Stochastic Nanoscale Addressing for Logic

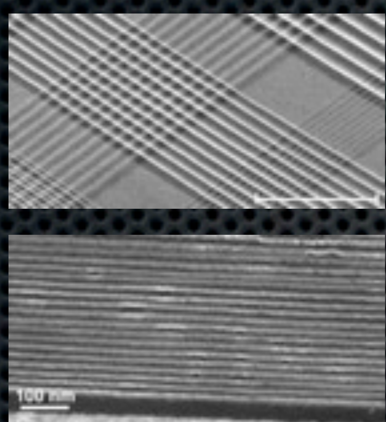
Eric Rachlin and John E. Savage

Brown University Department of Computer Science

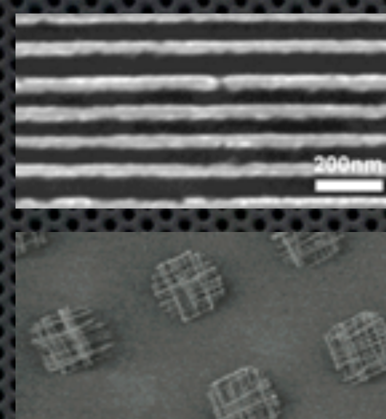
Nanoarch 2010

Nanowire Crossbar-based Architectures

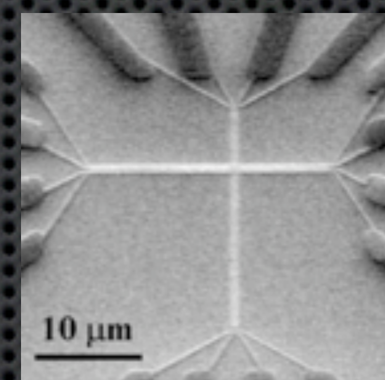
Nanowire crossbar-based devices offer a promising near-term path toward nanoscale computing.



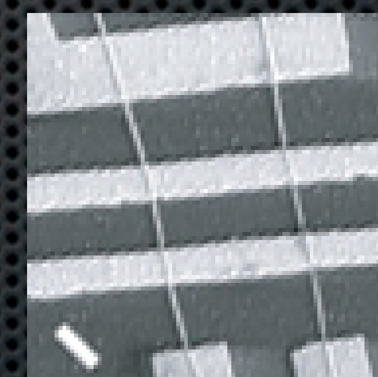
SNAP NWs
(Heath, Caltech)



CVD NWs
(Lieber, Harvard)



Programmable Crossbar
(Williams, HP)



Axially Encoded NWs
(Lieber, Harvard)

Many individual NW-based devices have already been demonstrated, but incorporating these devices into large-scale architectures remains a key challenge.

Stochastic Assembly

- Today's architectures are reliably produced using **top down** photolithographic assembly. Scaling this approach to the nanoscale is extremely challenging.
- While **bottom up** assembly of nanoscale architectures appears more feasible, it implies substantial variation in device functionality, placement, and interconnect.
- By modeling device assembly probabilistically we can derive tight analytic bounds on the area required by various stochastically assembled nanoscale devices.
- Here we consider **NW decoders** for logic circuits.

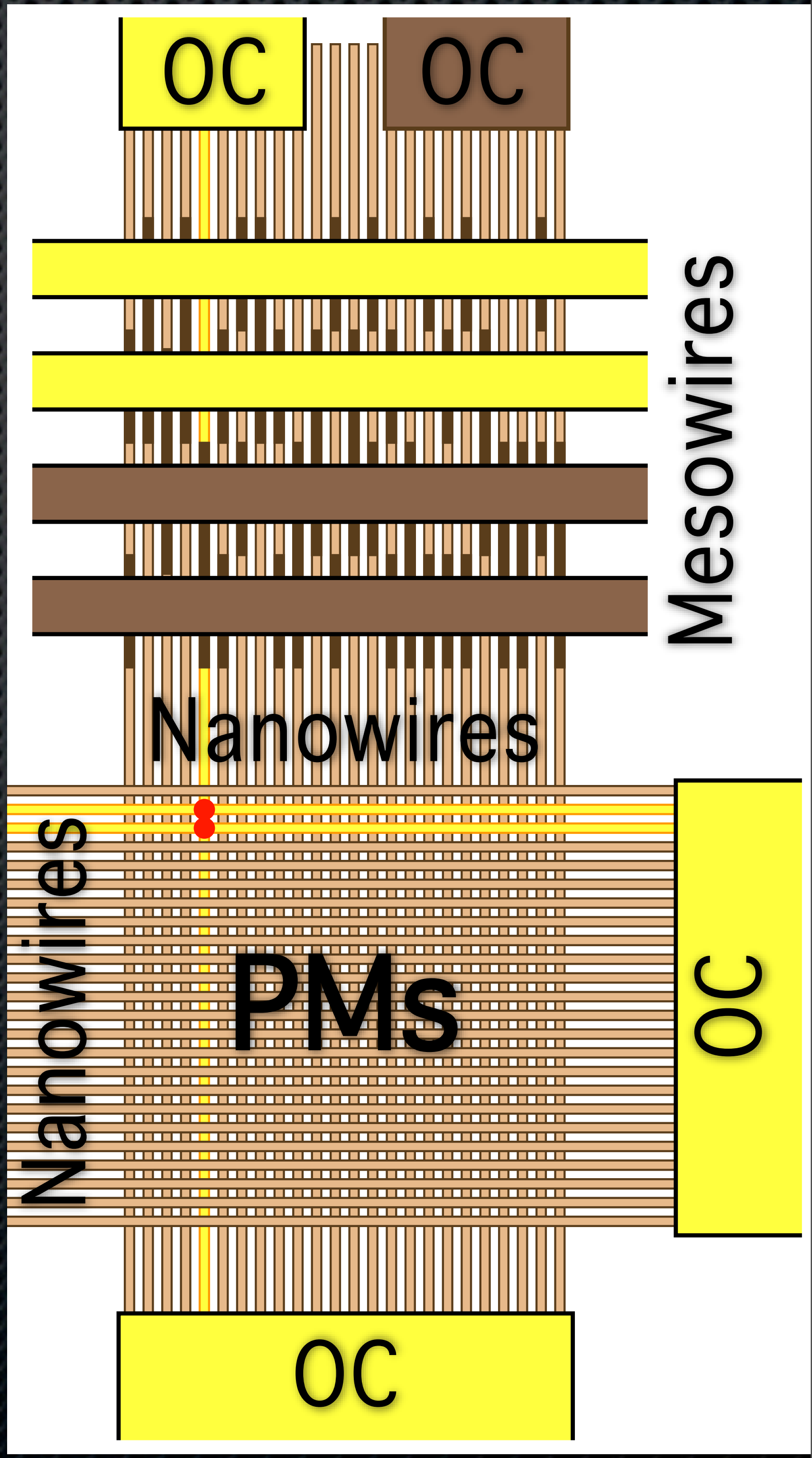
NW Addressing via NW Decoders

To impliment nanoscale architectures, we must gain control over individual NWs.

A **NW decoder** interfaces ***N*** NWs with ***M*** **mesoscale wires**.

In a NW decoder, **mesoscale contacts** place voltages across blocks of *N* NWs. Each MW then provides control over a random subset of the activated NWs.

The addressed NWs can supply inputs to a **NW crossbar**, a molecular device layer sandwiched by two sets of parallel NWs.



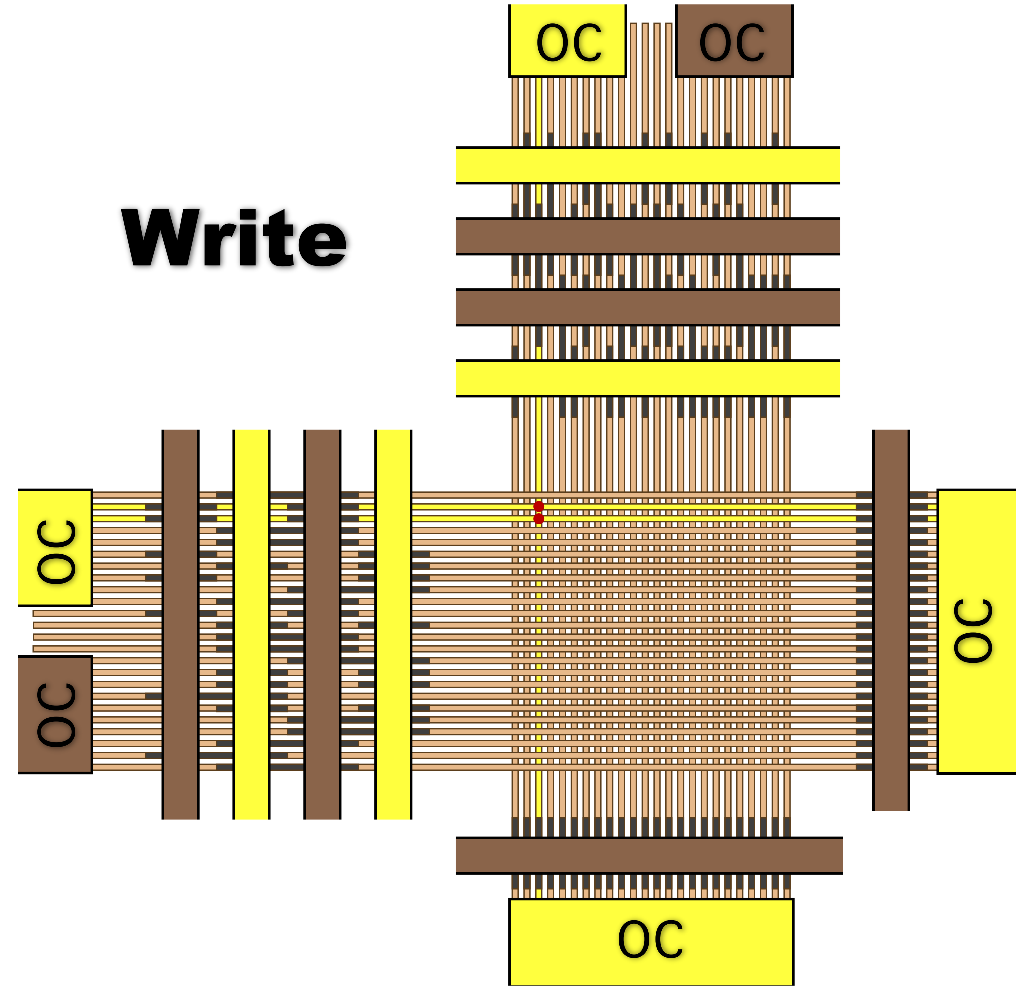
Crossbar-based Memories

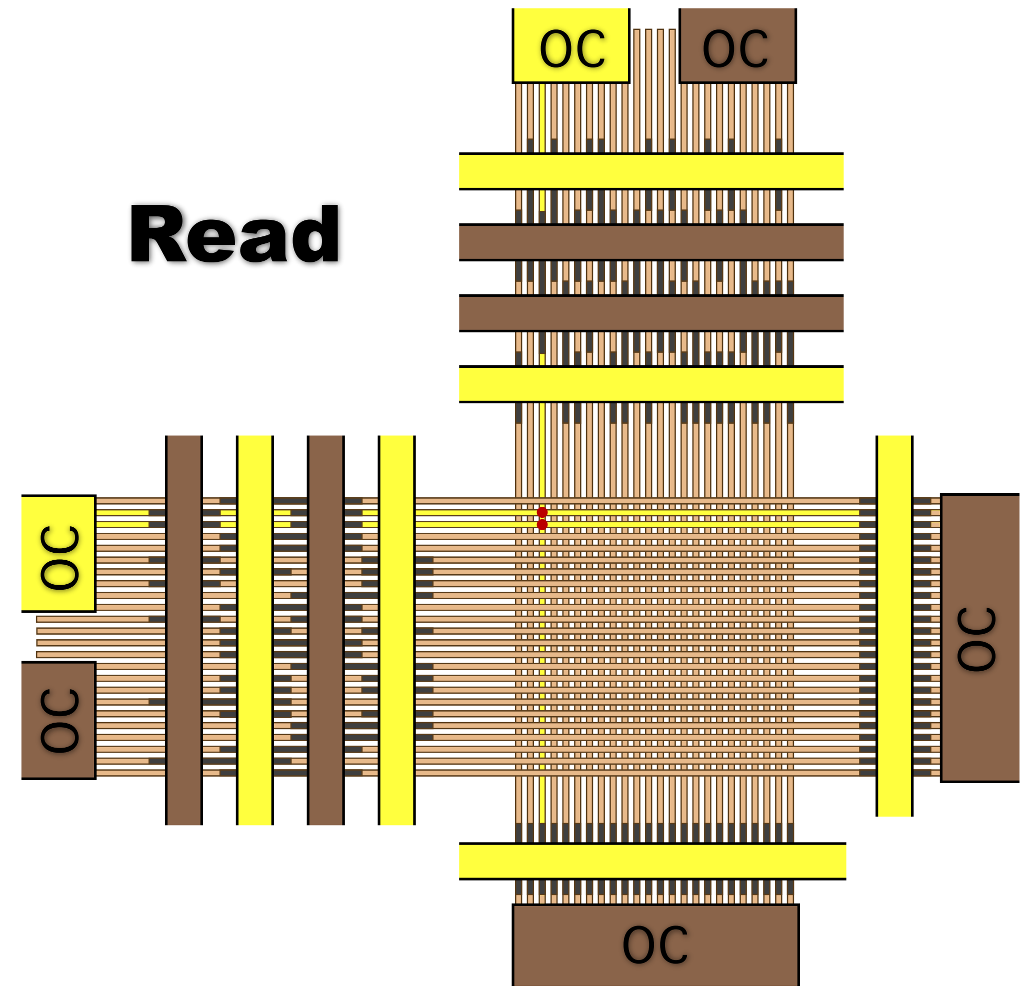
Perpendicular NWs provide control over molecular devices. This allows the crossbar to act as a memory

In a **write operation** a large voltage is used to set the conductivity of crosspoints.

In a **read operation** a smaller voltage is used to measure their conductivity.

Many NWs along each dimension must be addressable. It is acceptable to store the same bit at multiple crosspoints.



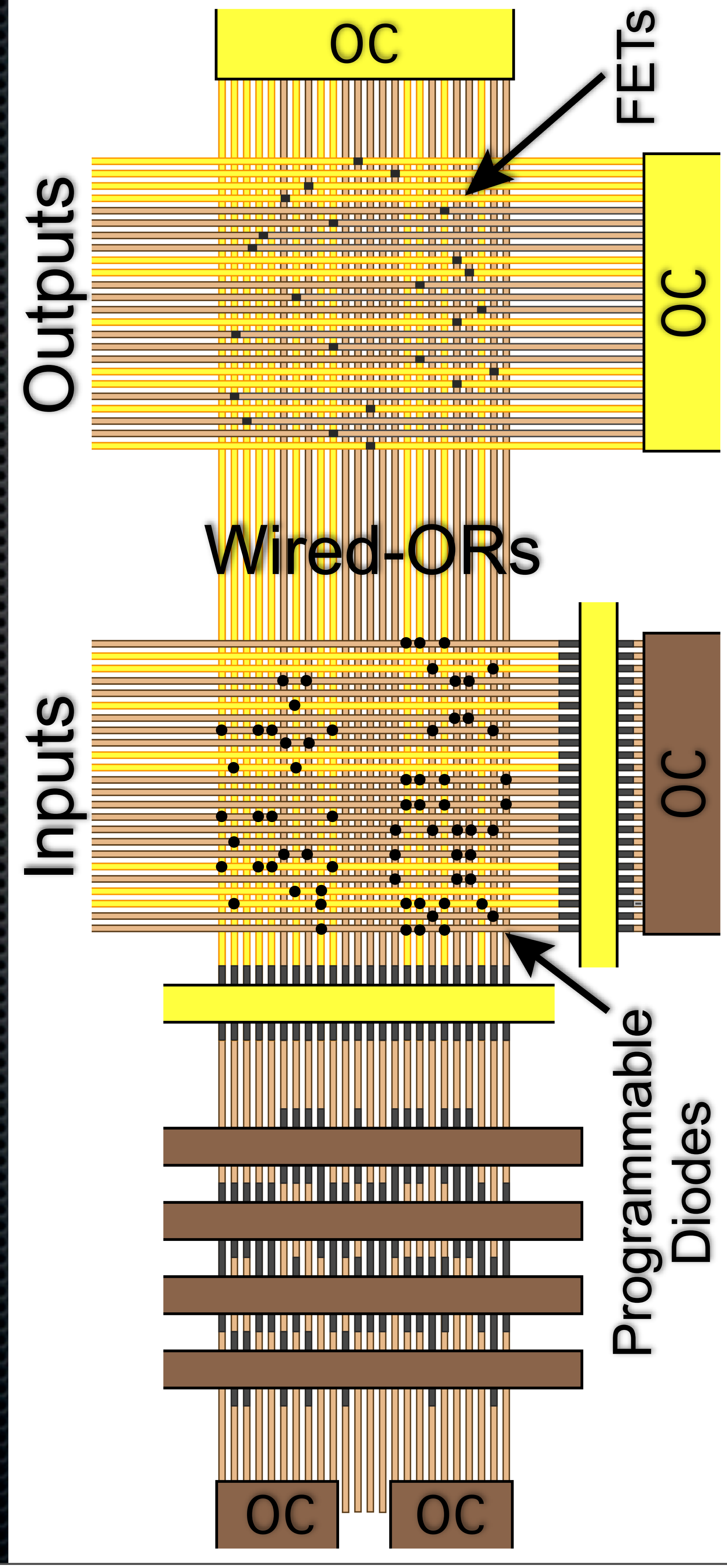


Crossbar-based Logic

By addressing multiple NWs in one dimension of a crossbar memory, the diode connections along NWs in the other dimension perform **wired-ORs**.

This allows for programmable logic, although signal restoration is needed.

To this end, an **inversion operation** can be performed via stochastically placed field-effect transitors (FETs).



NW Decoder Requirements

- A set of NWs is **addressed** if all NWs in the set are on (e.g. conducting) while all other NWs are off.
- **Decoders for Memories:** In a crossbar memory with N NWs along each dimension, decoders that address N_A disjoint sets of NWs provide control over $(N_A)^2$ individually addressable storage locations.
- **Decoders for Logic:** In a circuit with N_A input bits, a decoder must address all 2^{N_A} subsets of the N_A NWs.
- In a logic decoder N_A out of N NWs must each be controlled by a unique MW.

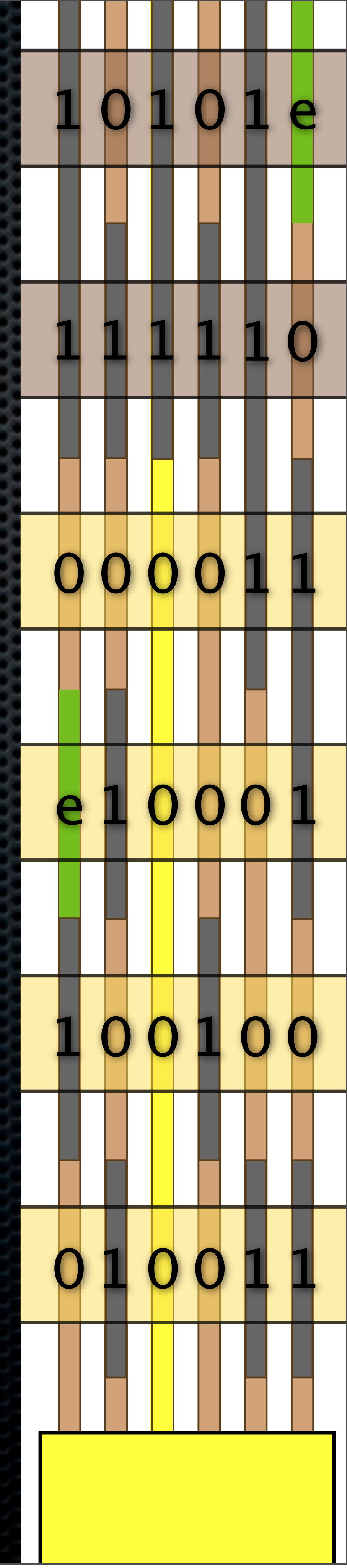
Modeling NW Decoders

In a NW decoder, each of the *M* MWs controls a random subset of the *N* NWs.

During decoder assembly, ***M*-bit codewords, c_i,** are stochastically assigned to NWs.

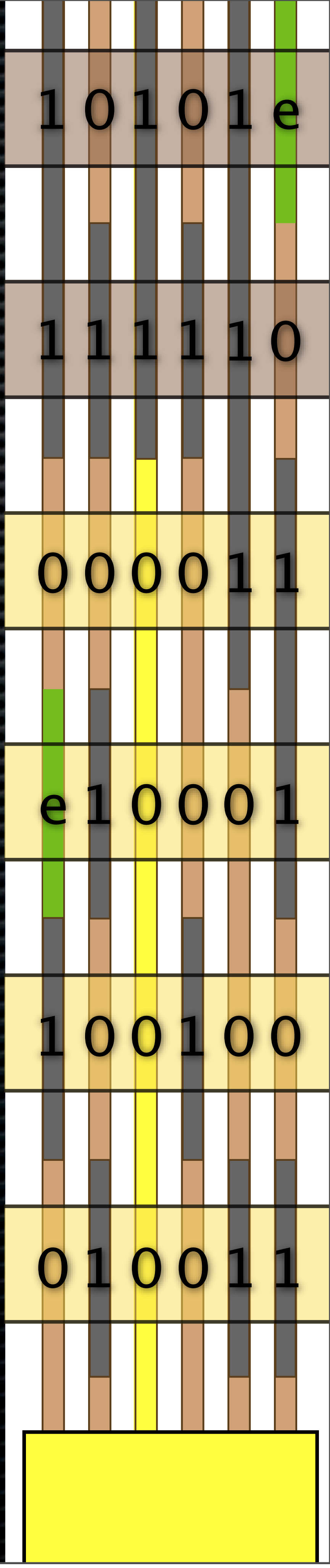
- $c_{ij} = 1$ if the i^{th} NW is controlled (turned off) by the j^{th} MW
- $c_{ij} = 0$ if the i^{th} NW is unaffected (left on) by the j^{th} MW.
- $c_{ij} = e$ if the i^{th} NW is partially controlled by the j^{th} MW.

A set of NWs is **addressable** if activating all MWs that do not affect those NWs reliably turns off all other NWs.



NW Decoders for Logic

- Codewords are assigned to NWs stochastically. For a given decoder, we model the distribution with which codewords are assigned.
- For logic decoders, we then bound the number of MWs, M, required for at least N_A NWs to be “fully addressable” with probability 1 - ε.
- A set of N_A NWs is **fully addressable** if all 2^{N_A} subsets of the NWs can be addressed. This implies that each of the N_A NWs is controlled by a unique MW.

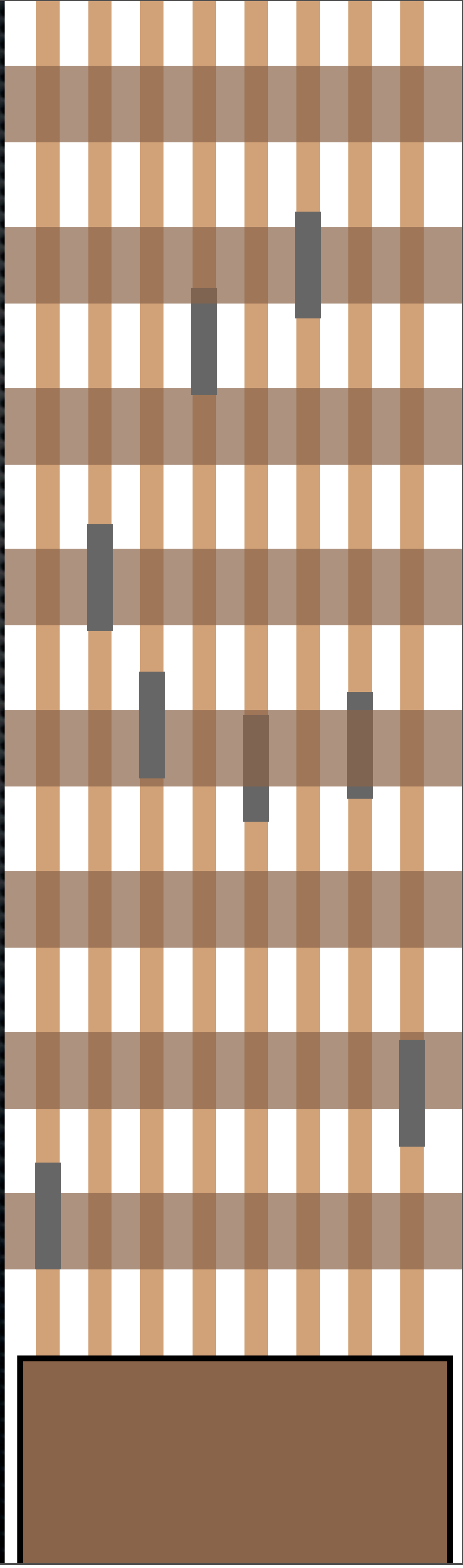


Encoded NW Decoders

- NWs can be grown with sequences of lightly and heavily doped regions along their axis.
- To produce an encoded NW decoder, many copies of differently encoded NWs are grown and collected in a single large ensemble.
- A random subset of these NWs is deposited onto a chip. This assigns each NW one of *C* codewords independently at random.
- For memories, such decoders only need close to *log N* MWs. Axial misalignment is a concern and may cause codeword errors.

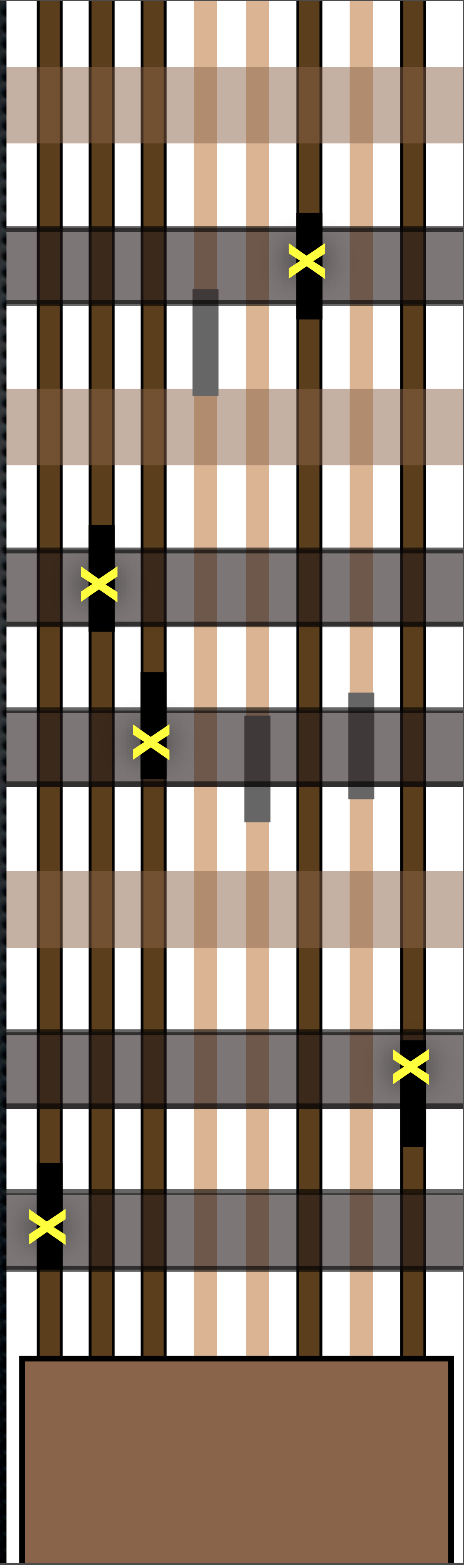
Upper bounding M

- In a logic decoder, NWs can be encoded such that each NW is controlled by one MW.
- We bound M x N such that there exists a set of N_A MWs, each of which controls a distinct NW with probability $1 - \varepsilon$.
- We consider the case where $M = N = \beta N_A$, and model the decoder assembly as a "coupon collectors problem".
- Here each of N NW "collects" one of N MWs independently at random. We bound N such that N_A MWs are collected with prob. $1 - \varepsilon$



Upper bounding M

- In a logic decoder, NWs can be encoded such that each NW is controlled by one MW.
- We bound $M \times N$ such that there exists a set of N_A MWs, each of which controls a distinct NW with probability 1 - ε.
- We consider the case where $M = N = \beta N_A$, and model the decoder assembly as a "coupon collectors problem".
- Here each of N NW "collects" one of N MWs independently at random. We bound N such that N_A MWs are collected with prob. 1 - ε



Upper bounding *M*

Once i MWs are collected, the probability an aligned NW collects a new MW is $p_i = (C - i)/C$.

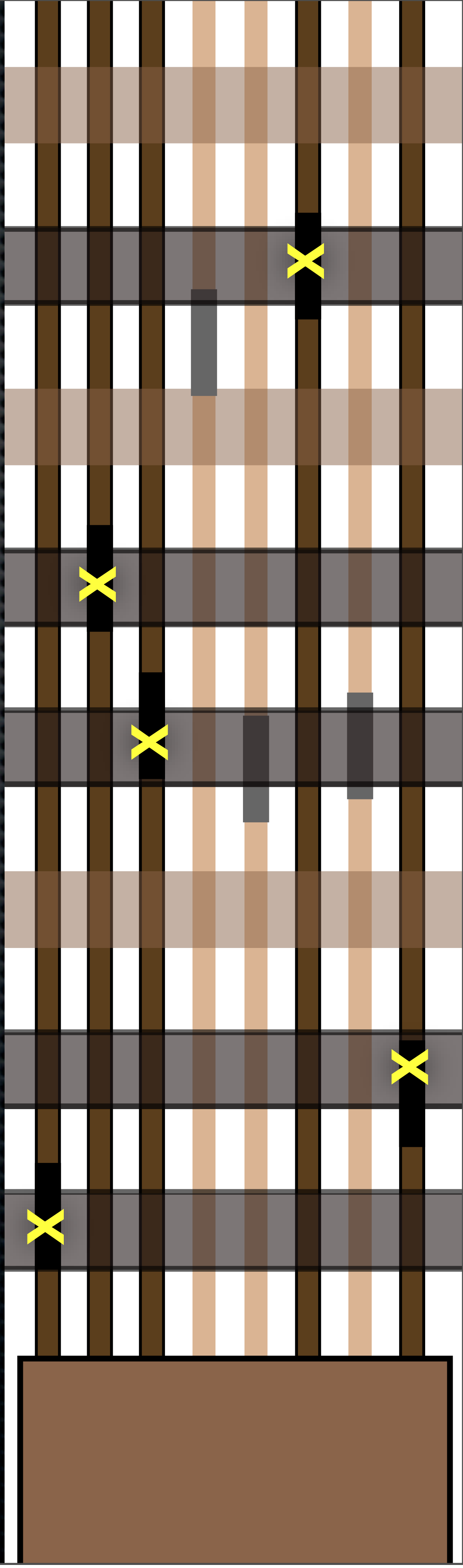
The expected number of aligned NWs needed to collect $N_A = N/\beta$ MWs can be expressed as

$$1/(p_0) + \ldots + 1/(p_{N/\beta - 1}) \approx -N \ln(1 - 1/\beta)$$

We can also show that for any ε, if N_A is sufficiently large, then N_A MWs are collected with probability 1 - ε when

$$\beta = p_S^{-1}(2 - \sqrt{2})^{-1}$$

and each NW is aligned with probability p_S.

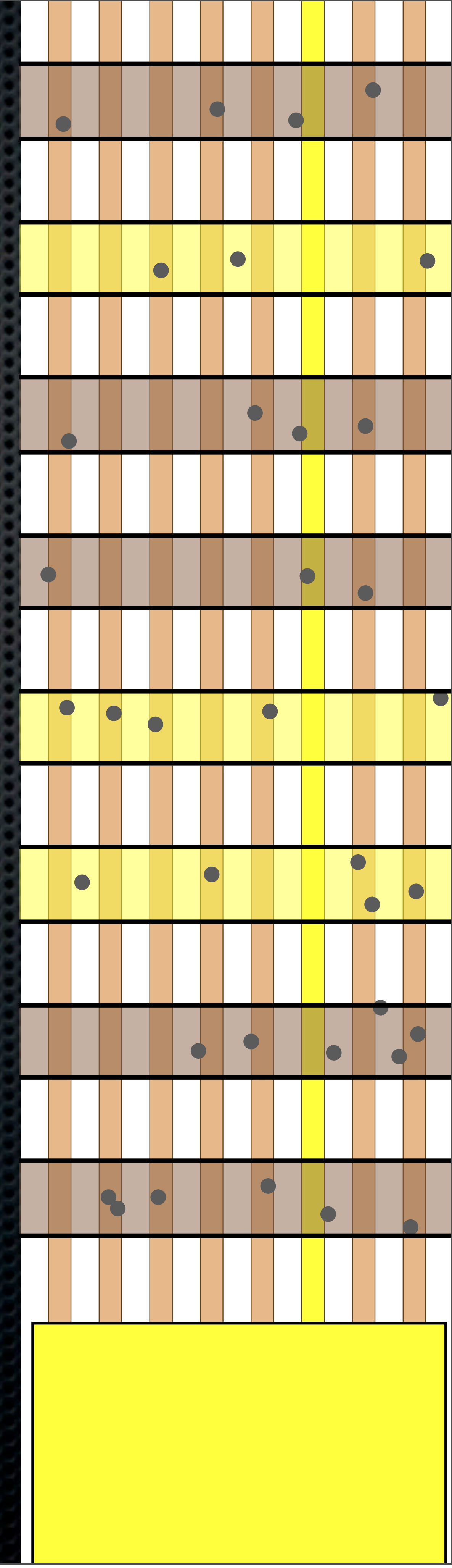


Randomized-Contact Decoders (RCDs)

- Encoded NWs are challenging to make. As an alternative, we consider RCDs.
- An RCD refers to <u>any NW decoder</u> where NW/MW junctions can be modeled as <u>i.i.d. random variables</u>. For each variable, c_{ij}

$$P(c_{ij} = 1) = p,\ P(c_{ij} = 0) = q,\ P(c_{ij} = e) = 1 - p - q$$

- This <u>more general model</u> of decoder manufacture <u>explicitly accounts for errors</u>.
- When $p = \alpha/N$, $q = 1 - 1/N$, our previous upper bound applies with $p_s \approx \alpha e^{-1}$.



Lower Bounding *M*

- When $M = N = \beta N_A$, we have upper bounded β for stochastically assembled NW logic decoders. We can also obtain an information theoretic lower bound on β.
- When a decoder is stochastically assembled, it is given a random **configuration of codewords**, ***C***. The entropy of this configuration, $h(\mathbf{C})$, is easy to compute.
- A "successful" configuration contains a subset of N_A NWs, $\mathbf{S_N}$, uniquely coupled to a set of N_A MWs, $\mathbf{S_M}$.
- Let ***S*** denote the subset of ***C*** that describes the N_A^2 crosspoints of $\mathbf{S_N}$ and $\mathbf{S_M}$.

Lower Bounding *M*

$\mathbf{S} \subset \mathbf{C}$ is defined by $\mathbf{S}_N$, $\mathbf{S}_M$, and an <u>ordering</u>, $\boldsymbol{\pi}$, of NWs in $\mathbf{S}_N$. The <u>entropy of **S**</u> is $h(\mathbf{S}) \leq h(\mathbf{S}_N) + h(\mathbf{S}_M) + h(\boldsymbol{\pi})$.

Let $h(\mathbf{C}\text{-}\mathbf{S}|\mathbf{S})$ denote the entropy of the $N^2 - N_A{}^2$ additional junction of **C** given **S**. If **C** "succeeds" with probability $1\text{-}\varepsilon$

$$(1\text{-}\varepsilon)h(\mathbf{C}) \leq h(\mathbf{S}) + h(\mathbf{C}\text{-}\mathbf{S}|\mathbf{S})$$

Error-free RCDs: $h(\mathbf{C}) = N^2 h(p)$, $h(\mathbf{C}\text{-}\mathbf{S}|\mathbf{S}) \geq (N^2\text{-}N_A{}^2)h(p^*)$ where $h(p) = N^2(p\log_2 p - q\log_2 q)$ and $p \leq p^* \leq p\beta(1\text{-}\beta^2)$.

$$(1\text{-}\varepsilon)\, N^2 h(p) \leq h(\mathbf{S}_N) + h(\mathbf{S}_M) + h(\boldsymbol{\pi}) + (N^2 - N_A{}^2)h(p^*)$$

Here $h(\mathbf{S}_N) = h(\mathbf{S}_M) = \log_2(N \text{ choose } N_A)$, $h(\boldsymbol{\pi}) = \log_2(N_A!)$. Considering p in terms of $1/N = 1/\beta N_A$ reveals $\beta \geq 1.25$

Conclusions

- By probabilistically modeling the assembly of nanoscale structures, manufacturing errors can be accounted for, and tight analytic bounds on area can be obtained
- For RCD and encoded NW decoders, for both memories and circuits, stochastic assembly introduces only a small constant factor overhead.
- Entropy-based lower bounds offer a general technique for bounding the overhead required to reliably obtain functional stochastically assembled structures.