.com

computer systems already is underway. Despite significant challenges, this effort is likely to produce functioning prototype nanomemories and nanoprocessors within a few years [1–9].

Such development is essential to fulfill the promise and the expectations that have been raised by the dramatic recent successes in demonstrating electronic devices and simple circuits on the molecular scale, as is discussed in the foregoing chapters of this book [10–13] and elsewhere [3, 14–21]. In this regard, much progress already has been made toward employing molecular-scale devices in building and demonstrating extended nanomemory systems [4, 5, 21]. Pressing on toward much more complex, extended nanosystems, such as nanoprocessors, does not alleviate the need for more research and development on nanodevices. In fact, as we explain below, it places even more stringent demands for understanding, predictability, uniformity, and reliability of performance at the device level.

Nonetheless, the process of developing true nanocomputers does open up an entirely new frontier of systems objectives and issues that require research and development beyond that which presently is being conducted upon isolated nanodevices. It is this new frontier that is the primary topic of the present chapter, which addresses the problem of how to design and simulate an entire nanoprocessor system that is integrated on the molecular scale.

This survey and analysis of nanoprocessor system architectures may be considered as a companion to a recent paper on nanomemories by several of the present authors. In that work, we simulated and analyzed a nanowire-based nanomemory array as a vehicle for considering a range of issues that arise in the development of nanomemory systems [22]. The much more difficult challenges presented by the frontier problem of nanoprocessor design and development are considered here in several ways and at several levels of resolution, as follows:

- In Sect. 17.2 of this work, we describe how molecular-scale devices fit in and are harnessed within an extended nanowire-based circuit system.
- Section 17.3 of this work provides a general overview of the nanoprocessor design and architecture problem from a system-level perspective. This review of the issues that must be faced includes a consideration of the problems of migrating conventional microelectronic architectures to the nanoscale, as well as an analysis of the difficulties that might arise with novel nanoelectronic architectures.
- Section 17.4 contains a brief survey of the various system architecture approaches that have been proposed.
- Then, in Sect. 17.5, there is detailed consideration of one promising architectural design approach, due to DeHon and Wilson [23–25]. This approach utilizes imprinted or self-assembled nanowires for both the devices and the interconnect structures. It draws from and builds upon well-tested ideas for constructing programmable logic, such as the programmable

logic
array (PLA) [26], in formulating the overall architecture. The section begins with a brief description of the structure and function of a PLA. This is followed by a brief survey and perspective on the microscale and nanoscale antecedents of this architectural approach [27]. This survey may be valuable in assisting others toward synthesizing still further design approaches that may be better suited to molecular-scale devices other than those built solely from nanowires.

- We continue in Sect. 17.6 with a detailed simulation of key nanowire circuits for such a PLA-based nanoarchitecture. This is intended to illustrate in a very specific manner the types of issues that will be encountered in building and operating a nanoprocessor, well before an entire system of this type actually is fabricated and integrated on the molecular scale.

By integration on the molecular scale, we mean that the basic switching devices, as well as the wire widths and the pitch dimensions (i.e., spacing between the centers of neighboring wires), all will measure only a few nanometers – the size of a small molecule – in the computer systems of interest here. Such systems may function using only one or a few molecules within their basic devices [3,6,16,17,21,28]. On the other hand, the systems may not use molecules at all, employing instead solid-state quantum dots [29–33] and/or imprinted or self-assembled nanowires [34–36].

From a systems perspective, the very small dimensions of any of these device and interconnect structures open up new design possibilities because of the very high density of function the structures can provide. However, at the same time, the repertoire of structures available to the system designer is limited by the present difficulty of performing precise, flexible, and economical fabrication or assembly at these molecular dimensions for the very large number of conductive components that are required. Especially, the basic structures available and the demonstrated performance of the devices are not yet as diverse or robust as computer-system designers have come to take for granted when developing systems that are integrated only at the microscale. Also, the nanometer-scale switches and interconnects are likely to exhibit a higher degree of structural and functional variability than is common in the much more highly evolved technology for building microelectronics. Thus, the nanoprocessor designer must seek a strategy that takes advantage of and can walk the line between these countervailing facts of life on the nanometer scale – very high densities, but less precision, uniformity, and operational robustness.

One example of this compromise is embodied in the nanowire-based PLAs [25] that are discussed in detail in Sects. 17.5 and 17.6 of this work. The fact that the PLA architecture takes advantage of the available density to ameliorate some of the limitations implicit in present-day nanodevice and nanofabrication technologies [24,37] seems to suggest that this architectural approach can be used successfully to develop a functioning nanoprocessor in the reasonably near term. That is, system simulations indicate that it should not be necessary to push back very far the thresholds of present limitations on devices and fabrication in order to make progress on the system challenges.

Further, we illustrate here that as the research community attempts to move forward with detailed designs for an entire nanocomputer, system simulation can illuminate the detailed consequences of both the architecture-level design choices and the a priori constraints. Still further, the results of the simulation serve to provide focus for nanodevice and nanofabrication research, showing where it may be necessary to push back on the limits of these technologies, and where such efforts can have the most benefit for the ultimate objective of building a nanocomputer.

17.2 Starting at the Bottom: Molecular Scale Devices in Device-Driven Architectures for Nanoprocessors

Whether one considers the design, simulation, or fabrication of an entire processor system, there is a hierarchy of structure and function. In the usual approach of modern electrical engineering, this hierarchy is taken to start at the highest level of abstraction, the architecture level. Then it descends down to the level of its component circuits, and finally, proceeds down to the level of the component switch and interconnect devices [38]. To a great extent, this viewpoint mirrors the “top-down” approach used in the design and fabrication of microprocessors, in which the robust performance of the devices and the ability to tune precisely the structure and performance of those devices – i.e., microelectronic transistors – is somewhat taken for granted. Architectures often are optimized to suit first the high-level, system objectives, such as computational latency and throughput, then the circuits, and finally, the behavior of the devices may be adjusted to suit particular needs of the architecture.

At present, the situation is different when one sets out to design, simulate, or fabricate an entire nanoprocessor system integrated on the molecular scale. The ability to tune the performance of nanodevices still is limited. This is partly because these molecular-scale devices are so new. Thus, the experiments [10–13,39] and the theory [40–47] necessary to understand them, design them, and make them to order still are very much in development, as is evident in the foregoing chapters of this volume. In addition, however, the ability to tune precisely the structure and performance of nanometer-scale devices may be limited inherently by the quantization of those properties, which is ubiquitous on that tiny scale.

Further, designs for nanoelectronic circuits and systems are constrained by the very small size and small total currents associated with molecular-scale switches. This is coupled with the difficulty of making contact with them using structures and materials that are large and conductive enough to provide sufficient current and signal strength to serve an entire nanoprocessor

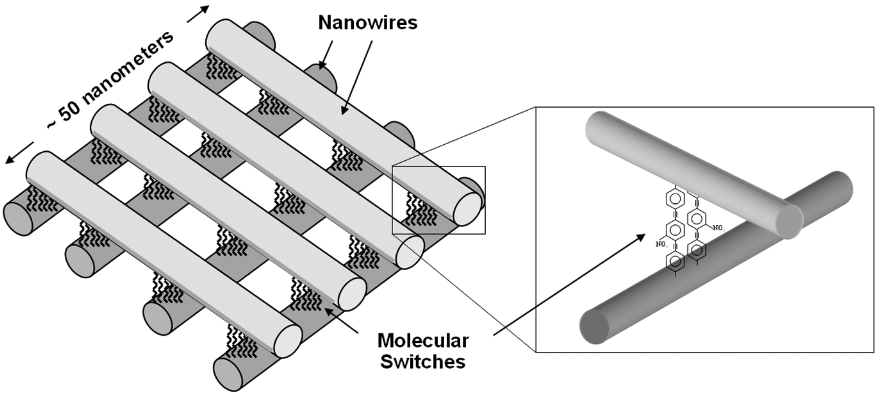


Fig. 17.1. “Crossbar” array of nanowires with molecular devices at junctions

system. This system will be at least tens of square microns, if not tens of square millimeters, in extent, which is millions or trillions of times larger than the molecular-scale devices themselves.

Regardless of whether all these limitations are temporary or fundamental, for now they constrain both the circuits and the architectures that are achievable in the relatively near term. Further, these limitations force us to begin consideration of the design and the simulation of nanoprocessor systems at the bottom-most level of the hierarchy, the device level.

As is true in most experiments on the electrical properties of molecules [10, 11, 15, 48, 49], for the purposes of discussing circuits and systems a molecular-scale device consists of a junction between two metal or semiconductor surfaces, with a molecular-scale structure sandwiched between. This molecular-scale structure may be one or a few molecules, as depicted in Fig. 17.1. Or else, it may be a layer of molecules or atoms only a few nanometers thick, as in the nanowire junction diode depicted in Fig. 2(a). While many electrical properties may be very important (especially capacitance), the electrical behavior of such junction nanoswitches is characterized primarily by the current response I to an applied voltage V, a so-called I-V curve, such as is shown in Fig. 2(b).

I-V behaviors of such junctions include: simple resistance at low voltage [6], rectification [13, 22, 50], negative differential resistance (NDR) [18] and hysteresis [6, 22]. A variety of such junction nanodevices have been realized that might be useful for building extended nanoelectronic systems. The hysteretic behavior illustrated in Fig. 2(b) is particularly valuable, as it allows the “programming” of a junction into one of two states. Such bistable switches are essential components of any computing system.

Development of molecular-scale switches with appropriate I-V behaviors is essential to be able to construct functional circuits that can be used to build up processor systems. It is of particular importance to have nanoscale

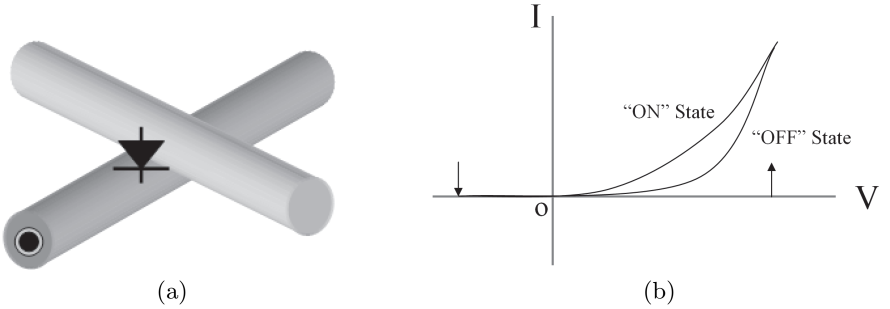


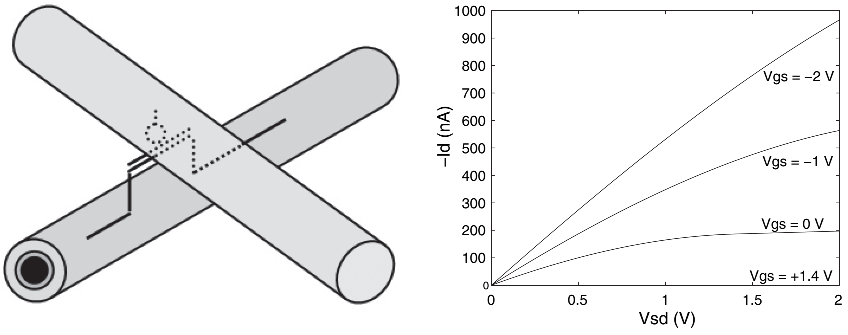
Fig. 17.2. Illustrations of (**a**) a rectifying junction switch made of crossed nanowires that sandwich a molecule or layer of molecules or atoms and (**b**) a representative I-V characteristic for a hysteretic, rectifying device. Hysteresis is indicated by the multiple conductance states. The high-conductance "on" state and low-conductance "off" state are depicted, and the voltage thresholds at which the device switches between states are labeled with arrows. Rectification is indicated by the unequal responses to positive and negative voltages

switches that can be used to produce signal restoration and gain, e.g., nanotransistors.[1] These two features are essential to maintaining electrical signals as they move through multiple levels of logic. Nanotransistors have been fabricated using carbon nanotubes (CNTs) [20, 52–54], although it remains very difficult to use them in building extended systems. There also have been some suggestions for fabricating transistors from smaller molecules [13, 55]. A few individual molecular transistors have been demonstrated based on small molecules, but only in very sensitive experiments under cryogenic conditions [56, 57]. On the other hand, robust nanoscale transistors built from crossed nanowires have been demonstrated in a number of experiments at room temperature [19].[2] A diagramm of such a nanowire nanotransistor is displayed alongside models of its I-V curves in Fig. 17.3. Thus, following the architects DeHon and Wilson [25], in the simulations described here, we employ these nanowire transistors in analyzing systems that might be fabricated and operated under realistic conditions in the near term.

In addition to obtaining gain and signal restoration, simpler I-V behaviors, such as strong rectification from two-terminal nanodevices, also are very important. Simulations show that even when using devices that provide good gain, rectification remains important to ensure that signals do not take unin-

[1] Small circuits, e.g., latches incorporating molecular diodes, also can produce signal restoration [51].

[2] Note that this transistor is not a junction nanoswitch since, ideally, no current flows between the nanowires. Rather, the top nanowire serves as a gate for the bottom "channel" nanowire, and the two are isolated from each other by a dielectric layer. This is in contrast to the nanowire diode shown in Fig. 17.2, which is a junction nanoswitch.



(a) (b)

Fig. 17.3. Illustrations of (**a**) a crossed-nanowire p-channel field effect transistor (PFET) and (**b**) a model of the I-V characteristic for this device. The experimental basis for this model was obtained from [19]. For this transistor, the threshold voltage, at which the device produces essentially zero current and turns "off," is observed to be approximately +1.4 V

tended and undesirable paths through circuits, especially in crossbar arrays. A strong rectifier can fulfill this role by permitting current to pass only in one direction in the circuit at the designed operating voltages.

The molecular-scale electronics community is just beginning to succeed in taking the key steps required for actually building and operating an extended nanoprocessor system that integrates two-terminal junction nanodevices, as well as three-terminal nanotransistors. The steps form a hierarchy from the device to the system level, as follows: (a) development of nanofabrication approaches to build large numbers of these devices with precision and regularity, (b) development of interconnect and circuit design approaches that can incorporate such junction structures into extended circuit systems, and (c) determination of architectural approaches that include the aforementioned circuits designs and that can accommodate the limitations of the I-V behaviors available in present-day molecular electronic devices. Challenges exist at each level of this hierarchy.

17.3 Challenges for Nanoelectronics in Developing Nanoprocessors

17.3.1 Overview

In order to utilize recent advances in molecular-scale devices and circuits to build extended systems, many challenges must be faced at all levels of design and fabrication. Foremost, the structure and ultra-high density of these novel molecular-scale devices make difficult the use of conventional microprocessor architectures. Such difficulties motivate fundamental departures in design. This then necessitates the development of new circuits, interconnection strategies, and fabrication methods, each of which, in turn, presents additional challenges. The following sections discuss some of the challenges posed by the use of conventional architectures, as well as the new difficulties that arise in novel architectures.

17.3.2 Challenges Posed by the Use of Conventional Microprocessor Architectures

The principal challenge of using conventional architectures [58] for the development of nanoprocessor systems is that such architectures have too much heterogeneity and complexity for existing nanofabrication methods. Conventional processor architectures are heterogeneous at every level of the design hierarchy. At the top level, a modern microprocessor consists of logic, cache memory, and an input/output interface. In conventional microscale integration, these three architectural components may be designed using different circuit styles or even different fabrication methods. The logic component itself consists of arithmetic and control subcomponents, both of which require circuits that may be either combinational (e.g., AND, OR, XOR gates) or sequential (i.e., clocked elements such as registers) [58]. Further still, the synthesis of the aforementioned combinational logic gates requires multiple kinds of devices for optimal performance [38]. This differentiation into a wide variety of devices, circuits, and subsystems is an advantageous structural feature provided by present microfabrication. Providing such differentiation is beyond the reach of present nanofabrication techniques. As a result, nanoelectronics research has targeted the development of architectures for nanoprocessors that provide comparable function while avoiding as much as possible the introduction of heterogeneity at the hardware level.

17.3.3 Challenges in the Development of Novel Nanoprocessing Architectures

Most of the nanoprocessor architectures presently proposed [25,32,33,59–72] are essentially homogeneous at the hardware level and introduce diversification at the programming stage. In this way, they are able to do without the complexity of fabrication characteristic of conventional microprocessors.

Many of these nanoprocessor architectures inherit their design characteristics from microscale programmable logic [27], especially field-programmable gate arrays (FPGAs) [73] and PLAs [26]. As described in detail in Sect. 17.5, FPGAs and PLAs are regular arrays of logic gates whose inter-gate wiring can be reconfigured. Software is used to configure FPGAs and PLAs to compute particular logic functions. In contrast, the logic functions in conventional microprocessors are hard-wired during construction. Thus, in FPGAs and PLAs, the use of software to "complete" the hardware construction allows the hardware design to be simplified to a homogeneous form.

Although these physically homogeneous architectures simplify fabrication, they do introduce a new set of challenges. For nanoprocessing, these challenges may be illustrated by considering the example of a nanoscale crossbar switch array. This is a homogeneous approach that combines a high degree of scalability with some of the smallest circuit structures demonstrated to date [3, 21]. The basic crossbar architecture consists of the combination of planes of parallel wires that are laid out in orthogonal directions, such as is shown in Figs. 17.1 and 17.4. Computation and communication rely on molecular-scale junction switches formed at the crosspoints of the wires as the fundamental devices.

These ultra-dense arrays are fabricated using specialized techniques such as nanoimprinting [3, 74] or flow-based alignment [34]. Prototype nanoelectronic circuits and reasonably large memory arrays already have been constructed using these techniques [7, 19, 21, 75, 76]. Moreover, a number of architectural proposals have been put forth that involve the tiling of crossbar subarrays to form programmable fabrics, including the design shown in Fig. 17.4 [7, 23, 25, 66].

Among the reasons that these regular crossbar structures are attractive is because it is possible to assemble them using presently available nanofabrication techniques. However, the structural regularity can increase the complexity of realizing logic at nearly every other level of the design hierarchy. One pays a penalty in the use of area and time in order to program topologically-irregular logic circuits into a physically homogeneous crossbar architecture. For example, programmable microscale circuits such as FPGAs incur approximately a 20 to 50-fold area penalty [77] and a 15-fold delay penalty [78] when compared to heterogeneous, custom-designed solutions. Thus, one significant challenge for nanoprocessing lies in developing programming algorithms that can produce area- and time-efficient realizations of heterogeneous logic using regular structures.

Furthermore, microscale PLAs and FPGAs are "mostly" regular, but some irregularity often is introduced at the lowest levels of the hardware hierarchy in order to promote more efficient utilization of physical resources [73]. Likewise, the ability to provide even a limited amount of irregularity with future nanofabrication methods might have a large, beneficial impact on the overall density and performance of a nanoprocessor.

In addition to the challenges enumerated above, the task of designing and developing novel nanoprocessor architectures must confront further difficulties in the circuit and device domains. Some of these challenges also are faced in the development of nanomemories, as described and illustrated in previous work [22]. For nanoprocessing, such issues are compounded. For example, in nanomemories, the use of two-terminal devices without gain

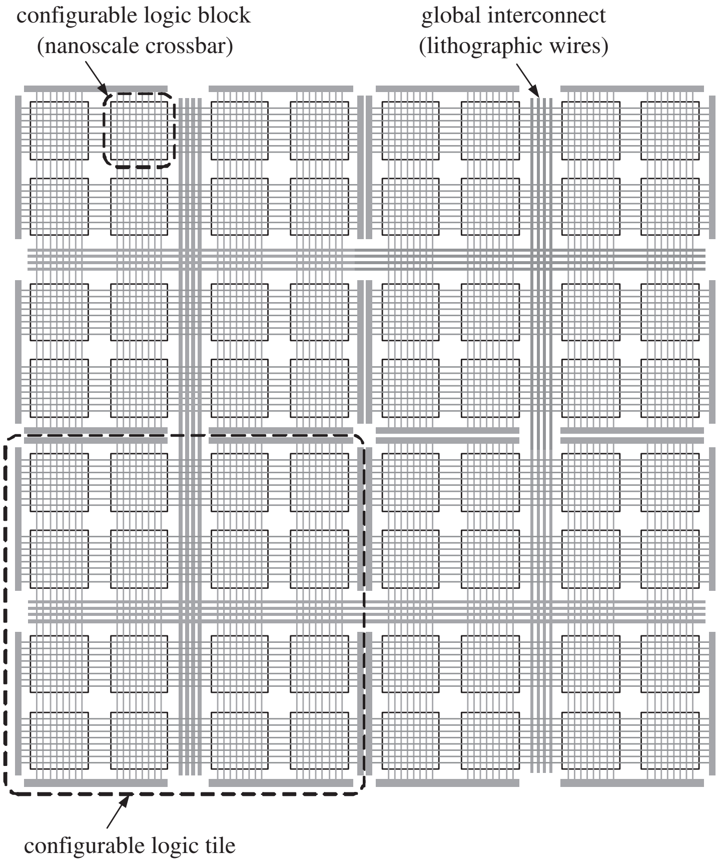


Fig. 17.4. A programmable fabric incorporates molecular-scale devices into the crossbar structures shown in Fig. 17.1. The fabric builds from them an extended structure of molecules or molecular devices, crossed nanowires, and microwires, such as is shown above. This can provide a platform for realizing a nanoprocessor [23]

imposes system-level constraints due to requirements for signal restoration. In nanoprocessors, requirements for signal restoration are more stringent, because the signals may need to traverse larger portions of nanoscale circuitry without the aid of the microscale amplifier circuits proposed for use with nanomemories [79]. Also, wires and the signals they carry must fan out in order to construct the complex logic required for processing, such as arithmetic functions. Still further, as in nanomemories, there are issues of signal integrity due to the signal coupling that arises when devices and interconnects are as densely packed as is proposed for nanoprocessors. The high density of devices also will make difficult the task of maintaining a low enough power density so that system temperature can be controlled [80].

A challenge for nanoprocessing that does not arise in nanomemories is that sequential (clocked) elements will be required. Such elements can be inefficient to realize using the combinational logic that is most readily available using crossbars that incorporate molecular-scale resistors and rectifiers. Specialized nanocircuits have been proposed to serve as sequential elements [68, 81–84]. These circuits operate using Goto pairs [85] in implementations that were used previously in solid-state nanoelectronic circuit designs [86, 87]. In crossbars, these circuits may be built by incorporating NDR molecules [18].

One virtue of using Goto-pair-based circuits for nanoelectronic systems is that they can provide restoration using only two-terminal devices. In effect, these circuits can provide some of the gain required to restore logic signals, thus reducing the gain requirements for the other circuits in the system. Such circuits might be able to limit, and possibly even eliminate, the need for nanotransistors. However, a potential drawback is that, unlike transistor-based circuits, Goto-pair circuits may require additional components in order to provide electrical isolation between logic stages. Such isolation might be provided by distinct nanodevices such as rectifiers. However, with or without such additional devices for isolation, localized insertion and placement of Goto-pair-based clocked elements into a crossbar array probably would require introducing a degree of heterogeneity into an otherwise regular nanofabric.

A recent development that has the potential to alleviate some of these difficulties is the crossbar latch designed by the Hewlett-Packard Corporation [51, 88]. This latch has been demonstrated to produce signal restoration and inversion using only molecular two-terminal devices. It is a clocked element that is designed to be fabricated using junction molecular devices within the same homogeneous crossed-nanowire molecular-scale circuit systems (see Fig. 17.4) that have been used to fabricate nanomemories [4, 6, 21, 89]. Such latches could be introduced into nanoprocessor systems based on crossbars, without requiring a heterogeneous set of devices. Furthermore, as with the Goto-pair circuits, the use of these crossbar latches in a nanoelectronic system might reduce gain requirements for other circuits in the system, even to the point where nanotransistors may not be required. Nanoprocessor system architectures based on these latches still are under development [90].

For all approaches to nanoprocessor system design based upon molecular switches, it is well understood that many device-level challenges also must be addressed [2, 59]. Impedance matching between bulk solid contacts and molecular-scale devices, precise characterization of device behaviors, variability, and yield of devices are among the chief examples. These challenges will be discussed further in connection with the nanoprocessor simulations described in Sect. 17.6. These issues must be managed either by improving fabrication capabilities or by introducing defect and variation tolerance into system architectures.

17.4 A Brief Survey of Nanoprocessor System Architectures

17.4.1 Overview

The previous section discussed some of the challenges facing the design and fabrication of future nanoprocessors based on novel nanodevices and new nanofabrication techniques. In this section, we survey the major architectural approaches that have been proposed to address these challenges. Some of these approaches rely on new architectural paradigms that are very different from those applied in conventional microprocessors. Others borrow heavily from these microprocessor architectures. However, all of these nanoscale approaches attempt to harness molecules or molecular-scale structures to build up electronic circuits and systems. These approaches and the nanoelectronic systems that will be developed in accordance with them have the potential to utilize effectively the much higher device densities that are possible at the nanoscale. Further, because they take advantage of potentially inexpensive, novel nanofabrication techniques, it may be possible to address the issue of exponentially rising costs that presently plagues the microelectronics industry [91, 92].

Substantial progress also continues to be made in the scaling of complementary metal-oxide-semiconductor (CMOS)-based conventional microprocessors. Thus, some nanocomputer architects propose to leverage the substantial knowledge and infrastructure available in CMOS technology. Rather than devise new or modified architectures to accommodate the properties of novel nanodevices, these architects attempt to use them simply to augment the CMOS devices employed in conventional microprocessors. For the most part, such efforts retain conventional microprocessor architectural designs.

In the following sections, both the scaling of conventional architectures and the development of novel approaches are discussed. First, in Sect. 17.4.2, the aggressive miniaturization of conventional architectures to the molecular scale is described. Second, in Sect. 17.4.3, alternatives to conventional architectures are described for cases in which recent nanodevice and nanofabrication developments have made such architectures especially relevant.

17.4.2 Migration of Conventional Processor Architectures to the Molecular Scale

Virtually all conventional microprocessor architectures use CMOS to implement a basic architectural design originally due to von Neumann, Mauchly, and Eckert [93–95]. First described in the 1940's, this architecture divides a computer into four main "organs:" arithmetic, control, memory, and input/output. Present examples of such CMOS-based processors include the well-known Intel Pentium® 4 and the AMD Opteron$^{\mathrm{TM}}$ chips. As Fig. 17.5 shows for the AMD Opteron,$^{\mathrm{TM}}$ the organ structure still is evident.

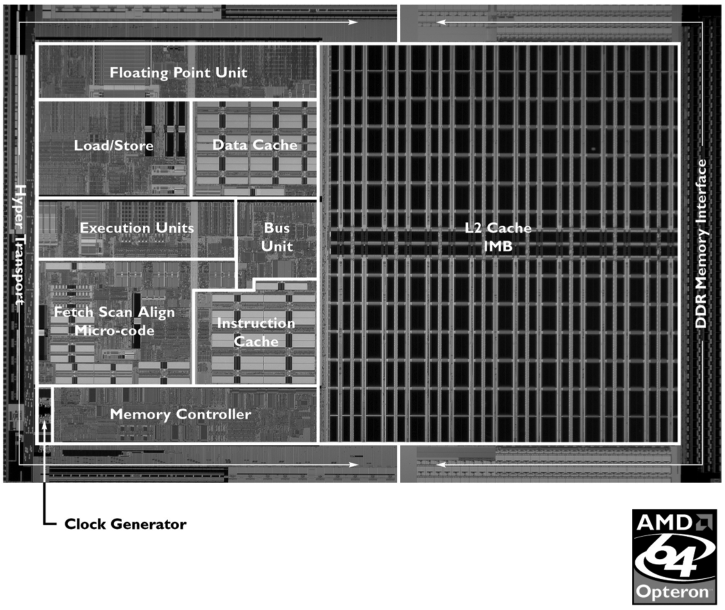


Fig. 17.5. AMD Opteron$^{\text{TM}}$ die photo with annotated block structure [96]

Because of its long-term investment, industry places a high premium on maintaining these architectures as it seeks to achieve ultra-dense integration on the nanometer scale. The primary industry approach today to building nanoprocessors is the aggressive scaling of CMOS technology to nanometer dimensions.[3] However, for a number of years, industry investigators and others have examined the likely limits of CMOS technology [98–100, 102, 103] and the possibility that it might not be cost-effective to use it to build commercial systems with devices scaled down to a few tens of nanometers. This is one of the reasons that new architectural ideas inspired by nanotechnology and molecular-scale electronics are so compelling.

An alternative to the straightforward, two-dimensional, aggressive scaling of CMOS is to expand silicon technology into a third dimension [102]. Three-dimensional integration, or 3-D CMOS [104, 105], refers to any of several methods that take conventional, "flat" CMOS wafers and stack them together with an inter-wafer interconnect [106–111]. For microprocessors, it has been shown that 3-D integration allows for a substantial improvement in performance, and, furthermore, that this improvement increases as device and interconnect dimensions decrease [112]. Therefore, 3-D architectures may

[3] This topic has been reviewed and discussed extensively elsewhere [97–101]. We include a brief discussion of it here both for completeness and to provide a reference point for the other, more novel approaches we discuss.

have particular utility in combination with novel molecular-scale devices, such as might be implemented using a 3-D crossbar array.

So-called "hybrid" approaches that incorporate novel nanostructures into CMOS devices constitute a third avenue by which conventional processor architectures may be migrated toward the molecular scale. Major industrial research laboratories have begun to explore how nanowires and CNTs might be employed to enhance CMOS and CMOS-like structures. For example, some of the Intel Corporation's designs for future transistors call for the incorporation of silicon nanowire channels to increase current density and to control short-channel effects [113]. Similarly, work at IBM has examined the increased current that results from the insertion of CNTs into CMOS field-effect transistor (FET) channels [114].

Another hybrid approach involves the use of self-assembled monolayers of redox-active molecules to enhance the function of traditional silicon devices. Thresholds and conductances of the underlying silicon substrate can be altered by the incorporation of these monolayers. In addition, new and novel devices might be enabled. For example, the redox states of the molecules in the SAMs may be used to form multi-level bits (i.e., n-ary digits) [115, 116]. Such so-called molecular FETs, or MoleFETs, which employ NDR molecules or charge-storage porphyrin molecules on silicon, might be used to implement multi-level memories or logic. It appears that molecules and molecular layers can be inserted into CMOS production processes for this purpose. For example, the porphyrin molecules proposed for some of these hybrid devices have been shown to be able to survive the 400°C processing temperature used for conventional CMOS components [117]. Also, Nantero Corporation is succeeding in introducing novel carbon nanotube-based devices and circuits into a CMOS production line [118].

Hybridization also may be employed at the architectural level. An example of such a hybrid design is the CMOL architecture. As is depicted in the previous chapter of this volume [69], CMOL circuits combine CMOS with crossed nanowires and molecular devices (see also Fig. 17.1 and Sect. 17.3.3 of the present chapter). Specifically, CMOL circuits are to be fabricated in two layers, with one layer consisting of CMOS blocks, or "cells," and the other layer containing an array of crossed nanowires employed as interconnects between the CMOS cells. As with many other crossbar architectures, the nanowire crosspoints are designed to contain programmable molecular devices. These devices should permit reconfiguration of the nanowire-based connections between the CMOS cells. Therefore, if physical experiments confirm the designers' preliminary analyses [69, 70], it is likely that CMOL may be used to implement any architecture based upon programmable interconnects. Thus far, quantitative analyses of the CMOL designs seem promising, but no fabrication experiments have been undertaken to build and test CMOL circuits.

Two types of CMOL circuit architectures have been proposed, neural networks and FPGAs [69, 70]. The salient features of these architectures serve here to illustrate the potential advantages and challenges of hybrid CMOS/nano designs in general. For example, like microscale neural networks, the CMOL neural-network architecture exploits parallelism in order to reduce operating speed and thereby save power. However, the designers assert that, unlike microscale neural networks, CMOL may be able to achieve the density of cells and interconnects required in order to emulate advanced neural networks, such as organic brains. As another example, CMOL FPGAs may be able to improve performance significantly, relative to conventional CMOS FPGAs, by utilizing nanoscale interconnects. This is because conventional FPGA circuit performance is limited predominantly by interconnect performance [119]. In both examples, CMOL is designed to build upon conventional CMOS circuitry, yet circumvent its limitations by exploiting the ultra-high density of devices available at the nanoscale.

In general, hybridizing at device, circuit, or architectural levels may allow the semiconductor industry to leverage the best features of both conventional CMOS and novel nanostructures. However, this combination does introduce additional challenges. One potential difficulty lies in designing the interface between CMOS and nanoscale components. For systems built solely from nanodevices, such an interface is required only at a relatively small number of points at the periphery of the nanoelectronic circuit system. In contrast, hybrid architectures necessitate tighter and denser integration of the many, many individual CMOS components and nanostructures *within* the circuit system.

For example, the CMOL approach proposes novel interface pins to accomplish this task [69]. However, such pins must be manufactured to tight, sublithographic tolerances. Also, to contact these pins, precise linear and angular alignment of the nanowire array is likely to be required. In addition, because of the high bandwidth of communication proposed between CMOS and nanoscale components, impedance matching between these components is a potential source of difficulty. Experimental measurement of the impedances of proposed interface pins must be undertaken in order to determine the suitability of such pins for hybrid approaches.

A more fundamental difficulty introduced by combining CMOS with nanostructures is that overall scalability may be limited by the scalability of CMOS technology. Such technology is almost certain to hit physical barriers to further scaling. Thus, new processor architectures must be devised that can operate solely with novel nanodevices. Proposals for such architectures are reviewed in the following section.

17.4.3 Overview of Novel Architectures for Nanoelectronics

A set of clever, yet profound architectural concepts underlies the prototype nanomemory and nanoprocessor circuit systems that just now are emerging

[23,25,32,59,61,63,65]. These architectural innovations seek to take advantage of the strengths of novel nanodevices (especially, high device density and non-volatile, low-power operation), as well as to ameliorate some of the limitations discussed in Sect. 17.3 in the techniques presently available for fabrication and assembly at the nanoscale (e.g., the inability to place nanostructures precisely or to make them readily with arbitrary shape or complexity). At the highest level, one may view these architectural innovations as falling into two classes, as discussed in the subsections immediately below.

Radical Departures from Microelectronic Architectures

One broad class of architectures has been devised strictly by taking demonstrated nanodevices and considering how to combine them into circuits or circuit-like structures that may then be fashioned into complex systems. This bottom-up style of nanoprocessor design has resulted in a number of architectural approaches that differ drastically from conventional architectures. These novel approaches, which are considered in detail elsewhere, include quantum cellular automata (QCA) [32, 33, 60–62], nanoscale neural networks [63, 69], nanocells [28, 64, 65], and biologically inspired electronic system structures such as the virus nanoblock (VNB) [120, 121]. Each of these encompasses important ideas and has virtues either in ease of fabrication or in ultra-low power consumption.

The QCA approach seeks to use electric fields, rather than currents, to set bits and propagate signals by moving the charge distributions in arrays of multi-quantum-dot structures termed quantum-dot cells. The primary virtue of this approach is that it is predicted to have ultra-low power dissipation, which is highly desirable in a very dense array of nanostructures. Also, the very small size of molecular quantum dots may permit this scheme to operate at room temperature, in contrast to solid-state QCA approaches that require cryogenic operation. However, a circuit employing a molecular QCA approach has not yet been demonstrated.

The nanocell architecture employs an array of nanoparticles randomly distributed and randomly connected by self-assembled molecules that typically exhibit negative differential resistance and voltage-dependent switching. No attempt is made to control the placement of the molecules that make up the individual interconnects; rather, the designer takes advantage of the molecules' switching characteristics to program the nanocell after it has been assembled. Input and output connections are fabricated on the lithographic scale using conventional techniques. This permits relative ease in manufacturing nanocells, as well as in connecting them to form higher-order circuits. As such, high-level designs may be possible that are similar to today's Very Large Scale Integrated (VLSI) circuits [64].

The nanocell architecture avoids potential difficulties in precise nanoscale fabrication. Instead, the desired connectivity is established by intensive post-fabrication testing and programming. Because of its random assembly and post-fabrication programming, the nanocell approach is inherently defect and fault tolerant [64]. Experimental nanocell memories recently have been fabricated [28] and logic gates have been simulated, but not yet demonstrated.

These architectures, which depart significantly in their operational and organizational principles from those of present-day computers, may make important contributions over the long term. However, their differences from almost all industry architectures mean that they cannot harness easily the significant infrastructure developed by the existing electronics industry. Thus, at the moment, they have more hurdles to overcome and appear to be further from being applied to build extended nanoprocessing systems than the regular array structures discussed below.

Regular Array Architectures Derived from Microelectronics

This second class of novel nanoelectronic architectures is derived via the adaptation and ultra-miniaturization of microelectronic FPGAs and PLAs so that they can be implemented with novel nanodevices and new nanofabrication techniques. For the purposes of achieving some near-term successes in developing and operating prototype nanoprocessors, these regular arrays occupy an important middle ground between the radical departures discussed above and the very inhomogeneous architectures used in conventional microprocessors. Nanoarray architectures have an appealing structural simplicity that takes advantage of a number of the strengths of novel nanodevices and nanofabrication techniques. Thus, physical prototypes of extended nanoarray processors are approaching realization based upon much systematic effort [3, 5–7, 21, 25, 75], including the detailed simulations described in Sect. 17.6.

There have been criticisms of the use of PLAs to develop nanoprocessors [69]. Some of these criticisms are premised on the assumption that nanoPLAs will not incorporate gain-producing or restoration-producing nanodevices. However, this is not necessarily the case. For example, the nanoPLA architecture due to DeHon and Wilson [25] does incorporate gain-producing nanowire-based nanotransistors, as is described in detail below. Other criticisms focus on the issue of heat dissipation. This is a valid concern, due to the high density of current-based devices. However, circuit techniques, such as the use of dynamic instead of static logic, may alleviate this problem [25].

Thus, because the path to the realization of these novel nanoelectronic architectures seems clearer and nearer at hand, the rest of this chapter will focus on a discussion of the operational principles, advantages, and trade-offs of FPGA- and PLA-type nanoarray processor architectures.

17.5 Principles of Nanoprocessor Architectures Based on FPGAs and PLAs

17.5.1 Overview

Having provided a brief survey above of various architectural approaches for nanoprocessors, we now focus our attention exclusively on regular arrays such as FPGAs and PLAs. Until recently, the use of such regular arrays in general-purpose, microscale computation has been disfavored relative to the use of conventional, heterogeneous architectures. Thus, to understand how regular arrays may be leveraged for nanoprocessing, it is important to review their use in conventional processing systems and to illustrate the benefits and challenges. Following this brief review, a specific regular architecture for a nanoprocessor will be explored, the DeHon-Wilson PLA.

17.5.2 Description of Regular Arrays, FPGAs, and PLAs: Advantages and Challenges

A regular array is a homogeneous two- or three-dimensional grid of configurable logic elements (such as four-input logic tables) interconnected by wires with embedded programmable switches (i.e., "programmable wires") [73]. The array is configured by programming the individual logic elements and switches to define a hardware implementation of a desired logic function. Thus, regular arrays attempt to eliminate heterogeneity at the hardware level, introducing it at the software level instead. Present fabrication methods for nanoelectronics, which rely on bottom-up, self-assembly approaches, produce such homogeneous systems of nanostructures relatively easily [3, 5, 75].

In conventional microelectronics, regular structures are employed for special-purpose applications in the form of circuits such as FPGAs and PLAs. A schematic diagram of a PLA is given in Fig. 6(a). Figure 6(b) shows an extended system architecture based on PLAs. This system structure is similar to that used for FPGAs. (See Sect. 17.3.3 for a brief description of FPGAs.)

Because of the underlying homogeneity of such structures, thus far they have been outperformed by classical microprocessor architectures at carrying out general-purpose computation. For a given application, an FPGA may be programmed to outperform a general-purpose microprocessor. However, a key capability of general-purpose microprocessors is their ability to switch rapidly between various applications. If the FPGA is configured to provide an equal amount of so-called "context switching" capability, the FPGA implementation usually lags in performance [77].

This is because the general class of functions that can be computed by a conventional processor is quite large, and the best way to compute the whole class of functions on an FPGA has been to program the FPGA as a conventional processor. This is inefficient. However, this inefficiency is not believed to be fundamental. It may be the case that migration to the nanoscale will

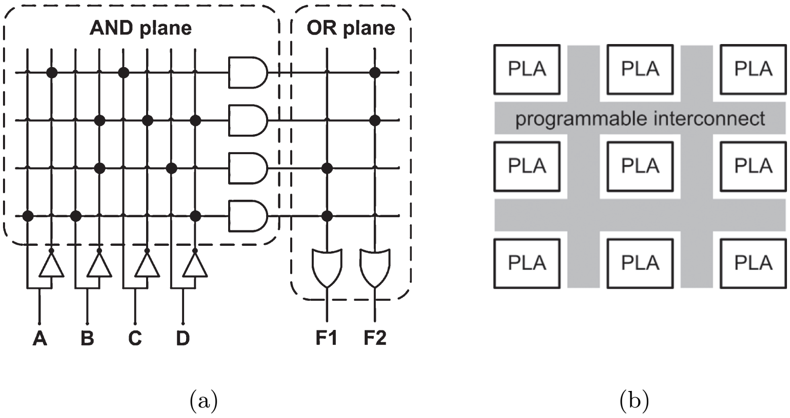


Fig. 17.6. Schematic illustrations of (**a**) a single PLA and (**b**) an extended system architecture based on an array of PLAs. A single PLA consists of a plane of AND gates followed by a plane of OR gates. The interconnections between these gates are reconfigurable after fabrication. In this example, output F_1 is programmed to compute $(A\ AND\ B\ AND\ (NOT\ D))\ OR\ ((NOT\ B)\ AND\ D)$, based on the configured connections shown by the black dots. More complex, hierarchical logic can be constructed using an array of PLAs, such as is shown in part (**b**). Here, outputs such as F_1 and F_2 can be used as inputs to other PLAs in the array

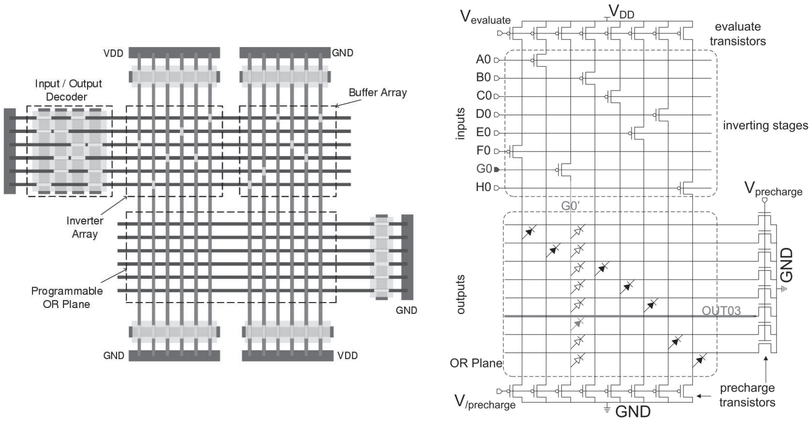
address this problem. At the nanoscale, it is conceivable to operate with many trillions of devices per processor. With so many devices, it may be possible to implement simultaneously all the required functions that make up a given set of programs [122]. Similarly, the existence of programmable nanoscale interconnects may improve the efficiency of array-based implementations, since the area overhead of each switch can be reduced.

Thus, due to the large number of available devices and the inherent regularity produced by several nanofabrication methods, array architectures have become prominent in nanocomputation research. In the next section, we will describe one such promising architecture, due to DeHon and Wilson [25].

17.5.3 The DeHon-Wilson PLA Architecture

A very thoroughly thought-out example of a nanoarray architecture that utilizes nanowires in readily realizable crossbar structures is the DeHon-Wilson PLA architecture [23–25, 37]. A high-level diagram of this architecture is shown in Figs. 4 and 6 (b), while Fig. 7(a) provides a detailed view of the low-level implementation. As with microelectronic PLA-based designs [26], the large-scale architecture of this nanoprocessor combines a number of PLAs into still larger arrays.

In general, a PLA consists of a programmable AND plane (with a number of AND gates in parallel) followed by a programmable OR plane (with



(a) (b)

Fig. 17.7. Illustrations of (**a**) the DeHon-Wilson PLA Architecture and (**b**) an 8×8 inverting block. The eight vertical wires shown in part (**b**) correspond approximately to the vertical wires in the left-hand side of the subarray in part (a)

a number of OR gates in parallel), as shown in Fig. 6(a). Inverters also are available for all inputs. Since any combinational logic function can be written as the OR of some number of AND terms, any such function can be synthesized using a PLA, assuming the PLA is large enough to contain all the logic terms [38].

In the DeHon-Wilson design, a crossbar subarray is used to provide the logical equivalents of the AND and OR planes of the PLA, as shown in Fig. 7(a). The system is extended by tiling crossbar subarrays, as illustrated in Fig. 4. Figure 7(a) shows the four major subsystems of the DeHon-Wilson PLA implementation: an array of crossed-nanowire diodes used as a programmable OR plane, one inverting subarray of crossed nanowire transistors, a similar buffering subarray, plus an input/output decoder. The inverting and buffering subarrays each are used to regenerate signals and maintain their strengths.

In this PLA scheme, the AND planes are replaced by logically-equivalent pairs of inverting subarrays and OR planes. Figure 7(b) shows a more detailed circuit-level characterization of the left-hand side of the system in Fig. 7(a). In the bottom half of the subarray shown in Fig. 7(b), all the crossed-wire junctions are taken to contain switchable or "programmable" diodes. By programmable, we mean that the diode can be set to either a high ("on") or low ("off") conductance state in the conductive direction. Where the diodes are not shown, they are taken to be always off, so that the block depicted produces the desired function. Where the solid diodes are shown explicitly, they are taken to be always on. The hollow diodes are turned on or off within a simulation in order to test fan-out in the system. In addition to the transistors and diodes shown, the representation in a simulation can incorporate parasitic resistances and capacitances that have been omitted for visual clarity.

The DeHon-Wilson architecture is notable because it is designed explicitly to tolerate shortcomings in present-day nanofabrication. Within the crossbars of the DeHon-Wilson architecture, redundant wires are used to overcome potential failures due to misalignment or physical defects. A stochastic scheme is used to connect to and thereby address specific wires so that unique addressing can be nearly guaranteed without the need to pick and place individual wires [24]. Also, the inverter and buffer arrays can function in two modes, static and dynamic [25]. In dynamic mode, static power consumption is reduced [38]. This ameliorates the potential problem [69] of heat dissipation in ultra-dense, current-based designs.

Efforts are underway to implement the DeHon-Wilson architecture. Prior to its actual fabrication, there are parameters that remain to be tuned and assumptions that remain to be verified. The most cost-effective method for doing this is the use of nanoprocessor system simulation, as has been demonstrated convincingly in the development of conventional microprocessors [123] and as is discussed further below.

17.6 Sample Simulation of a Circuit Architecture for a Nanowire-Based Programmable Logic Array

System simulation can produce an integrated, multi-level view of candidate nanocomputer architectural performance. This view considers optimization at the device level simultaneously with the problems of designing the system at the circuit and architecture levels. At this early stage of nanocomputer development, it is possible to provide useful insights and guidance to device developers, as well as system architects, by simulating even small component circuits and subsystems. In the following subsections, we describe a simulation and analysis of the DeHon-Wilson PLA [25].

17.6.1 Methodology for the Simulation and Analysis of Nanoprocessors

The details of our nanoelectronic system simulation methodology have been described previously [22]. Thus, only a brief overview is presented here.

The simulation methodology consists of three parts. First, empirical I-V behavior models are developed for the fundamental component nanodevices and small prototype circuits that have been demonstrated experimentally. Second, these device models are incorporated into schematic descriptions of the nanoprocessor system, as well as extended subsystems, based on the architectural design. Finally, system simulations are carried out using representative inputs. The simulations have the effect of extrapolating from the experimentally known device behaviors and the planned circuit designs to obtain a projection of the behavior of an entire nanoprocessor system. From these results, the tuning of device, circuit, and architectural design parameters may be investigated for its effect on the overall system performance. Significantly, this may be done in advance of time-consuming, costly, and difficult trial-and-error experiments.

The primary simulation software that has been used in the work described here is the DFII integrated-circuit computer-aided design package available from Cadence Design Systems of San Jose, California [124]. This commercial off-the-shelf software tool was chosen for the substantial time savings and reliability associated with the use of readily available, well-tested software. The Cadence package also incorporates particularly useful features, such as a graphical interface and modeling languages, that have been developed over many years specifically for the flexible modeling of custom circuits and devices. This enabled the authors to adapt the simulation tools particularly for the novel nanoelectronic devices and circuit structures described here. Individual device models were developed from empirical fits of experimental data, either from published literature or provided by the developers, using the hardware description language (HDL) Verilog-A. These empirical models then were incorporated into the component Spectre circuit simulator, which supports co-simulation of both novel components modeled in Verilog-A and conventional devices modeled using SPICE.

17.6.2 Device Models for System Simulation of the DeHon-Wilson NanoPLA

Construction of a nanoprocessor according to the DeHon-Wilson nanowire-based PLA architecture requires four distinct nanodevices, each of which requires a distinct I-V behavior model within the system simulation. All four of these devices are represented, for example, in the schematic in Fig. 7(b). Three of these devices are the nonvolatile nanowire (NVNW) diode, the microwire top-gated FET (TG-FET), and the nanowire interconnects. A diagram of the nanowire diode is depicted in Fig. 2(a). It is made from two crossed, bandgap-engineered nanowires. The microwire TG-FET resembles the crossed-nanowire FET shown in Fig. 3(a), except that for the TG-FET, the top wire is a much larger microwire. Detailed descriptions of the I-V behavior models derived for all three of these devices are published in prior work [22].

The fourth device and device model required for the nanoPLA is the crossed-nanowire FET (cNWFET) [5, 34–36], which acts as the input transistor for the restoration blocks. The cNWFETs are constructed by crossing a nanowire over another nanowire that is coated with silicon dioxide, as depicted in Fig. 3(a) [34]. The oxide isolates the coated nanowire and allows it to act as the channel of a field-effect transistor, while the uncoated nanowire serves as the gate. Figure 3(b) shows an I-V behavior model that has been developed for this device and incorporated into the simulations. This model reproduces published experimental I-V characteristics [19], although some extrapolation beyond the measured voltages was necessary.

One important observation from the I-V characteristics of the cNWFETs is that the experimentally-observed threshold voltage (V_T) of the p-channel FETs (PFETs) ranges into positive values. In contrast, conventional microelectronic circuits employ PFETs that have a negative threshold [38]. Some circuits, including the ones we explore here, can be made to function correctly using PFETs with positive thresholds. However, such operation is disadvantageous. In static mode, these circuits consume a great deal of power and usually are not capable of providing adequate signal restoration. Thus, dynamic-mode operation would be preferable. However, for the dynamic operation of the circuits we examine, the PFET V_T threshold must be negative.

Recent experimental results suggest that nanowire p-channel transistors can be fabricated with the desired negative thresholds [35] and that the value of this threshold can be controlled [36]. Based on these experimental results, we have extrapolated a cNWFET model with a reasonable negative value for the PFET threshold voltage. Use of this model permits simulation of these circuits in dynamic mode.

With the device models developed for all required devices, as described above, system simulations were conducted in accordance with the proposed architecture shown in Fig. 17.7. Parasitic behaviors of the nanowire arrays, such as coupling capacitance, also were incorporated.

17.6.3 Simulations and Analyses of the NanoPLA

The simulations described here consider primarily the performance of a 64-bit PLA. This is represented by an 8×8 OR plane driven by eight inverting stages, as shown in Fig. 7(b). The PLA is programmed with the pattern of diodes depicted there and described in Sect. 5.3. The input vectors to the PLA are given in Table 1.

The generally accepted method for determining the viability of a circuit system is to assess its operation under the least favorable circumstances. Thus, analysis is performed here by examining the worst-case high and low output voltages. The signal OUT_{03}, which is labeled in Fig. 7(b) and is the inversion of the G_0 input, is likely to produce the worst-case measurements. This is because, given the switch configuration shown, the length of wire traversed for this output is greatest, which results in the largest parasitic resistance and capacitances.

Functionality of the circuit can be determined by providing a specific input waveform and programmed function, then simulating the output waveform to determine if the function is realized. Such a simulation is illustrated in Fig. 8, which shows an output waveform for OUT_{03} when the circuit in Fig. 7(b) is programmed to implement the inversion of G_0. Also shown is the clocking scheme (i.e., the precharge and evaluate signals) for operating the inverting

Table 17.1. PLA Input Vectors

A_0	B_0	C_0	D_0	E_0	F_0	G_0	H_0	
1	1	1	1	1	1	0	1	High Output
0	0	0	0	0	0	1	0	Low Output

block in dynamic mode. To understand this scheme, it is first necessary to appreciate that the circuit operates in dynamic mode by storing charge on the wires and the terminals of the devices. Thus, the precharge signals serve to set the charge state of all these elements (e.g., to a charge state that produces a low voltage equivalent to logic "0"). Then, the evaluate signal is used to change the charge state appropriately on some of the wires and terminals (e.g., those for which the correct logic value would be "1").

The dynamic precharge-evaluate cycle first begins when the precharge signal goes high. This has the effect of switching on the n-channel FETs at the right of Fig. 7(b), to discharge the outputs of the inverting block to a low voltage. After the precharge is completed, the evaluate signal transitions to a low voltage, which turns on the evaluate PFETs at the top of Fig. 7(b) in order to produce the desired output signal on OUT_{03}. As can be seen in Fig. 8, the OUT_{03} waveform will continue to be pulled to a high voltage until the evaluate signal is turned back high. After the evaluate transistors turn off, the signal begins to drop, due primarily to leakage through the transistors.

Analyses based upon simulations of this type allow the determination of system behavior and limits. For example, by setting *a priori* the levels for the minimum logic "1" voltage and maximum logic "0" voltage, a minimum operating frequency may be calculated from the signal decay data shown in the bottom graph of Fig. 8. Thus, these simulations can help characterize how transistor leakage impacts the performance of the system.

Alternative simulations can examine still other effects. For example, diode loading can affect system operation. Simulations suggest that there is a limit to the number of diodes that may be turned on and permitted to load a single input column of the inverting stage. For one such simulation, Fig. 17.9 shows the output-voltage dependence of the number of diodes programmed in the "on" state along the G_0' column (see Fig. 7(b)), which drives the OUT_{03} output row. The high output voltage, and thus the voltage swing, is reduced as more diodes are programmed "on" and load the driving column. This is a result of current being divided among multiple outputs.

From another simulation for which results are plotted in the bottom curve of Fig. 17.9, it is seen that the low or "0" output voltage signal remains relatively constant as the number of "on" diodes is increased. This is because the input vector in Table 17.1 used in this simulation for the low output drives all the row wires in Fig. 7(b) except OUT_{03} to logic "1." This has the effect of

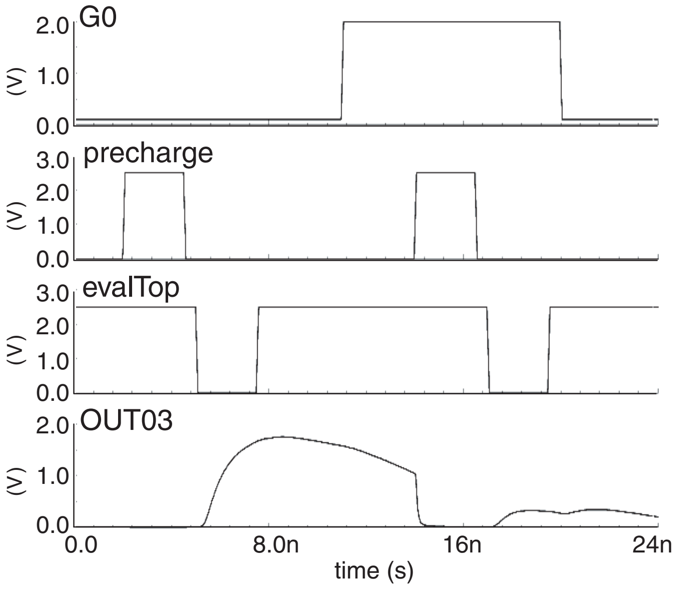


Fig. 17.8. Waveforms describing how the circuit in Fig. 7(b) inverts input signal G_0 to produce output signal OUT_{03}. See discussion in text

reverse-biasing all the diodes on the G'_0 column that connect to rows other than OUT_{03}. Thus, little current will flow through the diodes into those rows.

While these results show that the circuits can function correctly, they also suggest a limit to the number of "on" diodes that can load the restoring columns. The simulations suggest the maximum number of diodes that can load each column is approximately five. Otherwise, it is found that the voltages representing "1" and "0" get so close together that they cannot be distinguished by the gates in the downstream logic stages. Thus, there is a limit on the number of functions that may use the same input.

There are a number of ways to increase this limit. One way would be to reduce leakage through the nanowire transistors. This requires that difficult experiments be carried out in order to alter device performance appropriately. Another way to increase the limit would be to increase the capacitance at each output. However, this increased capacitance, which takes longer to discharge, also takes longer to charge. This reduces the maximum operating speed of the system. Still a third way would be to introduce duplicate columns, where the input transistors are driven by the same row nanowire.

Also, the restoration-producing portions of the nanoPLA array are likely to be particularly sensitive to variability in the nanodevices. In simulations we have performed on the buffering subarrays, it is seen that a buffer can fail to restore signals adequately if the control signals that would derive from

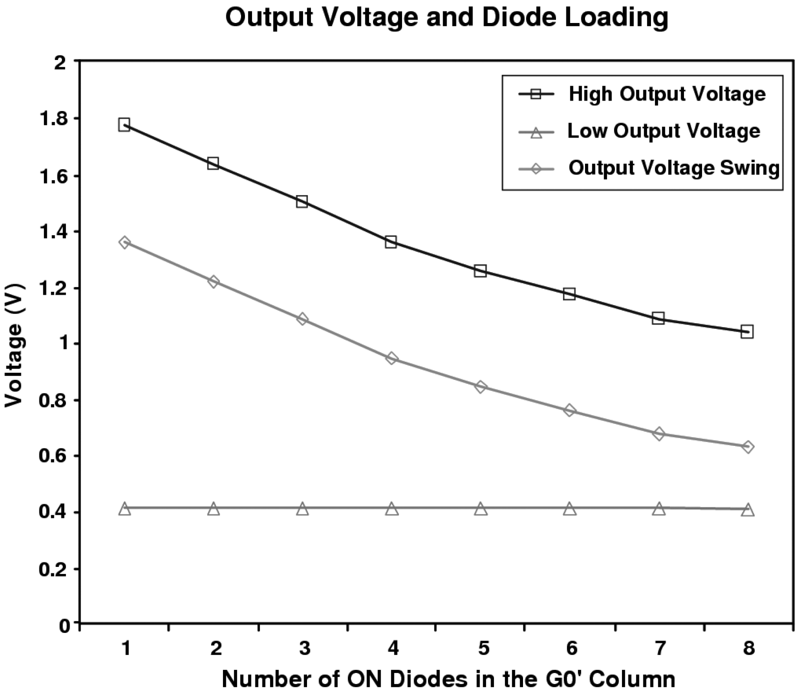


Fig. 17.9. High and low output voltages and output voltage swing plotted against the number of diodes programmed ON in the G_0' column

other logic subsystems vary outside of a small acceptable range. A likely source of control signal variation is variation in the structures of devices.

Specific results and design guidance, such as are described in the examples above, illustrate that system simulation is an effective way to extrapolate from device experiments to consider and improve various nanoelectronic system design options.

17.6.4 Further Implications and Issues for System Simulations

Although the results shown above are derived from simulations of a particular nanoprocessor system, the implications are significant for a wide variety of potential designs and architectures. Any system based on electronic currents flowing through densely-packed circuits must consider issues such as signal integrity, power density, fan-in, fan-out, and gain. For example, we have shown explicitly in Sect. 17.6.3 how the design of such systems must consider fan-out, which in the DeHon-Wilson architecture is the number of diode-connected rows a single inverting column can drive. Fan-out is an important issue to the design of any nanoscale architecture, in that greater fan-out capability aids in reducing the number of logic levels and the area required when implementing complex functions. Several of the nanoscale architectures proposed to date are based on PLAs, much as is envisioned in the DeHon-Wilson architecture [23,25,59,66,67]. As such architectures move toward realization, it will be up to device and circuit designers to find ways to address issues like fan-out for the purpose of optimizing system robustness.

It is important to note that the simulations presented here represent only the first steps toward detailed, extensive simulations of complete nanocomputer architectures. There are further issues that must be explored for the DeHon-Wilson architecture and other architectures. These issues include system impacts of crosstalk, transistor leakage, and power density. Crosstalk, the loss of signal through coupling capacitances between neighboring wires, can impair significantly the performance of any system consisting of closely-packed wires. Understanding the extent of crosstalk, and devising means for controlling it, can provide design flexibility to improve signal integrity, while possibly reducing power density. Leakage current is another factor that contributes to increased power consumption and to signal degradation. Preliminary experimental data suggest that leakage currents can be relatively large for many of the devices used in this architecture. This would result in increased static power consumption and decreased output voltage-level stability. While it probably will be feasible to reduce the leakage, this will require further careful experimentation.

Well in advance of such time-consuming experiments, system simulations can indicate the extent to which such enhancements in devices might improve system performance. If such improvements are significant, then it becomes worthwhile for experimentalists to invest in enhancing designs and techniques for fabricating nanodevices.

17.7 Conclusion

In this chapter, we have surveyed a range of possible architectural approaches to the development of electronic nanoprocessors. Following this survey, we have focused upon architectures that occupy an important middle ground between conventional microelectronic architectures and a set of more radical nanoelectronic architectures. To explore this middle ground, we have adapted the simulation tools and techniques used by the microelectronics industry. In so doing, we are attempting to bridge the gap between the present realm of pure research in nanoelectronics and the application of the resultant innovations in functional, manufacturable systems.

Using the detailed simulations of the subsystems embodied in one such middle-ground nanoprocessor architecture, the DeHon-Wilson PLA, we have examined some of the trade-offs that affect such a system based upon molecular-scale devices. Many of these trade-offs apply to almost any nanoprocessor architecture that might be adopted to harness molecules or molecular-scale devices in ultra-dense electronic computing structures. Thus, we believe that the simulations described here could assist experimentalists to understand better the path they must follow if they are to take steps toward applying their structures and devices.

Work of the type described above translates the hard-won results of difficult experiments upon nanodevices and small circuits into insights that illuminate the new frontier of nanoprocessor systems development. Thus, by simulations such as we have described, coupled closely with device and system experiments, it may be possible both to speed the realization and optimize the performance of ultra-dense electronic computers integrated on the molecular scale.

Acknowledgments

The authors thank Professor André DeHon of the California Institute of Technology, Professor Charles Lieber of Harvard University, Professor Konstantin Likharev of Stony Brook University, plus Phil Kuekes, Duncan Stewart, and Greg Snider of the Hewlett-Packard Corporation for their many generous discussions and for providing detailed information regarding their nanoscale devices and system designs. This research was supported by the MITRE Technology Program.

References

1. K. S. Kwok, J. C. Ellenbogen: Moletronics: future electronics, Materials Today, **5**, 28 (2002).
2. J. R. Heath: Wires, switches, and wiring. A route toward a chemically assembled electronic nanocomputer, Pure Appl. Chem., **72**, 11 (2000).
3. N. A. Melosh, A. Boukai, F. Diana, B. Gerardot, A. Badolato, P. M. Petroff, J. R. Heath: Ultrahigh-density nanowire lattices and circuits, Science, **300**, 112 (2003).
4. P. J. Kuekes, J. R. Heath, R. S. Williams: Molecular wire crossbar memory, United States Patent 6,128,214 (2000).
5. D. Whang, S. Jin, Y. Wu, C. M. Lieber: Large-scale hierarchical organization of nanowire arrays for integrated nanosystems, Nano Lett., **3**, 1255 (2003).
6. Y. Chen, G. Jung, D. A. A. Ohlberg, X. Li, D. R. Stewart, J. O. Jeppesen, K. A. Nielsen, J. F. Stoddart, R. S. Williams: Nanoscale molecular-switch crossbar circuits, Nanotechnology, **14**, 462 (2003).
7. G. Snider, P. Kuekes, R. S. Williams: CMOS-like logic in defective, nanoscale crossbars, Nanotechnology, **15**, 881 (2004).
8. Overview of the DARPA Moletronics Program at `http://www.darpa.mil/MTO/mole/`.
9. Overview of the DARPA MoleApps Program at `http://www.darpa.mil/dso/thrust/matdev/moleapps.htm`.

10. J. Tomfohr, G. Ramachandran, O. F. Sankey, S. M. Lindsay: Making contacts to single molecules: Are we there yet?, in *Introducing Molecular Electronics*, ed. by G. Cuniberti, G. Fagas, K. Richter (Springer New York, 2005), Lecture Notes in Physics, **680**, 301 (2005).
11. W. Wang, T. Lee, M. Reed: Intrinsic electronic conduction mechanisms in self-assembled monolayers, in *Introducing Molecular Electronics*, ed. by G. Cuniberti, G. Fagas, K. Richter (Springer New York, 2005), Lecture Notes in Physics, **680**, 275 (2005).
12. J. van Ruitenbeek, E. Scheer, H. B. Weber: Contacting individual molecules using mechanically controllable break junctions, in *Introducing Molecular Electronics*, ed. by G. Cuniberti, G. Fagas, K. Richter (Springer New York, 2005), Lecture Notes in Physics, **680**, 253 (2005).
13. R. M. Metzger: Six unimolecular rectifiers and what lies ahead, in *Introducing Molecular Electronics*, ed. by G. Cuniberti, G. Fagas, K. Richter (Springer New York, 2005), Lecture Notes in Physics, **680**, 313 (2005).
14. J. C. Ellenbogen, J. C. Love: Architectures for molecular electronic computers: 1. logic structures and an adder designed from molecular electronic diodes, Proc. IEEE **88**, 386 (2000).
15. M. A. Reed, C. Zhou, C. J. Muller, T. P. Burgin, J. M. Tour: Conductance of a molecular junction, Science, **278**, 252 (1997).
16. C. P. Collier, E. W. Wong, M. Belohradsky, F. M. Raymo, J. F. Stoddart, P. J. Kuekes, R. S. Williams, J. R. Heath: Electronically configurable molecular-based logic gates, Science, **285**, 391 (1999).
17. C. P. Collier, G. Mattersteig, E. W. Wong, Y. Luo, K. Beverly, J. Sampaio, F. M. Raymo, J. F. Stoddart, J. R. Heath: A [2]catenane-based solid state electronically reconfigurable switch, Science, **289**, 1172 (2000).
18. J. Chen, M. A. Reed, A. M. Rawlett, J. M. Tour: Large on-off ratios and negative differential resistance in a molecular electronic device, Science, **286**, 1550 (1999).
19. Y. Huang, X. Duan, Y. Cui, L. J. Lauhon, K. H. Kim, C. M. Lieber: Logic gates and computation from assembled nanowire building blocks, Science, **294**, 1313 (2001).
20. A. Bachtold, P. Hadley, T. Nakanishi, C. Dekker: Logic circuits with carbon nanotube transistors, Science, **294**, 1317 (2001).
21. Y. Luo, C. P. Collier, J. O. Jeppesen, K. A. Nielson, E. Delonno, G. Ho, J. Perkins, H. Tseng, T. Yamamoto, J. F. Stoddart, J. R. Heath: Two-dimensional molecular electronics circuits, ChemPhysChem, **3**, 519 (2002).
22. M. M. Ziegler, C. A. Picconatto, J. C. Ellenbogen, A. DeHon, D. Wang, Z. H. Zhong, C. M. Lieber: Scalability simulations for nanomemory systems integrated on the molecular scale, in *Molecular Electronics III* (2003), **1006** of Ann. N.Y. Acad. Sci., pp. 312
23. A. DeHon: Array-based architecture for FET-based, nanoscale electronics, IEEE TNANO **2**, (2003).
24. A. DeHon, P. Lincoln, J. E. Savage: Stochastic assembly of sublithographic nanoscale interfaces, IEEE TNANO **2**, 165 (2003).
25. A. DeHon, M. J. Wilson: Nanowire-based sublithographic programmable logic arrays, in *Proc. ACM/SIGDA FPGA* (ACM Press, Monterey, CA, 2004), pp. 123.
26. B. Zeidman: *Designing with FPGAs and CPLDs* (CMP Books, Lawrence, KS, 2002).

27. M. Barr: Programmable logic: What's it to ya?, Embedded Systems Programming magazine, 75 (1999). Also available online at http://www.embedded.com/1999/9906/ 9906sr.htm.
28. J. M. Tour, L. Cheng, D. P. Nackashi, Y. X. Yao, A. K. Flatt, S. K. S. Angelo, T. E. Mallouk, P. D. Franzon: Nanocell electronic memories, J. Am. Chem. Soc., **125**, 13279 (2003).
29. A. Fijany, B. N. Toomarian: New design for quantum dots cellular automata to obtain fault tolerant logic gates, J. Nanop. Res., **3**, 27 (2001).
30. C. S. Lent, B. Isaksen: Clocked molecular quantum-dot cellular automata, IEEE Trans. Elect. Dev., **50**, 1890 (2003).
31. C. S. Lent, B. Isaksen, M. Lieberman: Molecular quantum-dot cellular automata, J. Am. Chem. Soc., **125**, 1056 (2003).
32. C. S. Lent, P. D. Tougaw: Device architecture for computing with quantum dots, Proc. IEEE **85**, 541 (1997).
33. W. Porod, C. S. Lent, G. H. Bernstein, A. O. Orlov, I. Amlani, G. L. Snider, J. L. Merz: Quantum-dot cellular automata: computing with coupled quantum dots, Intl. J. Elect., **86**, 549 (1999).
34. Y. Huang, X. Duan, Q. Wei, C. M. Lieber: Directed assembly of one-dimensional nanostructures into functional networks, Science, **291**, 630 (2001).
35. A. B. Greytak, L. J. Lauhon, M. S. Gudiksen, C. M. Lieber: Growth and transport properties of complementary germanium nanowire field-effect transistors, Appl. Phys. Lett., **84**, 4176 (2004).
36. Y. Cui, Z. Zhong, D. Wang, W. U. Wang, C. M. Lieber: High performance silicon nanowire field effect transistors, Nano Lett., **3**, 149 (2003).
37. H. Naeimi, A. DeHon: A greedy algorithm for tolerating defective crosspoints in nano PLA design, in *Proc. IEEE Intl. Conf. Field Prog. Tech.* (2004).
38. J. M. Rabaey, A. P. Chandrakasan, B. Nikolic: *Digital Integrated Circuits*, 2nd edn. (Prentice-Hall, Inc., Englewood Cliffs, NJ, 2002).
39. E. Thune, C. Strunk: Quantum transport in carbon nanotubes, in *Introducing Molecular Electronics*, ed. by G. Cuniberti, G. Fagas, K. Richter (Springer New York, 2005), Lecture Notes in Physics, **680**, 351 (2005).
40. J. Jortner, A. Nitzan, M. A. Ratner: Foundation of molecular electronics – charge transport in molecular conduction junctions, in *Introducing Molecular Electronics*, ed. by G. Cuniberti, G. Fagas, K. Richter (Springer New York, 2005), Lecture Notes in Physics, **680**, 13 (2005).
41. P. Hänggi, S. Kohler, J. Lehmann, M. Strass: AC-driven transport through molecular wires, in *Introducing Molecular Electronics*, ed. by G. Cuniberti, G. Fagas, K. Richter (Springer New York, 2005), Lecture Notes in Physics, **680**, 55 (2005).
42. R. DiFelice, A. Calzolari, D. Versano, A. Rubio: Electronic structure calculations for nanomolecular systems, in *Introducing Molecular Electronics*, ed. by G. Cuniberti, G. Fagas, K. Richter (Springer New York, 2005), Lecture Notes in Physics, **680**, 77 (2005).
43. K. Stokbro, J. Taylor, M. Brandbyge, H. Guo: Ab-initio nonequilibrium Green's function formalism for calculating electron transport in molecular devices, in *Introducing Molecular Electronics*, ed. by G. Cuniberti, G. Fagas, K. Richter (Springer New York, 2005), Lecture Notes in Physics, **680** 117 (2005).
44. A. DiCaarlo, A. Pecchia, L. Latessa, T. Frauenheim, G. Seifert: Tight-binding DFT for molecular electronics (gDFTB), in *Introducing Molecular Electronics*,

ed. by G. Cuniberti, G. Fagas, K. Richter (Springer New York, 2005), Lecture Notes in Physics, **680**, 153 (2005).

45. N. Bushong, M. DiVentra: Current-induced effects in nanoscale conductors, in *Introducing Molecular Electronics*, ed. by G. Cuniberti, G. Fagas, K. Richter (Springer New York, 2005), Lecture Notes in Physics, **680**, 185 (2005).
46. M. Wegewijs, M. H. Hettler, J. Konig, A. Thielmann, C. Romeike, K. Nowack: Single electron tunneling in small molecules, in *Introducing Molecular Electronics*, ed. by G. Cuniberti, G. Fagas, K. Richter (Springer New York, 2005), Lecture Notes in Physics, **680** 207 (2005).
47. M. Thorwart, R. Egger, M. Grifoni: Transport through intrinsic quantum dots in interacting carbon nanotubes, in *Introducing Molecular Electronics*, ed. by G. Cuniberti, G. Fagas, K. Richter (Springer New York, 2005), Lecture Notes in Physics, **680**, 229 (2005).
48. L. A. Bumm, J. J. Arnold, M. T. Cygan, T. D. Dunbar, T. P. Burgin, L. Jones, D. L. Allara, J. M. Tour, P. S. Weiss: Are single molecular wires conducting?, Science, **271**, 1705 (1996).
49. Y. Selzer, M. A. Cabassi, T. S. Mayer, D. L. Allara: Temperature effects on conduction through a molecular junction, Nanotechnology, **15**, S483 (2004).
50. R. M. Metzger: All about (*N*-hexadecylquinolin-4-ium-1-yl) methylidenetricyanoquinodimethanide, a unimolecular rectifier of electrical current, J. Mater. Chem., **10**, 55 (2000).
51. P. J. Kuekes, D. R. Stewart, R. S. Williams: The crossbar latch: Logic value storage, restoration, and inversion in crossbar circuits, J. Appl. Phys. **97** (2005).
52. S. J. Tans, A. R. M. Verschueren, C. Dekker: Room-temperature transistor based on a single carbon nanotube, Nature, **393**, 49 (1998).
53. H. W. C. Postma, T. Teepen, Z. Yao, M. Grifoni, C. Dekker: Carbon nanotube single-electron transistors at room temperature, Science, **293**, 76 (2001).
54. S. Heinze, J. Tersoff, P. Avouris: Carbon nanotube electronics and optoelectronics, in *Introducing Molecular Electronics*, ed. by G. Cuniberti, G. Fagas, K. Richter (Springer New York, 2005), Lecture Notes in Physics, **680**, 381 (2005).
55. J. C. Ellenbogen: Monomolecular electronic device, United States Patent 6,339,227, issued to the MITRE Corporation (2002).
56. J. Park, A. N. Pasupathy, J. I. Goldsmith, C. Chang, Y. Yaish, J. R. Petta, M. Rinkoski, J. P. Sethna, H. D. Abruna, P. L. McEuen, D. C. Ralph: Coulomb blockade and the Kondo effect in single-atom transistors, Nature **417**, 722 (2002).
57. W. Liang, M. P. Shores, M. Bockrath, J. R. Long, H. Park: Kondo resonance in a single-molecule transistor, Nature, **417**, 725 (2002).
58. J. L. Hennessy, D. A. Patterson: *Computer Architecture: A Quantitative Approach*, 3rd edn. (Morgan Kaufmann, San Mateo, CA, 2002).
59. J. R. Heath, P. J. Kuekes, G. S. Snider, R. S. Williams: A defect-tolerant computer architecture: Opportunities for nanotechnology, Science, **280**, 1716 (1998).
60. G. L. Snider, A. O. Orlov, I. Amlani, G. H. Bernstein, C. S. Lent, J. L. Merz, W. Porod: Quantum-dot cellular automata: Line and majority logic gate, Japanese J. Appl. Phys., Part 1 **38**, 7227 (1999).
61. V. P. Roychowdhury, D. B. Janes, S. Bandyopadhyay: Nanoelectronic architecture for Boolean logic, Proc. IEEE **85**, 574 (1997).

62. G. Toth, C. S. Lent, P. D. Tougaw, Y. Brazhnik, W. W. Weng, W. Porod, R. W. Liu, Y. F. Huang: Quantum cellular neural networks, Superlattices and Microstructures, **20**, 473 (1996).
63. O. Turel, J. H. Lee, X. Ma, K. K. Likharev: Neuromorphic architectures for nanoelectronic circuits, Int. J. Circ. Theor. Appl., **32**, 277 (2004).
64. C. P. Husband, S. M. Husband, J. S. Daniels, J. M. Tour: Logic and memory with nanocell circuits, IEEE Trans. Elect. Dev., **50**, 1865 (2003).
65. J. M. Tour, W. L. van Zandt, C. P. Husband, S. M. Husband, L. S. Wilson, P. D. Franzon, D. P. Nackashi: Nanocell logic gates for molecular computing, IEEE TNANO **1**, 100 (2002).
66. S. C. Goldstein, M. Budiu: Nanofabrics: Spatial computing using molecular electronics, in *Proc. Intl. Symp. Comp. Arch.* (2001).
67. M. R. Stan, P. D. Franzon, S. C. Goldstein, J. C. Lach, M. M. Ziegler: Molecular electronics: From devices and interconnect to circuits and architecture, Proc. IEEE **91**, 1940 (2003).
68. G. S. Rose, M. R. Stan: (2005), Programmable logic using molecular devices in a three-dimensional architecture, presentation at the Engr. Intl. Conf. on Mol. Elect., San Diego, CA (unpublished).
69. K. K. Likharev, D. B. Strukov: CMOL: Devices, circuits, and architectures, in *Introducing Molecular Electronics*, ed. by G. Cuniberti, G. Fagas, K. Richter (Springer New York, 2005), Lecture Notes in Physics, **680**, 447 (2005).
70. D. B. Strukov, K. K. Likharev: CMOL FPGA: A reconfigurable architecture for hybrid digital circuits with two-terminal nano devices, Nanotechnology, **16**, 888 (2005).
71. J. Han: Fault-tolerant architectures for nanoelectronic and quantum devices, Ph.D. thesis, Delft University of Technology, Delft, The Netherlands (2004).
72. V. Beiu: A novel highly reliable low-power nano architecture when von Neumann augments Kolmogorov, in *Proc. IEEE Intl. Conf. on App. Spec. Sys., Arch. and Proc. (ASAP)* (2004).
73. S. Trimberger (Ed.): *Field Programmable Gate Array Technology* (Kluwer Academic Publishers, Boston, 1994).
74. M. D. Austin, H. Ge, W. Wu, M. Li, Z. Yu, D. Wasserman, S. A. Lyon, S. Y. Chou: Fabrication of 5 nm linewidth and 14 nm pitch features by nanoimprint lithography, Appl. Phys. Lett., **84**, 5299 (2004).
75. G. Y. Jung, S. Ganapathiappan, D. A. A. Ohlberg, D. L. Olynick, Y. Chen, W. M. Tong, R. S. Williams: Fabrication of a 34 × 34 crossbar structure at 50 nm half-pitch by UV-based nanoimprint lithography, Nano Lett., **4**, 1225 (2004).
76. Z. Zhong, D. Wang, Y. Cui, M. W. Bockrath, C. M. Lieber: Nanowire crossbar arrays as address decoders for integrated nanosystems, Science, **302**, 1377 (2003).
77. A. DeHon: Reconfigurable architectures for general-purpose computing, Technical Report AITR-1586, Massachusetts Institute of Technology (1996).
78. P. S. Zuchowski, C. B. Reynolds, R. J. Grupp, S. G. Davis, B. Cremen, B. Troxel: A hybrid ASIC and FPGA architecture, in *Proc. Intl. Conf. Comp. Aid. Des.* pp. 187 (2002).
79. M. M. Ziegler, M. R. Stan: CMOS/nano co-design for crossbar-based molecular electronic systems, IEEE TNANO (2003).
80. M. Forshaw, R. Stadler, D. Crawley, K. Nikolic: A short review of nanoelectronic architectures, Nanotechnology, **15**, S220 (2004).

81. R. P. McConnell: Diode-based power gain for molecular-scale electronic digital computers, report MP 00W0000310, The MITRE Corporation, McLean, VA (2000).
82. R. P. McConnell, J. C. Ellenbogen, T. S. Mayer, T. E. Mallouk, S. P. Goldstein: Requirements and designs for molecular computer architectures that incorporate gain-producing elements, presentation at the Engr. Found. Conf. on Mol. Elect., Kona, HI (unpublished) (2000).
83. S. C. Goldstein, D. Rosewater: Digital logic using molecular electronics, in *Proc. Intl. Sol. St. Circ. Conf.* (2002).
84. G. S. Rose, M. R. Stan: Memory arrays based on molecular RTD devices, in *Proc. IEEE-NANO* pp. 453, (2003).
85. E. Goto, K. Murata, K. Nakazawa, K. Nakagawa, T. Moto-Oka, Y. Matsuoka, Y. Ishibashi, T. Soma, , E. Wada: Esaki diode high speed logical circuits, IRE Trans. Elect. Comp. pp. 25, (1960).
86. H. C. Liu, T. C. L. G. Sollner: High-frequency resonant-tunneling devices, in *Semiconductors and Semimetals*, ed. by R. A. Kiehl, T. C. L. G. Sollner, **41** (Academic Press, Boston, 1994), pp. 359.
87. R. H. Mathews, J. P. Sage, T. C. L. G. Sollner, S. D. Calawa, C.-L. Chen, L. J. Mahoney, P. A. Maki, K. M. Molvar: A new RTD-FET logic family, Proc. IEEE **87**, 596 (1999).
88. P. J. Kuekes: Molecular crossbar latch, United States Patent 6,586,965 (2003).
89. Y. Chen, D. A. A. Ohlberg, X. Li, D. R. Stewart, J. O. Jeppesen, K. A. Nielsen, J. F. Stoddart, D. L. Olynick, E. Anderson: Nanoscale molecular-switch devices fabricated by imprint lithography, Appl. Phys. Lett., **82**, 1610 (2003).
90. G. Snider, P. Kuekes, T. Hogg, R. Stanley Williams: Nanoelectronic Architectures, Appl. Phys., A **80**, 1183 (2005).
91. M. van den Brink: Litho roadmap shows difficult terrain – part 2 – technology information, *Electronic News*, Jan. 17, 2000.
92. Semi industry to reach $360 billion by 2010, says report, *Silicon Strategies*, Dec. 2, 2003.
93. A. W. Burks, H. H. Goldstine, J. von Neumann: Preliminary discussion of the logical design of an electronic computing instrument, in *John von Neumann Collected Works*, ed. by A. H. Taub, Vol. V (The Macmillan Co., New York, 1963), pp. 34.
94. H. H. Goldstine, J. von Neumann: On the principles of large scale computing machines, in *John von Neumann Collected Works*, ed. by A. H. Taub, Vol. V (The Macmillan Co., New York, 1963), pp. 1.
95. J. von Neumann: First draft of a report on the EDVAC, in *From ENIAC to Univac: An Appraisal of the Eckert-Mauchly Computers*, ed. by N. Stern (Digital Press, Bedford, MA, 1981).
96. Reprinted from the AMD Virtual Pressroom at http://www.amd.com.
97. Semiconductor Industry Association: International technology roadmap for semiconductors: 2003 edition, Technical report, SEMATECH (2003).
98. J. D. Meindl, Q. Chen, J. A. Davis: Limits on silicon nanoelectronics for terascale integration, Science, **293**, 2044 (2001).
99. V. V. Zhirnov, R. K. Cavin, J. A. Hutchby, G. I. Bourianoff: Limits to binary logic switch scaling – a gedanken model, Proc. IEEE **91**, 1934 (2003).

100. D. J. Frank, R. H. Dennard, E. Nowak, P. M. Solomon, Y. Taur, H. S. P. Wong: Device scaling limits of Si MOSFETs and their application dependencies, Proc. IEEE **89**, 259 (2001). This article is one of several that appeared in a Special Issue on Limits of Semiconductor Technology.
101. M. T. Bohr: Nanotechnology goals and challenges for electronic applications, IEEE TNANO **1**, 56 (2002).
102. J. A. Davis, R. Venkatesan, A. Kaloyeros, M. Beylansky, S. J. Souri, K. Banerjee, K. C. Saraswat, A. Rahman, R. Reif, J. D. Meindl: Interconnect limits on gigascale integration (GSI) in the 21st century, Proc. IEEE **89**, 305 (2001). This article is one of several that appeared in a Special Issue on Limits of Semiconductor Technology.
103. M. Ieong, B. Doris, J. Kedzierski, K. Rim, M. Yang: Silicon device scaling to the sub-10-nm regime, Science, **306**, 2057 (2004).
104. A. Rahman: System-level performance evaluation of three-dimensional integrated circuits, Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, MA (2001).
105. R. Reif, A. Fan, K.-N. Chen, S. Das: Fabrication technologies for three-dimensional integrated circuits, in *Proc. Intl. Symp. Qual. Elect. Des.* pp. 33, (2002).
106. S. F. Al-Sarawi, D. Abbott, P. D. Franzon: A review of 3-D packaging technology, IEEE Trans. CPMT B **21**, 2 (1998).
107. A. Fan, A. Rahman, R. Reif: Copper wafer bonding, Elect. Sol. St. Lett., **2**, 534 (1999).
108. J. A. Burns, C. Keast, K. Warner, P. Wyatt, D. Yost: Fabrication of 3-dimensional integrated circuits by layer transfer of fully depleted SOI circuits, in *Proc. Mat. Res. Soc. Symp. G* **768** (2003).
109. Y. Kwon, A. Jindal, J. J. McMahon, J.-Q. Lu, R. J. Gutmann, T. S. Cale: Dielectric glue wafer bonding for 3-D ICs, in *Proc. Mat. Res. Soc.* (Spring 2003).
110. L. Xue, C. C. Liu, H. S. Kim, S. Kim, S. Tiwari: Three-dimensional integration: Technology, use, and issues for mixed-signal applications, IEEE Trans. Elect. Dev., **50**, 601 (2003).
111. V. Subramanian, P. Dankoski, L. Degertekin, B. T. Khuri-Yakub, K. C. Saraswat: Controlled two-step solid-phase crystallization for high-performance polysilicon TFT's, IEEE Elect. Dev. Lett., **18**, 378 (1997).
112. S. Das: Design automation and analysis of three-dimensional integrated circuits, Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, MA (2004).
113. R. Chau, B. Boyanov, B. Doyle, M. Doczy, S. Datta, S. Hareland, D. Jin, J. Kavalieros, M. Metz: Silicon nanotransistors for logic applications, Physica E: Low-dimensional systems and nanostructures **19**, 1 (2003).
114. S. J. Wind, J. Appenzeller, R. Martel, V. Derycke, P. Avouris: Vertical scaling of carbon nanotube field-effect transistors using top gate electrodes, Appl. Phys. Lett., **80**, 3817 (2002).
115. Z. M. Liu, A. A. Yasser, J. S. Lindsey, D. F. Bocian: Molecular memories that survive silicon device processing and real-world operation, Science **302**, 1543 (2003).
116. C. Li, W. Fan, B. Lei, D. Zhang, S. Han, T. Tang, X. Liu, Z. Liu, S. Asano, M. Meyyappan, J. Han, C. Zhou: Multilevel memory based on molecular devices, Appl. Phys. Lett., **84**, 1949 (2004).

117. Q. L. Li, S. Surthi, G. Mathur, S. Gowda, Q. Zhao, T. A. Sorenson, R. C. Tenent, K. Muthukumaran, J. S. Lindsey, V. Misra: Multiple-bit storage properties of porphyrin monolayers on SiO_2, Appl. Phys. Lett., **85**, 1829 (2004).

118. B. J. Feder: Nanotech memory chips might soon be a reality, *New York Times*, June 7, 2004.

119. E. Kusse: Analysis and circuit design for low power programmable logic modules, M.S. thesis, Univ. of California, Berkeley, CA (1997).

120. A. S. Blum, C. M. Soto, C. D. Wilson, J. D. Cole, M. Kim, B. Gnade, A. Chatterji, W. F. Ochoa, T. W. Lin, J. E. Johnson, B. R. Ratna: Cowpea mosaic virus as a scaffold for 3-D patterning of gold nanoparticles, Nano Lett., **4**, 867 (2004).

121. J. Y. Fang, C. M. Soto, T. W. Lin, J. E. Johnson, B. Ratna: Complex pattern formation by cowpea mosaic virus nanoparticles, Langmuir, **18**, 308 (2002).

122. P. Beckett, A. Jennings: Towards nanocomputer architecture, in *Proc. ACS Conf. Res. Prac. Inf. Tech.* **6**, 141 (2002).

123. N. Weste, K. Eshraghian: *Principles of CMOS VLSI Design*, 2nd edn. (Addison-Wesley Publishing Company, Reading, MA, 1994).

124. Cadence Design Framework II, Version IC 5.0.33, Cadence Design Systems, Inc., San Jose, CA, 2004.