Welcome to
048704/236803
Seminar on Coding for Non-Volatile Memories

1956: IBM RAMAC
5 Megabyte Hard Drive

Above: *An IBM Model 350 Disk File being delivered. Yes, that's ONE hard disk drive unit.*

A 2014 Terabyte Drive

Some of the main goals in designing a computer storage:

Price

Capacity (size)

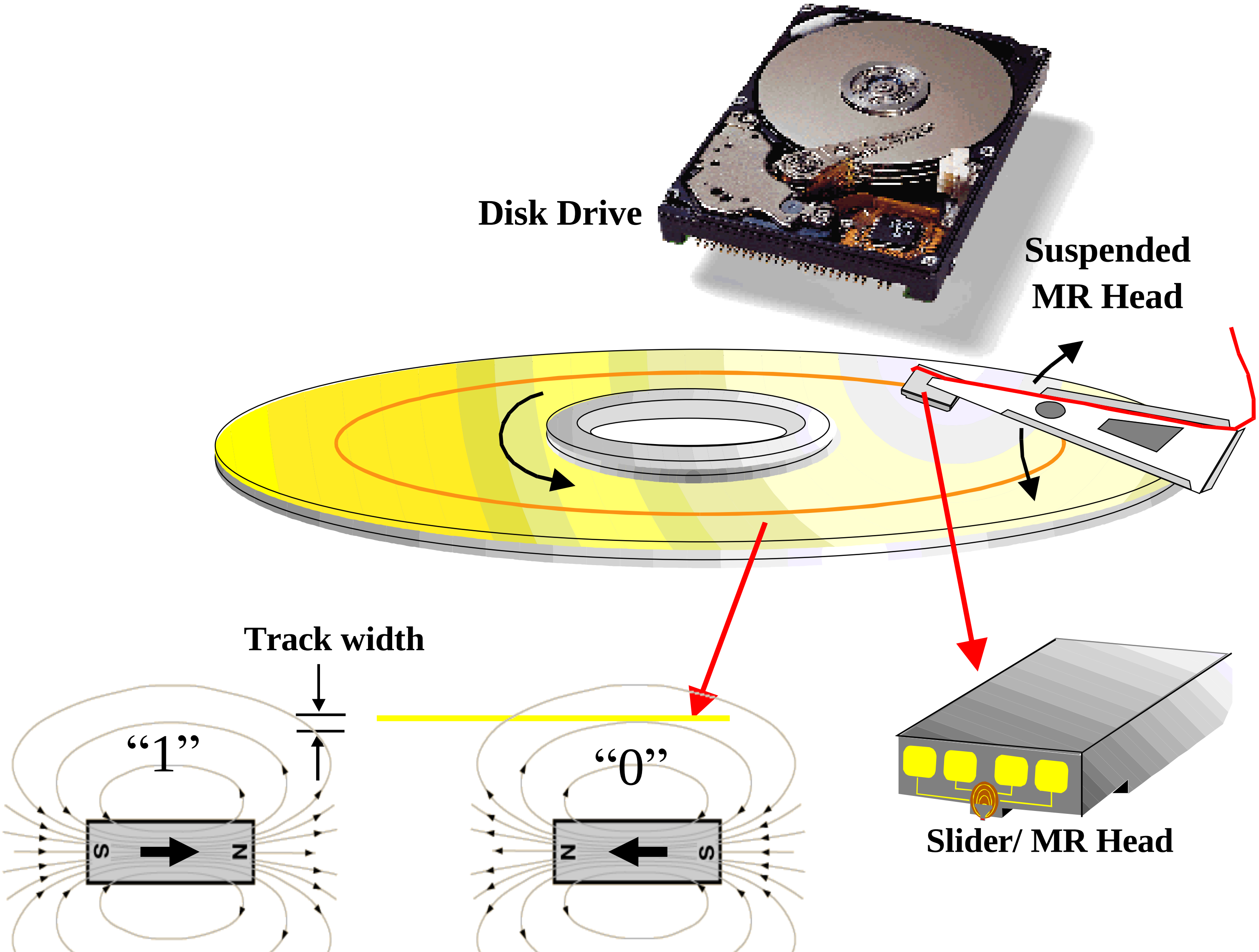
Endurance

Speed

Power Consumption

The Evolution of HDD

Disk Drive Basics



Disk Drive

Suspended MR Head

Track width

Slider/ MR Head

Memories Today

- RAM memories, DRAM, SRAM

Volatile Memories

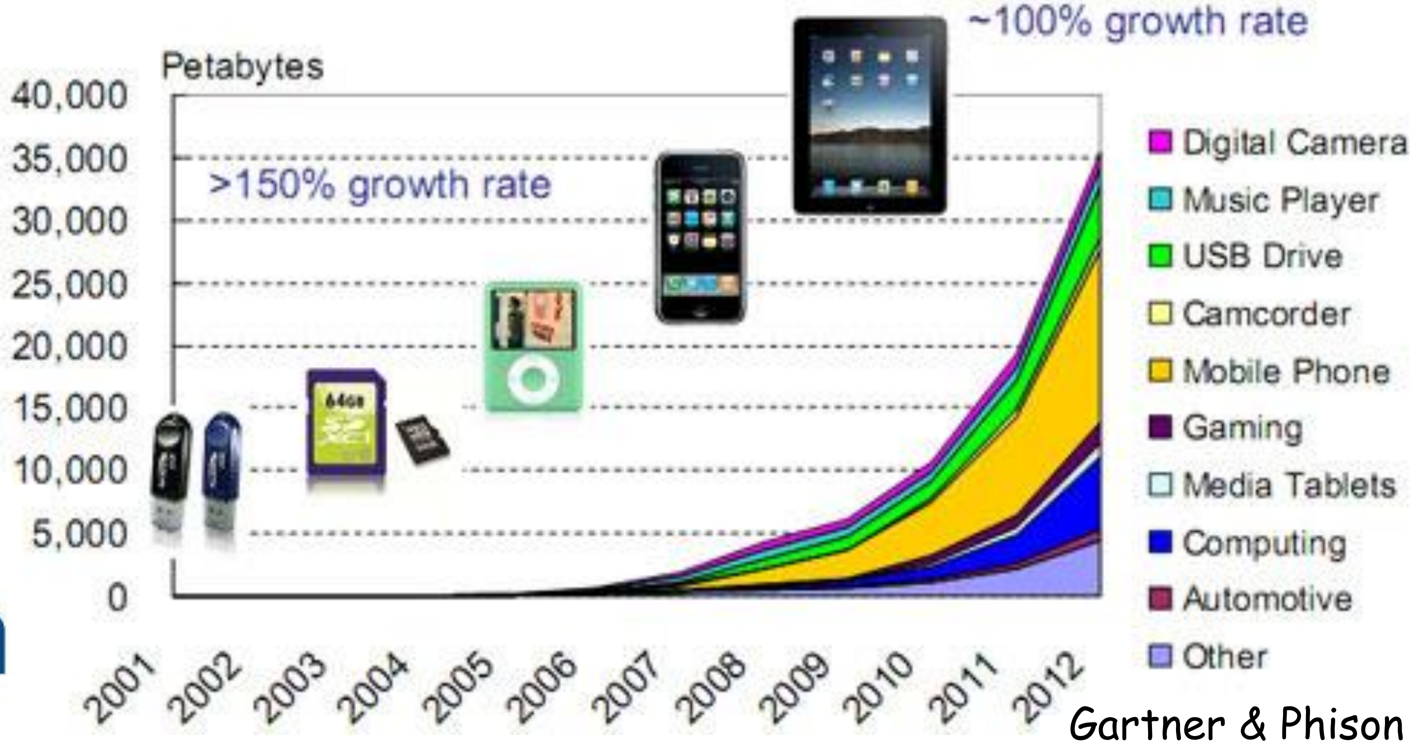
- Hard disk
- Tape
- Floppy disk
- Optical disc: CD, DVD, BlueRay
- Punch cards
- Flash memories
- Phase change memories
- Memristors/STTRAM/MRAM

Non-Volatile Memories

SanDisk

SAMSUNG

Micron



Gartner & Phison

SPANSION™

EMC²

where information lives®

=

XtremIO

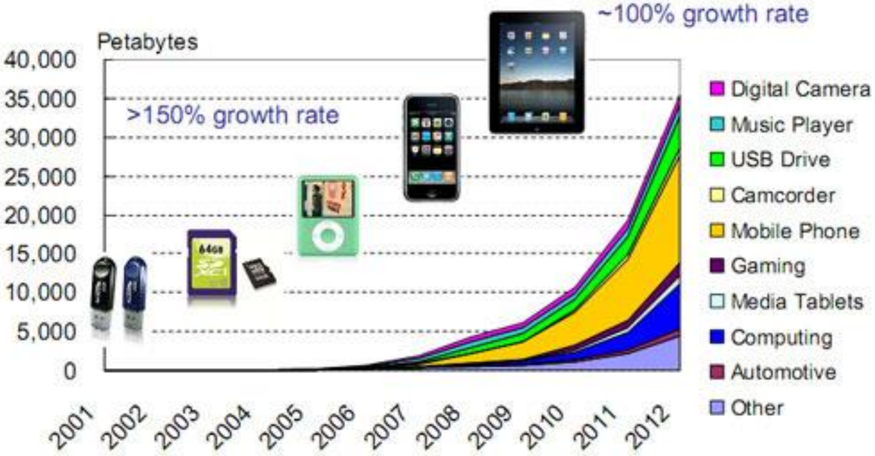
inside

densbits

TECHNOLOGIES

anobit

making flash better



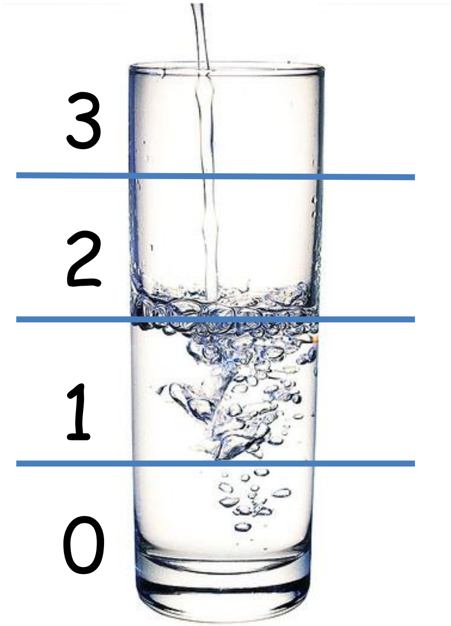
Fast

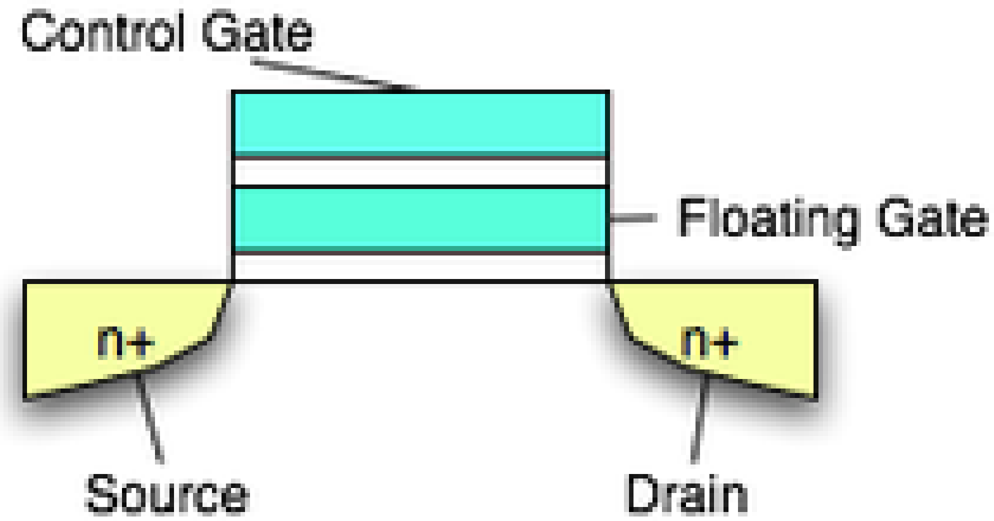
Low Power

Reliable

~10^5 P/E Cylces

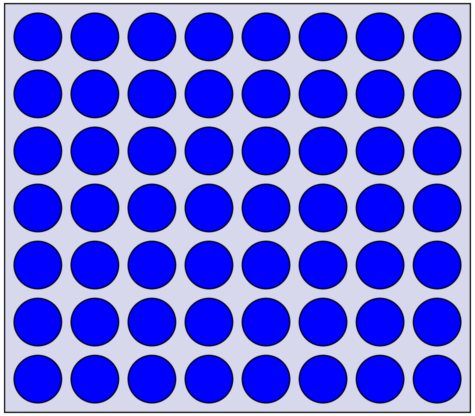
Flash Memory Cell

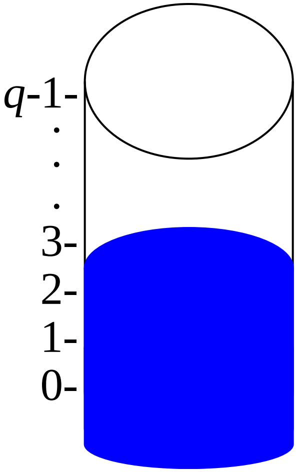




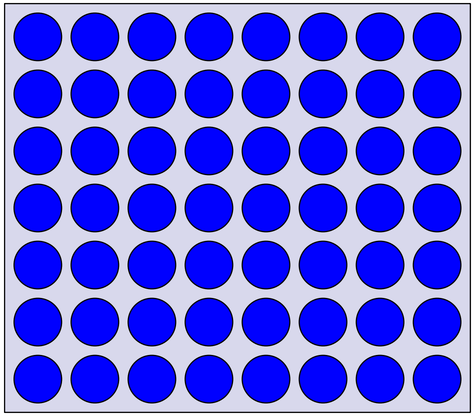
Multi-Level Flash Memory Model

- Array of cells, made of floating gate transistors
  - Each cell can store **q** different values.
  - Today, **q** typically ranges between **2** and **16**.

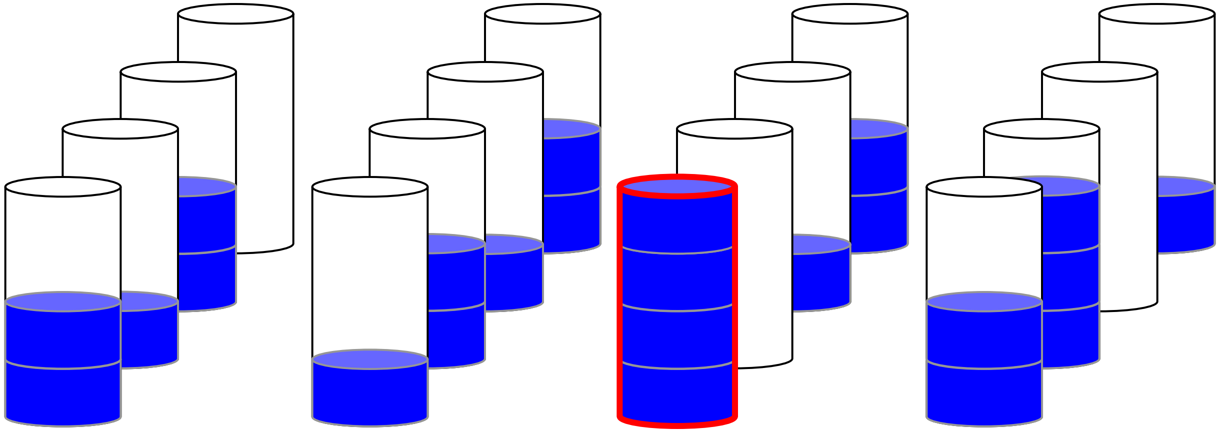




Multi-Level Flash Memory Model



- Array of cells, made of floating gate transistors
  - Each cell can store **q** different values.
  - Today, **q** typically ranges between **2** and **16**.
  - The cell's level is increased by pulsing electrons.
  - Reducing a cell level requires **resetting** all the cells in its containing block to level 0 – ***A VERY EXPENSIVE OPERATION***

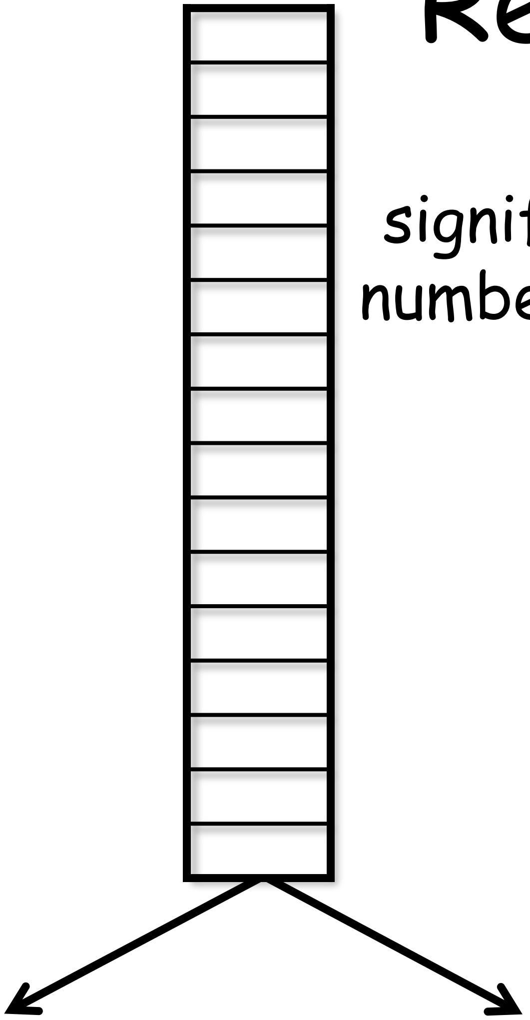


Flash Memory Constraints

- The **lifetime/endurance** of flash memories corresponds to the number of times the blocks can be erased and still store reliable information
- Usually a block can tolerate **~10^4-10^5** erasures before it becomes unreliable
- **The Goal:** Representing the data efficiently such that block erasures are postponed as much as possible - **Rewriting codes**

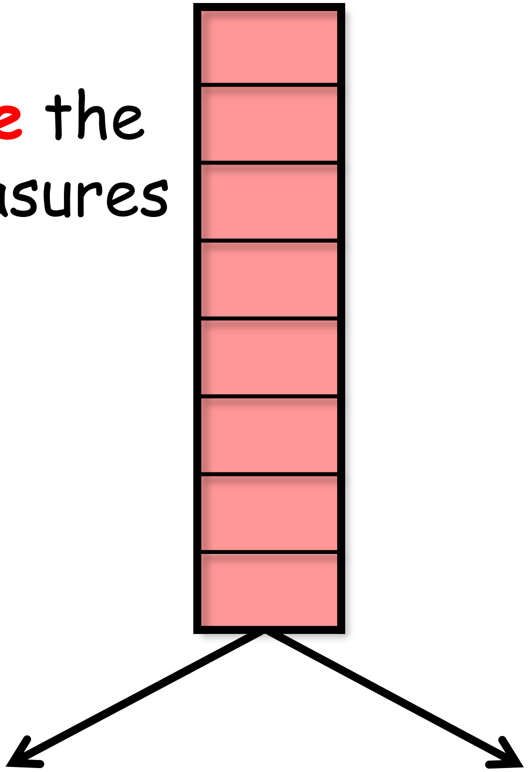
Rewrite Codes

Rewrite codes significantly **reduce** the number of block erasures



Store 4 bits once

Store 1 bit 16 times



Store 3 bits once

Store 1 bit 8 times

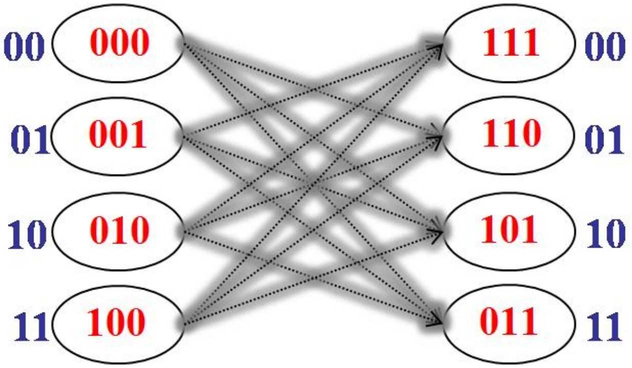
Write-Once Memories (WOM)

- Introduced by **Rivest and Shamir**, "*How to reuse a write-once memory*", 1982
- The memory elements represent bits (2 levels) and are irreversibly programmed from **'0'** to **'1'**

Bits Value	1st Write	2nd Write
00	000	111
01	001	110
10	010	101
11	100	011

1st Write

2nd Write



Write-Once Memories (WOM)

- Examples:

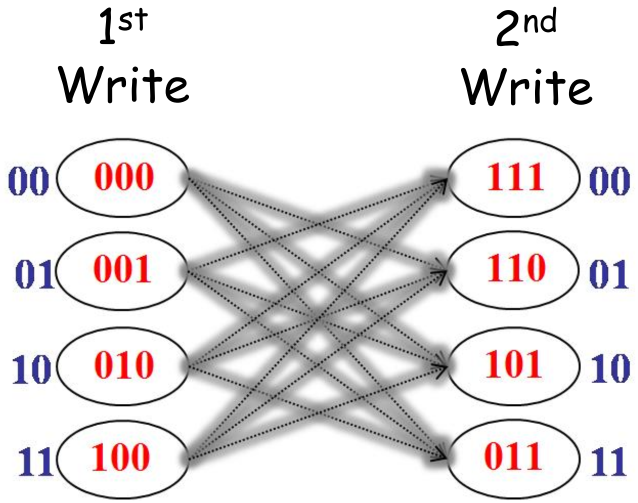
data	Memory State

data	Memory State

data	Memory State

data	Memory State

Bits Value	1st Write	2nd Write
00	000	111
01	001	110
10	010	101
11	100	011



- **The problem:**
  What is the total number of bits that is possible to write in **n** cells in **t** writes?

Binary WOM-Codes

- An **[n,t;M_1,…,M_t] t-write WOM-code** has **n** cells and guarantees any **t** writes of alphabet size **M_1,…,M_t** by programming cells from **0** to **1**
  - Example: the Rivest-Shamir code is
- The **sum-rate** of the WOM-code is

$$R = (\Sigma_1^t \log M_i)/n$$

  - Example: the Rivest-Shamir sum-rate is
- **Remark**: There are two cases:
  - Individual rates on each write must **all be the same (fixed-rate)**
  - Individual rates are **allowed to be different (unrestricted-rate)**

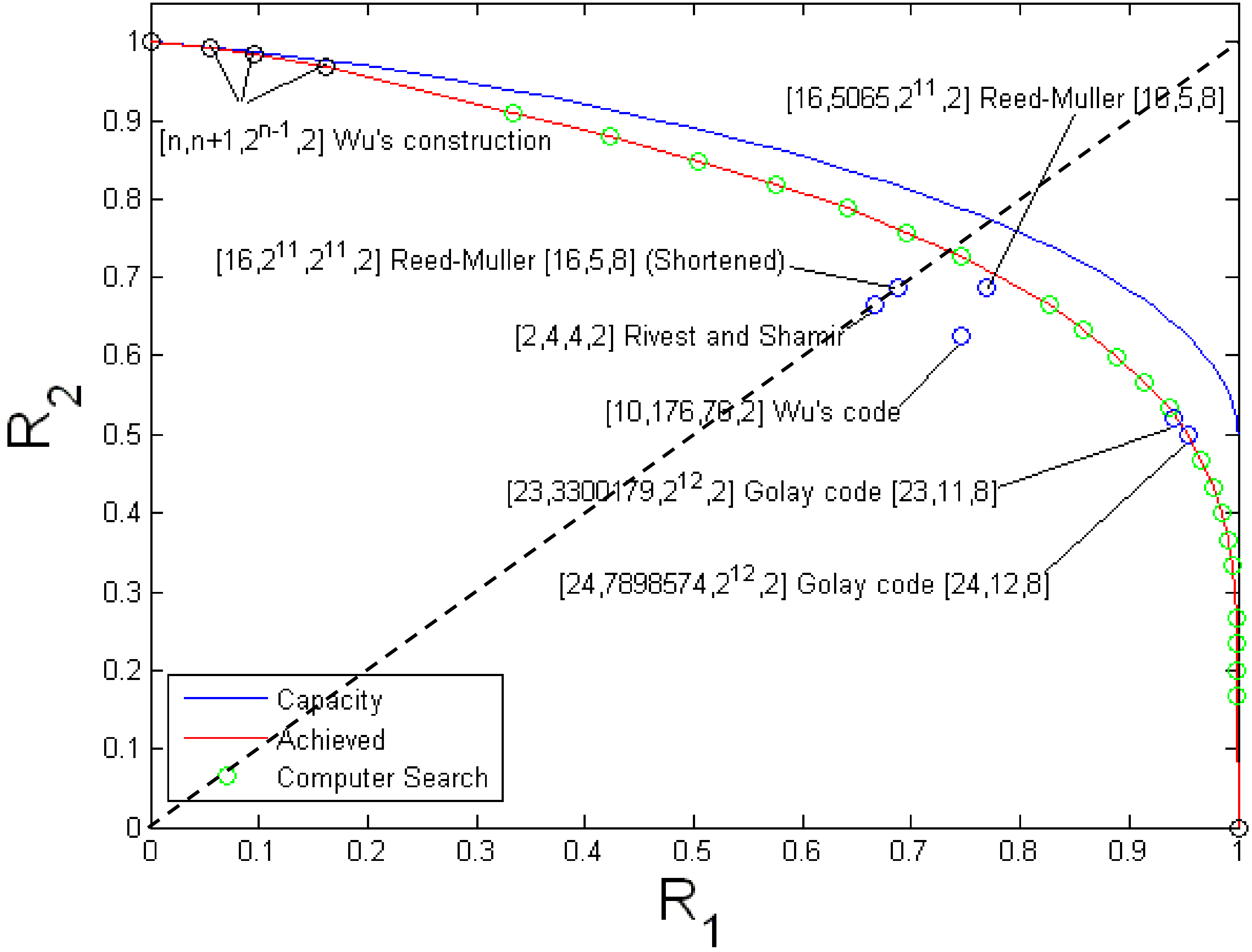
WOM Capacity

- Capacity region (**Heegard 1986, Fu and Han Vinck 1999**)

$$\begin{aligned} C_{t\text{-}WOM}=\{(R_1,\ldots,R_t)|\ R_1 &\leq h(p_1),\\ R_2 &\leq (1-p_1)h(p_2),\ldots,\\ R_{t-1} &\leq (1-p_1)\cdots(1-p_{t-2})h(p_{t-1})\\ R_t &\leq (1-p_1)\cdots(1-p_{t-2})(1-p_{t-1})\} \end{aligned}$$

- **Unrestricted-rate**: Maximum achievable sum-rate is $\log(t+1)$
- **Fixed-rate**: There is a recursive formula to calculate the maximum achievable sum-rate
- **Example:**
  - For two writes $C_{2\text{-}WOM}=\{(R_1,R_2)|\ R_1 \leq h(p), R_2 \leq (1-p)\}$
  - The maximum sum-rate is $\max_p\{h(p)+1-p\}=\log 3$
  - The max fixed-rate sum-rate is 1.54

WOM Capacity and Achievable Rates



Non-Binary WOM Codes

- **Definition**: An $[n,t; M_1,\dots,M_t]_q$ **t-write WOM code** is a coding scheme that consists of **n q**-ary cells and guarantees any **t** writes of alphabet size $M_1,\dots,M_t$ only by increasing the cell levels

- The **sum-rate** of the WOM-code is

$$R = \left(\sum_{i=1}^{t} \log M_i\right)/n$$

WOM Capacity

- The capacity of non-binary WOM-codes was given by **Fu and Han Vinck**, '99
- The maximal sum-rate using **t** writes and **q**-ary cells is

$$C = \log \binom{t+q-1}{q-1}$$

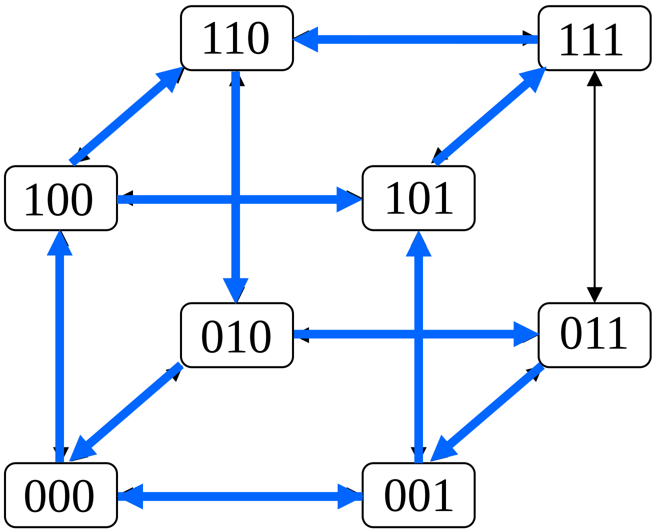
- There is no tight upper bound on the sum-rate in case equal amount of information is written on each write

Flash/Floating Codes

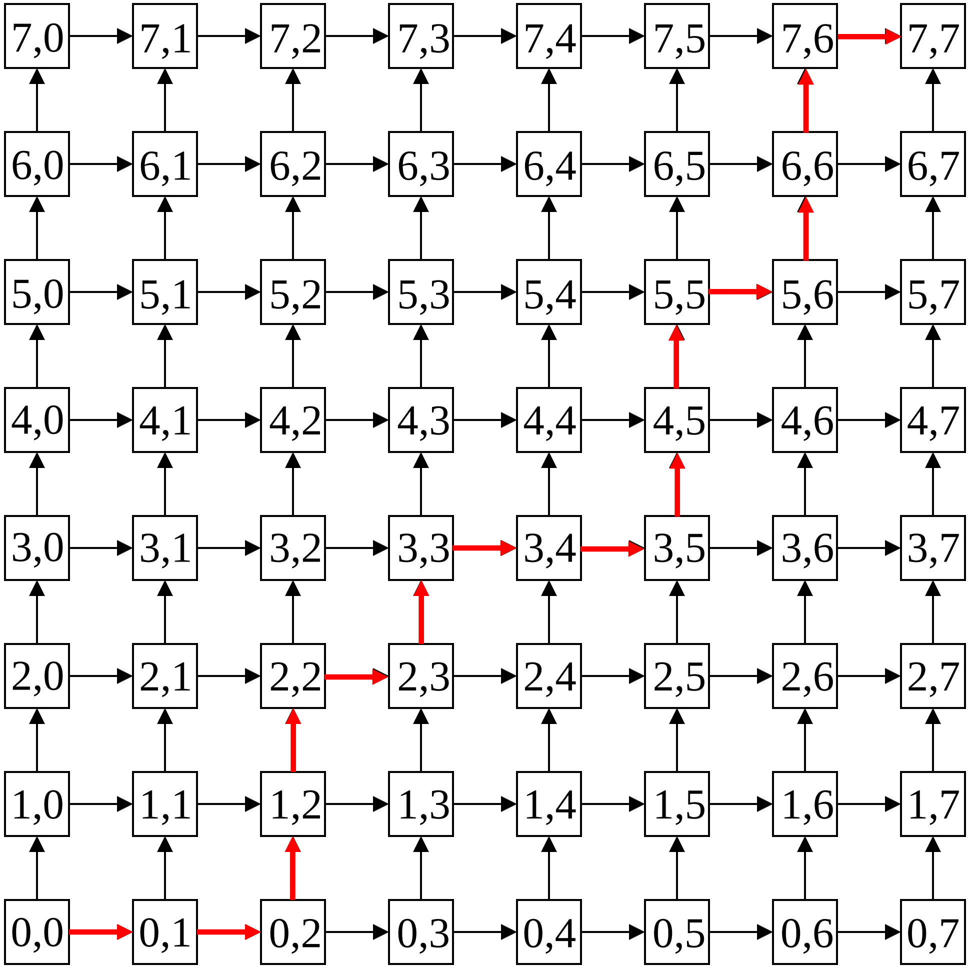
- k bits (more generally symbols) are stored using n cells
- A write is a change **0** → **1** or **1** → **0** of **one** of the k bits
- **Definition – Flash Codes**: An $(n, k, t)_q$ ***Flash Code*** is a coding scheme that accommodates ***any sequence of up to t writes*** of k bits, using n q-level cells, in such a way that a block erasure is never required.
- **Goal**: Given k, n, q, maximize the number of writes t

Flash/Floating Codes

Example: Storing **three bits** using **two 8-level cells**



Bits Diagram

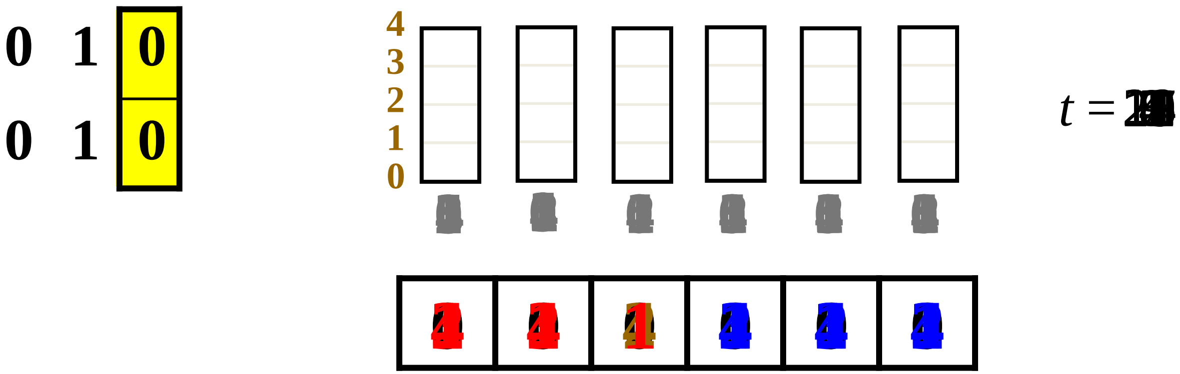


Cells Diagram

Example – Two Bits Construction

- Every cell is filled to the top before moving to the next one
- When the cells coincide, the last cell represents two bits. The cell's residue modulo 4 sets the bits value:

  0 – (0,0) 1 – (0,1) 2 – (1,0) 3 – (1,1)
- The maximum number of writes (worst case) is n(q-1) – [(q-1)/2] (optimal) before erasing is required.

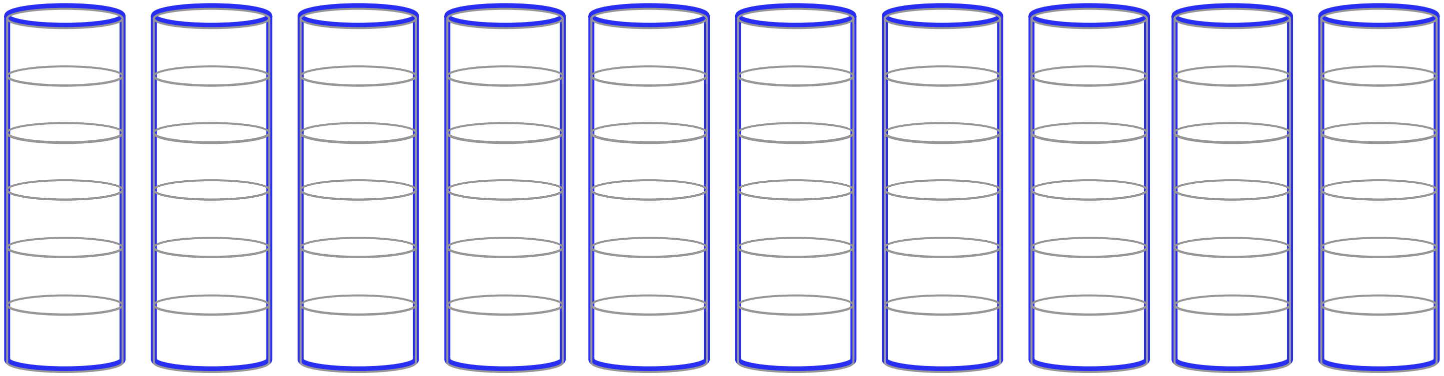


Trajectory Codes

- Flash/floating codes suffer from a restricted rewrite model – on each rewrite, only a single bit can be updated
- **Trajectory codes** extend this model
  Jiang, **Langberg**, **Schwartz**, **Bruck**, "*Universal Rewriting in Constrained Memories*", ISIT 09'
- The update transitions are depicted in a graph
- This extended model can fit all type of codes: flash/floating codes, buffer codes, rank modulation, WOM codes

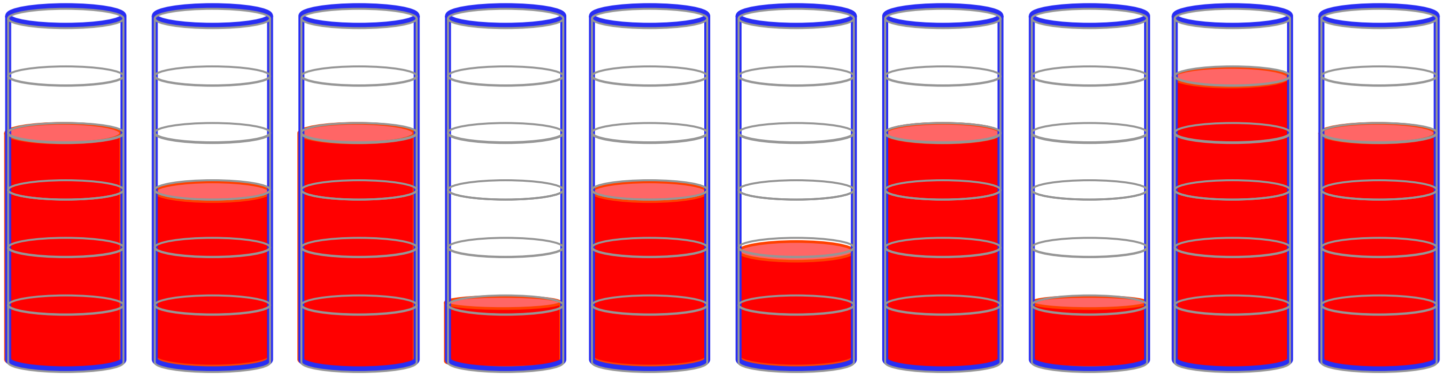
Asymmetric ECC

- Many storage applications, e.g. **flash memories**, **phase-change memories** and more, share the following common properties:
  - Cells have multiple levels: **$0,1,\ldots,q$-1**
  - Errors have an **asymmetric** behavior



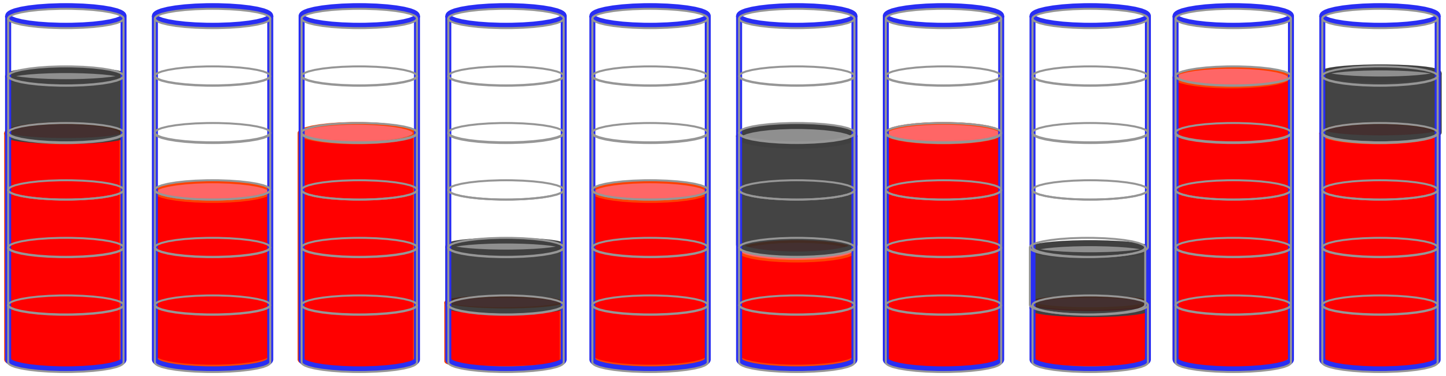
Asymmetric ECC

- Many storage applications, e.g. **flash memories**, **phase-change memories** and more, share the following common properties:
  - Cells have multiple levels: **$0, 1, \ldots, q-1$**
  - Errors have an **asymmetric** behavior
  - If a cell error occurs, then the cell level increases (or decreases) by at most **l** levels



Asymmetric ECC

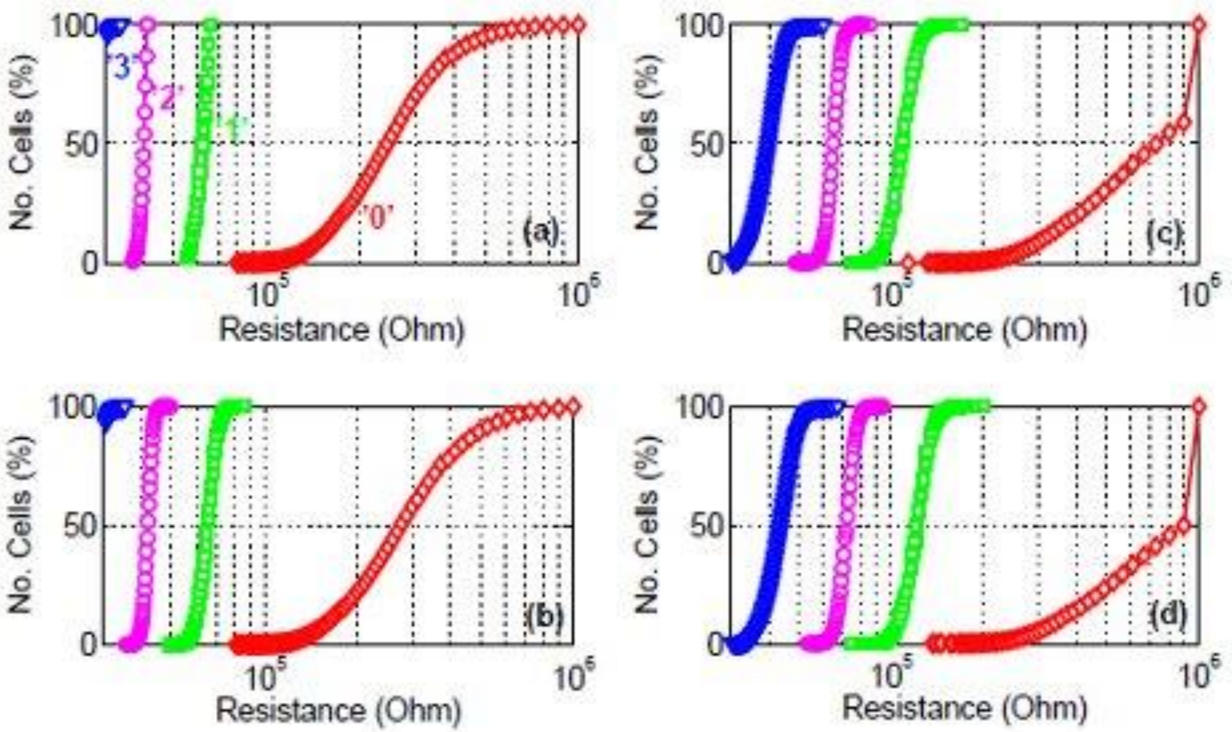
- Many storage applications, e.g. **flash memories**, **phase-change memories** and more, share the following common properties:
  - Cells have multiple levels: $0, 1, \ldots, q-1$
  - Errors have an **asymmetric** behavior
  - If a cell error occurs, then the cell level increases (or decreases) by at most l levels



Asymmetric ECC

- **Flash memories**
  - Cells **increase** their level during the programming process due to **over-shooting**
  - Cells **decrease** their level due to **data retention**
  - Errors become more prominent as the device is **cycled**
- **Phase change memories**
  - The **drift** in these memories changes the cells' levels in one direction

Asymmetric ECC

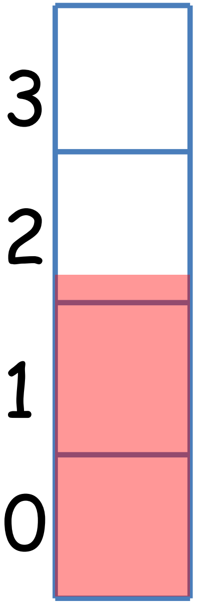


Time evolution of programmed resistance distributions of 200 kcells due to drift: (a) as programmed, and (b) 40μs, (c) 1000s, (d) 46,000s after programming.

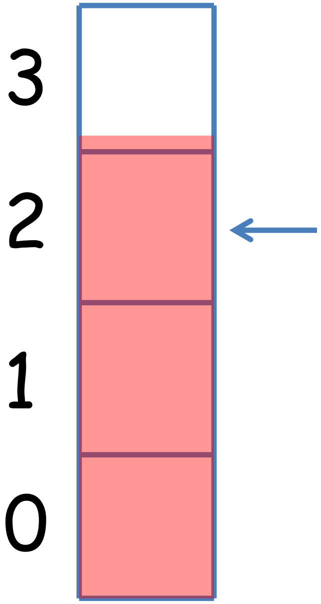
Figure from: N. Papandreou, H. Pozidis, T. Mittelholzer, G. F. Close, M. Breitwisch, C. Lam, and E. Eleftheriou, "Drift-Tolerant Multilevel Phase-Change Memory", 3rd IEEE Memory Workshop, May 2011

The Leakage Problem

Error!



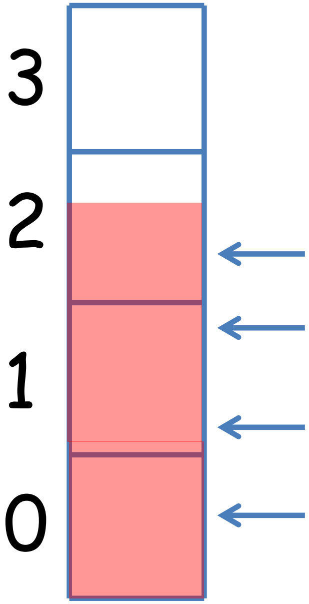
The Overshooting Problem



Need to erase the whole block

Possible Solution – Iterative Programming

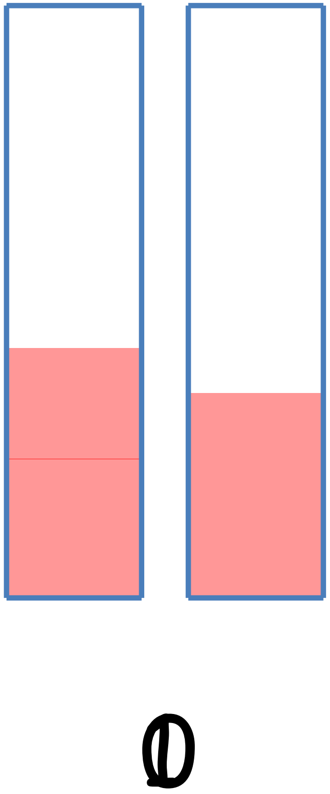
Slow…



Relative Vs. Absolute Values

Less errors

More retention



Jiang, Mateescu, Schwartz, Bruck,

"Rank modulation for Flash Memories", 2008

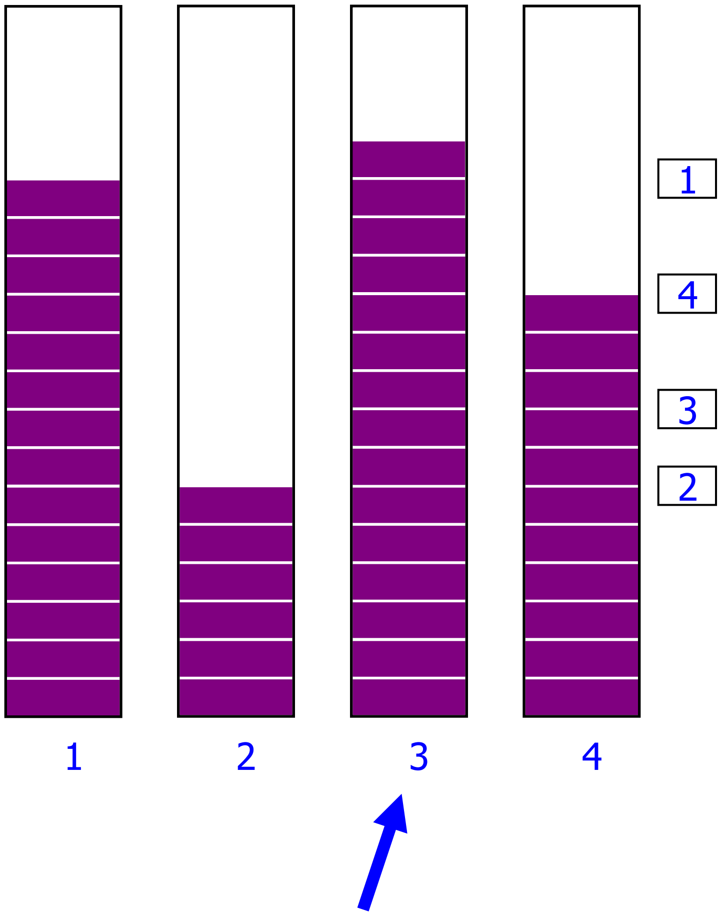
The New Paradigm
Rank Modulation

Absolute values → Relative values

Single cell → Multiple cells

Physical cell → Logical cell

Rank Modulation



Ordered set of **n** cells

Assume discrete levels

Relative levels define a **permutation**

Basic operation: **push-to-the-top**

Overshoot is not a concern

Writing is much faster

Increased reliability (data retention)

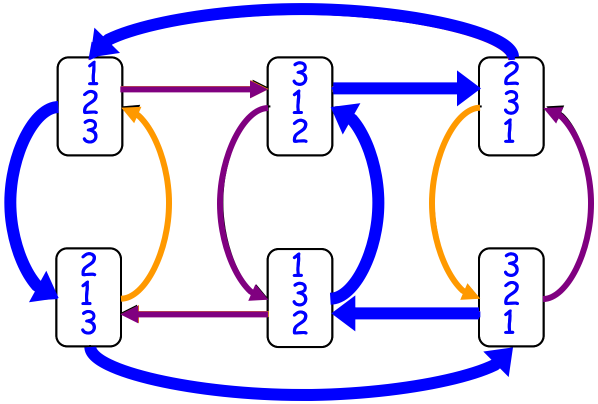
Gray Codes for Rank Modulation

The problem: Is it possible to transition between all permutations?

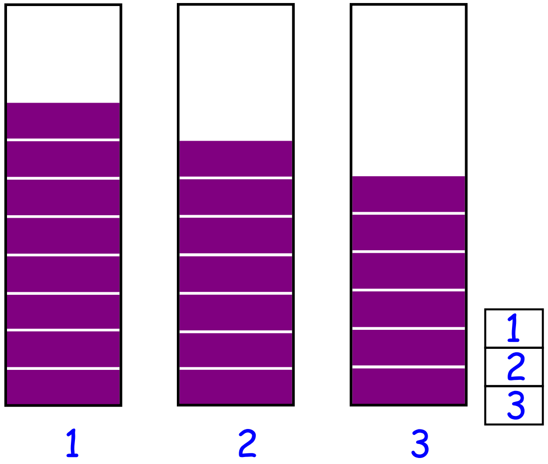
Find **cycle** through n! states by push-to-the-top transitions

n=3

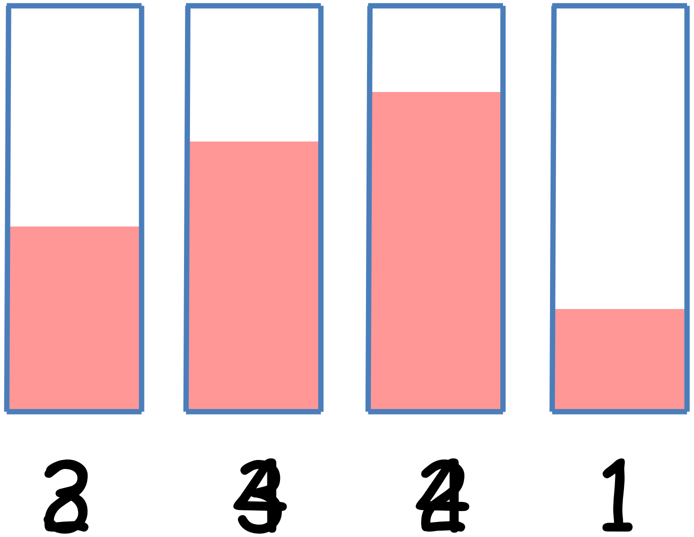
3 cycles



Transition graph, n=3



Multiple Cells Permutation



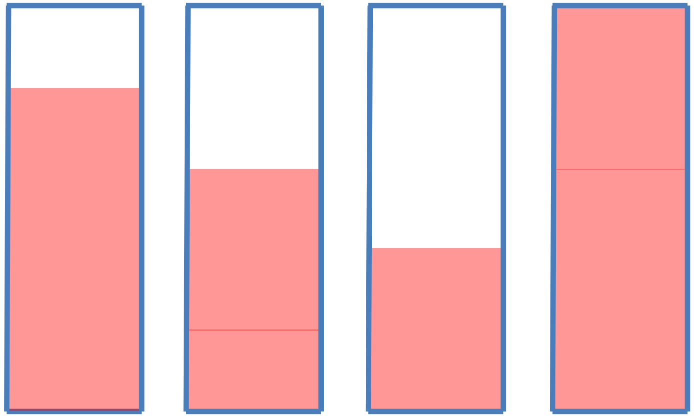
Goal: Guarantee large number of rewrites

Multiple Cells Permutation

Example:

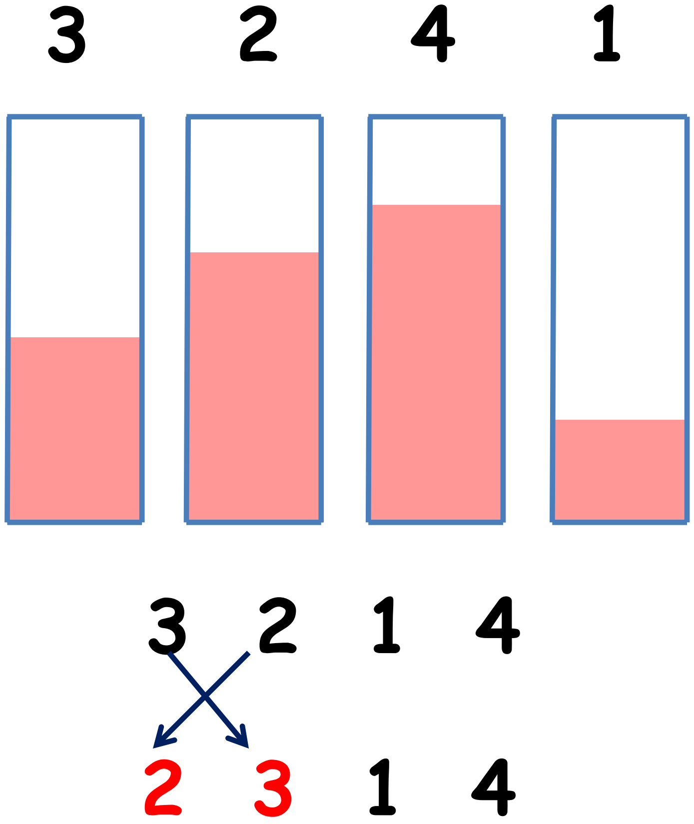
n=4 cells
q=5 levels in each cell

p = 3,2,1,4 c = 4,3,2,5
p = 1,2,3,4 c = 0,1,2,3
c = 0,0,0,0



T=2 writes

Goal: Guarantee large number of rewrites



3 2 1 4

2 3 1 4

Kendall's Tau Distance

- For a permutation σ an **adjacent transposition** is the local exchange of two adjacent elements
- For $\sigma, \pi \in S_m$, $d_\tau(\sigma,\pi)$ is the **Kendall's tau** distance between σ and π

= Number of adjacent transpositions to change σ to be π

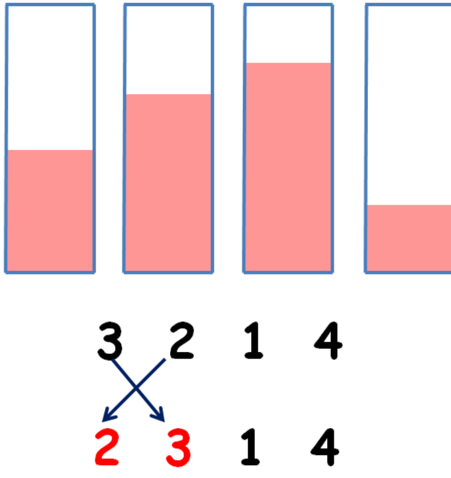
$\sigma=2413$ and $\pi=2314$

$2\underline{41}3 \rightarrow 21\underline{43} \rightarrow 2\underline{13}4 \rightarrow 2314$

$d_\tau(\sigma,\pi) = 3$

It is called also the **bubble-sort** distance

The Kendall's tau distance is the number of pairs that do not agree in their order

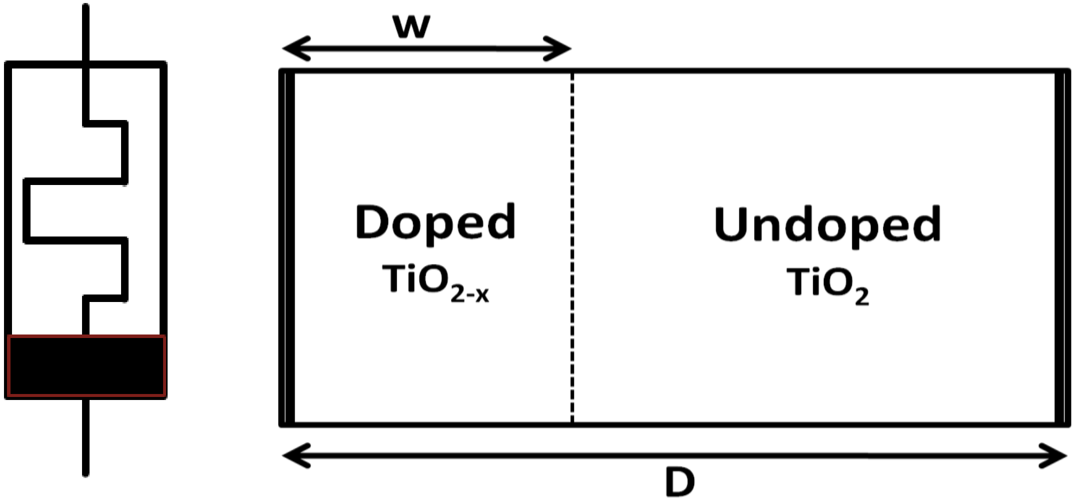


Other Types of Non-volatile Memories

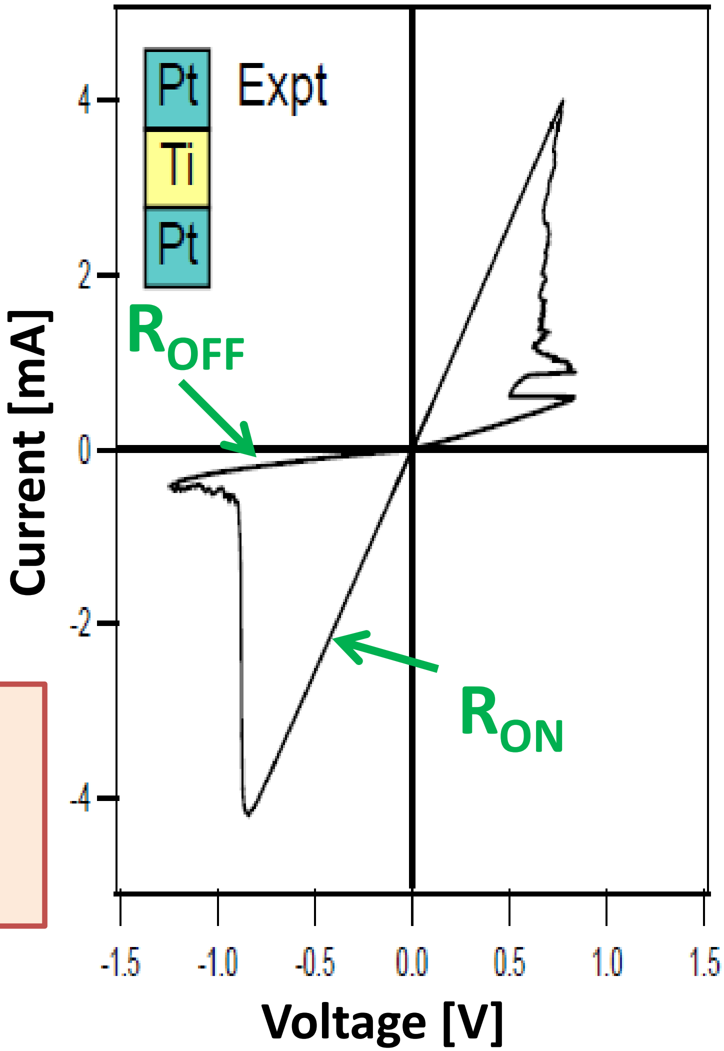
- Phase Change Memories (PCM)
- STTRAM
- MRAM
- Memristors

Practical Memristors

- 2008 Hewlett Packard

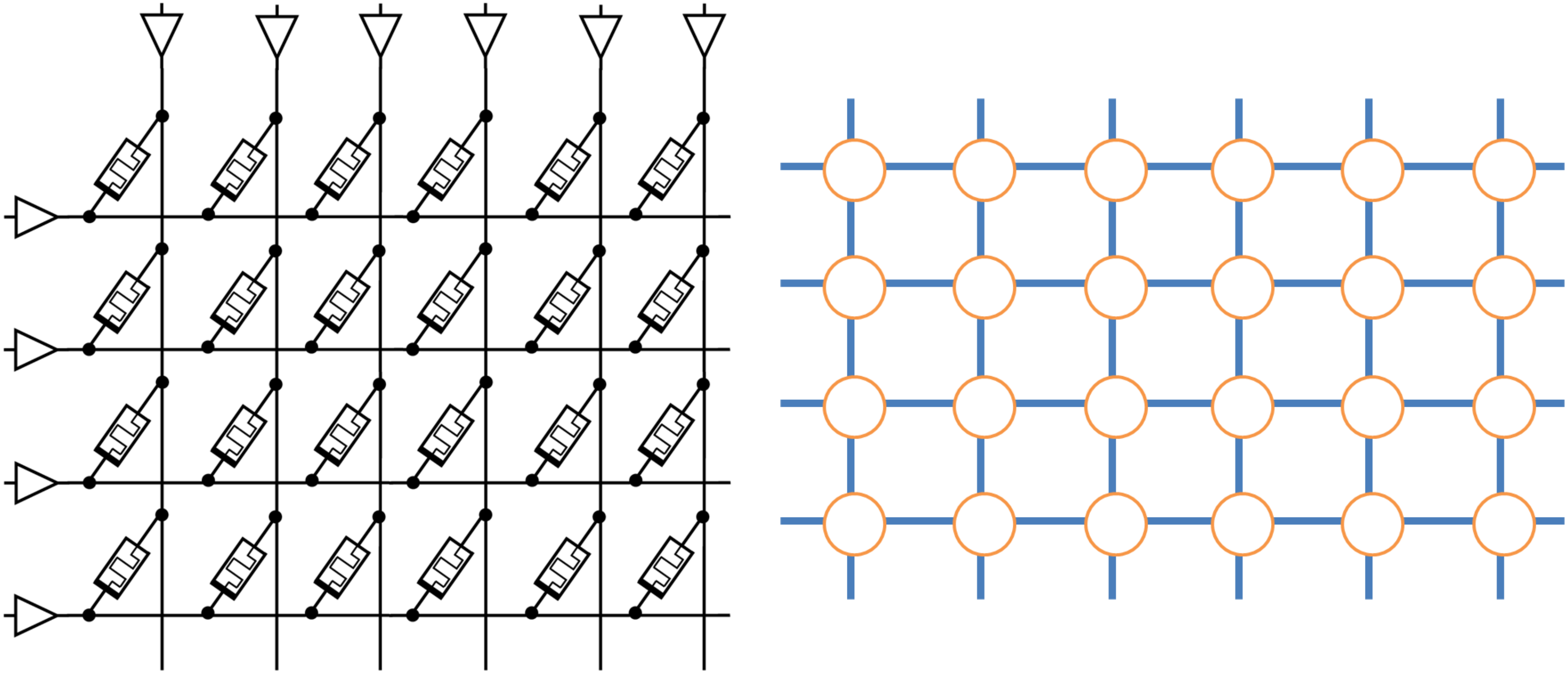


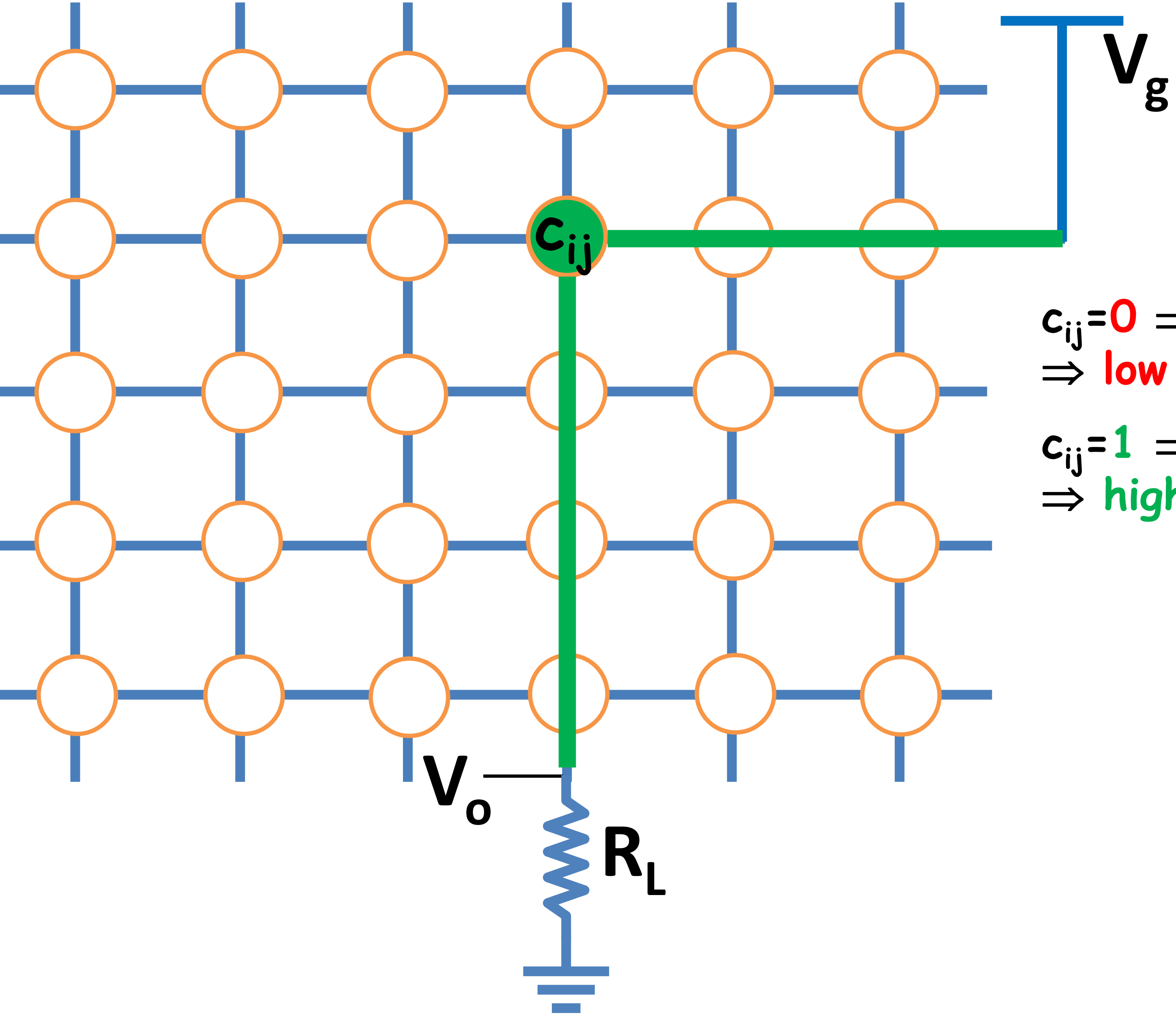
$$M(q) = R_{OFF}\left(1-\frac{\mu_v R_{ON}}{D^2}q(t)\right)$$



D.B. Strukov et al, "The missing memristor found," Nature, 2008

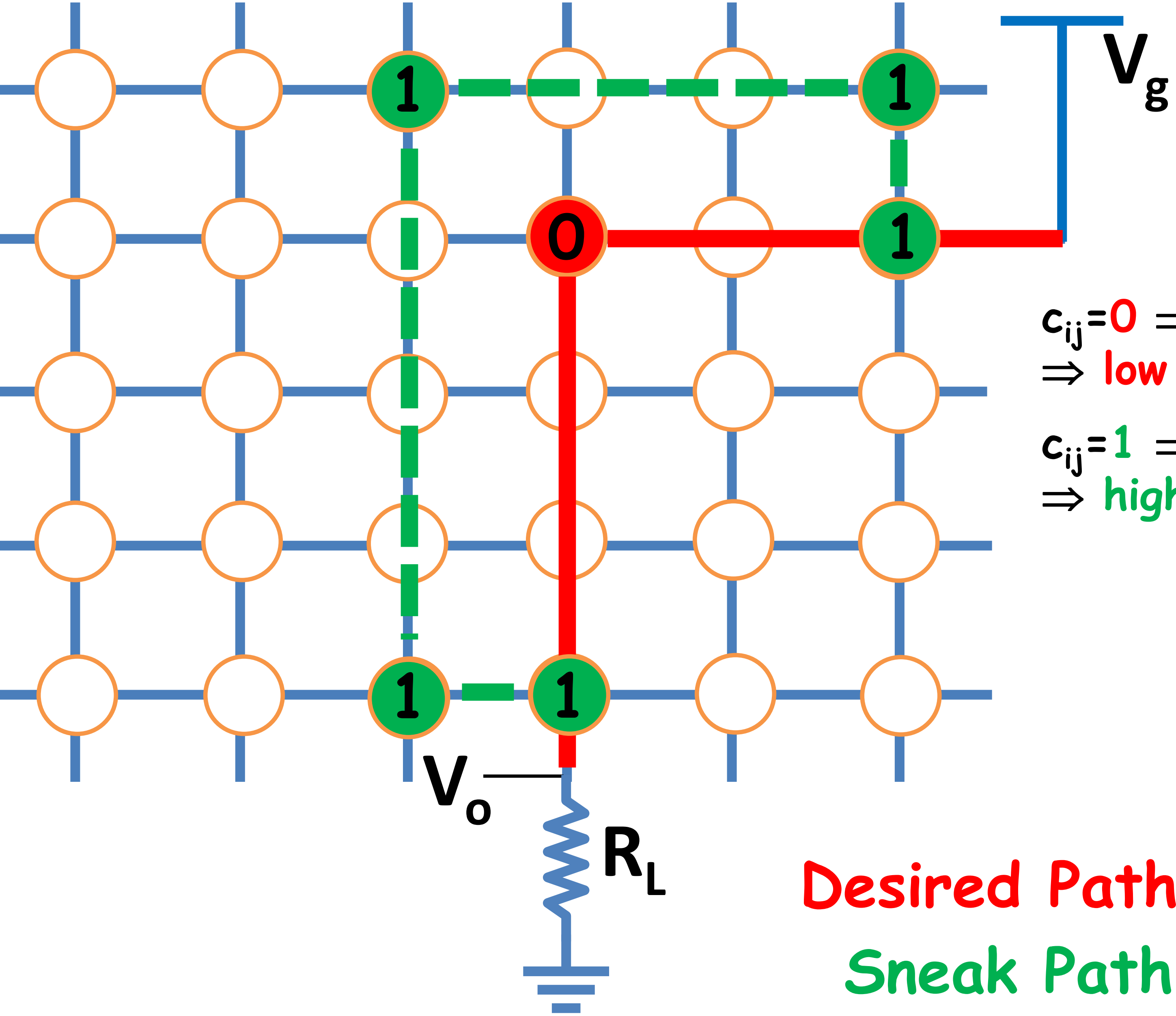
Crossbar Arrays





c_{ij}=0 ⇒ high resistance
⇒ low current sensed

c_{ij}=1 ⇒ low resistance
⇒ high current sensed



$c_{ij}=0$ ⇒ high resistance
⇒ low current sensed

$c_{ij}=1$ ⇒ low resistance
⇒ high current sensed

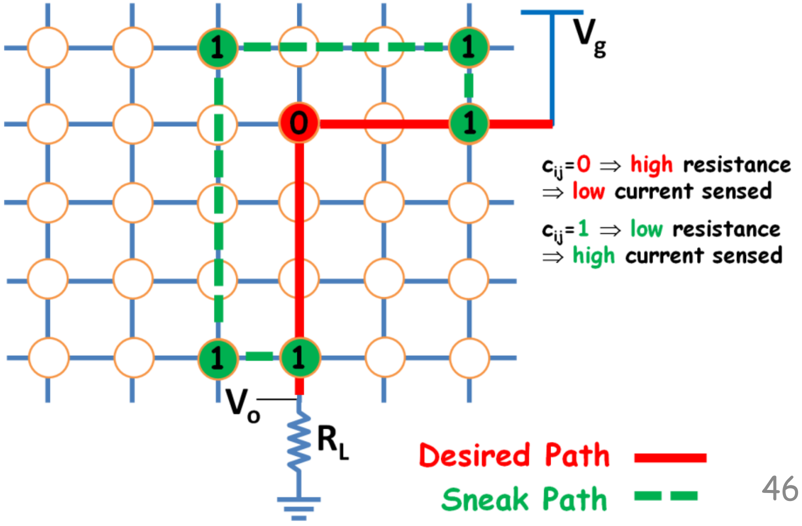
Desired Path ━━

Sneak Path ╍╍

Sneak Path

- An array **A** has a **sneak path** of length **2k+1** affecting the **(i,j)** cell if
  - $a_{ij}=0$
  - There exist $r_1,\dots,r_k$ and $c_1,\dots c_k$ such that
    $a_{ic_1} = a_{r_1c_1} = a_{r_1c_2} = \cdots = a_{r_kc_k} = a_{r_kj} = 1$
- An array A satisfies the **sneak-path constraint** if it has no sneak paths and then is called a **sneak-path free array**

		1			1
			0		1
		1	1		



$c_{ij}=0$ ⇒ high resistance
⇒ low current sensed

$c_{ij}=1$ ⇒ low resistance
⇒ high current sensed

Desired Path 🟥

Sneak Path - -