This chapter appears in the CRC Nano and Molecular Electronics Handbook, May 2007.

System-Level Design and Simulation of Nanomemories and Nanoprocessors

Shamik Das,[a] Carl A. Picconatto,[a] Garrett S. Rose,[a,b]
Matthew M. Ziegler,[c] and James C. Ellenbogen[a]

(a) Nanosystems Group, The MITRE Corporation, McLean, VA, USA
(b) Department of Electrical and Computer Engineering,
Polytechnic University, Brooklyn, NY, USA
(c) IBM T. J. Watson Research Center, Yorktown Heights, NY, USA

October 9, 2006

Abstract

This paper describes in detail system designs and system simulations for electronic nanocomputers that are integrated on the molecular scale. These systems are considered here as consisting primarily of the combination of two component subsystems, nanomemories and nanoprocessors. Challenges are enumerated for the design and development of both of these ultra-densely integrated components. Various system-level designs or architectures are presented that have been proposed to meet these challenges. Detailed consideration is given for both nanomemories and nanoprocessors to system designs that are based upon arrays of crossed nanowires. In each case, a system simulation is performed to assess and to help optimize the prospective performance of the system component in advance of its fabrication. In the ongoing development of crossbar nanocomputer systems, these simulations have been integral to the refinement of designs because they assist in reducing the time and cost of such development.

1 Introduction

Much progress has been made recently in the field of molecular electronics. In particular, dramatic successes in the demonstration of nanoelectronic devices and simple molecular-scale circuits [1–11] suggest that soon we may be able to design, fabricate, and demonstrate an entire, ultra-dense nanoelectronic computer that is integrated on the molecular scale. In fact, development of such nanoelectronic computer systems already is underway. Despite significant challenges, this effort has produced functioning prototype nanomemories [12–14] and is likely to produce functioning prototype nanoprocessors within a few years [2, 8, 15–18].

In this paper, we describe system designs [16, 19, 20] and system simulations [21, 22] that have been and continue to be integral to those advances in nanoelectronic system hardware development. That design and simulation work has focused on approaches for novel, ultra-dense nanoelectronic circuits and systems that use crossed nanowire arrays [8, 23–25] as their underlying circuit structures. Such arrays may be fabricated either from patterned nanowires [8, 24, 25] or from self-assembled nanowires [23]. It is expected that operational nanomemory and nanoprocessor systems based upon such crossed-nanowire array structures can achieve integration densities in excess of 10^{11} devices per square centimeter [26]. This is well beyond the densities presently envisioned [27] for electronic computer systems that use circuits based upon complementary metal-oxide-semiconductor (CMOS) devices, as do conventional microelectronic computers. Thus, higher-density nanomemory and nanoprocessor systems designed and fabricated from crossed nanowires might even be used to enhance CMOS-based electronics in a post-CMOS era. Here, however, we focus on the design and simulation of "pure" or "true" nanomemory and nanoprocessor systems that incorporate only nanometer-scale devices, such as crossed nanowires and molecules.

The process of developing such true nanocomputers opens up an entirely new frontier of systems objectives and issues that require research and development, beyond that in the much more numerous investigations that presently are being conducted upon isolated nanodevices and small nanocircuits [28–32]. Thus, in addition to describing our specific design and simulation investigations of crossed-nanowire nanocomputer systems, we also discuss the broader range of system issues that are being encountered at this new frontier. Further, we survey some of the other device and system design approaches [33–46] that are being advanced to help address the problem of building entire nanomemory and nanoprocessor systems that are 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 [4, 5, 8, 10, 12, 14, 47]. On the other hand, the systems may not use molecules at all, employing instead solid-state quantum dots [35, 38, 48–50] and/or patterned or self-assembled nanowires [8, 23, 24, 51, 52], as mentioned above.

Consideration of the range of topics described above in this introductory section proceeds below as follows:

- In Section 2 of this work, we discuss the electrical behaviors that are required of molecular-scale devices in order to develop extended nanoelectronic systems. We describe the presently-available nanoelectronic devices that exhibit and yield the types of behavior necessary for computation.

- Section 3 considers the prospective performance of a crossbar-based nanomemory system that utilizes some of the nanoelectronic devices described in Section 2. An overview is given of the architecture and the operational principles of this system. Then, metrics and a simulation methodology for the evaluation of system performance are described. This unique, bottom-up simulation methodology facilitates the detailed prediction of the performance of entire systems integrated on the molecular scale. Specific system simulation results are provided, followed by a discussion of the implications of these results for the construction of extended nanomemory systems.

- Building upon this analysis of nanomemories, Section 4 considers the more complex problem of nanoprocessor design. It begins with a review of the difficulties facing the design of nanoprocessor architectures, and it surveys the various system architecture approaches that have been proposed. Detailed consideration is given to one promising system-level design approach, a crossed-nanowire approach due to DeHon and Wilson [19]. Section 4 concludes by describing a detailed simulation of key circuits of a notional nanoprocessor based upon the DeHon-Wilson design approach.

The simulations described here are intended to illustrate in a very specific manner the types of issues that will be encountered in building and operating a nanocomputer. It is significant that these simulations can be and have been conducted well before an entire system of this type actually is fabricated and integrated on the molecular scale. 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* device-level 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.

2 Molecular Scale Devices in Device-Driven Nanocomputer Design[1]

Whether one considers the design, the simulation, or the fabrication of an entire computer system, there is a hierarchy of structure and function. In the usual approach of modern electrical engineering, this hierarchy

[1]Some of the material in this section has appeared previously in Das *et al.*, "Architectures and simulations for nanoprocessor systems integrated on the molecular scale," *Lect. Notes Phys.*, vol. 680, pp. 479–513, 2005.

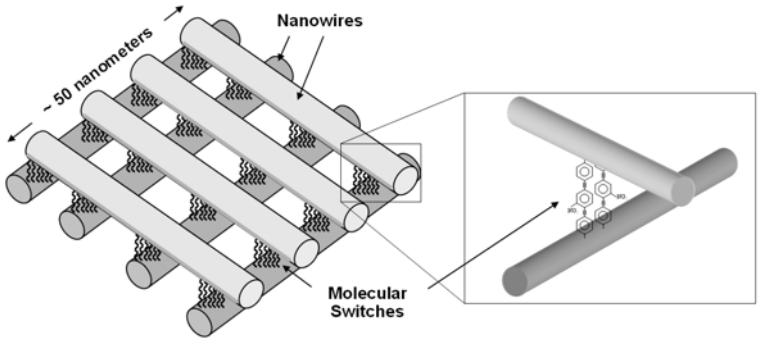


Figure 1: “Crossbar” array of nanowires with molecular devices at junctions.

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 [53]. 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.

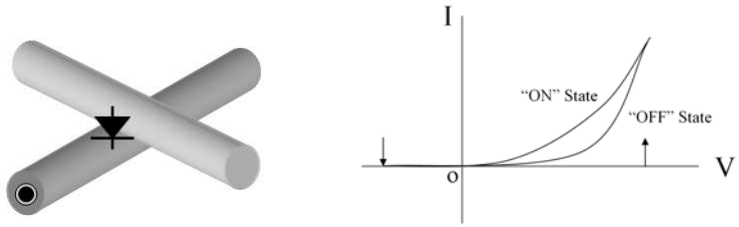
At present, the situation is different when one sets out to design, simulate, or fabricate an entire nanocomputer system integrated on the molecular scale. The ability to tune the performance of nanodevices still is limited. This is partly because these molecular-scale devices are so new. Thus, the experiments [54–58] and the theory [59–66] necessary to understand them, design them, and make them to order still are very much in development. In addition, the ability to tune precisely the structure and performance of nanometer-scale devices may be limited inherently by the quantization of those structures and 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 nanocomputer system. Such a system will be at least tens of square micrometers, 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 nanocomputer systems at the bottom-most level of the hierarchy, the device level.

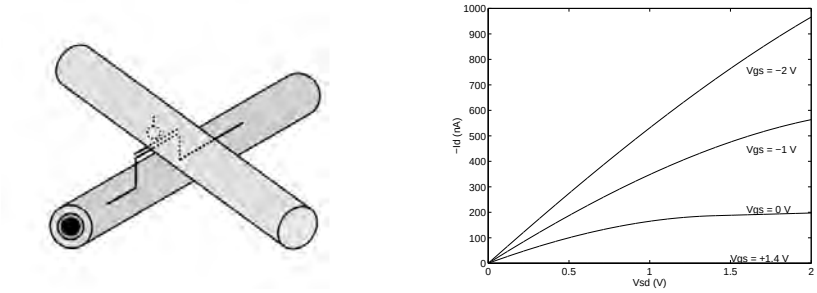
As is true in most experiments on the electrical properties of molecules [3, 55, 56, 67, 68], for the purposes of discussing circuits and systems, a molecular-scale device consists of a junction between two metal or semiconductor surfaces with a molecular-scale structure sandwiched between. This molecular-scale structure may be one or a few molecules, as depicted in Fig. 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 [69], rectification [57, 70], negative differential resistance (NDR) [6] and hysteresis [69]. A variety of such junction nanodevices have been realized that might be useful for building extended nanoelectronic systems. The hysteretic behavior illustrated



(a) (b)

Figure 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.



(a) (b)

Figure 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 Huang *et al.* [7]. 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.

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 computer systems. For the logic components of such systems, i.e., nanoprocessors, it is of particular importance to have nanoscale switches that can be used to produce signal restoration and gain. These two features are essential to maintaining electrical signals as they move through multiple levels of logic. Nanoscale switches that produce signal restoration and gain likely would be implemented using nanotransistors, although small circuits, e.g., latches incorporating molecular diodes, also can produce signal restoration [11]. Nanotransistors have been fabricated using carbon nanotubes (CNTs) [9, 71–74], although it remains very difficult to use them in building extended systems. There also have been some suggestions for fabricating transistors from smaller molecules [57, 75]. A few individual molecular transistors have been demonstrated based on small molecules, but only in very sensitive experiments under cryogenic conditions [76, 77]. On the other hand, robust nanoscale transistors built from crossed nanowires have been demonstrated in a number of experiments at room temperature [7].[2] A diagram of such a nanowire nanotransistor is displayed alongside models of its I-V curves in Fig. 3.

In addition to obtaining gain and signal restoration, other I-V behaviors, such as rectification from two-

[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. 2, which is a junction nanoswitch.

terminal nanodevices, are very important. Simulations show that even when using devices that provide good gain, rectification is necessary to ensure that signals do not take unintended 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 nanocomputer system that integrates two-terminal junction nanodevices, such as rectifiers, as well as three-terminal nanotransistors. These steps form a hierarchy from the device to the system level, as follows: (a) development of nanofabrication approaches to build large numbers of the requisite junction nanodevices 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 circuit designs and that can accommodate the limitations imposed by the constrained I-V behaviors available in present-day molecular electronic devices.

Challenges exist at each level of this hierarchy. However, these challenges may be mitigated by considering the development of a nanocomputer system separately for each of its two primary component systems, nanomemories and nanoprocessors. Nanomemory systems present fewer challenges by virtue of their less complex system architecture. Also, the lessons learned in designing and simulating nanomemory systems, as discussed in the next section, provide a foundation for addressing in Section 4 the more numerous and severe challenges of nanoprocessor design and simulation.

3 Crossbar-Based Design for Nanomemory Systems[3]

3.1 Overview

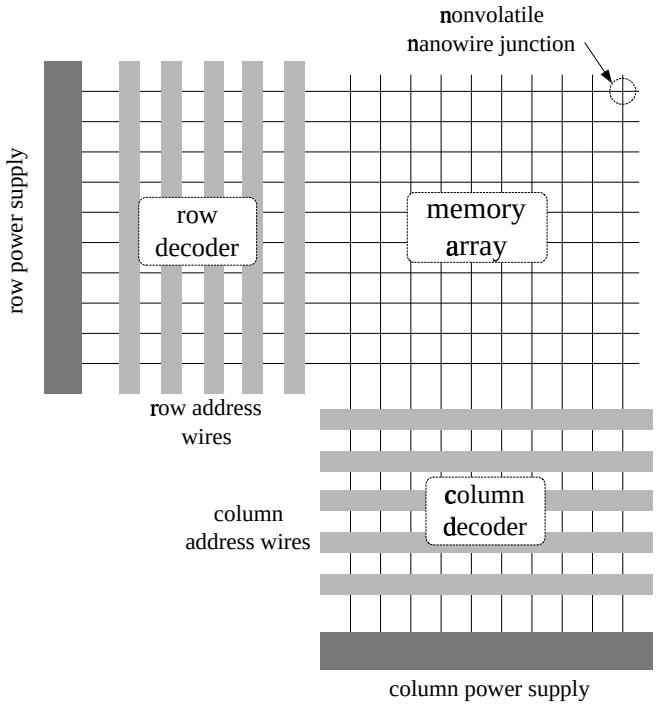
The crossbar architecture [10, 12, 14, 16, 25, 69, 78] is the most prevalent framework or approach now being employed for the design and fabrication of nanomemory systems that are integrated on the molecular scale. The basic crossbar architecture consists of the combination of planes of parallel wires that are laid out in orthogonal directions, such as is shown in Fig. 1. The fundamental devices for memory storage are the molecular-scale junction switches formed at the crosspoints of the wires. For the purposes of this work, it is assumed that one bit is stored in each such fundamental crosspoint device.

A nanomemory system design based upon this architecture consists of three major subsystems: a nanowire crossbar memory array and two decoders, one for the array rows and one for the columns. Ultra-dense arrays of crossed nanowires are fabricated using specialized techniques such as nanoimprinting [8, 24] and flow-based alignment [23]. Reasonably large memory arrays have been constructed using these techniques [12, 14].

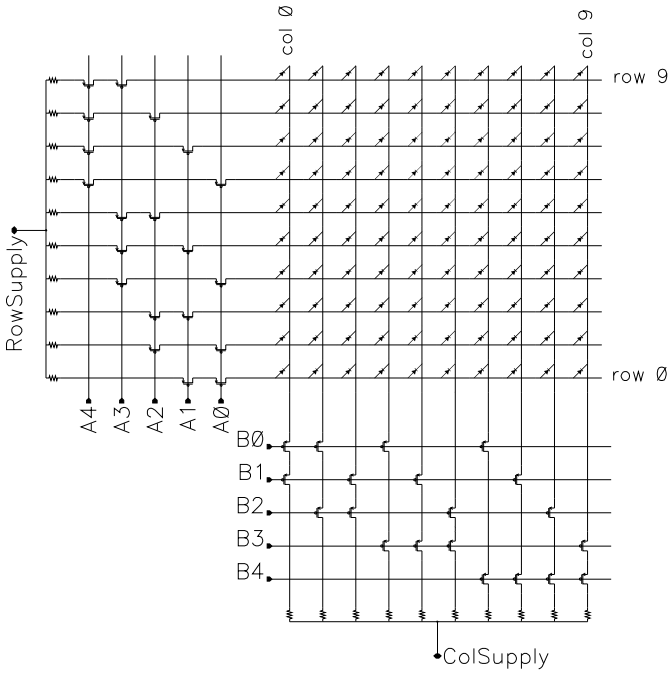
Fig. 4 shows a system diagram and a corresponding circuit schematic of a notional 10×10 nanomemory based on the architectural design by DeHon [79]. In Fig. 4, the nanowires forming the crossbar array are represented by thin black lines, as are the nanowires in the decoders. The decoders also contain much longer and much thicker micrometer-scale wires or "microwires" of the type used in conventional microelectronics. These are represented by thick gray lines. The crossbar array stores the data, whereas the decoders serve as an interface to this nanomemory array. The decoders permit an external microelectronic system to access a unique crossbar junction within the densely-integrated array. In addition, each decoder is connected to a microwire that supplies power to the system. These power supply lines are represented by thick black lines. They also serve to read or write a bit to the individual nanowire junction selected by the decoders, by imposing a voltage upon it.

Variations have been proposed for the nanomemory system design depicted in Fig. 4. For example, Strukov and Likharev propose a "hybrid" nanomemory architecture [80] that utilizes nanowire crossbars for storage, but places these crossbars on top of a decoder structure that is fabricated entirely in conventional CMOS circuitry. Another example is provided by Nantero Corporation, which has demonstrated prototype nanomemories using an altogether different crossbar composition [13, 81]. In the Nantero crossbar, one plane of wires is constructed using CMOS technology, while the other, orthogonal plane is created from a mesh of carbon nanotubes.

[3] Some of the material in this section has appeared previously in Ziegler *et al.*, "Scalability simulations for nanomemory systems integrated on the molecular scale," *Ann. N.Y. Acad. Sci.*, vol. 1006, pp. 312–330, 2003.



(a)



(b)

Figure 4: A sketch of (a) the structure and (b) a circuit schematic for a nanomemory design. This design consists of a crossbar nanowire memory array composed of nonvolatile nanowire diodes, plus two decoders composed of top-gated nanowire field-effect transistors.

Several switch options have been proposed to store individual memory bits at the crosspoints of the ultra-dense nanowire arrays. For example, in the Lieber-DeHon nanomemory system [78], each nanowire-nanowire crosspoint in the crossbar array forms a bistable, nonvolatile nanowire (NVNW) diode, as depicted in Fig. 2. In the Heath nanomemory system [10], a monolayer of bistable rotaxane molecules serves as an electronically rewritable memory bit [5]. Similarly, the system design of the Hewlett-Packard corporation relies on a monolayer of 100 to 1,000 organic molecules sandwiched between the inorganic contacts at each crosspoint [82]. The hybrid nanomemory proposed by Strukov and Likharev would employ single-electron latching switches [83]. In the Nantero nanomemory architecture, the carbon nanotubes that form part of the crossed-wire array also serve as the fundamental devices – individual memory bits are stored electromechanically by introducing reversible deflections or "kinks" at the desired crosspoints in the nanomemory.

A different set of molecular-scale devices is required to synthesize the decoder circuits that provide the interface to this storage array [84]. The Lieber-DeHon [85] and Heath [86] nanomemory architectures employ transistors that utilize semiconducting nanowires as their channels. Microwires, which gate these channels, are used to connect to the nanomemory from the microscale. Alternatively, Hewlett-Packard proposes a scheme in which the nanowire-microwire interface is generated stochastically by the random deposition of gold colloidal nanodots between the microwires and nanowires [87]. Through proper control of the deposition process, decoding of each of the individual nanowires in the nanomemory can be achieved with high probability. In contrast to the Lieber-DeHon and Hewlett-Packard approaches, the hybrid systems of Strukov and Likharev [80] and of Nantero Corporation [13, 81] employ conventional CMOS in the decoder circuits.

For any of these nanomemory designs, it is costly, time-consuming, and difficult experimentally to determine whether such a nanomemory system will function correctly. In fact, for most of the architectures proposed for nanomemory systems, fabrication and physical testing have yet to be carried out. Thus, to shorten the design cycle and reduce costs, it is desirable to conduct full-system simulation of these nanomemory system designs before they are fabricated.

3.2 Simulation of an Example Nanomemory System

In this section, we describe simulations of a notional nanomemory system based upon the Lieber-DeHon architecture [79]. The fundamental devices [51, 52, 88] and small prototype circuits [23, 85] of this nanomemory already have been demonstrated experimentally. Here, we utilize computer simulation to evaluate how extended system prototypes might perform if built using the same devices.

In the system simulation described here, the nanomemory storage array consists of nanowire diodes. Within the decoders, the microwire-nanowire crosspoints form field-effect transistors. These transistors permit the selection, or "addressing," of individual rows and columns in the memory array. The transistors are organized in a "2-hot" coding scheme [79]. The 2-hot scheme requires asserting a voltage on exactly two microwires in a decoder in order to select a unique nanomemory location, no matter the size of the storage array. This coding scheme differs from the binary schemes typically used in CMOS circuitry [53]. Binary coding would require asserting $\log_2 N$ microwires to select a unique wire from a set of N wires. The 2-hot addressing scheme is chosen for its additional defect tolerance. With 2-hot addressing, any failure of a single microwire impacts significantly fewer bits than a comparable failure in a binary scheme [79]. In addition, the 2-hot coding scheme requires the selected nanowire to encounter exactly two transistors in series, regardless of the size of the array, whereas binary coding would meet $\log_2 N$ transistors in series. Reducing the number of transistors in series is beneficial because it ensures that an ample amount of the supply voltage reaches the selected crossbar junction, rather than being dissipated by the decoders.

3.3 Nanomemory System Evaluation Metrics

First and foremost, we evaluate the ability of a nanomemory to read and write information accurately, with strong signals that are not easily lost in circuit noise or prone to other sources of error. To evaluate read operations, we focus on the output current differences (ΔI_{out}) between reading logic "1" and reading logic "0." This current difference is evaluated for the worst-case memory configuration (i.e., the worst-case pattern of

"1"s and "0"s in the array) in order to ensure that a logic "1" can be distinguished from a logic "0" for each bit in every configuration of the entire nanomemory array. For write operations, we examine the voltage applied to the desired crossbar junction in order to verify that sufficient voltage is being applied. At the same time, we verify that no other junction receives enough voltage to alter its logical state.

Consideration also is given here to the nanomemory speed and power consumption. Output current switching times are analyzed, and methods are suggested to improve speed and reduce power.

3.4 Simulation Methodology and Device Modeling

The simulation of devices and complex circuit systems can be performed at a number of different levels of design abstraction [53]. The appropriate level of abstraction is determined by the goals of the simulation and by the ability of the simulation tools to handle complexity. Often, it is necessary to neglect levels involving fine details in order to capture the important overall behaviors of complex systems.

There are three categories of electronic design abstraction: the device level, the circuit level, and the architectural level. The device level focuses on a single device, e.g., a diode or transistor, in great detail. Simulations at this level provide information about the operation and physics of individual devices, but generally do not consider the interactions among distinct devices in a circuit. In contrast, the architectural level considers very large systems, but typically does not include the physics or the behavior of individual devices. The circuit level bridges these two approaches and considers relatively large systems (on the order of tens of thousands of devices), while still retaining a connection to the underlying physical behavior. The simulations described here take place at this level.

Many concepts and techniques from conventional microelectronics are borrowed here for use in simulating nanoelectronic memories. For example, the commonly-utilized commercial Cadence Spectre VLSI CAD software tool [89] is our primary simulation program. One reason for applying such commercial off-the-shelf software tools from the microelectronics industry is the obvious timesaving and reliability associated with the use of readily available, well-tested software. This software also incorporates powerful features, such as modeling languages and graphics, developed specifically for the flexible modeling of extended circuitry. Finally, the use of conventional VLSI tools provides a seamless approach to the design and simulation of the nanomemory together with the peripheral microelectronic circuitry required for operation and communication with the outside world [90, 91].

The work presented here also relies heavily on the conventional microelectronic concept of the device model, which captures the essential properties and response behavior of a circuit element. Models of experimentally observed behavior are required for all of the devices utilized in the nanomemory. In particular, the current-voltage transfer characteristics (I-V curves) are necessary for the simulation of steady-state behavior, and the capacitance-voltage transfer characteristics (C-V curves) are required for time-varying, or transient, simulation.

Typical models for microelectronic devices consist of compact equations based upon the well-understood, underlying physics of such devices. However, this physics-based approach is not workable, at present, for simulations involving molecular-scale devices, because the fundamental physics of most molecular-scale devices is not well understood. Thus, in this work, we utilize empirical models based on measured device characteristics.

Incorporating new models into conventional circuit simulators can be difficult. The addition of a new model often can require modifying proprietary source code. Open-source simulators do exist, such as SPICE3 [92], but adding new device models to these simulators is tedious [93]. Furthermore, these open-source simulators lack the robustness and simulation speed that are necessary to model large circuit systems and that are found in many commercial simulators.

Thus, to develop and simulate efficiently models for molecular-scale devices, we utilized the commercial Cadence Spectre simulator. This software permits the description of the empirical behavior of devices using the analog hardware description language (analog HDL) Verilog-A. This modeling approach is similar to one described elsewhere [94–96], except that the empirical equations derived in this work were tailored to the devices employed in the Lieber-DeHon nanomemory system. These empirical equations were incorporated

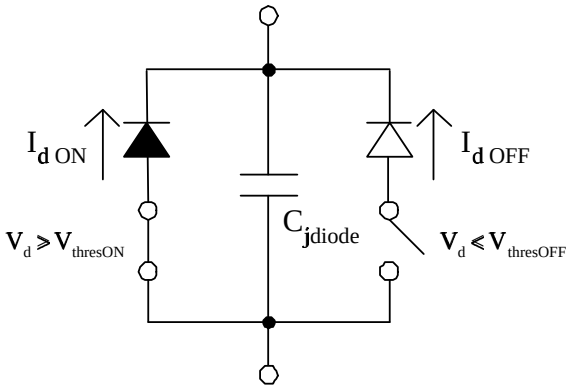


Figure 5: A sketch of a circuit schematic for the nonvolatile nanowire diode shown in Fig. 2(a). The device model consists of two conventional diodes in parallel with a capacitor. The two individual diodes model the high current state (on-state) and low current state (off-state) respectively.

into the Spectre circuit simulator, which supports co-simulation of both Verilog-A components and conventional SPICE-level devices.

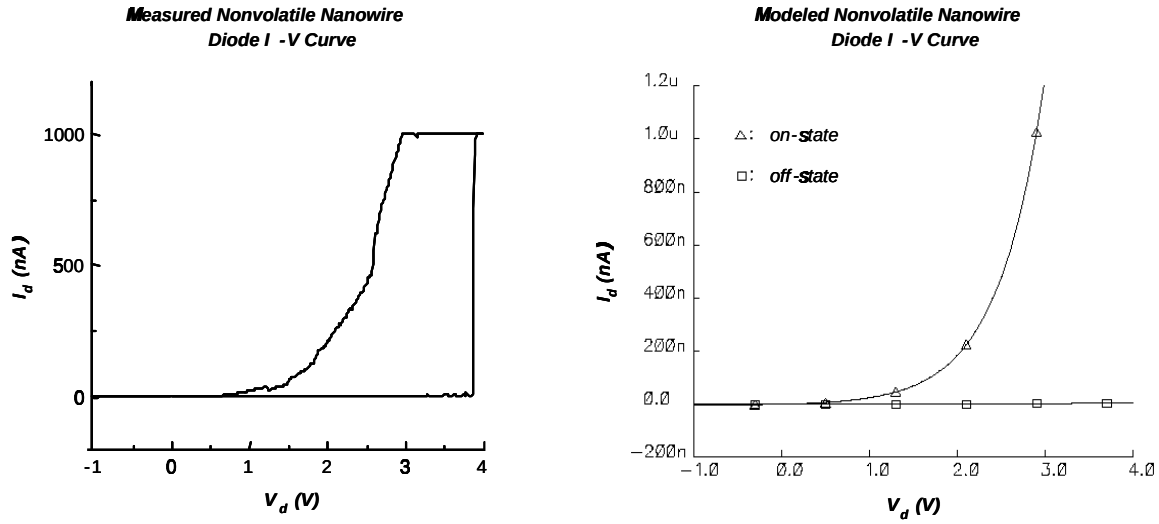
Simulations of the crossbar nanomemory system required three device models. The first two are models of nanowire devices: the nonvolatile nanowire (NVNW) diodes used in the storage array and the top-gated nanowire field-effect transistors (TGNW-FETs) used in the decoders. The third is a model of the nanowire interconnects of the nanomemory system.

3.4.1 Nonvolatile Nanowire (NVNW) Diode Model

Fig. 5 shows a schematic diagram of a circuit that models the behavior of the NVNW diodes developed at Harvard University [78]. The model consists of two conventional, non-hysteretic diodes connected in parallel with a capacitor (C_{jdiode}). The model can be switched between a high current state (on-state) and a low current state (off-state) by switching which diode is connected to the circuit. This reproduces the hysteretic I-V behavior seen in the experimental device. The measured I-V characteristics of the actual NVNW diodes and the corresponding model I-V curve are shown in Fig. 6. The measured I-V curves were fitted to empirical equations to produce the model. The apparatus used to collect the experimental data shown in Fig. 6 was limited to measuring currents of up to 1,000 nA, a limit that the device attains at a bias voltage of approximately 3 V. In the model, values for the current passing through the diode at bias voltages greater than 3 V were extrapolated from the available data.

The NVNW diode switches from the on-state to the off-state when a reverse bias voltage that is more negative than $V_{thresOFF}$ is applied across the device. In a similar fashion, a bias voltage greater (i.e., more positive) than $V_{thresON}$ switches the device from the off-state to the on-state. Device threshold values for the experimental diodes are -2.75 V and 3.80 V for $V_{thresOFF}$ and $V_{thresON}$, respectively. One issue for this simulation research is whether device characteristics, such as the threshold voltages, are optimal from the perspective of designing and building an extended memory circuit system, and whether this device behavior might be improved for that purpose. This question is addressed in Section 3.5.

Although this diode switch appears to exhibit relatively simple behavior, the hysteretic I-V curve creates a complicated device modeling task. A smooth transition between curves occurs when switching from the on-state to the off-state at $V_{thresOFF}$, but the device experiences an abrupt jump in current when switching from the off-state to the on-state at $V_{thresON}$. This discontinuity in the current requires special provisions in the mathematical models used in the simulation. We avoid any possible difficulties at the discontinuity by simply recording when $V_{thresON}$ has been surpassed, without actually changing the underlying state of the device. This is sufficient for the purposes of the work describe here, because the memory array is simulated for only one configuration at a time. Thus, it is necessary only to determine which of its constituent diodes has crossed



(a) (b)

Figure 6: Hysteretic I-V curve for the nonvolatile nanowire diode. (a) The measured I-V curve for an experimentally fabricated nonvolatile nanowire diode. (b) The simulated I-V curve for the nonvolatile nanowire diode model.

its switching threshold. Subsequent analysis of the nanomemory system with the diodes in the switched state is not required.

This technique is not suitable in all situations. Multiconfiguration simulations, such as those that calculate power consumption during write-read combinations, require simulation of diode transitions between the ON and OFF states. This cannot be modeled with the methodology described above. Nevertheless, the single-configuration simulations presented here are sufficient to determine whether the proposed memory system can be made to operate if constructed from presently-available devices.

For time-varying simulation, information concerning the device capacitance is needed in addition to the I-V behavior. Ideally, we would obtain a transfer curve relating capacitance to voltage in a manner similar to that of obtaining the curve describing the I-V behavior. However, sufficiently detailed experimental data is not yet available to describe the change in capacitance versus voltage. Instead, we used a constant value of 1 aF for the NWNV diode junction capacitance (C_j) [78]. In the absence of detailed data, this first-order estimate must suffice for use in simulating overall memory performance. Nonetheless, the simulations developed in this work can incorporate more detailed capacitance characteristics as they are measured.

3.4.2 Top-Gated Nanowire Field Effect Transistor (TGNW-FET) Model

The decoders are composed of TGNW-FETs that are constructed by crossing a microscale wire over a nanowire covered with silicon dioxide. The silicon dioxide isolates the microwire from the nanowire and allows the device to behave like a field-effect transistor, with the microscale wire acting as the gate. Changing the voltage on the microwire gate controls the current flow through the nanowire channel. These field-effect devices are similar to the crossed nanowire FETs (cNWFETs) described by Huang *et al.* [7]. An illustration of a TGNW-FET and a circuit schematic of the device model are shown in Fig. 7. The experimental I-V characteristics for p-type silicon nanowires coated with silicon dioxide and the corresponding TGNW-FET simulation model are shown in Fig. 8. The device behaves as a p-channel MOSFET (PFET), where applying a positive voltage to the gate reduces the conductivity of the channel [53]. The I-V equations for the model are modified versions of first-order MOSFET I-V equations. The modifications to the MOSFET equations involve scaling the input voltages and adding an error correction term. These modifications are empirical in nature and remove any direct

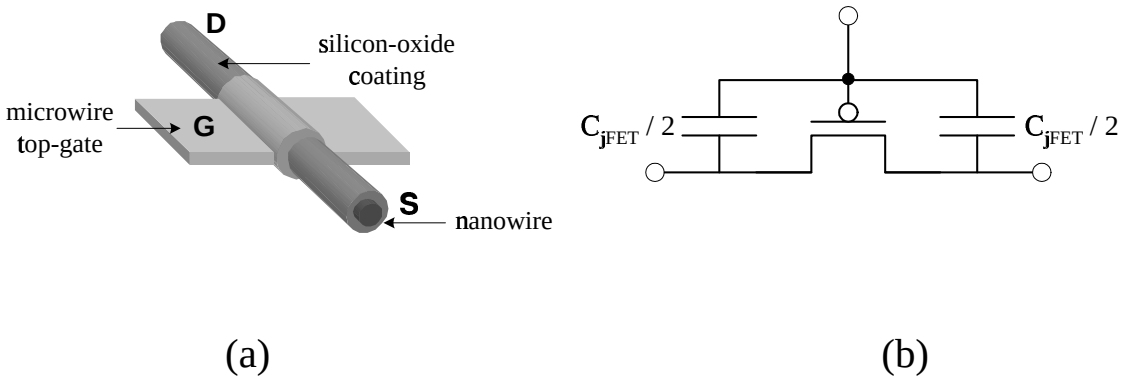


Figure 7: A sketch of (a) the structure and (b) a circuit schematic for a top-gated nanowire FET formed by depositing a microwire over a silicon-dioxide coated nanowire. The device model consists of a PFET transistor and two capacitors.

connection to the underlying physics. However, this is sufficient for the simulations presented here. It is not necessary to represent the underlying physics of the device, only to mimic its experimental behavior.

In addition, a capacitance between the nanowire and microwire is present in the model (C_{jFET}). We assume this capacitance is similar to that of the NVNW-diode junction, i.e., we set $C_{jFET} = C_{jdiode}$. This is a safe assumption, especially for large nanomemory arrays, because in these arrays C_{jFET} is dominated by C_{jdiode}.

3.4.3 Nanowire Interconnect Model

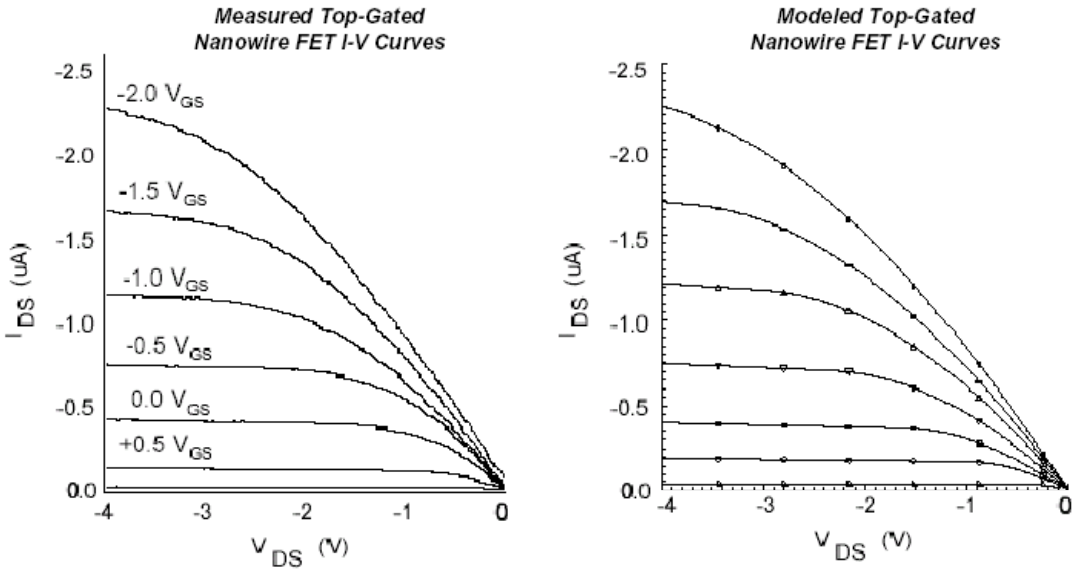
In conventional microelectronics, there is a clear-cut distinction between the devices and the wires that connect them. This distinction does not exist in the crossbar nanomemory considered here. Nanowires in the nanomemory form the devices and also connect these devices to one another. For simulation purposes, these two roles were divided artificially into separate models. The device behavior was captured in the models described in Sections 3.4.1 and 3.4.2. The interconnect behavior was captured in a third model.

Fig. 9 shows an illustration and a circuit schematic of the interconnect model. This is a Π model [53] composed of a resistor and two capacitors. The figure details a unit crossbar (i.e., two crossing nanowires), each the length of the nanowire pitch. The resistance of the unit crossbar determines the values of the resistances (R_{NW}) in Fig. 9, whereas the capacitors (C_{NWsub}) model the capacitances to the substrate below.

The interconnect model shown in Fig. 9 optionally may incorporate a contact resistance R_c. This resistance models the contact between the microwire power supply lines and the nanowires. Its value is approximately 1 MΩ in present devices [52]. This value of R_c is dominant in comparison to the nanowire resistance R_{NW}. Thus, we assume that R_{NW} is negligible in our simulations. Although R_{NW} is not employed in the simulations presented in this paper, including it in the interconnect model provides the capability to account for the nanowire resistance when improvements in the fabrication techniques reduce R_c to a value where the two resistances are comparable.

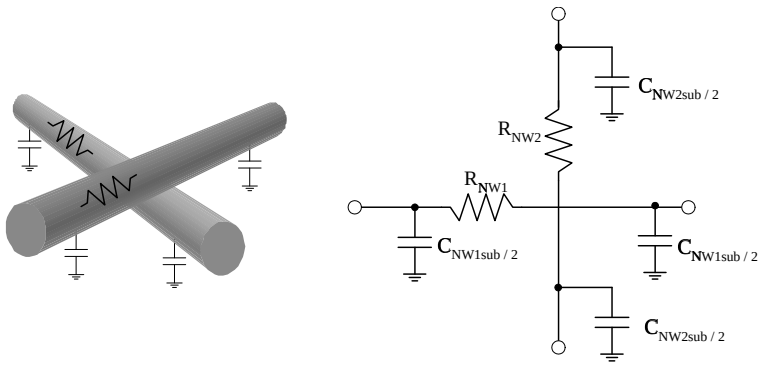
The experimental value of the capacitance C_{NWsub} can be altered by changing the separation distance between the nanomemory array and the substrate or by changing the insulating dielectric between the array and the substrate. Thus, within simulations, C_{NWsub} is treated as a variable parameter. Its value has an important influence on system performance, as is shown in Section 3.5.

Two additional parasitic influences were not included in the interconnect model, but may play a role in nanoscale systems. These are crosstalk capacitance between neighboring wires and parasitic inductances along the wires. These two effects may manifest themselves in systems with small wire pitches or in systems with long and narrow wires operating at high frequencies, respectively. However, these effects should not influence strongly the functionality of a low speed, low frequency prototype nanomemory, such as is considered here. That is, although these two parasitics may impact the speed and energy efficiency of the nanomemory, they will



(a) (b)

Figure 8: I-V curves for the top-gated nanowire FETs as a function of gate voltage. (a) Measured I-V curves for a p-silicon top-gated nanowire FET. (b) Simulated I-V curves for the top-gated nanowire transistor model.



(a) (b)

Figure 9: A sketch of (a) the structure and (b) a circuit schematic for the nanowire interconnect model, which consists of networks of resistors and capacitors.

not affect whether or not the system can be made to operate.

3.5 Nanomemory Simulation and Analysis

The nanomemory is accessed by providing an address to the row and column decoders and then adjusting the supply voltages to force either a read or write operation. The decoders assert a row and a column by turning on the TGNW-FETs in the selected row and column, while turning off at least one TGNW-FET in each nonselected row and column. This procedure isolates a unique point or address in the nanomemory array.

When the TGNW-FETs are turned off, they create an open circuit and leave the voltage upon the non-selected rows and columns "floating," in the absence of a connection to a strong power supply. Allowing the rows and columns to float in this manner risks having non-selected diode junctions inadvertently reprogrammed if these diodes are subjected to voltages from elsewhere in the array that exceed programming thresholds. To help control the voltages across the non-selected rows and columns, a precharge signaling scheme is used. The precharge places a fixed charge on all of the non-selected diodes prior to evaluation. This limits the voltage difference across them.

Each operation is thereby divided into a precharge phase and an evaluation phase. Fig. 10 shows the waveforms of the input and output signals of these two phases for a read operation on the 10×10 nanomemory shown in Fig. 4. The simulation first reads diode (8,8), that is, the diode in row 8 and column 8, followed by a read of diode (9,9). It is of particular importance to be able to simulate the reading of diode (9,9) because it is the worst-case diode for both read and write operations, that is, it is the farthest from the power supplies. Simulation of the reading of diode (8,8) provides an example of the precharge scheme over successive memory accesses. In principle, any address location would do.

The precharge phase asserts all address lines and places a voltage on all the rows and columns. Then, during the evaluation phase, only the selected row and column are asserted. The junction and parasitic capacitances on the non-selected lines hold the precharge voltage while they are isolated from the rest of the circuit. During the evaluation phase, at least one TGNW-FET in each non-selected row and column is turned off, leaving the only path between the row supply and column supply through the selected diode, enabling the reading or writing of a single bit.

When reading a bit from memory, voltages are placed on the row and column supplies such that the selected diode is forward biased, allowing the output current of the nanomemory to reflect the resistance of the selected diode. It is particularly important to choose operating voltages that only forward bias the selected diode. Forward biasing non-selected diodes will cause them to contribute, inadvertently, to the overall output current. In the worst-case memory configuration for reading a logic "0" bit (i.e., when the selected diode is in the off-state and the rest of the diodes are in the on-state), even a slight forward biasing of the non-selected junctions may make the state of the selected diode unreadable. This problem increases with the size of the array since there are more non-selected diodes that can contribute to the overall current.

To avoid this interference from non-selected diodes, we choose precharge and evaluation voltages for reading the memory that force non-selected diodes into a reverse bias or near zero bias. This strategy prevents non-selected diodes from contributing to the output current. The right half of Fig. 10 shows simulation results for the strategy described above. The memory configuration is set to the worst case for reading logic "0". The worst-case diode, that is, diode (9,9), is set to logic "0" and the rest of the diodes are set to logic "1". The top four waveforms are the voltage biases across the diodes being read and two neighboring diodes. The simulation results show that the diodes in non-selected rows and columns are either reverse biased or have a very small forward bias during the evaluation phase.

Although placing non-selected diodes under a reverse bias is effective for reducing unwanted current contributions to the output current, this scheme does run the risk of inadvertently programming on-state devices to off-state devices if the reverse bias exceeds $V_{thresOFF}$. Therefore, it is necessary to use supply voltages that are small enough to ensure $V_{thresOFF}$ is not surpassed. This, in turn, limits the bias that can be placed across the selected diode.

Nevertheless, in the simulation it is possible to achieve excellent ON/OFF current differences for a variety

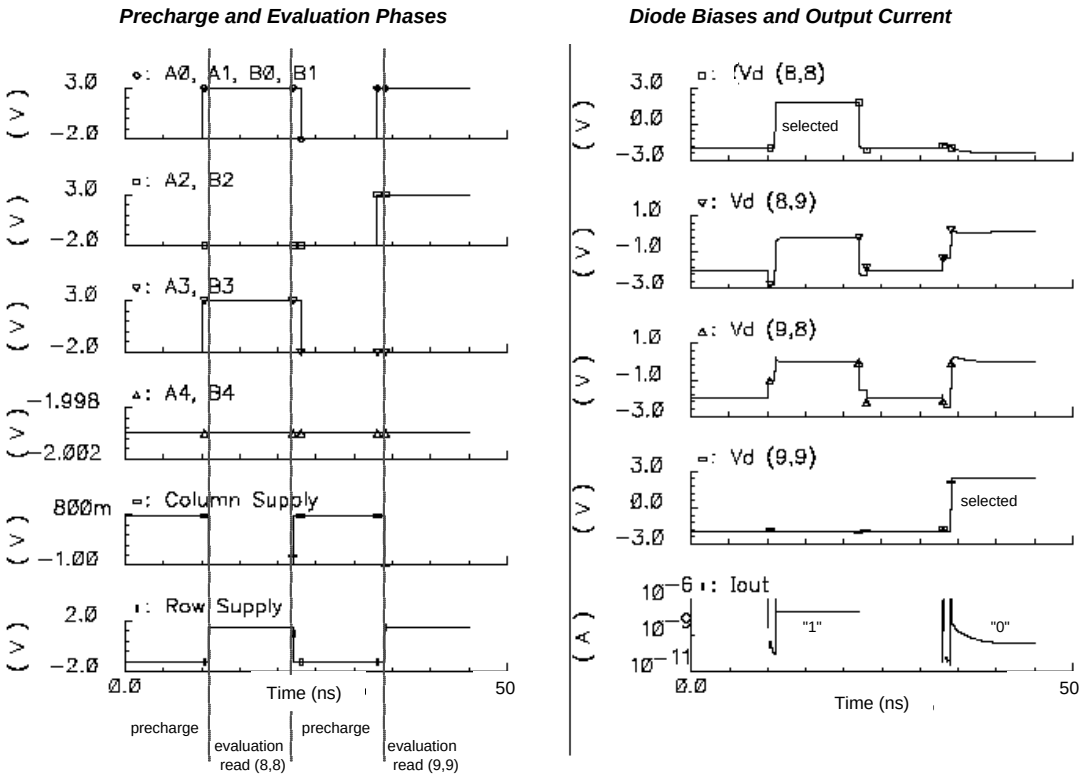


Figure 10: Input and output waveforms for the precharge and evaluation phases of two sequential read operations. The left half of the figure shows the input signals for a read of diode (8,8) followed by a read of diode (9,9). The voltage biases across the selected diodes, (8,8) and (9,9), their neighboring diodes, (8,9) and (9,8), and the memory's output current are shown in the right half of the figure.

Table 1: Simulation results for a read operation performed on the nanomemory shown in Fig. 4.

Nanomemory Array Size	I_{out} (nA) logic "1"	I_{out} (nA) logic "0"	ΔI_{out} (nA)	"1"/"0" Current Ratio
3×3	134	0.8	133	168
10×10	134	0.9	133	149
15×15	134	1.1	133	122
21×21	134	1.6	132	84
45×45	134	16.7	117	8

NOTE: The simulations are performed with zero capacitance to ground and the reported values occur 10 nsec after the evaluation phase begins (see text for details).

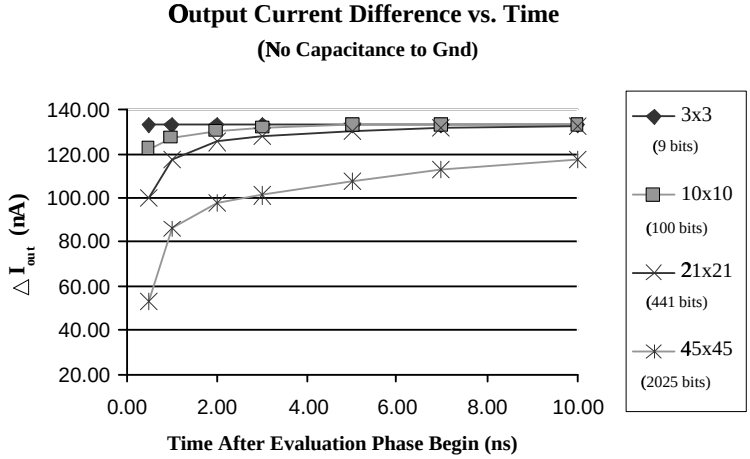


Figure 11: Plot of ΔI_{out} versus time. T=0 corresponds to the beginning of the evaluation phase. The time it takes for ΔI_{out} to reach its maximum value has implications on the speed of the memory. The simulations above have zero capacitance between the nanowires and the substrate.

of different memory arrays, as is shown in Fig. 11. Similarly, Tbl. 1 provides details of the output currents I_{out} for worst-case read operations for both logic "1" and logic "0", as well as the current difference ΔI_{out} and "1"/"0" current ratio between them. These differences are sufficient to read each memory successfully. Furthermore, the high current ratios suggest that read operations can be performed successfully in memory arrays that have been scaled up to include even more rows and columns.

Data is written to the nanomemory by subjecting the selected diode to a bias exceeding the switching threshold. As discussed in Section 3.4.1, a diode in the on-state is switched to the off-state at $V_{thresOFF} \approx -2.75$ V and a diode in the off-state is switched to the on-state at $V_{thresON} \approx 3.8$ V. As with the read operations, care must be taken to avoid inadvertently programming non-selected diodes. However, simulations performed in this work suggest it is feasible to write either logic value to the memory. It was always possible to identify operating conditions that programmed the selected diode without subjecting non-selected diodes to voltages that exceeded thresholds.

The simulations shown in Fig. 11 and Tbl. 1 assume no capacitance between the nanowires and the substrate, that is, $C_{NWsub1} = C_{NWsub2} = 0$. This is a reasonable approximation that can be realized experimentally by raising the crossbar nanomemory sufficiently high above the substrate or using a low-k dielectric between the nanomemory and substrate. Likewise, it should be possible to add a controlled amount of capacitance to the nanowires by reducing the height above the substrate or by using an alternative dielectric. Recent experiments have shown that the capacitances between the memory cell of interest and the substrate may be estimated to be approximately 1aF. Thus, the simulations described above were repeated with this small capacitance to ground added to each unit crossbar in the nanowire interconnect model, that is, $C_{NWsub1} = C_{NWsub2} = 1$ aF. As is shown in Fig. 12, adding capacitance to ground reduces the ΔI_{out} settling times, particularly for the larger arrays. This reduction in settling times occurs because the capacitance to ground provides a better environment for holding the precharge. Without capacitance to ground, the junction capacitance dominates and capacitive coupling to crossing wires can reduce the effectiveness of the precharge.

The simulations developed in this work also can evaluate the effects of varying design parameters on specific aspects of nanomemory performance or evaluate the tradeoffs between traditionally disparate design goals, such as high speed versus low power. For example, the output current difference ΔI_{out} can be improved either by shifting $V_{thresOFF}$ to a lower voltage (more negative voltage) or by increasing $V_{thresON}$. Increasing ΔI_{out} should lead to increased speed and array size. However, altering the programming threshold in this manner requires more energy during write operations. This, of course, increases power consumption. Simulation is an effective way to examine these tradeoffs in a quantitative manner. It can be used to identify optimal operating

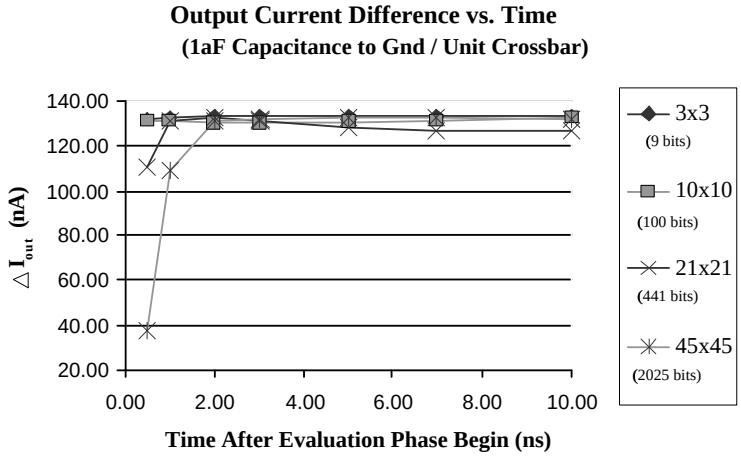


Figure 12: Same as Fig. 11 except that the simulations above have 1 aF of capacitance between the nanowires and the substrate. These simulations show that a small amount of capacitance produces shorter ΔI_{out} settling times.

parameters for specific design goals. For all of the simulations performed to date, the voltage swing for the input signals is relatively large, requiring the address lines to vary by 5 V, while the row supply and column supply vary by 2.75 V and 1.75 V, respectively. These large voltage swings most likely will consume significant dynamic power and require level shifting circuits to interface with conventional electronics. Thus, reducing the signal swing should be an experimental goal. This will reduce power consumption and ease integration with conventional circuits. However, achieving this goal may require smaller diode thresholds. This may reduce the memory speed and could affect functionality. Additional simulations that explicitly incorporate external CMOS circuits are required to explore this issue more fully.

Nevertheless, the simulation results, thus far, suggest that a 45×45 nanomemory would function correctly if built using the Lieber-DeHon architecture and devices. The general trends of these results suggest that larger memories will be functional, as well. Furthermore, as is shown in Section 3.6, the ability to assemble 45×45 nanomemories could be of considerable utility, because their use in a banked topology provides a route to realizing even larger nanomemories.

Although the simulations to date have suggested that the memory is scalable and will function under present device and design parameters, other factors should be considered in future simulations. For example, as the size of the nanomemory array grows, so does the capacitance and resistance of the rows and columns, which can hamper memory performance. Fig. 11 shows the time dependence of ΔI_{out} for simulations of four different memory sizes. The figure shows that increasing the size of the memory also increases the time needed for ΔI_{out} to reach its maximum value. This settling time may reduce the speed of the memory. However, detailed information and models for the connection of the nanomemory to conventional microscale CMOS circuitry, in this case signal amplifiers, are necessary for any realistic estimation of the memory speed.

3.6 Banking Topologies and Area Estimates

Increasing the size of a single nanomemory array may not be the most effective approach for producing memories with very high bit counts. As the size of a memory array increases, so do the resistances and capacitances associated with the array, which increase delay and power consumption. Ultimately, this may threaten functionality.

Further, large memory arrays are more susceptible to fabrication defects, since a single defect in a wire can render all the memory cells along it unusable. Reducing the vulnerability of nanomemories to defects is important. This is because, based on statistical and thermodynamic arguments, it is anticipated that the hierar-

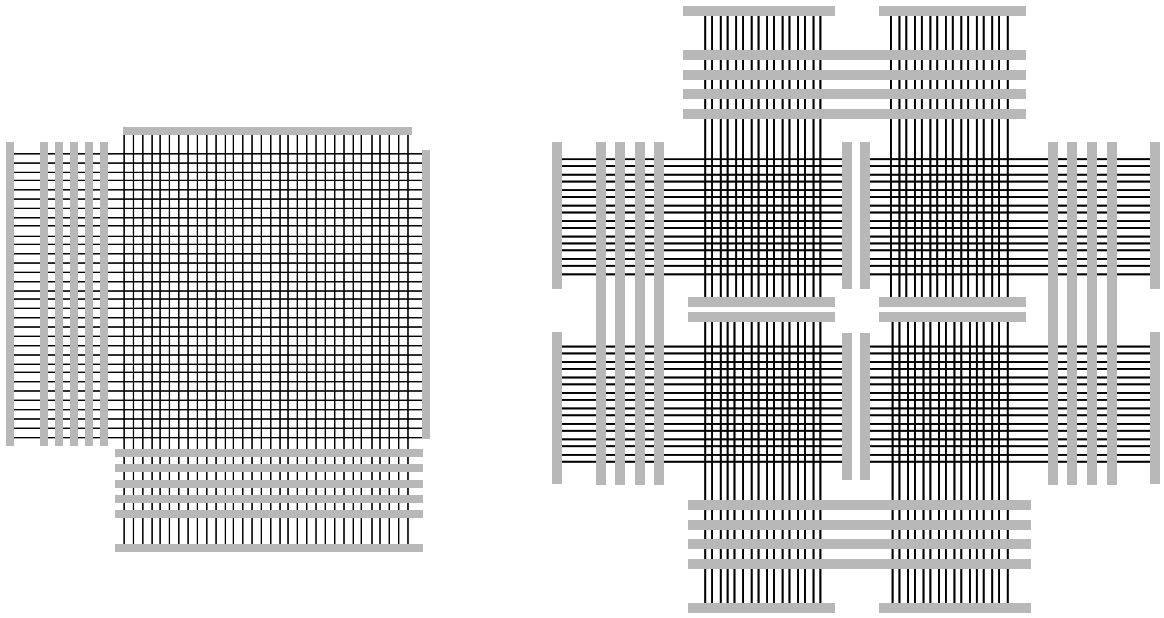


Figure 13: Illustration of two different topologies for realizing a 1-kilobit memory. (a) A single array. (b) A bank of four arrays with an equivalent number of bits.

chical self-assembly strategies being pursued for molecular-scale electronic circuits may produce a significant fraction of defective devices or devices that are imprecisely positioned [97].

To increase defect and fault tolerance, instead of using a single large array to achieve a high bit count, banks of smaller memories might be employed. Fig. 13 illustrates the notion of banking by showing how a one-kilobit memory array can be represented as a single 32×32 array or four 16×16 arrays. This strategy allows for the same level of defect tolerance with less redundancy, since any single defect impacts a smaller number of individual memory bits. Generally speaking, as the degree of banking increases, the amount of required redundancy should decrease, since smaller arrays pay a lower price per defect.

Adopting a banking strategy also increases the overall data throughput for the memory. First, the lower resistances and capacitances of the shorter nanowires in the smaller arrays allow faster access times. Second, banked arrays can be accessed in parallel (i.e., a bit can be accessed from each bank simultaneously) significantly increasing memory performance. Although banked architectures can create more complex fabrication patterns, the regularity of the banks would seem to provide a feasible route to nanomemory assembly. For one example, Harvard University already has made significant progress in the parallel fabrication of multiple arrays in a tiled pattern [17].

The one significant tradeoff generally associated with employing a banking strategy is an increase in area per usable bit. This occurs because each additional bank requires additional wires for encoding and decoding the memory array. Although some of these wires can be shared among the banks (see Fig. 13), banking always results in an increase in the number of address wires. Thus, an optimal banking strategy will employ moderately sized arrays that not only take advantage of the coding scheme to increase density, but also achieve the requisite degree of defect tolerance, parallel access, and other design goals.

Despite the various banking topologies possible for producing a given extended nanomemory system, first-

Table 2: Estimated area for four different memory arrangements targeting a 16-kilobit nanomemory.

Memory Arrangement	Total Locations	Percent Redundancy	Total Area (sq. μm) 20 nm pitch	15 nm pitch	10 nm pitch
136×136 - 1 array	18,496	15.6%	16.6	11.1	6.5
153×153 - 1 array	23,409	46.3%	20.4	13.5	7.8
66×66 - 4 arrays	17,424	8.9%	19.6	13.4	8.1
45×45 - 8 arrays	16,200	1.3%	20.9	14.4	8.8

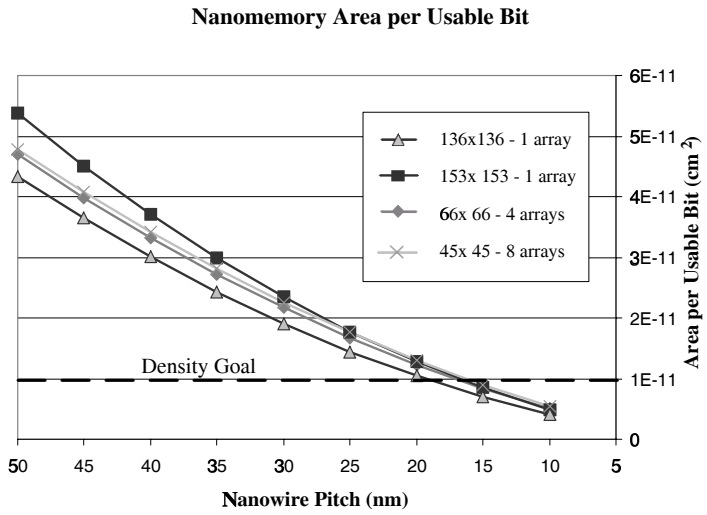


Figure 14: Plot of estimated area per usable bit versus nanowire pitch for four memory arrangements. The calculations assume that 16 kilobits of data can be accessed and the remaining memory locations are reserved for redundancy. The microwire pitch is set at 100nm for all four arrays.

order area calculations suggest that the target nanowire pitch should be similar for a variety of topologies. Fig. 14 shows the estimated bit density for three different banking strategies as a function of nanowire pitch. We also consider two different amounts of redundancy for a single array implementation. To compare these various strategies, these area estimations are premised on a goal of providing 16 kilobits of accessible memory, where any additional memory locations are assumed to be used only as replacements for faulty bits. In other words, the area per usable bit is calculated by dividing the total area for each topology by 16,000, regardless of the actual number of bits. The microwire pitch was set to 100 nm for all of the area calculations. Details of these four memory arrangements are given in Tbl. 2.

The four different memory arrangements described in Fig. 14 and in Tbl. 2 all reach a density of (10^{11} bits/cm^2) when the nanowire pitch is approximately 15 nm. However, a more appropriate measure of the nanotechnology employed in the fabrication of the nanomemory might ignore the area occupied by the microwires and just consider the area occupied by the nanowires. In that case, a nanowire pitch of approximately 30 nm would suffice to achieve this density. Clearly, a variety of topological strategies will be viable to fabricate functional, extended nanomemory systems.

3.7 Summary of Nanomemory System Simulation

Simulations performed on a crossbar nanomemory system based upon the work of Lieber and DeHon [78, 79] suggest that if such a system were built, it would operate. The simulation results suggest that a 45×45 nanomemory array would function properly if constructed from presently fabricated experimental devices. Fur-

thermore, such arrays could be banked to build more extended nanomemory systems, such as a 16 kilobit molecular-scale electronic nanomemory with a bit density of 10^{11} bits/cm^2.

The favorable results from these simulations are encouraging for ongoing and future experiments in the fabrication and prototyping of post-CMOS, crossed-nanowire nanomemory systems. In addition, these results suggest that more complex, extended nanoprocessing systems also could be made to operate using crossed-nanowire architectures that build upon those described above for nanomemories. Thus, the next section of this paper addresses the additional challenges that must be faced in the design of nanoprocessor systems.

4 Beyond Nanomemories: Design of Nanoprocessors Integrated on the Molecular Scale[4]

4.1 Challenges for Developing Nanoprocessors

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

4.1.1 Challenges Posed by the Use of Conventional Microprocessor Architectures

The principal challenge of using conventional architectures [98] 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) [98]. Further still, the synthesis of the aforementioned combinational logic gates requires multiple kinds of devices for optimal performance [53]. This differentiation into a wide variety of devices, circuits, and subsystems is an advantageous structural feature provided by the sophistication of modern microfabrication techniques. 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 structural heterogeneity at the hardware level.

4.1.2 Challenges in the Development of Novel Nanoprocessing Architectures

Most of the nanoprocessor architectures presently proposed [19, 35–44, 46, 83, 97, 99–101] 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 [102], especially field-programmable gate arrays (FPGAs) [103] and programmable logic arrays (PLAs) [104]. As described in detail below in Section 4.3, 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

[4]Some of the material in this section has appeared previously in Das *et al.*, "Architectures and simulations for nanoprocessor systems integrated on the molecular scale," *Lect. Notes Phys.*, vol. 680, pp. 479–513, 2005.

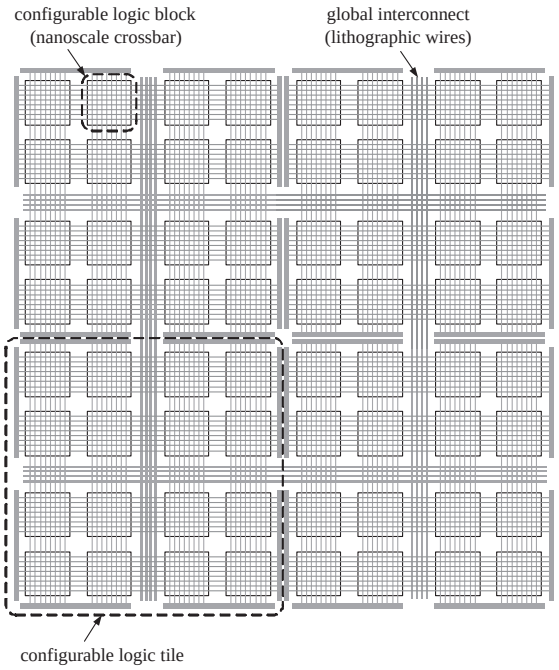


Figure 15: A programmable fabric incorporates molecular-scale devices into the crossbar structures shown in Fig. 1. The fabric builds from them an extended structure of molecules or molecular devices, crossed nanowires, and microwires, such as is shown above. This can provide a platform for realizing a nanoprocessor [79].

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. As discussed in Section 3, this is a homogeneous approach that combines a high degree of scalability with some of the smallest circuit structures demonstrated to date [8, 10]. A number of architectural proposals for nanoprocessors have been put forth that involve the tiling of crossbar subarrays to form programmable fabrics, including the design shown in Fig. 15 [18, 19, 43, 79].

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 [105] and a 15-fold delay penalty [106] 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 relatively homogeneous 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 [103]. 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, but for nanoprocessing, such issues are compounded. For example, in nanomemories, the use of two-terminal devices without gain 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 [107]. 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, 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 [108].

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 [44, 109–112]. These circuits operate using Goto pairs [113] in implementations that were used previously in solid-state nanoelectronic circuit designs [114, 115]. In crossbars, these circuits may be built by incorporating NDR molecules [6].

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.

The need for heterogeneity might be reduced through the use of the crossbar latch designed by the Hewlett-Packard Corporation [11, 116]. 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. 15) that have been used to fabricate nanomemories [10, 16, 69, 117]. 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 are under development [118, 119].

For all approaches to nanoprocessor system design based upon molecular switches, it is well understood that many device-level challenges also must be addressed [15, 97]. 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 Section 4.4. Such challenges must be managed either by improving fabrication capabilities or by introducing defect and variation tolerance into system architectures.

4.2 A Brief Survey of Nanoprocessor System Architectures

Section 4.1 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 [120, 121].

Substantial progress also continues to be made in the scaling of CMOS-based conventional microprocessors.

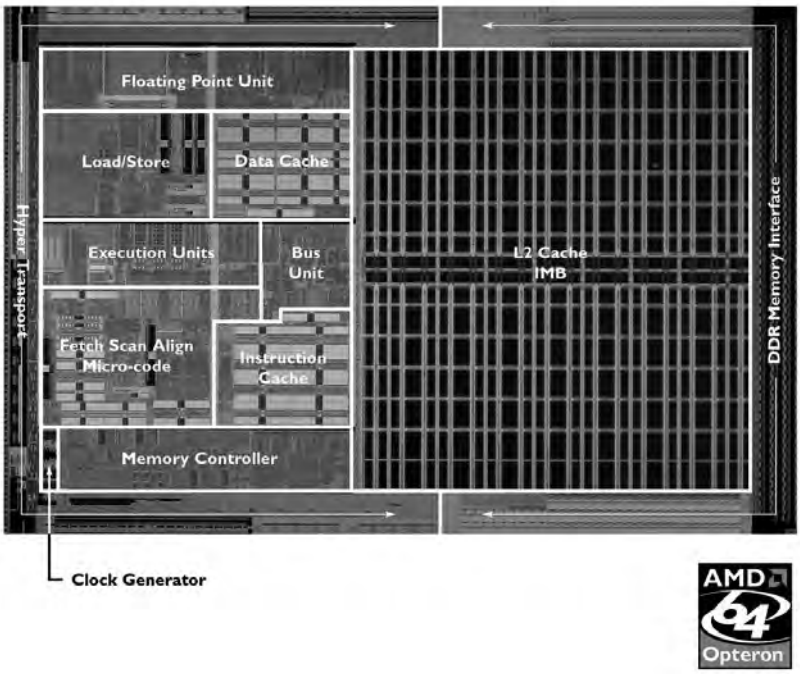


Figure 16: AMD Opteron^{TM} die photo with annotated block structure [125].

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 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 Section 4.2.1, the aggressive miniaturization of conventional architectures to the molecular scale is described. Second, in Section 4.2.2, alternatives to conventional architectures are detailed for cases in which recent nanodevice and nanofabrication developments have made such architectures especially relevant.

4.2.1 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 [122–124]. 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^{TM} chips. As Fig. 16 shows for the AMD Opteron,^{TM} the organ structure still is evident.

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.[5] However, for a number of years, industry investigators and others have examined the likely limits of CMOS technology [126–128, 130, 131] 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.

[5]This topic has been reviewed and discussed extensively elsewhere [27, 126–129]. 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.

An alternative to the straightforward, two-dimensional, aggressive scaling of CMOS is to expand silicon technology into a third dimension [130]. Three-dimensional integration, or 3-D CMOS [132, 133], refers to any of several methods that take conventional, "flat" CMOS wafers and stack them together with an inter-wafer interconnect [134–139]. 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 [140]. Therefore, 3-D architectures may 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 nanowire-like silicon channels to increase current density and to control short-channel effects [141]. Similarly, work at IBM has examined the increased current that results from the use of CNTs in field-effect transistor channels [73, 142].

Another hybrid approach involves the use of self-assembled monolayers (SAMs) 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) [33, 34]. 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 [143]. Also, as mentioned in Section 3, Nantero Corporation is succeeding in introducing novel carbon nanotube-based devices and circuits into a CMOS production line [13, 81, 144].

Hybridization also may be employed at the architectural level. An example of such a hybrid design is the CMOL architecture [83, 145]. CMOL circuits combine CMOS with crossed nanowires and molecular devices. 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 [83, 100, 145], 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 completed to build and test CMOL circuits.

In general, hybridization 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 many interfaces and problematic contacts to achieve 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 [83]. 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.

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.

4.2.2 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 [19, 35, 37, 40, 42, 79, 97]. 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 Section 4.1 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 below.

1. 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) [35–39], nanoscale neural networks [40, 83], nanocells [41, 42, 47], and biologically inspired electronic system structures such as the virus nanoblock (VNB) [146, 147]. Each of these encompasses important ideas and has virtues either in ease of fabrication or in ultra-low power consumption.

The QCA approach [35–39] 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 experimentally.

The nanocell architecture [41, 42, 47] 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 [41].

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 [41]. Experimental nanocell memories recently have been fabricated [47] 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 present 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.

2. 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 in-

homogeneous 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 [8, 10, 17–19, 25, 69], including the detailed simulations described in Section 4.4.

There have been criticisms of the use of PLAs to develop nanoprocessors [83]. 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 [19] does incorporate gain-producing nanowire-based nanotransistors, as is described in detail in Section 4.3.2. 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 [19].

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

4.3 Principles of Nanoprocessor Architectures Based on FPGAs and PLAs

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 array architecture for a nanoprocessor will be explored, the DeHon-Wilson PLA.

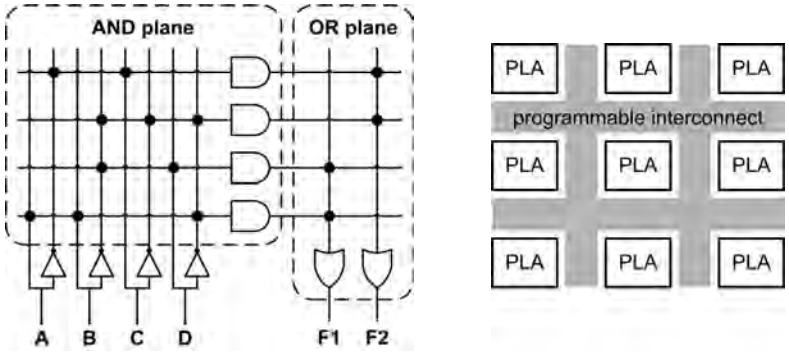
4.3.1 Description of Regular Arrays, FPGAs, and PLAs: Advantages and Challenges

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”) [103]. 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, can produce such homogeneous systems of nanostructures [8, 17, 25].

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. 17(a). Fig. 17(b) shows an extended system architecture based on PLAs. This system structure is similar to that used for FPGAs. (See Section 4.1.2 for a brief description of FPGAs.)

Because of the underlying homogeneity of such structures, thus far they have been outperformed by classical microprocessor architectures in carrying out general-purpose computation. For example, in 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 [105].

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 address this problem. At the nanoscale, it is conceivable that a system may 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 [148]. 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.



(a) (b)

Figure 17: 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.

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 [19].

4.3.2 The DeHon-Wilson PLA Architecture

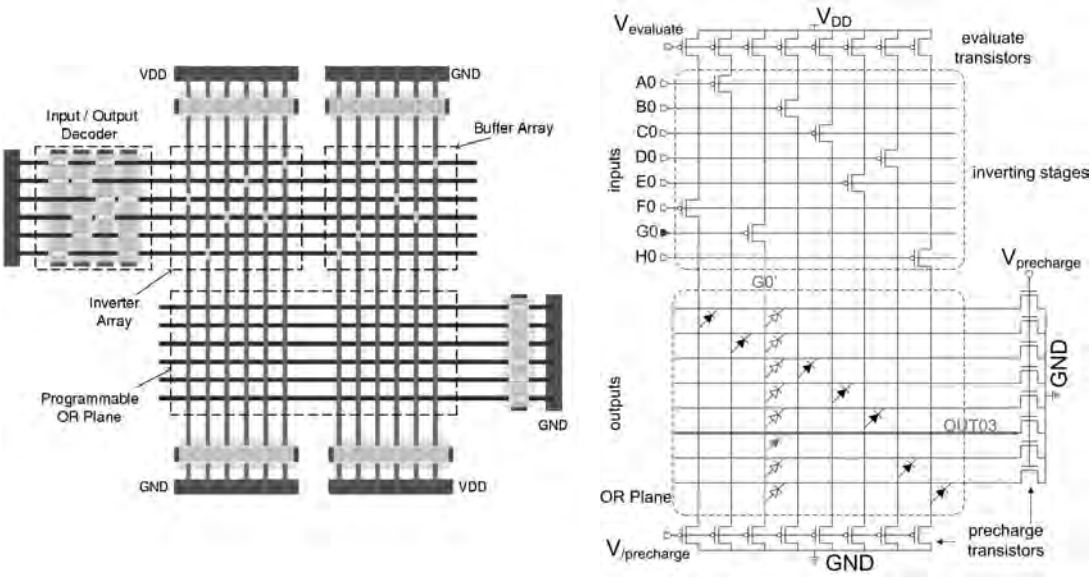
A detailed example of a nanoarray architecture that utilizes nanowires in readily realizable crossbar structures is the DeHon-Wilson PLA architecture [19, 79, 149, 150]. A high-level diagram of this architecture is shown in Figs. 15 and 17(b), while Fig. 18(a) provides a detailed view of the low-level implementation. As with microelectronic PLA-based designs [104], 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 number of OR gates in parallel), as shown in Fig. 17(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 [53].

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. 18(a). The system is extended by tiling crossbar subarrays, as illustrated in Fig. 15. Fig. 18(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. Fig. 18(b) shows a more detailed circuit-level characterization of the left-hand side of the system in Fig. 18(a). In the bottom half of the subarray shown in Fig. 18(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.

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



(a) (b)

Figure 18: 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).

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 [149]. Also, the inverter and buffer arrays can function in two modes, static and dynamic [19]. In dynamic mode, static power consumption is reduced [53]. This ameliorates the potential problem [83] 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 [151] and as is discussed further below.

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

As stated earlier, system simulation can produce an integrated, multi-level view of the performance of a candidate nanocomputer architecture. 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. Here, we describe such a simulation and analysis of the DeHon-Wilson PLA [19].

4.4.1 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. 18(b). Three of these devices also are employed in the construction of nanomemory prototypes and are described in Section 3.4. These are the nonvolatile nanowire (NVNW) diode, the microwire top-gated FET (TGNW-FET), and the nanowire

Table 3: 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

interconnects.

The fourth device and device model required for the nanoPLA is the crossed-nanowire FET (cNWFET) [17, 23, 51, 52], 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) [23]. 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. Fig. 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 [7], 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 [53]. 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 [51] and that the value of this threshold can be controlled [52]. 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 or system design shown in Fig. 18. Parasitic behaviors of the nanowire arrays, such as coupling capacitance, also were incorporated.

4.4.2 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. 18(b). The PLA is programmed with the pattern of diodes depicted there and described in Section 4.3.2. The input vectors to the PLA are given in Table 3.

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. 18(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. 19, which shows an output waveform for OUT_{03} when the circuit in Fig. 18(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 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

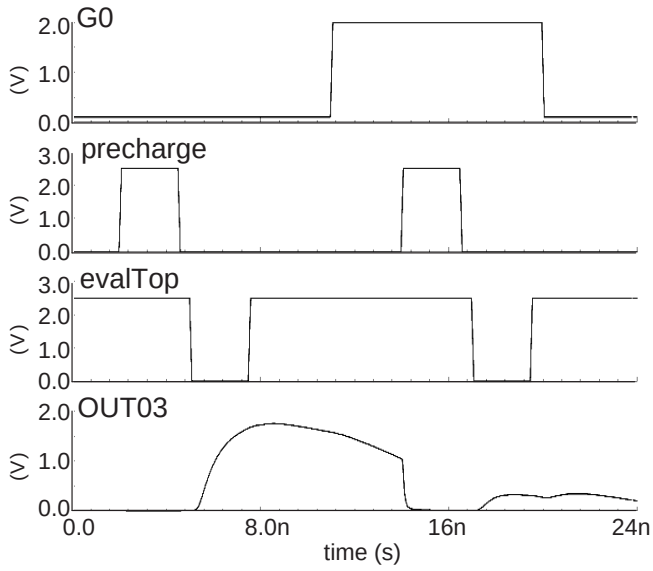


Figure 19: Waveforms describing how the circuit in Fig. 18(b) inverts input signal G_0 to produce output signal OUT_{03}. See discussion in text.

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. 18(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. 18(b) in order to produce the desired output signal on OUT_{03}. As can be seen in Fig. 19, 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. 19. 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. 20 shows the output-voltage dependence of the number of diodes programmed in the "on" state along the G'_0 column (see Fig. 18(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. 20, 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 shown in Table 3, and used in this simulation for the low output, drives all the row wires in Fig. 18(b) except OUT_{03} to logic "1." This has the effect of 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, i.e., the fan-out, 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

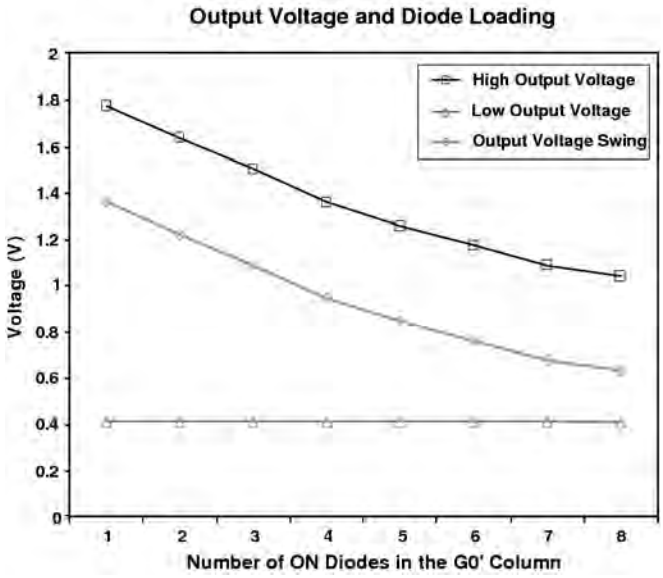


Figure 20: High and low output voltages and output voltage swing plotted against the number of diodes programmed ON in the G_0' column.

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 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.

4.5 Further Implications and Issues for System Simulations

Although the results shown above are derived from simulations of a particular nanoprocessor system design, 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 Section 4.4.2 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 for 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 [19, 43, 79, 97, 99]. 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.

5 Conclusion

In this paper, we have examined potential approaches to the system-level design and simulation of an extended nanocomputer system that is integrated on the molecular scale. We have considered such systems to be the union of two component subsystems, nanomemories and nanoprocessors.

For each of these components, we have focused upon ultra-dense, array-based system-level design strategies or architectures that offer significant promise for the fabrication and demonstration of extended system prototypes in the near term. In the case of nanomemory systems, recent research in nanoelectronic devices and in the nanofabrication of prototype nanomemory arrays [12, 14] has provided evidence of the efficacy of the crossbar array architecture. For nanoprocessor systems, we have surveyed a range of possible architectural approaches. Following this survey, we have focused upon crossbar-based architectures that occupy an important middle ground between conventional microelectronic architectures and a set of more radical nanoelectronic architectures.

To explore the prospective performance of nanocomputer systems based upon these crossbar-based architectures, we have adapted the simulation tools and techniques used widely 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 detailed simulations of the circuits and subsystems embodied in these architectures for nanomemory and nanoprocessor systems, we have examined some of the trade-offs that affect nanoelectronic systems built from molecular-scale devices. Many of these trade-offs apply to almost any nanocomputer architecture that might be adopted to harness molecules or molecular-scale devices in ultra-dense electronic computing structures. System simulations such as we have described in this paper can indicate the extent to which enhancements in devices might improve system performance. If such improvements are significant, then it becomes worthwhile for experimentalists to invest in enhancing designs for nanodevices and techniques for fabricating them.

Thus, 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 nanocomputer systems development. Innovative system design and simulation strategies, coupled closely with device and system experiments, may both speed the realization and optimize the performance of ultra-dense electronic computers integrated on the molecular scale.

Acknowledgments

The authors thank Professors André DeHon and James Heath of the California Institute of Technology, Professor Charles Lieber of Harvard University, plus R. Stanley Williams, 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. Thanks also are due to Professor Konstantin Likharev of Stony Brook University and Brent Segal of Nantero Corporation for so kindly providing us with

copies of their forthcoming papers. This research was supported by the Intelligence Technology Innovation Center (ITIC) Nano-Enabled Technology Initiative (NETI).

References

[1] J. C. Ellenbogen and J. C. Love. Architectures for molecular electronic computers: 1. logic structures and an adder designed from molecular electronic diodes. *Proc. IEEE*, 88(3):386–426, 2000.

[2] K. S. Kwok and J. C. Ellenbogen. Moletronics: future electronics. *Materials Today*, 5(2):28–37, 2002.

[3] M. A. Reed, C. Zhou, C. J. Muller, T. P. Burgin, and J. M. Tour. Conductance of a molecular junction. *Science*, 278(5336):252–254, 1997.

[4] C. P. Collier, E. W. Wong, M. Belohradsky, F. M. Raymo, J. F. Stoddart, P. J. Kuekes, R. S. Williams, and J. R. Heath. Electronically configurable molecular-based logic gates. *Science*, 285:391–394, 1999.

[5] C. P. Collier, G. Mattersteig, E. W. Wong, Y. Luo, K. Beverly, J. Sampaio, F. M. Raymo, J. F. Stoddart, and J. R. Heath. A [2]catenane-based solid state electronically reconfigurable switch. *Science*, 289:1172–1175, 2000.

[6] J. Chen, M. A. Reed, A. M. Rawlett, and J. M. Tour. Large on-off ratios and negative differential resistance in a molecular electronic device. *Science*, 286:1550–1552, November 1999.

[7] Y. Huang, X. Duan, Y. Cui, L. J. Lauhon, K. H. Kim, and C. M. Lieber. Logic gates and computation from assembled nanowire building blocks. *Science*, 294:1313–1317, November 2001.

[8] N. A. Melosh, A. Boukai, F. Diana, B. Gerardot, A. Badolato, P. M. Petroff, and J. R. Heath. Ultrahigh-density nanowire lattices and circuits. *Science*, 300(5616):112–115, 2003.

[9] A. Bachtold, P. Hadley, T. Nakanishi, and C. Dekker. Logic circuits with carbon nanotube transistors. *Science*, 294:1317–1320, 2001.

[10] Y. Luo, C. P. Collier, J. O. Jeppesen, K. A. Nielson, E. Delonno, G. Ho, J. Perkins, H. Tseng, T. Yamamoto, J. F. Stoddart, and J. R. Heath. Two-dimensional molecular electronics circuits. *ChemPhysChem*, 3:519–525, 2002.

[11] P. J. Kuekes, D. R. Stewart, and R. S. Williams. The crossbar latch: Logic value storage, restoration, and inversion in crossbar circuits. *J. Appl. Phys.*, 97, 2005.

[12] W. Wu, G. Y. Jung, D. L. Olynick, J. Straznicky, Z. Li, X. Li, D. A. A. Ohlberg, Y. Chen, S. Y. Wang, J. A. Liddle, W. M. Tong, and R. Stanley Williams. One-kilobit cross-bar molecular memory circuits at 30-nm half-pitch fabricated by nanoimprint lithography. *Appl. Phys. A*, 80:1173–1178, 2005.

[13] D. K. Brock, J. W. Ward, C. Bertin, B. M. Segal, and T. Rueckes. Fabrication and applications of single-walled carbon nanotube fabrics. In A. Busnaina, editor, *CRC Nanomanufacturing Handbook*, pages 41–64. CRC Press, 2006.

[14] J. E. Green, J. W. Choi, A. Boukai, Y. Bunimovich, E. Johnston-Halperin, E. DeIonno, Y. Luo, B. A. Sheriff, K. Xu, Y. S. Shin, H.-R. Tseng, J. F. Stoddart, and J. R. Heath. A defect-tolerant molecular electronic memory. in press.

[15] J. R. Heath. Wires, switches, and wiring. A route toward a chemically assembled electronic nanocomputer. *Pure Appl. Chem.*, 72(1-2):11–20, 2000.

[16] P. J. Kuekes, J. R. Heath, and R. S. Williams. Molecular wire crossbar memory. United States Patent 6,128,214, October 2000.

[17] D. Whang, S. Jin, Y. Wu, and C. M. Lieber. Large-scale hierarchical organization of nanowire arrays for integrated nanosystems. *Nano Lett.*, 3(9):1255–1259, September 2003.

[18] G. Snider, P. Kuekes, and R. S. Williams. CMOS-like logic in defective, nanoscale crossbars. *Nanotechnology*, 15(8):881–891, August 2004.

[19] A. DeHon and M. J. Wilson. Nanowire-based sublithographic programmable logic arrays. In *Proc. ACM/SIGDA FPGA*, pages 123–132, Monterey, CA, 2004. ACM Press.

[20] P. J. Kuekes, G. S. Snider, and R. S. Williams. Crossbar nanocomputers. *Scientific American*, pages 72–80, November 2005.

[21] M. M. Ziegler, C. A. Picconatto, J. C. Ellenbogen, A. DeHon, D. Wang, Z. H. Zhong, and C. M. Lieber. Scalability simulations for nanomemory systems integrated on the molecular scale. In *Molecular Electronics III*, volume 1006 of *Ann. N.Y. Acad. Sci.*, pages 312–330. 2003.

[22] S. Das, G. S. Rose, M. M. Ziegler, C. A. Picconatto, and J. C. Ellenbogen. Architectures and simulations for nanoprocessor systems integrated on the molecular scale. *Lect. Notes Phys.*, 680:479–513, 2005.

[23] Y. Huang, X. Duan, Q. Wei, and C. M. Lieber. Directed assembly of one-dimensional nanostructures into functional networks. *Science*, 291:630–633, January 2001.

[24] M. D. Austin, H. Ge, W. Wu, M. Li, Z. Yu, D. Wasserman, S. A. Lyon, and S. Y. Chou. Fabrication of 5 nm linewidth and 14 nm pitch features by nanoimprint lithography. *Appl. Phys. Lett.*, 84(26):5299–5301, June 2004.

[25] G. Y. Jung, S. Ganapathiappan, D. A. A. Ohlberg, D. L. Olynick, Y. Chen, W. M. Tong, and R. S. Williams. Fabrication of a 34 $\times$ 34 crossbar structure at 50 nm half-pitch by UV-based nanoimprint lithography. *Nano Lett.*, 4(7):1225–1229, 2004.

[26] A. DeHon, S. C. Goldstein, P. J. Kuekes, and P. Lincoln. Nonphotolithographic nanoscale memory density prospects. *IEEE Trans. Nano.*, 4(2):215–228, March 2005.

[27] International technology roadmap for semiconductors. Technical report, Semiconductor Industry Association, 2005.

[28] M. Ratner and V. Mujica. Molecular electronics from the bottom up. In S. E. Lyshevski, editor, *CRC Handbook on Molecular and Nano Electronics*. CRC Press, 2007.

[29] M. Reed and W. Wang. Electrical characterization of self-assembled monolayers. In S. E. Lyshevski, editor, *CRC Handbook on Molecular and Nano Electronics*. CRC Press, 2007.

[30] R. Metzger. Molecular rectifiers. In S. E. Lyshevski, editor, *CRC Handbook on Molecular and Nano Electronics*. CRC Press, 2007.

[31] B. Yu and M. Meyyappan. Inorganic nanowires in electronics. In S. E. Lyshevski, editor, *CRC Handbook on Molecular and Nano Electronics*. CRC Press, 2007.

[32] B. Kanshibotla, S. Pramanik, and S. Bandyopadhyay. Self assembly of nanostructures using nanoporous alumina templates. In S. E. Lyshevski, editor, *CRC Handbook on Molecular and Nano Electronics*. CRC Press, 2007.

[33] Z. M. Liu, A. A. Yasser, J. S. Lindsey, and D. F. Bocian. Molecular memories that survive silicon device processing and real-world operation. *Science*, 302(5650):1543–1545, 2003.

[34] C. Li, W. Fan, B. Lei, D. Zhang, S. Han, T. Tang, X. Liu, Z. Liu, S. Asano, M. Meyyappan, J. Han, and C. Zhou. Multilevel memory based on molecular devices. *Appl. Phys. Lett.*, 84(11):1949–1951, March 2004.

[35] C. S. Lent and P. D. Tougaw. Device architecture for computing with quantum dots. *Proc. IEEE*, 85(4):541–557, 1997.

[36] G. L. Snider, A. O. Orlov, I. Amlani, G. H. Bernstein, C. S. Lent, J. L. Merz, and W. Porod. Quantum-dot cellular automata: Line and majority logic gate. *Japanese J. Appl. Phys. Part 1*, 38(12B):7227–7229, 1999.

[37] V. P. Roychowdhury, D. B. Janes, and S. Bandyopadhyay. Nanoelectronic architecture for Boolean logic. *Proc. IEEE*, 85(4):574–588, 1997.

[38] W. Porod, C. S. Lent, G. H. Bernstein, A. O. Orlov, I. Amlani, G. L. Snider, and J. L. Merz. Quantum-dot cellular automata: computing with coupled quantum dots. *Intl. J. Elect.*, 86(5):549–590, 1999.

[39] G. Toth, C. S. Lent, P. D. Tougaw, Y. Brazhnik, W. W. Weng, W. Porod, R. W. Liu, and Y. F. Huang. Quantum cellular neural networks. *Superlattices and Microstructures*, 20(4):473–478, 1996.

[40] O. Turel, J. H. Lee, X. Ma, and K. K. Likharev. Neuromorphic architectures for nanoelectronic circuits. *Int. J. Circ. Theor. Appl.*, 32:277–302, 2004.

[41] C. P. Husband, S. M. Husband, J. S. Daniels, and J. M. Tour. Logic and memory with nanocell circuits. *IEEE Trans. Elect. Dev.*, 50(9):1865–1875, 2003.

[42] J. M. Tour, W. L. van Zandt, C. P. Husband, S. M. Husband, L. S. Wilson, P. D. Franzon, and D. P. Nackashi. Nanocell logic gates for molecular computing. *IEEE Trans. Nano.*, 1(2):100–109, 2002.

[43] S. C. Goldstein and M. Budiu. Nanofabrics: Spatial computing using molecular electronics. In *Proc. Intl. Symp. Comp. Arch.*, June 2001.

[44] G. S. Rose and M. R. Stan. Programmable logic using molecular devices in a three-dimensional architecture. Presentation at the Engr. Intl. Conf. on Mol. Elect., San Diego, CA (unpublished), January 2005.

[45] J. Han and P. Jonker. A defect- and fault-tolerant architecture for nanocomputers. *Nanotechnology*, 14(2):224–230, 2003.

[46] V. Beiu. From unreliable nanodevices to low power nanoarchitectures. In S. E. Lyshevski, editor, *CRC Handbook on Molecular and Nano Electronics*. CRC Press, 2007.

[47] J. M. Tour, L. Cheng, D. P. Nackashi, Y. X. Yao, A. K. Flatt, S. K. St. Angelo, T. E. Mallouk, and P. D. Franzon. Nanocell electronic memories. *J. Am. Chem. Soc.*, 125(43):13279–13283, 2003.

[48] A. Fijany and B. N. Toomarian. New design for quantum dots cellular automata to obtain fault tolerant logic gates. *J. Nanop. Res.*, 3(1):27–37, 2001.

[49] C. S. Lent and B. Isaksen. Clocked molecular quantum-dot cellular automata. *IEEE Trans. Elect. Dev.*, 50(9):1890–1896, 2003.

[50] C. S. Lent, B. Isaksen, and M. Lieberman. Molecular quantum-dot cellular automata. *J. Am. Chem. Soc.*, 125(4):1056–1063, 2003.

[51] A. B. Greytak, L. J. Lauhon, M. S. Gudiksen, and C. M. Lieber. Growth and transport properties of complementary germanium nanowire field-effect transistors. *Appl. Phys. Lett.*, 84(21):4176–4178, May 2004.

[52] Y. Cui, Z. Zhong, D. Wang, W. U. Wang, and C. M. Lieber. High performance silicon nanowire field effect transistors. *Nano Lett.*, 3(2):149–152, 2003.

[53] J. M. Rabaey, A. P. Chandrakasan, and B. Nikolic. *Digital Integrated Circuits*. Prentice-Hall, Inc., Englewood Cliffs, NJ, 2nd edition, 2002.

[54] J. van Ruitenbeek, E. Scheer, and H. B. Weber. Contacting individual molecules using mechanically controllable break junctions. *Lect. Notes Phys.*, 680:253–274, 2005.

[55] W. Wang, T. Lee, and M. Reed. Intrinsic electronic conduction mechanisms in self-assembled monolayers. *Lect. Notes Phys.*, 680:275–300, 2005.

[56] J. Tomfohr, G. Ramachandran, O. F. Sankey, and S. M. Lindsay. Making contacts to single molecules: Are we there yet? *Lect. Notes Phys.*, 680:301–312, 2005.

[57] R. M. Metzger. Six unimolecular rectifiers and what lies ahead. *Lect. Notes Phys.*, 680:313–350, 2005.

[58] E. Thune and C. Strunk. Quantum transport in carbon nanotubes. *Lect. Notes Phys.*, 680:351–380, 2005.

[59] J. Jortner, A. Nitzan, and M. A. Ratner. Foundation of molecular electronics - charge transport in molecular conduction junctions. *Lect. Notes Phys.*, 680:13–54, 2005.

[60] P. Hanggi, S. Kohler, J. Lehmann, and M. Strass. AC-driven transport through molecular wires. *Lect. Notes Phys.*, 680:55–76, 2005.

[61] R. DiFelice, A. Calzolari, D. Versano, and A. Rubio. Electronic structure calculations for nanomolecular systems. *Lect. Notes Phys.*, 680:77–116, 2005.

[62] K. Stokbro, J. Taylor, M. Brandbyge, and H. Guo. Ab-initio based nonequilibrium Green's function formalism for calculating electron transport in molecular devices. *Lect. Notes Phys.*, 680:117–152, 2005.

[63] A. DiCaarlo, A. Pecchia, L. Latessa, T. Frauenheim, and G. Seifert. Tight-binding DFT for molecular electronics (gDFTB). *Lect. Notes Phys.*, 680:153–184, 2005.

[64] N. Bushong and M. DiVentra. Current-induced effects in nanoscale conductors. *Lect. Notes Phys.*, 680:185–206, 2005.

[65] M. Wegewijs, M. H. Hettler, J. Konig, A. Thielmann, C. Romeike, and K. Nowack. Single electron tunneling in small molecules. *Lect. Notes Phys.*, 680:207–228, 2005.

[66] M. Thorwart, M. Grifoni, and R. Egger. Transport through intrinsic quantum dots in interacting carbon nanotubes. *Lect. Notes Phys.*, 680:229–249, 2005.

[67] L. A. Bumm, J. J. Arnold, M. T. Cygan, T. D. Dunbar, T. P. Burgin, L. Jones, D. L. Allara, J. M. Tour, and P. S. Weiss. Are single molecular wires conducting? *Science*, 271(5256):1705–1707, 1996.

[68] Y. Selzer, M. A. Cabassi, T. S. Mayer, and D. L. Allara. Temperature effects on conduction through a molecular junction. *Nanotechnology*, 15(7):S483–S488, 2004.

[69] Y. Chen, G. Jung, D. A. A. Ohlberg, X. Li, D. R. Stewart, J. O. Jeppesen, K. A. Nielsen, J. F. Stoddart, and R. S. Williams. Nanoscale molecular-switch crossbar circuits. *Nanotechnology*, 14:462–468, 2003.

[70] R. M. Metzger. All about (N-hexadecylquinolin-4-ium-1-yl) methylidenetricyanoquinodimethanide, a unimolecular rectifier of electrical current. *J. Mater. Chem.*, 10:55–62, 2000.

[71] S. J. Tans, A. R. M. Verschueren, and C. Dekker. Room-temperature transistor based on a single carbon nanotube. *Nature*, 393:49–52, May 1998.

[72] H. W. Ch. Postma, T. Teepen, Z. Yao, M. Grifoni, and C. Dekker. Carbon nanotube single-electron transistors at room temperature. *Science*, 293:76–79, July 2001.

[73] S. Heinze, J. Tersoff, and P. Avouris. Carbon nanotube electronics and optoelectronics. *Lect. Notes Phys.*, 680:381–410, 2005.

[74] A. Javey, R. Tu, D. B. Farmer, J. Guo, R. G. Gordon, and H. Dai. High performance n-type carbon nanotube field-effect transistors with chemically doped contacts. *Nano Lett.*, 5(2):345–348, 2005.

[75] J. C. Ellenbogen. Monomolecular electronic device. United States Patent 6,339,227, issued to the MITRE Corporation, 2002.

[76] 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, and D. C. Ralph. Coulomb blockade and the Kondo effect in single-atom transistors. *Nature*, 417:722–725, June 2002.

[77] W. Liang, M. P. Shores, M. Bockrath, J. R. Long, and H. Park. Kondo resonance in a single-molecule transistor. *Nature*, 417:725–729, June 2002.

[78] C. M. Lieber et al. Design and hierarchical assembly of nanowire-based moletronics. DARPA Moletronics PI Meeting, 2002.

[79] A. DeHon. Array-based architecture for FET-based, nanoscale electronics. *IEEE Trans. Nano.*, 2(1):23–32, 2003.

[80] D. B. Strukov and K. K. Likharev. Defect-tolerant architectures for nanoelectronic crossbar memories. *J. Nanosci. and Nanotech.*, August 2006.

[81] G. Stix. Nanotubes in the clean room. *Scientific American*, pages 82–85, February 2005.

[82] D. R. Stewart, D. A. A. Ohlberg, P. A. Beck, Y. Chen, R. Stanley Williams, J. O. Jeppesen, K. A. Nielsen, and J. Fraser Stoddart. Molecule-independent electrical switching in Pt/organic monolayer/Ti devices. *Nano Lett.*, 4(1):133–136, 2004.

[83] K. K. Likharev and D. B. Strukov. CMOL: Devices, circuits, and architectures. *Lect. Notes Phys.*, 680:447–478, 2005.

[84] J. E. Savage, E. Rachlin, A. DeHon, C. M. Lieber, and Y. Wu. Radial addressing of nanowires. *ACM J. Emerg. Tech. in Comp. Sys.*, 2:129–154, 2006.

[85] Z. Zhong, D. Wang, Y. Cui, M. W. Bockrath, and C. M. Lieber. Nanowire crossbar arrays as address decoders for integrated nanosystems. *Science*, 302(5649):1377–1379, November 2003.

[86] R. Beckman, E. Johnston-Halperin, Y. Luo, J. E. Green, and J. R. Heath. Bridging dimensions: Demultiplexing ultrahigh-density nanowire circuits. *Science*, 310:465–468, 2005.

[87] T. Hogg, Y. Chen, and P. J. Kuekes. Assembling nanoscale circuits with randomized connections. *IEEE Trans. Nano.*, 5(2):110–122, March 2006.

[88] Y. Cui and C. M. Lieber. Functional nanoscale electronic devices assembled using silicon nanowire building blocks. *Science*, 291:851–853, 2001.

[89] Cadence Design Framework II, Version IC 5.0.33. Cadence Design Systems, Inc., San Jose, CA, 2004.

[90] M. M. Ziegler and M. R. Stan. A case for CMOS/nano co-design. In *Proc. IEEE/ACM Intl. Conf. Comp. Aid. Des.*, pages 348–352, 2002.

[91] M. M. Ziegler and M. R. Stan. The CMOS/nano interface from a circuits perspective. In *Proc. Int'l Symp. Circuits and Systems*, volume 4, pages 904–907, 2003.

[92] D. O. Pederson and A. Sangiovanni-Vincentelli. SPICE 3 version 3F5 user's manual. Department of EECS, University of California, Berkeley, CA.

[93] M. Bhattacharya and P. Mazumder. Augmentation of SPICE for simulation of circuits containing resonant tunneling diodes. *IEEE Trans. Computer-Aided Design Integrated Circuits Systems*, 20:39–50, 2001.

[94] M. M. Ziegler and M. R. Stan. Design and analysis of crossbar circuits for molecular nanoelectronics. In *Proc. IEEE Conf. on Nano.*, pages 323–327, 2002.

[95] G. S. Rose, M. M. Ziegler, and M. R. Stan. Large-signal two-terminal device model for nanoelectronic circuit analysis. *IEEE Trans. VLSI*, 12(11):1201–1208, November 2004.

[96] G. S. Rose, A. C. Cabe, N. Gergel-Hackett, N. Majumdar, M. R. Stan, J. C. Bean, L. R. Harriott, Y. Yao, and J. M. Tour. Design approaches for hybrid CMOS/molecular memory based on experimental data. In *Proc. Great Lakes Symp. on VLSI*, pages 2–7, Philadelphia, PA, May 2006.

[97] J. R. Heath, P. J. Kuekes, G. S. Snider, and R. S. Williams. A defect-tolerant computer architecture: Opportunities for nanotechnology. *Science*, 280(5370):1716–1721, 1998.

[98] J. L. Hennessy and D. A. Patterson. *Computer Architecture: A Quantitative Approach*. Morgan Kaufmann, San Mateo, CA, 3rd edition, 2002.

[99] M. R. Stan, P. D. Franzon, S. C. Goldstein, J. C. Lach, and M. M. Ziegler. Molecular electronics: From devices and interconnect to circuits and architecture. *Proc. IEEE*, 91(11):1940–1957, 2003.

[100] D. B. Strukov and K. K. Likharev. CMOL FPGA: A reconfigurable architecture for hybrid digital circuits with two-terminal nanodevices. *Nanotechnology*, 16:888–900, 2005.

[101] J. Han. *Fault-Tolerant Architectures for Nanoelectronic and Quantum Devices*. Ph.D. thesis, Delft University of Technology, Delft, The Netherlands, 2004.

[102] M. Barr. Programmable logic: What's it to ya? *Embedded Systems Programming magazine*, pages 75–84, June 1999. Also available online at `http://www.embedded.com/1999/9906/9906sr.htm`.

[103] S. Trimberger, editor. *Field Programmable Gate Array Technology*. Kluwer Academic Publishers, Boston, 1994.

[104] B. Zeidman. *Designing with FPGAs and CPLDs*. CMP Books, Lawrence, KS, 2002.

[105] A. DeHon. Reconfigurable architectures for general-purpose computing. Technical Report AITR-1586, Massachusetts Institute of Technology, 1996.

[106] P. S. Zuchowski, C. B. Reynolds, R. J. Grupp, S. G. Davis, B. Cremen, and B. Troxel. A hybrid ASIC and FPGA architecture. In *Proc. Intl. Conf. Comp. Aid. Des.*, pages 187–194, 2002.

[107] M. M. Ziegler and M. R. Stan. CMOS/nano co-design for crossbar-based molecular electronic systems. *IEEE Trans. Nano.*, December 2003.

[108] M. Forshaw, R. Stadler, D. Crawley, and K. Nikolic. A short review of nanoelectronic architectures. *Nanotechnology*, 15(4):S220–S223, 2004.

[109] R. P. McConnell. Diode-based power gain for molecular-scale electronic digital computers. report MP 00W0000310, The MITRE Corporation, McLean, VA, 2000.

[110] R. P. McConnell, J. C. Ellenbogen, T. S. Mayer, T. E. Mallouk, and 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), December 2000.

[111] S. C. Goldstein and D. Rosewater. Digital logic using molecular electronics. In *Proc. Intl. Sol. St. Circ. Conf.*, 2002.

[112] G. S. Rose and M. R. Stan. Memory arrays based on molecular RTD devices. In *Proc. IEEE-NANO*, pages 453–456, 2003.

[113] E. Goto, K. Murata, K. Nakazawa, K. Nakagawa, T. Moto-Oka, Y. Matsuoka, Y. Ishibashi, T. Soma, and E. Wada. Esaki diode high speed logical circuits. *IRE Trans. Elect. Comp.*, pages 25–29, 1960.

[114] H. C. Liu and T. C. L. G. Sollner. High-frequency resonant-tunneling devices. In R. A. Kiehl and T. C. L. G. Sollner, editors, *Semiconductors and Semimetals*, volume 41, pages 359–418. Academic Press, Boston, 1994.

[115] R. H. Mathews, J. P. Sage, T. C. L. G. Sollner, S. D. Calawa, C.-L. Chen, L. J. Mahoney, P. A. Maki, and K. M. Molvar. A new RTD-FET logic family. *Proc. IEEE*, 87(4):596–605, April 1999.

[116] P. J. Kuekes. Molecular crossbar latch. United States Patent 6,586,965, 2003.

[117] Y. Chen, D. A. A. Ohlberg, X. Li, D. R. Stewart, J. O. Jeppesen, K. A. Nielsen, J. F. Stoddart, D. L. Olynick, and E. Anderson. Nanoscale molecular-switch devices fabricated by imprint lithography. *Appl. Phys. Lett.*, 82(10):1610–1612, March 2003.

[118] G. Snider, P. Kuekes, T. Hogg, and R. Stanley Williams. Nanoelectronic architectures. *Appl. Phys. A*, 80:1183–1195, 2005.

[119] G. S. Snider and P. J. Kuekes. Nano state machines using hysteretic resistors and diode crossbars. *IEEE Trans. Nano.*, 5:129–137, 2006.

[120] M. van den Brink. Litho roadmap shows difficult terrain - part 2 - technology information. *Electronic News*, Jan. 17, 2000.

[121] Semi industry to reach $360 billion by 2010, says report. *Silicon Strategies*, Dec. 2, 2003.

[122] A. W. Burks, H. H. Goldstine, and J. von Neumann. Preliminary discussion of the logical design of an electronic computing instrument. In A. H. Taub, editor, *John von Neumann Collected Works*, volume V, pages 34–79. The Macmillan Co., New York, 1963.

[123] H. H. Goldstine and J. von Neumann. On the principles of large scale computing machines. In A. H. Taub, editor, *John von Neumann Collected Works*, volume V, pages 1–32. The Macmillan Co., New York, 1963.

[124] J. von Neumann. First draft of a report on the EDVAC. In N. Stern, editor, *From ENIAC to Univac: An Appraisal of the Eckert-Mauchly Computers*. Digital Press, Bedford, MA, 1981.

[125] Reprinted from the AMD Virtual Pressroom at `http://www.amd.com`.

[126] J. D. Meindl, Q. Chen, and J. A. Davis. Limits on silicon nanoelectronics for terascale integration. *Science*, 293(5537):2044–2049, 2001.

[127] V. V. Zhirnov, R. K. Cavin, J. A. Hutchby, and G. I. Bourianoff. Limits to binary logic switch scaling – a gedanken model. *Proc. IEEE*, 91(11):1934–1939, November 2003.

[128] D. J. Frank, R. H. Dennard, E. Nowak, P. M. Solomon, Y. Taur, and H. S. P. Wong. Device scaling limits of Si MOSFETs and their application dependencies. *Proc. IEEE*, 89(3):259–288, 2001. This article is one of several that appeared in a Special Issue on Limits of Semiconductor Technology.

[129] M. T. Bohr. Nanotechnology goals and challenges for electronic applications. *IEEE Trans. Nano.*, 1(1):56–62, March 2002.

[130] J. A. Davis, R. Venkatesan, A. Kaloyeros, M. Beylansky, S. J. Souri, K. Banerjee, K. C. Saraswat, A. Rahman, R. Reif, and J. D. Meindl. Interconnect limits on gigascale integration (GSI) in the 21st century. *Proc. IEEE*, 89(3):305–324, March 2001. This article is one of several that appeared in a Special Issue on Limits of Semiconductor Technology.

[131] M. Ieong, B. Doris, J. Kedzierski, K. Rim, and M. Yang. Silicon device scaling to the sub-10-nm regime. *Science*, 306:2057–2060, 2004.

[132] A. Rahman. *System-Level Performance Evaluation of Three-Dimensional Integrated Circuits*. Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, MA, 2001.

[133] R. Reif, A. Fan, K.-N. Chen, and S. Das. Fabrication technologies for three-dimensional integrated circuits. In *Proc. Intl. Symp. Qual. Elect. Des.*, pages 33–37, 2002.

[134] S. F. Al-Sarawi, D. Abbott, and P. D. Franzon. A review of 3-D packaging technology. *IEEE Trans. CPMT B*, 21(1):2–14, 1998.

[135] A. Fan, A. Rahman, and R. Reif. Copper wafer bonding. *Elect. Sol. St. Lett.*, 2(10):534–536, 1999.

[136] J. A. Burns, C. Keast, K. Warner, P. Wyatt, and D. Yost. Fabrication of 3-dimensional integrated circuits by layer transfer of fully depleted SOI circuits. In *Proc. Mat. Res. Soc. Symp. G*, volume 768, April 2003.

[137] Y. Kwon, A. Jindal, J. J. McMahon, J.-Q. Lu, R. J. Gutmann, and T. S. Cale. Dielectric glue wafer bonding for 3-D ICs. In *Proc. Mat. Res. Soc.*, Spring 2003.

[138] L. Xue, C. C. Liu, H. S. Kim, S. Kim, and S. Tiwari. Three-dimensional integration: Technology, use, and issues for mixed-signal applications. *IEEE Trans. Elect. Dev.*, 50(3):601–609, 2003.

[139] V. Subramanian, P. Dankoski, L. Degertekin, B. T. Khuri-Yakub, and K. C. Saraswat. Controlled two-step solid-phase crystallization for high-performance polysilicon TFT's. *IEEE Elect. Dev. Lett.*, 18(8):378–381, 1997.

[140] S. Das. *Design Automation and Analysis of Three-Dimensional Integrated Circuits*. Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, MA, 2004.

[141] R. Chau, B. Boyanov, B. Doyle, M. Doczy, S. Datta, S. Hareland, D. Jin, J. Kavalieros, and M. Metz. Silicon nanotransistors for logic applications. *Physica E: Low-dimensional systems and nanostructures*, 19(1-2):1–5, 2003.

[142] S. J. Wind, J. Appenzeller, R. Martel, V. Derycke, and P. Avouris. Vertical scaling of carbon nanotube field-effect transistors using top gate electrodes. *Appl. Phys. Lett.*, 80(20):3817–3819, 2002.

[143] Q. L. Li, S. Surthi, G. Mathur, S. Gowda, Q. Zhao, T. A. Sorenson, R. C. Tenent, K. Muthukumaran, J. S. Lindsey, and V. Misra. Multiple-bit storage properties of porphyrin monolayers on SiO_2. *Appl. Phys. Lett.*, 85(10):1829–1831, 2004.

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

[145] D. B. Strukov and K. K. Likharev. A reconfigurable architecture for hybrid CMOS/nanodevice circuits. In *Proc. ACM/SIGDA FPGA*, Monterey, CA, 2006. ACM Press.

[146] 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, and B. R. Ratna. Cowpea mosaic virus as a scaffold for 3-D patterning of gold nanoparticles. *Nano Lett.*, 4(5):867–870, 2004.

[147] J. Y. Fang, C. M. Soto, T. W. Lin, J. E. Johnson, and B. Ratna. Complex pattern formation by cowpea mosaic virus nanoparticles. *Langmuir*, 18(2):308–310, 2002.

[148] P. Beckett and A. Jennings. Towards nanocomputer architecture. In *Proc. ACS Conf. Res. Prac. Inf. Tech.*, volume 6, pages 141–150, 2002.

[149] A. DeHon, P. Lincoln, and J. E. Savage. Stochastic assembly of sublithographic nanoscale interfaces. *IEEE Trans. Nano.*, 2(3):165–174, 2003.

[150] H. Naeimi and A. DeHon. A greedy algorithm for tolerating defective crosspoints in nano PLA design. In *Proc. IEEE Intl. Conf. Field Prog. Tech.*, 2004.

[151] N. H. E. Weste and D. Harris. *CMOS VLSI Design: A Circuits and Systems Perspective*. Addison-Wesley Publishing Company, Reading, MA, 3rd edition, 2004.