Molecular Electronics: From Devices and Interconnect to Circuits and Architecture

MIRCEA R. STAN, SENIOR MEMBER, IEEE, PAUL D. FRANZON, SENIOR MEMBER, IEEE, SETH COPEN GOLDSTEIN, MEMBER, IEEE, JOHN C. LACH, MEMBER, IEEE, AND MATTHEW M. ZIEGLER, MEMBER, IEEE

Invited Paper

As the dominating CMOS technology is fast approaching a "brick wall," new opportunities arise for competing solutions. Nanoelectronics has achieved several breakthroughs lately and promises to overcome many of the limitations intrinsic to current semiconductor approaches. Most of the results in this area reported until now focus on devices and interconnect; this work goes several steps further and presents issues related to circuits and architecture. Based on proposed nanoscale interconnect and device structures, we explore the design space available to the nanoelectronic circuit designer and system architect.

Keywords*—*Crossbar architectures, defect tolerance, molecular electronics, nanocircuits, nanodevices, nanoscience, nanotechnology, nanotubes.

I. INTRODUCTION

While traditional silicon electronics should continue industrial dominance for the next decade, novel nanoelectronic solutions will be needed to surmount the physical and economic barriers of current semiconductor technologies and continue along the exponential projections of Moore's Law. Although most new nanoelectronic solutions are still in their infancy, they present the potential for unprecedented levels of device density, low power computing, and tight integration with other biological and chemical functions such as sensors. Recent progress in molecular nanoelectronics provides hope that functional large-scale nanocircuits will be viable in the not too distant future. Common to many of these emerging nanotechnologies is the assumption that some form of self-assembly will be required to fabricate nanoscale circuits. Until recently, the focus has been mainly on materials and device science and engineering, but recent work suggests that the field is ready to undertake the challenge of integrating numerous devices into functional circuits. Viable concepts for molecular integrated circuits are starting to appear [1]–[3].

Just as the design of modern CMOS circuits relies heavily on abstraction due to increased design complexity, so must nanoelectronic-based circuit design. Abstraction is used to allow for design entry at the gate level, the logic level, and on up to the architecture level. This abstraction-based design methodology has been successful primarily due to accurate modeling, which allows for simulation at different levels. A high-level design can be validated without going through the lengthy computer-aided design (CAD) processes of deabstraction to the physical design level. For this methodology to be effective for nanotechnology-based circuit and system design, accurate models must be developed up through the abstraction hierarchy. Not only must individual device characteristics be modeled, but also their behavior when interacting with other devices to form gates and so on up the hierarchy.

Such models are necessary not only to ensure proper functionality via reliable high-level simulation but also to establish accurate cost functions for all relevant figures of metrics. Area, performance, power, reliability, and other metrics must be accurately modeled to enable synthesis, mapping, and physical design algorithms to implement circuits of minimal

Manuscript received October 10, 2002; revised May 18, 2003. The work of M. R. Stan and M. M. Ziegler was supported in part by the National Science Foundation (NSF) under Grant 0210585 and by a UVa FEST award. The work of P. D. Franzon was supported by the Defense Advanced Research Projects Agency and the Office of Naval Research under Contracts N00014-01-1-0657 and F29601-00-D-0244. The work of S. C. Goldstein was supported in part by grants from DARPA. The work of J. C. Lach was supported by the NSF under Grant CCR-0105626.

M. R. Stan, J. C. Lach, and M. M. Ziegler are with the Electrical and Computer Engineering Department, University of Virginia, Charlottesville, VA 22904-4743 USA (e-mail: mircea@virginia.edu; jlach@virginia.edu; ziegler@virginia.edu).

P. D. Franzon is with North Carolina State Univeristy, Department of Electrical and Computer Engineering, Raleigh, NC 27695-7911 USA (e-mail: paulf@ncsu.edu).

S. C. Goldstein is with the School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213-3891 USA (e-mail: seth@cs.cmu.edu).

Digital Object Identifier 10.1109/JPROC.2003.818327

cost, as evaluated by the cost function. These models are essential for designers to establish the proper cost function based on the metrics of greatest importance to their design goals.

The purpose of this paper is to explore approaches to designing molecular electronics above the device level. Intentionally we look mainly at the class of electronic nanoscale solutions that are amenable to self-assembly, which for the time being are mostly limited to two-terminal devices.[1] Questions addressed include the following. How is molecular computing technology viewed from a circuit and technology perspective? How are the devices modeled and treated at circuit-level abstractions? How can digital logic circuits be built from available devices? What are the logic architectures that are potentially useful given the nature of the underlying technology? This paper is organized to address these questions in the sequence described.

II. Nano Fabrication

The differences between molecular electronics and traditional CMOS are nowhere greater than in how the circuits will be fabricated. Typical molecular switches are typically only 2 nm long. They are synthesized in a test tube, rather than defined using dopants or epitaxy on solid state materials. Therefore, fabricating integrated circuits (ICs) out of molecular electronic devices requires a very different approach than is used in conventional IC fabrication.

The most interesting of the proposals, at least in long-term potential, are aimed at assembling circuits using bottom-up manufacturing, as opposed to today's top-down approaches. Bottom-up manufacturing is a hierarchical approach that first creates the individual components and then assembles them together into ever larger structures. This approach has the potential advantage of reducing the manufacturing precision required, thus reducing the cost of creating chips. Furthermore, it appears that it may be the only method which can scale both in the number of components assembled together and in the sizes of the individual components. It has the drawback that, unlike photolithography, it may not be able to create the components and their connections simultaneously. In addition, the hugely complex and highly controlled patterns enabled by photolithography will likely be impossible with nanoelectronic fabrication techniques.

The first step in bottom-up manufacturing is to create the devices and wires. There has been tremendous progress in this area. Researchers have created a variety of interesting molecular devices, including resonant-tunneling diode (RTD) devices [4], [5], programmable molecular switches [6], carbon nanotube transistors, [7] and diodes [8], to name but a few. Wires which have diameters of only a few nanometers have also been fabricated: single-crystal nanowires [9]–[12] and carbon-nanotube wires [13] are but a few of the many conductors that have been created. Some nanoscale wires can be used as more than just conductors, they can also be used as active devices [7], [14], [15].

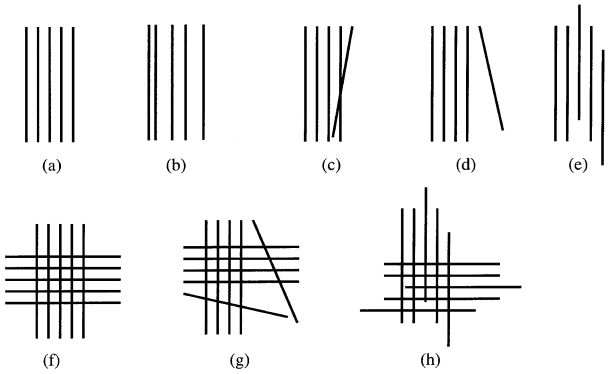


Fig. 1. Different scenarios using flow based assembly.

Bottom-up, scalable integration techniques include Langmuir–Blodgett films, flow-based alignment, nanoimprinting, random assembly, biologically assisted assembly, self-assembled monolayers, and catalyzed growth. The common feature among all these techniques, with the possible exception of nanoimprinting,[2] is that they can only form random or regular structures. An additional characteristic is that the resulting structures usually contain some defects; i.e., the results are not perfect. Finally, it is not possible to predetermine exactly where a particular element will be located in the structure. That is, one cannot deterministically assemble an array of different types of wires into a particular aperiodic pattern.

To date, the most complex structures made using scalable bottom-up techniques are two-dimensional (2-D) meshes of wires.[3] For example, fluidic assembly was used to create a 2-D mesh of nanowires [11]. The nanowires are suspended in a fluid which flows down a channel. The nanowires align with the fluid flow and occasionally stick to the surface they flow over. The average spacing between wires and the wire density can be controlled by varying the rate and duration of the flow. Further refinements can be made by patterning the underlying substrate with different molecular end-groups to which the nanowires will show preferential binding. By performing the flow operation twice, first in one direction and then in the orthogonal direction, arrays of wires at right angles can be formed.

The above method points out the possibilities and the limitations of bottom-up assembly. On the positive side, one can form one-dimensional arrays and 2-D meshes of nanowires [Fig. 1(a) and (f)]. Drawbacks to this method include its imprecision and lack of determinism. Wires will rarely all be equidistant from each other [Fig. 1(b)]. Wires that should be parallel may intersect [Fig. 1(c)] or be askew [Fig. 1(d)]. The wires may be parallel, but may be offset from each other

[1]Note that there are also some organic electronic devices with sizes much larger than the nanometer range. Such large devices, while useful in special low-cost applications, will automatically lead to low levels of integration and are not further considered in this paper.

[2]Nanoimprinting has been used successfully to create lithographic-like structures. However, the process of creating masters combined with direct contact printing may limit the smallest achievable pitch [16]. Further, it will difficult to precisely align the multiple masks.

[3]An exception exists for the recent advances in DNA-based assembly, e.g., [17]–[19] DNA-based assembly has been known to create ordered structures with heterogeneous materials. However, DNA-based methods are even less developed than the methods we explore here.

[Fig. 1(e)]. Finally, it is not possible to create a predetermined, aperiodic pattern of different types of wires. The resulting arrays may contain shorts [Fig. 1(c)] or open connections [Fig. 1(g) and Fig. 1(h)]. Finally, the technique can never deterministically produce complex aperiodic arrangements of the wires.

Another approach for creating crossbars is to use nanoimprinting, in which a master is made using techniques such as electron-beam lithography. The result is a stamp that has ridges and trenches. When the master is pressed onto a target surface, usually coated with a soft material, such as plastic deformation magnetic assembly (PDMA), the result is that the pattern on the master is transferred to the target. The resulting surface can be then be functionalized. For example, the trenches on the target can be filled with metal to create wires. Researchers at Hewlett-Packard (HP) have recently used this technique to make a 64-b molecular memory. They use the master to create parallel wires in one direction, fill the trenches with wires, coat the surface with a molecular switch, and then rotate the master by 90° to make a second imprint. The result is a crossbar made with 40-nm-wide wires at a pitch of slightly more than 100 nm. At each junction between two wires, there are approximately 1000 molecules which can serve as a bit of memory.

A. Implications on Circuits and Architectures

The use of self-assembly as the dominant means of circuit assembly imposes the most severe limitations on nanoscale architectures: it will be difficult to create either precise alignment between components or deterministic aperiodic structures. Chemical self-assembly, as a stochastic process, will not always produce precise alignment of structures, and manipulation of single nanoscale structures to construct large-scale ordered circuits, as currently built in silicon, is impractical. Furthermore, the methods used to assemble nanoscale components are most effective at creating random, or, at best, crystal-like structures. These two facts have significant implications on the kinds of circuits and structures that can be realized at the time of fabrication.

The structural implications of bottom-up assembly are as follows.

1) *Two-terminal devices are preferred*. The easiest devices to incorporate into a circuit will be devices with two terminals, e.g., diodes, configurable switches, and molecular negative-differential resistance (NDR) devices. It is much easier to bring two terminals into precise proximity than three.
2) *Connections by overlapping wires*. Lack of precise alignment means that end-to-end connections between groups of nanoscale wires will be nearly impossible to achieve. A scheme where all connections between nanoscale wires occur only when the wires are orthogonal and overlap, reduces the need to precisely align the wires.
3) *Meshes are a common basic unit*. Since practical foreseeable nanocomputer design will have to rely on the placement of active components at wire intersections, many concepts rely on using a crossbar array or mesh as a basic repeated unit. More complicated structures will have to be created either by combining meshes together or cutting the wires in a mesh to break it up into subarrays.
4) *Randomness has to be accepted*. Random assembly is a matter of fact. There are different ways of viewing randomness. One is to view it as a defect pattern that has to be coped with, e.g., a crossbar array with a number of shorts and opens in it. Another is to use elements to program around defects or to create higher levels of order—for example, using programmable devices as fuses or antifuses.
5) *Nanoscale to microscale connections will have to be sparse*. A direct implication of the size difference between the nanoscale and the microscale is that connections between the two will have to be few. If there are many connections between the two worlds, the density of the nanoscale components will be dictated by the density of the microscale. For example, using a demultiplexor to connect microscale wires to nanoscale wires would at any particular instant allow one of n nanoscale wires to be addressed using $\log n$ microscale wires. As n grows large, the nanoscale wires, not the microscale wires, dominate the device.

The architectural implications of molecular electronics and self-assembly are as follows.

1) *Fine-grained reconfigurable*. The most likely assembly processes are best at creating crystal-like structures, e.g., 2-D meshes. Thus, the resulting structures cannot directly implement a complex, aperiodic circuit. To create useful aperiodic circuits will require that the device be configured after it is manufactured.
2) *Defect tolerance*. The stochastic process behind molecular self-assembly will inevitably give rise to defects in the manufactured structures. Instead of defect densities in the range of one part per billion (as one gets in silicon), defect densities for bottom-up assembly may be as high as a few percent. The architecture will have to be designed to include spares, which can be used in place of defective components. Another useful design criterion will be to ensure that when a set of components is assembled into a larger structure, i.e., wires into parallel wires, the individual components are interchangeable. For example, in a set of parallel wires, it should not matter *a priori* which working wire is used to implement a particular circuit. Rather than attempting to eliminate all the defects completely with manufacturing techniques, one can also rely on postfabrication steps that allow the chip to work *in spite of* its defects. A natural method of handling the defects, first used in the Teramac [20], is to design the device to be reconfigurable and then exploit its reconfigurable nature. After fabrication, the device can be configured to test itself, the result of testing being a map of the device's defects. The defect map can then be used to configure the device to

implement a particular function in a way that avoids or incorporates the detected defects.

3) *Locality*. As devices and wires scale down to molecular dimensions, the wires become an increasingly important part of the total design. This is true not only for molecular computing, but also for end-of-the-roadmap CMOS [21]. Architectures with significant locality (and, thus, the ability to communicate most frequently over shorter wires), will have an advantage.

III. Nanodevices and Models

We begin our discussion of molecular electronics abstractions by examining molecular device modeling. The push to scale electronics to nanometer dimensions is bringing new device phenomena to the forefront. Experiment and theory show nanoscale devices may exhibit nonclassical characteristics due to electron energy discreteness, electron tunneling, and Coulomb blockade effects. Although some of these phenomena may be considered parasitic in conventional devices, it is possible that nanoelectronic circuits and systems may achieve greater performance through the utilization of such nonclassical behavior. As discussed above, modeling the behavior of nanoscale devices in a manner that allows complex circuits to be simulated in a reasonable amount of time requires new approaches to device modeling. Furthermore, with the field of nanoelectronics being in such an infant stage, there have been many new devices proposed. The ability to quickly and easily create robust device models will aid device designers and circuit designers in evaluating the applicability of new devices.

Many efforts are underway for developing solid-state models for nanoscale systems [22], [23]. However, the majority of the higher level work in molecular nanoelectronics has for the most part skipped the circuit level and focused on architectures [24]–[27]. Thus, modeling molecular devices at the level of abstraction needed for circuit design requires a jump start to fill the gap between the device level work and the architectural proposals.

The conventional method for modeling devices for use in circuit simulation, such as MOSFETs, incrementally builds on a well-established model framework. However, newly designed nanoscale devices do not have this legacy and need to be modeled from the ground up. Furthermore, because the nonclassical characteristics observed in these devices and the underlying physics are not yet completely understood, it is difficult to develop compact physics-based models for all the devices that require evaluation at the circuit level.

We classify underlying modeling approaches as physics-based, component-based, and empirical models. Previous work on nonclassical device models for SPICE simulation includes models for RTDs [22], [28] and single-electron transistors (SETs) [23]. The RTD models mentioned above exhibit the two extremes for modeling and implementation approaches. Bhattacharya and Mazumder [22] have developed a physics-based RTD using the compact RTD equations presented in [29]. This compiled model was added to the open-source Berkeley SPICE simulator. The advantages of this model [22] include fast and robust simulation, since it is compiled with the simulator and contains underlying equations based on physical parameters. However, the downsides include a difficult implementation and reduced model flexibility. On the other hand, Yan and Deer [28] present an empirical interpreted model added to PSPICE. While this model provides an easy implementation, which increases flexibility, it incurs increased simulation time because it must be interpreted. Furthermore, the empirical nature of the model loses any connection to the underlying physical structure of the device. The SET "macromodel" presented in [23] corresponds to what we call a component-based approach, since the complete model is built from multiple components with simple behavior.

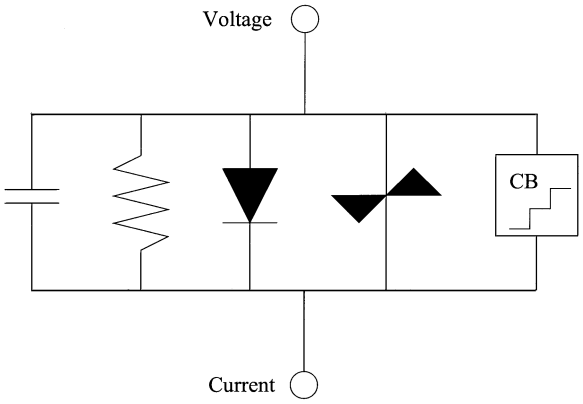


Fig. 2. Two-terminal UDM with parameterized contributions for a variety of fundamental effects (capacitance, resistance, themionic emission, NDR, and Coulomb blockade).

While previous solutions for nanoelectronics have demonstrated the many options in device modeling, the variety of emerging nanoscale devices without compact physical models calls for a robust modeling methodology capable of developing models in an automated fashion. Based on the fundamental classical and quantum phenomena in nanoscale devices, we proposed a generic universal device model (UDM) for two-terminal devices [30] that captures the device behavior such that circuit design and simulation become possible at the nanoscale level.

Our concept for the UDM uses empirical equations that describe each fundamental quantum and classical effect that may be relevant to an electronic device. Since the aim of this model is to promote the design and simulation of circuits containing nanoscale devices, the fundamental effects are represented in terms of their current versus voltage (I–V) characteristics. The properties currently included in the model are resistance, thermionic emission (diode-like behavior), resonant tunneling (RTD-like behavior), and Coulomb blockade (SET-like behavior) effects. Implicit resistors in series with each of the four parallel effects are also included to model imperfect interfacing and linear slopes in the I–V curve.

With the various parts of the UDM defined, the current can be obtained from the combination of all contributing phenomena [30], as can be seen in Fig. 2. There are four "basis" functions, each representing a separate physical phenomenon, which construct an overall dc model that typically is a combination of multiple such phenomena (e.g., the RTD

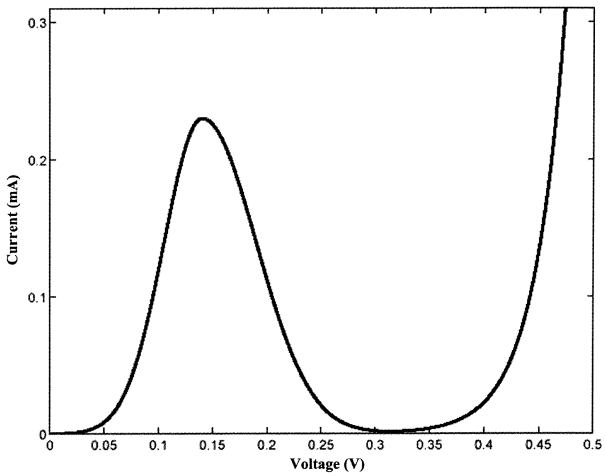


Fig. 3. Generic *I–V* characteristic of an RTD as the combination of a tunneling effect with a thermionic effect. A positive differential resistance (PDR) region is followed by an NDR region with an underlying exponential increase in the base current.

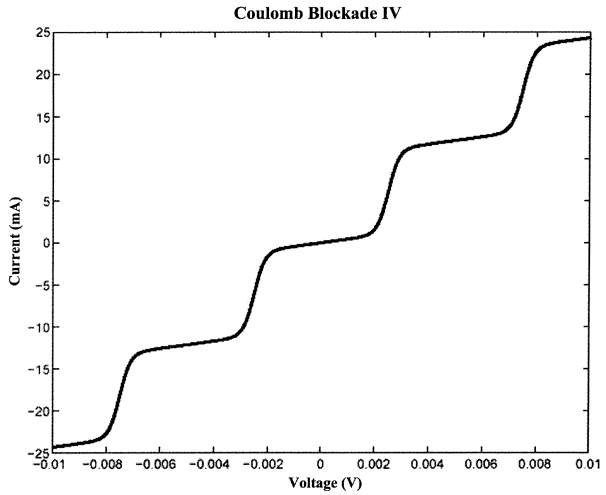


Fig. 4. Generic *I–V* characteristic of a Coulomb blockade device as a step-like behavior with a an implicit resistive effect that provides the slope in between the steps.

in Fig. 3 is formed by combining tunneling and thermionic functions, while the Coulomb blockade in Fig. 4 uses a single function). In addition to the parameters of each individual function, weights are assigned to each phenomenon to control the magnitude of the contribution, enabling the UDM to mimic almost any two-terminal device. The current equation including all of the contributing factors and their respective weights is given by

$$I_{\mathrm{UDM}}(V) = \alpha_R I_R(V) + \alpha_D I_D(V) + \alpha_T I_T(V) + \alpha_{CB} I_{CB}(V) \quad (1)$$

One goal of the UDM is to provide a method for rapidly using a device with characterized behavior in circuit simulation without investing time in coding a device model or modifying a simulator. The input required for extracting the UDM parameters is a file containing a set of current and voltage data points, i.e., points along the *I–V* curve. The data file is fed to a UDM parameter extraction tool. The extracted parameters are then used during simulation to customize the UDM to the desired device. The UDM can, thus, be used to model devices characterized either theoretically or experimentally, providing the *I–V* curve is described by a set of data points.

Fig. 5 shows the device extraction flows using the UDM for both of these approaches. The upper portion of Fig. 5 shows the theoretical design flow, whereas the lower portion of the figure depicts the experimental design flow. The *I–V* curve in Fig. 5 from the theoretical design flow was obtained using Purdue's Huckel-IV simulator [31], while the *I–V* shown in the experimental design flow was recreated from the molecular RTD in [32]. The key aspect is that the interface to the UDM parameter extraction tool is an *I–V* curve. The right portion of the figure illustrates that a schematic can contain different devices modeled by the UDM either from theoretical or from experimental data. The schematic shown in Fig. 5 is a design for an XOR gate [33]. We use this example simply to illustrate the UDM design flows, as the particular device characteristics in the figure would not yield a functional gate.

In addition to nonclassical behavior, many nanoelectronic devices exhibit hysteretic *I–V* characteristics. Hysteresis in terms of nanoscale devices typically refers to a device's ability to switch between stable behaviors. Such devices may ultimately provide the means for memory and programmable logic [3], [34]. Modeling hysteresis with the UDM requires a data set for each stable *I–V* characteristic as well as a set of toggle definitions. A toggle definition is an action (e.g., an applied bias) that causes the behavior to switch between stable points. When modeling a bistable device, the UDM will initially reference one parameter set and then switch to the alternate parameter set when a toggle condition occurs.

Simulators that support cosimulation of both SPICE-level netlists and analog hardware description language (analog HDL) blocks easily adapt to simulating hysteretic devices as well as conventional device models. Thus, we use a UDM model implemented in Verilog-A with the Spectre circuit simulator from Cadence [35] to simulate circuits containing hysteretic devices like the crossbar circuits in Section IV-C.

A simple example of the UDM used in circuit simulation is a latch consisting of two RTDs as shown in Fig. 6(a) [36], [37]. In the circuit, one RTD works as the load device, while the other drives it, based on the bias voltage *CLK*. Due to the PDR and NDR regions of the RTDs, the circuit effectively has three stable points. Varying the bias voltage *CLK* and the input current causes the circuit to settle in either the high or low stable states. The process by which the RTD pair evolves toward one of these two states is known as the monostable–bistable transition (MBT).[4] As can be seen in Fig. 6(b), the circuit latches the input voltage after the bias is pulsed low. Since the circuit is driven by the input current, the resistance at the input had to be chosen carefully to obtain the desired results. Another useful property of the RTD latch can be seen from simulating several cascading latches as in Fig. 6(c). Fig. 6(d) shows the output waveform of such a cascaded RTD latch that demonstrates gain, since the output

[4]A detailed description of how this device works can be found in [36].

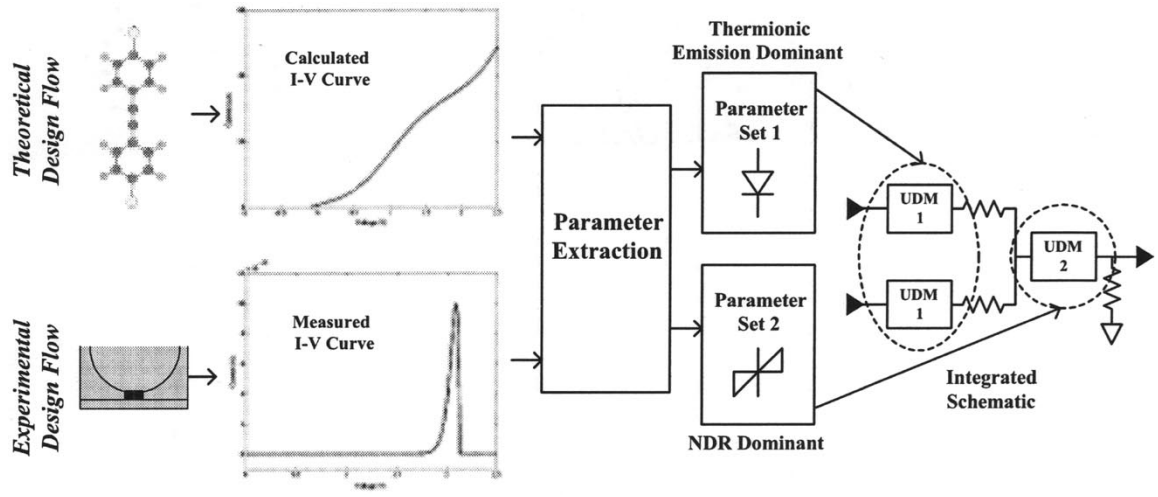


Fig. 5. Circuits containing multiple device types can be designed with the UDM. The devices can be characterized either theoretically or experimentally, providing the behavior of the devices is presented in *I–V* curve data sets.

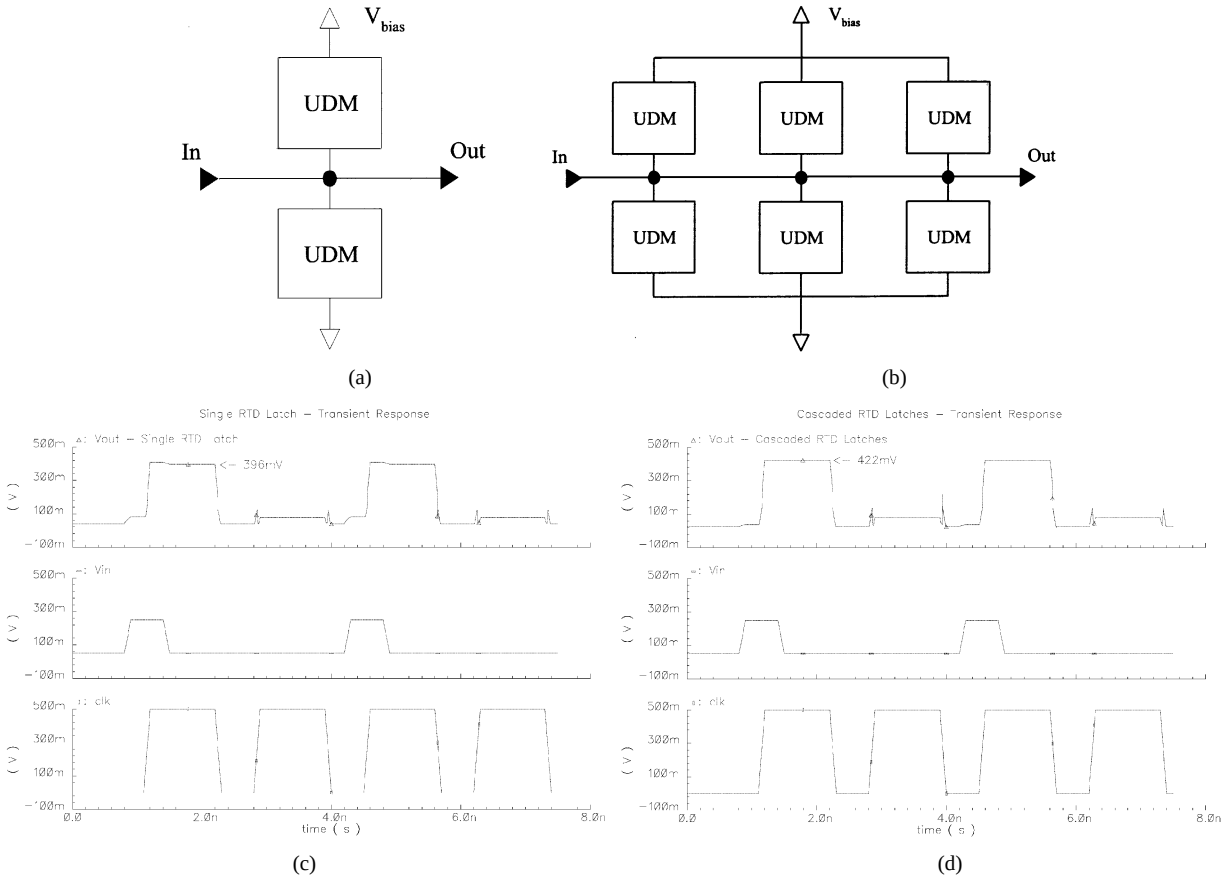


Fig. 6. Schematics and simulations for RTD latch circuits using the UDM. (a) Single RTD latch schematic. (b) Cascaded RTD latches schematic. (c) Simulation of a single RTD latch showing voltage gain at Vout. (d) Simulation of the cascaded RTD latches showing increased voltage gain over the single latch.

levels settle closer to the stable points after each level of the cascade.

The RTD latch simulated using the UDM behaves as expected demonstrating that the model works correctly for both tunneling and diode-like devices. More work needs to be done to verify the other properties included in the model; single-electron devices and circuits which utilize phenomena such as the Coulomb blockade are currently being investigated.

IV. Nanocircuits and Logic

A number of challenges exist in turning molecular devices and possible integration schemes into useful circuit struc-

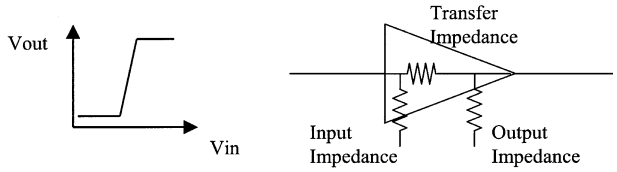


Fig. 7. Digital circuits require the ability to tolerate noise and support fan-out. These requirements are met by designing the circuits with nonlinear transfer gain characteristics, and appropriate equivalent impedances.

tures. In this section, the nature of those challenges and some potential approaches are explored.

A. Digital Circuit Requirements

Scalable digital circuits, especially those based on MOSFETs, have a number of useful properties that must also exist in molecular digital circuits for the latter to be useful. As Keyes pointed out in 1985 [38], digital circuits have been successful due to a number of well-established reasons.

1) *Noise tolerance*. Unlike analog circuits, which always add noise to input signals, digital circuits reject noise. They can typically reject noise voltages up to 10%–20% of the normal voltage swing. The ability to reject noise arises because of the nonlinear gain characteristic from input to output as shown in Fig. 7. It is the presence of this nonlinear gain characteristic that enables digital circuits to be scaled to very high integration levels.
2) *Ability to support fan-out*. Fan-out refers to the number of similar circuits that can be driven by a logic gate without a significant degradation in noise tolerance. High fan-out requires that any circuit has high input impedance, low output impedance and high transfer impedance (Fig. 7.) The latter is sometimes referred to as input–output isolation.

Furthermore, in order to potentially replace CMOS digital circuits, it is important for molecular circuits to exceed the capabilities of future CMOS circuits in one or more key metrics. As a CMOS reference, this paper uses the capabilities expected at the "end of the roadmap," 2016 for 10-nm gate length CMOS devices, referred to as the 22-nm "node" (or half the smallest wire pitch).[5] The metrics include the following.

1) *Small size*. Due to its use of self-aligned lithography and low transistor count per logic gate, CMOS achieves the lowest area footprint per gate of any current semiconductor device family. At the 22-nm node, dynamic RAM is expected to have a density of 5×10^{10} b/cm^2, static RAM at 7×10^{9} transistors/cm^2, and logic an average density of 1×10^{9} transistors/cm^2.
2) *High switching speed*. In a CMOS digital circuit the logic transition time, and, thus, the clock frequency, is determined by the product of the output current of the gate and its input capacitance. Traditionally, as devices scale down in dimensions, the output current has remained fairly constant while the input capacitance has decreased, thus enabling ever faster clock speeds. One useful speed metric is the delay of a gate with a fan-out of 4 (FO4), typically employed at the logic level. The intrinsic switching speed time constant is

$$\tau = C_{\mathrm{gate}} * \frac{V_{\mathrm{dd}}}{I_{\mathrm{ds-sat}}} \tag{2}$$

(here, C_{gate} is the gate capacitance, V_{dd} is the supply voltage, and I_{ds}-sat is the device saturation current) for an NMOS device, useful for device evaluation.[6]
3) *Low power*. CMOS digital circuits dominate today largely because they can perform any function with less energy than any other manufacturable device family. This has obvious importance for battery operated applications but is also very important for wall-powered systems, as the ability to dissipate heat is always finite. To a first order, in a CMOS logic gate, the energy consumed during a transition is

$$E = \frac{1}{2}C_{\mathrm{gate}}V_{\mathrm{dd}}^2 + \tau I_{\mathrm{leakage}}V_{\mathrm{dd}} \tag{3}$$

where I_{leakage} is the leakage current.

Useful power metrics include device switching energy (CV_{dd}^2), and static device standby power $(I_{\mathrm{leakage}}V_{\mathrm{dd}})$. For anticipated 10-nm gate length high-performance transistors, these metrics can be calculated as 2 pJ and 1.1×10^{-7} W, respectively.

A very useful metric that combines the last two items above, and, additionally prevents "cheating" by (uselessly) improving one metric at the expense the other, is the *energy-delay product*. For a device, this can be measured as the product of the intrinsic switching speed time constant and the device switching energy, around 10^{-29} J $\cdot$ s for a high-speed 10-nm device.

Molecular circuits face a number of challenges in meeting, or exceeding, these metrics based on the limitations of nanofabrication described earlier. Many of these challenges come about due to the limitations of using two-terminal devices. Two-terminal devices do not naturally result in circuits that have gain and good input/output impedance properties. Current devices can benefit from better on/off ratios, better conductance when on, and larger voltage swings when off, than are currently demonstrated.[7] CMOS achieves low power largely because little current flows from V_{dd} to ground. This result is much more difficult to achieve with two-terminal devices.

B. RTD Circuits

As described above, a latch can be built using a device showing an NDR characteristic, reminiscent of RTDs. The principle of operation of the NDR latch, when used as a memory, is illustrated in Fig. 8. The NDR device is loaded

[5]The International Technology Roadmap for Semiconductors (ITRS) [21] provides a good reference describing anticipated future CMOS capabilities.

[6]The ITRS predicts $\tau = 0.15$ ps for a device with a 10-nm-long high-performance gate.

[7]Improved devices are anticipated, however.

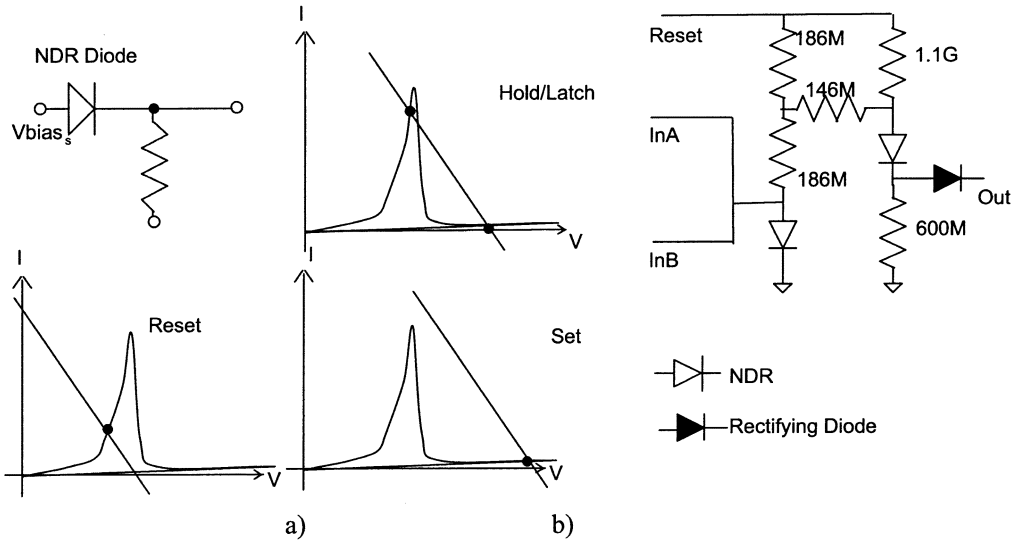


Fig. 8. (a) NDR latch, as used in memory circuits, and principle of operation. (b) NDR-based NAND gate.

with a resistor and, thus, can be understood with a simple load line analysis. The device can be forced into a high or low state by changing the bias voltage. At an intermediate bias, it stays in the previously forced state. The utility of this circuit, and the one shown in Fig. 6, is somewhat limited, mainly by the poor input–output isolation due to the resistor connecting the input and output. This circuit can only support a fan-out of one and has a low noise margin. On the other hand, NDR-based latches that have input–output isolation and a fan-out of greater than one have been simulated [37].

The latch in Fig. 8(a) can be modified as shown in Fig. 8(b) to enable NAND and NOR gates that exhibit suitable digital characteristics [39], [40]. This circuit is based on devices as those demonstrated in [4], [41], and [42], and operates as follows. A clock is applied to the "reset" line; there is no current in the bridge resistor. After the clock (reset) goes low, both NDRs transition to a high-current, low-voltage (low-impedance) state; a small current flows left to right through the bridge resistor. As the input current increases, the voltage across the bridge resistor increases adding more current to the second NDR stage. The resistors are sized such that the input threshold current is at a point which results in a bridge current that forces a transition of the second stage NDR to a low-current, high-voltage (high-impedance) state.

Once both NDRs have transitioned, the voltage across the bridge resistor is zero; therefore, no current flows in either direction. This two-stage isolation circuit helps to prevent the propagation of noise through the gate. All the devices in this circuit, including the resistors and rectifying diode have been demonstrated in a molecular form. From a circuit perspective, this gate is quite practical. It supports high fan-outs and has good noise immunity. Its speed is determined by the conductivity of the NDR gates and, thus, gets faster as the device characteristic improves. With currently available devices, operation at about 1 MHz is expected.

However, the gate also has some limitations that will restrict its utility. The energy-delay product is worse than for CMOS by a few orders of magnitude. It has a relatively high device count of eight per gate, compared to four for the equivalent CMOS gate. Its biggest limitation is that it is a very structured circuit and, thus, almost impossible to integrate given current nanofabrication techniques.

This leads to the recognition of a fundamental tradeoff in molecular circuits, based on anticipated devices. The available integration technologies lead naturally to structures like the crossbar, which in turn, lead to the use of simple latch type circuits constructs, such as shown Figs. 6 and 8(a). However, these latches cannot support a fan-out greater than one, and have low noise margins. Thus, only point-to-point circuits can be used (between crossbars) and high levels of transient faults can be anticipated. Also, the size of possible array structures is going to be limited, as described in the next section.

C. Crossbar Circuits

As described earlier, the potential and limitations of self-assembly lead naturally to regular crossbar architectures. A number of architectures based on crossbar circuits employing two-terminal devices have been recently suggested for memory and logic [25], [26], [43], [44]. Fig. 9 shows an abstract representation of such a crossbar, consisting of two sets of parallel nanowires crossing perpendicularly; the wire crossings form junctions with hysteretic properties. One possible realization of such junctions is an electrically configurable monolayer of bistable molecules, such as the rotaxane and catenane families of molecules being pursued by HP [3], [45] and the University of California, Los Angeles (UCLA) [44]. Another conceptually similar implementation is magnetic RAM (MRAM), which relies on electron spins for bistability [46]. The electromechanical manipulation of carbon nanotubes [47] and

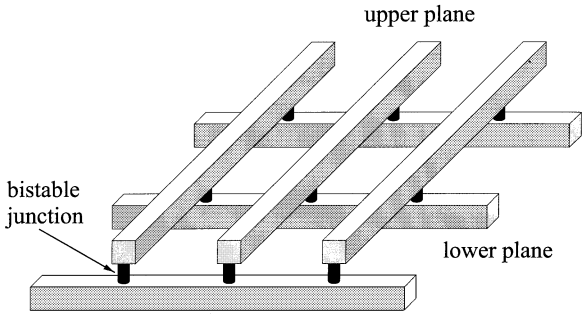


Fig. 9. The crossbar paradigm consists of perpendicular sets of parallel wires with bistable junctions at each wire crossing.

nonvolatile crossed nanowires [34] demonstrated at Harvard also fits into the crossbar circuit paradigm.

While there have been significant efforts aimed at investigating nanoscale crossbars at the device and architecture levels, little consideration has been given to the intermediate levels of abstraction.

In this section, we examine a crossbar from a circuit design perspective [48] and expose both the strong and weak points of this paradigm. While much of this analysis is applicable to several nanotechnology approaches based on regularly structured circuits, we use the crossbar technology proposed by HP and UCLA for demonstrating ideas at the circuit level [3], [43], [44], [49], [50]. This crossbar technology is composed of arrays of crossed nanoscale wires with molecules present at each junction, forming two-terminal devices that can be electrically configured to behave as low or high resistance diodes. These molecules, such as rotaxanes or cantenanes, create a programmable computing fabric that can be used for memories, logic arrays, etc. Harvard has demonstrated a similar circuit paradigm consisting of crossed nanowire p-n junctions [51] as well as logic gates from crossed nanowire FETs (cNW-FETs) [1].

Molecular crossbar technologies present an opportunity for high computational densities; however, they do suffer from some significant drawbacks. Most of the suggested bistable devices are two-terminal devices having the ability to switch between a low and high resistance state. Rectifying (diode-like) behavior is also generally present in these devices, depending on the technology. Being restricted to a diode–resistor logic style results in an inability to achieve signal gain. The lack of gain will restrict the array size and require interfacing to technology capable of achieving gain for extended computation. Another drawback of diode–resistor logic is the inability to implement an inverter. However, this can be easily overcome either by computing the complement of every function or by incorporating an NDR latch and a clock into the circuit.

The generic circuit we consider is shown in Fig. 10. The figure shows a crossbar designed for crosspoint addressing, with the input vector coming horizontally on the row wires and the output presented vertically on the column wires. While this structure alone is not sufficient for computation, it can serve as a crosspoint array. The crosspoint structure will require a decoder external to the crossbar, such as suggested in [26] and [50].

D. Crossbar Logic and Memory

The crosspoint circuit alone is not sufficient for memory or logic. However, combining a decoder with the crosspoint provides the framework for addressing memory for data storage as well as memory-based logic. Previous decoder schemes suggested for molecular nanoelectronics include a stochastic decoder implementation [49] and a prepatterned nanoscale decoder [26]. Another approach for decoding entails programming the decoder in the crossbar fabric. The advantages of this scheme include a straightforward implementation as well as an inherent defect tolerance, since defective wires can be located before programming and then avoided. Fig. 11 shows an implementation of a full-adder using a decoder programmed into the crossbar. The circuit is essentially a lookup table (LUT) comprised of rows of diode–resistor AND gates connecting to columns of diode–resistor OR gates. This same AND–OR plane arrangement can also be used to implement a programmable logic array (PLA). Since logic implemented in the AND–OR planes does not provide the means for an inversion function, signals and their complements are needed for inputs and possibly outputs. The schematic shows explicit pull-up and pull-down resistors, labeled as $R_{\rm and}$ and $R_{\rm or}$, respectively. The lumped wire resistance of a vertical wire is $R_{\rm wc}$, i.e., the column wire resistance. The horizontal wire resistance is broken into two lumped resistors, the resistance of a row wire in the AND plane $R_{\rm wrAND}$ and the resistance of a row wire in the OR plane $R_{\rm wrOR}$.

We simulated the crossbar adder using the devices models in Fig. 12 to investigate $\Delta V_{\rm out}$ [48]. We expect that a crossbar circuit with this arrangement will have inherent signal integrity problems; we also explore the effect of increasing the supply voltage as a method to boost $\Delta V_{\rm out}$. The results in Fig. 13 are representative of the characteristics for $\Delta V_{\rm out}$. As the figure shows, the output voltage $\Delta V_{\rm out}$ is quite small but the supply voltage is one lever for controlling it. However, various other tradeoffs associated with changing the supply voltage must be taken into account to select the optimal supply voltage, such as signal delay and power consumption.

The programmed decoder is a feasible implementation for logic that requires only the basic assumption that each junction in the crossbar can be individually addressed and programmed. However, for memory, this scheme incurs the problem that programming the crosspoint memory via the decoder may overwrite the programmed decoder. To avoid this scenario, we propose a decoder plane consisting of junctions that are programmed at different voltages than the crosspoint array [48]. We feel such an implementation is feasible, since the borders between the decoder and the crosspoint array are not constricted by nanoscale features. For example, the decoder for a memory could be fabricated to consist of junctions which are programmed at voltages

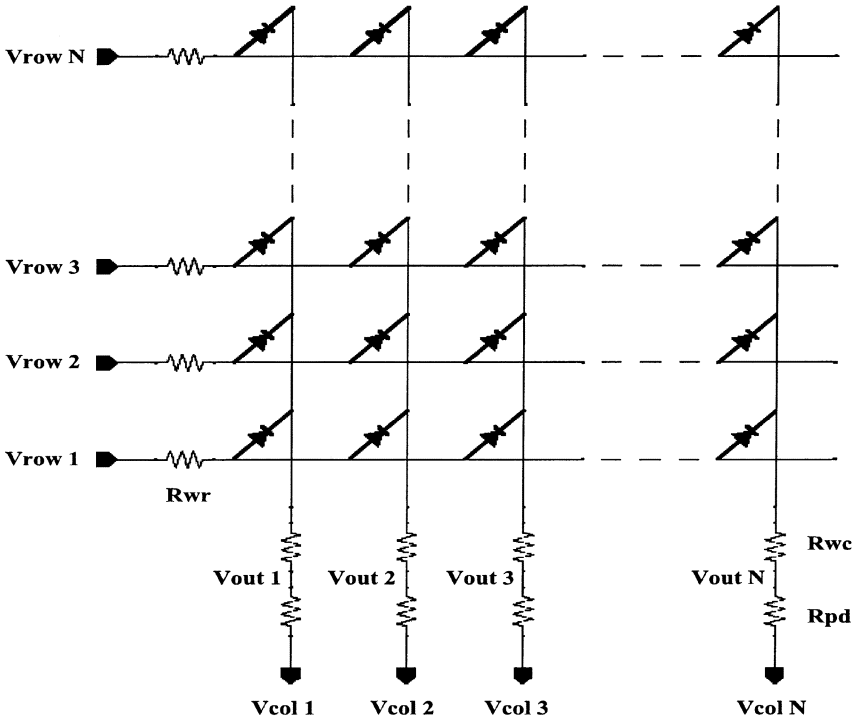


Fig. 10. The generic crosspoint crossbar structure selects a row by placing one of the horizontal inputs to V_{dd} and the remaining rows to GND.

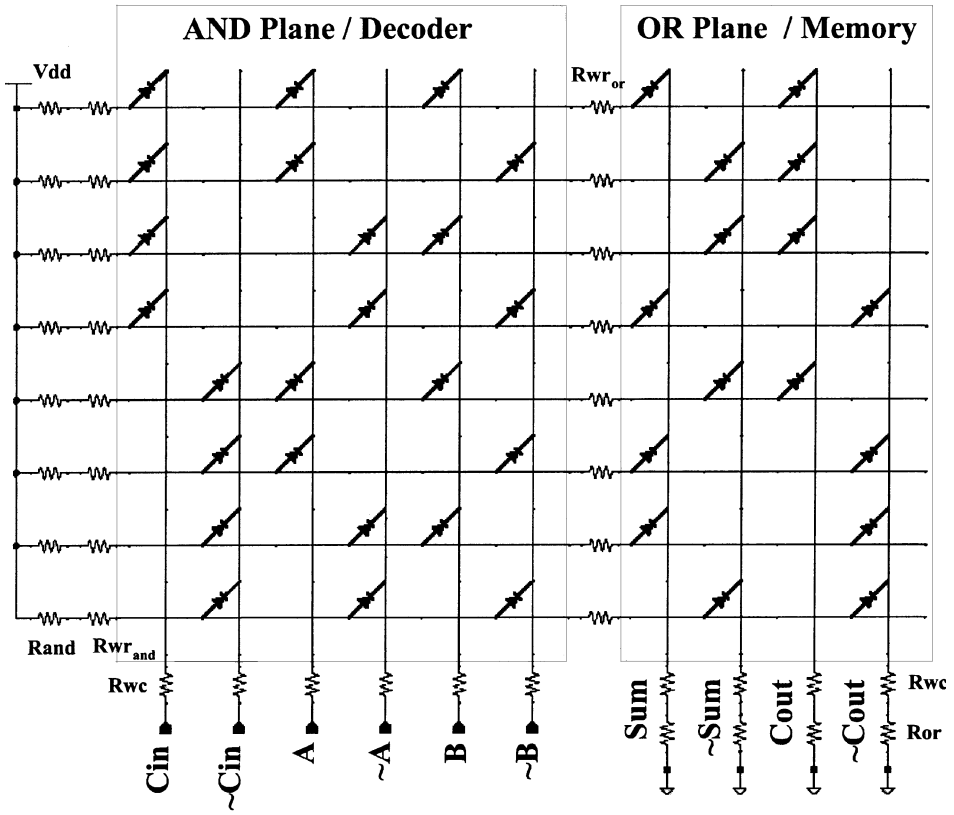


Fig. 11. Schematic of a full-adder employing programmed decoder.

higher than the devices in the crosspoint array; the spacing between the wires bordering the decoder and crosspoint could also be set at a pitch that allows the sections to be fabricated with different characteristics.

The potential utility of the crossbar is determined by the availability of suitable devices. Noise and fan-out issues were discussed in the previous section. Potential size is limited by the available wire pitch and the scalability of the crossbar.

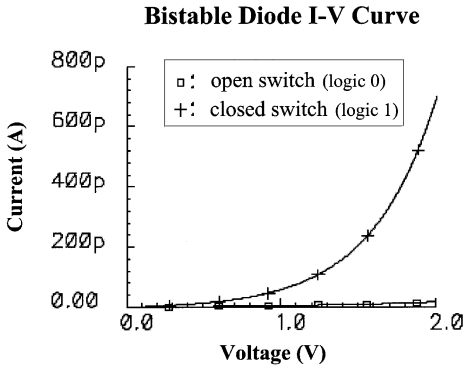


Fig. 12. *I–V* curve modeling bistable devices.

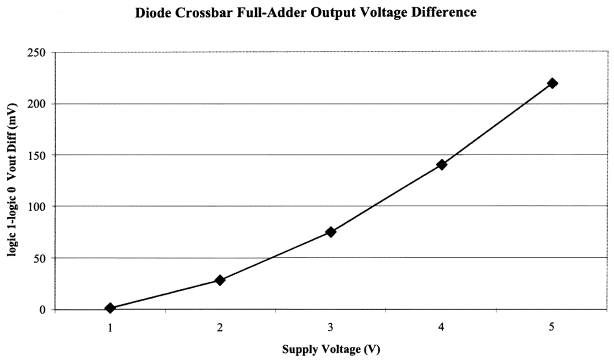


Fig. 13. Boosting the supply voltage is one method of increasing the difference between worst case high and low voltages.

At a 40-nm pitch, the raw density will be almost 10^{11} crosspoints per square centimeter. However, since not all crosspoints will be used, the functional density could be one or more orders of magnitude less. Obviously speed is limited by the *RC* time constants, and will be fairly low. Power depends on the end-device and wire characteristics.

E. Multistate Device Circuits

Due to their small noise margins, NDR-latch based circuits and memory arrays might be difficult to build in large scales. Instead it is useful to consider devices that have shown the capability to switch between several states [52]–[54]. These devices provide noise margin through their on/off ratio between the states and, due to long state retention times, are highly suited to form the basis of a nonvolatile memory. A representative *I–V* characteristic, and the one used as a basis for the circuits in this section, is shown in Fig. 14.

A number of approaches to building molecular memories are discussed in [55]. In general, a memory array is built with a two-state device by combining it with an isolation device, usually a molecular diode. The overall structure is very similar to the crossbar presented in the previous section.

During the write operation, the cells are reset by biasing the wordline low and the reset plane high. Thus, all two-state devices in that wordline enter the conductive state without any other cells being affected, as just the voltage bias of the reset plane is not enough to affect any memory cells. In the second part of a write, the wordline is now pulled to a high voltage. A particular bit sequence is written into the word

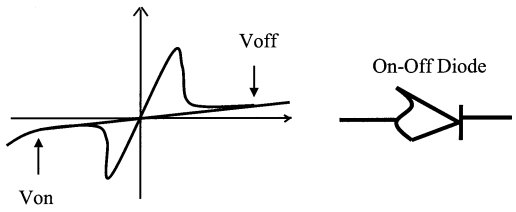


Fig. 14. Representative *I–V* characteristic for a two-state device. The device changes to the "on" state whenever the voltage across it goes below $V_{\rm on}$ and changes to the "off" state upon reaching $V_{\rm off}$.

by selectively pulling the bitline of a cell in which a zero is to be stored to a low voltage, while keeping the bitlines of cells that are to store a one at the reference voltage. Thus, two-state devices in cells that are to store a zero are flipped into the nonconductive state, while those that are to store a one remain at a conductive state; other wordlines remain unaffected. Reading out the information is accomplished by biasing the wordline to be read high. At read the devices will form voltage dividers with the load resistors on the end of the bitline (conductive devices actually form an NDR latch but will remain in the high current/low voltage state). This results in appreciable current being drawn through the load resistor and a relatively high voltage on the bitline for a conductive-state cell, but low voltage for a nonconductive-state cell. Thus, the difference in conductivity of the two states of the device can be read out as a voltage difference on the bitline.

The scalability and memory standby power consumption are largely determined by the resistance looking back into a memory cell from the bitline, which in turn is dominated by a reverse-biased diode; the analysis is similar to the one described above for crossbar arrays. The ratio of the "on" resistance to the wire resistance and to the "off" resistance determines the how large these arrays can be made. The "off" resistance also provides a lower bound for static power consumption. In comparison, in the year 2016 transistors are anticipated to have an on–off ratio in the 10^3–10^5 range. Similar goals for molecular electronics are reasonable. Today, devices that have been demonstrated in practice have on–off ratios only in the 10^1–10^2 range, with typical "on" values in the $10^6 - 5 \times 10^8$ range and with "off" values $> 4 \times 10^9$ [45].

On–off devices are also potentially useful for making programmable logic trees. For example, they present one approach to overcoming the programming problem described for the crossbar.

F. Molecular Electronics—Comparison Metrics

Molecular electronics has a lot of potential to enable electronic functionality to continue scaling beyond the end of CMOS scaling. In Table 1, potential moletronic circuit approaches are compared with predictions made in the ITRS [21] for end of the roadmap CMOS in terms of speed, power, and density. To keep the comparison simple, only high-performance CMOS is used in this comparison. The table compares footprint, energy/transition, delay, power density, and compute density (all the densities are peak metrics). The

Table 1
Metrics Comparing Predicted 22-nm CMOS, Directed Assembly Molecular Circuits Based On NDR Molecules, and Self-Assembled Crossbars Based on Multistate Devices

Metric	CMOS	NDR	Crossbar
Device active length	10 nm (Leff) 22 nm (Ldrawn)	2 -3 nm	
Device footprint	15,000 nm^2	8,000 nm^2	2,000 nm^2
Energy/transition	2 pJ	0.01 fJ	1 fJ
Delay/transition	0.15 ps	1 μs	1 μs
Power Density	800 W/μm^2	0.125 W/μm^2	0.5 W/μm^2
Computation Density	444×10^12 transitions/s/μm	125×10^6 transitions/s/μm	500×10^6 transitions/s/μm

overhead associated with each architecture is ignored for simplicity. Peak density is based on a $F = 22$ nm half-pitch patterning capability. A CMOS transistor is about 8 F $\times$ 4 F in area. The NDR–RTD gate shown in Fig. 8 would consume an area of around 16 F $\times$ 4 F if the chemicals were self assembled onto features built using 22-nm lithography. Since this circuit is equivalent to a four-transistor CMOS circuit, the result was divided by four to obtain the number in the table. A crosspoint in a crossbar takes an area of 2 F $\times$ 2 F. The energy and delay numbers are drawn from the ITRS or from simulations based on characteristics of currently demonstrated molecular devices. The crossbar has a higher energy per transition than the RTD-based circuit due to the capacitance of long lines, and uses devices with higher current density to be scalable. These numbers are manipulated to obtain the peak power density and transition rate density. As can be seen, molecular circuits show clear potential for superiority in energy but not in performance. Of course, as such circuits are still in the construction stage; these numbers could be off by orders of magnitude. The production cost of building such circuits is very hard to predict. The cost of building CMOS chips continues to follow an exponential law with time. It is reasonable to expect that molecular chips will be less expensive to build, as chemical self-assembly is used to build the devices, rather than many very precise lithography steps.

V. Molecular Electronic Architecture

Given the constraints introduced by bottom-up manufacturing, architectures for molecular electronics look significantly different than traditional architectures. Traditional architectures depend on the ability to create arbitrary patterns that can specify exactly where each active component and wire is placed and how they are interconnected. Bottom-up manufacturing, on the other hand, will be best at building very dense "crystalline" structures which do not depend on exact placement or arbitrarily specified patterns. As a result, the architecture-level modeling of molecular electronics will be significantly different from current CMOS models.

A. Information Content

The available assembly primitives require that molecular electronic-based architectures be built from structures with low information content and with repeating regular patterns or even from random patterns. Furthermore, they must be inherently defect tolerant. These requirements yield two basic classes of nanoelectronic architectures: unstructured and quasi-regular. Unstructured architectures assume only the most basic manufacturing primitives and only require that molecules and wires be assembled stochastically. These architectures are the easiest to build but the hardest to use. The quasi-regular approach assumes that some order can be imposed at the nanoscale level, e.g., that 2-D crossbars are manufacturable. Such an architecture would, however, not require that the wires in the crossbar be laid out in any particular order.

The main hurdle for unstructured architectures is that the manufacturing process produces a device that has no predetermined information content. Instead, the user of the device must probe it to determine what kinds of structures are present. After probing the device, if useful structures are found, they can be connected to form a circuit. The probing task is similar in spirit to defect detection, but instead of finding the defects, it must determine the functionality on the device.

The quasi-regular architectures depend on the manufacturing process to create a known pattern of regular structures. The goal is to configure the structures to implement whatever functionality is desired. Architectures in this class take the middle road in terms of difficulty in manufacturing and difficulty in using the device.

Another less ambiguous way to categorize the different approaches is to examine information binding time. Regardless of the approach, the desired result is a circuit that performs some task. The circuit itself requires a certain amount of information to describe it. There are primarily two times at which information about the desired circuit can be bound into the circuit: at manufacturing time and at configuration time. Traditional methods, which we call deterministic architectures, bind the information that describes the circuit completely at manufacturing time. These methods require precise layout and the ability to create arbitrary patterns; otherwise, they would not be able to put all the information in at manufacturing time. On the other end of the spectrum are unstructured architectures which bind the circuit information at configuration time. These architectures require little manufacturing-time precision, as all of the functionality is embedded into the circuit during configuration. However, the manufactured structure must be examined so that the circuit information can be properly bound at configuration time. In between these two extremes are quasi-regular architectures that bind some information at manufacturing time and some at configuration time. There is a clear tradeoff between the cost of manufacturing and the cost of configuration. Both deterministic and unstructured architectures take extreme points, which appears to be very costly at the nanoscale.

B. Unstructured Approaches

While it is true that unstructured architectures are costly in terms of configuration time, it is also true that such

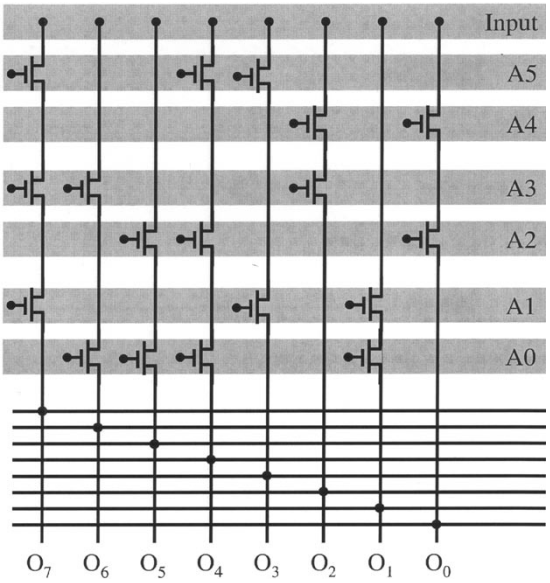


Fig. 15. Schematic of a molecular demultiplexor based on random assembly of gold particles to act as gates on nanoscale wires. The horizontal, wide, gray lines represent microwires, while the thin lines represent nanowires. The "dots" on the transistor gates represent connections between the microscale and the nanoscale.

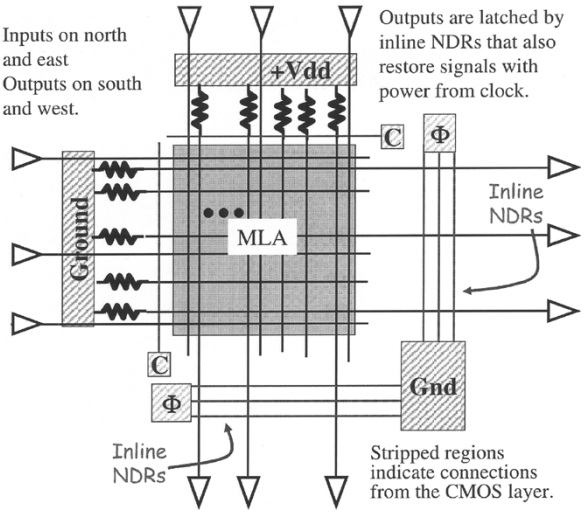


Fig. 16. A nanoBlock. The MLA is reconfigurable and combines with the resistors attached to $V_{\rm dd}$ and ground to create a reconfigurable diode–resistor logic array. Signal restoration is performed in the molecular latches which are orthogonal to the output wires.

architectures have some clear advantages over approaches requiring more order. A good example of how unstructured approaches can reduce the cost of manufacturing without significantly increasing the cost of configuration is the nondeterministic demultiplexor proposed by HP [50]. This demultiplexor (demux) is formed by intersecting a parallel set of microscale wires with an orthogonal set of parallel nanoscale wires. Between the microscale wires and the nanoscale wires is a layer consisting of randomly placed gold particles. Wherever the nanoscale wires contact a gold particle a transistor gate is formed. If the gold particle also contacts a microscale wire, then that microscale wire can be used to turn on that particular "transistor." The microscale wires essentially provide an address which is used to select one of the nanoscale wires (see Fig. 15.) A traditional demux using binary signals requires $\log_2 n$ address wires to select from one of n outputs; however, it requires substantial precision at the time of manufacture. The demux proposed by HP, on the other hand, requires more microscale address lines, since the address of each nanoscale wire is formed "randomly"; furthermore, one does not know at manufacturing time which address will select which nanoscale wire. However, the difficulty in determining the addresses is low, and with a sufficient number of micrometer-scale wires, each of the output wires can be selected ($5 \log_2 n$ address lines will select each one of the n outputs with greater than 50% probability). Note that the three-terminal devices in this example can be modeled as conventional FETs with no need for the UDM, which right now is limited to modeling two-terminal devices.

A more complete architecture based on the unstructured approach is the Nanocell [56]. The Nanocell architecture is based on the random assembly of a small grid of on–off devices. The functionality of each grid is determined post-fabrication through a detection phase using external voltage pulses. Gain and input–output isolation are established between cells using NDR-latch type circuits. Preliminary calculations show that this approach, when combined with CMOS at the 22-nm node, can achieve a functional density equivalent to 10^{10} devices/cm^2.

C. Array-Based Architectures

Array-based or crossbar-based architectures typify the quasi-regular class of architectures. Each basic unit (crossbar) consists of wires and programmable molecular devices. The crossbars are then connected together to form a larger mesh of configurable elements. Unlike unstructured architectures, the potential functionality of array-based devices does not have to be discovered; however, these architectures do depend on postfabrication programming in order to create logical circuits. Furthermore, since the fabrication primitives are unlikely to yield perfect meshes, these architectures require reprogrammable components in order to provide defect tolerance.

One such architecture is the nanoFabric [25], of which the basic unit of logic is a nanoBlock (see Fig. 16). Each nanoBlock is based around a molecular logic array (MLA). At each intersection of the MLA is a reconfigurable switch (e.g., a pseudo-rotaxene) in series with a diode. Diode–resistor logic is used to perform logical operations. To create a complete logic family, signals and their complements are brought into each circuit and produce both the desired functions and their complements. Logic values are restored using molecular latches [37], which also provide a mechanism for latching values and isolating outputs from one nanoBlock from the inputs of another nanoBlock. The nanoBlocks are grouped together into clusters and arranged so that the outputs of a nanoBlock intersect the inputs of two other

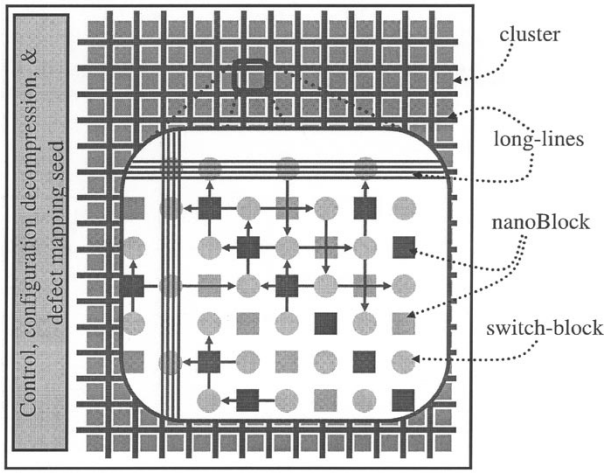


Fig. 17. The nanoFabric.

nanoBlocks. The area where the wires for four nanoBlocks intersect (two outputs, two inputs) is called a switchblock (see Fig. 17); the result is a 2-D mesh of nanoBlocks.

There are several things to note about the nanoFabric. First, it is manufactured hierarchically (devices and wires are manufactured first, wires are aligned and crossbars are created resulting in a nanoBlock, the nanoBlocks themselves are then put together to form the nanoFabric). Second, the architecture itself is hierarchical. Finally, it is reconfigurable.

Another example of this class of architectures is the Nanowire-based approach used by DeHon [26]. It, too, is based on meshes of intersecting wires. Silicon nanowires are arranged into arrays which can implement wide-fanin logic functions. The resulting architecture can be viewed as an array of PLAs.

Both of these architectures exhibit the features necessary for a quasi-regular architecture. They are based on 2-D meshes where the intersections of the mesh contain some form of configurable switch. Logic functions are formed by configuring the meshes and the connections between the meshes, which are made using nanoscale wires. There is a separation between devices that compute logical functions and those that provide isolation and signal restoration. The nanoscale components are supported by microscale devices which provide infrastructure such as power, ground, clock, etc. Finally, they both support defect tolerance.

D. Defect Tolerance

One significant disadvantage of bottom-up manufacturing is that it is likely to have significantly higher defect densities than current technologies: we expect that the very nature of chemical fabrication will result in defect densities of as much as 10%.[8] Such high defect densities require a completely new approach to manufacturing computational devices. No longer will it be possible to test a device and throw it away if it has a defect, since it is likely that every chip will have a significant number of defects. Instead, one must devise a way to use defective chips, i.e., build working systems from nonworking components.

Most modern fault-tolerant circuit design techniques (e.g., triple-mode redundancy and error correction coding) are not applicable to the bottom-up molecular electronics manufacturing process discussed here. Such techniques work reliably only if the number of defects is below a certain hard threshold; which will likely be exceeded in this case.

Memory chips are able to tolerate some defects by having redundancy built into them: for instance, a row containing a defect might be replaced with a spare row after fabrication. However, nanoscale electronic fabrics will not be able to simply follow this approach, since it is unlikely that an entire row or column of any appreciable size will be defect free.

Another potential solution is suggested by looking at reconfigurable fabrics, such as field-programmable gate arrays (FPGAs) and the Teramac custom computer [57], [58]. An FPGA is a regular array of programmable (reconfigurable) logic elements connected by programmable interconnect, allowing any function to be configured onto the device. The Teramac is essentially a very large FPGA with a rich interconnect that works in spite of the fact that 75% of the chips contained in the Teramac had some number of defects. Like memories, the Teramac have significant built-in redundancy; therefore, postmanufacturing diagnosis can identify and locate faults, and the reconfigurable structures can be programmed to avoid them. This approach trades off manufacturing complexity for postfabrication programming.

The reduction in manufacturing time makes reconfigurable fabrics a particularly attractive architecture for bottom-up manufactured nanoelectronics circuits, since directed self-assembly will most easily result in highly regular, homogeneous structures. The fabrication process for these fabrics can be followed by a testing phase, where a defect map will be created and shipped with the fabric. The defect map can then be used by compilers to route around the defects, which can be made sufficiently tractable if the underlying circuits have sufficient routing resources, i.e., they are wire-rich.

However, the diagnosis process for fault localization requires some discussion. In general, any methodology for locating defects should meet the following criteria.

- It should not require access to the individual components of the fabrics, such as individual gates or wire crossings.
- It should scale with the number of defects.
- It should scale with fabric size, so that testing does not become a bottleneck in the manufacturing process.

So the question becomes, how does one find defects without requiring each component to be probed? There is a large body of work in statistics and information theory on techniques for finding subsets of a population in which all members satisfy a given property. Various flavors of this technique, called *group testing*, have been applied to a variety of problems [59]–[62], none of which, however, had constraints as demanding as those of nanoscale electronics.

One approach for defect detection in molecular scale reconfigurable circuits consists of configuring the components

[8]The techniques discussed in this section are also potentially important for next-generation lithography, e.g., Extreme Ultra Violet (EUV) and other approaches for sub-90-nm lithography.

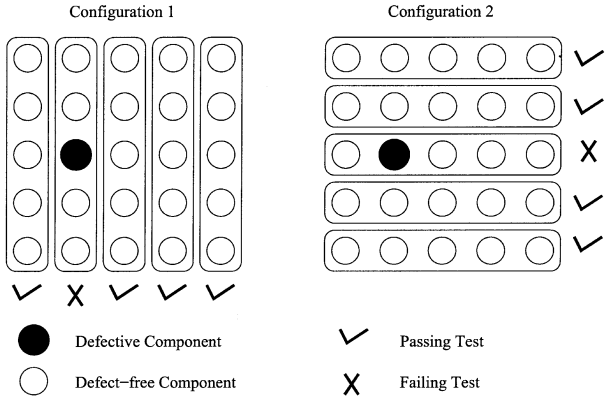


Fig. 18. An example showing how a defective component is located using two different test circuit configurations. The components within one rectangular block are part of one test circuit.

on the fabric[9] into test circuits, which will give us information about the presence or absence of defects in their constituent components. Each component is made part of several different test circuits, and information about the error status of each of those circuits is collected. This information is then used to deduce and confirm the exact location of the defects.

As an example, consider the situation in Fig. 18. Five components are configured into one test circuit, which computes a simple mathematical function. This function is such that defects in one or more circuit components causes the answer to diverge from the correct value. Therefore, by comparing the circuit's output with the correct answer, the presence or absence of any defects in the circuit components can be detected. In the first run, the components are configured vertically, and test circuit 2 detects a defect. In the next run, the components are configured horizontally, and test circuit 3 fails. Since no other errors are detected, we can conclude that the component at the intersection of these two circuits is defective, and all others are good.

Since the tester cannot have access to individual fabric components, the test circuits will be large, consisting of tens and perhaps even hundreds of components. With high defect rates, each circuit will on average have multiple defective components, complicating the simple picture presented in the example above. In particular, test circuits which give information only about the presence or absence of defects (such as the ones used above) will be useless: almost each and every test circuit will report the presence of defects. The key idea is to use more powerful test circuits that return more information about the defects in their components, such as a count of defects. An example would be a circuit that computes a mathematical function whose output will deviate from the correct value if any of the circuit's components are defective; if the amount of this deviation deterministically depends on the number of defective components, then a comparison of the circuit's output with the correct result can indicate the number of defects present in the circuit.

[9]We are deliberately leaving the meaning of "component" unspecified. It will depend on the final design of the fabric: a component may be one or more simple logic gates, or a LUT implementing an arbitrary logic function; the on-fabric interconnects will also be "components" in the sense that they may also be defective.

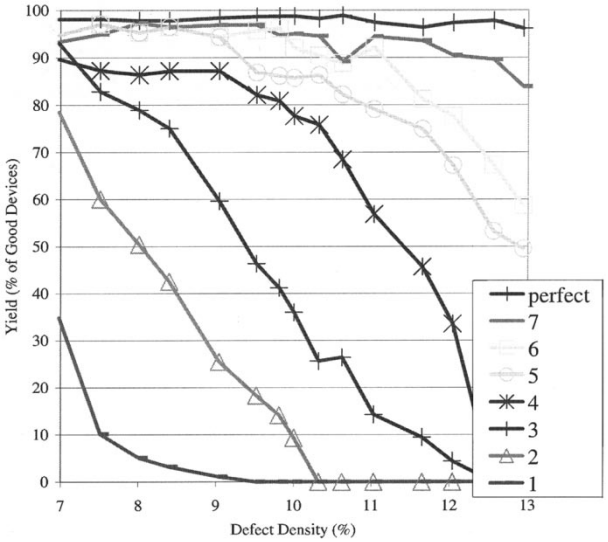


Fig. 19. Yields achieved by varying the defect densities and the number of defects our test circuits could count. The x axis represents the defect density of the fabric, the y axis shows the yield achieved (or, in other words, the fraction of the fabric's defect-free components that are identified as such), and each line represents a counter that can count defects up to a different threshold.

Using such defect-counting circuits, [63] proposes splitting the process of defect-mapping into two phases: a *probability assignment phase* and a *defect location phase*. The probability assignment phase attempts to separate the components in the fabric into two groups: those that are probably good and those that are probably bad. The former will have an expected defect density that is low enough so that in the defect location phase, one can use circuits that return zero–one information about the presence of defects to locate them exactly.

To test the effectiveness of this procedure and to measure the impact of the defect-counting threshold on the results, we ran a number of simulations, the results of which are presented in Fig. 19. From these results, it is apparent that it is possible to achieve high yields even with test circuits that can count a small number of defects, particularly if the defect density is low. For example, for densities less than 10%, a test circuit that can count up to four errors achieves yields of over 80%. With more powerful test circuits, yields of over 95% are achievable.

This testing strategy currently requires time proportional to n for testing a fabric of $n \times n$ components, if a test circuit of size n is being used. To speed up testing even further, the reconfigurability of the fabric can be leveraged. Once part of the fabric is mapped, it can be used to test other parts of the fabric. This can decrease test time by eliminating the off-chip bandwidth bottleneck and allowing for parallel testing.

E. The Case for CMOS/Nano Mixed Architectures

New nanoelectronic solutions present a potential for unprecedented levels of device density, low power computing, and possibly high operating speed. Despite this high potential, it will be very difficult for any new technology to compete head to head with silicon's large-scale fabrication

infrastructure, proven design methodologies, and economic predictability.

An alternative approach to an abrupt technology change is the integration of silicon with nanoelectronics, i.e., mixed CMOS/nano integrated circuits. This route would allow a smooth transition and permit leveraging the beneficial aspects of both technologies [64]. Such an approach is embodied in the architectures proposed by Goldstein [25] and DeHon [26]. We refer to this design paradigm as nano on CMOS (NoC); similar ideas have also been suggested in [65].

The NoC paradigm allows for significant design versatility. One extreme uses the CMOS as the primary computation medium while the nano on top is used as a supplement to better achieve integration goals. For example, the nano crossbar can act as a memory or as large logic arrays. At the other extreme, the nano portion would be the primary computation medium while the underlying CMOS would be used simply to provide signal gain and latching capabilities. This latter approach is the one described in Section V-C. A more balanced solution uses both mediums for primary computation with portions of the circuit being allocated either to CMOS or nano at a finer grain.

The concept of scaling is also affected by the NoC paradigm. While the silicon CMOS era has been fueled by a continuous scaling of device performance and sizes as expressed by the different flavors of "Moore's law," nanoelectronics promises to take miniaturization beyond the "brick wall" CMOS scaling is predicted to hit but without the same scaling scenario (especially for molecular electronics). Instead, nanoelectronic scaling will be in the form of comprising increasingly large percentages of mixed CMOS/nano ICs. At first, small amounts of nano will be embedded on a predominantly CMOS chip. In successive generations, the amount of nano can be increased as the amount of CMOS is decreased. Increasing the nano-to-CMOS ratio over time can provide a means to ease into a new technology paradigm. Furthermore, the possibility of mixed CMOS/nano circuits permits using the best aspects of both technologies simultaneously, while the undesired aspects of a technology can be compensated by the partner technology.

We proposed a new definition of scaling for mixed nano/CMOS circuits that still allows an exponential increase in "effective density" even as individual component sizes stay constant [66]. For CMOS the density doubles by scaling the dimensions by 0.7 times for every technology node, but for molecular electronics no such continuous scaling seems feasible; instead the "effective density" (number of nano/CMOS devices per unit area) can still double simply by increasing the ratio of nano/CMOS [66].

VI. Conclusion

Molecular electronics have the potential to deliver unprecedented levels of computing per dollar-watt-cm^2. Recent progress in manufacturing and understanding molecular components has been significant. However, just as complex CMOS-based system design is made possible by hierarchical abstraction, molecular electronics requires the development of accurate device, logic, circuit, and architecture models. Molecular devices may behave like traditional devices, but the method by which they achieve this effect is significantly different, requiring new modeling methods. We introduced a UDM which can be used to quickly model any two-terminal device. This is important as physical scientists are generating new devices at a rapid rate.

In addition to different kinds of devices, the very method of manufacturing requires that we rethink our circuit methodologies and our assumptions of what can be incorporated into an architecture. The manufacturing implications that are most prominent are the inability to create arbitrary patterns and the increased defect density of the manufactured parts. These two factors combine to limit the amount of information that can be manufactured into the final product. Instead, postfabrication programming will be necessary to avoid defects and create the desired functionality.

One approach which takes into account the limitations of both the devices and the fabrication methods is to create systems from interlinked reconfigurable crossbars. The crossbars are programmed, around the defects, to implement logic functions. Such architectures can be constructed solely from molecular electronics or from a combination of CMOS and molecular electronics. The latter design is an example of NoC. As molecular electronics comes into its own, the amount of nano in the architecture will increase, thereby increasing the effective density of the resulting device. This increase in density, without a decrease in CMOS feature size, constitutes a new form of scaling.

Acknowledgment

This work would not have been possible without the help of the authors' students. In particular, the authors would like to thank G. Rose, M. Mishra, M. Budiu, and others. The authors would also like to thank Dr. J. Ellenbogen of MITRE Corporation for fruitful discussions, and the anonymous reviewers for their comments.

References

[1] Y. Huang, X. Duan, Y. Cui, L. J. Lauhon, K. Kim, and C. M. Lieber, "Logic gates and computation from assembled nanowire building blocks," *Science*, vol. 294, pp. 1313–1316, Nov. 2001.

[2] A. Bachtold, P. Hadley, T. Nakanishi, and C. Dekker, "Logic circuits with carbon nanotube transistors," *Science*, vol. 294, pp. 1317–1320, Nov. 2001.

[3] 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*, vol. 285, pp. 391–394, July 1999.

[4] J. Chen, M. A. Reed, A. M. Rawlett, and J. M. Tour, "Observation of a large on–off ratio and negative differential resistance in an electronic molecular switch," *Science*, vol. 286, pp. 1550–1552, 1999.

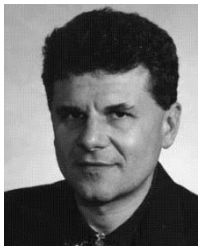
[5] J. Chen and M. A. Reed, "Molecular wires, switches and memories," presented at the Int. Conf. Molecular Electronics, Kona, HI, Dec. 2000.

[6] C. Collier, G. Mattersteig, E. Wong, Y. Luo, K. Beverly, J. Sampaio, F. Raymo, J. Stoddart, and J. Heath, "A [2]catenane-based solid state reconfigurable switch," *Science*, vol. 289, pp. 1172–1175, 2000.

[7] Y. Cui and C. Lieber, "Functional nanoscale electronic devices assembled using silicon nanowire building blocks," *Science*, vol. 291, p. 851, 2001.

[8] Y. Huang, X. Duan, Y. Cui, L. Lauhon, K.-H. Kim, and C. Lieber, "Logic gates and computation from assembled nanowire building blocks," *Science*, vol. 294, p. 1313, 2001.
[9] Y. Cui, C. Lieber, L. Lauhon, M. Gudiksen, and J. Wang, "Diameter-controlled synthesis of single crystal silicon nanowires," *Appl. Phys. Lett.*, vol. 78, no. 15, pp. 2214–2216, 2001.
[10] T. I. Kamins, R. S. Williams, Y. Chen, Y.-L. Chang, and Y. A. Chang, "Chemical vapor deposition of Si nanowires nucleated by TiSi2 islands on Si," *Appl. Phys. Lett.*, vol. 76, pp. 562–564, 2000.
[11] Y. Huang, X. Duan, Q. Wei, and C. Lieber, "Directed assembly of onedimensional nanostructures into functional networks," *Science*, vol. 291, pp. 630–633, Jan. 2001.
[12] A. Morales and C. Lieber, "A laser ablation method for the synthesis of crystalline semiconductor nanowires," *Science*, vol. 279, pp. 208–211, 1998.
[13] C. Soh, C. Quate, C. Morpurgo, C. Marcus, C. Kong, and C. Dai, "Integrated nanotube circuits: controlled growth and ohmic contacting of single-walled carbon nanotubes," *Appl. Phys. Lett.*, vol. 75, no. 5, pp. 627–629, 1999.
[14] T. Rueckes, K. Kim, E. Joselevich, G. Tseng, C. Cheung, and C. Lieber, "Carbon nanotube based nonvolatile random access memory for molecular computing," *Science*, vol. 289, pp. 94–97, 2000.
[15] D. J. Pena, B. Razavi, P. A. Smith, M. J. Natan, T. S. Mayer, T. E. Mallouk, and C. D. Keating, "Electrochemical synthesis of multi-material nanowires as building blocks for functional nanostructures," in *Proc. MRS Symp.*, vol. 636, 2001, pp. D4.6.1–D4.6.6.
[16] Y. Xia, J. Rogers, K. Paul, and G. Whitesides, "Unconventional methods for fabricating and patterning nanostructures," *Chem. Rev.*, vol. 99, pp. 1823–1848, 1999.
[17] E. Winfree, F. Liu, L. A. Wenzler, and N. C. Seeman, "Design and self-assembly of two-dimensional DNA crystals," *Nature*, vol. 394, pp. 539–544, 1998.
[18] C. Mirkin, "Programming the assembly of two- and three-dimensional architectures with DNA and nanoscale inorganic building blocks," *Inorg. Chem.*, vol. 39, pp. 2258–2272, 2000.
[19] S. R. Nicewarner-Pena, S. Raina, G. P. Goodrich, N. V. Fedoroff, and C. D. Keating, "Hybridization and extension of au nanoparticle-bound oligonucleotides," *J. Amer. Chem. Soc.*, vol. 124, pp. 7314–7323, 2002.
[20] W. B. Culbertson, R. Amerson, R. Carter, P. Kuekes, and G. Snider, "Defect tolerance on the teramac custom computer," in *Proc. 1997 IEEE Symp. FPGAs Custom Computing Machines*, 1997, pp. 116–124.
[21] SIA/Sematech. (2000) International Technology Roadmap for Semiconductors. [Online] Available: http://public.itrs.net
[22] M. Bhattacharya and P. Mazumder, "Augmentation of SPICE for simulation of circuits containing resonant tunneling diodes," *IEEE Trans. Computer-Aided Design*, vol. 20, pp. 39–50, Jan. 2001.
[23] Y. S. Yu, Y. I. Jung, J. H. Park, S. W. Hwang, and D. Ahn, "Simulation of single-electron/CMOS hybrid circuits using SPICE macromodeling," *J. Korean Phys. Soc.*, vol. 20, no. 35, pp. S991–S994, 1999.
[24] J. R. Heath, P. J. Kuekes, G. S. Snider, and R. S. Williams, "A defect-tolerant computer architecture: opportunities for nanotechnology," *Science*, vol. 280, pp. 1716–1721, June 1998.
[25] S. C. Goldstein and M. Budiu, "Nanofabrics: spatial computing using molecular nanoelectronics," in *Proc. 28th Int. Symp. Computer Architecture*, 2001, pp. 178–189.
[26] A. DeHon, "Array-based architecture for molecular electronics," presented at the 1st Workshop Non-Silicon Computation (NSC-1), Boston, MA, 2002.
[27] L. J. K. Durbeck and N. J. Macias, "The cell matrix: an architecture for nanocomputing," *Nanotechnology*, vol. 12, pp. 217–320, Aug. 2001.
[28] Z. Yan and M. J. Deer, "New RTD large-signal DC suitable for PSPICE," *IEEE Trans. Computer-Aided Design*, vol. 14, pp. 167–172, Feb. 1995.
[29] J. N. Schulman, H. J. De Los Santos, and D. H. Chow, "Physics-based RTD current-voltage equation," *IEEE Electron Device Lett.*, vol. 17, pp. 220–222, May 1996.
[30] M. Ziegler, G. Rose, and M. Stan, "A universal device model for nanoelectronic circuit simulation," presented at the IEEE Conf. Nanotechnology (IEEE-NANO), Washington, DC, 2002.
[31] The Nanotechnology Simulation Hub. [Online] Available: http://nanohub.purdue.edu/
[32] 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*, vol. 286, pp. 1550–1552, Nov. 1999.
[33] J. C. Ellenbogen and J. C. Love, "Architectures for molecular electronic computers. I. Logic structures and an adder designed from molecular electronic diodes," *Proc. IEEE*, vol. 88, pp. 386–426, Mar. 2000.
[34] X. Duan, Y. Huang, and C. M. Lieber, "Nonvolatile memory and programmable logic from molecule-gated nanowires," *Nano Lett.*, vol. 2, no. 5, pp. 487–490 , 2002.
[35] Cadence. [Online] Available: http://www.cadence.com/
[36] R. H. Mathews, J. P. Sage, T. C. L. G. Sollner, S. D. Calawa, C.-L.Chang-Lee Chen, L. J. Mahoney, P. A. Maki, and K. M. Molvar, "A new RTD-FET logic family," *Proc. IEEE*, vol. 87, pp. 596–605, Apr. 1999.
[37] S. C. Goldstein and D. Rosewater, "Digital logic using molecular electronics," in *Proc. Int. Solid-State Circuits Conf.*, 2002, p. 12.5.
[38] R. W.Robert W. Keyes, "What makes a good computer device?," *Science*, vol. 230, no. 4722, pp. 138–144, 1985.
[39] D. Nackashi and P. Franzon, "Moletronics: a circuit design perspective," in *Proc. SPIE, Int. Conf. SPIE Smart Electronics and MEMS*, vol. 4236, 2000, pp. 80–88.
[40] D. P. Nackashi, C. J. Amsinck, and P. D. Franzon, "Molecular electronic circuits and circuit elements," *IEEE J. Nanotechnol.*, submitted for publication.
[41] R. M. Metzger *et al.*, "Unimolecular electrical rectification in hexadecylquinolinium tricyanoquinodimethanide," *J. Amer. Chem. Soc.*, vol. 119, pp. 10 455–10 466, 1997.
[42] P. S. Weiss *et al.*, "Probing electronic properties of conjugated and saturated molecules in self-assembled monolayers," *Ann. NY Acad. Sci.*, vol. 852, pp. 145–168, 1998.
[43] P. J. Kuekes, J. R. Heath, and R. S. Williams, "Molecular wire crossbar memory," U.S. Patent 6 128 214, Oct. 2000.
[44] A. R. Pease, J. O. Jeppesen, J. F. Stoddart, Y. Luo, C. P. Collier, and J. R. Heath, "Switching devices based on interlocked molecules," *Acc. Chem. Res.*, vol. 34, no. 6, pp. 433–444, 2001.
[45] 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*, no. 14, pp. 462–468, 2003.
[46] W. J. Gallagher, J. H. Kaufman, S. S. Papworth, and R. E. Scheuerlein, "Magnetic memory array using magnetic tunnel junction devices in the memory cells," U.S. Patent 5 640 343 (IBM), June 1997.
[47] T. Rueckes, K. Kim, E. Joselevich, G. Y. Tseng, C. Cheung, and C. M. Lieber, "Carbon nanotube-based nonvolatile random access memory for molecular computing," *Science*, vol. 289, pp. 94–97, July 2000.
[48] M. Ziegler and M. Stan, "Design and analysis of crossbar circuits for molecular nanoelectronics," presented at the IEEE Conf. Nanotechnology (IEEE-NANO), Washington, DC, 2002.
[49] P. J. Kuekes, J. R. Heath, and R. S. Williams, "Molecular-wire crossbar interconnect (MWCI) for signal routing and communications," U.S. Patent 6 314 019, Nov. 2001.
[50] P. J. Kuekes and R. S. Williams, "Demultimplexer for a molecular wire crossbar network (MWCN DEMUX)," U.S. Patent 6 256 767, July 2001.
[51] Y. Cui and C. M. Lieber, "Functional nanoscale electronic devices assembled using silicon nanowire building blocks," *Science*, vol. 291, pp. 851–853, Feb. 2001.
[52] M. A. Reed, J. Chen, A. M. Rawlett, D. W. Price, and J. M. Tour, "Molecular random access memory cell," *App. Phys. Lett.*, vol. 78, pp. 3735–3737, June 2001.
[53] Z. J. Donhauser, B. A. Mantooth, K. F. Kelly, L. A. Bumm, J. D. Monnell, J. J. Stapleton Jr., D. W. Price, A. M. Rawlett, D. L. Allara, J. M. Tour, and P. S. Weiss, "Conductance switching in single molecules through conformational changes," *Science*, vol. 292, pp. 2303–2307, June 2001.
[54] F.-R. F. Fan, J. Yang, S. M. Dirk, D. W. Price, D. Kosynkin, J. M. Tour, and A. J. Bard, "Determination of the molecular electrical properties of self-assembled monolayers of compounds of interest in molecular electronics," *J. Amer. Chem. Soc.*, vol. 123, pp. 2454–2455, Mar. 2001.
[55] C. J.Christian J. Amsinck, D. P.David P. Nackashi, N. H.Neil H. Di Spigna, and P. D.Paul D. Franzon, "Electrically accessible molecular memories," *IEEE J. Nanotechnol.*, submitted for publication.
[56] J. M. Tour, V. Zandt, C. Husband, S. Husband, E. Libby, D. Ruths, K. Young, L. Wilson, P. Franzon, and D. Nackashi, "Nanocell logic gates for molecular computing," *IEEE J. Nanotechnol.*, to be published.
[57] B. Culbertson, R. Amerson, R. Carter, P. Kuekes, and G. Snider, "Defect tolerance on the teramac custom computer," in *1997 IEEE Symp. FPGAs Custom Computing Machines (FCCM '97)*, 1997, pp. 116–123.

[58] J. R.James R. Heath, P. J.Philip J. Kuekes, G. S.Gregory S. Snider, and R. S.R. Stanley Williams, "A defect-tolerant computer architecture: opportunities for nanotechnology," *Science*, vol. 280, pp. 1716–1721, June 12, 1998.
[59] R. Dorfman, "The detection of defective members of large populations," *Ann. Math. Stat.*, vol. 14, pp. 436–440, Dec. 1943.
[60] M.Milton Sobel and P. A.Phyllis A. Groll, "Group testing to eliminate efficiently all defectives in a binomial sample," *Bell Syst. Tech. J.*, pp. 1179–1224, Sept. 1959.
[61] J. K. Wolf, "Born again group testing: multiaccess communications," *IEEE Trans. Inform. Theory*, vol. IT-31, pp. 185–191, Mar. 1985.
[62] E. Knill, W. J. Bruno, and D. C. Torney, "Non-adaptive group testing in the presence of errors," Los Alamos Natl. Lab., Los Alamos, NM, Tech. Rep. LAUR-95-2040, 1996.
[63] M. Mishra and S. Goldstein, "Scalable defect tolerance for molecular electronics," presented at the 1st Workshop Non-Silicon Computation (NSC-1), Boston, MA, 2002.
[64] M. Ziegler and M. Stan, "A case for CMOS/nano co-design," presented at the Int. Conf. Computer Aided Design (ICCAD), San Jose, CA, 2002.
[65] Microelectronics Advanced Research Initiative MELARI NANO. (2000) Technology Roadmap for Nanoelectronics. [Online] Available: http://www.cordis.lu/esprit/src/melna-rm.htm
[66] M. Stan, "A scaling scenario for nanoelectronic technologies," in *Proc. Georgia Tech Conf. Nanoscience and Nanotechnology*, 2001, p. 103.



Mircea R. Stan (Senior Member, IEEE) received the Diploma degree in Electronics and Communications from Politehnica University, Bucharest, Romania, in 1984 and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of Massachusetts, Amherst, in 1994 and 1996, respectively.

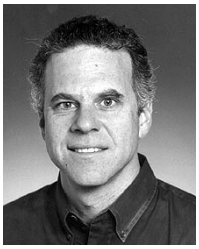
He has more than eight years of industrial experience as an R&D Engineer in Bucharest, Romania; Tokyo, Japan; and Atlanta, GA. Since 1996, he has been with the Department of Electrical and Computer Engineering at the University of Virginia, Charlottesville, first as an Assistant Professor, and as an Associate Professor since 2002. He has also been a Visiting Faculty Member at IBM in 2000, and at Intel in 1999 and 2002. His research interests include the areas of high-performance and low-power very large scale integration (VLSI), mixed-mode analog and digital circuits, computer arithmetic, embedded systems, and nanoelectronics.

Prof. Stan is a Member of the Association for Computing Machinery, Usenix, Eta Kappa Nu, Phi Kappa Phi, and Sigma Xi. He was the General Chair for the Great Lakes Symposium on VLSI (GLSVLSI) 2003. Since 2001, he has been an Associate Editor for the IEEE TRANSACTIONS ON VLSI SYSTEMS. In 1997, he received the National Science Foundation CAREER Award for investigating low-power design techniques.

Paul D. Franzon (Senior Member, IEEE) received the B.S., B.E. (with honors), and Ph.D. degrees from the University of Adelaide, Adelaide, Australia, in 1983, 1984, and 1988, respectively.

He has been with AT&T Bell Laboratories, DSTO Australia, Australia Telecom, and Communica Ltd. He is currently an Alumni Distinguished Professor at North Carolina State University (NCSU), Raleigh. He has led several major efforts and published over 100 papers in his areas of expertise. His current interests center on the technology and design of complex systems incorporating very large scale integration, microelectromechanical systems (MEMS), advanced packaging, and molecular computing. Application areas currently being explored include novel advanced packaging structures, network processors, silicon-on-insulator baseband radio circuit design for deep space, on-chip inductor and inductance issues, radio frequency MEMS, and moleware circuits and characterization.

Dr. Franzon received the National Science Foundation Young Investigators Award in 1993 and was selected to join the NCSU Academy of Outstanding Teachers in 2001.



Seth Copen Goldstein (Member, IEEE) received the B.S. degree from Princeton University, Princeton, NJ, in 1985 and the M.S. and Ph.D. degrees in computer science from the University of California, Berkeley, in 1994 and 1997, respectively.

He was CEO and Founder of Complete Computer Corporation. research focuses on computing systems and nanotechnology. Since 1997, he has been on the faculty of Carnegie Mellon University, Pittsburgh, PA. He is currently working on architectures and compilers for computer systems built with electronic nanotechnology. Because he believes that the fundamental challenge for computer science in the 21st century is how to effectively harness systems which contain billions of potentially faulty components, he is also working on novel circuit techniques, defect and fault tolerance, reconfigurable architectures, scalable optimizing compilers for spatial computing, and self-organizing systems.

John C. Lach (Member, IEEE) received the B.S. degree from Stanford University, Stanford, CA, in 1996 and the M.S. and Ph.D. degrees in electrical engineering from the University of California, Los Angeles, in 1998 and 2000, respectively.

Since 2000, he has been an Assistant Professor in the Electrical and Computer Engineering Department at the University of Virginia, Charlottesvills. He is also the Principal Investigator on the Dynaptable Computing project funded by the National Science Foundation. His primary research interests include dynamically adaptable and real-time embedded systems, computer-aided design techniques for very large scale integration, general purpose and application specific processor design, intellectual property protection, and wearable technologies for aged independence.

He is a Member of the IEEE Computer Society, the Association for Computing Machinery, the Special Interest Group on Design Automation, and Eta Kappa Nu. While at UCLA, he twice received the School of Engineering and Applied Sciences Dean's Award. He received the 2001–2002 University of Virginia Teaching Fellowship and the 2002–2003 Electrical and Computer Engineering New Faculty Teaching Award.

Matthew M. Ziegler (Member, IEEE) received the B.S. degree in electrical engineering from the University of Virginia, Charlottesville, in 2000. He is currently working toward the Ph.D. degree in electrical engineering at the University of Virginia.

He has been an Intern with Annapolis Micro Systems, Inc., and Thomson Consumer Electronics. Since 2001, he has been with the MITRE Corporation, McLean, VA, part-time, where he has worked on the simulation of nanoscale systems. His current research interests include very large scale integration (VLSI) design, computer-aided design, and developing VLSI approaches for nanoelectronics and molecular electronics.