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

Amit Solomon
The Viterbi Department of Electrical Engineering
Technion - Israel institute of Technology
email: samitsolom@gmail.com

Yuval Cassuto
The Viterbi Department of Electrical Engineering
Technion - Israel institute of Technology
email: ycassuto@ee.technion.ac.il

***Abstract*—For an array of non-volatile memory cells read by threshold measurements, read performance is determined by the number of measurements needed to read the memory word. In this work we wish to explore the time savings when the measurement sequence is allowed to terminate with some prescribed amount of uncertainty. This residual uncertainty will be treated by a specially designed code. The model we introduce in this work allows the read process to terminate the measurement sequence when up to T cells have their levels known only up to a set of W possible levels. For this model we propose efficient read algorithms, we analyze the expected number of read measurements, and plot results for interesting parameters. Our results show that allowing uncertainty in the readout and cleaning this uncertainty with a special code gives a better read performance compared to a non-redundant representation that is read at full without uncertainty.**

I. Introduction

Non-Volatile Memory (NVM) based storage devices become more and more attractive, as their density and read/write throughput continue to scale. In order to increase the density of NVM-based storage devices, multi-level cells are used. Using more levels in each cell improves the storage density, but has an adverse effect on the read and write speeds. In this work, we propose a novel method to speed up the storage device's readout. A faster physical readout is achieved by allowing uncertainty in the returned values. This uncertainty will be subsequently resolved by a proper error-correcting code. The focus of this paper is developing read algorithms that minimize read time given the allowed uncertainty, and analyzing the performance advantage of these algorithms. The memory model we consider here is one in which n memory cells of q levels are read in parallel using a sequence of threshold measurements. The sequence of measurements is complete when all the cell levels are known, or as we propose in this paper, when a prescribed level of (un)certainty is reached. The main conclusion from this work is that the information read rate can be higher if we implement an imperfect physical readout and a code, compared to a perfect physical readout and no code. Our results are practically interesting when the number of cells n is not very large, hence this work is applicable to emerging memory technologies with a relatively low level of parallelism.

A. System Model

Based on [1], let us define the model formally.

Storage device: A storage device comprises elementary storage units called *cells*. Each cell stores a *level* $v \in \{0, ..., q-1\}$. There are n cells in a memory word. The n values stored in a memory word are arbitrary, but for the sake of performance analysis we will assume that they are i.i.d with the uniform distribution over the set $\{0, ..., q-1\}$.

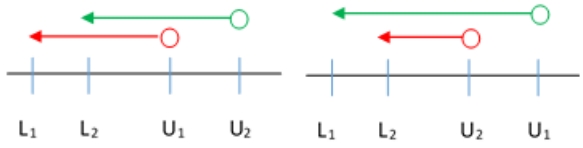
Thresholds and measurements: A *threshold* is an integer $\tau \in \{1, ..., q-1\}$ with the following function on cells. A cell is said to be *active* with respect to τ if its value is greater than or equal to τ, i.e., $\tau \leq v$, and *inactive* otherwise. A *measurement* is an operator applying a threshold τ on a given set of cells. Throughout the paper we assume that a measurement is applied to all the n cells in the word in parallel, and returns n active/inactive outputs.

Windows: A *window* is a set of two values $L < U$. A cell is said to be within a window if $L \leq v < U$. At the beginning of the reading process, each cell is within the window $[0, q)$. Applying a measurement with threshold $\tau \in \{L, ..., U-1\}$ on a set of cells having a window $[L, U)$ results in two windows: all the cells that are active with respect to τ would result in the window $[\tau, U)$, and the other cells in the window $[L, \tau)$ (note that if $\tau \notin \{L, ..., U-1\}$, the window remains unchanged). A cell's *uncertainty* is denoted by w and defined as $w \equiv U - L$. Note that a cell's value is known without uncertainty iff $w = 1$.

The goal of this work is as follows: given a system with n cells, whose values are in $\{0, ..., q-1\}^n$, we wish to devise an algorithm that by applying a minimal number of measurements on the cells, would result in all n cells being within windows whose uncertainty is 1, except for no more than T cells whose uncertainty is no more than W. T and W are system parameters. Note that in the base case where no uncertainty is allowed (i.e., $T = 0$), [1] used a read sequence for n cells based on a parallel version of the binary search, and analyzed its average number of measurements until full readout of all cells. We review the n-cell binary search from [1] as an important background for our algorithms addressing $T > 0$.

***n*-cell binary search:** The idea of the algorithm proposed in [1] is the following. For q that is a power of 2, start by

978-1-5090-2152-9/16/$31.00



(a) Case 1 (b) Case 2

Fig. 1: Example of different cases shown in claim 1.

measuring at threshold level $q/2$. Then proceed recursively: for each window of at least one cell, bisect the window by applying a measurement at its central level $(U-L)/2$ if its uncertainty is greater than 1. Stop when all cells are within windows whose uncertainty is 1. The n-cell binary search differs from the classical single-cell binary search in that in some iterations it may require halving *both* the upper and the lower intervals.

B. General claims

We now wish to prove a few claims that will be used throughout this paper.

Claim 1. Window independence: Suppose $win_1 = [L_1, U_1), win_2 = [L_2, U_2)$ are two windows that were obtained by parallel measurement of an n cell word with q levels, i.e., at the beginning of the read all n cells start at the same window $[0, q)$. Then either $win_1 \cap win_2 = win_1 = win_2$ or $win_1 \cap win_2 = \emptyset$. This means that different windows are disjoint, and can be dealt with separately.

Proof: Suppose by contradiction that the above condition is not met. There are two cases:

Case 1: $win_1 \cap win_2 \neq \emptyset$ and neither window contains the other: $win_1 \subseteq win_2$ does not hold, $win_2 \subseteq win_1$ does not hold. Without loss of generality, assume $L_1 < L_2 < U_1 < U_2$ (see Fig. 1a). From win_2 we know that L_2 was a threshold, therefore win_1 is impossible, since either U_1 would decrease to L_2 and the two windows would be disjoint, or L_1 would increase to L_2. By repeating this with U_1, we would get $win_1 = win_2$ or $win_1 \cap win_2 = \emptyset$, which contradicts the assumption.

Case 2: $win_1 \subset win_2$ or $win_2 \subset win_1$. Without loss of generality, assume $L_1 < L_2 < U_2 \leq U_1$ (see Fig. 1b). From win_2 we know that L_2 was a threshold, therefore win_1 is impossible, since either U_1 would decrease to L_2 and the two windows would be disjoint, or L_1 would increase to L_2. By repeating this with U_2 we would again get $win_1 = win_2$, or $win_1 \cap win_2 = \emptyset$. ■

II. Read Algorithms for $T = 1$

Let us discuss read algorithms under the following assumptions: $q > 1$ is power of 2, and $T = 1$, i.e., at the end of the reading process, all cells are known without uncertainty, except for up to 1 cell that is allowed to have uncertainty of no more than W. We distinguish between two cases: W that is a power of 2, and W that is not a power of 2. We discuss these two cases in the following two sub-sections.

A. W is a power of 2

The read algorithm we use is a variant of the n-cell binary search of [1] (see Section I-A above) that is adapted to allowing $T = 1$ uncertainty. The change we apply to the algorithm is a condition that the first encountered window of exactly one cell, having uncertainty W or less, is skipped from further measuring within it. By that we reduce the number of required measurements, as stated quantitatively in the following theorem.

Theorem 2. *When $W = 2^m$ for some positive integer m, the expected number of measurements that the T=1 algorithm above saves relative to the full n-cell binary search is given by*

$$\sum_{r=1}^{m}\left[\sum_{k=1}^{q/w}(-1)^{k-1}\binom{n}{k}\binom{q/w}{k}\cdot\frac{k!}{(q/w)^k}\left(\frac{q/w-k}{q/w}\right)^{n-k}\right]$$

where in the inner sum $w \equiv 2^r$.

To prove the theorem, we need the following lemmas.

Lemma 3. *A binary search allowed $T = 1$ uncertainty of $W = 2^m$ levels will save $r \leq m$ measurements relative to full n-cell binary search if for $w = 2^r$, there exists an interval of levels of the form $[iw, (i+1)w)$ that has exactly one cell in it, and if r is the largest integer for which such an interval exists.*

Proof: At each iteration, the n-cell binary search bisects intervals of size 2^r, where r is decreasing with the growing iteration count. For $r \leq m$, if there exists an interval $[i2^r, (i+1)2^r)$ with exactly one cell, the modified binary search will skip further bisecting this interval. Without allowed uncertainty the algorithm must continue bisecting this interval until resolving the precise level of the cell, and this costs r more measurements by the classical single-cell binary search algorithm. ■

Example 4. For $q = 8$ and $n = 6$, suppose that the cell levels are given by the vector $(1, 0, 3, 2, 6, 1)$. The standard n-cell binary search will first measure the threshold level $q/2 = 4$. Then for the window $[0, 4)$ the recursion will measure the threshold levels $2, 1, 3$. For the window $[4, 8)$ the recursion will measure the threshold levels $6, 7$ (no need to measure 5 because there are no cells in $[4, 6)$). These last two measurements will be skipped by the modified algorithm allowed uncertainty. The cell vector has an interval of 4 levels $[4, 5, 6, 7]$ with only one cell, and this interval is the maximal one having the form $[iw, (i+1)w)$, $w = 4$, $i = 1$. Lemma 3 indeed states that the modified algorithm saves $r = \log_2 w = 2$ measurements.

Lemma 5. *For $w = 2^r$ with r a positive integer, the probability that a uniformly selected cell-level vector has an interval of the form $[iw, (i+1)w)$ with one cell equals*

$$\sum_{k=1}^{q/w}(-1)^{k-1}\left(\begin{array}{c} n \\ k \end{array}\right)\left(\begin{array}{c} q/w \\ k \end{array}\right)\frac{k!}{(q/w)^k}\left(\frac{q/w-k}{q/w}\right)^{n-k}. \quad (1)$$

Proof: We use the balls-and-bins model [2], with each bin representing a level interval of the form $[iw, (i+1)w)$. Hence there are q/w bins. Each ball represents a cell, whose level is chosen by throwing the ball into one of the q/w bins. Hence there n balls, and by the uniform distribution on the cell levels we have a uniform and i.i.d. selection of bins. We need to calculate the probability that there is a bin with exactly one ball. Let A_i count the number of balls-to-bins assignments with the following property: "for a given set of i bins, each of the i bins has exactly one ball, and the other $q/w - i$ bins have $n - i$ balls, in any division". The number of assignments with *any* i bins having one cell each is counted by the inclusion-exclusion principle [3]. We need to apply inclusion-exclusion because the $\left(\begin{array}{c} q/w \\ i \end{array}\right) \cdot A_i$ obtained by naively summing A_i for every i-subset of bins counts assignments multiple times. By inclusion-exclusion, the number of assignments with *any* i bins having one cell each is $\sum_{k=i}^{q/w}(-1)^{k-i}\left(\begin{array}{c} q/w \\ k \end{array}\right)A_k$. We get the desired probability by taking $i = 1$, substituting $A_k = \left(\begin{array}{c} n \\ k \end{array}\right)k!(q/w-k)^{n-k}$, and dividing by the total number of balls-to-bins assignments $(q/w)^n$. ∎

We now prove Theorem 2.

Proof: (of Theorem 2) When $W = 2^m$, the maximal interval $[iw, (i+1)w)$ with one cell has size $w = 2^{r'}$, where $r' \in 0, \ldots, m$. For each r' in this range, denote by $p_{r'}$ the probability that a uniform cell assignment results in r' as the maximum. The probability that this maximal r' is *at least* r is calculated in Lemma 5. Denote this probability by $\underline{p_r}$. We can now write

$$E(r') = \sum_{r'=1}^{m} r' p_{r'} = \sum_{r=1}^{m} \underline{p_r}\,, \quad (2)$$

where the second equality follows from the fact that $\underline{p_r} = \sum_{r'=r}^{m} p_{r'}$ by definition, and the two sums have the same summands in their expansions. To get the expectation stated in the theorem we need to substitute $\underline{p_r}$ from (1) in the right-hand side of (2). ∎

In Fig. 2 we use Theorem 2 to plot the expected number of measurements as a function of W, for $q = 16$, $n = 4$. We compare these results to the analysis of the no-uncertainty n-cell binary search from [1]. It can be observed that higher W gives more measurement savings, but also that the benefit of uncertainty diminishes as we further grow W. In particular, for $T = 1$ there is no added benefit with the proposed modified binary-search algorithm to increasing W beyond $q/2$. That is because the first measurement is always necessary when $n > 1$, and it already reduces the uncertainty to $q/2$.

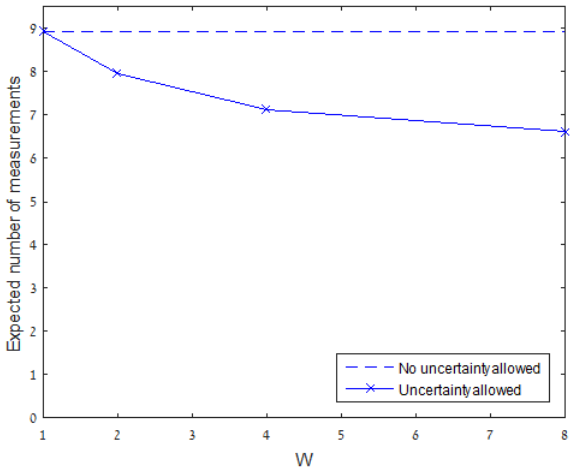


Fig. 2: The expected number of measurements needed to read $n = 4$ $q = 16$-ary cells, when uncertainty is allowed (solid line), and not allowed (dashed line).

B. W is not power of 2

In order to get a complete picture of the case of $T = 1$, let us now discuss the case where W is not power of 2 (we still assume that q is power of 2). Binary search would never utilize such a W entirely, and effectively we will reduce to the results of the previous sub-section with W rounded down to the closest power of 2. Hence, we suggest an enhancement of the modified binary search of Section II-A that can do better if W is not power of 2. In this enhanced algorithm we seek to improve the expected number of measurements, but while guaranteeing that in no instance it will need more measurements than what the algorithm of Section II-A would take.

Algorithm - W not power of 2. *The enhanced algorithm works as follows. Proceed with the previous modified binary search as long as all window sizes satisfy $w > 3W$. The first time we get a window $[L, U)$ whose size satisfies $2W < w \leq 3W$, and which includes $T = 1$ cell, replace the usual halving measurement by the threshold $L + W$.*

We now analyze the number of measurements used by the algorithm. With some probability, the cell level in the $[L, U)$ window is in the range $[L, L+W)$, and we can stop measuring this window after reaching an allowed uncertainty. This took only 1 measurement, while binary search would require 2 measurements in this window because $2W < w$. It may happen that the enhanced algorithm will also require 2 measurements: when the cell level is in the range $[L + W, L + w)$. But because of the upper limit of w, the enhanced algorithm will not require more than 2 measurements, because an additional threshold at $L + 2W$ will complete the read with an allowed uncertainty (since the remaining window's size is at most

W). Note that since the cell levels are distributed uniformly, measuring threshold $L+W$ is equivalent to threshold $U-W$ in the expected number of measurements. In the sequel we use r to denote the smallest natural number such that $2^r > 2W$, and define $W' \equiv 2^r$.

Example 6. $q = 32, n = 1, T = 1, W = 3$, and the cell's level is 0. Note that $W' = 8$ and $2W < W' \leq 3W$ holds. After two binary-search measurements, we reach a window $[0, 8)$. According to the algorithm we now measure at the threshold $L+W = 3$, and we get a window $[0, 3)$ that satisfies the allowed uncertainty in one measurement. In comparison, the binary search measures this window at the thresholds 4 and 2, one more measurement. Note that in the worst case, if the value of the cell was 7, after threshold 3 we would need another threshold 6, which completes the read in two measurements as the binary search.

In the following we derive the probability to save a measurement compared to binary search.

Theorem 7. *When W is not power of 2, let r denote the smallest natural number such that $2^r > 2W$, and define $W' \equiv 2^r$. If $W' \leq 3W$, then the enhanced algorithm saves one measurement over binary search with probability* $\left(\frac{W}{W'}\right) \cdot \sum_{k=1}^{q/W'} (-1)^{k-1} \binom{n}{k} \binom{q/W'}{k} \cdot \frac{k!}{(q/W')^k} \left(\frac{q/W'-k}{q/W'}\right)^{n-k}$.

Proof: The enhanced algorithm saves a measurement if there is a binary-search window of W' levels with one cell in it, and if the cell level is in the first W levels of this window. We denote the probability of the former by $p_{W'}$, and the probability of the latter equals $\frac{W}{W'}$. To calculate $p_{W'}$ we use the balls-and-bins model, but first group each W' levels as one bin, such that in total the q levels give q/W' bins. Standard balls-and-bins analysis gives $p_{W'} = \sum_{k=1}^{q/W'} (-1)^{k-1} \binom{n}{k} \binom{q/W'}{k} \frac{k!}{(q/W')^k} \left(\frac{q/W'-k}{q/W'}\right)^{n-k}$. Combining with the probability to be in the "correct" side of the window we get

$$\Pr(save\ one\ more\ measurement) = p_{W'} \cdot \left(\frac{W}{W'}\right) = \left(\frac{W}{W'}\right) \cdot \sum_{k=1}^{q/W'} (-1)^{k-1} \binom{n}{k} \binom{q/W'}{k} \cdot \frac{k!}{(q/W')^k} \left(\frac{q/W'-k}{q/W'}\right)^{n-k}.$$

In Fig. 3 we plot a numeric example of using Theorem 7 to obtain the probability to save one measurement over binary search, for $q = 16$ and $W = 3$. ■

III. Results

We now present the read-performance results of the read algorithms proposed in the paper. In Section III-A, we plot the average number of measurements required by different algorithms run with different read parameters n and W (all

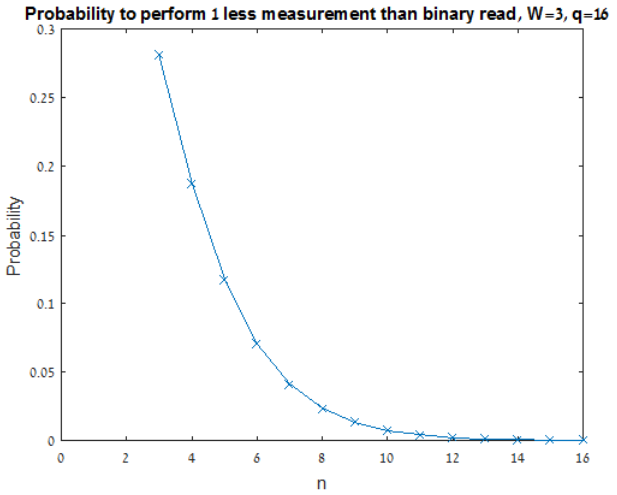


Fig. 3: A numeric example of the probability to save one measurement over the modified binary search.

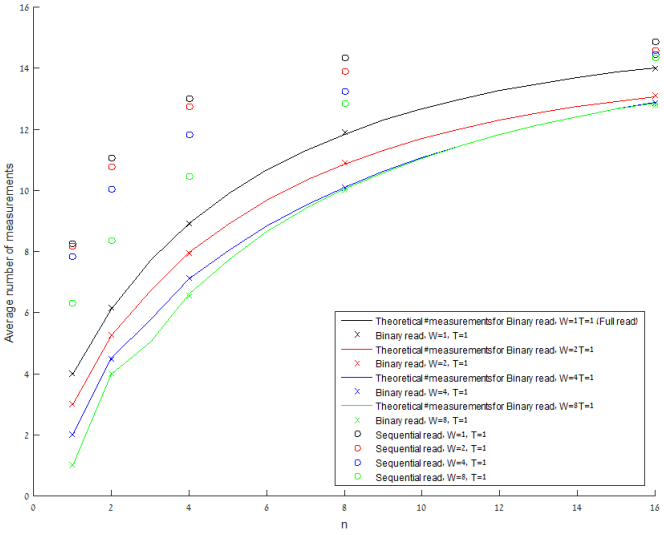


Fig. 4: Simulation results vs. analysis. Average number of measurements for $q = 16$.

with $T = 1$). In Section III-B, we compare the read performance of algorithms with uncertainty to read algorithms that read the information at full. For this comparison, we encode the information prior to writing it in memory, such that the redundancy is sufficient to decode the full information in the presence of the uncertainty W. This comparison takes into account the redundancy in evaluating the read performance.

A. Comparing different read algorithms and parameters

We now show a few results of implementation in Matlab environment of Theorem 2. Results are shown in Fig. 4

The plots show that the analytical expressions of Section II-A match the simulation results. In addition, the results show a clear advantage for the modified binary search with

uncertainty over other algorithms with and without uncertainty. *Sequential* refers to a read algorithm that starts at the lowest threshold level and proceeds in sequence upward until reaching the allowed uncertainty.

B. Comparing the average time until full-readout

When the algorithm returns a read with uncertainty, we cannot fulfill the read request to the memory. It is, however, possible to encode the information such that a decoder can remove the uncertainty and return the full information to the read request. For the case of $T = 1$ and W a power of 2 (Section II-A), we use the following simple code. If one cell is allowed an uncertainty of W, then out of the $log_2(q)$ bits of the symbol, at most the $log_2(W)$ lower significance bits will be missing. As a result we can apply the binary single-parity code on the $log_2(W)$ low bits, which costs a total redundancy of $log_2(W)$ for the entire block of n cells.

In that case, we need to take into account the redundancy when we evaluate the read performance. On the one hand, we saw in Section II-A that allowing uncertainty can speed up the reads. On the other hand, each full read of n cells returns less information when redundancy is incorporated.

For a fair comparison, we define the *normalized read time* as the number of read measurements divided by the information rate R that is stored in the cell block. When we substitute the information rate for the code we propose for $T = 1$,W uncertainty, we obtain $R = \frac{nlog_2(q)-log_2(W)}{nlog_2(q)} = 1 - \frac{log_q(W)}{n}$. In Fig. 5 we compare for different n values the normalized read times of algorithms with uncertainty, to those that read at full without redundancy. The dashed line shows the normalized read time without uncertainty, and the solid lines show the normalized time for three different values of W (each requiring a different amount of redundancy). It is shown that the read algorithms with uncertainty used with coding indeed give perfectly reliable readout with better normalized read performance. The plots reveal an interesting behavior: depending on the value of n, different uncertainty parameters W become attractive for the normalized read time. This is because increasing W has an adverse affect on the information rate alongside its good effect on the read time. Specifically for the parameters of Fig. 5, the choice of $W = 4$ seems to offer the best tradeoff between read performance and information rate.

IV. Conclusion

In this work, we studied the read performance of algorithms that use uncertainty. We developed algorithms for the case of $T = 1$ uncertain symbols, and analyzed their performance. The results show that algorithms with uncertainty together with appropriate coding can improve the read performance of non-volatile memories. For future work there are many interesting directions:

- Algorithms and analysis for the case $T > 1$.
- Obtaining an analytical lower bound for the number of measurements.

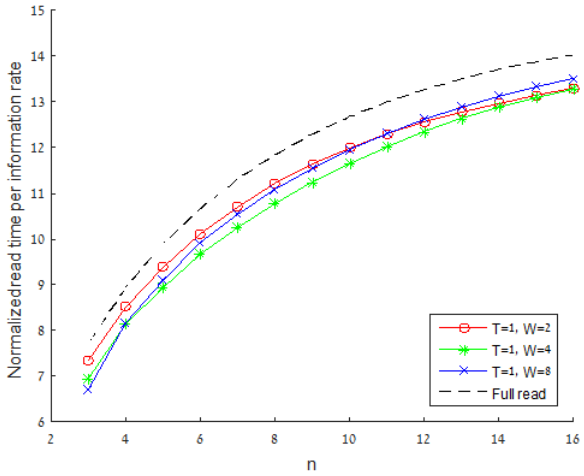


Fig. 5: Normalized time to read information as a function of n and W, for $q = 16$.

- Matching codes with read algorithms for different parameters.

V. Acknowledgement

The authors wish to thank Evyatar Hemo for valuable discussions. This work was supported in part by the Israel Science Foundation, by the Israel Ministry of Science and Technology, and by a GIF young investigator award.

References

[1] E. Hemo and Y. Cassuto, *Adaptive Threshold Read Algorithms in Multi-Level Non-Volatile Memories*, in *IEEE* Journal on selected areas in communications. Vol. 32, No. 5, May 2014.
[2] M. Mitzenmacher and E. Upfal, *Probability and Computing Randomized Algorithms and Probabilistic Analysis.*
[3] J.H. van Lint and R.M. Wilson, *A* course *in Combinatorics*.
[4] R.J. Baker, CMOS Circuit Design, Layout, and Simulation, Third Edition, Wiley-IEEE, 2010. ISBN 978-0-470-88132-3.