Adaptive Threshold Read Algorithms in Multi-level Non-Volatile Memories

Evyatar Hemo and **Yuval Cassuto**
Department of Electrical Engineering, Technion – Israel Institute of Technology
evyatarhemo@gmail.com, ycassuto@ee.technion.ac.il

***Abstract*— For an array of memory cells that are read by threshold measurements, we ask the question of how to choose the measurements in the read sequence to minimize the number of measurements before the array is fully read. We propose and study analytically various adaptive read algorithms, and provide a corresponding lower bound on the average number of measurements. We show that new two-dimensional read algorithms improve over the best one-dimensional ones.**

I. Introduction

Solid-state storage technologies, most prominently flash memories, constantly grow in their storage densities and become the most attractive media for many applications. One way in which improved density is achieved is by increasing the number of levels to which a cell can be programmed. Commercial products exist with 16 levels, and this number is likely to grow further in future technologies. Unfortunately, using more levels per cell is not a "magic formula" to squeeze more bits into the same hardware, but rather an action with significant ramifications on the read/write performance and reliability of the device.

There is a significant body of known work on how to represent data in multi-level flash memories, so as to endow the storage device with different goodness features. These features include error correction, rewrite capability, and write optimization. The works in that area are so numerous and diverse, that we avoid the daunting task of listing a fair set of references. One direction that is less explored theoretically is the optimization of *reading* information from multi-level memory arrays – in particular, memory arrays that are read by threshold measurements applied to a group of cells in parallel. We choose this read problem as the topic of study reported in this paper. The motivation to consider the read problem comes from a concern that with threshold reads, continued growth in the number of levels will introduce a significant toll on the read performance. The driving idea of this work is that read algorithms that only optimize for the worst-case (across information contents) read time are suboptimal, since they fail to benefit from information contents that are "easier" to read.

To understand the problem at hand, suppose we have n memory cells with q discrete levels $\{0, \ldots, q-1\}$. The cells are read by applying a sequence of threshold measurements, each applied to the n cells in parallel and returns n binary values of whether the cell levels are above or below the threshold. For small n relative to q, it may be possible to read the array content with fewer measurements than trying all $q-1$ threshold levels. For example, when $n = 4$ and $q = 8$ we intuitively feel that reading cell levels $(0, 1, 2, 1)$ is easier than reading the more spread levels $(0, 2, 4, 6)$. This intuitive feel is pursued in the paper, both algorithmically: minimizing the number of measurements, and analytically: calculating and bounding the average number of required measurements. The primary regime of operation to benefit from this work is when n is not much larger than q. This setup is the most interesting for the adaptive read problem, since the savings potential (over trivial read algorithms) is significant. This fact limits the applicability of the framework to current NAND flash technology, which uses an especially high level of parallelism. Nevertheless, it is plausible that future non-volatile memory technologies will work in a lower parallelism regime, due to technology limitations or cost issues. We also believe that threshold reading is a fundamental and interesting problem in general, worthy of the detailed study that follows. In Section II the scope is on one-dimensional (1D) read algorithms, where the set of measured cells is fixed. In Section III we move to two-dimensional (2D) algorithms, where the measured cell can be chosen with certain degrees of freedom. The results show that adaptive choice of threshold measurements can improve read performance over fixed predetermined read sequences. 2D algorithms are further shown to be superior over 1D algorithms, including over the 1D lower bound that any 1D algorithm must satisfy. In terms of prior work, the studied problem is related to the problem of adaptive sorting algorithms, but we have not found a simple way to directly apply existing knowledge to the threshold-read problem. Unless stated otherwise, all log functions are base 2.

A. Parallel threshold read model

We first give some formal definitions for the threshold-read model. Let the state of the storage cell be represented as a discrete **cell level** c, taken from the integer set $\{0, \ldots, q-1\}$.

Definition 1. *A* **threshold** τ *is an integer from the set* $\{1, \ldots, q-1\}$. *Given a threshold* τ, *a cell is said to be* **active** *with respect to* τ *if its cell level satisfies* $c \geqslant \tau$. *In the complementary case of* $c < \tau$, *the cell is said to be* **inactive** *with respect to* τ.

Note that this definition of threshold is equivalent to applying a (non-integer) threshold in the real interval $(i, i+1)$.

Definition 2. *A* **measurement** *is an operator acting on a set of cells* $\mathcal{S}$ *by applying a threshold of* τ, *and obtaining* $s = |\mathcal{S}|$ *binary values reflecting the activity of each cell in* $\mathcal{S}$ *with respect to* τ. *We denote the measurement as a vector* $M_\tau(\mathcal{S}) = (m_1, \ldots, m_s)$, *where* $m_i \in \{0, 1\}$. $m_i = 1$ *represents an active cell with respect to* τ, *and* $m_i = 0$ *represents an inactive cell.*

The accumulated information on the cell levels after some sequence of iterations is represented by *uncertainty windows*:

Definition 3. *The* **uncertainty window** *of a cell level* c *is given as a pair of integers* $[L, U]$, *if it is known that* $L \leqslant c \leqslant U$.

It is straightforward to observe that for any threshold-measurement sequence, the level of every cell is known up to a set of consecutive integer values given by $L \leqslant c \leqslant U$. When $L = U$, the cell level c is completely known (no uncertainty).

II. Read Algorithms

In a standard storage setup involving non-volatile memories, we want to determine the cell levels of a block $\mathcal{N}$ of n memory cells by applying measurements to the cell block until all levels become known. The sequence of measurements applied to the cell block is allowed to be *adaptive*, i.e., the parameters of the next measurement are chosen based on the accumulated state of the cells measured thus far. In particular, the measurement sequence will stop once all cell levels have been determined, which may happen well before reaching the number of measurements needed for the worst-case cell-level combination. We assume throughout the section that all measurements act on the full block of cells, hence $\mathcal{S} = \mathcal{N}$ and $s = n$. This assumption will be lifted in subsequent sections.

A. Sequential scan

The simplest way to determine the cell levels of all cells is by a sequential scan – applying measurements starting from $\tau = 1$ upward, and stopping when all cell levels are fully determined. In the worst case, all $q-1$ possible τ values will be used, but an earlier stop is possible if none of the n cells in the block currently store the upper levels.

Proposition 1. *The expected number of measurements needed for sequential-scan read, assuming uniform level distribution, is given by*

$$T(n,q) = (q-1) - \sum_{k=1}^{q-2} \left(\frac{k}{q}\right)^n . \tag{1}$$

Proof: If all the cells in the block have levels in $\{0,...,k-1\}$, for some $k \leqslant q-1$, then k measurements are clearly sufficient. This is because the k-th measurement, having $\tau = k$, gives $M_k(\mathcal{N}) = (0,\ldots,0)$ (all cells inactive), making all higher-threshold measurements redundant. Therefore, for every $k \leqslant q-2$, $\Pr[\#meas. \leqslant k] = (k/q)^n$ and for $k = q-1$, we trivially get $\Pr[\#meas. \leqslant q-1] = 1$. For $k \leqslant q-2$, if all levels fall in $\{0,...,k-1\}$ but not in $\{0,...,k-2\}$, then k measurements are both sufficient and necessary, and the number of *saved* measurements is $q-1-k$. The expected number of saved measurements is thus given by

$$E[q-1-\#meas.] = \sum_{k=1}^{q-2}(q-1-k)\left[\left(\frac{k}{q}\right)^n - \left(\frac{k-1}{q}\right)^n\right]. \tag{2}$$

Splitting the sum and shifting the summation indices of the second sum, (2) becomes

$$\sum_{k=1}^{q-2}(q-1-k)\left(\frac{k}{q}\right)^n - \sum_{k=1}^{q-3}(q-2-k)\left(\frac{k}{q}\right)^n = \sum_{k=1}^{q-2}\left(\frac{k}{q}\right). \tag{3}$$

Subtracting the right-hand side of (3) from $q-1$ we get the expected number of measurements in the claim (1). ■

From the expression for $T(n,q)$ in (1), it is observed that as n grows, the expected number of measurements tends to the worst case of $q-1$.

B. Binary search

A potentially better read-algorithm than sequential scan is the binary search. For reading a single cell level ($n = 1$), it is clear that the binary search, requiring $\log q$ measurements, is optimal. The binary search is the basis of a widely used analog to digital converter (ADC) family called *successive approximation register (SAR)* [1]-[3]. To use the binary search for multiple cells ($n > 1$), the following simple extension of the algorithm is needed. Recall the cell uncertainty window from Definition 3 denoted by $[L,U]$. Initially, each cell has an uncertainty window of $[0,q-1]$. After the first binary-search measurement $M_{q/2}(\mathcal{N})$ (we assume for simplicity that q is a power of two), the cells that are active will have an uncertainty window of $[q/2,q-1]$; those that are inactive will have $[0,q/2-1]$. The n-cell binary-search algorithm proceeds by finding the largest uncertainty window in $\mathcal{N}$ and applying a measurement to cut it by half. It stops when all uncertainty windows are of size 1. We give a formal specification of the algorithm in recursive presentation. In the sequel we denote by L_i and U_i the upper and lower limits, respectively, of the i-th cell in $\mathcal{N}$. Complete measurement of all the cells in $\mathcal{N}$ is achieved by calling BinarySearch$(\mathcal{N},0,q-1)$.

Algorithm 1. *BinarySearch*$(\mathcal{N},L,U)$

if $L = U$ *return*
$\tau = (L+U+1)/2$
$\boldsymbol{m} = M_\tau(\mathcal{N})$
// update uncertainty windows
For all i *with* $m_i = 0$, *set* $U_i = \min(U_i, \tau-1)$
For all i *with* $m_i = 1$, *set* $L_i = \max(L_i, \tau)$
// recursive calls
if $\exists i : m_i = 0, U_i \geqslant L$ *then BinarySearch*$(\mathcal{N},L,\tau-1)$
if $\exists i : m_i = 1, L_i \leqslant U$ *then BinarySearch*$(\mathcal{N},\tau,U)$

A recursive BinarySearch call in Algorithm 1 is invoked (if and) only if there is a cell in the corresponding sub-interval. If there is no cell i whose uncertainty window overlaps with $[L,\tau-1]$, then the call BinarySearch$(\mathcal{N},L,\tau-1)$ is skipped. Similarly, if there is no cell i whose uncertainty window overlaps with $[\tau,U]$, then the call BinarySearch$(\mathcal{N},\tau,U)$ is skipped. These skipped intervals result in saved measurements. In the extreme case of $n = 1$, all measurements return (degenerate, size 1) all-0 or all-1 vectors, and it is always the case that only one sub-interval is chosen in the recursion. We now turn to analyze the expected number of measurements applied by the BinarySearch algorithm.

Proposition 2. *Let* $F(n,\log_2 q)$ *be the expected number of measurements needed for binary-search read, assuming uniform level distribution. Then* $F(n,\log_2 q)$ *can be calculated by the recursive formula*

$$F(n,l) = \sum_{i=0}^{n} \frac{\binom{n}{i}}{2^n}\left(1 + F(i,l-1) + F(n-i,l-1)\right), \tag{4}$$

where $F(n,l) = 0$ *if either* $n = 0$ *or* $l = 0$.

An explicit analytic expression for $F(n,l)$ *is given by*

$$F(n,l) = \sum_{k=0}^{l-1} 2^k\left[1 - \left(1 - \frac{1}{2^k}\right)^n\right]. \tag{5}$$

Proof: Proving the recursive formula in (4) is straightforward. Each entry to BinarySearch in Algorithm 1 applies one measurements, and calls BinarySearch again recursively to measure the i inactive cells at the lower sub-interval, and the $n-i$ active cells at the upper sub-interval. Note that when either i or $n-i$ are zero, one recursive call is skipped, which is captured by setting $F(0,l) = 0$. The probability of an $i, n-i$

split between active and inactive cells is determined to be $\binom{n}{i}/2^n$ by the uniform distribution.

The closed-form expression in (5) is now proved by induction on l. For the induction base we verify that (5) gives $F(n,0) = 0$, as required by the initial conditions. We now assume that (5) is correct for $l-1$, and prove it for l. First we simplify (4) using symmetry to get

$$F(n,l) = 1 + \sum_{i=0}^{n} \frac{\binom{n}{i}}{2^{n-1}} F(i, l-1). \tag{6}$$

Substituting (5) in the right-hand side of (6) and reorganizing gives

$$F(n,l) = 1 + \frac{1}{2^{n-1}} \sum_{k=0}^{l-2} 2^k \sum_{i=0}^{n} \binom{n}{i} \left[1 - \left(1 - \frac{1}{2^k}\right)^i\right]. \tag{7}$$

The inner sum can now be written as

$$\sum_{i=0}^{n} \binom{n}{i} \left[1 - \left(1 - \frac{1}{2^k}\right)^i\right] = 2^n - \left(2 - \frac{1}{2^k}\right)^n = 2^n \cdot \left[1 - \left(1 - \frac{1}{2^{k+1}}\right)^n\right]. \tag{8}$$

Substituting (8) back in (7) now gives

$$F(n,l) = 1 + \sum_{k=0}^{l-2} 2^{k+1} \left[1 - \left(1 - \frac{1}{2^{k+1}}\right)^n\right] = \sum_{k=0}^{l-1} 2^k \left[1 - \left(1 - \frac{1}{2^k}\right)^n\right],$$

where the last equality is obtained by shifting the summation index and inserting the 1 into the sum.

■

C. Lower bound on the average number of measurements

For the purpose of evaluating the efficiency of the simple read algorithms of Sections II-A and II-B, we now derive a lower bound on the average number of measurements. Given a size-n block $\mathcal{N}$ of q-ary cells, we wish to find a lower bound $LB(n,q)$ defined in the following.

Definition 4. *$LB(n,q)$ is called a* **lower bound** *if any read algorithm for n q-ary cells requires on average at least $LB(n,q)$ measurements.*

To obtain such a bound, the key observation we make is that given an assignment of levels to the n cells $(c_1, \ldots, c_n)$, any read algorithm must apply a measurement in every threshold level that appears as one of the c_i, and also in every threshold level that is immediately above one of the c_i. If at least one of these two measurements is missing, then the corresponding cell remains with an uncertainty window of size at least two. More formally, we have the following definitions.

Definition 5. *Given a vector of cell levels $\boldsymbol{c} = (c_1, \ldots, c_n)$, with $c_i \in \{0, \ldots, q-1\}$, define the* **incidence set** *as the set $\mathcal{I}(\boldsymbol{c}) = \{s \in \{1, \ldots, q-1\} | \exists i,\ c_i = s\}$. The* **shifted incidence set** *is defined as $\mathcal{I}^*(\boldsymbol{c}) = \{s \in \{1, \ldots, q-1\} | \exists i,\ c_i + 1 = s\}$.*

Therefore, we have the following.

Proposition 3. *For a given cell-level vector $\boldsymbol{c}$, a lower bound on the number of measurements is given by $|\mathcal{I}(\boldsymbol{c}) \cup \mathcal{I}^*(\boldsymbol{c})|$.*

Observe that $|\mathcal{I}(\boldsymbol{c}) \cup \mathcal{I}^*(\boldsymbol{c})| \leqslant \min(2n, q)$.

Example 1. *For the following $q = 8$, $n = 4$ cell-level vector $\boldsymbol{c} = (2,2,4,5)$, we have $\mathcal{I}(\boldsymbol{c}) = \{2,4,5\}$ and $\mathcal{I}^*(\boldsymbol{c}) = \{3,5,6\}$. Since $\mathcal{I}(\boldsymbol{c}) \cup \mathcal{I}^*(\boldsymbol{c}) = \{2,3,4,5,6\}$, the lower bound is 5.*

In order to obtain an analytic lower bound for the *average* number of measurements, we need to find the expectation of $|\mathcal{I}(\boldsymbol{c}) \cup \mathcal{I}^*(\boldsymbol{c})|$ over uniformly distributed vectors $\boldsymbol{c} \in \{0, \ldots, q-1\}^n$. As seen in Proposition 3, the lower bound for a particular $\boldsymbol{c}$ depends on both the size of $\mathcal{I}(\boldsymbol{c})$ (how many levels appear), and the overlap between $\mathcal{I}(\boldsymbol{c})$ and $\mathcal{I}^*(\boldsymbol{c})$ (how many levels in the union serve as both incident and shifted). It can be seen that

$$|\mathcal{I}(\boldsymbol{c}) \cup \mathcal{I}^*(\boldsymbol{c})| = |\mathcal{I}(\boldsymbol{c})| + L(\boldsymbol{c}),$$

where $L(\boldsymbol{c})$ is the number of runs of consecutive levels in $\mathcal{I}(\boldsymbol{c})$. In Example 1, for $\boldsymbol{c} = (2,2,4,5)$ we have $L(\boldsymbol{c}) = 2$, since the set $\mathcal{I}(\boldsymbol{c}) = \{2,4,5\}$ can be split into two runs: $\{2\}$ and $\{4,5\}$. The number of runs captures the number of necessary measurements, because in each run only the last level contributes an element to $\mathcal{I}^*$ not already appearing in $\mathcal{I}$.

To calculate the distribution of $L(\boldsymbol{c})$, we first regard the set $\mathcal{I}(\boldsymbol{c})$ as a length-q binary indicator vector whose i-th entry is 1 if $i \in \mathcal{I}(\boldsymbol{c})$. We then define the combinatorial object $D(q,l,L)$ to be the number of length-q binary vectors that have l ones falling into exactly L runs of consecutive coordinates. To handle the special extreme cases of levels 0 and $q-1$, we further refine $D(q,l,L)$ to $D_0(q,l,L)$, $D_1(q,l,L)$ and $D_2(q,l,L)$, where $D_j(q,l,L)$ is the number of (q,l,L) vectors that have a one on j of the locations 0 and $q-1$. This refinement is needed because the elements 0 and $q-1$ are special in that a run that contains them necessitates one fewer measurement than a run that does not touch the edges. Based on [4], we now write the closed-form expressions for $D_j(q,l,L)$.

$$D_0(q,l,L) = \binom{l-1}{L-1}\binom{q-l-1}{L}, D_1(q,l,L) = 2\binom{l-1}{L-1}\binom{q-l-1}{L-1},$$
$$D_2(q,l,L) = \binom{l-1}{L-1}\binom{q-l-1}{L-2}.$$

We are now ready to present a lower bound on the average number of measurements.

Theorem 4. *A lower bound on the average number of measurements given uniformly distributed cell levels is given by*

$$LB(n,q) = \frac{1}{q^n} \sum_{k=1}^{n} k! \cdot S(n,k) \cdot \sum_{L=1}^{k} \sum_{j=0}^{2} D_j(q,k,L) \cdot (k + L - j),$$

where $S(n,k)$ is the Stirling number of the second kind [6].

Proof: Every vector $\boldsymbol{c} \in \{0, \ldots, q-1\}^n$ can be uniquely obtained by choosing a size-k set $\mathcal{I}(\boldsymbol{c})$, and then applying a surjection from the n-set of coordinates to the k-set $\mathcal{I}(\boldsymbol{c})$. It is well known [6] that the number of surjections from an n-set to a k-set equals $k!S(n,k)$. The number of necessary measurements depends only on k and the number of runs in the set $\mathcal{I}(\boldsymbol{c})$. The two inner sums count all choices of size-k sets $\mathcal{I}(\boldsymbol{c})$, and weight each choice with its corresponding number of necessary measurements $k + L - j$. The number of necessary measurements does not depend on the particular surjection applied to $\boldsymbol{c}$, and hence the numbers of surjections appear at the outer sum. The overall sum, normalized by the number of vectors q^n, gives the expected number of necessary measurements given the uniform distribution on $\boldsymbol{c}$. It is clear that any read algorithm will have at least the number $k + L - j$ of necessary measurements on every input $\boldsymbol{c}$, and therefore on average must apply no less than $LB(n,q)$ measurements. ■

To better understand the proof of Theorem 4, we give an example of a mapping between vectors $\boldsymbol{c} \in \{0, \ldots, q-1\}^n$ and incidence sets $\mathcal{I}$.

Example 2. *Suppose $n = 3$, $q = 8$, and we have the incidence set $\mathcal{I} = \{2, 5\}$ (with size $k = 2$). The vectors $\boldsymbol{c}$ that map to $\mathcal{I}$ are the 6 vectors $(2,2,5)$, $(2,5,2)$, $(5,2,2)$, $(5,5,2)$, $(5,2,5)$, $(2,5,5)$. Substituting $n = 3$, $k = 2$ in $k!S(n,k)$ indeed gives 6. All these $\boldsymbol{c}$ vectors will have the same number of necessary measurements, which depends solely on $\mathcal{I}$.*

D. Performance comparison

To summarize the section, we take the analytic expressions for the average measurement counts of two read algorithms: sequential scan ($T(n,q)$ in Section II-A) and binary search ($F(n, \log q)$ in Section II-B), and plot their values in comparison with the lower bound ($LB(n,q)$ in Section II-C). The case of fixed $n = 4$ is shown in Fig. 1. The average numbers of measurements are plotted as a function of $\log(q)$.

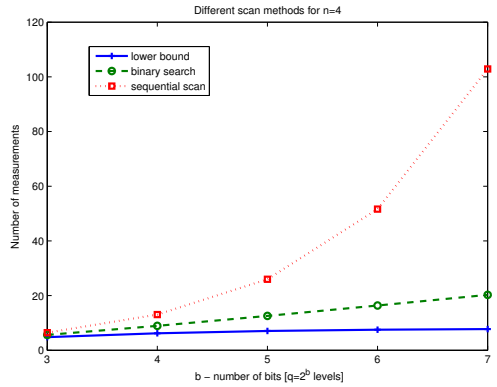

Figure 1. Analytic average measurement counts for $n = 4$: sequential scan (squares), binary search (circles), and the lower bound (crosses).

It is observed in Fig. 1 that the read complexity of sequential search grows exponentially in $\log(q)$. In contrast, binary search grows more gracefully. The lower bound grows even slower than binary search, converging to $2n = 8$.

Another interesting case to examine is when n and q grow together, while maintaining a fixed ratio. Fig. 2 shows the results for $n = q/2$. Here the three curves follow a similar shape, but with widening gaps as q grows. The fixed-ratio case is important because it makes n small enough to have "easy" read instances that improve the average counts (when n is very large, with high probability all q levels will be used, and sequential scan is optimal), but n is also large enough to motivate advanced read algorithms (when n is very small, e.g. $n = 1$, the binary search is likely close to optimal). Our objective in the remainder of the paper is to improve over binary search using more advanced algorithms.

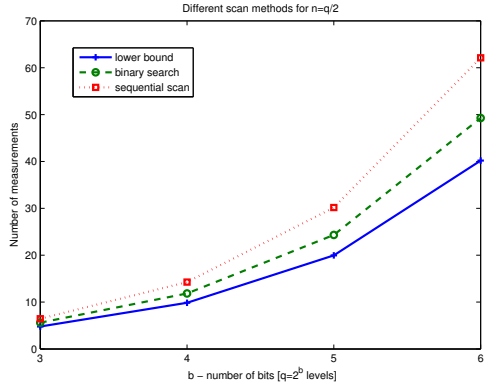

Figure 2. Analytic average measurement counts for $n = q/2$.

III. Two-Dimensional Read Algorithms

In order to improve on the average number of measurements, we will move to a more realistic storage setup where the cells are organized as a two-dimensional (2D) array, and each measurement is applied to a subset of the memory cells. In the 2D problem, the memory block $\mathcal{N}$ is an array of q-ary cells with dimensions $n \times n$. The set of cells $\mathcal{S}$ to which a measurement $M_\tau(\mathcal{S})$ is applied is no longer the full set $\mathcal{N}$, but a set of size n. Setting the measurement set size to n allows comparisons between the proposed 2D algorithms and using the 1D algorithms of Section II row-by-row. We start the discussion of 2D read algorithms with a motivating example.

Example 3. *Suppose we read the following 2×2, $q = 8$ array row-by-row.*

1	*2*
0	*3*

Then from Proposition 3 we know that we need for the top row at least 3 measurements: $\mathcal{I}((1,2)) \cup \mathcal{I}^*((1,2)) = \{1,2,3\}$, and for the bottom row at least 3 measurements as well: $\mathcal{I}((0,3)) \cup \mathcal{I}^*((0,3)) = \{1,3,4\}$. In total we need 6 measurements.

*Alternatively, if we can choose whether to measure a row **or** a column, it is possible to reduce the number of measurements to 4. We first measure the top row with $\tau = 2$, then we measure the left column with $\tau = 1$. At this point the 1 and 0 in the left column are fully known, and the 2 on the top right is known to be $\geqslant 2$. Now we measure the right column with $\tau = 3$ and $\tau = 4$, after which the 2 and 3 on the right are also known.*

By showing a gap in the lower bound on the number of measurements, Example 3 shows that a 2D algorithm can *in principle* improve over a 1D algorithm. However, it is still not clear how to reduce the number of measurements *algorithmically* using 2D flexibility. In the remainder of the section we address the algorithmic 2D read problem. We first define the general optimization problem formally.

Problem 5. *Given a 2D memory array $\mathcal{N}$, find a minimal-length sequence of measurements $M_{\tau_1}(\mathcal{S}_1), M_{\tau_2}(\mathcal{S}_2), \ldots$ that reads all the cells in $\mathcal{N}$. The sets $\mathcal{S}_j$ are of size n, and are chosen according to some prescribed specifications. The choice of $\mathcal{S}_j$ and τ_j may depend on the outcomes of preceding measurements.*

A. CRDF algorithm

In an $n \times n$ array $\mathcal{N}$, suppose the prescribed specifications are that $\mathcal{S}_j$ is either a row or column of $\mathcal{N}$. For this model we now present the *columns and rows degrees of freedom (CRDF)* algorithm. The core of the CRDF algorithm is a criterion that chooses a row or a column and a threshold τ_j, based on the current uncertainty state of the array cells. The objective of the implemented criterion is to read $\mathcal{N}$ at full with a short measurement sequence.

The criterion that we choose for the CRDF algorithm is minimization of the *expected uncertainty* after the measurement. The idea behind this criterion (defined shortly), is to greedily choose the measurement that makes the largest step toward eliminating the uncertainty in the array.

Definition 6. *The* **uncertainty** Ω *of a cell with uncertainty window* $[L, U]$ *is defined as* $\Omega = \log(U - L + 1)$.

The uncertainty of a cell ranges from $\log q$ initially (before any measurement) to 0 when $L = U$ (level fully determined). Let $[L_i, U_i]$ be the uncertainty window of cell i, with its uncertainty value denoted Ω_i. It is easy to see that if $L_i < \tau \leqslant U_i$, then after the measurement the uncertainty window of cell i will be $[L_i, \tau - 1]$ if $c_i < \tau$ and $[\tau, U_i]$ if $c_i \geqslant \tau$. Hence we can find the expected uncertainty of cell i after a measurement with threshold τ to be

$$H_i(\tau) = \Pr(c_i < \tau)\log(\tau - L_i) + \Pr(c_i \geqslant \tau)\log(U_i - \tau + 1) = \frac{\tau - L_i}{U_i - L_i + 1}\log(\tau - L_i) + \frac{U_i - \tau + 1}{U_i - L_i + 1}\log(U_i - \tau + 1), \quad (9)$$

where (9) follows by a simple substitution of the uniform distribution into the probabilities above. Another way to write $H_i(\tau)$ is now given using the binary entropy function.

$$H_i(\tau) = \Omega_i - h\left(\frac{\tau - L_i}{U_i - L_i + 1}\right), \quad (10)$$

where $h(p) = -p\log p - (1-p)\log(1-p)$ is the binary entropy function. Hence the criterion for choosing a legal $\mathcal{S}$ (row/column in CRDF) and a corresponding τ is set to be

$$\underset{\mathcal{S},\tau}{\operatorname{argmax}} \sum_{i \in \mathcal{S}} h\left(\frac{\tau - L_i}{U_i - L_i + 1}\right).$$

Note that this is a generalization of the 1D binary search algorithm, for which $\mathcal{S}$ is fixed and τ is chosen as the mid point between L_i and U_i to reduce the uncertainty by 1 bit.

B. Results

A more flexible version of the CRDF algorithm is the *any n degrees of freedom (ANDF)* algorithm, in which $\mathcal{S}_j$ can be any n arbitrary cells out of $\mathcal{N}$ (and not just a row or a column as in CRDF). The performance of the 2D algorithms is now evaluated by simulation. A 4×4 memory array with uniformly distributed level assignment was read by both the CRDF and ANDF algorithms 1000 times, providing the results presented in Fig. 3. The average number of measurements, normalized by $n = 4$, of CRDF and ANDF are marked as a function of q. The normalization by n allows to compare the results with the 1D binary search and lower bound, also marked on the same plot. We first observe that CRDF improves over the 1D binary search for all q, with a growing gap. ANDF is clearly superior to CRDF, given its increased flexibility. Note that the 1D lower bound (solid line) does not apply to the 2D algorithms, since the necessary measurements within a row can be shared by multiple rows, e.g. in a column measurement. Indeed ANDF outperforms the 1D lower bound for low q.

The results for a fixed ratio $n = q/2$ are shown in Fig. 4.

As we can see, for $n = q/2$ the ANDF significantly outperforms both CRDF and the lower bound, which is a motivation to implement more flexible measurement-set selection.

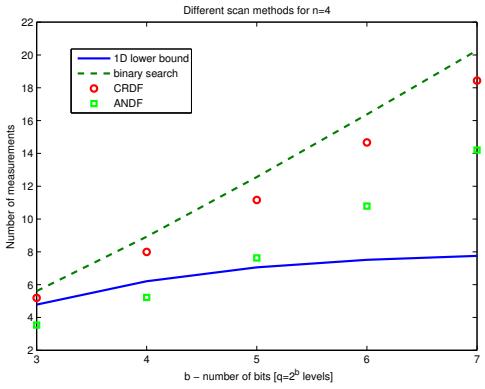

Figure 3. Simulated average measurement counts (normalized by n) for 4×4: lower bound (solid), binary search (dashed), CRDF (circle markers), and the ANDF (square markers).

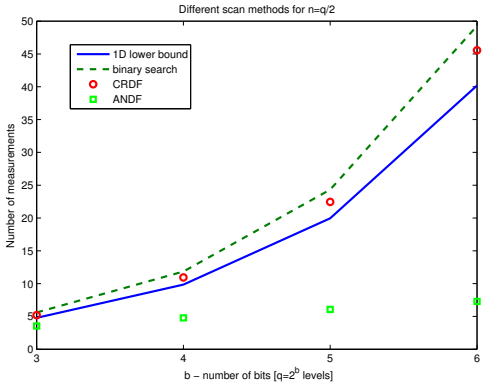

Figure 4. Average measurement counts (normalized by n) for $n = q/2$.

IV. Conclusion

Adaptive threshold read algorithms can reduce the number of measurements required to read a memory array. In particular, 2D reading algorithms were shown to improve over pure 1D measurements.

V. Acknowledgment

This work was supported in part by the Marie Curie CIG grant and by the Technion Funds for Security Research.

References

[1] R.J. Baker, CMOS Circuit Design, Layout, and Simulation, Third Edition, Wiley-IEEE, 2010. ISBN 978-0-470-88132-3.
[2] Ogawa, Tomohiko, et al. "SAR ADC algorithm with redundancy and digital error correction. ", IEICE transactions on fundamentals of electronics, communications and computer sciences 93.2 (2010): 415-423.
[3] Lin, Ying-Zu, et al. "An asynchronous binary-search ADC architecture with a reduced comparator count.", Circuits and Systems I: Regular Papers, IEEE Transactions on 57.8 (2010): 1829-1837.
[4] Y. Cassuto and M. Blaum, "Codes for symbol-pair read channels", IEEE Transactions on Information Theory, Vol 57, No. 12, December 2011.
[5] A. Mohr and T.D. Porter, "Applications of Chromatic Polynomials Involving Stirling Numbers", Department of Mathematics Southern Illinois University, 2008.
[6] J. van Lint and R. Wilson, *A Course in Combinatorics, second edition*. Cambridge UK: Cambridge University Press, 2001.