Nonphotolithographic Nanoscale Memory Density Prospects

André DeHon, *Member, IEEE*, Seth Copen Goldstein, *Member, IEEE*, Philip J. Kuekes, *Member, IEEE*, and Patrick Lincoln

***Abstract*—Technologies are now emerging to construct molecular-scale electronic wires and switches using bottom-up self-assembly. This opens the possibility of constructing nanoscale circuits and memories where active devices are just a few nanometers square and wire pitches may be on the order of ten nanometers. The features can be defined at this scale without using photolithography. The available assembly techniques have relatively high defect rates compared to conventional lithographic integrated circuits and can only produce very regular structures. Nonetheless, with proper memory organization, it is reasonable to expect these technologies to provide memory densities in excess of 10^{11} b/cm^2 with modest active power requirements under 0.6 W/Tb/s for random read operations.**

***Index Terms*—Defect tolerance, electronic nanotechnology, memory density, memory organization, molecular electronics.**

I. Introduction

SCIENTISTS are now reliably fabricating nanowires (NWs) which are just a few atoms in diameter and carbon nanotubes (NTs) which are just over 1 nm in diameter. These wires are created and their feature size controlled without relying on lithographic processing. Devices have been built from individual molecules which exhibit diode rectification, negative differential resistance, field-effect gating, or nonvolatile memory storage behavior; these, too, are synthesized without lithographic processing. Using self-assembly techniques, these building blocks have been organized into sets of crossed wires with devices at the crosspoint junctions.

With switchable devices at the crosspoints, such arrays form memory banks. NT and NW feature sizes are smaller than the lithographic road map envisions for 2016 [1], and these techniques may provide a fabrication alternative which avoids the challenges and costs of lithography at this scale.

To exploit these devices, they must be interfaced with lithographic-scale processing and tolerate the inevitable defects which occur during the self-assembly processes. Both the interfacing and the defects will reduce the net density delivered by these devices.

In this paper, we explore how these nanoscale memories can be organized and integrated (Section III) into a conventional, lithographic-scale scaffolding. We analyze the basic design to calculate the raw area (Section IV), delay (Section V), and energy (Section VI) characteristics of these memories. We use a simple defect model to understand the net effect of various defect rates and summarize the net densities which this suggests and the memory organizations necessary to achieve them (Section VII).

II. Technology

A. Devices

Chen *et al.* [2] demonstrate a nanoscale Pt-rotaxane-Ti/Pt sandwich which exhibits hysteresis and nonvolatile state storage showing an order of magnitude resistance difference between "on" and "off" states for several write cycles. With 1600-nm^2 junctions, the "on" resistance was roughly 500 kΩ, and the "off" resistance was 9 MΩ. After an initial "burn-in" step, the state of these devices can be switched at ± 2 V and read at ± 0.2 V. The basic hysteretic molecular memory effect is not unique to the rotaxane, and the junction resistance is continuously tunable [3]. The exact nature of the physical phenomena involved is the subject of active investigation.

Appendix I highlights two additional molecular-scale, nonvolatile crosspoint technologies and others are being developed. So far, they all seem to share the following characteristics:

1) programmable resistance—significant resistance change between "on" and "off" states;
2) rectification;
3) voltage switching—a suitable application of voltage can turn them "on" or "off" (sometimes in the presence of other environment factors, like temperature control).

B. Wires

Today, chemists can synthesize carbon NTs which are nanometers in diameter and micrometers long [4]. Ultimately, these carbon NTs can be a single nanometer wide. To date, we cannot control the detailed electrical properties (conducting versus semiconducting) for these NTs during growth, but the conduction of even the worst conductors is often adequate for many uses; further, techniques for separating them after growth are now being developed (e.g., [5]).

At the same time, chemists are developing technologies to grow silicon and germanium NWs [6], [7] which are also only

Manuscript received December 1, 2003; revised June 28, 2004. This work was supported by the Defense Advanced Research Projects Agency Moletronics Program under Grant ONR N00014-01-0651, Grant MDA972-01-03-0005, and Grant ONR N00014-01-0659.

A. DeHon is with the Computer Science Department, California Institute of Technology, Pasadena, CA 91125 USA (e-mail: andre@acm.org).

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

P. J. Kuekes is with the Quantum Science Research Group, Hewlett-Packard Laboratories, Palo Alto, CA 94304 USA.

P. Lincoln is with the Computer Science Laboratory, SRI International, Menlo Park, CA 94025-3493 USA (e-mail: lincoln@csl.sri.com).

Digital Object Identifier 10.1109/TNANO.2004.837849

nanometers in width (e.g., wires as small as 3 nm in diameter have been reported). These NWs can be hundreds of micrometers long [8]. The electrical properties of these NWs can be controlled with dopants, yielding semiconducting wires [9]. NWs can be assembled along with NTs when their respective properties complement each other.

C. Field-Effect Gating

Conduction through doped NWs can be controlled via an electrical field like field-effect transistors (FETs) [10]. "Off" resistances can be over 10 GΩ and "on" resistances under 0.1 MΩ; OFF–ON resistance ratios are at least 10^4 [11]. An NW field-effect gating has sufficient gain to build restoring gates [12]. The threshold voltage for the NW can be controlled by material properties (e.g., doping or composition) and geometry factors.

D. Assembly

NTs and discrete NWs can be aligned and assembled into parallel rows of conductors which are several nanometers apart. These can be layered into arrays of orthogonal wires [13], [14]. Alternate techniques grow NWs on a lattice mismatched substrate so they naturally grow in a single dimension yielding straight parallel NWs which are a few nanometers across and spaced several nanometers apart [15].

Imprint lithography [2], [16], [17] can also be used to produce aligned parallel NWs. Masks are produced by a variety of techniques, including direct E-beam write and timed etches to reduce feature sizes. Chou *et al.* demonstrate 50-nm-pitch features using imprint lithography [18]. Chen *et al.* demonstrate crossbar arrays built at 125-nm pitch using imprint lithography [19].

Whang *et al.* demonstrates that flow-aligned NWs can be used as masks for patterning tight-pitch parallel NW masks, demonstrating features with an average spacing of 90 nm [14]. NW pitch is a function of NW diameter and composition—features which are controlled by catalyst sizes and growth times, not by lithography or E-beam. Techniques such as this may ultimately allow the construction of NW masks with pitches below 20 nm.

In yet another technique, a vertically grown superlattice structure is used to create a pattern for etching; timed vertical growth of different materials defines features down to a few nanometers without lithography. The resulting structure can be cut orthogonally, differentially etched to expose one of the materials, and used to transfer an etch mask pattern. Parallel wires with pitches as small as 16 nm have been demonstrated [20].

Using Langmuir–Blodgett techniques, switchable molecules (e.g., [21]) can be formed into a single monolayer and transfered onto a set of parallel NWs or NTs [22]. An orthogonal set of NT/NWs can then be transfered on top, creating a conductor-device-conductor sandwich for the memory crossbar. Chen *et al.* demonstrate an 8×8 molecular crossbar at 125-nm pitch using this approach [19].

III. Memory Organization

Large blocks of memory are conventionally organized as sets of smaller submemories, which are called banks. The reason for breaking a large memory into smaller banks is to trade off overall memory density for access speed and reliability. Excessively small bank sizes will incur a large area overhead for memory drivers and receivers, while excessively large banks will suffer from large shared row and column capacitance and larger defect rates.

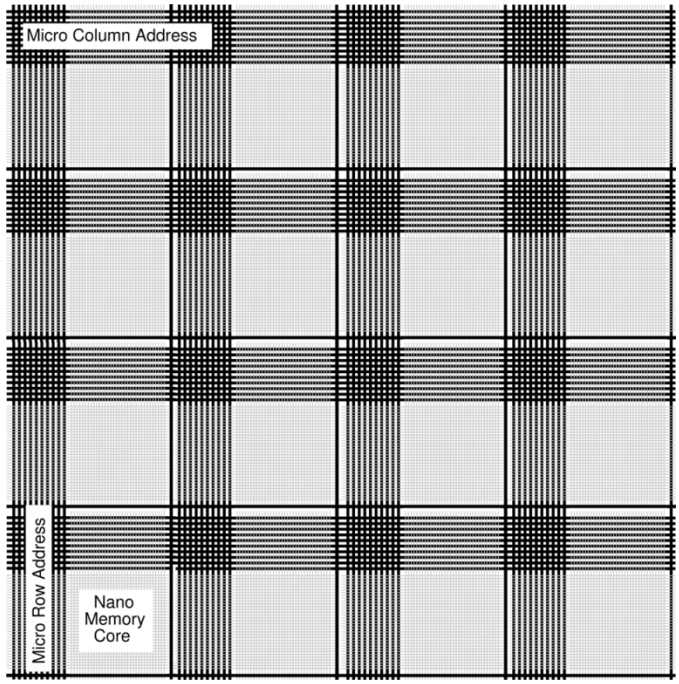


Fig. 1. Large memory assembled from nanoscale memory array banks.

The organization of nanoscale memory is no different, except that the overhead per bank is larger due to the scale difference between the size of a memory bit (a single wire crossing) and the support logic (a mix of nanoscale wires and microscale wires). In this section, we describe the basic mechanisms and parameters for a bank of memory and show how the banks would be integrated into a complete memory system.

A. Banks

Banks are tiled together to build a large-scale memory (see Fig. 1). Each bank is composed of a set a crossed nanoscale wires supported by a set of interface microscale wires (see Fig. 2). For the purpose of this description, we assume that logic, multiplexing, and decoding for interbank connectivity are all performed using conventional microscale circuitry and, consequently, merit little discussion here.

B. Microscale Interfacing

The easiest way to integrate nanoscale building blocks with a microscale infrastructure is to provide a bridge between the microscale wires we can build reliably with conventional lithography and the NWs or NTs used in our nanoscale memory banks. In this manner, we use a conventional lithographic integrated circuit as a substrate for assembling these nanoscale memory blocks just as we use printed-circuit boards or multichip modules as a packaging hierarchy for integrated circuits. This is particularly important since the NWs must typically be short (e.g., tens of micrometers) and are error-prone. The lithographic scaffolding allows us to connect them together into larger ensembles

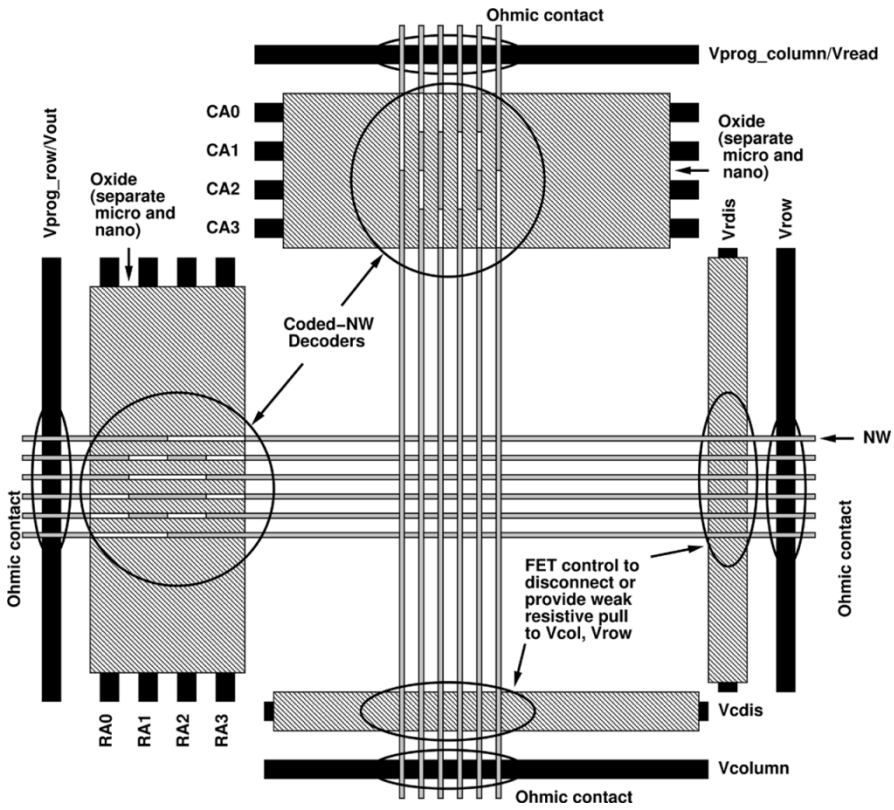


Fig. 2. Nanoscale memory array bank; for clarity, only a small (6×6) NW crossbar is shown. A typical array might have 100–3000 nanoscale wires on each side and be addressed by 35–60 microscale wires.

and gives us a set of fault-free wires from which to bootstrap testing and configuration.

We use a decoder to interface between the nanoscale and microscale wires. The decoder allows a large number of nanoscale wires to be addressed by a small number of microscale wires, allowing us to exploit the tight pitch promised by nanoscale wires. The key manufacturing challenge here is to create differentiating features in the decoder interface which allow individual nanoscale lines to be addressed in the memory core.

Kuekes and Williams describe a scheme for bridging microscale and nanoscale wires with a decoder based on randomly deposited gold nanoparticles [23]. If the gold particle density is properly controlled and an appropriate address code space is used, then, with high probability, most of the nanoscale wires can be uniquely addressed. They show that $5 \log_2(N)$ address wires should be sufficient to address a core array with N wires.

DeHon *et al.* [24] harness a more complex fabrication technique to address N lines using no more than $\lceil 2.2 \log_2(N) \rceil + 11$ address wires. Their scheme uses a fabrication technique which can control the doping profile or material composition along the axial dimension of an NW [25]–[27]. By controlling the doping profile, one can effectively control the field-effect conduction threshold voltage along the length of the wire (see Section II-C), making some regions gateable and others oblivious to the normal operating voltage of the crossed wire.

Both schemes work by placing a number of field-effect controllable regions on each of the NWs (*NB*: white bands at the top and left of the NWs in Fig. 2). The NW will conduct only when all of the controllable regions see the appropriate field from the microscale address wires. Unique code population from a suitable code space guarantees that at most one NW has its control regions satisfied for conduction. Decoders built in this way serve both as demultiplexers that place a target voltage on a single selected NW and as multiplexers that allow the voltage from a single selected NW to be coupled to a common output.

The random-particle scheme requires no features on the NWs. Consequently, it is self-aligning to the extent that a gold nanoparticle creates a connection where it exists. The coded-NW-design engineers features into the NWs themselves; however, we cannot, currently, control the alignment of the NWs relative to the microscale circuitry. Nonetheless, the coded-NW scheme can be designed to tolerate misalignments and take only a small yield loss for particularly unfortunate alignments [24].

C. Basic Operation

The minimal circuitry shown in Fig. 2 allows us to program each of the diode junctions, one at a time, and read back memory bits one at a time. Fig. 3 shows a rough circuit equivalent for the memory bank. Note that the NW array shown in Figs. 2 and 3 includes the address decoder and the output demultiplexer. All control lines to each memory bank are global except for the write enable (not shown) and the input and output data bit. For each global signal, a local buffer isolates the bank capacitance from the global signal runs. We assume separate metal layers are used for global runs and local runs, and thus they can be stacked on top of each other. The only logic we need to perform locally is the selection of the appropriate row and column supply voltages

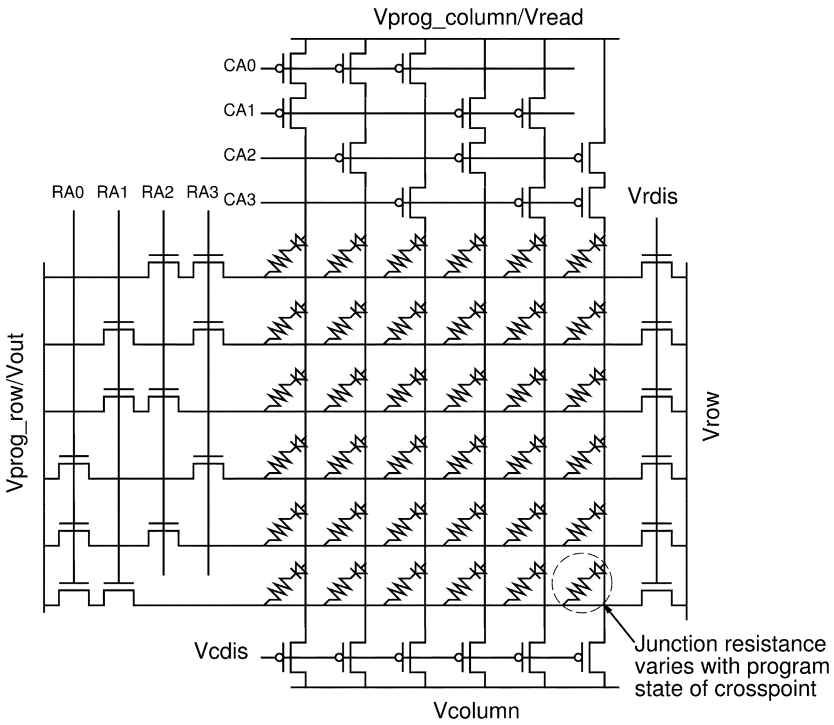


Fig. 3. Rough circuit equivalent for nanoscale memory.

for programming or reading which can be done with four logic gates and five transistor switches.

To program a junction, we set the row and column supplies ($V_{\mathrm{prog\_row}}$ and $V_{\mathrm{prog\_column}}$) to the appropriate programming voltages (e.g., $+1$ and -1 V for the aforementioned molecular switches from [2]), and we drive the intended row and column addresses into the row and column lines (RAi and CAi). To ensure that only the selected junction is programmed, V_{row} and V_{column} pull the nonselected row and column NWs (via a weak resistance) to a nominal voltage level (V_{nominal}). V_{rdis} and V_{cdis} are used to control the weak resistance. Note that V_{nominal} is a voltage that is midway between the high and low programming voltages. If the high and low voltages used for programming are symmetric about ground, as is the case for the example above, V_{nominal} can be ground; however, in general, it may not be ground. The net result is that we place the intended voltage difference across a single junction point, which allows us to set or reset the device (e.g., molecule or electrostatic switch) at that junction.

We assume each nonvolatile crosspoint is rectifying, i.e., the contact is a diode from the column to the row. To read a crosspoint's state, we drive the intended row and column addresses and drive a high read voltage onto V_{read} (e.g., 0.2–1 V for molecular switches from [2]). This way, only the intended column is driven, and only the intended row is coupled onto the output line V_{out} (which plays a dual role as $V_{\mathrm{prog\_row}}$). Naturally, V_{out}, is not driven, but instead is read. The current flow from V_{read} to V_{out} then tells us whether the crosspoint is programmed into the "on" or "off" state. We pull V_{column} to ground during this operation so no current flows along the nondriven lines.

Unlike a DRAM, we are not pulling charge off of the memory crosspoint. Consequently, if we wait long enough, we should be able to charge a capacitive output up to a high voltage level—ideally all the way up to the input read voltage. To contain speed and power, we will typically arrange to expose minimal microscale capacitance to the nanoscale circuitry and sense a low-voltage swing on this capacitor using traditional sense-amp techniques.

The above description demonstrates that it is possible to implement a nanoscale memory with full read and write capabilities by reading and writing a single bit at a time from the lithographic-scale CMOS circuitry. The logic to drive and control the microscale control lines and to construct the sense-amps are all implemented in microscale circuitry. As we show in Appendix IV-A, reasonable bank sizes are large enough so that the per-bank CMOS circuitry can be placed underneath the microscale control wire runs.

This version of the memory, while completely operational, has two problems which would make it slower than necessary.

1) The diode memory points can couple a column read line to every row line, forcing us to charge all rows in order to read a single bit. This forces the read time to scale as the product of the number of rows and columns rather than the sum.
2) Using stochastically populated address wires forces us to indirect through a separate memory to find out which addresses are actually live in the array. This can be done via multiple reads on the array, but the number of sequential reads needed to address a single bit out of a full device could be prohibitively large.

D. Precharge Read

It is possible to avoid the worst-case coupling capacitance for read operations, making read time scale as $O(N_{\mathrm{row}} + N_{\mathrm{column}})$

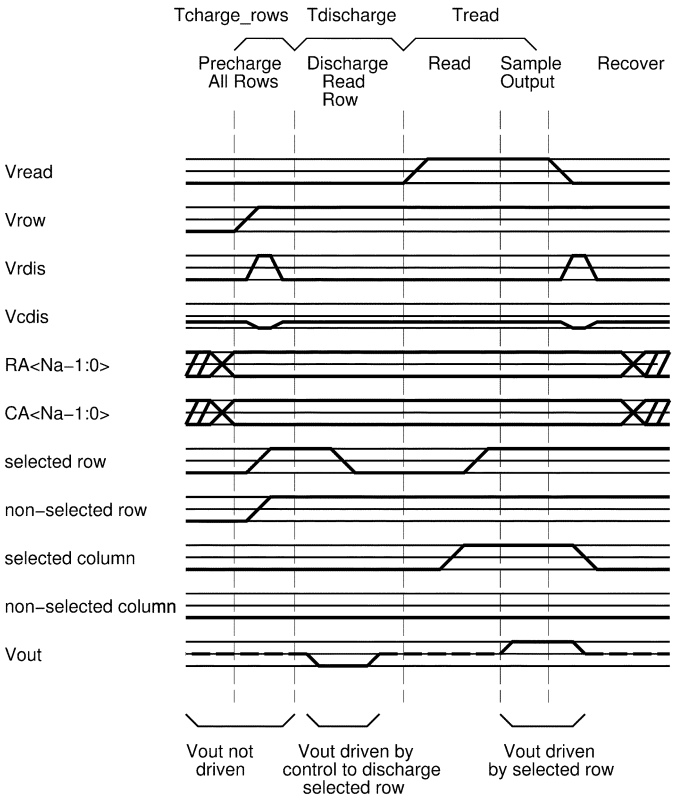


Fig. 4. Precharge read timing diagram. Diagram shows a read of an "on" junction.

rather than $O(N_{\text{row}} \times N_{\text{column}})$. We first precharge all of the row lines to the high read voltage. These can be driven in parallel so that it takes no more time than charging a single row line. Then we discharge the single row line we wish to read. Now, when we perform the read operation as before, the row lines associated with bits which we do not wish to read are already charged high and will not need to be charged while driving the intended row line (see Fig. 4). As a power optimization for back-to-back reads, we keep the row lines high between reads so we only have to discharge and recharge the single row being read.

E. Address-Corrected Memory Scheme

Addressability can be achieved using address translation circuitry at the nanoscale which can be programmed to present externally deterministic addresses. Detailed development of address translation schemes is beyond the scope of this paper and will be reported elsewhere. In the remainder of this paper, we account for the overhead of address correction by increasing the number of microscale address bits. To provide a feel for the impact of different addressing schemes, we show data for both the current lower bound of $\lceil 2.2 \log_2(N) \rceil + 11$ microscale address bits and the larger overhead of $7\lceil \log_2(N) \rceil$ microscale address lines.

F. Off/On Resistance Ratios

As noted in Sections II-A and C, the diode crosspoint and field-effect $R_{\text{off}}/R_{\text{on}}$ ratios are large, but finite. We must assure that these are large enough for correct memory operation. With addressing schemes based on field-effect gating for column selection and row multiplexing (Section III-B), the field-effect off/on ratio $(R_{\text{offfet}}/R_{\text{onfet}})$ is the critical one that determines how large of an array (how many NWs, N), we can build. The diode off/on ratio $(R_{\text{offdiode}}/R_{\text{ondiode}})$ does not effect scaling; it simply needs to be large enough to be distinguishable. However, $R_{\text{offfet}}/R_{\text{ondiode}}$ does affect scaling. We define D as the current ratio needed to guarantee that we can discriminate "on" and "off" crosspoint cases. In Appendix II, we derive the following set of contraints:

1) $D \times N \times R_{\text{ondiode}} < R_{\text{offfet}}$;
2) $D \times N \times R_{\text{onfet}} < R_{\text{offfet}}$;
3) $D \times R_{\text{onfet}} < R_{\text{offdiode}}$;
4) $D \times R_{\text{ondiode}} < R_{\text{offdiode}}$.

We further show that we can use the crosspoint from [2] with the NW field-effect gating from [11] to support memory arrays as large as 2000×2000. The key parameters are: $R_{\text{offfet}} > 10\ \text{G}\Omega$, $R_{\text{onfet}} < 1\ \text{M}\Omega$, $R_{\text{ondiode}} \approx 500\ \text{K}\Omega$, and $R_{\text{offdiode}} \approx 9\ \text{M}\Omega$.

IV. Raw Area

A. Parameters

The main design parameters for our memory banks are the number of row N_{row} and column N_{column} nanoscale wires used in a bank and the number of address wires N_a needed to address the nanoscale wires. N_a is a function of the number of wires addressed. Using the two schemes described in Section III-E, it is either

$$N_a(N) = 7\lceil \log_2(N) \rceil \tag{1}$$

or

$$N_a(N) = \lceil 2.2 \log_2(N) \rceil + 11. \tag{2}$$

For simplicity, we will assume that the number of rows and columns are equal.

The side length of a bank is

$$S = W_{\text{litho}} \times (N_a(N_{\text{row}}) + 5) + W_{\text{nano}} \times N_{\text{row}}. \tag{3}$$

W_{litho} is the wire pitch for the lithographic address wires, and W_{nano} is the pitch for the nanoscale wires. The extra five microscale wires account for the programming, disconnect, and pull-down lines shown in Fig. 2 along with spacing around the array core.

Raw bit density is

$$A_{\text{bit}} = \frac{S^2}{N_{\text{row}} \times N_{\text{column}}}. \tag{4}$$

Assuming $N_{\text{row}} = N_{\text{column}}$, Fig. 5 plots the raw bit density as a function of bank row length (N_{row}). Around $N_{\text{row}} = 1300$, we achieve half the density (800 nm^2 per bit for $W_{\text{nano}} = 20$ nm) of the raw wire crossings (20 nm $\times$ 20 nm $=$ 400 nm^2 per bit).

In Appendix IV-A, we estimate the area required for the bank-level CMOS support (Section III-C) and show that it will fit in the area under the microscale wire runs.

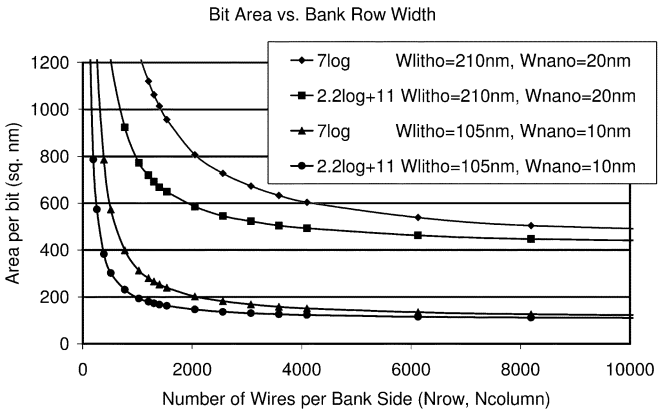


Fig. 5. Raw memory bit density as a function of N_{row}.

V. Delay

A. Random Read/Write Mode

Write time is dictated by the capacitance on the row (or column) lines and the resistance to $V_{\mathrm{prog\_row}}$ (or $V_{\mathrm{prog\_column}}$) through the microcontact-to-nanocontact (R_{contact}) and the decoder network (R_{decode}). We split the contact and decoder resistances to be explicit about the contributions of each. In the worst-case, "on" diodes may connect the desired column to all rows, forcing us to charge all rows in order to effectively charge up the column

$$T_{\mathrm{write}} = (R_{\mathrm{contact}} + R_{\mathrm{decode}}) \times (C_{\mathrm{column}} + N_{\mathrm{row}} \times C_{\mathrm{row}}). \tag{5}$$

We keep the row and column lines which we are not programming or reading at a safe nominal voltage V_{nominal}. V_{rdis} and V_{cdis} can be used to make the resistance between V_{row} (or V_{column}) and the NWs moderately large so that we can drive V_{row} (or V_{column}) to V_{nominal} in order to ensure that all undriven lines are pulled to V_{nominal} (see Figs. 2 and 3). Assuming a square array, row and column resistances and capacitances will be identical. If the pulldown resistance to V_{nominal} is large, we will want to make sure to actively drive any row or column lines to V_{nominal} between writes. We can do this after the write either by driving V_{nominal} onto $V_{\mathrm{prog\_row}}$ and $V_{\mathrm{prog\_column}}$ or by driving V_{nominal} onto V_{row} and V_{column} while at the same time lowering the effective resistance through the pulldown resistors by changing the voltage on V_{rdis} and V_{cdis}

$$T_{\mathrm{recover}} = (R_{\mathrm{contact}} + R_{\mathrm{decode}}) \times C_{\mathrm{column}} \tag{6}$$

$$T_{\mathrm{write\_cycle}} = T_{\mathrm{write}} + T_{\mathrm{recover}}. \tag{7}$$

Using the precharge read scheme, we obtain

$$T_{\mathrm{charge\_rows}} = T_{\mathrm{recover}} \tag{8}$$

$$T_{\mathrm{discharge}} = T_{\mathrm{recover}} \tag{9}$$

$$\begin{aligned} T_{\mathrm{chg\_read}} = {} & (R_{\mathrm{contact}} + R_{\mathrm{decode}}) \\ & \times (C_{\mathrm{column}} + C_{\mathrm{row}} + C_{\mathrm{out}}) \\ & + R_{\mathrm{ondiode}} \times (C_{\mathrm{row}} + C_{\mathrm{out}}) \\ & + (R_{\mathrm{contact}} + R_{\mathrm{decode}}) \times C_{\mathrm{out}} \end{aligned} \tag{10}$$

$$T_{\mathrm{chg\_read\_cycle}} = T_{\mathrm{charge\_rows}} + T_{\mathrm{discharge}} + T_{\mathrm{chg\_read}}. \tag{11}$$

B. Bulk Write Mode

Writing a memory bit to zero instead of one avoids worst-case coupling because the rectification at each crosspoint means that all of the memory points are reverse-biased during the write operation and there is no parasitic current flow through the memory junctions. Consequently, in cases where we can afford to rewrite the whole memory bank, we can program the entire bank "on" with a single write operation and then selectively write zeros. The zero write timing is then linear in the number of rows or columns

$$T_{\mathrm{write\_zero}} = (R_{\mathrm{contact}} + R_{\mathrm{decode}}) \times C_{\mathrm{column}} \tag{12}$$

$$T_{\mathrm{write\_zero\_cycle}} = T_{\mathrm{write\_zero}} + T_{\mathrm{recover}}. \tag{13}$$

This operation is similar to FLASH memory where one clears the bank and then writes individual memory bits. This mode is appropriate when an application can bulk rewrite a suitably large memory region, such as when writing a large media file or when performing nonvolatile data logging.

C. Key Electrical Parameters

Assuming the dominant capacitance is junction capacitance between crossed NW junctions and NW over microwire junctions, we can calculate the nanoscale junction capacitance (C_{nanoj}) and nanojunction–microjunction capacitance (C_{microj}) and estimate C_{column}

$$C_{\mathrm{row}} = C_{\mathrm{column}} = N_{\mathrm{row}} \times C_{\mathrm{nanoj}} + N_a(N_{\mathrm{row}}) \times C_{\mathrm{microj}}. \tag{14}$$

As developed in Appendix III-A, we will use $C_{\mathrm{nanoj}} = 10^{-18}$ F as a conservative approximation for nanoscale junction capacitance. At 90 nm, $C_{\mathrm{microj}} \approx 10^{-17}$ F and is half and a quarter this capacitance at 105 and 45 nm, respectively. We assume $C_{\mathrm{out}} \approx 10$ fF.

Rounding up typical resistance parameters (Appendix III-B), we have

$$R_{\mathrm{decode}} < 100\ \Omega \tag{15}$$

$$R_{\mathrm{contact}} \approx 0.1 \text{ to } 1\ \mathrm{M}\Omega \tag{16}$$

$$R_{\mathrm{ondiode}} \approx 0.1 \text{ to } 32\ \mathrm{M}\Omega. \tag{17}$$

At $R_{\mathrm{ondiode}} = 8\ \mathrm{M}\Omega$ or 32 MΩ and $R_{\mathrm{offfet}} = 10\ \mathrm{G}\Omega$, our constraint $D \times N \times R_{\mathrm{ondiode}} < R_{\mathrm{offfet}}$ says: $D \times N < 1000$ or 300. This suggests a need to reduce R_{ondiode} or increase R_{offfet} in order to use molecular switches in the large arrays preferred from a density standpoint. Slightly more sophisticated memory bank addressing schemes (e.g., hybrid control memory in [24]) can also reduce the array sizing constraints.

D. Read and Write Time Summary

Figs. 6 and 7 show the resulting read and write cycle times assuming $R_{\mathrm{contact}} = 1\ \mathrm{M}\Omega$ and $R_{\mathrm{ondiode}} = 0.1\ \mathrm{M}\Omega$. Note that all times are essentially linear in R_{contact} so a reduction in contact resistance of one or two orders of magnitude will translate directly into a decrease in read and write cycle times of one or two orders of magnitude. Fig. 8 shows precharge read cycle times for various values of R_{contact} and R_{ondiode}.

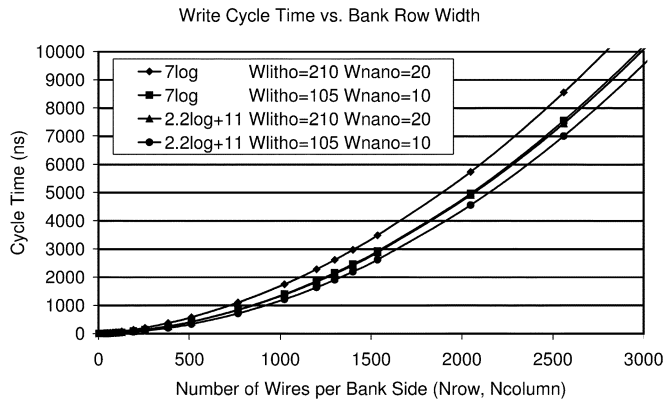


Fig. 6. Random write cycle time as a function of N_{row} with $R_{\mathrm{contact}} = 1\ \mathrm{M}\Omega$ and $R_{\mathrm{ondiode}} = 0.1\ \mathrm{M}\Omega$.

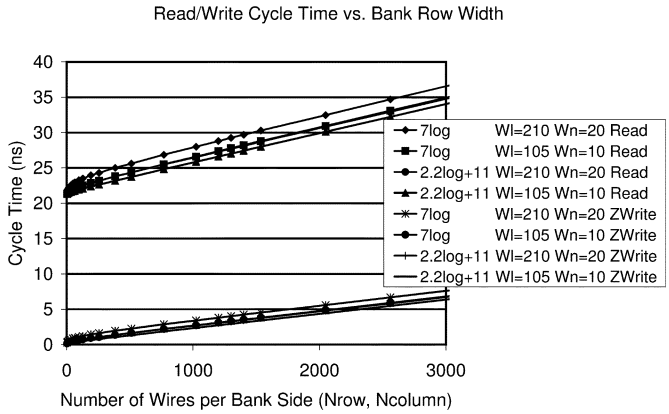


Fig. 7. Precharge read and zero write cycle time as a function of N_{row} with $R_{\mathrm{contact}} = 1\ \mathrm{M}\Omega$ and $R_{\mathrm{ondiode}} = 0.1\ \mathrm{M}\Omega$.

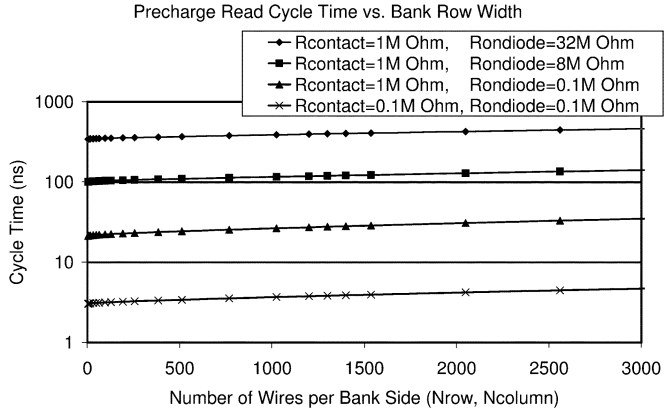


Fig. 8. Precharge read cycle time as a function of N_{row} for various resistances. $W_{\mathrm{litho}} = 105$ nm, $W_{\mathrm{nano}} = 10$ nm, $7\ \lceil \log_2(N) \rceil$ addressing.

VI. Energy

Active energy in the array is burned charging the rows and columns for reads and writes. Worst-case random writes require that we charge the intended column and the intended row and may end up coupling to all rows and columns. We burn equivalent energy, returning the array to a nominal state

$$E_{\mathrm{write}} = \frac{1}{2} C_{\mathrm{write}} (\Delta V_{\mathrm{prog}})^2 \tag{18}$$

$$C_{\mathrm{write}} = N_{\mathrm{row}} \times C_{\mathrm{row}} + N_{\mathrm{column}} \times C_{\mathrm{column}} \tag{19}$$

$$E_{\mathrm{write\_recover}} = E_{\mathrm{write}}. \tag{20}$$

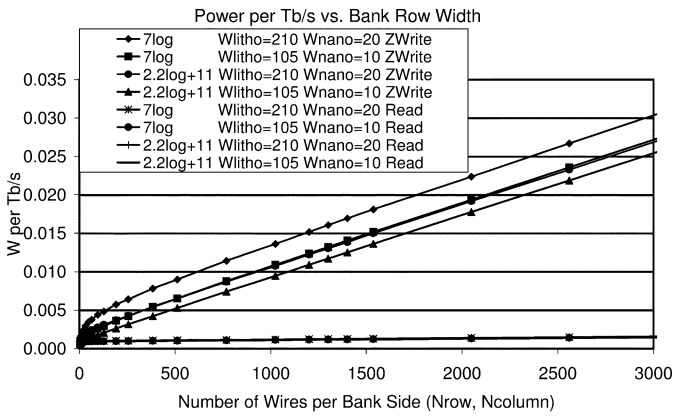


Fig. 9. Power per Tb/s a function of N_{row} for precharge read and zero write.

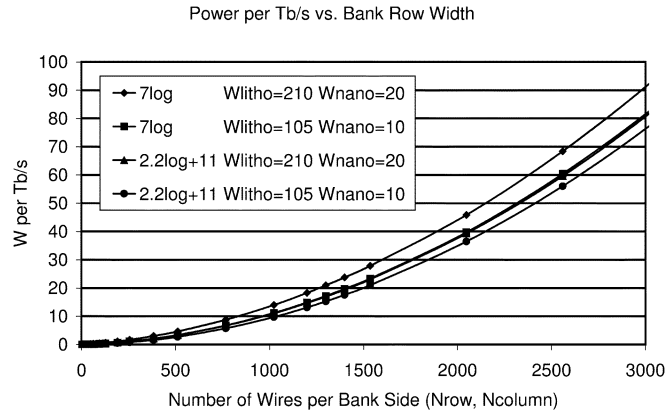


Fig. 10. Random write power per Tb/s a function of N_{row}.

Back-to-back precharge reads will discharge and charge a single row and a single column

$$E_{\mathrm{chg\_read}} = \frac{1}{2} C_{\mathrm{chg\_read}} (V_{\mathrm{read}})^2 \tag{21}$$

$$C_{\mathrm{chg\_read}} = C_{\mathrm{column}} + C_{\mathrm{row}} + C_{\mathrm{out}} \tag{22}$$

$$E_{\mathrm{chg\_read\_recover}} = E_{\mathrm{chg\_read}}. \tag{23}$$

Writing ones into the entire array is the same as a worst-case random write above. Writing a zero will only require charging a single row and column.

$$E_{\mathrm{zero\_write}} = \frac{1}{2} C_{\mathrm{zero\_write}} (\Delta V_{\mathrm{prog}})^2 \tag{24}$$

$$C_{\mathrm{zero\_write}} = C_{\mathrm{row}} + C_{\mathrm{column}} \tag{25}$$

$$E_{\mathrm{zero\_write\_recover}} = E_{\mathrm{zero\_write}}. \tag{26}$$

Power depends on how fast we actually run back-to-back memory operations. A useful metric may be the power required to read (or write) one terabit per second (1 Tb/s). Power will scale linearly with greater frequency, so we can simply scale this metric to understand the power density of higher data rates.

For write and read voltage levels, we assume

$$\Delta V_{\mathrm{prog}} \approx 2\ \mathrm{V} \tag{27}$$

$$V_{\mathrm{read}} \approx 0.3\ \mathrm{V}. \tag{28}$$

Figs. 9 and 10 show the resulting power density per Tb/s. Random writes are the only case which require more than 1 W/Tb/s, suggesting that random writes may need to be

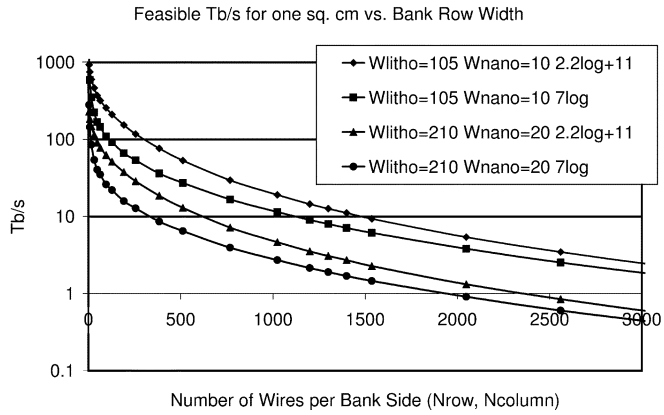


Fig. 11. Feasible read bandwidth for one cm^2 as a function of $N_{\rm row}$.

slowed relative to other operations if power density is a concern. At row and column widths of 2000, precharge read only requires 1.3 mW and zero writes only 22 mW/Tb/s.

Appendix IV-B shows that the 1.3 mW/Tb/s for the precharge read is dominated by the CMOS-level power for read operations, which we estimate at 0.3–0.6 W/Tb/s for random reads and 15–30 mW/Tb/s for sequential reads.

Feasible data bandwidths per cm^2 are $(N_{\rm bank}/t_{\rm chg\_read\_cycle})$ where we calculate $N_{\rm banks}$ as

$$N_{\rm banks} = \frac{1\ {\rm cm}^2}{S^2}. \tag{29}$$

The feasible region is plotted in Fig. 11.

VII. Defects

At this scale, we do not expect perfect devices. Wires may break or fail to make connections, molecules may be missing at junctions or statistical fluctuations may yield unusable characteristics. To accommodate these defects, we combine row and column sparing along with error-correcting codes to obtain a functional memory bank. In this section, we calculate net bit area based on the number of rows and columns we expect to yield and the overhead incurred for error correction.

A. Defect Types

The microscale-to-nanoscale junction may be difficult to make—or, at least, difficult to make completely consistently. We identify P_c as the likelihood of achieving a good microcontact-to-nanocontact. We believe this can be high but will have a few percent chance of being faulty. For example, Lieber *et al.* report greater than a 95% yield of junctions with controllable electronic characteristics [10]. The memory architectures described here require each NW to have contacts on both ends for proper operation, so a good wire will have adequate contacts with probability only $(P_c)^2$. Memories that use precharge instead of static voltage dividers may be feasible with a single contact between the microscale wires and the NWs. We will assume dual contacts here to be conservative.

A line may be broken at a junction. $P_{\rm nobreak}$ is the probability a line is not broken at a junction (i.e., each unit of $W_{\rm nano}$ length). Gudiksen *et al.* report reliable growth of SiNWs which are over 9-μm long [28] (i.e., no breaks over a distance equivalent to 900 10-nm junction lengths). This means $(P_{\rm nobreak})^{900}$ is a reasonably large value. If these can be produced with 90% likelihood (i.e., $(P_{\rm nobreak})^{900} > 0.9$), that suggests $P_{\rm nobreak} \approx 0.9999$. Sophisticated addressing and error-correction schemes might be able to tolerate breaks in lines, but we will stay with simpler schemes here.

A line may be shorted at the junction. $P_{\rm noshort}$ is the probability a line is not shorted at a junction.

A junction may be malformed such that it fails to hold a bit value reliably, but otherwise does not effect connectivity of attached lines. $P_{\rm gbit}$ is the probability that a junction can reliably store state. Chen *et al.* [19] suggest 85% of all fabricated junctions have shown good switching properties ($P_{\rm gbit} = 0.85$). As with any new technology, we are just entering the learning curve for these devices, so we expect to see much higher $P_{\rm gbit}$ values as the technology matures.

B. Net Bit Area

Appendix V shows how we compose these defect rates to compute device yield rates. We introduce $N_{\rm cyield}$ and $N_{\rm ryield}$ to represent the number of rows and columns yielded. Due to error correction (Appendix V-B), the number of corrected data bits on a column is $N_{\rm cdata}$.

Substituting the yielded row wires and net column data payload into the bank area from (4), we get the effective net bit area for useful data bits

$$A_{\rm netbit} = \frac{S^2}{N_{\rm cdata} \times N_{\rm ryield}}. \tag{30}$$

To illustrate the entire net bit area calculation, consider the calculation for: $W_{\rm litho} = 105$ nm (45-nm half-pitch node [1]), $W_{\rm nano} = 10$ nm, $P_c = 0.95$, $P_j = 0.9999$, $P_{\rm gbit} = 0.95$, $7\lceil \log_2(N_{\rm row})\rceil$ addressing. We use $N_{\rm row} = N_{\rm column} = 2038$. $N_a = 77$. This makes $S = 82 \times 105\ {\rm nm} + 2038 \times 10\ {\rm nm} = 28\,990$ nm. $P_{\rm cw} = 0.95^2 \times 0.9999^{28\,990} = 0.68$. We choose $P_{\rm correct} = 0.97$ (probability the line has few enough errors to be correctable with our error-correcting code). This makes the row NW yield rate $P_{\rm rw} = 0.95^2 \times 0.9999^{28\,990} \times 0.97 = 0.65$. We are able to yield $N_{\rm cyield} = 1335$ good column wires 99% of the time; similarly we yield $N_{\rm ryield} = 1293$ good row wires 99% of the time. To achieve $P_{\rm correct} = 0.97$, we must tolerate $N_{\rm err} = 82$ bad junctions (failures from $P_{\rm gbit}$). The Gilbert bound [29, eq. (46)] tells us that we can make our 1335 raw bits distance $d = 2 \times 82 + 1 = 185$ apart and yield at least $N_{\rm cdata} = 625$. Our net bit area is finally $((28990\ {\rm nm})^2/\ 1293 \times 625) = 1040$ nm. Since we yield our designated column and row wires each with 99% probability, that means the entire bank yields 98% of the time ($P_{\rm bank\_yield} = 0.99^2 = 0.98$).

Performing similar calculations, Fig. 12 plots the resulting net bit area for 98% bank yield probability. For $P_{\rm junc} = 0.9999$ minimum net bit area occurs when $N_{\rm row}$ is around 1800–2000. Fig. 13 shows an example of how net bit area scales with $P_{\rm gbit}$.

Table I lists several nanomemory bank organizations and summarizes their size, speed, and power requirements. For comparisons, the farthest out lithography projections from the ITRS Roadmap [1] are also included. We consider $P_c = 0.95$,

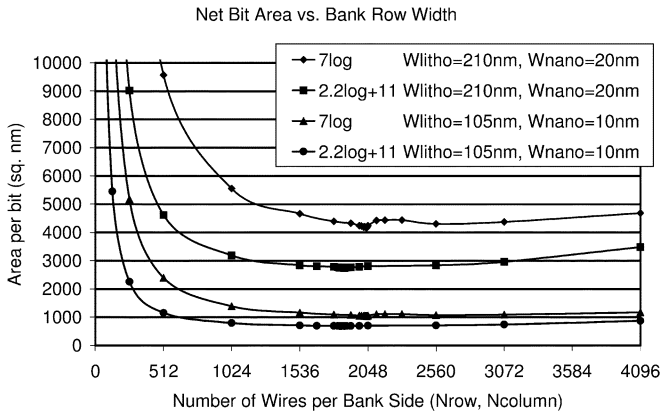


Fig. 12. Net memory bit density as a function of N_{row} ($P_c = 0.95$, $P_{\text{junc}} = 0.9999$, $P_{\text{gbit}} = 0.95$, $P_{\text{bank\_yield}} = 0.98$).

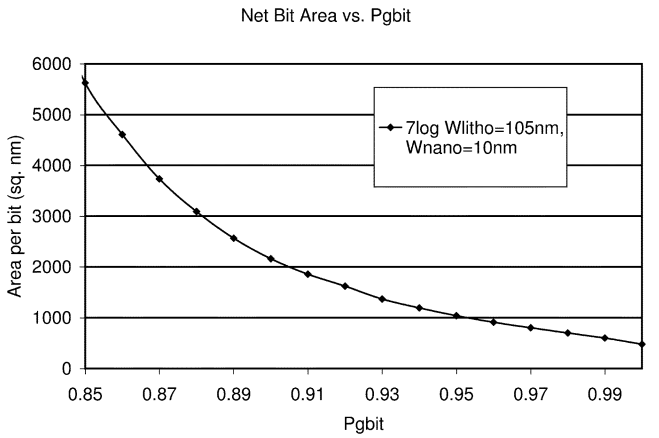


Fig. 13. Net memory bit density as a function of P_{gbit} ($7\log$, $W_{\text{litho}} = 105$, $W_{\text{nano}} = 10$, $P_c = 0.95$, $P_{\text{junc}} = 0.9999$, $P_{\text{bank\_yield}} = 0.98$).

$P_j = 0.9999$, $P_{\text{gbit}} = 0.95$ a plausible scenario; at 1040 nm^2 net area when assuming $7\lceil\log_2(N_{\text{row}})\rceil$ microscale address wires ($W_{\text{litho}} = 105$ nm), the $W_{\text{nano}} = 10$ nm bank density achieves roughly 10^{11} b/cm^2, making it denser by a factor of three than 22-nm dynamic random-access memory (DRAM) ($W_{\text{litho}} = 50$ nm). Fig. 14 shows the level of reliability for contacts (P_c) and functioning data bits (P_{gbit}) necessary to achieve 10^{11}b/cm^2 ($A_{\text{netbit}} \leq 1000$ nm^2) for various wire sizes and encoding scheme assumptions.

VIII. Discussion and Open Issues

Today, the technologies discussed here are in their infancy. They have largely been demonstrated in small batches in science labs. We have not reached the point where anyone is mass producing these device assemblies and carefully controlling their properties. Many characteristics (e.g., temperature dependence, variation in resistance or gain, robustness, failure mechanisms, and endurance) still need to be studied. Even the fundamental mechanisms at play in some of these devices are still questions of open research and inquiry. While the particular devices and fabrication techniques mentioned here are promising candidates for construction, we expect some of these devices to turn out to have problems that prevent their use, and we expect most to be superceded by devices with better characteristics. Nonetheless, the wealth of different devices and techniques now being demonstrated raises our confidence that there will be several viable solution in this space.

With all of the "Manufacturable solutions are NOT Known" entries in the conventional lithographic roadmap [1], these technologies offer a different approach to reaching nanometer-scale electronics which may compliment or be a valuable alternative to traditional approaches.

Our goal in this paper has been to anticipate some of the prospects and pitfalls of electronic memory devices built from emerging nonlithographic technologies. We deliberately consider ranges of parameters (e.g., resistances, NW widths, architectural choices, and defect rates), because there is still some uncertainty about eventual device characteristics. Our parameterized analysis may aid technologists working in device synthesis and manufacture to understand the leverage of specific device aspects. For example, we show that certain defect rates are tolerable, and we show how varying defect rates can greatly impact net memory density.

IX. Summary

Small memory arrays fabricated without photolithography have been demonstrated. With appropriate bank and defect-tolerant organizations, it will be feasible to build high-capacity memories with these techniques. After accounting for lithographic-scale addressing, defective wires, and defective bits, it appears possible these techniques will provide high densities. We show a number of plausible scenarios which achieve or exceed net usable densities of 10^{11} b/cm^2. If we can engineer the identified level of reliability in making contacts, wires, and bit storage, this density level may be attainable using modest lithographic support technologies for decoding. These technologies suggest a path to meet or exceed the semiconductor roadmap density targets without the extremely fine photolithographic patterning and registration typically assumed. Further, these nonlithographic technologies may have advantages in power consumption (tens of milliwatts per Tb/s for block reads) and fabrication capital costs, though much research, engineering, and risk remain.

Appendix I
Additional Crosspoint Technologies

Tour *et al.* demonstrate that mononitro oligo(phenylene ethynylene) molecules bridging discontinuous gold islands and gold nanorods also exhibit hysteretic switching with off/on resistance ratios separated by up to four orders of magnitude [30] at room temperature. These are set at 8 V and can be read from the 0- to -2-V range. Metal nanofilament and molecular electronic effects have been hypothesized as mechanisms for the voltage controllable resistances, with more recent evidence pointing toward filamentary metal effects [30].

Rueckes *et al.* have shown switched devices using suspended NTs to realize a bistable junction with an energy barrier between the two states [31]. The top tube is held in the "far" state above the lower conductor by mechanical forces and the distance between conductors is large enough to make tunneling current

TABLE I
DESIGN POINT ROUNDUP

Type	P_c	P_{junc}	P_{gbit}	W_{nano} (nm)	W_{litho} (nm)	Bank Rows	T_{read} (ns)	A_{netbit} (nm^2)
SRAM	[1] 22nm node				50			130000
DRAM	[1] 22nm node				50			3400
FLASH	[1] 22nm node NOR				(F=25)			8750
FLASH	[1] 22nm node single level NAND				(F=25)			2800
FLASH	[1] 22nm node multiple level NAND				(F=25)			1400
7log	0.95	0.9999	0.95	20	105	1750	30	2772
2.2log+11	0.95	0.9999	0.95	20	105	1535	28	2123
7log	0.95	0.9999	0.95	10	105	2038	31	1040
7log	0.95	0.9999	0.95	10	50	2034	30	673
2.2log+11	0.95	0.9999	0.95	10	50	1536	28	525

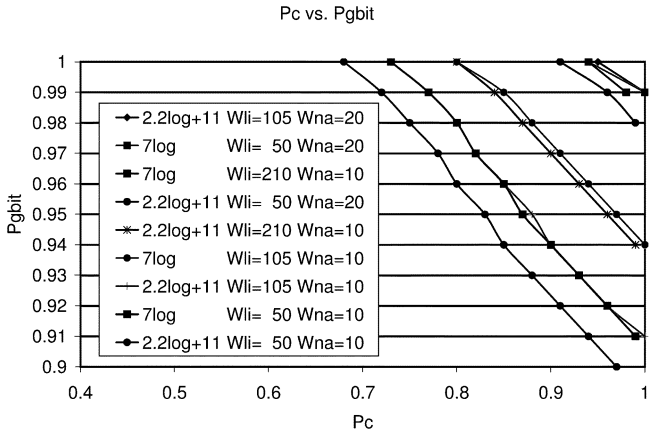


Fig. 14. Feasibility contours for achieving 10^{11} b/cm^2 ($P_c = 0.95$, $P_{\text{junc}} = 0.9999$, $P_{\text{bank\_yield}} = 0.98$).

small (gigaohms of resistance). When the tubes come into contact they are held together via molecular forces and there is little resistance (100 kΩ) between the tubes. By charging the tubes to the same or opposite polarities with an applied voltage, electrical charge attraction/repulsion allows the tubes to cross the energy gap between the two stable states, effectively setting or resetting the programming of the connection. Junctions can be directional such that the connected state exhibits pn-diode rectification behavior to support independent addressability of the memory crosspoints.

APPENDIX II
REQUIREMENTS FOR OFF/ON RESISTANCE RATIO

Both addressing schemes (Section III-B) use NW field-effect gating for the address lines. As noted, NW field-effect gating has demonstrated $R_{\text{off}}/R_{\text{on}}$ ratios in excess of 10^4. Field-effect isolation plays two roles in our memory bank design, which are: 1) output row multiplexer and 2) column selector (demultiplexer).

The row multiplexer needs to isolate nonselected rows from the output. In other words, for the precharge read, the output multiplexer needs to be able to distinguish between a crosspoint that is "on" from the leakage from all of the nonselected crosspoints. A high row would have driven the output to the high read voltage in

$$T_{\text{charge\_out\_onxpoint}} \approx R_{\text{onfet}} \times C_{\text{out}}.$$

Due to imperfect isolation, leakage from the nonselected rows will pull the output high over

$$T_{\text{charge\_out\_leak}} \approx (N-1) \times R_{\text{offfet}} \times C_{\text{out}}.$$

We can distinguish these two cases as long as

$$(N-1)R_{\text{offfet}} > R_{\text{onfet}}. \tag{31}$$

At experimentally demonstrated $R_{\text{offfet}}/R_{\text{onfet}}$ ratios, bank sizes as large as $N \approx 2000$ are workable. As we show in Section VII, defect densities also suggest bank sizes of this order.

The column selector needs to isolate the nonselected columns from V_{read}. With strong isolation in the field-effect-based column-selector and rectification in the diode crosspoints, crosspoint $R_{\text{off}}/R_{\text{on}}$ is not a limiting factor for bank scaling. First note that, regardless of the crosspoint $R_{\text{off}}/R_{\text{on}}$ ratio, we must design for the case where all of the junctions in a row or column are "on." Dealing with a low R_{off} relative to R_{on} does not change the design challenge of parasitic leakage through other junctions in the selected row or column. With strong isolation of the nonselected columns, we guarantee that the current into, and hence through, the crosspoints from the nonselected columns to the selected row is low. The precharge scheme further guarantees that we do not share current from the selected column line onto the the nonselected row lines through "on" or "off" junctions by guaranteeing that the nonselected rows are already at the same or higher potential from the read voltage; this puts all the diodes to nonselected row lines in reverse bias mode during the read.

Quantitatively, the current into the selected row wire, I_{srow}, will be

$$I_{\text{srow}} = \Delta V_{rs} \left(\sum_{0 \le i < N} \left(\frac{1}{R_{\text{decode}}[i] + R_{\text{diode}}[i]} \right) \right) \tag{32}$$

where $\Delta V_{rs} = (V_{\text{read}} - V_{\text{srow}})$. Here we lump together column NW contact resistance and the aggregate resistance of the decoder portion of the NW into $R_{\text{decode}}[i]$. For the selected column, i_{select}, we know $R_{\text{decode}}[i_{\text{select}}] = R_{\text{onfet}}$. For all others, $R_{\text{decode}}[i] = R_{\text{offfet}}$. $R_{\text{diode}}[i]$ is the resistance of the diode crosspoints between each of the columns and our selected row line. To find the largest current, we look at the worst case, i.e., where $R_{\text{diode}}[i] = R_{\text{ondiode}}$ for all $i \neq i_{\text{select}}$. We want

to be able to sense the resistance of the intended junction, $R_{\text{diode}}[i_{\text{select}}]$. Refactoring (32), we get

$$\begin{aligned} I_{\text{srow}} &= \Delta V_{rs} \left(\sum_{\substack{0 \le i < N \\ i \ne i_{\text{select}}}} \left(\frac{1}{R_{\text{decode}}[i] + R_{\text{diode}}[i]} \right) \right) \\ &\quad + \Delta V_{rs} \left(\frac{1}{R_{\text{decode}}[i_{\text{select}}] + R_{\text{diode}}[i_{\text{select}}]} \right) \\ &= \Delta V_{rs} \left(\frac{N-1}{R_{\text{offfet}} + R_{\text{ondiode}}} \right) \\ &\quad + \Delta V_{rs} \left(\frac{1}{R_{\text{onfet}} + R_{\text{diode}}[i_{\text{select}}]} \right). \end{aligned} \tag{33}$$

First, we expect $R_{\text{offfet}} \gg R_{\text{ondiode}}$. We will take this as our first contraint.

1) $R_{\text{offfet}} \gg R_{\text{ondiode}}$.

This allows us to simplify (33) to

$$I_{\text{srow}} = \Delta V_{rs} \left(\frac{N-1}{R_{\text{offfet}}} + \frac{1}{R_{\text{onfet}} + R_{\text{diode}}[i_{\text{select}}]} \right). \tag{34}$$

We can differentiate the current through the selected column junction from the leakage from the nonselected junctions when the current through the "on" crosspoint dominates the current through diodes from all the nonselected columns. We define D as the current ratio needed to guarantee that we can discriminate between these two cases:

$$\frac{1}{R_{\text{onfet}} + R_{\text{ondiode}}} > D \times \left(\frac{N-1}{R_{\text{offfet}}} \right). \tag{35}$$

This allows us to refine constraint 1) and derive a second constraint as follow:

1) $D \times N \times R_{\text{ondiode}} < R_{\text{offfet}}$.
2) $D \times N \times R_{\text{onfet}} < R_{\text{offfet}}$.

To be able to distinguish the "off" case from the "on," case we also need

$$\frac{1}{R_{\text{onfet}} + R_{\text{ondiode}}} > D \left(\frac{1}{R_{\text{onfet}} + R_{\text{offdiode}}} \right). \tag{36}$$

We start by assuming

3) $R_{\text{offdiode}} > R_{\text{onfet}}$.

Equation (36) tells us to refine this constraint and add a fourth constraint

3) $D \times R_{\text{onfet}} < R_{\text{offdiode}}$.
4) $D \times R_{\text{ondiode}} < R_{\text{offdiode}}$.

This shows that $R_{\text{off}}/R_{\text{on}}$ for the crosspoint diode does not affect scaling. We do have a scaling issue for R_{ondiode} (constraint 1 above): $R_{\text{offfet}}/R_{\text{ondiode}} > N \times D$.

From this, we can see that the molecules from [2] with the NW field-effect gating from [11] can support memory arrays as large as 2000×2000. The key parameters are: $R_{offfet} > 10\ \text{G}\Omega$, $R_{\text{onfet}} < 1\ \text{M}\Omega$, $R_{\text{ondiode}} \approx 500\ \text{K}\Omega$, and $R_{\text{offdiode}} \approx 9\ \text{M}\Omega$. With $N = 2000$ and $D = 4$, we meet all four constraints.

1) $4 \times 2000 \times 500\ \text{k}\Omega = 4\ \text{G}\Omega < 10\ \text{G}\Omega$.
2) $4 \times 2000 \times 1\ \text{M}\Omega = 8\ \text{G}\Omega < 10\ \text{G}\Omega$.
3) $4 \times 1\ \text{M}\Omega = 4\ \text{M}\Omega < 9\ \text{M}\Omega$.
4) $4 \times 500\ \text{k}\Omega = 2\ \text{M}\Omega < 9\ \text{M}\Omega$.

APPENDIX III
ELECTRICAL PARAMETERSS

A. Capacitance

From [9]

$$C_{\text{junc}} \approx 2\pi\epsilon_k \frac{L}{\ln\left(\frac{2h}{r}\right)}$$

where r is the NW radius, h is the distance between conductors, and L is the length of the overlap. For NW junctions, we assume: $r = 1$ nm, $h = 1$ nm, $L = 2r$, and an SiO_2 dielectric $\epsilon_{Si0_2} = 3.4 \times 10^{-11}$ F/m

$$C_{\text{nanoj}} \approx 2\pi \left(3.4 \times 10^{-11}\ \frac{\text{F}}{\text{m}} \right) \left(\frac{2\text{nm}}{\ln(2)} \right) \approx 6 \times 10^{-19}\ \text{F}. \tag{37}$$

We will use $C_{\text{nanoj}} = 10^{-18}$ F as a conservative approximation. For the NW runs over the microwires, $L = W_{\text{litho}}/2$ assuming half the pitch is in the wire and half in the spacing between wires. Here, height is oxide thickness between the address lines and the nanoscale core memory wires, which is likely to be 5–10 nm (assume $h = 5$ nm)

$$C_{\text{microj}} \approx 2\pi \left(3.4 \times 10^{-11}\ \frac{\text{F}}{\text{m}} \right) \left(\frac{\frac{W_{\text{litho}}}{2}}{\ln\left(2 \times \frac{5}{1}\right)} \right). \tag{38}$$

B. Resistance

Rueckes *et al.* [31] suggests diode "on" resistance around $R_{\text{ondiode}} = 100\ \text{k}\Omega$. Scaling the junctions in Chen *et al.* [2] to 10 nm × 10 nm crosspoints ($W_{\text{nano}} = 20$ nm) gives a diode "on" resistance of around 8 MΩ; similarly, at $W_{\text{nano}} = 10$ nm, we have 25-nm^2 junctions with 32-MΩ diode "on" resistance.

Typical contact resistance is currently in the 1-MΩ range [10], [32]. For NTs, the fundamental limit on contact resistance is 6.5 kΩ, and it appears possible to approach this limit [33]. Consequently, it should be possible to decrease contact resistance with further technology mastery. In recent work it has already been shown how to bring this down to 100 kΩ [11]. Note that low values for R_{onfet} for field-effect NWs mentioned earlier (Section II-C and Appendix II) are limited by the contact resistance and not the actual NW resistance. Reduced contact resistance may actually help increase $R_{\text{offfet}}/R_{\text{onfet}}$.

Rounding up typical resistance parameters, we have

$$R_{\text{decode}} < 100\ \Omega \tag{39}$$

$$R_{\text{contact}} \approx 0.1 \text{ to } 1\ \text{M}\Omega \tag{40}$$

$$R_{\text{ondiode}} \approx 0.1 \text{ to } 32\ \text{M}\Omega. \tag{41}$$

TABLE II
SYMBOL ROUNDUP FOR EXPRESSIONS IN DEFECT CALCULATIONS

Symbol	Description	Intro
A_{netbit}	Net effective area for one yielded bit	Eq. 30
d	Distance between code words	Append. V-B
N	Number of raw wires used generically for N_{row} or N_{column}	Sec. III
N_a	Number of address bits	Eq. 1 and 2
N_{cdata}	Number of net data bits will get from a single row	Eq. 46
N_{cerr}	Number of errors tolerable on yielded column wires	Append. V-B
N_{column}	Number of raw column wires	Sec. IV
N_{cyield}	Number of good column wires	Append. V-A
N_{row}	Number of raw row wires	Sec. IV
N_{ryield}	Number of good row wires	Append. V-A
N_{yield}	Number of good wires used generically for N_{cyield} or N_{ryield}	Append. V-A
P_{bank_yield}	Probability bank will yield	Append. V-A
P_c	Probability of good micro-to-nano contact	Sec. VII-A
P_{cw}	Probability a column wire yields	Eq. 43
P_{gbit}	Probability a junction can be set, cleared, and will hold its value	Sec. VII-A
P_{junc}	Probability a wire has no fault at a junction	Eq. 42
$P_{nobreak}$	Probability a wire is not broken at a junction	Sec. VII-A
$P_{noshort}$	Probability a wire is not shorted at a junction	Sec. VII-A
P_{rw}	Probability a row wire yields	Eq. 47
P_w	Generic Probability a row or column wire yields	Append. V-A
$P_{wcorrect}$	Probability a wire is correctable	Eq. 45
P_{yield}	Probability will yield adequate row or column wires	Eq. 48
S	Length of a side of the bank	Eq. 3

APPENDIX IV
CMOS SUPPORT

A. Circuit Area

As noted in Section III-C, the CMOS circuitry to support each bank is restricted to the following:

- buffers for row and column addresses $(2 \times N_a(N_{\text{row}}))$;
- buffers for V_{row}, V_{rdis}, V_{col}, and V_{cdis} (4);
- gates and transistors for locally switching the appropriate power rails onto $V_{\text{prog\_column}}/V_{\text{read}}$ and $V_{\text{prog\_row}}/V_{\text{out}}$ (four three-input gates and five transistors);
- read buffer or sense-amp (1).

The drive strength on these devices does not need to be high as they each only drive about 20 μm of local metal. Counting each three-input gate as two four-transistor gates and each pass transistor as one, we need about $(2 \times N_a(N_{\text{row}}) + 18)$ four-transistor gates. At $N_{\text{row}} = 1024$, and using (1), we end up needing around 160 four-transistor gates. In a 90-nm process, each gate is about 5 μm^2 [1]. This gives us an upper bound of 800 μm^2. Note that the horizontal and vertical microscale wire runs included in this array for this technology ($W_{\text{litho}} =$ 210 nm, $W_{\text{nano}} =$ 20 nm) will each be $(75 \times 210 \text{ nm} \approx 15\ \mu\text{m}) \times (1024 \times 20 \text{ nm} \approx 20\ \mu\text{m}) = 300\ \mu\text{m}^2$. That is, the area under the horizontal and vertical metal runs, not including the area where they overlap, is about 75% of the area required for these buffers. If we can share each of the address tap buffers across a pair of adjacent arrays, this logic will easily fit completely underneath the microscale wire runs. Note that the area under each of these wire runs scale with $N_a(N_{\text{row}})$, as does the dominant buffer requirement. The logic-area-to-wire-area ratio decreases with increasing N_{row}. For all of the technology combinations in this work, the ratio is less than one for $N_{\text{row}} > 2000$; the ratio is less than 2 for $N_{\text{row}} > 700$. The example used here is actually the worst-case example, with the ratio decreasing with more compact addressing schemes and smaller CMOS geometries.

B. Circuit Energy and Power

Each of the microscale wire runs is, conservatively, under 10 fF per bank. We have about $2 \times N_a(N_{\text{row}}) + 10$ such microscale wire runs and pay for each of them twice to have both a global and a local version of each line. For $N_{\text{row}} = 1024$, and using the address upper bound (1), this means we have 3 pF of wire capacitance to drive on each operation. Running the address and control lines at 0.6 V requires roughly 0.54-pJ/bank/read operation ($W_{\text{litho}} = 105$, $W_{\text{nano}} = 10$). With 160 four-transistor gates, a 2:1 P:N width ratio, and $W/L = 3$ for the nmos transistors, we have 1000 $W/L = 3$ transistors switching. Each such transistor switching costs 0.015 fJ [1], giving us a total logic energy of 0.015-pJ/bank/read operation. Together, this gives us 550 mW/Tb/s. By comparison, the 1.3 mW/Tb/s for the NW array is negligible. The lower bound addressing scheme (2) requires half of this power.

The preceeding calculation is for random reads that require all of the address lines to switch. Reading within a row or column will keep one address constant and effectively cut the power in half. Further, sequential read sequences will, on average, only change a few bits of the address, reducing this power by an order of magnitude. Adding together the advantages of the lower bound addressing scheme and sequential reads along one row, it may be possible to read or write a block-addressed memory with as little as 15 mW/Tb/s used in the memory component we consider here.

APPENDIX V
DEFECT CALCULATION

In this appendix, we develop the complete defect calculations used in Section VII. Table II summarizes the symbols used in these calculations.

A. Composing Defects

A single break or a single short will force us to discard an entire row or column wire, so we will combine the lack of a

break and lack of a short probabilities into a probability that a junction is good

$$P_{\text{junc}} = P_{\text{nobreak}} \times P_{\text{noshort}}. \tag{42}$$

For a column wire to yield, the wire must connect on both ends and have a good set of junctions

$$P_{\text{cw}} = (P_c)^2 \times (P_{\text{junc}})^{N_{\text{row}}}. \tag{43}$$

For a row wire to yield, the wire must connect on both ends and have a good set of junctions and reliable crosspoints

$$P_{\text{rw}} = (P_c)^2 \times (P_{\text{junc}})^{N_{\text{column}}} \times (P_{\text{gbit}})^{N_{\text{column}}}. \tag{44}$$

B. Bit Error Correction

We can improve on (44) by using an error-correcting code (ECC) so we can tolerate a number of bad bits on a row wire. For simplicity, we assume each row line is a separate code word. After accounting for failed column lines, the row line has a total of N_{cyield} bits equal to the number of yielded column lines. We pick an error correcting code that will tolerate N_{cerr} bit errors. In general, the N_{cyield} bits will encode only $N_{\text{cdata}} < N_{\text{cyield}}$ bits. If a line has more than N_{cerr} bad bits, then an N_{cerr}-correcting code will not be able to repair it, and we will consider the line unusable and discard it as a line that did not yield. We can calculate the probability that an otherwise good wire is correctable as an M-of-N calculation

$$P_{w\text{correct}} = \sum_{i=(N_{\text{cyield}}-N_{\text{cerr}})}^{N_{\text{cyield}}} \left(\binom{N_{\text{cyield}}}{i} (P_{\text{gbit}})^i \times (1-P_{\text{gbit}})^{N_{\text{cyield}}-i} \right). \tag{45}$$

To correct N_{err} errors, we must make our codewords a distance $d = 2 \times N_{\text{err}} + 1$ apart. The Gilbert bound [29] gives us a lower bound on the useful data N_{cdata} that we can extract making our stored codes distance d apart within the N_{cyield} bits available:

$$N_{\text{cdata}} \geq \log_2 \left(\sum_{i=0}^{d-2} \binom{N_{\text{cyield}}-1}{i} \right). \tag{46}$$

The Gilbert bound is for practical achievable codes. Good error-correcting codes can typically extract even more data. For example, if we have $N_{\text{cyield}} = 266$ and $N_{\text{err}} = 8$, then the Gilbert bound tells us we can achieve $N_{\text{data}} \geq 232$. A 266-b Reed–Solomon subfield code [34] can recover 240 b for the same eight errors. We use the Gilbert bound as a conservative estimate of the bit yield in the net density calculations in this paper.

Using error correction, we can modify (44) to allow a number of bit errors. Now, for a row wire to yield, the wire needs to connect on both ends and have a good set of junctions, but need only have few enough bit errors to be correctable.

$$P_{\text{rw}} = (P_c)^2 \times (P_{\text{junc}})^{N_{\text{column}}} \times P_{w\text{correct}}. \tag{47}$$

Discarding column wires with too many bad bits could also be used to improve upon this yield. We will not perform such an optimization for this calculation.

C. Bank Yield

Knowing P_w (P_{rw} or P_{cw}, respectively), we can calculate the number of row or column wires we expect to yield to a given confidence level (N_{ryield}, N_{cyield})

$$P_{\text{yield}} = \sum_{i=N_{\text{yield}}}^{N} \left(\binom{N}{i} (P_w)^i (1-P_w)^{N-i} \right). \tag{48}$$

P_{yield} is the target probability (for rows this is P_{ryield} and for columns, P_{cyield}) of yielding at least N_{yield} wires given a bank with N wires each of which is functional with probability P_w. Typically we will pick a suitable P_{yield} level based on our desired bank yield rate and use that to determine the appropriate relationship between N_{yield} and N. The probability we yield a good bank is the product of the probability we yield the desired number of row wires and the probability we yield the appropriate number of column wires, so $P_{\text{bank\_yield}} = P_{\text{ryield}} \times P_{\text{cyield}}$.

ACKNOWLEDGMENT

Architecture work at this early stage is only feasible and meaningful in close cooperation with scientists working on device properties and fabrication. Special thanks to J. Heath, C. Lieber, S. Williams, and D. Stewart for their support in this work. The authors are also grateful to M. Ziegler and J. Ellenbogen for drawing attention to some of the parasitic coupling effects in the diode memory banks. The anonymous reviewers made good observations and suggestions that helped make this a more complete and accessible paper

REFERENCES

[1] International Technology Roadmap for Semiconductors (2001). [Online]. Available: http://public.itrs.net/Files/2001ITRS/

[2] Y. Chen, D. A. A. Ohlberg, X. Li, D. R. Stewart, R. S. Williams, J. O. Jeppesen, K. A. Nielsen, J. F. Stoddart, D. L. Olynick, and E. Anderson, "Nanoscale molecular-switch devices fabricated by imprint lithography," *Appl. Phys. Lett.*, vol. 82, no. 10, pp. 1610–1612, 2003.

[3] D. R. Stewart, D. A. A. Ohlberg, P. A. Beck, Y. Chen, R. S. Williams, J. O. Jeppesen, K. A. Nielsen, and J. F. Stoddart, "Molecule-independent electrical switching in Pt/organic monolayer/Ti devices," *Nanoletters*, vol. 4, no. 1, pp. 133–136, 2004.

[4] C. Dekker, "Carbon nanotubes as molecular quantum wires," *Phys. Today*, pp. 22–28, May 1999.

[5] R. Krupke, F. Hennrich, H. von Löhneysen, and M. M. Kappes, "Separation of metallic from semiconducting single-walled carbon nanotubes," *Science*, vol. 301, pp. 344–347, Jul. 2003.

[6] Y. Cui, L. J. Lauhon, M. S. Gudiksen, J. Wang, and C. M. Lieber, "Diameter-controlled synthesis of single crystal silicon nanowires," *Appl. Phys. Lett.*, vol. 78, no. 15, pp. 2214–2216, 2001.

[7] A. M. Morales and C. M. Lieber, "A laser ablation method for synthesis of crystalline semiconductor nanowires," *Science*, vol. 279, pp. 208–211, 1998.

[8] Y. Wu and P. Yang, "Germanium nanowire growth via simple vapor transport," *Chem. Materials*, vol. 12, pp. 605–607, 2000.

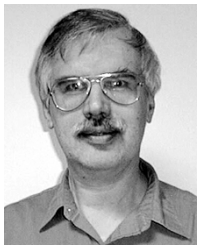
[9] Y. Cui, X. Duan, J. Hu, and C. M. Lieber, "Doping and electrical transport in silicon nanowires," *J. Phys. Chem. B*, vol. 104, no. 22, pp. 5213–5216, Jun. 2000.
[10] Y. Huang, X. Duan, Y. Cui, L. Lauhon, K. Kim, and C. M. Lieber, "Logic gates and computation from assembled nanowire building blocks," *Science*, vol. 294, pp. 1313–1317, 2001.
[11] Y. Cui, Z. Zhong, D. Wang, W. U. Wang, and C. M. Lieber, "High performance silicon nanowire field effect transistors," *Nanoletters*, vol. 3, no. 2, pp. 149–152, 2003.
[12] A. DeHon, "Array-based architecture for fet-based, nanoscale electronics," *IEEE Trans. Nanotechnol.*, vol. 2, no. 2, pp. 23–32, Mar. 2003.
[13] Y. Huang, X. Duan, Q. Wei, and C. M. Lieber, "Directed assembley of one-dimensional nanostructures into functional networks," *Science*, vol. 291, pp. 630–633, Jan. 2001.
[14] D. Whang, S. Jin, and C. M. Lieber, "Nanolithography using hierarchically assembled nanowire masks," *Nanoletters*, vol. 3, no. 7, pp. 951–954, July 2003.
[15] Y. Chen, D. A. A. Ohlberg, G. Medeiros-Ribeiro, Y. A. Chang, and R. S. Williams, "Self-assembled growth of epitaxial erbium disilicide nanowires on silicon (001)," *Appl. Phys. Lett.*, vol. 76, no. 26, pp. 4004–4006, 2000.
[16] S. Y. Chou, P. R. Krauss, and P. J. Renstrom, "Imprint lithography with 25-nanometer resolution," *Science*, vol. 272, pp. 85–87, 1996.
[17] Y. Xia and G. M. Whitesides, "Soft lithography," *Annu. Rev. Mat. Sci.*, vol. 28, pp. 153–84, 1998.
[18] S. Y. Chou, P. R. Krauss, W. Zhang, L. Guo, and L. Zhuang, "Sub-10 nm imprint lithography and applications," *J. Vac. Sci. Technol. B, Microelectron. Process. Phenom.*, vol. 15, no. 6, pp. 2897–2904, Nov.–Dec. 1997.
[19] Y. Chen, G.-Y. 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*, vol. 14, pp. 462–468, 2003.
[20] N. A. Melosh, A. Boukai, F. Diana, B. Gerardot, A. Badolato, P. M. Petroff, and J. R. Heath, "Ultrahigh-density nanowire lattices and circuits," *Science*, vol. 300, pp. 112–115, Apr. 2003.
[21] 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.
[22] C. L. Brown, U. Jonas, J. A. Preece, H. Ringsdorf, M. Seitz, and J. F. Stoddart, "Introduction of [2] catenanes into langmuir films and langmuir-blodgett multilayers. A possible strategy for molecular information storage materials," *Langmuir*, vol. 16, no. 4, pp. 1924–1930, 2000.
[23] S. Williams and P. Kuekes, "Demultiplexer for a molecular wire crossbar network," U.S. Patent 6 256 767, Jul. 3, 2001.
[24] A. DeHon, P. Lincoln, and J. Savage, "Stochastic assembly of sublithographic nanoscale interfaces," *IEEE Trans. Nanotechnol.*, vol. 2, no. 3, pp. 165–174, Sep. 2003.
[25] M. S. Gudiksen, L. J. Lauhon, J. Wang, D. C. Smith, and C. M. Lieber, "Growth of nanowire superlattice structures for nanoscale photonics and electronics," *Nature*, vol. 415, pp. 617–620, February 2002.
[26] Y. Wu, R. Fan, and P. Yang, "Block-by-block growth of single-crystalline Si/SiGe superlattice nanowires," *Nano Lett.*, vol. 2, no. 2, pp. 83–86, Feb. 2002.
[27] M. T. Björk, B. J. Ohlsson, T. Sass, A. I. Persson, C. Thelander, M. H. Magnusson, K. Depper, L. R. Wallenberg, and L. Samuelson, "One-dimensional steeplechase for electrons realized," *Nano Lett.*, vol. 2, no. 2, pp. 87–89, Feb. 2002.
[28] M. S. Gudiksen, J. Wang, and C. M. Lieber, "Synthetic control of the diameter and length of semiconductor nanowires," *J. Phys. Chem. B*, vol. 105, pp. 4062–4064, 2001.
[29] G. C. Clark, Jr. and J. B. Cain, *Error-Correction Coding for Digital Communications*. New York: Plenum, 1981.
[30] J. M. Tour, L. Cheng, D. P. Nackashi, Y. Yao, A. K. Flatt, S. K. S. Angelo, T. E. Mallouk, and P. D. Franzon, "Nanocell electronic memories," *J. Amer. Chem. Soc.*, vol. 125, no. 43, pp. 13279–13 283, 2003.
[31] T. Rueckes, K. Kim, E. Joselevich, G. Y. Tseng, C.-L. Cheung, and C. M. Lieber, "Carbon nanotube based nonvolatile random access memory for molecular computing," *Science*, vol. 289, pp. 94–97, 2000.
[32] C. M. Lieber and X. Duan, "NanoFET threshold voltages," unpublished, Dec. 2001.
[33] P. L. McEuen, M. Fuhrer, and H. Park, "Single-walled carbon nanotubes electronics," *IEEE Trans. Nanotechnol.*, vol. 1, no. 2, pp. 75–85, Mar. 2002.
[34] J. Bierbrauer and Y. Edel, "New code parameters from Reed–Solomon subfield codes," *IEEE Trans. Inform. Theory*, vol. 43, no. 3, pp. 953–968, May 1997.

André DeHon (S'92–M'96) received the S.B., S.M., and Ph.D. degrees in electrical engineering and computer science from the Massachusetts Institute of Technology (MIT), Cambridge, in 1990, 1993, and 1996, respectively.

From 1996 to 1999, he co-ran the BRASS Group, Computer Science Department, University of California at Berkeley. Since 1999, he has been an Assistant Professor of computer science with the Computer Science Department, California Institute of Technology, Pasadena. He is broadly interested in the physical implementation of computations from substrates, including very large scale integration (VLSI) and molecular electronics, up through architecture, computer-aided design (CAD), and programming models. He places special emphasis on spatial programmable architectures (e.g., field-programmable gate arrays) and interconnect design and optimization.

Seth Copen Goldstein (M'91) received the B.S degree from Princeton University, Princeton, NJ, in 1985, and the M.S. and Ph.D. degrees from the University of California at Berkeley, in 1994 and 1997, respectively, all in computer science.

In 1997, he joined the faculty of Carnegie Mellon University, Pittsburgh, PA, where he is currently an Associate Professor with the Department of Computer Science. His research focuses on computing systems and nanotechnology. Currently, he is involved with architectures, compilers, and tools for computer systems built with electronic nanotechnology. He believes that the fundamental challenge for computer science in the 21st Century is how to effectively harness systems that contain billions of potentially faulty components. In pursuit of meeting this challenge, he is working on novel circuit techniques, defect and fault tolerance, reconfigurable architectures, scalable optimizing compilers for spatial computing, and self-organizing systems.



Philip J. Kuekes (M'69) received the B.S. degree in physics from Yale University, New Haven, CT, in 1969.

He is currently a Member of the Technical Staff with Hewlett-Packard Laboratories, Palo Alto, CA. He designed mega-op array processors in the 1970s and giga-op systolic processors in the 1980s. In 1991, he joined Hewlett-Packard Laboratories as a Project Manager for Teramac, a trillion operations per second reconfigurable computer. Teramac is the largest defect tolerant processor ever made. Three quarters of the 864 chips in Teramac have defects. His current research interests are at the intersection of massively parallel computer architectures and chemistry. As a researcher in the HP Defect Tolerant Moletronics effort, he is investigating a reconfigurable architecture that allows one to electrically download the designed complexity of a computer into a chemically assembled regular, but imperfect nanostructure.

Patrick Lincoln received the S.B. degree from the Massachuestts Institute of Technology (MIT), Cambridge, in 1986, and the Ph.D. degree from Stanford University, Stanford, CA, in 1992, both in computer science.

He is currently the Director of the Computer Science Laboratory, SRI International, Menlo Park, CA, which he joined in 1989. He has previously held positions at MCC and Los Alamos National Laboratory. He directs research and is an active researcher in the fields of molecular electronics, formal methods, computer security and privacy, bioinformatics, scalable distributed systems, and programming language design and implementation. He has made significant contributions to the formal analysis of systems, languages, and protocols in computer security, privacy, and fault tolerance, and to their integration into scalable and survivable systems. He has co-chaired Defense Advanced Research Projects Agency (DARPA)-sponsored Information Science and Technology (ISAT) studies, and serves on the Scientific Advisory Boards of private and public companies including Cable & Wireless.