Contents lists available at ScienceDirect

Theoretical Computer Science

journal homepage: www.elsevier.com/locate/tcs

Nanowire addressing with randomized-contact decoders

Eric Rachlin, John E. Savage *

Computer Science, Brown University, Providence, RI 02912-1910, United States

ARTICLE INFO

Keywords:
Nanotechnology
Decoder
Memories
Stochastic assembly

ABSTRACT

Methods for assembling crossbars from nanowires (NWs) have been designed and implemented and methods for controlling individual NWs within a crossbar have also been proposed. However, implementation remains a challenge. A NW decoder is a device that controls many NWs with a much smaller number of lithographically produced mesoscale wires (MWs). Unlike traditional demultiplexers, all proposed NW decoders are assembled stochastically. In a randomized-contact decoder (RCD), for example, field-effect transistors are randomly created at about half of the NW/MW junctions.

In this paper, we tightly bound the number of MWs required to produce a correctly functioning RCD with high probability. We show that the number of MWs is logarithmic in the number of NWs, even when manufacturing errors occur. We also analyze the overhead associated with controlling a stochastically assembled decoder. As we explain, lithographically-produced control circuitry must store information regarding which MWs control which NWs. This requires more area than the MWs themselves, but has received little attention elsewhere. Finally we analyze several simple testing algorithms for configuring this control circuitry. We demonstrate an unexpected tradeoff between testing time and the number of MWs required by an RCD.

1. Introduction

The dream of nanoscale computing was first articulated by Richard Feynman in his 1959 speech to the American Physical Society. He argued that no physical law prevented the room-sized computers of the 1950's from being replaced with vastly more powerful pin-sized computers built from billions of nanoscale devices. Since then computers have become orders of magnitude smaller, faster and more powerful. Their wires and gates, however, have not yet reached the nanoscale (i.e., the dimensions of individual molecules).

Although individual nanoscale devices have been demonstrated, we lack the ability to place these devices with nanoscale precision. As a result, near-term nanoscale architectures will be assembled stochastically. The range of device variation these architectures must tolerate, makes them fundamentally different from today's CMOS. Our approach to nanoscale circuit design must change accordingly.

For the past 30 years, chips with ever shrinking features have been produced, using photolithography. Wires and gates are defined using light on a silicon substrate. This allows many copies of a chip to be produced from a single set of costly masks. Although photolithography allows for a very wide range of circuit designs, the wavelength of light (193 nm or greater) is too large to allow for features on the order of a few nanometers, the range considered in this paper. Nanoscale architectures require new manufacturing technology.

A particularly viable basis for nanoscale architectures that has received significant attention in the physical science and engineering communities is the **nanowire crossbar** [1,2] (see Fig. 1). Here a grid of nanowires (NWs) provides control over

* Corresponding author.
E-mail addresses: eerac@cs.brown.edu (E. Rachlin), jes@cs.brown.edu (J.E. Savage).

0304-3975/$ – see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.tcs.2008.08.011

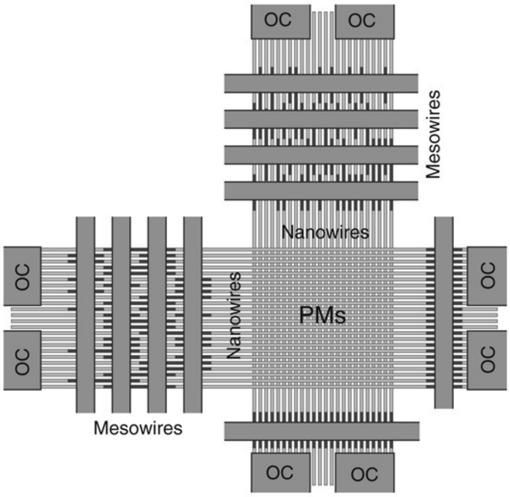

Fig. 1. A crossbar formed from two orthogonal sets of NWs with programmable molecules (PMs) at the crosspoints defined by intersecting NWs. NWs are divided into contact groups by connecting them to ohmic contacts (OCs). To activate a NW in one dimension, a contact group is activated and MWs are used to deactivate all but one NW in that group. Data is stored at a crosspoint by applying a large electric field across it. Data is sensed with a smaller field.

molecular devices that reside at their crosspoints. Like traditional crossbars, NW crossbars can act as memories (such as RAM) and circuits (such as PLAs) [3,4]. Unlike traditional crossbars, however, their assembly is stochastic. This results in three very important challenges:

(1) NWs are randomly assigned physical addresses.
(2) A testing procedure is required to configure a crossbar's control circuitry.
(3) Permanent and transient faults must be tolerated.

To overcome these challenges, nanoscale crossbar-based architectures rely on stochastically assembled NW decoders. A **nanowire decoder** is any device capable of controlling the resistances of individual NWs, using larger lithographically-produced mesoscale wires (MWs) and ohmic contacts (OCs). In Section 2 we explain how NW decoders can be used to control a NW crossbar-based memory.

To date, all proposed NW decoders rely on a stochastic assembly process. Three types of NW decoder have been analyzed probabilistically, "encoded NW decoders", "mask-based decoders", and "randomized-contact decoders". In Section 3 we describe these decoders and summarize their performance. We also provide a new bound on the number of MWs an encoded NW decoder requires to control a large fraction of all NWs.

In Section 4 we examine in detail the addressing of NWs by MWs and derive conditions that must be met by the resistances of NW/MW junctions in order to address NWs correctly. This allows us to give a model of NW decoders that explicitly takes manufacturing errors into account.

In Section 5 we use this model to derive probabilistic bounds on the number of MWs needed to address NWs with the randomized-contact decoder (RCD). We bound the number of MWs needed to address all NWs connected to a single OC. We also bound the number of MWs required to address some fixed fraction of NWs across all OCs. Our analysis demonstrates that RCDs are efficient and robust. They can reliably control a large number of NWs using a small number of MWs, even when some fraction of contacts between MWs and NWs are defective.

Our bounds improve upon the analysis of [5]. They take errors into account, and also make explicit the probability that the bounds hold. Additionally, their derivation is more precise, as it avoids an unnecessary independence approximation (see Section 3.3). This ensures that our bounds apply even to small groups of NWs connected to OCs. In the case of a large number of NWs connected to an OC, our bound on the number of MWs required to address all NWs agrees with the asymptotic analysis in in [5].

We also note that previous work on coping with manufacturing defects in NW decoders has focused on the use of error correcting codes [6–8]. In these coding-based approaches, a specific error correcting code is used to determine which subsets of MWs have the ability to address NWs. Viewed in this light, the bounds of Section 5 show that even randomly generated codes provide good defect-tolerance, with only constant factor overhead.

This is a significant insight, since it eliminates the requirement that a NW decoder provide designers with a great deal of control over which subsets of MWs can control NWs. This level of control is present in encoded NW decoders [8], or programmable NW decoders [9,7] (which themselves require a second NW decoder to configure), but is not present in RCDs, which are arguably simpler to manufacture.

Since we cannot predict in advance which MWs will control which NWs, NW addresses must be discovered after an RCD is assembled. These addresses must then be stored and mapped to fixed binary addresses, using programmable circuitry. Strategies for implementing this mapping and a method for estimating the area used by a crossbar as well as its addressing circuitry are discussed in Section 6. The need for such circuitry is also mentioned in [3], but the varying overhead associated with specific mapping strategies is not considered. In [10] specific mappings are considered in the context of encoded NW decoders, but not the "Almost All Wires Addressable" and "Take What You Get" strategies presented here.

The problem of discovering which subsets of MWs address NWs is discussed in Section 7. An exhaustive search is considered, as is a search randomized algorithm. The randomized algorithm is analyzed in Section 8. Again, we improve on work done in [5], relaxing an assumption about what testing circuitry is present. We also correct an independence assumption that is not met, unless the number of MWs used to control the NWs connected to an OC is much larger than what the bounds of Section 5 require. Conclusions are drawn in Section 9.

2. Crossbar overview

To motivate our analysis, we briefly describe how NW crossbars can be used as memories. This approach can be extended to circuits as well [4]. Since our research focuses on controlling individual NWs with MWs, crossbar-based memories offer sufficient motivation.

2.1. Crossbar assembly

There are multiple approaches to constructing a NW crossbar. Undifferentiated NWs can be stamped onto a chip, [11–13], or alternatively, many types of differentiated NWs can be grown off chip, collected in a large ensemble, then deposited fluidically [14,15]. In either approach, a molecular layer is deposited between two layers of parallel NWs. This layer can be comprised of molecular diodes that switch between a low and high resistance in a large electric field [16,17]. A layer of amorphous silicon has also been proposed as a storage medium [18]. Programmable devices that do not behave like diodes (e.g. nanoscale resistors or transistors), have also been considered [19,9]. Some of these alternatives have been compared to diodes with regard to their information storage capacity and ability to control NWs [20,7,21]. When used for information storage, resistors are inferior to other alternatives [20].

Once a NW crossbar is assembled, g OCs and M MWs can be placed along each dimension of the crossbar, using photolithography (see Fig. 1). Although each MW controls (makes nonconducting) a subset of the NWs, these subsets cannot be chosen deterministically. For each NW, we can describe the subset of MWs that control it using a binary M-tuple. We call this a NW **codeword**.

In the case of undifferentiated NWs, two methods have been proposed to control NWs with MWs. The first, the **randomized-contact decoder** (RCD), is analyzed here. A proposed approach for producing an RCD is to randomly deposit nanometer-sized particles between NWs and MWs, making each NW/MW junction controlling with some fixed probability [5,22]. The second method for controlling undifferentiated NWs relies on randomly shifted lithographically-defined regions of high-K dielectric material between NWs and MWs [23,24]. Here again, each MW is made to control some random subset of the NWs. Additional methods for producing decoders are possible using differentiated NWs [25,26].

2.2. Crossbar operation

In a crossbar memory, NWs along each dimension are divided up into g contact groups of N NWs each. NWs in each contact group are connected to a common OC. To use the crossbar as a memory, a voltage is applied to a single contact group of NWs along each dimension of the crossbar. Subsets of MWs along each dimension are then used to address NWs within each of the two groups. This operation can either read or write a single bit to the crosspoints of the NWs being addressed (see Fig. 2). If multiple NWs connected to an OC are addressed by the same set of MWs, it may be acceptable to store the same bit at multiple crosspoints.

In a **write operation**, the diodes at crosspoints are turned on or off by applying a large potential between one or more pairs of orthogonal NWs, by addressing (giving low resistance to) one or more NWs in each dimension. Both ends of the NWs are maintained at the same potential. The polarity of the potential determines the state of a crosspoint and the value written.

In a **read operation**, a smaller voltage is used, allowing the decoder to detect the state of crosspoints. In a read operation each NW is disconnected from one of its ohmic contacts. Current will either flow or not flow through a crosspoint, depending on its state. The amount of current reveals the resistive state of the crosspoints, and thus the value being stored.

Both read and write operations require that the NWs being addressed have a significantly lower resistance than the other NWs in the same contact group. This requirement is formalized at the beginning of Section 4.

2.3. Address translation circuitry

When a memory is supplied with a particular external binary address, **address translation circuitry** (ATC) along each dimension of the crossbar maps that address to a contact group and MW input. This mapping depends on the stochastic assembly of the decoder. To ensure each external address addresses some NW, the ATC must store information about which MWs control which NWs. In the next subsection, we discuss how this information can be obtained. For now, assume it is known and consider the storage overhead required.

Please cite this article in press as: E. Rachlin, J.E. Savage, Nanowire addressing with randomized-contact decoders, Theoretical Computer Science (2008), doi:10.1016/j.tcs.2008.08.011

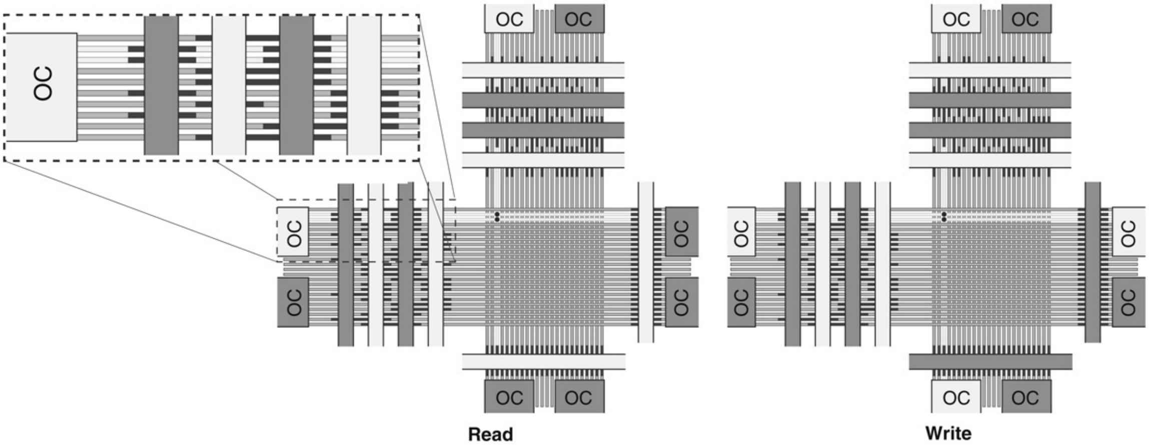

Fig. 2. A crossbar-based memory in which OCs and MWs read and write data to programmable molecules at crosspoints. The darkened segments along each NW indicate lightly doped regions. These regions become nonconducting when the adjacent MW is turned on. In a read operation an OC at each end of a NW is disconnected from ground. Current flows through any conducting NW crosspoints that are addressed by MWs. The amount of current reveals the value stored at the crosspoints. In a write operation, NWs along each dimension apply a larger electric field across their crosspoints. The direction of the field determines the value stored at the crosspoints. In this figure, the same bit of information is stored at two crosspoints.

In order to make address translation circuitry fast, reliable, and easy to manufacture, it may be implemented in CMOS. Any approximation of the area of the memory must take into account not just the area of MWs and ohmic contacts, but also the area used to store physical NW addresses using CMOS. We explicitly model the size of address translation circuitry in [10], but it has received less attention elsewhere. The appendix of [3], also estimates the area required for the ATC, but does so without exploring how different address mapping strategies affect area requirements. The prospect of implementing the ATC using nanoscale storage is considered in [9].

Most previous work on NW decoders has focused on the number of MWs required to control NWs. Although MWs are much wider than NWs, they are still relatively small. In an RCD, however, each NW/MW junction, corresponds to a bit of storage in address translation circuitry. As a result, these bits, when stored in mesoscale devices, collectively take up far more area than the NW/MW junctions.

The ATC must associate a contact group and codeword with each external address. In the worst case, this requires $\log_2 g + M$ bits of storage per NW. In some cases fewer bits are required. For example, if all NWs can be addressed, and the number of NWs per contact group is a power of 2, the high order bits of an external address can be used to index a contact group. The address translation circuitry now requires only M bits of storage for each NW. The way in which external addresses are mapped to NWs is called an **addressing strategy**. Later, several addressing strategies are discussed in detail. As we explain, some addressing strategies require more overall area than others.

2.4. Address discovery

We also explore the problem of testing. In an RCD, each NW has a physical address determined by which MWs control it. Since addresses are randomly generated during decoder assembly, they must be discovered. This is a difficult problem, as some addresses mask others, and faults make test outputs unreliable. We evaluate the effectiveness of several simple testing procedures that do not require read/write operations.

In [10], an efficient testing procedure involving read/write operations was given for differentiated NW decoders. The algorithm's reliance on nanoscale storage devices is a drawback. Read/write operations are relatively time consuming, and possibly faulty. Also, in circuits, they may not be possible at all (as not all NWs will be used to control nanoscale storage devices).

The testing algorithms we consider are only allowed to apply a voltage across a contact group, turn on a subset of the MWs, and observe if any NW remains conducting. This test does not reveal which NW is on, nor does it reveal if multiple NWs are on. Nonetheless, it is sufficiently powerful to determine which subsets of MWs address individual NWs. As it turns out, the algorithm in [10] can also be adapted to this model. Unfortunately it does not work for RCDs. A discovery algorithm for RCDs is given in [5], which we improve upon in Section 7 on this algorithm here. We also improve upon its analysis.

3. Decoding technologies

In this section we describe three types of NW decoder. Each type of decoder can itself be manufactured in multiple ways. As shown in Section 4, however, all three decoders can be modeled in a unified way. Using this model, we analyze the number of MWs required by an RCD in Section 5. In Section 6 we estimate the total amount of area RCDs require.

3.1. The encoded NW decoder

Encoded NW decoders work with two kinds of NWs, modulation-doped NWs [27,28], NWs with sequences of lightly and heavily doped regions, and radially encoded NWs [29], NWs with removable shells. In both cases many NW types are prepared separately, each with a different encoding. When modulation-doped NWs are used, encodings correspond to patterns of lightly and heavily doped regions. When radially encoded NWs are used, encodings correspond to sequences of shells. In either case many NWs of each type are all collected in a large ensemble then deposited onto a chip using fluidic methods that align the NWs in parallel [30].

When MWs are placed across the NWs, any NW/MW junction comprised of a lightly doped region forms a field effect transistor (FET). The application of an immobilizing electric field to the MW causes the resistance of the junction to become high. A NW is addressed by applying fields to all MWs that do not significantly increase its resistance. If doping sequences are properly chosen, only one type of NW will become nonconducting. (See Figs. 1 and 2.) In practice, lightly doped NW regions will not align perfectly with MWs [29]. Consequently, a MW's control over a NW can be ambiguous. Several methods for encoding modulation-doped NWs are studied in [10].

The encoded NW decoder also works with radially encoded NWs, that is, NWs that have shells composed of differentially etchable NWs [29]. There are several ways to control these NWs with MWs. The simplest method uses one MW to control each type of NW. Each NW type is grown with a different sequence of s shells surrounding a lightly doped core. Once NWs have been deposited, a different sequence of s shells are etched away in the space reserved for each MW. Under each MW, only the lightly doped cores of one NW type are exposed, and if shells are sufficiently thick, each MW controls only NWs with exposed cores. Radially encoded NWs do not suffer from misalignment but may require slightly larger radii than modulation-doped NWs.

3.1.1. Known results

In an encoded NW decoder, the number of controllable NWs depends on the number of differently-encoded NW types, C, being used. This, in turn, determines how many MWs, M, are required. In an encoded NW decoder, each NW is equally likely to contain each encoding. NWs with different encodings can be addressed separately. When NWs are encoded using "binary reflected codes" [10], $M = 2\log_2 C$. Using "h-hot codes" [25], M can be reduced to close to $\log_2 C$. Despite using more MWs, binary reflected codes require the same amount of ATC. The main advantage of binary reflected codes is that they allow for "wildcarding", which in turn yields a simple testing algorithm [10]. If the N NWs in a given contact group each have a different encoding, these encodings can be discovered in time $O(NM)$, which is optimal.

The area required for an encoded NW decoder also depends on C. In order for all NWs in a contact group to be addressable with probability $1-\epsilon$, the number of NW types, C, must be at least $N(N-1)/(-2\ln(1-\epsilon))$ [10]. Half of the NWs in a contact group are addressable with probability $1-\epsilon$ if C is at least $\mathrm{e}^{(N-1-2\ln\epsilon)/(N+1)}(N-1)/2$ [10]. It was demonstrated in [10] that requiring half of the NWs be addressable requires significantly less total area than requiring that all NWs be addressable. Additionally, it requires fewer NW encodings be manufactured.

As explained in [10], NWs in an encoded NW decoder are assigned encodings with equal probability. Since each encoding can be addressed separately, the probability that a NW is individually addressable is $(1-1/C)^{N-1}$, and the expected number of individually addressable NWs per contact group is $N(1-1/C)^{N-1}$. This observation allows us to directly apply the derivations of Theorem 5.2 and Corollary 5.2 that appear in Section 5.1. Doing this gives following result, which does not appear elsewhere.

Theorem 3.1. *Let N'_a be the total number of individually addressable NWs in an encoded NW decoder with g contact groups, N NWs per contact group, $N' = gN$ NWs in total and M MWs.*

$$P(N'_a > \kappa N') \geq 1-\epsilon$$

if $\kappa \leq (1-1/C)^{N-1} - \sqrt{-\ln\epsilon/(2g^*)}$ *where* $g^* = g(N/(N-1))^2$.

Also, as mentioned in Section 2.1, it is not always necessary to address NWs individually. If two NWs have the same encoding (see Fig. 2), they can still be addressed collectively and used to store the same bit at multiple NW crosspoints. In an encoded NW decoder, the expected number of different encodings per contact group is $C(1-(1-1/C)^N)$. If addressing groups of NWs is acceptable, the above theorem can be modified.

Theorem 3.2. *Let N'_a be the total number of addresses in an encoded NW decoder with g contact groups, N NWs per contact group, $N' = gN$ NWs in total and M MWs.*

$$P(N'_a > \kappa N') \geq 1-\epsilon$$

if $\kappa \leq (C/N)(1-(1-1/C)^N) - \sqrt{-\ln\epsilon/(2g^*)}$ *where* $g^* = g(N/(N-1))^2$.

3.2. The masked-based decoder

Mask-based decoders [23] work with uniform NWs [13,31]. Lithograpically-defined high-K dielectric rectangles are deposited between NWs and MWs. A rectangle amplifies a MW's electric field, allowing it to increase the resistance of the

Please cite this article in press as: E. Rachlin, J.E. Savage, Nanowire addressing with randomized-contact decoders, Theoretical Computer Science (2008), doi:10.1016/j.tcs.2008.08.011

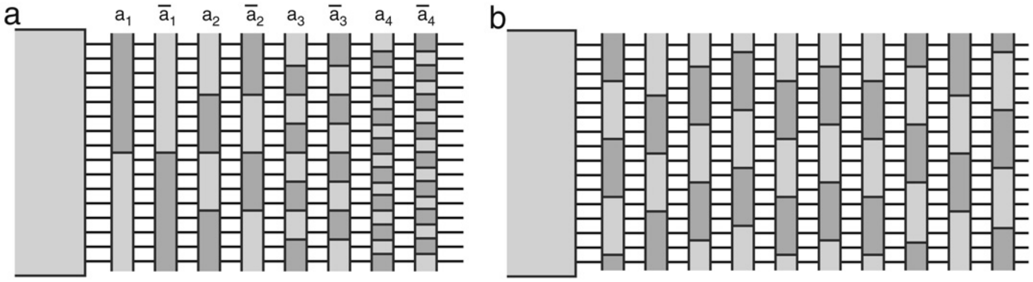

Fig. 3. (a) A masked-based NW decoder in which regions of high-K dielectric allow each MW to control a different subset of NWs. If arbitrarily small high-K dielectric regions could be manufactured and placed with nanoscale precision, $2\log_2(N)$ MWs could be used to address each of N NWs. (b) Since this is not the case, many randomly shifted copies of the smallest manufacturable region can be used to gain control over individual NWs.

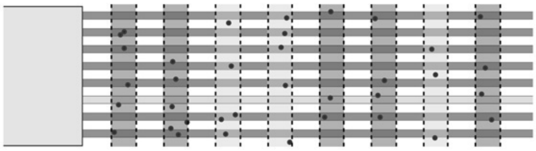

Fig. 4. A randomized-contact decoder in which random particle deposition causes each MW to control certain NWs.

lightly-doped NWs underneath. If rectangles can be as small as the pitch of NWs, they can be used with $M = 2\log_2 N$ MWs to cause all but one NW to have high resistance, as suggested in Fig. 3a. Unfortunately, rectangles cannot be made as small as the pitch of NWs. Instead, many randomly shifted copies of the smallest lithographically-defined rectangles can be deposited on a chip. The natural randomness in their locations provides control over NWs with high probability [23] (see Fig. 3b). One difficulty, however, is that nanoscale misalignment of a high-K dielectric regions can cause a particular MW to only partially turn off a particular NW. A similar problem will arise if the boundary of a high-K dielectric regions is not sufficiently sharp, although [23] appears to indicate that nanoscale transitions between high-K and low-K dielectric regions are feasible.

3.2.1. *Known results*

The number of MWs, M, needed to control N NWs is estimated to be at least six times the number required with an encoded NW decoder [32]. Unless masks can be placed with sub-NW pitch accuracy, the number of MWs required for all N NWs in a contact group to be addressable with probability ϵ is approximately $2(N-1)\ln(2(N-1)/\epsilon)$. Even though M is large, each NW can be addressed using a small number of MWs. As a result, the area required for ATC remains reasonable.

3.3. *The randomized-contact decoder*

A randomized-contact decoder is any decoder in which NW/MW connections can be modeled as independent random variables. In an RCD, a MW provides strong control over a NW with probability p, very weak or no control with probability q and intermediate or ambiguous control with probability $r = 1-(p+q)$. Since this third case models a manufacturing error, we do not assume that $p+q = 1$. This is a very practical generalization of the error-free model given in [5]. In Section 5 we bound the number of MWs M required to tolerate a given error rate.

Williams and Kuekes first proposed the randomized-contact decoder (RCD) in [33]. There is a number of ways an RCD might be produced. One approach is to randomly deposit impurities (such as gold particles) onto undifferentiated NWs (see Fig. 4). Another approach is to randomly deposit small regions of high-K dielectric. An RCD can also be constructed from axially encoded NWs. If many sets of axially encoded NWs are produced with randomly placed lightly doped regions, each NW/MW junction can be treated as an independent random variable. As a result, analysis of RCDs provides bounds that apply to axial (and similarly radial) decoders.

There is also another interesting relationship between RCDs, encoded NW decoders, and masked-based decoders. In a masked-based decoder, there is significant correlation regarding which NWs a given MW controls. In an encoded NW decoder there is correlation between which MWs control a given NW. In an RCD, neither correlation should exist, although in practice a small amount of spacial correlation between NW/MW junctions might be present.

Hogg et al. [5] have explored the conditions under which most of the N NWs in an RCD can be controlled by a set of M MWs. They demonstrate through simulation that when M passes a threshold, which is around $4.8\log_2 N$, the probability that all NWs are addressable grows rapidly as N increases. This is in agreement with our Corollary 5.3.

Their asymptotic analysis does not make explicit the dependence of M on N and the probability ϵ of failing to having all NWs be addressable. It also fails to capture the impact of manufacturing errors. We develop tight bounds for both purposes in Section 5, and do so without the independence approximation used in [5], namely that pairs of NWs can be analyzed

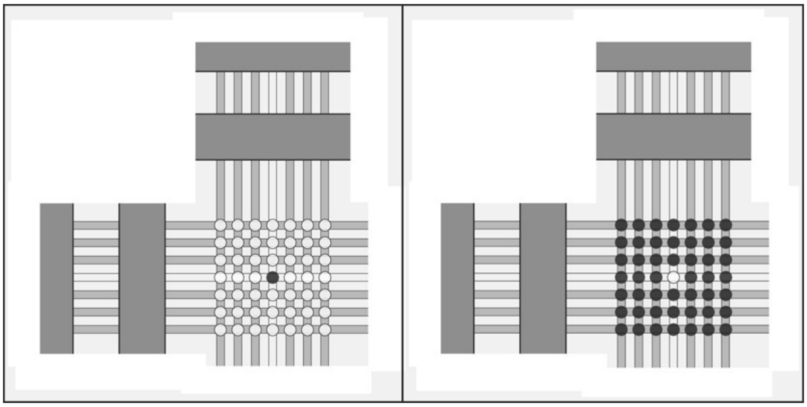

Fig. 5. On the left, the crosspoint being read has a high resistance, but all other crosspoints have a low resistance. On the right, however, the crosspoint being read has a low resistance, but all other crosspoints have a high resistance. To correctly determine the state of the crosspoint, the amount of current flowing from one dimension of the crossbar to the other must be greater on the the right than the left.

independently with regard to whether or not each can be controlled separately. We also give a more careful analysis of the value of M required for at least some fixed fraction of NWs to be addressable.

4. Decoder requirements

In Section 5, we bound the number of MWs, M, required by RCDs with N NWs per OC to control many individual NWs. To derive these bounds, we first define the requirements that decoders must meet. The conditions we obtain in this section apply to other types of decoders as well.

4.1. Nanowire addressing

As explained in Section 1, read/write operations are performed in a NW crossbar-based memory, by employing an address decoder in each dimension of the memory. If each decoder addresses at least D disjoint sets of NWs, they collectively control D^2 disjoint sets of NW crosspoints each of which can store a bit.

Since each of the two decoders is comprised of g contact groups, $D = \sum_{i=1}^{g} D_i$, where D_i is the number of disjoint sets of NWs that can be addressed within the ith contact group.

Let R_i be the resistance of NW $\boldsymbol{n_i}$. When a decoder addresses a set $\mathcal{S}$ of NWs within a single contact group, each NW in $\mathcal{S}$ has a low resistance, while the NWs not in $\mathcal{S}$ have a high resistance. In a **write operation**, every NW in $\mathcal{S}$ must have a much lower resistance than every NW not in $\mathcal{S}$, that is, $\max(R_i \mid \boldsymbol{n_i} \in \mathcal{S}) \ll \min(R_i \mid \boldsymbol{n_i} \notin \mathcal{S})$. This ensures that the bits associated with NWs in $\mathcal{S}$ are written whereas those not in $\mathcal{S}$ are not written. A **read operation** requires that the combined resistance of all NWs in $\mathcal{S}$, R_{IN}, be much less than the combined resistance of NWs not in $\mathcal{S}$, R_{OUT}, that is $1/R_{OUT} \ll 1/R_{IN}$. Consider the two extremes illustrated in Fig. 5. Since the resistance R of a set of n resistances, $R_1, \ldots, R_n$, placed in parallel satisfies $1/R = 1/R_1 + \cdots + 1/R_n$, this is equivalent to $\sum_{\boldsymbol{n_i} \notin \mathcal{S}} 1/R_i \ll \sum_{\boldsymbol{n_i} \in \mathcal{S}} 1/R_i$.

Definition 4.1. A set, $\mathcal{S}$, of NWs is **addressed** if and only if a) every NW not in $\mathcal{S}$ has a resistance that is at least α times that of every NW in $\mathcal{S}$ and b) the combined resistance of all NWs not in $\mathcal{S}$ is at least α times that of the combined resistance of all NWs in $\mathcal{S}$, where $\alpha \gg 1$.

In general, the choice of an actual value of α is application specific. A larger value, for example, would be required to read data from molecular devices with poor on/off ratios. A larger value would also facilitate reading data more quickly or more reliably.

Following the above analysis, if $R_i \le R_L$ when $\boldsymbol{n_i} \in \mathcal{S}$ and $R_i \ge R_H$ when $\boldsymbol{n_i} \notin \mathcal{S}$, the condition on writing is satisfied when $R_H \ge \alpha R_L$ and that on reading is satisfied when $R_H/(N - |\mathcal{S}|) \ge \alpha R_L/|\mathcal{S}|$. This read condition is hardest to meet when $|\mathcal{S}| = 1$ in which case $R_H \ge \alpha(N-1)R_L$. This is clearly stronger than the write condition $R_H \ge \alpha R_L$.

4.2. Resistive and ideal models of control

A NW decoder addresses a set of NWs by applying an electric field to a subset of the MWs. These MWs are said to be **activated**. The set of activated MWs is called an **activation pattern**. A particular activation pattern, $\boldsymbol{a}$, is represented as a binary vector where $a_j = 1$ if and only if the jth MW is activated. Each activated MW increases each NW's resistance by some amount (possibly 0). More formally, NWs behave as follows.

Definition 4.2. In the **resistive model of NW control**, each NW $\boldsymbol{n_i}$ has initial resistance η_i when no MWs are activated. Associated with each NW is a length-M vector of reals, or a **real-valued nanowire codeword**, $\boldsymbol{r^i}$. The jth entry of $\boldsymbol{r^i}$, r_j^i, is the amount by which the jth MW increases the resistance of $\boldsymbol{n_i}$ when activated. When the decoder is supplied with $\boldsymbol{a}$, the resistance of NW $\boldsymbol{n_i}$ is $\eta_i + \boldsymbol{a} \cdot \boldsymbol{r^i}$ where $\boldsymbol{a} \cdot \boldsymbol{r^i}$ is the inner product of activation pattern $\boldsymbol{a}$ and codeword $\boldsymbol{r^i}$.

When the jth MW provides strong control over NW $\boldsymbol{n_i}$, r^i_j is large. r^i_j is small when the jth MW provides weak control over $\boldsymbol{n_i}$. In the ideal case, each r^i_j is either 0 or ∞ and a codeword is associated with each NW. Note that multiple NWs may have the same codeword.

Definition 4.3. In the **ideal model of NW control**, each NW, $\boldsymbol{n_i}$ is assigned a **binary codeword**, $\boldsymbol{c^i}$, where $c^i_j = 1$ if and only if $r^i_j = \infty$. For a particular activation pattern, $\boldsymbol{a}$, $\boldsymbol{a} \cdot \boldsymbol{c^i} > 0$ if and only if $\boldsymbol{a} \cdot \boldsymbol{r^i} = \infty$. A **set $\mathcal{S}$ of NWs is addressed** when $\boldsymbol{a} \cdot \boldsymbol{c^i} = 0$ for NWs in $\mathcal{S}$ and $\boldsymbol{a} \cdot \boldsymbol{c^j} \geq 1$ for NWs not in $\mathcal{S}$.

In either model of control, a set, $\mathcal{S}$, of NWs is considered **addressable** if there is some activation pattern such that $\mathcal{S}$ is addressed. Similarly, a particular NW $\boldsymbol{n_i}$ is **individually addressable** if there is an activation pattern such that $\{\boldsymbol{n_i}\}$ is addressed. A codeword c_i is **individually addressable** if each NW with that codeword is individually addressable.

Notice that in the ideal model of NW control, if a binary codeword is addressable, the NWs with that codeword are addressed by activation pattern $\boldsymbol{a} = \overline{\boldsymbol{c^i}}$. Furthermore, if $\boldsymbol{c^i}$ is not addressable, there is some other codeword $\boldsymbol{c^k}$ such that for each j it is not true that $c^i_j = 0$ and $c^k_j = 1$. This is the mathematical definition of implication; that is, c^k_j implies c^i_j. When this condition holds for all values of j, we say that $\boldsymbol{c^k}$ **implies** $\boldsymbol{c^i}$, and write $\boldsymbol{c^k} \Rightarrow \boldsymbol{c^i}$. The following is immediate.

Lemma 4.1. *In a simple NW decoder in the ideal model of control, a NW codeword $\boldsymbol{c^i}$ is addressable, if and only if, no other codeword that is present implies $\boldsymbol{c^i}$. The decoder can address D disjoint sets of NWs if and only if D distinct NW codewords are addressable.*

4.3. Modeling errors

If each r^i_j takes value $r_{low} = 0$ or $r_{high} = \infty$, each real-valued codeword can be mapped to a binary codeword, which are simple to work with. When r_{low} and r_{high} do not hold these extreme values we map $\boldsymbol{r^i}$ to $\boldsymbol{c^i}$ such that:

- $c^i_j = 0$ if $r^i_j \leq r_{low}$
- $c^i_j = 1$ if $r_{high} \leq r^i_j$
- $c^i_j = e$ if $r_{low} \leq r^i_j \leq r_{high}$, meaning that c^i_j is in error.

Our goal is to choose values for r_{low} and r_{high} so that a set $\mathcal{S}$ of NWs is addressed by an activation pattern $\boldsymbol{a}$ if the following conditions hold:

- for $\boldsymbol{n_i} \in \mathcal{S}$, $c^i_j = 0$ when $a_j = 1$,
- for $\boldsymbol{n_k} \notin \mathcal{S}$, there exists j such that $c^k_j = 1$ and $a_j = 1$.

Consider an activation pattern $\boldsymbol{a}$ that meets these two conditions. Let $r_{base} = \max_i \eta_i$. Observe that every NW in $\mathcal{S}$ has resistance at most $R_L = r_{base} + (M-1)r_{low}$ because at most $M-1$ MWs are activated. Also, note that every NW not in $\mathcal{S}$ has resistance at least $R_H = r_{high}$. From Definition 4.1 and the discussion that follows it is clear that $\mathcal{S}$ is addressed if $R_H \geq \alpha(N-1)R_L$ or $r_{high} \geq \alpha(N-1)(r_{base} + (M-1)r_{low})$. To simplify the discussion, let $r_{low} = cr_{base}$ for some constant $c > 0$. Then, $\mathcal{S}$ is addressed if

$$r_{high} \geq \alpha(N-1)(cM - c + 1)r_{base}$$

where $\alpha \gg 1$.

In the above model with errors we say that NW $\boldsymbol{n_i}$ is addressable if for each NW $\boldsymbol{n_k}$ there is at least one index (MW) j such that $c^i_j = 0$ and $c^k_j = 1$. The ensures that $\boldsymbol{c^i}$ has low resistance while $\boldsymbol{c^k}$ has high resistance. When this condition fails, $\boldsymbol{c^i}$ may still be addressable but this cannot be guaranteed. We say that a **codeword $\boldsymbol{c^i}$ fails to be addressable** if there exists a codeword $\boldsymbol{c^k}$ such that the conditions $c^i_j = 0$ and $c^k_j = 1$ fail to be satisfied for some j. In this case, and by analogy with the ideal model, we say that codeword $\boldsymbol{c^k}$ **possibly implies** $\boldsymbol{c^i}$, denoted $\boldsymbol{c^k} \overset{?}{\Rightarrow} \boldsymbol{c^i}$. If $\boldsymbol{c^k} \overset{?}{\Rightarrow} \boldsymbol{c^i}$, there is no guarantee that $\boldsymbol{n_i}$ can be addressed separately from $\boldsymbol{n_k}$.

Lemma 4.2. *In a simple decoder in the model with errors, a codeword, $\boldsymbol{c^i}$, is addressable if for no other codeword $\boldsymbol{c^k}$ does $\boldsymbol{c^k} \overset{?}{\Rightarrow} \boldsymbol{c^i}$. The decoder can address D disjoint sets of NWs if and only if D distinct NW codewords are addressable.*

If r_{high} is too low or r_{low} is too high to be realized using a particular manufacturing technology, NWs can still be addressed if we set $r_{low} = cr_{base}$ and $r_{high} = (\alpha/d)(N-1)(cM - c + 1)r_{base}$, but instead require that each NW is addressed by an activation pattern that activates at least d high resistance junctions in the other NWs. This ensure that $R_H = dr_{high}$.

It is possible for an RCD to be realized with diodes instead of FETs. The decoder model with errors can also be used in this case to capture diodes with imperfect behavior.

5. Analysis of the RCD

In an RCD, consider a simple decoder, consisting of single contact group with N NWs and M MWs. As mentioned, we assume that NW/MW junctions are controlling (i.e. $c_j^i = 1$) with probability p, noncontrolling (i.e. $c_j^i = 0$) with probability q, and ambiguous (i.e. c_j^i is in error) with probability $r = 1 - p - q$. We also assume that these events are statistically independent and identically distributed.

We now bound N_a, the number of individually addressable NWs in each contact group in terms of M, the number of MWs. Recall that for a NW with codeword $\boldsymbol{c^i}$ to be individually addressable there must be no other codeword $\boldsymbol{c^k}$ such that $\boldsymbol{c^k} \overset{?}{\Rightarrow} \boldsymbol{c^i}$ (see Lemma 4.2).

We take two approaches to deriving bounds on M. First, in Theorem 5.1 we bound the expected value of N_a, $E[N_a]$. This allows us to apply Hoeffding's Inequality and derive a lower bound on M, such that the total number of individually addressable NWs across all g contact groups is some at least a fixed fraction of gN, with probability $1 - \epsilon$.

Second, in Theorem 5.3, we use the principle of inclusion–exclusion, to derive upper and lower bounds on M such that all NWs in all (or almost all) contact groups are independently addressable. The first bound is used to evalute the **Take What You Get** addressing strategy evaluated in Section 6. The second is used to evaluate the **All Wires Addressable** and **Almost All Wires Addressable** strategies.

5.1. Bounds using expectation

We now bound the mean number of individually addressable NWs. We use this to bound the fraction of NWs in a compound RCD that are addressable with high probability.

Theorem 5.1. *In an RCD, let N_a be the number of independently addressable NWs in a contact group with N NWs and M MWs.*

$$N(1 - (N-1)(1-pq)^M) \le E[N_a] \le N(1 - (1-pq)^M).$$

Proof. Let $x_i = 1$ if NW $\boldsymbol{n_i}$ is independently addressable and 0 otherwise. Since $N_a = \sum_{i=1}^N x_i$, $E[N_a] = \sum_{i=1}^N E[x_i]$. Also, since the $\{x_i\}$ are identically distributed 0-1 random variables, $E[N_a] = NE[x_1] = NP(x_1 = 1)$.

Let $E_{k,i}$ be the event that $\boldsymbol{c^k} \overset{?}{\Rightarrow} \boldsymbol{c^i}$. $P(x_1 = 1) = 1 - P(x_1 = 0) = 1 - P(E_{2,1} \cup E_{3,1} \cup \cdots \cup E_{N,1})$. Since $P(E_{2,1}) \le P(E_{2,1} \cup E_{3,1} \cup \cdots \cup E_{N,1}) \le \sum_{k=2}^N P(E_{k,1})$ and $P(E_{2,1}) = P(E_{3,1}) = \cdots = P(E_{N,1})$, $1 - (N-1)P(E_{2,1}) \le P(x_1 = 1) \le 1 - P(E_{2,1})$.

$\boldsymbol{c^2} \overset{?}{\Rightarrow} \boldsymbol{c^1}$ if for all $1 \le j \le M$ it is not the case that both $c_j^1 = 0$ and $c_j^2 = 1$, thus $P(E_{2,1}) = (1-pq)^M$ and $1 - (N-1)(1-pq)^M \le P(x_1 = 1) \le 1 - (1-pq)^M$. ■

Corollary 5.1. *Let $N' = gN$ be the total number of NWs contained in the g contact groups of a RCD, and let N'_a be the number of those NWs that are individually addressable. Then,*

$$N'(1 - (N-1)(1-pq)^M) \le E[N'_a] \le N'(1 - (1-pq)^M).$$

Proof. N'_a is the sum of the number of individually addressable NWs in each contact group. Since each contact group has N NWs, $E[N'_a] = gE[N_a]$. Substituting the bounds from Theorem 5.1 yields the desired result. ■

Let $S = n_1 + n_2 + \cdots + n_t$ be the sum of t independent random variables, where each n_i ranges from a_i to b_i. Hoeffding's Inequality [34] states that

$$P(E[S] - S \ge d) \le \mathrm{e}^{-2d^2/\sum c_i^2}$$

where $c_i = b_i - a_i$, and $d \ge 0$. We use this to bound the total number of independently addressable NWs with high probability.

Theorem 5.2. *Let N'_a be the total number of addressable NWs in an RCD with g contact groups, N NWs per contact group, and $N' = gN$ NWs in total.*

$$P(N'_a \le E[N'_a] - N'k) \le \mathrm{e}^{-2k^2N'N/(N-1)^2} = \mathrm{e}^{-2k^2g^*}$$

for any $k \ge 0$ where $g^* = g(N/(N-1))^2$.

Proof. In Hoeffding's Inequality, let $t = g$, $d = N'k$, $S = N'_a$ and $c_i = (N-1)$. This gives $P(E[N'_a] - N'_a \ge N'k) \le \mathrm{e}^{-2(N'k)^2/g(N-1)^2} = \mathrm{e}^{-2k^2N'N/(N-1)^2}$. We can then rewrite $P(E[N'_a] - N'_a \ge N'k)$ as $P(N'_a \le E[N'_a] - N'k)$. ■

From this we obtain a corollary

Corollary 5.2. *Let N'_a be the total number of addressable NWs in an RCD with g contact groups, N NWs per contact group, $N' = gN$ NWs in total and M MWs.*

$$P(N'_a > \kappa N') \ge 1 - \epsilon$$

if $\kappa \le 1 - \sqrt{-\ln \epsilon/(2g^*)} - (N-1)(1-pq)^M$ where $g^* = g(N/(N-1))^2$.

Proof. From Theorem 5.1 we have $E[N'_a] \geq N'(1-(N-1)(1-pq)^M)$ and by the above theorem,

$$P(N'_a \leq N'(1-(N-1)(1-pq)^M) - N'k) \leq \mathrm{e}^{-2k^2 g^*}.$$

Thus, if $k = (1-(N-1)(1-pq)^M) - \kappa$, then

$$P(N'_a \leq \kappa N') \leq \mathrm{e}^{-2g^*(1-(N-1)(1-pq)^M-\kappa)^2}.$$

Thus, when $\mathrm{e}^{-2g^*(1-(N-1)(1-pq)^M-\kappa)^2} \leq \epsilon$ the desired conclusion follows. This occurs when $\ln \epsilon \geq -2g^*(1-(N-1)(1-pq)^M-\kappa)^2$ or $\sqrt{-\ln \epsilon/(2g^*)} \leq (1-\kappa)-(N-1)(1-pq)^M$. ■

As an example, suppose $p = q = 1/2$, $g = 175$, $N = 8$, $N' = 1400$, $\epsilon = .01$, and $\kappa = .733$. When $M = 13$, $\kappa = .733 \leq 1-\sqrt{-\ln .01/(2*175*(8/7)^2)} - 7*(3/4)^{13}$. Thus at least $\lceil .733*1400 \rceil = 1027$ NWs are addressable with probability .99.

If errors occur, that is, when $p+q < 1$, but g is held fixed, M must increase to keep κ constant. For example, if $pq = .2$ rather than $pq = .25$ in the error-free case, M must grow by a factor of $\ln(4/3)/\ln(5/4) = 1.29$. If $pq = .1$, the factor is $\ln(4/3)/\ln(10/9) = 2.73$. Even for relatively high error rates, M is not prohibitively large.

5.2. Bounds using inclusion/exclusion

In this section we derive bounds on the number of MWs required for all NWs to be individually addressable with high probability.

Theorem 5.3. *In an RCD, let Γ be the probability that M NWs fail to control all N NWs in a single contact group. Γ satisfies the following bounds*

$$Q(1-Q/2) - \Delta \leq \Gamma \leq Q \tag{1}$$

where $Q = N(N-1)\mu_1^M$ and $\Delta = 2N(N-1)(N-2)\left(\mu_3^M + \mu_5^M - 2\mu_1^{2M}\right)$ and $\mu_1 = (1-pq)$, $\mu_3 = (1-pq(p+2q))$, and $\mu_5 = (1-pq(2p+q))$.

Proof. See Appendix. ■

This theorem implies upper and lower bounds on M in terms of N and Γ. For the cases examined below when $p = q$ and Γ is small, these bounds are tight, meaning the upper and lower bounds they give on M agree. Slightly weaker but simpler bounds are given in the following corollary, in which upper and lower bounds on M differ by $\ln(2)/\ln(1-pq)$.

Corollary 5.3. *In an RCD, the minimum value of M such that all N NWs in a contact group are individually addressable with probability $1-\epsilon$ satisfies the following.*

$$\frac{\ln(N(N-1)/2\epsilon)}{-\ln(1-pq)} \leq M \leq \frac{\ln(N(N-1)/\epsilon)}{-\ln(1-pq)}$$

where the lower bound holds when $\epsilon \leq .05$ and the actual minimum value of M is itself at least $(1-pq)/(pq\min(p,q))$.

Proof. The upper bound follows from (1). For the lower bound, assume $Q \leq 0.1$, which implies that $M \geq \ln(10N(N-1))/(-\ln \mu_1)$. This is less than $\ln(N(N-1)/2\epsilon/(-\ln(1-pq))$ when $\epsilon \leq .05$. In Δ drop the last term and replace $\mu_3^M + \mu_5^M$ by $2\max(\mu_3,\mu_5)^M$. Since $\mu_3 = \mu_1 - pq^2$ and $\mu_5 = \mu_1 - p^2 q$, $\max(\mu_3,\mu_5) = \mu_1(1-\min(pq^2,p^2q)/\mu_1)$. The lower bound on Γ becomes $\Gamma \geq Q(.95 - 4N(1-pq\min(p,q)/\mu_1)^M)$. Using the inequality $(1-x)^n \leq 1-nx$, the lower bound is at least $Q/2$ if $M \geq (1-.45/4N)(1-pq)/(pq\min(p,q))$ which is less than $(1-pq)/(pq\min(p,q))$. ■

In this corollary, when $\epsilon \leq .05$, the second condition associated with our lower always holds when p and q are fairly close to 1/2. To see why, consider the extreme case when $N = 2$. Here the minimum value of M such that neither NW implies the other must satisfy $(1-pq)^M \leq \epsilon$, or equivalently $M \geq \ln(\epsilon)/\ln(1-pq)$. It is easy to verify numerically that $\ln(.05)/\ln(1-pq) \geq (1-pq)/(pq\min(p,q))$ when $pq \geq .21$.

Corollary 5.4. *In an RCD with N' NWs divided into g contact groups, all NWs are independently addressable with probability $(1-\epsilon)$ if*

$$M \geq \ln(N'((N'/g)-1)/\epsilon)/(-\ln(1-pq)).$$

Proof. Let δ be the probability of failure of all NWs in a contact group to be individually addressable. Then, the probability that one or more contact groups fails to have all its NWs be individually addressable is at most $g\delta$. If $g\delta \leq \epsilon$, the probability that all N' NWs are addressable is at least $1-\epsilon$. We use the upper bound on M given in Corollary 5.3 when N is replaced by N'/g and ϵ by ϵ/g. ■

Please cite this article in press as: E. Rachlin, J.E. Savage, Nanowire addressing with randomized-contact decoders, Theoretical Computer Science (2008), doi:10.1016/j.tcs.2008.08.011

When $N' = 1024, g = 128$ and $M \geq 47$, all N' NWs will be individually addressable with probability 0.99 or better. In fact, evaluating Theorem 5.3 numerically shows this threshold value of M to be exact. These parameters apply to the **All Wires Addressable** addressing strategy in which every NW address is used.

The number of MWs is reduced if we do not require that all NWs in each contact group be individually addressable. We illustrate this with an example. Corollary 5.3 says that a failure rate of at most $\epsilon = .01$ can be achieved with a simple RCD when $p = q = .5$ and $N = 8$ if $M \geq 30$. (As above, this threshold value of M is exact.) The number of individually addressable NWs in each contact group is statistically independent. If all N NWs in a particular contact group are individually addressable with probability $1 - \epsilon$, the probability that f or fewer contact groups fail to have all NWs addressable is $\phi(\epsilon, f, g) = \sum_{i=0}^{f} \binom{g}{i} \epsilon^i (1-\epsilon)^{g-i}$. Let $\epsilon = .01, g = 133$ and $f = 5$. Because $\phi(.01, 5, 133) \geq .99$, at least 128 of $g = 133$ contact groups have all NWs addressable with probability 0.99.

In summary, when $M = 30, g = 133$, and $N = 8 * 133 = 1064, N'_a = 8 * 128 = 1,024$ NWs are individually addressable with probability 0.99. These parameters apply to the **Almost All Wires Addressable** addressing strategy in which almost every NW address is used.

As discussed at the end Section 5.1, manufacturing errors only increase the number of required MWs by a small constant factor.

6. Addressing strategies

We now use the bounds on M to estimate the total area required for a crossbar-based memory that uses RCDs. As explained in Section 2.3, this area estimate depends not just on the number of MWs used, but also on the size of an ATC. In this section we consider three addressing strategies, that is, ways of using an ATC to map an external binary address $\boldsymbol{E}$ of $b = |\boldsymbol{E}|$ bits to an internal NW address consisting of a contact group σ and an activation pattern $\boldsymbol{a}$ on M MWs.

All Wires Addressable: Here we choose M so that, with probability $(1-\epsilon)$, all NWs in every contact group are individually addressable. If we assume that the number of NWs in each contact group is 2^k, we can simply use the first $b - k$ bits of $\boldsymbol{E}$ to select σ. This fixed mapping does not depend on the particular NW codewords that are present, although the mapping of $\boldsymbol{E}$ to $\boldsymbol{a}$ does. To execute the second mapping, the ATC stores each NW codeword that is present in a lookup table. This requires $N'_a M$ bits of storage where N'_a is the number of addressable NWs in the decoder.

All Wires Almost Always Addressable: Here we choose M so that with probability $(1 - \epsilon)$, all NWs in nearly all contact groups are addressable. Contact groups in which not all NWs are addressable are not used. Since the particular contact groups that are not used will vary from decoder to decoder, the ATC cannot use a fixed mapping from $\boldsymbol{E}$ to contact groups σ. Instead, a lookup table is used to obtain an integer to be added to the first $b - k$ bits of $\boldsymbol{E}$ so that it corresponds to the proper contact group. Let g be the number of contact groups and g' be the number for which all NWs are addressable. Then $g - g'$ is an upper bound on the values in the table. We also use a lookup table to map $\boldsymbol{E}$ to $\boldsymbol{a}$. The two tables combined require approximately $g' \lceil \log_2 g - g' \rceil + N'_a M$ bits.

Take What You Get: Here we choose M so that a fixed fraction of the NWs are individually addressable. In this case, some contact groups may have all NWs addressable, but some may not. Since the number of addressable NWs per contact group varies, we can no longer map fixed blocks of binary memory addresses to a particular contact group. Instead, we store a value of σ and $\boldsymbol{a}$ for each addressable NW. This requires $N'_a(\lceil \log_2 g \rceil + M)$ bits.

6.1. Area estimate

To estimate the total area, A_T, required to produce a crossbar memory using each of the three strategies, we use the approach of [10] and write:

$$A_T \approx 2\chi\beta + 2\lambda_{meso}^2 g \lceil \log_2 g \rceil + (\lambda_{meso} M + \lambda_{nano} N')^2.$$

Here λ_{meso} and λ_{nano} denote the pitch of MWs and NWs respectively, that is, the center-to-center distance between wires. Also, χ denotes the area of a mesoscale memory cell, and β denotes the number bits stored in each dimension of the crossbar's ATC. Thus, $2\chi\beta$ approximates the amount of programmable storage required, $2\lambda_{meso}^2 g \lceil \log_2 g \rceil$ approximates the area required to implement a standard demultiplexer used to activate contact groups, and $(\lambda_{meso} M + \lambda_{nano} N')^2$ approximates the area occupied by the NW crossbar.

6.2. Comparison of strategies

To compare the three addressing strategies, we estimate their area when used to produce a memory with a given storage capacity. In our comparison, we fix ϵ, the probability of failure, and N/g, the number of NWs per contact group. Given these values, we would ideally like to also fix N'_a, the number of addressable NWs along each dimension of the crossbar, then estimate A_T for all three strategies. Unfortunately, for a given strategy, it is difficult to choose M and N' to yield an exact value for N'_a, but in all three cases we show that about 1024 NWs are individually addressable along each dimension.

To compare the strategies, we consider the case when $p = q = 1/2$ and use the numerical results given above.

- **All Wires Addressable**:
  Here $M = 47$, $g = 128$, and $N' = N'_a = 1024$ with probability at least .99. The ATC requires $\beta = N'_a M = 47{,}990$ bits. This gives
  $$A_T \approx 95{,}982\chi + \lambda^2_{meso}1792 + (\lambda_{meso}49 + \lambda_{nano}1600)^2.$$
- **All Wires Almost Always Addressable**:
  Here $M = 30$, $g = 133$, and $N' = 1064$ yields $N'_a = 1024$ and $g' = 128$ with probability at least .99. The ATC requires $\beta = g'\lceil \log g - g' \rceil + N'_A M = 31{,}104$ bits. This gives
  $$A_T \approx 62{,}208\chi + 1877\lambda^2_{meso}+(\lambda_{meso}30 + \lambda_{nano}1064)^2.$$
- **Take What You Get**:
  Here $M = 13$, $g = 175$ and $N' = 1400$, yields N'_a of 1027 with probability at least .99. The ATC requires $\beta = N'_a(\lceil \log g \rceil + M) = 21{,}567$ bits. This gives
  $$A_T \approx 43{,}134\chi + 2800\lambda^2_{meso} + (\lambda_{meso}13 + \lambda_{nano}1400)^2.$$

Since the parameter χ, the area of a mesoscale memory unit, will be many times λ^2_{meso}, and it is expected that $\lambda_{meso} \geq 10\lambda_{nano}$, the **Take What You Get** addressing strategy is clearly best.

7. Codeword discovery

Although required of all NW decoders, codeword discovery has received significantly less attention than NW addressing. In this section, we consider several codeword discovery algorithms that do not require the addition of specialized testing circuitry. We make three assumptions: (a) the ideal decoder model applies so no NW/MW junctions are in error; (b) arbitrary subsets of MWs can be activated; and (c) for any subset of MWs the total amount of current flowing across all NWs in a contact group can be measured, but not with high precision. In Section 7.5 we re-examine (a).

Even when many or all NWs are individually addressable, their codewords (or at least some portion of their codewords) must be discovered to properly configure the ATC. It is not feasible to individually probe each NW/MW junction.

In the ideal model if all MWs in a contact group are activated, and all NWs are controlled by at least one MW, no current will flow. As MWs are turned off one by one, one or more NWs will become conducting. At this point, current is detected. In theory, accurate current measurements and knowledge of NW resistances could allow a testing procedure to estimate how many NWs are conducting, but we avoid this assumption.

In [5] it is assumed that one can distinguish how between all NWs being off, one NW being addressed, and two or more NWs being addressed. We avoid this assumption and assume only the ability to distinguish between all NWs being off and at least on NW being on. Since α in Definition 4.1 is much greater than 1, this is a very reasonable assumption. It is already met, for example, by circuitry used to read data from a crossbar-based memory.

We also examine the number of tests a discovery algorithm must perform. In doing so, we point out an important flaw in the less rigorous analysis given in [5].

7.1. Exhaustive search

The simplest codeword discovery algorithm is exhaustive search. For each contact group, we determine whether current flows for every possible MW activation pattern. The outputs of all 2^M tests are reviewed offline to determine which codewords are present on individually addressable NWs.

Suppose codeword $\boldsymbol{c^i}$ is present on the ith NW. Activation pattern $\boldsymbol{a} = \overline{\boldsymbol{c^i}}$ turns on $\boldsymbol{n_i}$ and turns off all other NWs. Also, any other activation pattern, $\boldsymbol{a}'$, that turns on a strictly larger subset of MWs turns off all NWs. For this reason we call $\boldsymbol{a}$ **maximal**.

An activation pattern $\boldsymbol{a}$ is maximal if and only if $\boldsymbol{c^i} = \bar{\mathbf{a}}$ is individually addressable. Once exhaustive testing is complete, the set of maximal activation patterns can be identified.

7.2. Parallel exhaustive search

The runtime of this algorithm is exponential in M, but as shown in the previous section M may be relatively small. In our analysis of the "Take What You Get" addressing strategy, we demonstrated through analysis that a $M = 13$ suffices. Smaller values of M are also possible if one is willing to tolerate a smaller fraction of addressable NWs.

The exponential running time of exhaustive search can be amortized across contact groups, if all contact groups can be tested in parallel. If current measurements for each contact group can be taken simultaneously, each of the algorithm's 2^M tests can be performed on all contact groups at once.

In an exhaustive search, every possible activation pattern is tested. A more efficient search algorithm would be adaptive. It would use the outcome of previous tests, to determine which activation pattern to apply next. For certain values of M and

g, however, parallel exhaustive search is superior to any adaptive search procedure in which contact groups are tested one at a time.

Suppose all contact groups can be tested in parallel when $M = 13$, $N = 8$, $g = 175$ and $p = q = 1/2$, the conditions on the **Take What You Get** strategy discussed in the previous section. The number of tests per contact group is $2^M/175 < 47$. We show that more tests are required by any adaptive discovery algorithm operating on contact groups one at a time using tests with binary outcomes (e.g. the current measured is "high" or "low").

An adaptive discovery procedure must produce the codeword for each individually addressable NW in a contact group. As shown in Theorem 5.1, the expected number of addressable NWs in a contact group is at least $N(1-(N-1)(1-pq)^M) = 1 - 7(3/4)13 > 6.6$. This indicates that at least six NWs are addressable at least 1/2 the time. (Given that $N = 8$, if less than six NWs are addressable half the time, the average number of addressable NWs is at most $(5 + 8)/2 = 6.5$ because at most five NWs are addressable half the time and at most eight NWs the rest of time.) There are 2^{MN} assignments of M-bit codewords to N NWs. We call these codes. Since all assignments are equally likely, at least $(1/2)2^{MN}$ of these have six individually addressable NWs. The codewords of these NWs will be produced by a discovery algorithm.

Let σ be the maximum number of codes containing any fixed set of six, seven or eight individually addressable codewords. When an adaptive discovery algorithm produces six or more codewords as output, one of at most σ codes is present in the contact group. If eight codewords are produced, one of 8! codes is present. If seven codewords are produced, one of at most $7!(8)2^M$ codes is present. These codes contain all 7! permutations of the codewords and eight locations for the remaining codeword, which takes at most 2^M values. Finally, if six codewords are produced, the number of associated codes is at most $6!\binom{8}{2}2^{2M}$. Since the last case yields the most codes, $\sigma \leq 6!\binom{8}{2}2^{2M}$.

It follows that any discovery algorithm must be able to identify at least $(1/2)2^{MN}/\sigma$ codes. Since it is assumed that tests produce binary outcomes, the number of testing steps for an adaptive algorithm must be at least $T = \log_2[(1/2)2^{MN}/\sigma = MN - 1 - 2M - \log_2(6! * 28)]$. When $M = 13$ and $N = 8$, $T = 67$. Thus, an adaptive algorithm that examines one contact group at a time will need to perform at least 67 tests per group.

7.3. Randomized codeword discovery

For large values of M exhaustive search is prohibitively slow. In this regime adaptive algorithms need to be explored. Also, if multiple searches cannot be run in parallel, an adaptive algorithm will always be faster. In this section we consider a simple adaptive algorithm and examine its runtime. A less efficient version of this algorithm appeared in [5]. As we explain, however, its analysis was based on faulty assumptions.

The goal of our algorithm is to discover the maximal activation patterns that address codewords. The algorithm, sketched below, chooses a random permutation of MWs, π, and activates the MWs in order specified by this permutation until no current is produced. When current is turned off, the last MW to be turned on is deactivated, and the process continues.

```
procedure Discover_Codewords
  $\pi = RandomPermutation(MW_1, MW_2, \ldots, MW_M)$
  for $i = 1$ to $M$ do
      Activate $\pi(i)$
      if All NWs are turned off, then deactivate MW $\pi(i)$
```

After each execution of this procedure, a maximal activation pattern is identified. Its complement yields the discovered codeword. For ease of simulation, it is convenient to note that the discovered codeword is the codeword that comes first when all codewords are sorted lexicographically according to π.

Each execution of the discovery procedure requires M tests. After each test, some codeword is discovered. The total time required for codeword discovery thus depends on the relative likelihood of discovering each codeword. If all codewords are equally likely to be discovered the well-known coupon collector problem, stated in Section 8, shows that approximately $N_a \log(N_a/\epsilon)$ executions are required to discover all N_a individually addressable codewords with probability $(1 - \epsilon)$.

As an optimization, we note that it is not actually necessary to activate subsets of MWs when they do not turn off all of the codewords that have already been discovered. This observation was also made in [35], which evaluates a similar codeword discovery algorithm through simulation.

This is the faulty assumption made in [5]. In fact, experiments indicate that for small values or medium-sized values of M, some codewords will often be much less likely to be discovered than others. For example, when M is 30, all NWs in a contact group of $N = 8$ NWs are addressable with very high probability. If all NWs were equally likely to be discovered, $N \log(N/.01) = 69$ executions are required with probability .01. Our simulations reported in Section 8.2 show this values to be approximately 270. When M is 100, however, the value shrinks to 72.

The reason for this discrepancy is that, when M is small, some NWs that are addressable are much less likely to be discovered than others. For example, more than 1/10 of the time there was at least one NW that had only a 1/70 chance being discovered on each run of the algorithm. For intuition as to why this occurs, consider the following four codewords: $\mathbf{c^1} = 111100000000$, $\mathbf{c^2} = 000011110000$, $\mathbf{c^3} = 000000001111$, $\mathbf{c^4} = 011101110111$. By symmetry, $\mathbf{c^1}$, $\mathbf{c^2}$ and $\mathbf{c^3}$ are equally likely to be discovered, but $\mathbf{c^4}$ can only be discovered if at least two of MW_1, MW_5 and MW_9 are activated before any of the other MWs. This observation reveals that $\mathbf{c^4}$ is discovered is with probability $3/12 * 2/12 * 1/4 = 1/96$, where

Please cite this article in press as: E. Rachlin, J.E. Savage, Nanowire addressing with randomized-contact decoders, Theoretical Computer Science (2008), doi:10.1016/j.tcs.2008.08.011

as all other codewords are discovered with probability $(1 - 1/96)/3 = 95/288$. When M is small, these sorts of extreme examples are much more likely to occur.

As M increases the probability of discovering a NW address approaches $1/N$. This points to an interesting trade-off not just between the number of MWs and the number of addressable NWs, but the number of NW codewords that can be quickly discovered. When the number of MWs is in an intermediate range, adding additional MWs may actually increase the speed with which codewords are discovered.

7.4. *Possible extensions*

Our codeword discovery algorithm does not require specialized testing circuitry, or the ability to measure current in individual NWs. This makes our algorithm highly practical. However, the algorithm can be improved if the testing is done in the context of a memory. Consider testing a horizontal contact group. First activate all the NWs in one vertical contact group to make contacts (i.e. write 1's) at the intersections of NWs in the two groups. When testing the horizontal contact group, measure current using the vertical contact group. After discovering a maximal activation pattern, use the corresponding codeword to open the contacts (i.e. write 0's) at any intersections formed from horizontal NWs with that codeword. This ensures that the codeword will not be discovered again. Using the discovery procedure and this method of eliminating previously discovered NWs, all NW address will be discovered. It will then be necessary to analyze the addresses to find the individually addressable NWs.

The main disadvantage of this modified randomized algorithm is the read/write requirement. In a circuitry (as opposed to a memory) molecular switches may not be present. Furthermore, read/write operations may be faulty and slow. If the number of MWs is small, it may still be faster to implement the parallel exhaustive search algorithm described above.

A codeword discovery algorithm that uses read/write operations for encoded NW decoders was described in [10]. In fact, the algorithm can be adapted to find codewords without the use of read/write operations, but the algorithm will not work with the randomly generated codewords found in an RCD. The read/write discovery algorithm described above uses at most M tests per NW and works for both encoded NW and RCD decoders.

7.5. *Coping with errors*

An even more important extension to codeword discovery, is learning to cope with errors. For simplicity, we only consider the exhaustive search case. In the case of errors, it is no longer possible to describe certain activations patterns as maximal, because it is no longer reasonable to treat the output of each test as binary. An error can produce an intermediate level of current flow along a NW.

We have already shown that, for sufficiently large M, all NWs are addressable with high probability. This result holds, even if errors occur. If N NWs are addressable, there is some activation pattern that causes each of these NWs to conduct while all other $N - 1$ NWs are turned off. Furthermore, if we combine any two of these activation patterns, all N NWs will be turned off. If we have a pair of activation patterns that satisfy this property, we call them *disjoint*.

If N NWs are addressable and we exhaustively test all 2^M activation patterns, we can then identify N patterns that are all disjoint. One method for identifying these patterns from the testing data, is to construct a graph G with vertex associated with each pattern that causes at least one NW to conduct. The testing data is then used to place an edge between between any two vertices that correspond to disjoint activation patterns. A clique of N vertices in G, corresponds to a set of N addresses that each address a distinct NW. Exhaustive testing is currently the only known method for discovery of codewords in the presence of errors.

8. Analysis of the discovery procedure

We now bound the number of tests required by the randomized codeword discovery algorithm given in Section 7.3. The number of runs needed to ensure that, with high probability, all codewords are discovered is modeled by the **coupon collection problem with non-uniform probabilities**. We now state bounds on the number of trials that are needed to collect all coupons with probability at least ϵ. They can be derived using established methods [32].

Lemma 8.1. *Consider the collection of N coupons in which each coupon is collected with probability at least u. The expected number of trials to collect all N coupons is at most* $1 + \frac{1}{u}H_{N-1}$.

Proof. The average time to collect N coupons is $\overline{T} = \sum_{i=1}^{N} \overline{x}_i$ where x_i is the time to collect the ith coupon. Let $\{p_1, p_2, \ldots, p_N\}$ be the probabilities of collecting the coupons and let $j_1, j_2, \ldots, j_N$ be the order in which they are collected. Because the first new coupon is collected on the first trial, $\overline{x}_1 = 1$. For $i \geq 2$ the probability distribution for x_i is geometric with probability $(1 - (p_{j_1} + p_{j_2} + \cdots + p_{j_{i-1}}))$. Thus, $\overline{x}_i = 1/(1 - (p_{j_1} + p_{j_2} + \cdots + p_{j_{i-1}}))$. It follows that $\overline{T}$ is maximized by maximizing $(p_{j_1} + p_{j_2} + \cdots + p_{j_{N-1}})$. Since $p_N \geq u$, $\overline{T}$ is largest when $p_N = u$. Similarly, the remaining terms in the sum for $\overline{T}$ are maximized by setting $p_j = u$ for $2 \leq j \leq N$, which provides the desired result. ■

8.1. The likelihood of generating discoverable codewords

We show that the probability $Q(u)$ of choosing a code such that each codeword can be discovered by Discover_Codewords with probability at least u is close to one when M, the number of MWs, is proportional to $\log_2 N$ where N is the number of NWs.

The codeword associated with the ith NW is defined in Section 5 as $\mathbf{c^i} = \{c_1^i, \ldots, c_M^i\}$ where $c_j^i = 1\,(0)$ if the jth NW/MW junction in the ith codeword is controlling (noncontrolling). If $c_j^i = e$, the control of the junction is ambiguous. A code $\mathbf{C}$ is a collection of codewords. We let p, q, and $r = 1 - p - q$ be the probabilities that $c_j^i = 1, 0$ and e, respectively. In this section we consider codeword discovery when there is no ambiguity, that is, when $r = 0$.

We consider codes containing codewords that are all about equally likely, $\mathcal{C}_0$, and $\overline{\mathcal{C}_0}$, the complement of that set. $\mathcal{C}_0$ is defined in terms of $B^i(0) = \{i \mid c_j^i = 0\}$, the indices for which $\mathbf{c^i}$ has value 0, and $B^i(0) \bigcap B^k(0)$, the indices for which $\mathbf{c^i}$ and $\mathbf{c^k}$ have 0s in common locations. The first condition on $\mathcal{C}_0$ is that each codeword must satisfy is the following:

$$|B^i(0)| \geq Mq - k_1. \tag{2}$$

This ensures that each codeword has approximately the average number of 1s and 0s. The second, given below, also ensures that pairs of codewords are typical, namely, that the number of 0s they have in common is approximately the average.

$$|B^i(0) \cap B^k(0)| \leq Mq^2 + k_2. \tag{3}$$

Let $D(\mathbf{C}, u)$ be the event that the each codeword in code $\mathbf{C}$ is discovered with probability at least u. It follows that probability $Q(u)$ satisfies the following bound.

$$\begin{aligned} Q(u) &= \sum_{C \in \mathcal{C}} P(D(\mathbf{C}, u) \cap E(\mathbf{C})) \\ &\geq \sum_{C \in \mathcal{C}_0} P(D(\mathbf{C}, u) \mid E(\mathbf{C}))P(E(\mathbf{C})). \end{aligned}$$

Let $D^i(\mathbf{C})$ be the event that codeword $\mathbf{c^i}$ in a code $\mathbf{C}$ is discovered and let $P(D^i(\mathbf{C}))$ be the probability of this event. If $P(D^i(\mathbf{C})) \geq u$ for all words in codes in $\mathcal{C}_0$, then $P(D(\mathbf{C}, u) \mid E(\mathbf{C})) = 1$. Below we derive such a bound, the proof of which in the Appendix.

Theorem 8.1. *If $\mathbf{C}$ is a code in the ensemble $\mathcal{C}_0$, the probability $P(D^i(\mathbf{C}))$ that the ith codeword in $\mathbf{C}$ is discovered is given below when $\gamma = (Mq - k_1)/(Mq^2 + k_2) > 1$ and $M \geq (\ln 4N)/\ln \gamma + 1$.*

$$P(D^i(\mathbf{C})) \geq \frac{1}{2}(4N)^{-\frac{1}{\gamma}(\ln q - k_1/M)} \mathrm{e}^{-\left(\frac{(\ln 4N)^2}{\gamma^2 M}\right)\left(\frac{1}{q - k_1/M} - 1\right)}. \tag{4}$$

If M is large relative to k_1 and $(\ln 4N)^2/\gamma^2$, the lower bound approaches $P(D^i) \geq \frac{1}{2}(4N)^{-\frac{\ln q}{\gamma}}$. When M is also large relative to k_2, γ approaches $1/q$ and the limiting value of $P(D^i)$ becomes $\frac{1}{2}(4N)^{-q \ln q}$ or $\frac{1}{2}(4N)^{-.35}$ when $q = 1/2$.

When u satisfies $P(D^i(\mathbf{C})) \geq u$, $Q(u)$ has the following lower bound.

$$Q(u) \geq \sum_{C \in \mathcal{C}_0} P(E(\mathbf{C})) = P(\mathcal{C}_0).$$

But $P(\mathcal{C}_0) = 1 - P(\overline{\mathcal{C}}_0)$ where $P(\overline{\mathcal{C}}_0)$ is the probability that either $|B^i(0)| \geq Mq - k_1$ is violated for one of the N codewords or $|B^i(0) \cap B^k(0)| \leq Mq^2 + k_2$ is violated for one of the $\binom{N}{2}$ pairs of codewords. Thus, $P(\overline{\mathcal{C}}_0)$ satisfies the following.

$$\begin{aligned} P(\overline{\mathcal{C}}_0) &\leq NP\left(|B^i(0)| < Mq - k_1\right) \qquad (5) \\ &\quad + \binom{N}{2} P\left(|B^i(0) \cap B^k(0)| > Mq^2 + k_2\right). \qquad (6) \end{aligned}$$

Bits in codewords are i.i.d 0-1 random variables in which 0s (1s)occur with probability q ($p = 1 - q$). A 0 occurs in a given position in two codewords simultaneously with probability q^2. We use the Chernoff bound cited below to bound these probabilities [34, p. 66].

Theorem 8.2. *Let X be the sum of n independent and identically distributed random variables, with mean μ. Then, the following holds when $k \leq \mu$.*

$$P(X \leq \mu - k) \leq \mathrm{e}^{-k^2/2\mu}.$$

Please cite this article in press as: E. Rachlin, J.E. Savage, Nanowire addressing with randomized-contact decoders, Theoretical Computer Science (2008), doi:10.1016/j.tcs.2008.08.011

Corollary 8.1. *Let Y be the sum of n independent and identically distributed random variables, with mean μ and maximum value P. Then, the following holds when $k \le P - \mu$.*

$$P(Y \ \ge \mu + k) \le \mathrm{e}^{-k^2/2(P-\mu)}.$$

To obtain the corollary let $X = P - Y$. Then, $P(Y \ge \mu + k) = P(X \le (P - \mu) - k)$ where $P - \mu$ is the average of X.

When applied these bounds are applied to the events in question the following holds.

$$P\left(|B^i(0)| < Mq - k_1\right) \le \mathrm{e}^{-k_1^2/(2Mq)}$$

$$P\left(|B^i(0) \cap B^k(0)| > Mq^2 + k_2\right) \le \mathrm{e}^{-k_2^2/(2M(1-q^2))}.$$

Thus, $P(\overline{\mathfrak{C}}_0)$ satisfies the following bound.

$$P(\overline{\mathfrak{C}}_0) \le N\mathrm{e}^{-k_1^2/(2Mq)} + \binom{N}{2}\mathrm{e}^{-k_2^2/(2M(1-q^2))}. \tag{7}$$

Summarizing, we have the following result concerning the performance of the Discover_Codewords procedure.

Theorem 8.3. *Consider RCD codes consisting of N codewords of length M, in which 0s (1s) occur independently with probability q (p). If $\gamma = (Mq - k_1)/(Mq^2 + k_2) > 1$ and u satisfies*

$$u \le \frac{1}{2}(4N)^{-\frac{1}{\gamma}(\ln q - k_1/M)}\mathrm{e}^{-\left(\frac{(\ln 4N)^2}{\gamma^2 M}\right)\left(\frac{1}{q-k_1/M}-1\right)}$$

$k_1 \ge \sqrt{2Mq\ln(2N/(1 - Q(u)))}$, and $k_2 \ge \sqrt{2M(1 - q^2)\ln(N^2/(1 - Q(u)))}$, then the probability that a code is selected for which Discover_Codewords discovers each codeword with probability at least u, is at least $Q(u)$.

Proof. The results follow from Theorem 8.1 and (5) if k_1 and k_2 are chosen so that $N\mathrm{e}^{-k_1^2/(2Mq)} \le (1 - Q(u))/2$ and $N^2\mathrm{e}^{-k_2^2/(2M(1-q^2)} \le (1 - Q(u))$. ■

When $Q(u) = .99$ and $q = 0.5$, $k_1 \ge \sqrt{M(\ln N + 5.3)}$ and $k_2 \ge \sqrt{M(2\ln N + 4.6)}$.

8.2. Experimental results

The bound on the probability $P(\overline{\mathfrak{C}}_0)$ given in (5) and the bound on u in Theorem 8.3 have been calculated for a variety of values of k_1 and k_2. Choosing $k_1 = 10$ and $k_2 = 12$ when $N = 8$ yields $Q(u) = 0.93$ when $u = .005$, that is, for 93% of the RNC codes each codeword is discovered with probability of at least 1/2 of one percent. In practice, a much higher value of u is achieved for a given value of $Q(u)$.

We simulated 2000 runs in Matlab of the Discover_Codewords procedure on each of 5000 randomly generated, error-free contact groups. Each contact group had 8 NWs. In Fig. 6 we plot the cumulative distribution of the number of runs before all individually addressable codewords were discovered for both 30 and 100 MWs. We also plot the cumulative distribution of the fraction of runs that discovered whichever codeword was discovered least often, that is, an empirical estimate of u.

As discussed at the end of Section 7.3 as the number of MWs increases from 30 to 100, the minimum probability with which a codeword is discovered increases. Similarly, the number of runs to discover nearly all codewords with high probability decreases as M increases. In fact, approximately 270 runs are needed to discover all codewords with probability 0.99 when $M = 30$ and approximately 72 when $M = 100$. The latter number is very close to the number predicted, when all codewords are equally likely to be discovered using the coupon collector problem.

9. Conclusions

We have shown analytically, that stochastically assembled RCD decoders can control large number of NWs using a smal number of MWs. Our results are obtained using a simple, but broadly applicable model, that quantifies the requirements a decoder must meet to address sets of NWs. Our model is robust in the sense that it takes manufacturing defects into account.

By applying our model to RCDs, we obtain tight bounds on the probability that M MWs control all N NWs in all, or almost all contact groups. We also bound the total fraction of individually addressable NWs. Both bounds allow us to investigate multiple addressing strategies for implementing a NW crossbar-based memories. We conclude that "Take What You Get" addressing strategy uses the smallest area to individually address at least 1024 NWs along each dimension of the crossbar. What is more, only 13 MWs are required.

We have also considered the problem of codeword discovery. We have given the first formal analysis of several codeword discovery algorithms. As explained, parallel exhaustive search may be preferable to an adaptive search algorithm that must test only one contact group at a time. When an adaptive algorithm is used, there appears to be a tradeoff between the number of MWs and its runtime. The specific algorithm we consider can be modeled as a coupon collection problem, where some coupons are more likely to be collected than others.

Although RCDs have not yet been demonstrated experimentally, we believe they are a very promising NW decoder technology. Their ability to cope with manufacturing errors, as well as to be produced using a range of manufacturing

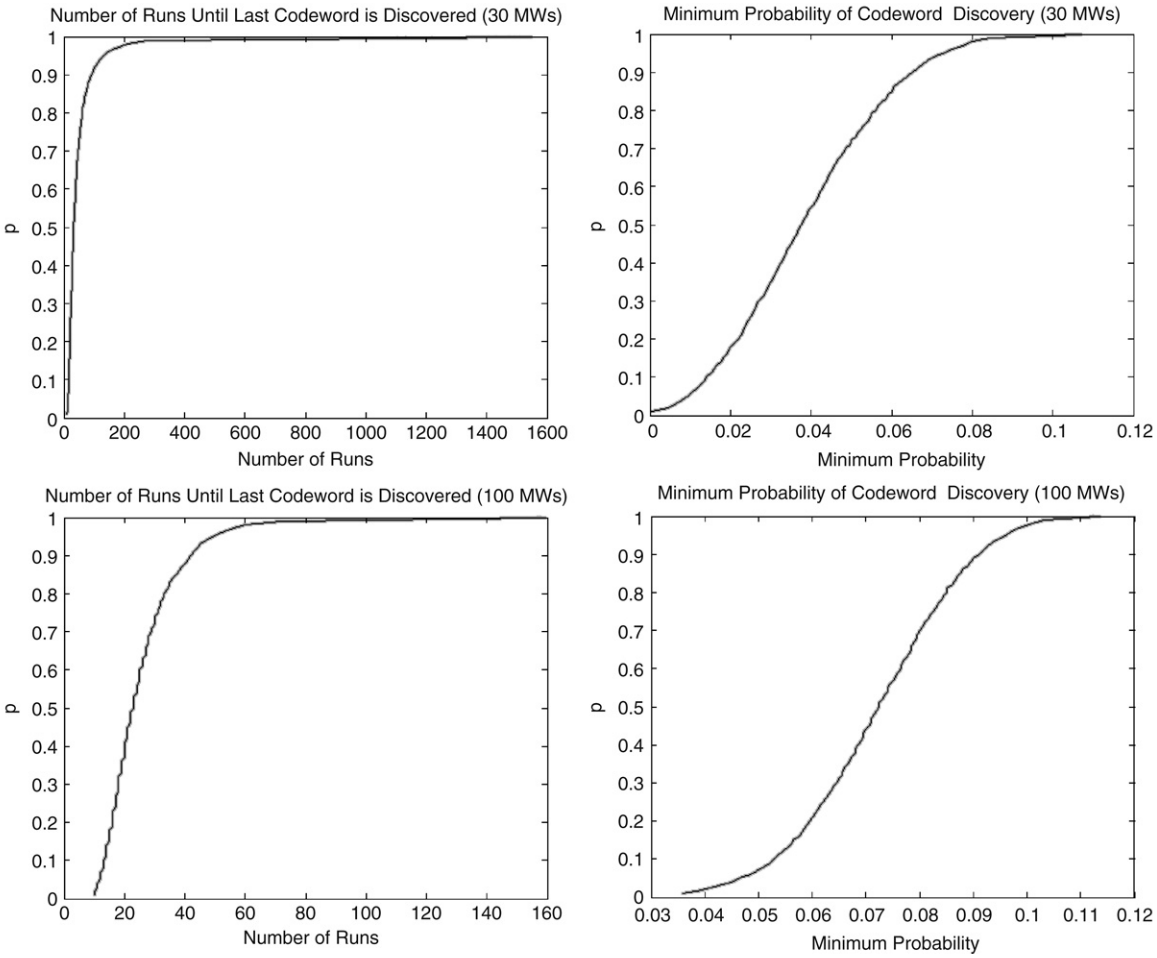

Fig. 6. Shown are empirical plots obtained by simulating 2,000 runs of Discover_Codewords on 5000 randomly generated, error-free contact groups each of which has 8 NWs. The plots show the cumulative distribution of the number of runs before all individually addressable codewords are discovered, and the fraction of the runs in which the least frequently discovered codeword were found.

methods, makes them practically appealing. Their highly stochastic assembly represents a significant departure from current lithographic manufacturing techniques. They serve as an important example of how nanoscale architectures can cope with randomness and still achieve significant gains over CMOS.

Acknowledgements

The authors acknowledge support by the National Science Foundation under NSF Grant CCF-0403674. A preliminary version of this paper but without the material on codeword discovery appeared in the Proceedings of ICCAD 2006.

Appendix

Theorem 5.3. *In an RCD, let Γ be the probability that M NWs fail to control all N NWs in a single contact group. Γ satisfies the following bounds*

$$Q(1 - Q/2) - \Delta \le \Gamma \le Q$$

where $Q = N(N-1)\mu_1^M$ *and* $\Delta = 2N(N-1)(N-2)\left(\mu_3^M + \mu_5^M - 2\mu_1^{2M}\right)$ *and* $\mu_1 = (1 - pq)$, $\mu_3 = (1 - pq(p + 2q))$, *and* $\mu_5 = (1 - pq(2p + q))$.

Proof. The principle of inclusion–exclusion states that $P(E_1 \cup E_0 \cup \cdots \cup E_n) \le \sum_{i=1}^{n} P(E_i)$ and $\sum_{i=1}^{n} P(E_i) - 1/2 \sum_{i \neq j} P(E_i \cap E_j) \le P(E_1 \cup E_0 \cup \cdots \cup E_n)$.

Let $E_{a,b}$ (where $a \neq b$) be the event that $\boldsymbol{c^a} \overset{?}{\Rightarrow} \boldsymbol{c^b}$. By Lemma 4.2, we know that all NWs are independently addressable, if no event $E_{a,b}$ occurs. The probability that not all NWs are individually addressable, Γ, satisfies $\Gamma = P(\cup_{(a,b)} E_{a,b})$. We use inclusion–exclusion to bound Γ.

Please cite this article in press as: E. Rachlin, J.E. Savage, Nanowire addressing with randomized-contact decoders, Theoretical Computer Science (2008), doi:10.1016/j.tcs.2008.08.011

As established in the proof of Theorem 5.1, $P(E_{a,b}) = \mu_1^M$ where $\mu_1 = (1 - pq)$. Let $Q = \sum_{a \neq b} P(E_{a,b})$. Since a and b can both take values from 1 to N, $Q = N(N-1)\mu_1^M$. We must now bound $\sum_{(a,b)\neq(c,d)} P(E_{a,b} \cap E_{c,d})$. Here $1 \leq a, b, c, d \leq N$ provided that $(a, b) \neq (c, d)$, i.e., either $a \neq b$ or $c \neq d$ or both.

To compute $P(E_{a,b} \cap E_{c,d})$, we consider 3 cases:

In case (1), a, b, c and d are all different. There are $N(N-1)(N-2)(N-3)$ ways of selecting them. Since $E_{a,b}$ and $E_{c,d}$ are independent, $P(E_{a,b} \cap E_{c,d}) = P(E_{a,b})P(E_{c,d}) = \mu_1^{2M}$.

In case (2), two of the four variables are equal. Here, either $a = c$, $a = d$, $b = c$ or $b = d$. As stated earlier, we do not allow $a = b$ or $c = d$. There are $N(N-1)(N-2)$ ways to choose indices in each case. These cases are considered below.

In case (3), there are only two different values for a, b, c, and d. Since $(a, b) \neq (c, d)$, $a = d$ and $b = c$, which can occur in $N(N-1)$ ways. Here $P(E_{a,b} \cap E_{c,d}) = P(E_{a,b} \cap E_{b,a})$, which is the probability that, for no j is c_j^a = 0 and c_j^b = 1, or c_j^a = 1 and c_j^b = 0. So $P(E_{a,b} \cap E_{b,a}) = \mu_2^M$ where $\mu_2 = (1 - 2pq)$.

Returning to case 2, we have four subcases to consider.

Let $F_{a,b}(m)$ be the event that $c_m^a = 0$ and $c_m^b = 1$. Let $E_{a,b}(m)$ be the complement of $F_{a,b}(m)$. Since the probability of $F_{a,b}(m)$ is pq, it follows that the probability of event $E_{a,b}(m)$ is $P(E_{a,b}(m)) = 1 - pq$. Since the event $E_{a,b}$ is $\prod_m E_{a,b}(m)$, $P(E_{a,b}) = \mu_1^M$.

(1) $\boldsymbol{n_a} = \boldsymbol{n_c}$. $F_{a,b}(m) \cup F_{a,d}(m)$ occurs only if $(c_{a,m}, c_{b,m}, c_{d,m})$ assumes the value (0, 1, 0), (0, 1, 1), or (0, 0, 1). Thus, $P(F_{a,b}(m) \cup F_{a,d}(m)) = pq(p + 2q)$ and $P(E_{a,b} \cap E_{c,d}) = \mu_3^M$ where $\mu_3 = (1 - pq(p + 2q))$.
(2) $\boldsymbol{n_a} = \boldsymbol{n_d}$. Thus, $F_{a,b}(m) \cup F_{c,a}(m)$ occurs if $(c_{a,m}, c_{b,m}, c_{c,m})$ assumes the value (0, 1, 0), (0, 1, 1), (1, 1, 0), or (1, 0, 0). Thus, $P(F_{a,b}(m) \cup F_{c,a}(m)) = 2pq(p + q)$ and $P(E_{a,b}) \cap E_{c,a}) = \mu_4^M$ where $\mu_4 = (1 - 2pq(p + q))$.
(3) $\boldsymbol{n_b} = \boldsymbol{n_c}$. Thus, $F_{a,b}(m) \cup F_{b,d}(m)$ occurs if $(c_{a,m}, c_{b,m}, c_{d,m})$ assumes the value (0, 1, 0), (0, 1, 1), (0, 0, 1), or (1, 0, 1). Thus, $P(F_{a,b}(m) \cup F_{c,b}(m)) = 2pq(p + q)$ and $P(E_{a,b}) \cap E_{b,d}) = \mu_4^M$.
(4) $\boldsymbol{n_b} = \boldsymbol{n_d}$. Thus, $F_{a,b}(m) \cup F_{c,b}(m)$ occurs if $(c_{a,m}, c_{b,m}, c_{c,m})$ assumes the value (0, 1, 0), (0, 1, 1), or (1, 1, 0). Thus, $P(F_{a,b}(m) \cup F_{c,b}(m)) = pq(2p + q)$ and $P(E_{a,b}) \cap E_{c,a}) = \mu_5^M$ where $\mu_5 = (1 - pq(2p + q))$.

Let $D = \sum_{(a,b)\neq(c,d)} P(E_{a,b} \cap E_{c,d})$. Then,

$$D/(N(N-1)) = (N-2)(N-3)\mu_1^{2M} + \mu_2^M + (N-2)\left(\mu_3^M + 2\mu_4^M + \mu_5^M\right)$$

where $\mu_1 = (1 - pq)$, $\mu_2 = (1 - 2pq)$, $\mu_3 = (1 - pq(p + 2q))$, $\mu_4 = (1 - 2pq(p + q))$, and $\mu_5 = (1 - pq(2p + q))$. The behavior of D is dominated by the largest term μ_i^M. Note that $\mu_2 \leq \mu_1^2$ and $\mu_4 \leq \min(\mu_3, \mu_5) \leq (\mu_3 + \mu_5)/2$. It follows that $(N-2)(N-3)\mu_1^{2M} + \mu_2^M \leq N(N-1)\mu_1^{2M}$ and $(\mu_3^M + 2\mu_4^M + \mu_5^M) \leq 2(\mu_3^M + \mu_5^M)$. Thus, D satisfies the following bound.

$$D \leq Q^2 + 2N(N-1)(N-2)\left(\mu_3^M + \mu_5^M - 2\mu_1^{2M}\right).$$

The lower bound to Γ follows directly from the above. ∎

Theorem 8.1. *If* $\mathbf{C}$ *is a code in the ensemble* $\mathfrak{C}_0$, *the probability* $P(D^i(\mathbf{C}))$ *that the ith codeword in* $\mathbf{C}$ *is discovered is given below when* $\gamma = (Mq - k_1)/(Mq^2 + k_2) > 1$ *and* $M \geq (\ln 4N)/\ln\gamma + 1$.

$$P(D^i(\mathbf{C})) \;\geq\; \frac{1}{2}(4N)^{-\frac{1}{\gamma}(\ln q - k_1/M)} e^{-\left(\frac{(\ln 4N)^2}{\gamma^2 M}\right)\left(\frac{1}{q - k_1/M} - 1\right)}. \tag{8}$$

If M *is large relative to* k_1 *and* $(\ln 4N)^2/\gamma^2$, *the lower bound approaches* $P(D^i) \geq \frac{1}{2}(4N)^{-\frac{\ln q}{\gamma}}$. *When* M *is also large relative to* k_2, γ *approaches* $1/q$ *and the limiting value of* $P(D^i)$ *becomes* $\frac{1}{2}(4N)^{-q\ln q}$ *or* $\frac{1}{2}(4N)^{-.35}$ *when* $q = 1/2$.

Proof. The event D^i that codeword $\boldsymbol{c^i}$ in a code $\mathbf{C}$ is discovered is the event that for some $1 \leq \rho \leq M$ after ρ MWs are activated $\boldsymbol{c^i}$ remains on and for no other codeword $\boldsymbol{c^k}$ do both $\boldsymbol{c^i}$ and $\boldsymbol{c^k}$ remain on. Let $E(\boldsymbol{c^i}, \rho)$ be the event that $\boldsymbol{c^i}$ remains on after the ρth trial. Then,

$$D^i = \bigcup_{1\leq\rho\leq M}\left(E(\boldsymbol{c^i}, \rho) - \bigcup_{k\neq i} E(\boldsymbol{c^i}, \rho) \cap E(\boldsymbol{c^k}, \rho)\right).$$

It follows that the probability that $\boldsymbol{c^i}$ is discovered, $P(D^i)$, satisfies the following bound

$$\begin{aligned} P(D^i) \;&\geq\; \max_{1\leq\rho\leq M}\left(P(E(\boldsymbol{c^i}, \rho)) - \sum_{k\neq i} P(E(\boldsymbol{c^i}, \rho) \cap E(\boldsymbol{c^k}, \rho))\right) \\ &=\; \max_{1\leq\rho\leq M}\left(P(E(\boldsymbol{c^i}, \rho))\left(1 - \sum_{k\neq i} R(i, j, \rho)\right)\right) \end{aligned}$$

where

$$R(i, j, \rho) \;= P(E(\boldsymbol{c^i}, \rho) \cap E(\boldsymbol{c^k}, \rho))/P(E(\boldsymbol{c^i}, \rho)). \tag{9}$$

Please cite this article in press as: E. Rachlin, J.E. Savage, Nanowire addressing with randomized-contact decoders, Theoretical Computer Science (2008), doi:10.1016/j.tcs.2008.08.011

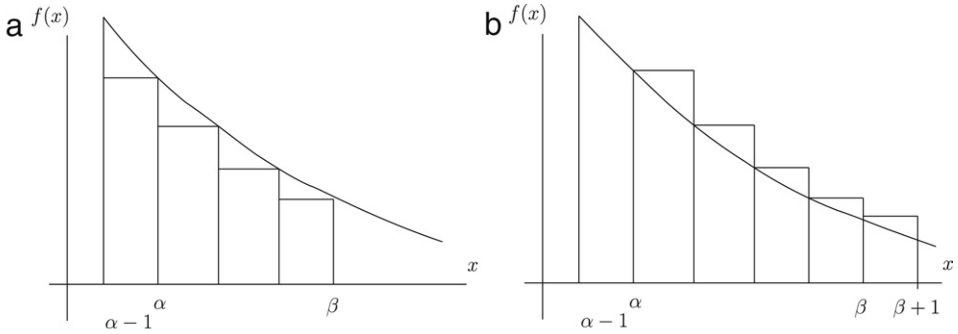

Fig. 7. When $f(x)$ is decreasing, $\sum_{\alpha}^{\beta} f(x) \le f(\alpha) + \int_{\alpha}^{\beta} f(x)\,\mathrm{d}x$, as suggested in (a). Also, $\sum_{\alpha}^{\beta} f(x) \ge \int_{\alpha}^{\beta} f(x)\,\mathrm{d}x + f(\beta)$, as suggested in (b).

Later, we show that for all i and j, $R(i, j, \rho) \le R_0(z)$, a function independent of i and j for $z = \rho - 1$, from which we have the following.

$$P(D^i) \ \ge \max_{1\le\rho\le M} \left(P(E(\boldsymbol{c^i}, \rho))\,(1 - (N-1)R_0(z))\right). \tag{10}$$

Because all permutations under Discover_Codewords are equally likely, $P(E(\boldsymbol{c^i}, \rho))$ is the probability that one of the $|B^i(0)|$ 0s of $\boldsymbol{c^i}$ is activated by the first MW, which occurs with probability $|B^i(0)|/M$, that one of the remaining $|B^i(0)| - 1$ 0s is activated by the second MW, which occurs with probability $(|B^i(0)| - 1)/(M - 1)$, etc, giving the following expression for $P(E(\boldsymbol{c^i}, \rho))$.

$$P(E(\boldsymbol{c^i}, \rho)) = \prod_{0\le t\le\rho-1} \frac{|B^i(0)| - t}{M - t}.$$

Similarly,

$$P(E(\boldsymbol{c^i}, \rho) \cap E(\boldsymbol{c^k}, \rho)) = \prod_{0\le t\le\rho-1} \frac{|B^i(0) \cap B^k(0)| - t}{M - t}.$$

It follows that $R(i, j, z)$ has the following form.

$$R(i, j, \rho) \ = \prod_{0\le t\le\rho-1} \frac{|B^i(0) \cap B^k(0)| - t}{|B^i(0)| - t}. \tag{11}$$

Fig. 7 illustrates the use of integration to obtain bounds on decreasing functions such as $\ln f(m, z)$ where $f(m, z) = \prod_{0\le t\le z}(m - t)$. Bounds are stated in terms of $h(\alpha, \beta, m) = \int_{\alpha}^{\beta} \ln(m - x)\,\mathrm{d}x = (y \ln y - y)\,\big|_{m-\beta}^{m-\alpha}$. The following is immediate.

$$\mathrm{e}^{h(\alpha,\beta,m)} = \frac{(m-\alpha)^{(m-\alpha)}}{(m-\beta)^{(m-\beta)}}\mathrm{e}^{(\alpha-\beta)}.$$

Consequently $f(m, z)$ satisfies the following bounds

$$\left(1 - \frac{z}{m}\right) F(m, z) \le f(m, z) \ \le F(m, z)$$

where

$$F(m, z) = m^z \left(1 - \frac{z}{m}\right)^{-(m-z)} \mathrm{e}^{-z}.$$

To simplify these bounds, consider the function $g(x) = (1 - x)\ln(1 - x)$. Because its Taylor series expansion is $g(x) = -x + \sum_{j=2} \frac{x^j}{j(j-1)}$, $g(x) \ge -x + x^2/2$. Also, because $\ln(1 - x) \le -x$, $g(x) \le -x + x^2$. These results imply the following bounds on $F(m, \rho)$.

$$m^z \mathrm{e}^{-(z)^2/(2m)} \le F(m, z) \le m^z \mathrm{e}^{-z^2/m}.$$

This leads to the following bounds on $f(m, z)$.

$$\left(1 - \frac{z}{m}\right) m^z \mathrm{e}^{-(z)^2/(2m)} \le F(m, z) \le m^z \mathrm{e}^{-z^2/m}.$$

Using the assumptions of (2) and (3) and these results provides the following upper bound on $R(i, j, \rho)$ where $z = \rho - 1$.

$$R(i, j, \rho) \ \le R_0(z) = \left(1 - \frac{z}{Mq - k_1}\right)^{-1} \left(\frac{Mq^2 + k_2}{Mq - k_1}\right)^z \mathrm{e}^{z^2\left(\frac{1}{Mq-k_1} - \frac{1}{Mq^2-k_2}\right)}.$$

Please cite this article in press as: E. Rachlin, J.E. Savage, Nanowire addressing with randomized-contact decoders, Theoretical Computer Science (2008), doi:10.1016/j.tcs.2008.08.011

Now let $\gamma = (Mq - k_1)/(Mq^2 + k_2)$ for $\gamma > 1$. Also, let $z \leq (Mq - k_1)/2$ so that $(1 - z/(Mq - k_1))^{-1} \leq 2$ and the following bound holds.

$$R_0(z) \leq 2\gamma^{-z} e^{-z^2 \frac{\gamma - 1}{Mq - k_1}}.$$

In (10) $R_0(z)$ is multiplied by $(N - 1)$. For the bound to be meaningful, $(N - 1)R_0(z)$ must be less than 1. Thus, we let $NR_0(z) \leq 1/2$ and solve the value of ρ for which this is true. But this holds when the following condition applies.

$$z^2 \left(\frac{\gamma - 1}{Mq - k_1}\right) + z \ln \gamma - \ln 4N \geq 0$$

z has two solutions, one positive and one negative. The positive solution, which is shown below, is the only viable alternative.

$$z_+ = \ln \gamma \left(-1 + \sqrt{1 + \frac{4(\gamma - 1)}{(Mq - k_1)} \frac{\ln 4N}{(\ln \gamma)^2}}\right) \frac{Mq - k_1}{2(\gamma - 1)}.$$

Using $\sqrt{1 + x} \leq 1 + x/2$, it follows that $NR_0(z) \leq 1/2$ is satisfied if $z \geq z_+ = (\ln 4N)/\ln \gamma$ when $\gamma = \frac{Mq - k_1}{Mq^2 + k_2} > 1$. Under these conditions $P(D^i)$ satisfies the following condition when the maximization is restricted to $\rho \geq \rho_0 = (\ln 4N)/\ln \gamma + 1$.

$$P(D^i) \ \geq \ \max_{\rho_0 \leq \rho \leq M} \left(P(E(\boldsymbol{c^i}, \rho))/2\right). \tag{12}$$

Because $\rho \leq M$, the condition $\rho \geq (\ln 4N)/\ln \gamma + 1$ implies that M must satisfy $M \geq (\ln 4N)/\ln \gamma + 1$.

To finish this analysis, we derive a lower bound to $P(E(\boldsymbol{c^i}, \rho))$.

$$\begin{aligned} P(E(\boldsymbol{c^i}, \rho)) \ &\geq \ \prod_{0 \leq t \leq \rho - 1} \frac{Mq - k_1 - t}{M - t} \\ &\geq \left(1 - \frac{\rho - 1}{Mq - k_1}\right)\left(q - \frac{k_1}{M}\right)^{\rho - 1} e^{-(\rho-1)^2 (\frac{1}{Mq - k_1} - \frac{1}{M})} \\ &\geq \frac{1}{2}\left(q - \frac{k_1}{M}\right)^{\rho - 1} e^{-(\rho-1)^2 (\frac{1}{Mq - k_1} - \frac{1}{M})}. \end{aligned}$$

The latter holds, because $\rho - 1 \leq (Mq - k_1)/2$. Because $q - k_1/M < 1$, this lower bound decreases with increasing ρ. Thus, we evaluate the lower bound to $P(D^i)$ at $\rho = \rho_0$, as shown below.

$$P(D^i) \ \geq \ \frac{1}{2}(4N)^{-\frac{1}{\gamma}(\ln q - k_1/M)} e^{-\left(\frac{(\ln 4N)^2}{\gamma^2 M}\right)\left(\frac{1}{q - k_1/M} - 1\right)}. \tag{13}$$

When M is large, relative to $(\ln 4N)^2/\gamma^2$, the lower bound approaches the following when $\gamma = (Mq - k_1)/(Mq^2 + k_2) > 1$ and k_1/M approaches 0.

$$P(D^i) \ \geq \ \frac{1}{2}(4N)^{-\frac{\ln q}{\gamma}}. \tag{14}$$

As k_2/M approaches 0, γ approaches $1/q$ and the limiting value of $P(D^i)$ is $\frac{1}{2}(4N)^{-q \ln q}$ or $\frac{1}{2}(4N)^{-.35}$ when $q = 1/2$. ■

References

[1] G.Y. Jung, S. Ganapathiappan, A.A. Ohlberg, L. Olynick, Y. Chen, William M. Tong, R. Stanley Williams, Fabrication of a 34 × 34 crossbar structure at 50 nm half-pitch by UV-based nanoimprint lithography, Nano Letters 4 (7) (2004) 1225–1229.

[2] P.J. Kuekes, R.S. Williams, J.R. Heath, Molecular wire crossbar memory, US Patent Number 6,128,214, October 3, 2000.

[3] André DeHon, Seth Copen Goldstein, Philip Kuekes, Patrick Lincoln, Nonphotolithographic nanoscale memory density prospects, IEEE Transactions on Nanotechnology 4 (2) (2005) 215–228.

[4] André DeHon, Nanowire-based programmable architectures, Journal on Emerging Technologies in Computing Systems 1 (2) (2005) 109–162.

[5] Tad Hogg, Yong Chen, Philip J. Kuekes, Assembling nanoscale circuits with randomized connections, IEEE Transactions on Nanotechnology 5 (2) (2006) 110–122.

[6] Philip J. Kuekes, Warren Robinett, Gabriel Seroussi, R. Stanley Williams, Defect-tolerant interconnect to nanoelectronic circuits, Nanotechnology 16 (2005) 869–882.

[7] G.S. Snider, W. Robinett, Crossbar demultiplexers for nanoelectronics based on n-hot codes, IEEE Transactions on Nanotechnology 4 (2) (2005) 249–254.

[8] Eric Rachlin, John E. Savage, Nanowire addressing in the face of uncertainty, in: J. Becker, A. Herkersdorf, A. Mukherjee, A. Smailagic (Eds.), Procs. 2006 Int. Symp. on VLSI, Karlsruhe, Germany, March 2–3, 2006, pp. 225–230.

[9] A. DeHon, Deterministic addressing of nanoscale devices assembled at sublithographic pitches, IEEE Transactions on Nanotechnology 4 (6) (2005) 681–687.

[10] Benjamin Gojman, Eric Rachlin, John E. Savage, Evaluation of design strategies for stochastically assembled nanoarray memories, Journal on Emerging Technologies in Computing Systems 1 (2) (2005) 73–108.

[11] S.Y. Chou, P.R. Krauss, P.J. Renstrom, Imprint lithography with 25-nanometer resolution, Science 272 (1996) 85–87.

Please cite this article in press as: E. Rachlin, J.E. Savage, Nanowire addressing with randomized-contact decoders, Theoretical Computer Science (2008), doi:10.1016/j.tcs.2008.08.011

[12] Yong Chen, Gun-Young Jung, Doublas A.A. Ohlberg, Xuema Li, Duncan R. Stewart, Jon O. Jeppeson, Kent A. Nielson, J. Fraser Stoddart, R. Stanley Williams, Nanoscale molecular-switch crossbar circuits, Nanotechnology 14 (2003) 462–468.

[13] Nicholas A. Melosh, Akram Boukai, Frederic Diana, Brian Gerardot, Antonio Badolato, Pierre M. Petroff, James R. Heath, Ultrahigh-density nanowire lattices and circuits, Science 300 (2003) 112–115.

[14] Dongmok Whang, Song Jin, Charles M. Lieber, Nanolithography using hierarchically assembled nanowire masks, Nano Letters 3 (7) (2003) 951–954.

[15] Zhaohui Zhong, Deli Wang, Yi Cui, Marc W. Bockrath, Charles M. Lieber, Nanowire crossbar arrays as address decoders for integrated nanosystems, Science 302 (2003) 1377–1379.

[16] C.P. Collier, E.W. Wong, M. Belohradský, F.M. Raymo, J.F. Stoddart, P.J. Kuekes, R.S. Williams, J.R. Heath, Electronically configurable molecular-based logic gates, Science 285 (1999) 391–394.

[17] Charles P. Collier, Gunter Mattersteig, Eric W. Wong, Yi Luo, Kristen Beverly, José Sampaio, Francisco Raymo, J. Fraser Stoddart, James R. Heath, A [2]catenate-based solid state electronically reconfigurable switch, Science 290 (2000) 1172–1175.

[18] K. Gopalakrishnan, R.S. Shenoy, C. Rettner, R. King, Y. Zhang, B. Kurdi, L.D. Bozano, J.J. Welser, M.B. Rothwell, M. Jurich, M.I. Sanchez, M. Hernandez, P.M. Rice, W.P. Risk, H.K. Wickramasinghe, The micro to nano addressing block, in: Procs. IEEE Int. Electron Devices Mtng., December 2005.

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

[20] P.P. Sotiriadis, Information capacity of nanowire crossbar switching networks, IEEE Transactions on Information Theory 52 (7) (2006) 3019–3032.

[21] W. Robinett, G.S. Snider, D.R. Stewart, J. Straznicky, R. Williams, Demultiplexers for nanoelectronics constructed from nonlinear tunneling resistors, IEEE Transactions on Nanotechnology 6 (3) (2007) 280–290.

[22] Eric Rachlin, John E. Savage, Nanowire addressing with randomized-contact decoders, in: Procs. ICCAD, November, 2006.

[23] Robert Beckman, Ezekiel Johnston-Halperin, Yi Luo, Jonathan E. Green, James R. Heath, Bridging dimensions: Demultiplexing ultrahigh-density nanowire circuits, Science 310 (2005) 465–468.

[24] Eric Rachlin, John E. Savage, Analysis of mask-based nanowire decoders, IEEE Transactions on Computers 57 (2) (2008) 175–187.

[25] André DeHon, Array-based architecture for FET-based, nanoscale electronics, IEEE Transactions on Nanotechnology 2 (1) (2003) 23–32.

[26] Chen Yang, Zhaohui Zhon, Charles M. Lieber, Encoding electronic properties by synthesis of axial modulation-doped silicon nanowires, Science 310 (2005) 1304–1307.

[27] André DeHon, Patrick Lincoln, John E. Savage, Stochastic assembly of sublithographic nanoscale interfaces, IEEE Transactions on Nanotechnology 2 (3) (2003) 165–174.

[28] Benjamin Gojman, Eric Rachlin, John E. Savage, Decoding of stochastically assembled nanoarrays, in: Procs 2004 Int. Symp. on VLSI, Lafayette, LA, February 19–20, 2004.

[29] John E. Savage, Eric Rachlin, André DeHon, Charles M. Lieber, Yue Wu, Radial addressing of nanowires, Journal on Emerging Technologies in Computing Systems 2 (2) (2006) 129–154.

[30] Franklin Kim, Serena Kwan, Jennifer Akana, Peidong Yang, Langmuir–Blodgett nanorod assembly, Journal of the American Chemical Society 123 (18) (2001) 4360–4361.

[31] E. Johnston-Halperin, R. Beckman, Y. Luo, N. Melosh, J. Green, J.R. Heath, Fabrication of conducting silicon nanowire arrays, Journal of Applied Physics Letters 96 (10) (2004) 5921–5923.

[32] Eric Rachlin, John E. Savage, Benjamin Gojman, Analysis of a mask-based nanowire decoder, in: Procs 2005 Int. Symp. on VLSI, Tampa, FL, May 11–12, 2005.

[33] R.S. Williams, P.J. Kuekes, Demultiplexer for a molecular wire crossbar network, US Patent Number 6,256,767, July 3, 2001.

[34] Michael Mitzenmacher, Eli Upfal, Probability and Computing: Randomized Algorithms and Probabilistic Analysis, Cambridge University Press, Cambridge, 2005.

[35] Jia Wang, Ming-Yang Kao, Hai Zhou, Address generation for nanowire decoders, in: GLSVLSI '07: Proceedings of the 17th Great lakes Symposium on VLSI, 2007, pp. 525–528.