Radial Addressing of Nanowires

JOHN E. SAVAGE and ERIC RACHLIN
Brown University
ANDRÉ DEHON
California Institute of Technology
and
CHARLES M. LIEBER and YUE WU
Harvard University

We introduce radial encoding of nanowires (NWs), a new method of differentiating and controlling NWs by a small set of mesoscale wires for use in crossbar memories. We describe methods of controlling these NWs and give efficient manufacturing algorithms. These new encoding and decoding methods do not suffer from the misalignment characteristic of flow-aligned NWs. They achieve comparable effective pitch and resulting memory density with axially encoded NWs, while avoiding potential cases of address ambiguity and simplifying NW preparation. We also explore hybrid axial/radial encodings and show that they offer no net benefit over pure codes.

Categories and Subject Descriptors: B.3.m [**Memory Structures**]: Miscellaneous; C.5.m [**Computer Systems Implementation**]: Miscellaneous

General Terms: Design, Performance

Additional Key Words and Phrases: Addressing schemes, core-shell nanowires, nanotechnology, nanowire crossbars

1. INTRODUCTION

Scientists have developed methods for growing nanowires (NWs) and carbon nanotubes (NTs) whose diameters are on the order of a few nanometers (a few molecular diameters) [Dekker 1999; Cui et al. 2001; Morales and Lieber 1998; Melosh et al. 2003; Johnston-Halperin et al. 2004]. They have also exhibited methods to assemble NWs into *nanoarrays*, crossbars consisting of two orthogonal sets of parallel wires on either side of a molecular layer [Kuekes et al.

This research was funded in part by National Science Foundation Grant CCF-0403674.
Authors' addresses: J. E. Savage, E. Rachlin, Department of Computer Science, Brown University, Providence, RI; email: jes@cs.brown.edu; A. DeHon, Department of Computer Science, California Institute of Technology, Pasadena, CA; C. M. Lieber, Y. Wu, Department of Chemistry and Chemical Biology, Harvard University, Cambridge, MA.

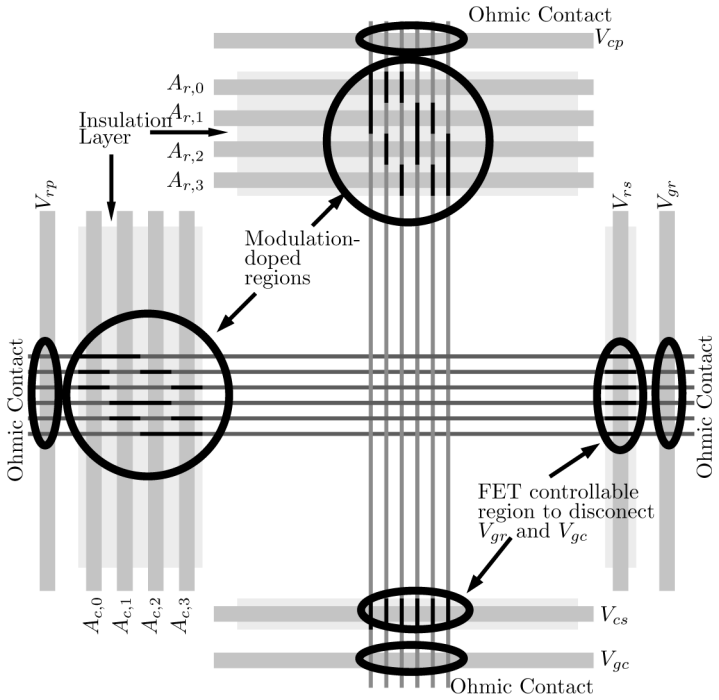

Fig. 1. Crossbar in which NWs in each dimension are addressed by a small set of MWs.

2000; Chen et al. 2003; Melosh et al. 2003; Whang et al. 2003; Zhong et al. 2003; Huang et al. 2001; Kim et al. 2001]. The molecules in this layer are chosen so that, under the application of large positive and negative electric fields, they change their conductivity [Collier et al. 2000; Collier et al. 1999; Rueckes et al. 2000; Duan et al. 2002]. The state of a switch at a crosspoint (defined by a pair of orthogonal NWs) can be sensed without changing its state by application of a smaller electric field. Nanocrossbars have the potential to serve as very high density memories and programmed logic arrays (PLAs) [DeHon 2003; DeHon et al. 2003; Gojman et al. 2004; Rachlin et al. 2005; DeHon et al. 2005]. A prototype 8×8 crossbar with a density of 6.4Gbits/cm^2 has been announced that is based on these technologies [Chen et al. 2003], and a memory with storage capacity of 10Gbits based on crossbars of NTs is promised (see http://www.nantero.com). DeHon et al. [2005] estimate that a memory density exceeding 10^{11} bits/cm^2 is possible.

To read and store data in nanoarrays requires that individual NWs be addressable. That is, it must be possible to select one NW from each orthogonal set of NWs and apply a voltage to it or pass a current through it. To control NWs from the lithographic level requires that mesoscale wires (MWs) be used to address NWs. If each NW is connected to a single MW, the close packing possible with NWs is lost. Thus, schemes are needed that use multiple MWs to control individual NWs. Several such schemes, called *decoders*, have been proposed. All assume that MWs are placed at right angles to NWs as suggested in Figure 1. Here MWs are labeled $A_{r,i}$ and $A_{c,j}$.

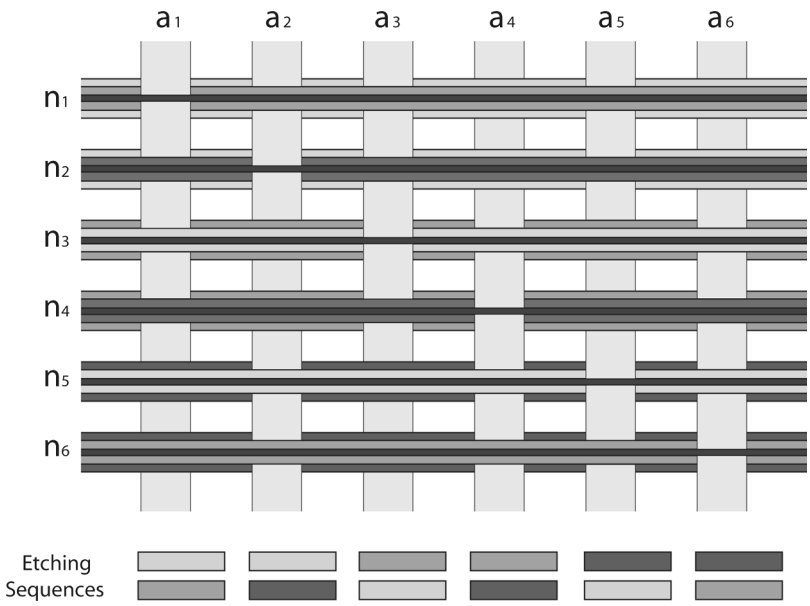

Fig. 2. Shown is the linear decoding method for six (horizontal) radially encoded NWs. An insulated lightly-doped core is exposed to the field on a MW if its sequence of shell types is the sequence associated with that MW. This example uses three different shell types. The boxes below a MW indicate the order of shell removal, from outer (top) to inner (bottom) shell. Each NW represents one of six possible encodings. The etching sequences have been chosen so that, the ith MW controls the ith NW encoding. Applying fields to all but one MW turns off all but one NW.

As explained in Section 2, three types of decoder have been proposed to control MWs. The first decoder assumes that gold particles are placed at random between undifferentiated NWs and MWs. The second assumes that high-K dielectric regions are placed between undifferentiated lightly-doped NWs (they can be controlled by electric fields) and MWs. The third assumes that NWs are differentiated during their manufacture by growing lightly-doped regions into NWs (modulation doping) that have a length equal to the width of MWs.

In this article, we introduce a fourth type of decoder that is competitive with the other three methods in the number of MWs used to control them but that has the advantage that it is less sensitive to the random displacements produced by stochastic assembly. This decoder operates on core-shell NWs [Lauhon et al. 2002], that is, lightly-doped NW cores to which shells of different types are added. In Section 4.2, we describe two variants of the fourth decoder, the linear and logarithmic decoders. They differ in the way shells are used to determine which MWs control which NWs. To illustrate the new decoding method, we give an example of the linear decoder in Figure 2. Here NWs have two shells with three types per shell.

If NWs can have three types of material in the first shell (e.g., a, b, or c), and a different material in the second shell, one MW is used for each type of shell encoding (e.g., $M_{a,b}$, $M_{a,c}$, $M_{b,a}$, $M_{b,c}$, $M_{c,a}$, $M_{c,b}$). In the region reserved for a particular MW (e.g., $M_{b,e}$, the one for shell encoding (b, e)), all shells are

removed from NWs with that shell encoding but at least one insulating shell is left over all other NWs. When an immobilizing electric field is applied to this MW (e.g., $M_{b,c}$), the NWs with the assigned shell encoding (e.g., (b, c)) are turned off, that is, become nonconducting, while the rest remain on, that is, conducting. If immobilizing fields are applied to all but one MW, just the NWs with the corresponding shell encoding remain on.

This new NW encoding method has several advantages: a) correct operation is more certain than with the first encoding method; b) it may use fewer MWs to control NWs than the second method; and c) registration with MWs of doped NW regions is guaranteed by construction, unlike the third method. The new core-shell encoding scheme has the disadvantage that it can increase the pitch of NWs because shells add to NW diameters. However, by making good architectural choices, this encoding method is competitive in NW pitch with other proposed NW encoding schemes. (See Tables III and V.)

Because methods of addressing NWs are also needed to program the junctions in PLAs [DeHon and Wilson 2004], the addressing methods presented in this article also have relevance to these applications.

Section 2 describes the first three methods of addressing NWs that have been developed. These are the randomized contact decoder, the mask-based decoder, and the differentiated NW decoder. Two types of axial NW doping patterns, the (h, λ_A)-hot and binary reflected codes, are described in Section 2.3.1. The problems that arise in controlling these NWs due to misalignment between doped and undoped NW regions and MWs are explored in Section 3. We ask if there exist fail-safe doping patterns, namely, those that guarantee that every NW is either on or off but not in an ambiguous state as a result of misalignment. We also explore whether it is possible that a NW can remain on for all fields applied to MWs.

Radially coded NWs are introduced in Section 4. We describe the method for etching NW shells to provide control of NWs with MWs and give examples of etchants. We introduce the linear decoder for multishell NWs and show that it allows a large number of NW types to be decoded using a small number of shell types and shells. We also describe a logarithmic decoder that controls NWs with a smaller number of MWs at the cost of fewer NW types for the same number of shell types and shells. The logarithmic decoder allows for fault tolerance with regard to etching. We also show that the etching procedures needed to implement each decoder can be executed efficiently in parallel.

Hybrid NW codes and a decoder for them are introduced in Section 5. These are codes for core-shell NWs in which the cores contain an axial code and the shells provide radial encoding. We present a very efficient decoder that assumes that each axial code is a binary reflected code. The sensitivity of the BRC hybrid decoder to displacements is examined in Section 5.2.

The effectiveness of the different NW encoding strategies is examined in Section 6. We find that hybrid codes are inferior to axial and radial codes in terms of effective pitch. When axial and radial codes are compared in the same terms, axial codes are seen to have a slight advantage except when the likelihood of NW misalignment of axial codes is high. Given that axial codes appear to be more difficult to manufacture than radial codes and that shells are useful for

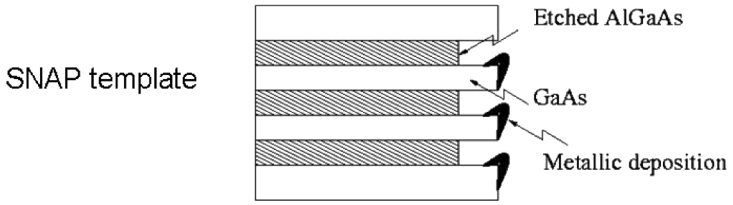

Fig. 3. The SNAP process grows NWs by a) forming a superlattice using molecular-beam epitaxy, b) etching away alternating layers in the superlattice, c) depositing metal on the superlattice edges, and d) pressing the metal wires onto an adhesive layer of a chip.

other purposes, radial codes are shown to be a viable alternative to axial codes. Conclusions are drawn in Section 7.

2. PRIOR WORK ON NANOWIRE ADDRESSING

We describe three known methods of addressing differentiated and undifferentiated NWs with MWs. Each has associated with it a circuit(s), called a *decoder* (s), that makes one NW conductive (turns it on) and the rest nonconductive (turns them off).

2.1 Randomly Addressed NWs

The first method of addressing NWs, the *randomized contact decoder*, assumes that undifferentiated NWs are arranged in parallel. Gold particles are deposited at random between MWs and the NWs with the goal of placing gold particles at about half of the junctions formed by MWs and NWs [Williams and Kuekes 2001]. The difficulty of achieving this goal has not been assessed.

Under these assumptions, it has been shown that, with high probability, the randomized contact decoder uses $5 \log_2 N$ MWs to control N NWs. That is, with this many MWs, it is possible to select an arbitrary one of N NWs to be conducting and the rest nonconducting.

2.2 Undifferentiated NWs

The second method uses long, undifferentiated NWs. They can be grown using molecular beam epitaxy (MBE) (the SNAP method) [Melosh et al. 2003] or nanoimprinting [Austin et al. 2004].

In the SNAP method a superlattice is formed consisting of alternating layers of two materials, such as Aluminum Gallium Arsenide (AlGaAs) and Gallium Arsenide (GaAs), and one type of material, such as AlGaAs etched back to create notches. The superlattice is turned and metal deposited on the exposed edges (see Figure 3.) The superlattice is then pressed onto a chip that contains a thin layer of an adhesive. When removed, long straight NWs are deposited. These NWs have uniform diameter and pitch unlike the modulation-doped NWs described in the following, that are assembled fluidically.

In the nanoimprinting method, a template is grown, perhaps using MBE, and the template is pressed against a soft polymer, thereby creating a contrast pattern in the polymer. Anisotropic etching removes the thin regions, thereby exposing the substrate for doping. The metallic NWs deposited by the SNAP

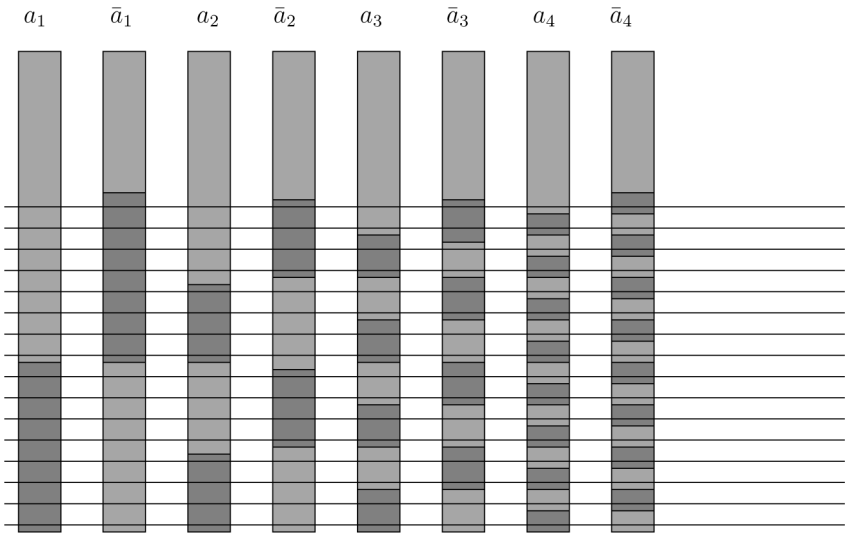

Fig. 4. A mask-based decoder for undifferentiated NWs. Rectangular high- and low-K dielectric regions are interposed between vertical MWs and horizontal NWs. The low-K regions shield NWs from the effect of electric fields applied by MWs. When a field is applied to either a_i or $\bar{a}_i$ for $1 \leq 4$, exactly one of the 16 NWs conducts.

method can also serve the same purpose. If the surface has a thin layer of Si on SiO_2 which in turn is on a substrate, the SNAP metallic wires can be used with etching to expose Si NWs [Johnston-Halperin et al. 2004].

The method proposed to address undifferentiated NWs, called the *randomized mask-based decoder* [Heath and Ratner 2003], uses lithographically defined rectangular regions of low-K dielectric to shield NWs from the fields associated with MWs. As suggested in Figure 4, if dielectric regions as small as the pitch of NWs can be produced lithographically, electric fields applied to one of a_i or $\bar{a}_i$ for $1 \leq i \leq \log_2 N$, cause exactly one of N NWs to remain conducting. Here a NW separated from a MW by high-K dielectric acts as a field effect transistor (*FET*); the application of an electric field of the appropriate strength to the MW immobilizes carriers and drives the conductance of the NW to near zero.

Because lithography puts a lower limit on the size of such regions, many randomly shifted copies of the smallest regions are used instead of the decreasingly smaller regions. The number of MWs needed to control N NWs with the mask-based decoder has been analyzed [Rachlin et al. 2005]. Under very reasonable assumptions, it has been shown that, as NW pitch decreases, at least $2\log_2 N + 46$ MWs will be needed. Although this number is large, the NWs grown with the SNAP process are expected to be much longer and more uniformly spaced than the modulation-doped NWs described next.

2.3 Modulation-Doped NWs

The third method, called the *differentiated NW decoder*, uses NWs that are grown from seed catalysts through a vapor-liquid-solid (VLS) process depicted in Figure 5 [Gudiksen et al. 2002; Wu et al. 2002; Björk et al. 2002]. In this

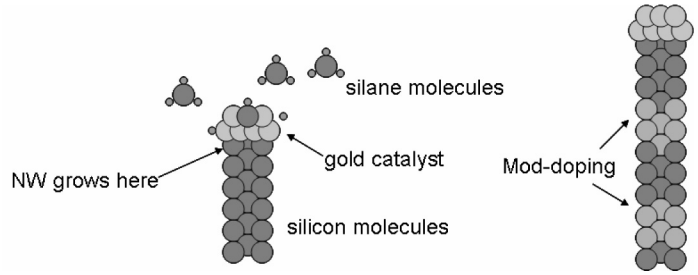

Fig. 5. (a) NWs grown through a vapor-liquid-solid process. (b) NWs are doped as they grow.

example, silane molecules (SiH_4) fall onto gold clusters, precipitating out Si atoms that solidify into crystalline silicon NWs. These NWs can be differentiated by adding dopant molecules to the gaseous mixture as they grow. NWs can be heavily and lightly doped over lengths that are determined by exposure time. This process is called *modulation doping* when referring to the doping process and *axial doping* when referring to the result.

As stated earlier, when a MW is placed at right angles to a lightly-doped region of a NW and separated from it by high-K dielectric, the MW and NW act as a FET. The doping levels are chosen so that the same field has no effect on heavily-doped regions. We say that lightly-doped regions are *controllable* and the heavily doped regions are *uncontrollable*.

We assume that each NW is given a pattern of controllable and uncontrollable regions, each of the same length. For example, two of four regions could be made controllable as suggested in Figure 1 where all six different doping patterns are shown.

Many axially-doped NWs with the same doping profile are assembled at the same time and collected in solution. The VLS process is repeated until each of the desired doping profiles is produced. NWs are then assembled on a chip using a fluidic process. NWs with different doping profiles are mixed and floated onto the surface of a liquid where baffles align them in parallel. NWs are deposited by passing the chip up through the liquid. After drying, lithography is used to trim the NWs deposited in this manner. To produce a crossbar, this procedure is applied again after turning the chip by 90°. Unfortunately, this process cannot guarantee that NWs will have a uniform separation nor can it guarantee that the boundaries of doped regions will be aligned with one another or any point on the chip.

When NWs are placed on a chip, insulation is used to separate the NWs from MWs that are superimposed on them. This combination of NWs and MWs forms an addressing circuit called an *encoded NW decoder*.

As with the previously described decoders, this encoder exhibits randomness. In this case, the types of NW doping patterns that fall on a chip cannot be predicted in advance. Thus, it is necessary to test the chip to discover which NW doping patterns are present. For applications that require deterministic addresses, such as memories, an auxiliary *translation memory* is then needed to translate fixed external addresses into the particular doping patterns that are deposited on the chip during assembly. An important factor affecting

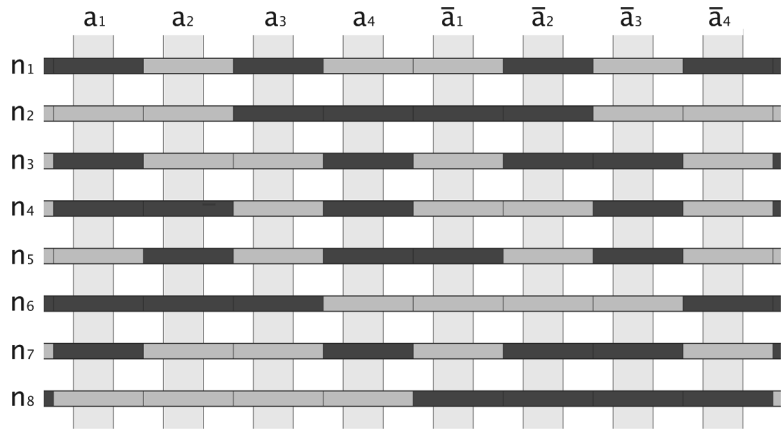

Fig. 6. Modulation-doped NWs encoded with a binary reflected code of length eight.

manufacturability is the number of different doping patterns, C, (the size of the code space) that is needed to ensure that all or nearly all of the NWs have different doping patterns.

2.3.1 *Axial Doping Patterns.* Two types of axial codes have been proposed, (h, λ_A)-hot codes [DeHon 2003] and length λ_A binary reflected codes (λ_A-BRCs) [Gojman et al. 2004, 2005]. To describe them, we assume that controllable NW regions are aligned with MWs [DeHon et al. 2003].

In an (h, λ_A)-hot code, exactly h of λ_A regions are controllable. To select one codeword, disabling fields are applied to $(\lambda_A - h)$ MWs. The one codeword type whose controllable regions coincide with the MWs to which no field is applied remains conductive.

A λ_A-BRC has an even number of regions. The doping pattern in the first $\lambda_A/2$ regions is denoted by an arbitrary binary $(\lambda_A/2)$-tuple $\boldsymbol{x}$ (1s (0s) denote controllable (uncontrollable) regions). The doping pattern in the second $\lambda_A/2$ regions is denoted by the Boolean complement of $\boldsymbol{x}$. A single λ_A-BRC codeword is selected by applying fields to the MWs that correspond to uncontrollable regions. The one codeword type whose controllable regions coincide with the MWs to which no field is applied remains conductive. The doping patterns for λ_A-BRC are a subset of the doping patterns of (h, λ_A)-hot code.

An example of a binary reflected code with eight controllable or uncontrollable regions that are aligned with MWs is shown in Figure 6. When the 2nd, 3rd, 5th, and 8th MWs are turned on, the 3rd and 7th NWs, both of which have the same doping pattern, are activated, while all others are turned off.

2.3.2 *Addressability of Modulation-Doped Nanoarrays.* Using (h, λ_A)-hot codes, DeHon et al. [2003] show that with high probability N modulation-doped NWs can be controlled with $\lceil 2.2 \log_2 N \rceil + 11$ MWs when the design goal is that all NWs doping patterns be different. Using binary reflected codes and the assumption that at least half of the NWs have different doping patterns,

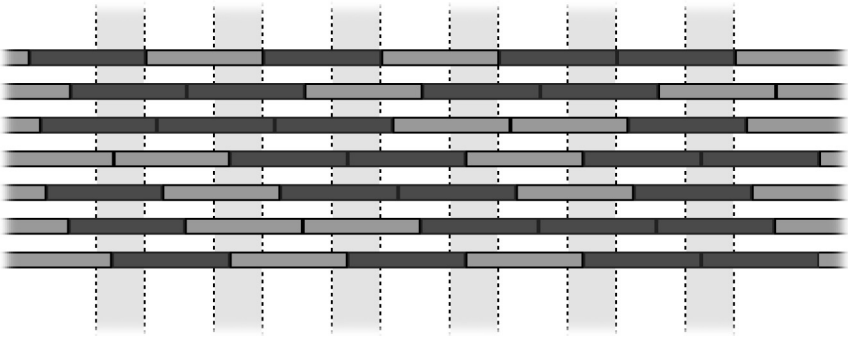

Fig. 7. NWs offset by random amounts from vertical MWs.

Gojman et al. [2004] show that this number can be reduced to $2\log_2 N + 8$ MWs, although a somewhat better upper bound might be obtained for (h, λ_A)-hot codes. They also analyze the area needed for the translation memory.

3. MISALIGNMENT OF AXIAL CODES

Because fluidic assembly methods cannot control the lengthwise displacement of NWs, alignment between MWs and NW controllable regions cannot be guaranteed (see Figure 7). To compensate for this problem, doping patterns are repeated along the length of NWs. Even with this accommodation, it remains possible that the overlap between NW controllable regions and MWs will be so small that the control of MWs cannot be definitely guaranteed; that is, a MW may be able to reduce the conductivity of a NW but not effectively turn it off. We say that such a NW is in an *ambiguous state*. The alignment problem is compounded by the difficulty of making sharp transitions between controllable and uncontrollable NW regions during the VLS manufacturing process.

To quantify the effect of misalignment, let $W_{overlap}$ be the minimal length overlap needed between the field of a mesoscale wire and a NW to reduce the conductivity to a satisfactory level (see Figure 8). Let W_{pitch} be the pitch of MWs. Since all shifts of NWs relative to MWs are equally likely, it follows that the probability, $P_{control}$, that a NW is controlled by a MW is $P_{control} = (1 - 2W_{overlap}/W_{pitch})$, a quantity that is used in Section 6 to compare NW encoding strategies [DeHon et al. 2003].

To cope with misalignment it would be desirable to find a control strategy consisting of a NW doping pattern and a MW activation strategy so that a NW is either on or off and not in an ambiguous state. Such strategies are called *fail-safe*. We show that such strategies don't exist. If a strategy is not fail-safe, we ask if it is possible to guarantee that misalignment can never result in some NW that is on for all applications of fields to MWs. These are called *on failures*. If such failures exist, they are disastrous because it would be impossible to control any NWs in the dimension of the crossbar containing such NWs. We show that on failures can't occur when doped regions are a bit longer than undoped regions.

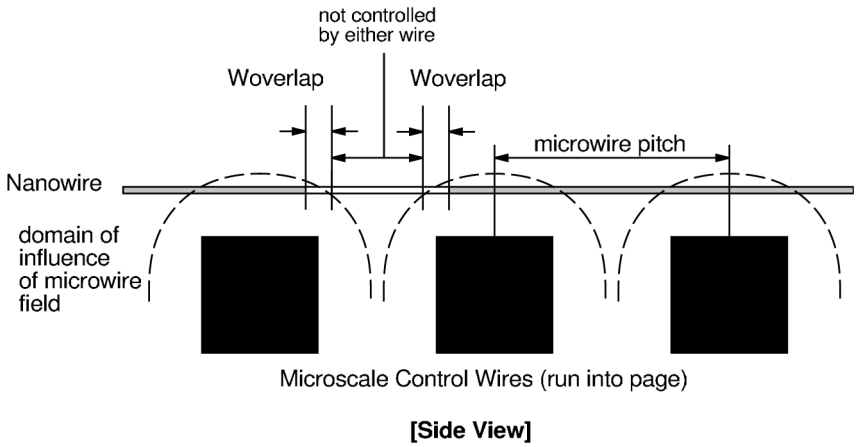

Fig. 8. Schema for calculating the probability of loss of control. The lightly shaded NW region is controllable. When the overlap of this region and the electric field is $W_{overlap}$ or less, the NW cannot be sufficiently controlled.

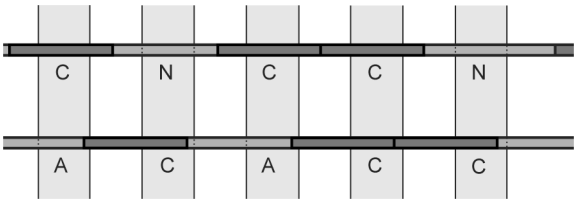

Fig. 9. Illustration of the status of MWs as a result of NW misalignment.

3.1 Fail-Safe Strategies for Axial Codes

Before formally showing that fail-safe doping patterns are not possible, we give an informal argument of this fact. Consider either the (h, λ_A)-hot code or the λ_A-BRC doping patterns. A given MW activation pattern causes a set of identically doped NWs to be conducting, while all differently doped NWs become nonconducting. If a conducting NW is shifted by a MW pitch, it becomes nonconducting because it represents another doping pattern. In between these two states, the NW must transition from being on to being off which means that its state must be ambiguous for some shift of less than a MW pitch. This informal argument shows that fail-safe doping patterns don't exist.

To formally characterize the misalignment problem we introduce some terminology. First, we recall that doping patterns are determined at manufacturing time.

Second, we examine the influence that MWs have on NWs (see Figure 9). We say that a MW is *controlling (C)* if it is aligned sufficiently well with a controllable region of a NW that the NW can be turned off. It is *noncontrolling (N)* if has the same relationship with the region but the region is uncontrollable. A MW is *ambiguous (A)* if it is sufficiently close to a controllable region to have some influence on its conductivity but not so much as to turn it off. Given the placement of a NW relative to MWs, we characterize the influence of λ MWs on

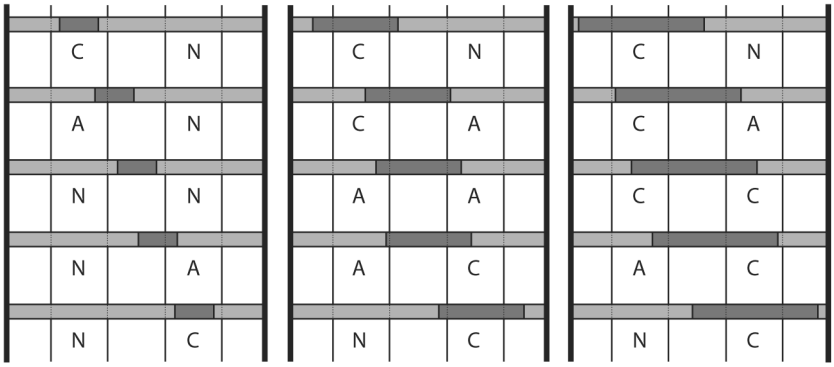

Fig. 10. The controllability of a pair of MWs as a function of the length of the lightly-doped regions. The leftmost case occurs when these regions are very small, and the rightmost case occurs when the lengths are large enough so that some MW is always controlling a lightly-doped region. The intermediate case is shown in the middle.

the NW by a *control pattern* $\boldsymbol{c}$, a λ-tuple over the set {C,N,A}. It is easy to see that some displacement of each NW with respect to MWs leads to a MW that is ambiguous relative to that NW.

An *activation pattern* $\boldsymbol{a}$ is the application of a field (1) or no field (0) to MWs. It is a λ-tuple over the set {0,1}. Thus, there are $A = 2^{\lambda}$ different activation patterns.

A NW with control pattern $\boldsymbol{c}$ is said to *fail* relative to an activation pattern $\boldsymbol{a}$ if the conductivity of the NW is not at its maximum or minimum value.

An activation pattern $\boldsymbol{a}$ *fails* for a set of NWs if no NW is fully conducting when $\boldsymbol{a}$ is applied.

If a control strategy is fail-safe, a prechosen activation strategy can be applied to a set of NWs assembled with a doping strategy without fear that any NW will be left in an ambiguous state.

LEMMA 3.1. *Fail-safe control strategies do not exist.*

PROOF. If a fail-safe control strategy exists, then every activation pattern must leave all NWs either conducting or nonconducting. Consider an activation pattern $\boldsymbol{a}$ that causes a NW with control pattern $\boldsymbol{c}$ to conduct. The control pattern $\boldsymbol{c}$ for that NW has N's under MWs that carry a field. A shift of that NW in one direction or the other will eventually result in at least one of those N's turning into an A. At the first occurrence of this event, the control pattern will behave ambiguously with respect to $\boldsymbol{a}$. □

3.2 On Failures Can Be Avoided with Axial Codes

Now that we know that fail-safe control strategies don't exist, we ask whether on failures can be avoided for the two proposed doping patterns. The good news is that they can be avoided for reasonable assumptions on the lengths of the controllable and uncontrollable regions as shown in the following.

We examine possible scenarios as a NW shifts to the right that are depicted in Figure 10. Assume that the NW has three consecutive regions that are heavily

doped, lightly doped, and heavily doped, moving from left to right, and that the two regions on the right are centered under two MWs. If the doped region is very short, an undoped region is long, and the field strength applied by MWs is low. As the NW shifts to the right, the influence of the two MWs begins as *CN* and changes to *AN*, *NN*, *NA*, *NC*. With a somewhat higher field strength or longer doped region, the influence of two adjacent MWs begins as *CN* and changes to *CA*, *AA*, *AC*, and *NC*. Finally, the change is from *CN* to *CA*, *CC*, *AC*, *NC*. We say that doped regions producing the last set of changes are *normal* because some control is always kept over a doped region. When the length of a doped region is the length of the region between two MWs (see Figure 8) plus $2W_{overlap}$, the doped regions are normal.

LEMMA 3.2. *The* (h, λ_A)*-hot codes and binary reflected codes of length* λ_A *do not exhibit on failures when their doped regions are normal.*

PROOF. For normal doped regions, the number of *C*s and *A*s in misaligned NWs is more than h for (h, λ_A)-hot codes and more than $\lambda_A/2$ for λ_A-BRCs. Since the number of MWs carrying fields is $\lambda_A - h$ for the first type of code and $\lambda_A/2$ for the second, misalignment may cause the conductivity of a NW to be ambiguous but it will not leave a NW fully conducting or on. □

In summary, misalignment in (h, λ_A)-hot and binary reflected codes results in wires that might not turn on when they should but will always turn off when they should. Consequently, misalignment errors are not catastrophic.

3.3 Codeword Activation

As mentioned, testing is needed to discover which NW doping patterns have been assembled on a chip. If there is some way to test a crossbar to determine which activation patterns are ambiguous, these activation patterns can be avoided. This may be possible by forming a voltage divider between the selected NW and an external resistance at V_{gr} (see Figure 1). We can then set a voltage threshold for acceptance/rejection of the NW selection. If the voltage is above the threshold, the selected NW resistance is low enough to be accepted; if the voltage is below the threshold, the NW selection resistance is too high and we reject it.

In light of the preceding discussion, we observe that an activation pattern may turn on many NWs. Thus, we can associate a set of NWs with an activation pattern. There are $A = 2^{\lambda}$ different activation patterns. If an activation pattern activates only NWs that are ambiguous, we say that the pattern ambiguously activates a set of NWs. If two different activation patterns activate overlapping sets of NWs, the result is problematic. In this case, the data recorded or read with one activation pattern may conflict with data read or recorded by the other pattern. This leads to the following definition.

Definition 3. A code is a set of activation patterns with the property that no activation pattern ambiguously activates a set of NWs and the sets of NWs that are activated are disjoint.

4. RADIALLY CODED NANOWIRES

Core-shell NWs are assembled by adding shells to NWs with lightly-doped cores [Lauhon et al. 2002]. The sequence of shell materials grown around a NW core are called its *radial encoding*. We propose using radial encodings to differentiate and control NWs.

Axial NW growth occurs when reactant activation and addition occurs at the catalyst site (the gold cluster in Figure 5) and not on the NW surface. Radial NW growth occurs when the conditions are changed to favor homogeneous vapor phase deposition on the NW surface. Multiple shells of nearly arbitrary composition are possible although epitaxial growth of these shells requires consideration of lattice structures.

Assume now that core-shell NWs are produced with $n + 1$ shells. The innermost shell is assumed to be an inert high-K dielectric, such as hafnium oxide, HfO_2, that covers a lightly-doped core. This dielectric is insensitive to etchants used to remove any of the other shell materials.

A core-shell, or radially encoded NW is controlled with a metallic MW as follows.

(1) Under the future location of the MW, etchants remove the NW's n outer shells, exposing its lightly-doped, high-K dielectric-covered core.
(2) The MW is deposited over the dielectric in the space from which the outer shells were etched; the NW/MW junction forms an FET.
(3) If an etching procedure does not remove all etchable shells, the remaining shell(s) is (are) assumed to be thick enough that the influence of the field applied by a MW to the lightly-doped core is sufficiently small that its impact on the resistance of the core is minimal.

A set of such junctions is called a *radial decoder* (see Figure 2) if for each NW encoding they allow only NWs with that encoding to be made conducting. In Section 4.2, we describe two types of radial decoder, linear and logarithmic decoders.

As we shall see, core-shell NWs have a key advantage. They are insensitive to random lengthwise displacement unlike modulation-doped NWs. As a consequence, NWs cannot be in an ambiguous state of conduction due to misalignment of NWs and MWs. Furthermore, when a MW is deposited over exposed cores, the MWs will self-align with them [Glasser and Dopperpuhl 1985], thereby obtaining far superior registration between controllable NW regions and MWs than can be obtained with modulation-doped NWs. The degree of sharpness of the vertical etched walls in the space reserved for MWs is not important because the variation in walls is measured in a few nanometers, whereas the MWs will have widths of several tens of nanometers. Also, variations in the radius of cores and dielectric shells are expected to be minimal and not present any difficulties with respect to metallic MW deposition. Evidence for this as well as for the ease with which four or more shells can be placed on a core are reflected in recent experiments on the deposition of shells for NW-based light-emitting diodes [Quian et al. 2005].

Table I.

	Etchant			
Etchable outer NW shells	$FeCl_3$	$NaOH$	$C_4H_4KNaO_6 + HCl + H_2O_2 + H_2O$	$C_6H_8O_7 + H_2O_2$
Cu	Will etch	Will not etch	Will not etch	Will not etch
Al_2O_3	Will not etch	Will etch	Will not etch	Will not etch
$GaSb$	Will not etch	Will not etch	15:1 with regard to $InAs$	1:100 with regard to $InAs$
$InAs$	Will not etch	Will not etch	1:15 with regard to $GaSb$	100:1 with regard to $GaSb$

Four configurations of core-shell NW are shown on the left. Each has a silicon (Si) core, a hafnium oxide shell (HfO_2) (a high-K dielectric), and outer shells made of one of four materials. The columns list the effect of etchants on the outer shells. The etchants in the two right columns etch both $InAs$ and $GaSb$. The third etchant etches $GaSb$ 15 times faster than $InAs$, while the fourth etchant etches $InAs$ 100 times faster than $GaSb$.

Core-shell NWs have the disadvantage that the addition of shells increases their diameter and the spacing between them, thereby sacrificing area. However, extra NWs do not have to be added to account for the fact that some NWs are in an ambiguous state due to misalignment. Furthermore, NWs appear to be more tolerant of variations in the manufacturing process and are compatible with other methods of controlling NWs.

4.1 Etching Shells

Creating a radial decoder requires an etching process that exposes the cores of NWs with a particular type of radial encoding, while leaving other cores insulated. To allow for this, shell materials must be sufficiently different that an etch for one type of material has little or no effect on another.

There are a number of possible core/multishell nanowire heterostructures that could be prepared using atomic layer and/or chemical vapor deposition (CVD) methods such that multiple shells could be selectively etched. Four distinct shell materials are highlighted in Table I, namely, Cu, Al_2O_3, $GaSb$, and $InAs$, along with specific wet etching solutions that can differentiate anyone of the materials from another as required to implement the decoder [Dier et al. 2004].

The specific implementation would further include two other specific features. First, an integral high-constant dielectric (high-K) shell of, for example, hafnium oxide (HfO_2), would be deposited using CVD prior to the growth of the selectively etchable shells in order to simplify the overall fabrication process. The high-K dielectric increases the coupling between the NW and controlling MWs. Second, the first shell outside of the high-K dielectric shell would be made thicker than subsequent shells to ensure sufficient differential gate response of the mesoscale wires in contact with the high-K dielectric on the inner shell.

In our discussion of radial decoders, we assume that each shell of each NW can be made from any one of α independently etchable materials. We denote the materials $m_1, \ldots, m_\alpha$ and use $Etch(m_i, W)$ to refer to the etching procedure that removes just material m_i from the region under MW W.

Writing $[Etch(m_i, W), \ldots, Etch(m_j, W)]$ means that $Etch(m_i, W)$ is followed by $Etch(m_{i+1}, W), \ldots, Etch(m_j, W)$. For an arbitrary set of materials, M, writing $Etch(M, W)$ means that for all $m_i \in M$, all $Etch(m_i, W)$ are applied one after the other in some arbitrary order.

In a radial encoding, the innermost shell material is listed first. In an etching sequence, operations on the outer shell appear first. Thus, if NWs N_a and N_b have radial encodings (m_i, m_j) and (m_j, m_i), respectively, the sequence $[Etch(m_j, W), Etch(m_i, W)]$ exposes the core of N_a under MW W, but only removes the outer shell of N_b. A MW W can then control N_a without affecting N_b.

4.2 Linear Decoder for MultiShell Radial Codes

A radial decoder allows MWs to control NWs manufactured with some set of radial encodings. We consider a family $\mathcal{F}$ of radial encodings in which each possible encoding uses n shells, and each shell is one of $\alpha \geq 1$ possible materials. In order to ensure that shells can be removed one at a time under one MW, we assume that consecutive shells are never made of the same material. When manufacturing a NW, the first shell can be of any of α types, but each additional shell must be a different type from that of the preceding shell, which implies that $N = |\mathcal{F}| \leq \alpha(\alpha-1)^{n-1}$. This means that 4 independently etchable materials and 2 shell layers can lead to 12 different shell encodings. If 5 independently etchable materials are used, 20 shell encodings can be generated with two shells and 80 encodings with three shells.

Given an arbitrary family of radial encodings, $\mathcal{F}$, a *linear decoder* uses a separate MW for each radial encoding. If MW W_i is associated with encoding $E_i = (m_{j_1}, \ldots, m_{j_n})$, then we apply $[Etch(m_{j_n}, W_i), Etch(m_{j_{n-1}}, W_i), \ldots, Etch(m_{j_1}, W_i)]$. The etching sequence exposes the cores of NWs with encoding E_i under W_i. The cores of NWs with other encodings remain unexposed since for these NWs at least one etching step will fail to remove a shell. Each of the $\lambda_R = |\mathcal{F}|$ MWs can turn off exactly the NWs with a particular radial encoding. The decoder shown in Figure 2 is a linear decoder.

Note that the linear decoder etches one material in each shell under each MW. Thus, it performs $n\lambda_R$ etching operations and an equal number of masking and unmasking operations, where n is the number of shells and λ_R is the number of MWs. This number can be reduced to $n\alpha$ etching operations by observing that the etching operations under each MW that remove the same material can be done together after first masking all other MWs from etchants and demasking these regions when done.

LEMMA 4.1. *The linear NW decoder for $N = \alpha(\alpha-1)^{n-1}$ NW types containing n shells of α types can be implemented using $\lambda_R = N$ MWs in $n\alpha$ etching operations.*

4.3 Logarithmic Decoder for Single-Shell Radial Codes

A standard decoder uses $\lceil \log_2 N \rceil$ binary inputs to select one of N outputs. In order to describe a radial decoder with a logarithmic number of MWs, we first consider the case when NWs have a single shell and the family of radial

encodings, $\mathcal{F}$, uses $\alpha = |\mathcal{F}|$ materials. We show that it is possible to control radial encoding NW type with $\lambda_R = 2\lceil \log_2 \alpha \rceil$ MWs.

To control the α radial encodings with λ_R MWs, we first associate each material with a distinct binary string of length $L \leq \lceil \log_2 \alpha \rceil$. Along each dimension of the nanoarray, a pair of MWs is associated with each of the L bits. Let MWs $W_{0,i}$ $W_{1,i}$ be the two MWs associated with the ith bit. Let $M_{0,i}$ and $M_{1,i}$ be the materials having 0 and 1 as their ith bit, respectively, in their binary encodings. Apply $Etch(M_{0,i}, W_{0,i})$ and $Etch(M_{1,i}, W_{1,i})$. NWs with a shell made of a material in $M_{0,i}$ have their core exposed under $W_{0,i}$, while all other NWs (those in $M_{1,i}$) have their core exposed under $W_{1,i}$.

Each radial encoding consists of a single shell made of some material. Each material is associated with a distinct binary sequence. Each material is etched away under a distinct set of exactly L MWs. To turn off all NWs that do not have a particular radial encoding, E, simply apply a voltage to all MWs that do not control E. Every other encoding must be controlled by at least one of these MWs, hence only NWs with encoding E will remain conducting. Thus, the number of controllable NWs in each dimension of a crossbar is α and the number of MWs used is $\lambda_R = 2L = 2\lceil \log_2 \alpha \rceil$.

Again, note that $|M_{0,i}|$ ($|M_{1,i}|$) etching operations are performed under MW $W_{0,i}$ ($W_{1,i}$). Since $|M_{0,i}| + |M_{1,i}| = \alpha$ and there are at most $\lceil \log_2 \alpha \rceil$ pairs MWs $W_{0,i}$ and $W_{1,i}$, the number of etching operations performed by the logarithmic method on one shell is at most $\alpha \lceil \log_2 \alpha \rceil$.

The number of operations can be reduced to α etching, masking, and demasking operations if these steps are done under multiple MWs at the same time. To see this, observe that, for each i, the materials in $M_{0,i}$ and $M_{1,i}$ are etched under $W_{0,i}$ and $W_{1,i}$ in some arbitrary order. Thus, we can order the materials and etch, mask, and demask NWs under MWs in parallel. In particular, the new algorithm executes three steps for each material of type t, $1 \leq t \leq \alpha$. It begins by removing masks over regions under MWs. It then applies a mask to the region under $W_{a,i}$ if $t \notin M_{a,i}$, after which it etches NWs under $W_{a,i}$ if $t \in M_{a,i}$.

LEMMA 4.2. *The logarithmic decoder for NWs with one shell and $N = \alpha$ shell types can be implemented with $2\lceil \log_2 N \rceil$ MWs and N etching operations.*

4.4 Decoders for MultipleShell Radial Codes

In order to extend our logarithmic radial decoder to multiple shell encodings, the encodings in $\mathcal{F}$ must be restricted. The restriction prevents every possible shell material from appearing in each shell but allows for more powerful etching operations.

The $\alpha = |\mathcal{F}|$ shell materials are divided into two disjoint sets of size α_1 and α_2, $\alpha_1 + \alpha_2 = \alpha$. The materials used to form the ith shell of each radial encoding is chosen from the first set when i is odd, and from the second set when i is even. This allows for $N = \alpha_1^{\lceil n/2 \rceil} \alpha_2^{\lfloor n/2 \rfloor}$ possible encodings, which implies that $N \leq (\alpha/2)^n$ when n is even.

Let σ_i denote the materials that can appear in the ith shell. σ_i simply depends on the parity of i. Let M be an arbitrary set of materials. When no

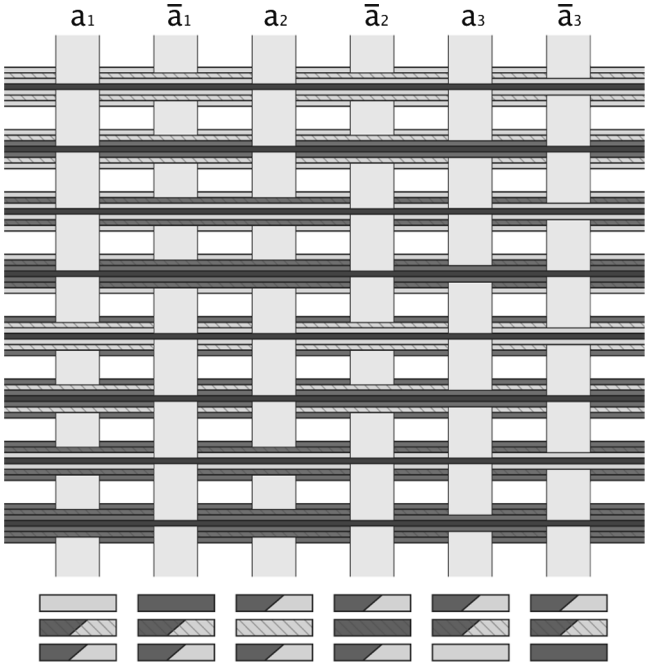

Fig. 11. Shown is a logarithmic decoder for eight (horizontal) NWs each having three shells of two types. The insulated core of a NW is exposed to the field on a MW if the NW has a particular type in a particular shell. Applying an immobilizing field to one of the (vertical) MWs labeled a_i and $\bar{a}_i$ causes one half of the NWs to have very high resistance. The three shaded rectangles below MWs indicate which shell material(s) is (are) removed in each shell; the top rectangle corresponds to the outer shell. If two types of shading occur in a rectangle, both materials in a shell are removed; otherwise only one type is removed.

other etchings have been done, $[Etch(\sigma_n, W), Etch(\sigma_{n-1}, W), \ldots, Etch(\sigma_{s+1}, W)]$ will remove the outermost $n - s$ shells of every NW in the region under W. $Etch(M, W)$ then removes a shell from only NWs with a material in M in their sth shell. $[Etch(\sigma_{s-1}, W), \ldots, Etch(\sigma_1, W)]$ then exposes the cores of these NWs. Denote this entire three-part procedure as $LayerEtch(M, W, s)$.

Our restriction on $\mathcal{F}$ ensures that σ_i and σ_{i+1} are disjoint and that $Etch(\sigma_i, W)$ does not remove more than one shell at a time. This implies that $LayerEtch(M, W, s)$ can only expose the cores of NWs that are affected by the $Etch(M, W)$ operation. $LayerEtch(M, W, s)$ thus exposes the cores of exactly those NWs with a material in M in their sth shell.

The procedure $LayerEtch(M, W, s)$ can be applied to any sets M of materials in the sth shell that are used by an etching procedure. In particular, it can be used with the etching procedure for the linear or logarithmic decoder on one shell. When the shell etching procedure is linear (logarithmic), call the decoders resulting from these procedures *LinearLog* and *FullyLog*, respectively. The two decoders which are the same when $|\sigma_i| = 2$ are illustrated in Figure 11.

4.4.1 *The FullyLog Decoder.* In etching for the FullyLog decoder, the set M assumes values $M_{0,i}$ or $M_{1,i}$, as defined above. Thus, the FullyLog uses

Table II.

Shell	W_r^1	W_g^1	W_b^2	W_y^2	W_r^3	W_g^3
1	r		r	r	r	r
		g	g	g	g	g
2	b	b	b		b	b
	y	y		y	y	y
3	r	r	r	r	r	
	g	g	g	g		g

Shown are the materials etched in each shell under each MW for the decoder of three-shell NWs shown in Figure 11. Shell types r and g are used in the outer and inner shells, and b and y are used in the inner shell. The etching procedure exposes under MW W_t^s the cores of NWs that have material t in shell s. For example, under W_b^2 the cores of NWs are exposed that have type b in the middle shell. Etching begins with the first or outer shell. The FullyLog procedure operates on one column at a time. All but one column is exposed at a time, and all cells in that column are removed in sequence. For example, under MW W_r^1 (W_g^1), it removes shells of type r (g) followed by both types of shell in successive shells. This procedure executes 36 etching steps. The fast version of FullyLog procedure etchs all cells in one row in parallel in one step except for the cell in one column that is masked. For example, it removes outer shell of type r under all MWs except for W_g^1. It them removes all shells of type g under all MWs except for W_r^1, etc. This procedure executes 6 etching steps.

$M_{FullyLog} = \sum_{r=1}^{n} 2\lceil \log_2 |\sigma_r| \rceil$ MWs. Since $|\sigma_s| = \alpha_1$ when s is odd and $|\sigma_s| = \alpha_2$ when s is even, $M_{FullyLog} = 2(\lceil n/2 \rceil \lceil \log_2 \alpha_1 \rceil + \lfloor n/2 \rfloor \lceil \log_2 \alpha_2 \rceil)$.

We now consider the number of operations that are needed to expose NWs to MWs. Let W_t^s be the tth MW corresponding to the sth shell. It follows that FullyLog requires $|\sigma_r|$ etching steps under this MW in the rth shell for $r \neq s$ for a total of $T - |\sigma_s|$ operations where $T = \sum_{r=1}^{n} |\sigma_r|$. There are $2\lceil \log_2 |\sigma_s| \rceil$ MWs corresponding to the sth shell, and they require $|\sigma_s| \lceil \log_2 |\sigma_s| \rceil$ additional etching operations. Thus, *FullyLogEtch* performs $2(T - |\sigma_s|)\lceil \log_2 |\sigma_s| \rceil + |\sigma_s| \lceil \log_2 |\sigma_s| \rceil$ operations for W_t^s, $1 \leq t \leq |\sigma_s|$, and a total of $E_{FullyLog} = 2T \sum_{s=1}^{n} \lceil \log_2 |\sigma_s| \rceil - \sum_{s=1}^{n} |\sigma_s| \lceil \log_2 |\sigma_s| \rceil$ operations overall.

It is easy to see that the FullyLog etching procedure implements NW decoders for N NW types with n shells using $O(n^2)$ etching steps. Faster algorithms can be implemented by performing operations in parallel.

As with previous decoders, the FullyLog decoder can be implemented with fewer etching steps when they are done in parallel. As suggested in Table II, they can be done in $\sum_{r=1}^{n} \sigma_r$ etching operations.

THEOREM 4.1. *The FullyLog decoder for n-shell NWs having α_1 (α_2) materials in odd-(even-)indexed shells and $N = \alpha_1^{\lceil n/2 \rceil} \alpha_2^{\lfloor n/2 \rfloor}$ NW types can be implemented with $M_{FullyLog} = 2(\lceil n/2 \rceil \lceil \log_2 \alpha_1 \rceil + \lfloor n/2 \rfloor \lceil \log_2 \alpha_2 \rceil)$ MWs, and $E = \lceil n/2 \rceil \alpha_1 + \lfloor n/2 \rfloor \alpha_2$ etching operations. When $\alpha_1 = \alpha_2 = \alpha/2$ and n is even, $N = (\alpha/2)^n$, $M_{FullyLog} = 2\lceil \log_2 N \rceil$, and $E = \log_2 N$.*

4.4.2 *The LinearLog Decoder.* In etching for the *LinearLog* decoder, the set M for the sth shell assumes the value of one shell material at each MW. Thus, the LinearLog uses $M_{LinearLog} = \sum_{r=1}^{n} |\sigma_r|$ MWs. Since $|\sigma_s| = \alpha_1$ when s is odd, and $|\sigma_s| = \alpha_2$ when s is even, $M_{LinearLog} = \lceil n/2 \rceil \alpha_1 + \lfloor n/2 \rfloor \alpha_2$. In some cases,

$M_{LinearLog}$ is smaller than $M_{FullyLog}$. For example, when $\alpha_1 = \alpha_2 = 3$ and $n = 2$, $M_{LinearLog} = 6$ whereas $M_{FullyLog} = 8$.

We now consider the number of operations that are needed to expose NWs to MWs. As with the FullyLog decoder, the LinearLog decoder under MW W_t^s, $1 \leq t \leq |\sigma_s|$, requires $(T - |\sigma_s|)$ operations on shells other than the sth, where $T = \sum_{r=1}^{n} |\sigma_r|$, and one operation on the sth shell for a total of $E_{LinearLog} = T^2 + T - \sum_{r=1}^{n} |\sigma_s|^2$ operations. This is quadratic in n, the number of shells.

As with previous decoders, the LinearLog decoder can be implemented within $\sum_{r=1}^{n} \sigma_r$ etching operations.

THEOREM 4.2. *The LinearLog decoder for n-shell NWs having α_1 (α_2) materials in odd-(even-)indexed shells and $N = \alpha_1^{\lceil n/2 \rceil} \alpha_2^{\lfloor n/2 \rfloor}$ NW types can be implemented with $M_{LinearLog} = \lceil n/2 \rceil \alpha_1 + \lfloor n/2 \rfloor \alpha_2$ MWs, and $E = \lceil n/2 \rceil \alpha_1 + \lfloor n/2 \rfloor \alpha_2$ etching operations. When $\alpha_1 = \alpha_2 = \alpha/2$ and n is even, $N = (\alpha/2)^n$, $M_{LinearLog} = n\alpha/2$, and $E = \log_2 N$.*

4.5 Code Discovery and Faults

Recall that codewords are randomly assigned to NWs in a nanoarray. As a result, a discovery process is required to determine which encodings are present. All radial decoders described previously allow for the use of the efficient code discovery algorithm given in Gojman et al. [2005].

The etching processes we have described may behave imperfectly. Shells which should remain may be removed, and shells that should be removed may remain. Either error can alter the subset of active MWs that control a NW. In the nanoarray, a binary tuple is assigned to each NW in which a 1 corresponds to MWs that turn NWs off, and a 0 corresponds to MWs that do not influence the NW. Etching errors can flip the bits in this tuple.

In order to protect against bit flips, the tuples associated with core-shell NWs must have a sufficiently high Hamming distance. This is easily accomplished with a minor modification to the single-shell logarithmic etching procedure. Instead of associating arbitrary binary strings with shell materials, we can use coding theory to assign binary strings with a sufficiently high Hamming distance. As explained in Kuekes et al. [2005], binary reflected NW codewords with Hamming distance of $2d + 2$ can tolerate up to d errors.

When this idea is extended to the FullyLog decoder, $2n$-bit binary tuples based on n-bit binary strings with a Hamming distance of d applied to each shell allows up to nd errors to be tolerated across n shells.

4.6 Examples of Radial Codes

We consider next the number of different NW types needed to ensure that N NWs can be addressed with high probability. In Figure 1, it is assumed that N NWs in each dimension of the crossbar are connected to one ohmic region at each end. As shown in DeHon et al. [2003], to ensure that all or most of the N NWs in each dimension are different with high probability requires that the number of differently encoded NWs, C_R, be enormous. Thus, we assume that the ohmic region at one end of the NWs in each dimension

Table III.

Min. # Codes C_R	# Shells n	Radial Distribution	Type of Decoder	# MWs λ_R	# Shell Types	NW Diameter
12	2	(4,3)	LinearLog	7	7	13nm
16	2	(4,4)	FullyLog	8	8	13nm
18	3	(3,3,2)	LinearLog	8	6	17nm
24	3	(4,3,2)	LinearLog	9	7	17nm
27	3	(3,3,3)	LinearLog	9	6	17nm

Parameters of some radial codes with $10 \leq C_R \leq 30$ are shown. Here n is the number of shells and λ_R is the number of MWs. The distribution of shell types is shown in the third column. The type of decoder is shown in the fourth column. The minimal number of types of shell materials that suffice to encode NWs, is shown in the sixth column (see Section 4.1). The last column contains the diameter of NWs (and their pitch when abutted one against the other) under the assumption that cores have a diameter of 5nm, and each shell adds 4nm to the diameter.

is subdivided into m ohmic regions each containing w NWs (also shown in DeHon et al. [2003]).

The diameter and pitch of radially encoded NWs grows with the number of shells. Thus, it is important to keep the number of shells to a reasonable minimum. In turn, the number of shells is related to C_R, the size of the code space. If the number of differentially etchable shell materials is small, the number of shells must be large to meet a minimum requirement on C_R.

We now ask how large C_R must be to ensure that with probability 0.99 or larger at least half of the NWs in each dimension of an array has a unique address specified by its ohmic region and NW type within that region. Analysis [Gojman et al. 2005] and empirical evidence [Rachlin and Savage 2005] indicate that it suffices to have $10 \leq C_R \leq 30$. In Table III examples of radial codes that meet this requirement are shown.

5. HYBRID NANOWIRE CODES AND DECODERS

A NW has *a hybrid encoding* if its core has an axial encoding and its shells have a radial encoding. To cope with random axial displacement of NWs, the axial encoding is repeated along the length of NWs.

This section describes how an axial and radial decoder can be efficiently combined to form a hybrid decoder. Our *BRC hybrid decoder* uses two sets of MWs. The first set functions as an axial decoder, the other as a radial decoder. The total number of MWs required by the hybrid decoder is proportional to the sum of the number of MWs required by the axial and radial decoders when used separately.

The BRC hybrid decoder is designed to work exclusively with binary reflected codes. Recall that in a λ_A-BRC, the doping pattern is a repeated sequence of λ_A heavily- or lightly-doped regions. The repeated sequence is such that the first $\lambda_A/2$ regions are the compliment of the second $\lambda_A/2$ regions. If two regions in the sequence lie $\lambda_A/2 - 1$ regions apart, exactly one will be lightly doped. The radial decoding portion of the BRC hybrid decoder relies on this.

Consider a radial code controlled by a radial decoder using λ_R MWs. Suppose the radial code is used in conjunction with a λ_A-BRC to generate hybrid NWs.

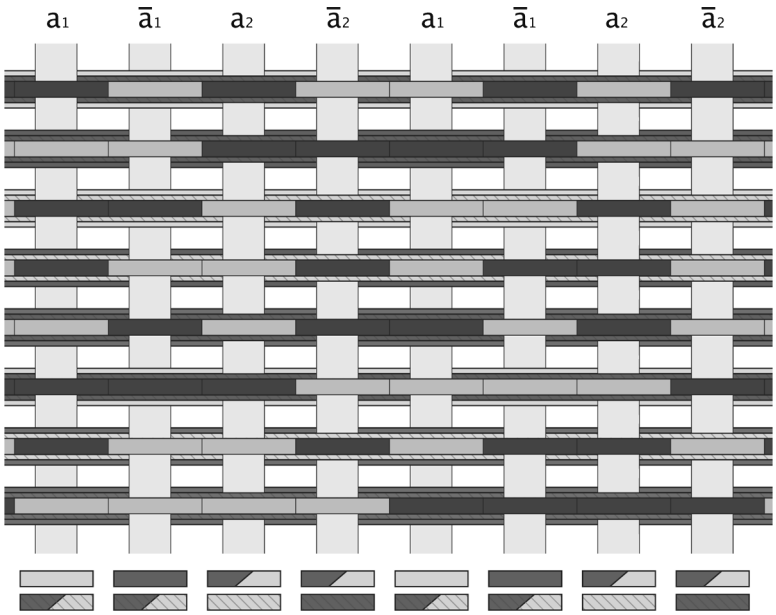

Fig. 12. The second portion of a BRC hybrid decoder devoted to radial decoding. An arbitrary radial decoder in which MWs are labeled $a_1, \bar{a}_1, a_2$, and $\bar{a}_2$ is implemented on the two halves of the BRC axial code.

A BRC hybrid decoder is constructed as follows (see Figure 12.)

(1) Use λ_A consecutive MWs for axial decoding. Under these MWs, all shells are removed from all NWs. The MWs then function as a standard axial decoder. The λ_A MWs are used to select hybrid NWs with a given axial codeword. In other words, the MWs can make nonconducting all NWs that do not have a particular binary reflected codeword.
(2) Use $2\lambda_R$ MWs for radial decoding. Each NW contains a repeated binary reflected codeword. If a pair of MWs are $\lambda_A/2 - 1$ regions apart, exactly one will lie over a lightly-doped region. Use λ_R such pairs (in $\lceil 2\lambda_R/\lambda_A \rceil$ repetitions of the axial code) to produce two identical radial decoders. Apply the same etching operations to both MWs in a pair.
(3) One MW in each pair is adjacent to an exposed lightly-doped region. If the two radial decoders are used simultaneously, they successfully simulate a standard radial decoder. The $2\lambda_R$ MWs can thus select MWs with a given radial codeword. When a radial and axial codeword are selected simultaneously, only NWs with a particular hybrid codeword will remain conducting.

As mentioned previously, $\lceil 2\lambda_R/\lambda_A \rceil$ repetitions of an axial λ_A-BRC codeword suffice to implement a radial code. Thus, $\lambda_A + \lambda_A \lceil 2\lambda_R/\lambda_A \rceil$ MWs are sufficient to realize a hybrid decoder. Since $\lceil 2a/b \rceil \leq (2a/b) + 1$, it follows that the BRC-hybrid decoder uses at most $2(\lambda_A + \lambda_R)$ MWs. This is within a factor of two of the information theoretic minimum.

Table IV.

C_A	λ_A	Radial Distr.	C_R	n	# Shell Types	λ_R	# MWs λ_2	# Codes C_AC_R	NW Diameter
16	8	(2)	2	1	2	2	16	32	9nm
8	6	(3)	3	1	3	3	12	24	9nm
		(2)	2	1	2	2	12	16	9nm
4	4	(3,2)	6	2	5	5	16	24	13nm
		(2,2)	4	2	4	4	12	16	13nm
		(3)	3	1	3	3	12	12	9nm
2	2	(4,3)	12	2	7	7	16	24	13nm
		(4,2)	8	2	6	6	14	16	13nm
		(3,2)	6	2	5	5	12	12	13nm

Parameters of hybrid codes that produce 12 to 32 different NW types (C_AC_R) are shown. C_A is the number of BRC axial code types and λ_A the number of MWs it uses. The shell distribution is shown for each radial code along with its number of NW types, C_R, the number of shells, n, shell types T_R, and MWs, λ_R, that it uses with a logarithmic decoder. The total number of MWs used with the λ_A-BRC decoder is $\lambda_2 = \lambda_A\left(1 + \lceil 2\lambda_R/\lambda_A \rceil\right)$. The number of codewords in the code and the diameter of NWs are also shown.

5.1 Examples of Hybrid Codes

Several combinations of axial and radial encodings that produce between 12 and 32 NW types while keeping the number of shells and shell types small are shown in Table IV. The reason for considering this small number of NW types is explained at the end of Section 4.6.

5.2 Sensitivity of the BRC Hybrid Decoder to Displacements

If a fluidic process is used to assemble NWs with hybrid codes into parallel arrays, NWs will be displaced axially or lengthwise during assembly. To cope with this problem, as mentioned above, the axial code is repeated as was done for purely axially-coded NWs [DeHon et al. 2003].

If a BRC is shifted axially by the pitch of MWs, the doping pattern under MWs corresponds to that of another BRC [Gojman et al. 2004]. In other words, the set of BRCs is closed under displacements by multiples of a MW pitch. The same is true of hybrid codes.

LEMMA 5.1. *The set of hybrid codes when decoded using a BRC decoder is closed under axial displacements of NWs by the pitch of a MW.*

The analysis of the sensitivity of axial decoders to displacements by less than a MW pitch given here is the same as that sketched in Section 3 for axial codes except that the MW pitch is larger for a given number of NW encodings. Since the probability of loss of control by MWs decreases with this parameter, hybrid codes are less sensitive to axial displacements than would be a comparable axial code. We repeat here the analysis given in DeHon et al. [2003].

We assume that the length of a doped region is the length of the region between two MWs that cannot be controlled by either MW plus $2W_{overlap}$. Under this assumption, NW doped regions are normal, the definition of which is given in Section 3.2.

As mentioned in Section 3, the probability, $P_{control}$, that a NW is controlled by a MW is $P_{control} = (1 - 2W_{overlap}/W_{pitch})$.

6. COMPARISON OF NW ENCODING STRATEGIES

Radial, axial, and hybrid codes can be compared along various dimensions. Some of these are a) the total area of a chip including the area of the crossbar as well as the area of a memory to translate external to internal addresses, b) the area of the crossbar alone under the assumption that the area of the translation memory can be ignored, c) the difficulty of manufacturing NWs with a given type of encoding, and d) the difficulty of reliably assembling a memory. In light of uncertainties that exist with respect to these issues, we compare NW encodings on the basis of the second measure, namely, the area of the crossbar alone. This is done by comparing the effective pitch of NWs.

The effective pitch of NWs is their actual pitch increased by a factor that takes into account the loss of NWs due to duplication and/or misalignment. Misalignment is possible with both axial and hybrid codes.

We only compare the effective pitch of radial and axial codes because the hybrid codes are generally inferior to axial codes. Hybrid codes incur almost the same misalignment penalty as axial codes but have increased pitch. Thus, in this comparison, they are inferior to both radial and axial codes. Hybrid codes may still be useful since a removable shell will always be added to axially coded NWs to ensure that they remain separated during fluidic assembly. Perhaps this shell can be used to advantage.

To compare the effective pitch of NWs with axial and radial coding, we compute their raw diameter (see Table V), which is also their raw pitch, under the assumption that each axially coded NW has one shell as indicated previously, and that the assembly process abuts one NW with another. Radially coded NWs will generally have more than one shell. We then ask how many individually addressable NWs remain after ignoring duplicates and misaligned NWs when the NWs are organized into 100 contact groups (also known as ohmic regions) each of which contains 10 NWs. We have done a Monte Carlo simulation with four code sizes, namely, $C = 12$, 20, 24, and 80, to determine how many addressable NWs are accessible among the 1,000 NWs in the 100 contact groups. These code sizes are those that are realizable with either two or three shells containing either four or five different types of material on radially coded NWs when linearly decoded. (See Section 4.2) The simulations were done under the assumption that either no misalignment occurs or it occurs with probabilities 0.09, 0.20, and 0.40. The three values were chosen because they are three of the values of $1 - P_{control}$ that arise when $W_{pitch} = 105$ or 50nm and $W_{overlap} = 5$ or 10nm. The fourth case in which $P_{control} = .81$ is ignored because it doesn't provide new information.

As these data indicate, radial codes have a slight disadvantage with regard to effective NW pitch. This translates directly into the crossbar storage density since one bit is stored per effective pitch squared of area. Using memory bank area models from [DeHon et al. 2005] and assuming $P_{gbit} = 0.95$, $P_c = 0.95$, and $P_j = 0.9999$, the radial and axial codes that achieve an effective pitch of

Table V.

$P_{control}$	Coding Style	Code Size	No. Shells	Actual Diameter	NW Yield With Prob. .99	Effective Pitch
1.00	radial	12	2	13	672	19.3
		20	2	13	778	16.7
		24	3	17	807	21.1
		80	3	17	928	18.3
0.91	axial	12	1	9	628	14.3
		20	1	9	716	12.6
		24	1	9	741	12.1
		80	1	9	839	10.7
0.80	axial	12	1	9	575	15.7
		20	1	9	647	13.9
		24	1	9	667	13.5
		80	1	9	743	12.1
0.60	axial	12	1	9	451	20.0
		20	1	9	492	18.3
		24	1	9	502	17.9
		80	1	9	544	16.5

Comparison of axial and radial codes in terms of their effective NW pitch for four different values of C, the code size, for radial codes having $s = 2$, or 3 shells and $\alpha = 4$, or 5 different types of shell material. When decoded with a linear decoder, the radial codes have $C_R = 12$, 20, 24 or 80 code types. The number of NWs with different addresses that occur with probability 0.99 among 1,000 NWs connected to 100 contact groups is shown. The value of $P_{control} = (1 - 2W_{overlap}/W_{pitch})$ is computed when $W_{pitch} = 105$ or 50nm and $W_{overlap} = 5$ or 10nm, that is, when $P_{control} = 0.91$, 0.81, 0.80 or 0.60. Results when $P_{control} = 0.81$ are not shown.

roughly 16.5nm (such as 13nm radial codes with 20 codewords or 9nm axial codes with 80 codewords and $P_{ctrl} = 0.6$) yield net memory densities of roughly 2000nm^2/bit and 1000nm^2/bit with W_{pitch}=105nm and 50nm, respectively.

Additionally, axial encodings appear to be more difficult to prepare because of variation in the length of doped regions and the difficulty of producing abrupt transitions between differently doped regions. Also, axially doped NWs can be in an ambiguous conducting state. Finally, because axially encoded NWs will need at least one shell to keep them separate under fluidic assembly, it makes sense to consider radially encoded NWs, especially if the number of differentially etchable shell materials can be increased beyond five.

7. CONCLUSIONS

Radially encoded NWs provide a promising method to differentiate NWs for their assembly into crossbars. To motivate the need for a new method of differentiating NWs, we described the problems that arise with axially encoded NWs as a result of their random misalignment during the fluidic assembly of crossbars.

To confirm that radially encoded NWs are feasible, we gave four examples of materials that are differentially etchable as well as their etchants. We then describe four methods for decoding radial encoded NWs, the linear method, the logarithmic method for shell, and the LinearLog and FullyLog on multiple shells. We demonstrated that a large number of NW types, C_R, can be created

from a small number of shell materials, especially when decoded by the linear method. However, the linear method requires many more MWs to address NWs than the logarithmic method. A disadvantage of the logarithmic method is that it requires more shell materials to achieve the same value of C_R. Because NW diameter and pitch generally grow with C_R, it is important that the latter be small.

For each decoder, we considered algorithms for etching NWs to expose their insulated, lightly-doped cores to the fields applied by MWs. For each of them, we demonstrated that the number of etching steps can be reduced to the sum of the number of material types in each shell when etching is done in parallel.

Because it is natural to consider NW encodings that are a hybrid of axial and radial encodings, we also considered such encodings. We find that they are generally inferior to the other two.

Finally, we performed a representative comparison of the effective NW pitch for axial and radial encodings. This comparison suggests that axial codes may be slightly better than radial codes as measured by their effective pitch and resulting crossbar storage density. However, any advantage the axial NWs may have in effective pitch is diminished by the common amount of circuitry that both types of NW encoding require to control the crossbar. Given that axial codes require a more delicate manufacturing process and that shells are useful for other purposes, it appears that radial codes will be a key type of NW encoding in the realization of nanoarray crossbars.

REFERENCES

AUSTIN, M. D., GE, H., WU, W., LI, M., YU, Z., WASSERMAN, D., LYON, S. A., AND CHOU, S. Y. 2004. Fabrication of 5nm linewidth and 14nm pitch features by nanoimprint lithography. *Appl. Physics Lett. 84,* 26 (June), 5299–5301.

BJÖRK, M. T., OHLSSON, B. J., SASS, T., PERSSON, A. I., THELANDER, C., MAGNUSSON, M. H., DEPPERT, K., WALLENBERG, L. R., AND SAMUELSON, L. 2002. One-dimensional steeplechase for electrons realized. *Nano Lett. 2,* 2, 87–89.

CHEN, Y., JUNG, G.-Y., OHLBERG, D. A. A., LI, X., STEWART, D. R., JEPPESON, J. O., NIELSON, K. A., STODDART, J. F., AND WILLIAMS, R. S. 2003. Nanoscale molecular-switch crossbar circuits. *Nanotechn. 14*, 462–468.

COLLIER, C. P., MATTERSTEIG, G., WONG, E. W., LUO, Y., BEVERLY, K., SAMPAIO, J., RAYMO, F., STODDART, J. F., AND HEATH, J. R. 2000. A [2]catenate-based solid state electronically reconfigurable switch. *Science 290*, 1172–1175.

COLLIER, C. P., WONG, E. W., BELOHRADSKÝ, M., RAYMO, F. M., STODDART, J. F., KUEKES, P. J., WILLIAMS, R. S., AND HEATH, J. R. 1999. Electronically configurable molecular-based logic gates. *Science 285*, 391–394.

CUI, Y., LAUHON, L., GUDIKSEN, M., WANG, J., AND LIEBER, C. M. 2001. Diameter-controlled synthesis of single crystal silicon nanowires. *Appl. Physics Lett. 78,* 15, 2214–2216.

DEHON, A. 2003. Array-based architecture for FET-based, nanoscale electronics. *IEEE Trans. Nanotechn. 2,* 1 (March) 23–32.

DEHON, A., GOLDSTEIN, S. C., KUEKES, P., AND LINCOLN, P. 2005. Nonphotolithographic nanoscale memory density prospects. *IEEE Trans. Nanotechn. 4,* 2, 215–228.

DEHON, A., LINCOLN, P., AND SAVAGE, J. E. 2003. Stochastic assembly of sublithographic nanoscale interfaces. *IEEE Trans. Nanotechn. 2,* 3, 165–174.

DEHON, A. AND WILSON, M. J. 2004. Nanowire-based sublithographic programmable logic arrays. In *Proceedings of the International Symposium on Field-Programmable Gate Arrays*. 22–24.

DEKKER, C. 1999. Carbon nanotubes as molecular quantum wires. *Physics Today*, 22–28.

DIER, O., LIN, C., GRAU, M., AND AMANN, M.-C. 2004. Selective and non-selective wet-chemical etchants for GaSb-based materials. *Semiconduc. Science Techn. 19,* 11, 1250–1253.

DUAN, X., HUANG, Y., AND LIEBER, C. M. 2002. Nonvolatile memory and programmable logic from molecule-gated nanowires. *Nano Lett. 2,* 5, 487–490.

GLASSER, L. A. AND DOPPERPUHL, D. W. 1985. *The Design and Analysis of VLSI Circuits*. Addison-Wesley, Reading, MA.

GOJMAN, B., RACHLIN, E., AND SAVAGE, J. E. 2005. Evaluation of design strategies for stochastically assembled nanoarray memories. *J. Emerg. Techn. Comput. Syst. 1,* 2, 73–108.

GOJMAN, B., RACHLIN, E., AND SAVAGE, J. E. 2004. Decoding of stochastically assembled nanoarrays. In *Proceedings of the International Symposium on VLSI*. Lafayette, LA.

GUDIKSEN, M. S., LAUHON, L. J., WANG, J., SMITH, D. C., AND LIEBER, C. M. 2002. Growth of nanowire superlattice structures for nanoscale photonics and electronics. *Nature 415*, 617–620.

HEATH, J. R. AND RATNER, M. A. 2003. Molecular electronics. *Physics Today 56,* 5, 43–49.

HUANG, Y., DUAN, X., WEI, Q., AND LIEBER, C. M. 2001. Directed assembly of one-dimensional nanostructures into functional networks. *Science 291*, 630–633.

JOHNSTON-HALPERIN, E., BECKMAN, R., LUO, Y., MELOSH, N., GREEN, J., AND HEATH, J. 2004. Fabrication of conducting silicon nanowire arrays. *J. Appl. Physics Lette. 96,* 10, 5921–5923.

KIM, F., KWAN, S., AKANA, J., AND YANG, P. 2001. Langmuir-Blodgett nanorod assembly. *J. Ameri. Chem. Soc. 123,* 18, 4360–4361.

KUEKES, P. J., ROBINETT, W., SEROUSSI, G., AND WILLIAMS, R. S. 2005. Defect-tolerant interconnect to nanoelectronic circuits. *Nanotechn. 16*, 869–882.

KUEKES, P. J., WILLIAMS, R. S., AND HEATH, J. R. 2000. Molecular wire crossbar memory, US Patent Number 6,128,214.

LAUHON, L. J., GUDIKSEN, M. S., WANG, D., AND LIEBER, C. M. 2002. Epitaxial core-shell and core-multishell nanowire heterostructures. *Nature 420*, 57–61.

MELOSH, N. A., BOUKAI, A., DIANA, F., GERARDOT, B., BADOLATO, A., PETROFF, P. M., AND HEATH, J. R. 2003. Ultrahigh-density nanowire lattices and circuits. *Science 300*, 112–115.

MORALES, A. M. AND LIEBER, C. M. 1998. A laser ablation method for synthesis of crystalline semiconductor nanowires. *Science 279*, 208–211.

QUIAN, F., GRADECAK, S., LI, Y., WEN, C.-Y., AND LIEBER, C. M. 2005. Core/multishell nanowire heterostructures as multicolor, high-efficiency light-emitting diodes. *Nano Lett. 5*, 2287–2291.

RACHLIN, E. AND SAVAGE, J. E. 2005. Small codespace addressing strategies for nanoarrays. Nano Note 3, Computer Science Department, Brown University.

RACHLIN, E., SAVAGE, J. E., AND GOJMAN, B. 2005. Analysis of a mask-based nanowire decoder. In *Proceedings of the International Symposium on VLSI*. Tampa, FL.

RUECKES, T., KIM, K., JOSELEVICH, E., TSENG, G. Y., CHEUNG, C.-L., AND LIEBER, C. M. 2000. Carbon nanotube-based nonvolatile random access memory for molecular computing. *Science 289*, 94–97.

WHANG, D., JIN, S., AND LIEBER, C. M. 2003. Nanolithography using hierarchically assembled nanowire masks. *Nano Lett. 3,* 7, 951–954.

WILLIAMS, R. S. AND KUEKES, P. J. 2001. Demultiplexer for a molecular wire crossbar network, US Patent Number 6,256,767.

WU, Y., FAN, R., AND YANG, P. 2002. Block-by-block growth of single-crystal Si/SiGe superlattice nanowires. *Nano Lett. 2,* 2, 83–86.

ZHONG, Z., WANG, D., CUI, Y., BOCKRATH, M. W., AND LIEBER, C. M. 2003. Nanowire crossbar arrays as address decoders for integrated nanosystems. *Science 302*, 1377–1379.

Received October 2005; revised March 2006; accepted April 2006