A Study of Polar Codes for MLC NAND Flash Memories

Yue Li[1,2], Hakim Alhussien[3], Erich F. Haratsch[3], and Anxiao (Andrew) Jiang[1]
[1]Texas A&M University, College Station, TX 77843, USA
[2]California Institute of Technology, Pasadena, CA 91125, USA
[3]LSI Corporation, San Jose, CA 95131, USA.
yli@caltech.edu, ajiang@cse.tamu.edu

***Abstract*—The increasing density of NAND flash memories makes data more prone to errors due to severe process variations and disturbance. The urgency to improve NAND flash reliability encourages searching for optimal channel coding methods. This paper reports our efforts towards a read channel for flash memories using polar coding. Our contributions include the solutions to several challenges raised when applying polar codes to NAND flash memories in practice. We propose efficient schemes for shortening both non-systematic and systematic polar codes, making polar codewords be easily adapted to flash page of any size. We demonstrate that the decoding performance of the shortened polar codes and LDPC codes are comparable using the data obtained by our NAND flash characterization platform. We show the feasibility of a practical adaptive decoding framework where it is not necessary to construct new polar codes for different channel parameters. Experimental results show that the decoding performance approaches the optimized performance where different codes are constructed for different channel conditions. To the best of our knowledge, this work is the first study of polar codes for error correction in flash memories.**

I. Introduction

NAND flash geometries have scaled beyond 20nm to achieve higher storage density. As a result, data are more prone to errors due to severe process variation and disturbance. The urgency to improve NAND flash reliability calls for continuous search for optimal channel coding schemes.

Polar codes proposed recently is the first class of capacity-achieving codes with efficient constructions [1]. However, the practical performance of polar codes in flash memories is still unknown due to many important challenges raised in practice. For instance, the length of a polar codeword needs to be an integer power of two, and such lengths do not directly fit in flash pages of different sizes; to conduct experimental analysis, the decoding performance of polar codes need to be compared with that of other error correcting codes (ECCs) on the same data from flash characterization platforms, and such testing data are not assumed to be the codewords of any ECC; moreover, the construction of polar codes uses channel statistics, and one concern is that new polar codes need to be frequently constructed for optimized performance as channel gradually degrades, which is prohibitively expensive in practice. This paper studies the solutions to the challenges above, and reports our efforts towards realizing polar decoders for flash memories using multi-level cells (MLCs).

One contribution of this paper is the design and experimental evaluations of shortened polar codes in NAND flash. Punctured polar codes have been studied recently [2] [3]. Puncturing has low implementation complexity, however it introduces additional erasures to received codewords and thus degrades decoding performance. This paper explores an alternative approach through shortening. We propose the schemes for shortening both non-systematic [1] and systematic polar codes [4]. Shortening obtains a shorter codeword by assigning selected codeword symbols to predetermined values made known both to encoder and decoder. The selected symbols are removed before transmission and are inserted back before decoding. Therefore, shortening does not introduce additional errors.

Another contribution of this paper is an adaptive polar decoder for NAND flash memories that gradually degrade. The decoder adaptively switches to use lower code rates as memory cells endure. Rate-compatible polar codes can be realized by adjusting the size of frozen sets without constructions of new codes [2]. This paper further shows that the property guarantees the feasibility of the practical adaptive polar decoding framework. We show that repeated polar code construction is not necessary when NAND flash channel degrades, and extensive experiments demonstrate that the decoding performance by constructing polar codes only once closely approaches the optimized performance.

To the best of our knowledge, this paper is the first work that studies the practical error correction performance of polar codes for flash memories. While the results indicate that polar codes are very promising for NAND flash, there still remain many open problems for further explorations.

II. Basic Models of MLC NAND Flash

An MLC NAND flash chip contains several planes. A plane consists of a set of blocks. A block has many pages, and each page consists many memory cells. Cells are the basic storage units of flash memories. The threshold voltage of a cell is quantized into multiple discrete levels to represent data. For MLC NAND flash, a cell has four levels and stores two bits. Cell levels are labeled using two-bit Gray codes. We refer the left bit as the most significant bit (MSB) and the right bit as the least significant bit (LSB). The statistical distributions of cell threshold voltages in a block can be appropriately approximated using Gaussian distributions. An example of cell level distributions is shown in Figure 1.

Data stored in cells are read using either hard or soft sensing. In both approaches, MSB and LSB are read independently. Hard sensing returns a possibly noisy version of the data, and soft sensing outputs the log-likelihood ratio (LLR) for each bit. Specifically, hard sensing applies one reference threshold voltage between two adjacent distributions. As an example, in Figure 1 let the reference threshold voltages used by hard sensing be $V_{\text{th},1,2}$, $V_{\text{th},2,2}$, and $V_{\text{th},3,2}$. To read LSB, the cell threshold voltage V_{th} is compared with the reference threshold voltage $V_{\text{th},2,2}$, returning 0 if $V_{\text{th}} > V_{\text{th},2,2}$, and 1 otherwise. Similarly, for MSB if $V_{\text{th},1,2} < V_{\text{th}} < V_{\text{th},3,2}$, bit 0 is returned, otherwise bit 1 is returned. Soft sensing uses k reference threshold voltages between two adjacent distributions. Figure 1 shows an example for $k = 3$. In general, for $i \in \{1, 2, 3\}$, $j \in \{1, 2, \cdots, k\}$, let $V_{\text{th},i,j}$ be the j-th smallest reference threshold voltage of the k reference threshold voltages between level i and $i+1$. The domain of cell threshold voltage is then divided into $3k+1$ bins by the reference voltages. During reading, the bin that V_{th} falls into can be determined by comparing V_{th} with different reference threshold voltages, according to which the LLRs are computed. Assume the threshold voltage V_{th} of a cell is found in the bin of the voltage interval $[V_{\text{th},i}, V_{\text{th},i+1})$, and let the mean and the standard deviation of the cell level distribution of level i be μ_i and σ_i. The LLRs of the LSB and the MSB are given by

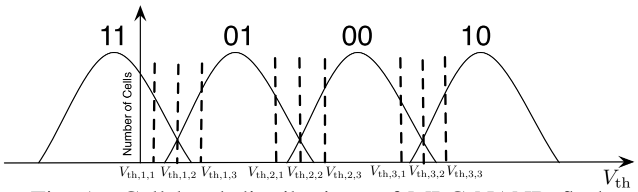


Fig. 1: Cell level distributions of MLC NAND flash.

$$
\begin{aligned}
L_{\text{lsb}} &\triangleq \ln \frac{\mathrm{P}(V \in [V_{\text{th},i}, V_{\text{th},i+1}) \mid \text{LSB} = 1)}{\mathrm{P}(V \in [V_{\text{th},i}, V_{\text{th},i+1}) \mid \text{LSB} = 0)} \\
&= \ln \frac{\sum_{l \in \{3,4\}} \mathrm{P}(V \in [V_{\text{th},i}, V_{\text{th},i+1}) \mid l)}{\sum_{l \in \{1,2\}} \mathrm{P}(V \in [V_{\text{th},i}, V_{\text{th},i+1}) \mid l)} \\
&= \ln \frac{Q(\frac{V_{\text{th},i}-\mu_3}{\sigma_3}) - Q(\frac{V_{\text{th},i+1}-\mu_3}{\sigma_3}) + Q(\frac{V_{\text{th},i}-\mu_4}{\sigma_4}) - Q(\frac{V_{\text{th},i+1}-\mu_4}{\sigma_4})}{Q(\frac{V_{\text{th},i}-\mu_1}{\sigma_1}) - Q(\frac{V_{\text{th},i+1}-\mu_1}{\sigma_1}) + Q(\frac{V_{\text{th},i}-\mu_2}{\sigma_2}) - Q(\frac{V_{\text{th},i+1}-\mu_2}{\sigma_2})},
\end{aligned}
$$

and

$$
\begin{aligned}
L_{\text{msb}} &\triangleq \ln \frac{\mathrm{P}(V \in [V_{\text{th},i}, V_{\text{th},i+1}) \mid \text{MSB} = 1)}{\mathrm{P}(V \in [V_{\text{th},i}, V_{\text{th},i+1}) \mid \text{MSB} = 0)} \\
&= \ln \frac{\sum_{l \in \{1,4\}} \mathrm{P}(V \in [V_{\text{th},i}, V_{\text{th},i+1}) \mid l)}{\sum_{l \in \{2,3\}} \mathrm{P}(V \in [V_{\text{th},i}, V_{\text{th},i+1}) \mid l)} \\
&= \ln \frac{Q(\frac{V_{\text{th},i}-\mu_1}{\sigma_1}) - Q(\frac{V_{\text{th},i+1}-\mu_1}{\sigma_1}) + Q(\frac{V_{\text{th},i}-\mu_4}{\sigma_4}) - Q(\frac{V_{\text{th},i+1}-\mu_4}{\sigma_4})}{Q(\frac{V_{\text{th},i}-\mu_2}{\sigma_2}) - Q(\frac{V_{\text{th},i+1}-\mu_2}{\sigma_2}) + Q(\frac{V_{\text{th},i}-\mu_3}{\sigma_3}) - Q(\frac{V_{\text{th},i+1}-\mu_3}{\sigma_3})},
\end{aligned}
$$

where $Q(\cdot)$ is the Q-function of the standard normal distribution. The sign of LLR determines the value of the bit that is more likely to be, and the absolute value of LLR measures the level of confidence. In this paper, the performance of polar coding using both sensing methods are studied.

The noise that happens to the MSB and the LSB of a cell is independently modeled using cascaded channels. A cascaded channel consists of more than one subchannels where two adjacent subchannels are connected such that the output of the first subchannel are the input of the second subchannel. Cascaded channels were used for modeling the errors for magnetic recording devices [5]. In this paper, a cascaded channel consists of two binary symmetric channels (BSCs) with cross-over probabilities p_{r} and p_{w} . We refer the first BSC as write channel and the second BSC as read channel. The write channel models the errors that occur in programming. Such errors include misprogram errors happened during the two-step MLC programming, cell-to-cell interference, and stuck cells. The read channel is implied by the cell level distributions of MLCs. When the threshold voltage of a cell is at the region where two distributions overlaps, the cell will be misread with high probability. In this work, the polar codes used for evaluations are constructed using the equivalent BSCs of the cascaded channels above.

III. Background on Polar Coding

A polar code is a linear block error correcting code which is provably capacity-achieving [1]. The encoder of a polar code transforms N input bits $\mathbf{u} = (u_1, u_2, \cdots, u_N)$ to N codeword bits $\mathbf{x} = (x_1, x_2, \cdots, x_N)$ through the linear transformation $\mathbf{x} = \mathbf{uG} = \mathbf{u}G_2^{\otimes m}$ where $G_2 = \left(\begin{smallmatrix} 1 & 0 \\ 1 & 1 \end{smallmatrix}\right)$ and $G_2^{\otimes m}$ is the m-th Kronecker product of G_2. The N codeword bits $(x_1, x_2, \cdots, x_N)$ are transmitted through N independent copies of a binary discrete memoryless channel (BDMC). For decoding, N transformed binary input channels $\{W_N^{(1)}, W_N^{(2)}, \cdots, W_N^{(N)}\}$ can be synthesized for $u_1, u_2, \cdots, u_N$, respectively. The channels are polarized such that for large N, the fraction of indices i for which the mutual information $I(W_N^{(i)})$ is nearly 1 approaches the capacity of the B-DMC, while the values of $I(W_N^{(i)})$ for the remaining indices i are nearly 0. The latter set of indices are called the frozen set. For error correction, the u_i's with i in the frozen set take fixed values, and the other u_i's are used as information bits. A successive cancellation (SC) decoding algorithm achieves diminishing block error probability as N increases.

In this work, we use the concept of upgrading and degrading channels, defined based on frozen sets. As in [6], a channel $W' : X \to Z$ is called "degraded with respect to a channel $W : X \to Y$" if a channel equivalent to W' can be constructed by concatenating W with an additional channel $Q : Y \to Z$, where the inputs of Q are linked with the outputs of W. That is, $W'(z|x) = \sum_{y \in Y} W(y|x) Q(z|y)$. We denote it by $W' \preceq W$. Equivalently, the channel W is called "an upgrade with respect to W'", denoted by $W \succeq W'$.

IV. Shortened Polar Codes in Flash Memories

Polar codes require the code length be 2^m where m is an integer. Without length-adaptation a codeword does not directly fit in flash memories of typical page sizes. For instance, current flash memories typically use BCH codes, where the user data length is 512 bytes or 1 Kbytes, and after applying the ECC parity bits the codeword size is not a power of 2. As a result, also the total flash page size is not a power of 2 either. We study the approaches to shortened polar codes defined below.

Definition 1. *An (N, K, K')-shortened polar code (SPC) is a polar code of length $N - K'$ obtained from an (N, K)-polar code with block length $N = 2^m$ and information bit length K by assigning K' predetermined input symbols to known values before encoding, and removing K' predetermined codeword symbols after encoding.*

Let us define the notations used later in this section. Consider an (N, K) binary polar code with $N = 2^m$. Let the non-frozen set of the code be $\mathcal{A} \triangleq \{a_1, a_2, \cdots, a_K\} \subseteq \{1, 2, \cdots, N\}$, and let the frozen set $\bar{\mathcal{A}} \triangleq \{b_1, b_2, \cdots, b_{N-K}\}$ be the complement. We also assume that $a_1 < a_2 < \cdots < a_K$ and $b_1 < b_2 < \cdots < b_{N-K}$. Denote the input bits to the encoder by $\boldsymbol{u} \triangleq (u_1, u_2, \cdots, u_N) = (\boldsymbol{u}_{\mathcal{A}}, \boldsymbol{u}_{\bar{\mathcal{A}}})$ to represent $\boldsymbol{u}$, where $\boldsymbol{u}_{\mathcal{A}} \triangleq (u_i : i \in \mathcal{A})$ contains the message bits and $\boldsymbol{u}_{\bar{\mathcal{A}}} \triangleq (u_i : i \in \bar{\mathcal{A}})$ contains the frozen bits. The codeword $\boldsymbol{x} \triangleq (x_1, x_2, \cdots, x_N)$ computed by encoding is written to cells. The reading process outputs a (possibly noisy) codeword $\boldsymbol{y} \triangleq (y_1, y_2, \cdots, y_N)$, and decoder computes the estimated codeword $\hat{\boldsymbol{x}} \triangleq (\hat{x}_1, \hat{x}_2, \cdots, \hat{x}_N)$.

A. Shortened non-systematic polar codes

We first study the shortening of non-systematic polar codes (NSPCs) whose encoding of an (N, K)-NSPC follows the linear transformation $\boldsymbol{x} \triangleq \boldsymbol{u}\mathbf{G}$.The shortening of NSPCs is based on the following lemma:

Lemma 2. *The input bits $(u_{N-K'+1}, u_{N-K'+2}, \cdots, u_N)$ are all 0s if and only if the bits $(x_{N-K'+1}, x_{N-K'+2}, \cdots, x_N)$ are all 0s.*

Proof: As the matrix $\mathbf{G}$ is a lower triangular matrix with ones on the diagonal, $\mathbf{G}$ is invertible and there is a one-to-one mapping between the $(u_{N-K'+1}, \cdots, u_N)$ and $(x_{N-K'+1}, \cdots, x_N)$, and when $(u_{N-K'+1}, \cdots, u_N)$ are all 0s, $(x_{N-K'+1}, \cdots, x_N)$ will be 0s. ■

The above lemma suggests we obtain an (N, K, K')-SPC from an (N, K)-NSPC by setting the last K' input bits to 0s, then removing the last K' codeword symbols after encoding. Among the K' input bits, there are K'' non-frozen bits and $K' - K''$ frozen bits where $K'' = |\{i | i \in \mathcal{A} \text{ and } N - K' + 1 \leqslant i \leqslant N\}|$.

Theorem 3. *An (N, K, K')-SPC obtained through the encoding above has rate $\frac{K-K''}{N-K'} \in [\frac{K-K'}{N-K'}, \frac{K}{N}]$.*

The encoding and the decoding algorithms are given below.

Encoding

1) For $j = a_{N-K-K'+K''+1}, a_{N-K-K'+K''+2}, \cdots, a_{N-K}$, let $u_j = 0$. For $j \in \bar{\mathcal{A}} - \{a_{N-K-K'+K''+1}, \cdots, a_{N-K}\}$, let u_j be any predetermined frozen bit (e.g. 0), completing $\boldsymbol{u}_{\bar{\mathcal{A}}}$. For $j \in \{b_{K-K''+1}, b_{K-K''+2}, \cdots, b_K\}$, let $u_j = 0$. Store $K - K''$ message bits in $(u_{b_1}, u_{b_2}, \cdots, u_{b_{K-K''}})$, completing $\boldsymbol{u}_{\mathcal{A}}$.
2) Compute $\boldsymbol{x} = \boldsymbol{u}\mathbf{G}$, and send the shortened codeword $(x_1, x_2, \cdots, x_{N-K'})$.

Decoding

1) After receiving the (possibly noisy) shortened codeword $(y_1, y_2, \cdots, y_{N-K'})$, let the codeword $\boldsymbol{y} = (y_1, y_2, \cdots, y_{N-K'}, 0, \cdots, 0)$ with K' 0s in the end.
2) Correct $\boldsymbol{y}$ using a polar decoder. The decoder treats the added bits $(y_{N-K'+1}, y_{N-K'+2}, \cdots, y_N)$ as if they went through a perfect channel and have unit probability of being 0.

B. Shortened systematic polar codes

In practice, systematic codes are used in flash memories to reduce the overhead for reading information bits. Systematic polar codes (SYPCs) was proposed by Arıkan [4]. Let $\mathbf{G}_{\mathcal{A}\mathcal{A}}$ be a submatrix of $\mathbf{G}$ such that for each element $G_{i,j}$, the indices $i, j \in \mathcal{A}$, the encoder computes the codeword $\boldsymbol{x} = (\boldsymbol{x}_{\mathcal{A}}, \boldsymbol{x}_{\bar{\mathcal{A}}})$ where the information part $\boldsymbol{x}_{\mathcal{A}} = \boldsymbol{u}_{\mathcal{A}}\mathbf{G}_{\mathcal{A}\mathcal{A}} + \boldsymbol{u}_{\bar{\mathcal{A}}}\mathbf{G}_{\bar{\mathcal{A}}\mathcal{A}}$ and the non-information part $\boldsymbol{x}_{\bar{\mathcal{A}}} = \boldsymbol{u}_{\mathcal{A}}\mathbf{G}_{\mathcal{A}\bar{\mathcal{A}}} + \boldsymbol{u}_{\bar{\mathcal{A}}}\mathbf{G}_{\bar{\mathcal{A}}\bar{\mathcal{A}}}$. To shorten SYPCs, we need the following theorem:

Theorem 4. *Let $\boldsymbol{u}_{\bar{\mathcal{A}}}$ be 0s. There is a one-to-one correspondence between $(u_{a_{K-K'+1}}, u_{a_{K-K'+2}}, \cdots, u_{a_K})$ and $(x_{a_{K-K'+1}}, x_{a_{K-K'+2}}, \cdots, x_{a_K})$.*

Proof: The matrix $\mathbf{G}_{\mathcal{A}\mathcal{A}}$ is a $K \times K$ lower-triangular matrix with ones on the diagonal. Let $\mathbf{G}_{\mathcal{C}\mathcal{C}}$ be a submatrix of $\mathbf{G}_{\mathcal{A}\mathcal{A}}$ where $\mathcal{C} = \{a_{K-K'+1}, \cdots, a_K\}$. We have

$$(x_{a_{K-K'+1}}, \cdots, x_{a_K}) = (u_{a_{K-K'+1}}, \cdots, u_{a_K}) \cdot \mathbf{G}_{\mathcal{C}\mathcal{C}}.$$

Since $\mathbf{G}_{\mathcal{C}\mathcal{C}}$ is a $K' \times K'$ lower-triangular matrix with ones on the diagonal, it is also invertible. ■

The theorem states that it is feasible to obtain an (N, K, K')-SPC from an (N, K)-SYPC by letting frozen bits be 0s, and setting the last K' bits of $\boldsymbol{u}_{\mathcal{A}}$ to predetermined values before encoding. The last K' bits of $\boldsymbol{x}_{\mathcal{A}}$ are removed after encoding.

Theorem 5. *An (N, K, K')-SYPC obtained through the encoding above has rate $\frac{K-K'}{N-K'}$.*

An instance of the encoding and the decoding algorithms for shortened SYPCs is given below, where we assign $(u_{a_{K-K'+1}}, \cdots, u_{a_K})$ to all 0s.

Encoding

1) Let $\boldsymbol{u}_{\bar{\mathcal{A}}}$ be 0s. Store $K - K'$ message bits in $(u_{a_1}, \cdots, u_{a_{K-K'}})$, and let $(u_{a_{K-K'+1}}, \cdots, u_{a_K})$ be 0s.
2) Do systematic encoding to compute $\boldsymbol{x} = (\boldsymbol{x}_{\mathcal{A}}, \boldsymbol{x}_{\bar{\mathcal{A}}})$. Send the shortened codeword $((x_{a_1}, x_{a_2}, \cdots, x_{a_{K-K'}}), \boldsymbol{x}_{\bar{\mathcal{A}}})$.

Decoding

1) After receiving a (possibly noisy) shortened polar codeword $((y_{a_1}, y_{a_2}, \cdots, y_{a_{K-K'}}), \boldsymbol{y}_{\bar{\mathcal{A}}})$, compute $\boldsymbol{y}_{\mathcal{A}} = (y_{a_1}, y_{a_2}, \cdots, y_{a_{K-K'}}, 0, \cdots, 0)$ with K' 0s appended at the end. We obtain the unshortened codeword $\boldsymbol{y} = (\boldsymbol{y}_{\mathcal{A}}, \boldsymbol{y}_{\bar{\mathcal{A}}})$.
2) Correct $\boldsymbol{y}$ with a polar decoder with frozen bits $\boldsymbol{u}_{\bar{\mathcal{A}}}$ (all 0s), treating the bits $(y_{a_{K-K'+1}}, \cdots, y_{a_K})$ as if they went through a perfect channel and have unit probability of being 0.

C. Polar codes with bit-reversal permutation

For the polar codes proposed in [1], codeword symbols are permuted by multiplying the generator matrix $\mathbf{G}$ with the bit-reversal permutation matrix $\mathbf{B}_N$. To adapt the shortening methods for the permuted codes simply requires modifying the locations of the symbols that are removed after encoding (and are inserted back before decoding): For permuted NSPCs, the K' indices of the bits that are removed are the images of the indices $(N-K'+1, N-K'+2, \cdots, N)$ under bit-reversal permutations; for permuted SYPCs, the K' indices are the images of the indices $\{a_{K-K'+1}, a_{K-K'+2}, \cdots, a_K\}$ under bit-reversal permutations.

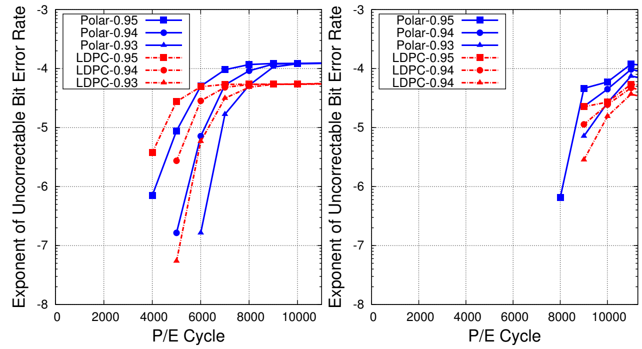
D. Performance evaluation

We evaluated the performance of shortened polar codes with the data from the characterizations of MLC flash chips using 2Y-nm technology. The characterization process sequentially programs each page in a block with random input bits, reads the stored (and possibly noisy) data, and erases the block for the next write. Such an iteration is referred as a program/erase cycle (PEC). Raw bit error rates increase as PEC grows, and the endurance of a cell is measured by the maximum PECs when data fail to decode. Starting with a new chip, we continue program-erase cycling the chip, recording the raw input and output data at multiple PECs during the lifetime of the block. As data written to the block are pseudo-random bits, coset coding technique is needed to view such random sequences as the codewords of the ECC being evaluated. Coset coding is feasible for polar codes, and the following results state that a random string of bits can be considered as a polar codeword of some message bits:

Lemma 6. *Given an (N,K)-polar code with frozen set $\bar{\mathcal{A}}$, $\forall \boldsymbol{x} \in \{0,1\}^N$, there is a unique $\boldsymbol{u}_{\mathcal{A}} \in \{0,1\}^K$ and a unique $\boldsymbol{u}_{\bar{\mathcal{A}}} \in \{0,1\}^{N-K}$ such that $\boldsymbol{x} = (\boldsymbol{u}_{\mathcal{A}}, \boldsymbol{u}_{\bar{\mathcal{A}}}) \cdot \mathbf{G}$.*

Corollary 7. *Given an (N,K,K')-SPC obtained from an (N,K)-NSPC with frozen set $\bar{\mathcal{A}}$, let $K'' = |\{i | i \in \mathcal{A}$ and $N-K'+1 \leqslant i \leqslant N\}|$, $\forall \boldsymbol{x}' \in \{0,1\}^{N-K'}$, there is a unique $\boldsymbol{u}'_{\mathcal{A}} \in \{0,1\}^{K-K'}$ and a unique $\boldsymbol{u}'_{\bar{\mathcal{A}}} \in \{0,1\}^{N-K-K'+K''}$, such that $(\boldsymbol{x}', \underbrace{0, \cdots, 0}_{K'}) = ((\boldsymbol{u}'_{\mathcal{A}}, \underbrace{0, \cdots, 0}_{K''})_{\mathcal{A}}, (\boldsymbol{u}'_{\bar{\mathcal{A}}}, \underbrace{0, \cdots, 0}_{K'-K''})_{\bar{\mathcal{A}}}) \cdot \mathbf{G}$.*

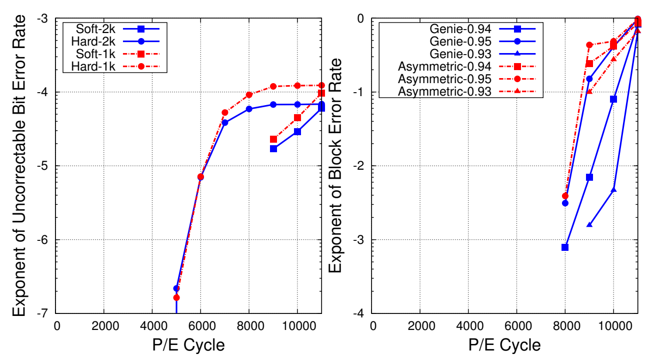
Figure 2 shows the average uncorrectable bit error rates (UBERs) of shortened polar codes at different PECs with both hard sensing (Figure 2(a)) and soft sensing (Figure 2(b)). We used the list decoding algorithm by Tal and Vardy [7] specified in probability domain with list size 32. For soft decoding, the input noisy codeword bits were determined by the signs of the LLRs, and the transition probability p of the BSC used for code construction is approximated from the LLR L by $p = e^{-|L|}/(1+e^{-|L|})$. We compared with the performance of the equivalent LDPC codes under min-sum decoding. Three rates (0.93, 0.94 and 0.95) of interest to flash memories are used. We assume each page stores 8 length-7943 polar codewords shortened from a length-2^{13} polar code constructed using the degrading merge algorithm [6] for the BSC with the



(a) Hard Decoding (b) Soft Decoding

Fig. 2: Comparison between polar codes and LDPC codes.

cross-over probability measured at the current PEC. The PECs when decoding failures first occurred are of special interest to flash memories. The results suggest the performance of both codes are comparable, and soft sensing significantly improves the endurance of MLCs. (Note that the endurance specified by the vendor for this chip is 3000 PECs.) Figure 3(a) compares the soft and the hard decoding performance between polar codes of lengths 2^{13} and 2^{14}. We found that the decoding of both codes failed at the same PECs, although the longer codes gave lower UBERs. Figure 3(b) compares the soft decoding performance of polar codes using a realistic soft sensing scheme with that of using a genie. The genie performed brute force search for the reference threshold voltages that maximize the degree of symmetry of the errors. The results show that lower BERs and higher decoding failure PEC were achieved by the genie by making errors more symmetric.



(a) (b)

Fig. 3: The performance of polar codes with (a) different block lengths as well as (b) realistic and genie soft sensing.

V. Adaptive Decoding

The channels of flash memories gradually degrade as PEC grows. Specifically, let the flash channel $\mathrm{W}(\alpha)$ be parameterized by PEC $\alpha \in \mathbb{N}$, $\mathrm{W}(\alpha) \succeq \mathrm{W}(\alpha')$ for any α, α' such that $\alpha \leqslant \alpha'$. To make the decoding error rates stay low, adaptive decoder is used in practice where lower code rates are used when the channel becomes more noisy.

Definition 8. *Let $R_1 > R_2 > \cdots > R_{k-1}$ be $k-1$ code rates of some channel code C, and let $\alpha_1 < \alpha_2 < \cdots < \alpha_k$ be k*

selected PECs. For $i \in \{1, 2, \cdots, k-1\}$, an adaptive decoder of C is the decoder which

1) *changes the rate of C to R_i at α_i.*
2) *uses rate R_i consistently for any $\alpha \in [\alpha_i, \alpha_{i+1})$.*

In this section, we show that polar codes is a good candidate for adaptive decoding in flash memories in the sense that the construction of new codes is not necessary through the lifetime of flash chips, and changing code rate only requires freezing additional input bits. Due to the relevance, we first state the following lemma, which restates Corollary 1 from [2].

Lemma 9. *(From [2]) Let F_W be the frozen set of the capacity-achieving polar codes for W. For any two channels W_i and W_j such that $W_i \succeq W_j$, the capacity achieving polar code for W_j can be obtained from the polar code for W_i by freezing additional input bits whose indices are in the set $F_{W_i} - F_{W_j}$.*

Consider an ideal adaptive polar decoder with unlimited code length, R_i being the capacity of $W(\alpha_i)$, and $\alpha_{i+1} = \alpha_i + 1$ for $i \in \{1, 2, \cdots, k-1\}$. The lemma above states that the ideal adaptive decoder can be realized by simply making additional input bits frozen when changing the rates at different PECs. In practice, adaptive decoders use finite block lengths, and it is prohibitively expensive to switch to a new code rate at each PEC. Therefore, we further consider a practical adaptive polar decoder with code rate R_i being smaller than the capacity of $W(\alpha_i)$, and $\alpha_{i+1} > \alpha_i + 1$ for $i \in \{1, 2, \cdots, k-1\}$. Let $W^{(1)}(\alpha), W^{(2)}(\alpha), \cdots, W^{(N)}(\alpha)$ be the N subchannels of the polar code for $W(\alpha)$. Let $\sigma_{W(\alpha)} = (x_1, x_2, \cdots, x_N)$ be the length-N permutation induced by the polarization order of the subchannels such that the sequence $P_e(W^{(x_1)}(\alpha)), P_e(W^{(x_2)}(\alpha)), \cdots, P_e(W^{(x_N)}(\alpha))$ is in ascending order where $P_e(\cdot)$ computes the maximum a posteriori (MAP) decoding error rate of a channel.

Theorem 10. *For any α, α' such that $\alpha \leqslant \alpha'$, and rate-R and rate-R' codes are used at α and α', respectively ($R > R'$), the polar code for $W(\alpha')$ can be obtained from the polar code for $W(\alpha')$ by further freezing the input bits in $F_{W(\alpha')} - F_{W(\alpha)}$ if*

$$\sigma_{W(\alpha)} = \sigma_{W(\alpha')}. \tag{1}$$

The condition (1) is motivated by our experimental observations. Figure 4 shows the theoretical decoding error probabilities of some randomly selected subchannels for the polar code constructed for the upper-odd pages. The figure suggests the error rates almost increase with PEC (due to process variation, the rates do not increase monotonically between 0 and 1000 PECs), and that the order of polarization be well preserved. Assume (1) holds for any $\alpha, \alpha' \in [\alpha_1, \alpha_k]$. The next corollary states that the constructions of new codes can be avoided for the practical adaptive decoders in Definition 8.

Corollary 11. *For $i \in \{1, 2, \cdots, k-1\}$, when the decoder changes the code rate R_i previously used at $\alpha_{i+1} - 1$ to R_{i+1} to be used at α_{i+1}, it only needs to further make the input bits in $F_{w_{\alpha_{i+1}}} - F_{w_{\alpha_{i+1}-1}}$ frozen, and given any two PECs $\alpha, \alpha' \in [\alpha_i, \alpha_{i+1})$, with the same code rate R_i the polar codes for $W(\alpha)$ and $W(\alpha')$ are equivalent.*

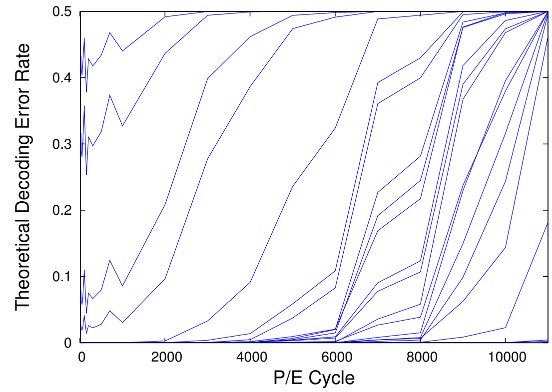
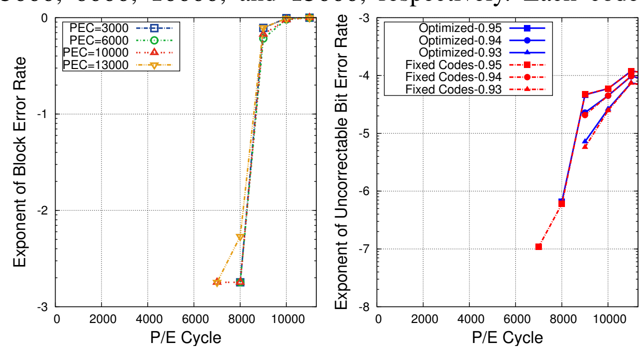


Fig. 4: The theoretical decoding error rates of some subchannels at different PECs. Each curve is for one subchannel.

Figure 5(a) shows the block error rates of four polar codes of rate-0.94 for the upper-odd pages constructed at PECs 3000, 6000, 10000, and 13000, respectively. Each code is



(a) Average BERs of upper-odd pages (b) Average UBERs over all pages

Fig. 5: The soft decoding performance of the codes constructed at fixed PECs.

tested through the whole lifetime of the flash chips. The results suggest the codes yield very similar decoding performance due to the polarization order preservation shown in Figure 4. Figure 5(b) compares the average UBERs of the codes constructed at 6000 PECs with the optimized performance yield by codes constructed at different PECs. The performance of the scheme without construction of new code closely approaches the optimized performance.

Acknowledgment

The authors would like to thank Sundararajan Sankaranarayanan for the many constructive discussions. This work was supported by the University Research Program of LSI Corporation.

References

[1] E. Arıkan, "Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels," *IEEE Trans. Information Theory.*, vol. 55, no. 7, pp. 3051–3073, July 2009.

[2] A. Eslami and H. Pishro-Nik, "A practical approach to polar codes," in *Proc. ISIT*, 2011, pp. 16–20.

[3] D.-M. Shin, S.-C. Lim, and K. Yang, "Design of length-compatible polar codes based on the reduction of polarizing matrices," *IEEE Transactions on Communications*, vol. 61, no. 7, pp. 2593–2599, 2013.

[4] E. Arıkan, "Systematic polar coding," *IEEE Communications Letters*, vol. 15, no. 8, pp. 860–862, 2011.

[5] A. Iyengar, P. Siegel, and J. Wolf, "Ldpc codes for the cascaded bsc-bawgn channel," in *47th Annual Allerton Conference on Communication, Control, and Computing*, 2009, pp. 620–627.

[6] I. Tal and A. Vardy, "How to construct polar codes," *IEEE Transactions on Information Theory*, vol. 59, no. 10, pp. 6562–6582, 2013.

[7] ——, "List decoding of polar codes," in *Proc. ISIT*, 2011, pp. 1–5.