electronics

MDPI

Article

Error Correction for TLC and QLC NAND Flash Memories Using Cell-Wise Encoding

Daniel Nicolas Bailon, Johann-Philipp Thiers and Jürgen Freudenberger *

Institute for System Dynamics (ISD), HTWG Konstanz, University of Applied Sciences, 78462 Konstanz, Germany; dnicolas@htwg-konstanz.de (D.N.B.); jthiers@htwg-konstanz.de (J.-P.T.)
* Correspondence: jfreuden@htwg-konstanz.de; Tel.: +49-7531-206-647

Abstract: The growing error rates of triple-level cell (TLC) and quadruple-level cell (QLC) NAND flash memories have led to the application of error correction coding with soft-input decoding techniques in flash-based storage systems. Typically, flash memory is organized in pages where the individual bits per cell are assigned to different pages and different codewords of the error-correcting code. This page-wise encoding minimizes the read latency with hard-input decoding. To increase the decoding capability, soft-input decoding is used eventually due to the aging of the cells. This soft-decoding requires multiple read operations. Hence, the soft-read operations reduce the achievable throughput, and increase the read latency and power consumption. In this work, we investigate a different encoding and decoding approach that improves the error correction performance without increasing the number of reference voltages. We consider TLC and QLC flashes where all bits are jointly encoded using a Gray labeling. This cell-wise encoding improves the achievable channel capacity compared with independent page-wise encoding. Errors with cell-wise read operations typically result in a single erroneous bit per cell. We present a coding approach based on generalized concatenated codes that utilizes this property.

Keywords: non-volatile memory; channel capacity; error correction coding; concatenated codes

check for updates

Citation: Nicolas Bailon, D.; Thiers, J.-P.; Freudenberger, J. Error Correction for TLC and QLC NAND Flash Memories Using Cell-Wise Encoding. *Electronics* **2022**, *11*, 1585. https://doi.org/10.3390/electronics11101585

Academic Editor: Marco Vacca

Received: 16 April 2022
Accepted: 12 May 2022
Published: 16 May 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Many novel applications, such as medical devices, IoT, and autonomous vehicles, require large storage capacities and fast data access [1,2]. NAND flash-based non-volatile storage is well suited for such applications. Moreover, the development of NAND flash memory with increased storage density is progressing steadily.

The NAND flash cells consist of floating gate transistors in which data are stored in the form of electric charge states. There are different technologies for NAND flash memory. The multi-level cell (MLC) and triple-level cell (TLC) flash memories can store two and three bits, respectively. Quadruple-level cells (QLC) can store four bits per cell [3], and penta-level cells (PLC), which store five bits per cell, are already in development [4].

With the growing storage capacity, error correction codes (ECC) are becoming increasingly important. There are numerous sources of disturbances for a flash cell, such as charge loss over time, cell-to-cell interference, program and erase (P/E) cycle stress, and external influences, such as temperature fluctuations, that reduce the memory's reliability and influence the threshold voltage distribution [5,6]. Therefore, the noise level and the error probability of the flash channels change during the device's lifetime.

With MLC flash memories, hard-decision decoding based on Bose–Chaudhuri–Hocquenghem (BCH) codes was sufficient for error correction [7,8]. Due to the high memory density and fabrication tolerances, errors during the readout are becoming more probable, and more sophisticated error correction algorithms are necessary. To improve the error correction performance, soft reading is applied to the flash cell [9–12]. Low-density parity-check codes (LDPC) are well-suited for flash systems with hard-input or soft-input decoding [10,13–17]. However, the error floor of LDPC codes may cause reliability issues

for industrial applications, which have to guarantee word error rates below 10^{-16} [18]. Such error rates cannot be analyzed by simulations. Generalized concatenated (GC) codes are suitable for flash-based storage systems that require low guaranteed residual error rates [18–22]. For GC codes, it is possible to bound the residual error probability.

During the flash readout, different reference voltages are applied to infer the charge state of the cell. The soft information for soft-input decoding is obtained by applying additional reference voltages. These additional read operations improve the reliability but increase the latency and the power consumption. Furthermore, soft-input decoding is more complex than hard-input decoding, which also leads to higher power consumption.

In this work, we consider a coding approach with the objective of avoiding the soft reading operation as long as possible. This approach is based on a cell-wise encoding instead of the typically used page-wise encoding. Flash memories are organized in pages where usually the different bits of a cell are assigned to different pages and different codewords of the error-correcting code. This page-wise encoding minimizes the read latency with hard-input decoding. Depending on the bit labeling, a different number of read operations is needed for the different pages, and accordingly, the read latency depends on the bit labeling. In addition, bit labeling affects the error probability of the pages [23].

In the case of a cell-wise read, code bits from all pages associated with a flash cell must be read together. Depending on the flash technology, this reading procedure may increase the read latency. On the other hand, the dependencies between the different bits stored in a cell can be exploited with cell-wise read operations. For instance, a method that improves the decoding performance of LDPC codes with cell-wise reading was presented in [24,25]. This method calculates log-likelihood ratios for the different bits depending on the estimated charge state. However, this method requires soft-input decoding, which is more complex and causes a higher power consumption than hard-input decoding. Moreover, channel estimation is required to determine the log-likelihood ratios. In contrast, we consider a joint encoding approach for GC codes that requires only algebraic hard-input decoding and avoids additional channel estimation.

The bit labeling or coding of the cell states was addressed in the literature with different aims. In [26], a coding scheme was introduced that reduces inter-cell interference for various flash types, including QLC and PLC. These constrained codes exclude problematic bit patterns from the set of possible code patterns. However, these constrained codes significantly reduce the code rate. The achievable code rates are low considering the spare space provided by today's flash memories. Another coding approach was pursued with write-once memory (WOM) codes [27,28]. This coding technique attacks the aging due to program and erase cycles by reducing the number of erasures. The bit labeling of available flash memories is usually based on Gray codes [7]. A Gray code is a binary code where the codewords of neighboring charge states differ only in one bit position. Such an encoding scheme minimizes latency for page-wise random access. However, the error probability and the total cell capacity is not completely balanced with Gray codes [23].

The multiple sources of errors in flash memory due to different physical effects are problematic for modeling the flash channel [6]. There are numerous publications on modeling flash memories [29–34]. Similarly to [9,23], we consider a capacity analysis for the flash cells. For the capacity calculation, we fit a model to measured voltage distributions. Based on these measured voltage distributions for a TLC flash, we demonstrate that the page-wise read operations lead to a loss in terms of the overall achievable channel capacity which is comparable to the capacity gain that is obtained by the soft read operation. In other words, the joint processing of all pages with hard-input decoding can achieve a channel capacity similar to a page-wise soft-input decoding approach.

To demonstrate that a cell-wise encoding provides practical advantages, we present a coding approach based on generalized concatenated codes. This coding approach improves the average error correction performance without increasing the number of reference voltages. We consider TLC flashes where all bits are jointly encoded using a Gray labeling. The proposed GC codes exploit the fact that errors with cell-wise reading typically result in

a single erroneous bit per cell. This results from the fact that almost all errors are caused by detecting a charge state adjacent to the programmed state. This is described in more detail in [35].

In the following, we review the bit mapping with Gray codes for flash-based memory. Then, we describe the channel model and consider the achievable channel capacities in Section 3. We present the GC code construction with inner BCH and outer RS code in Section 4. In Section 5, we investigate the error correction performance with page-wise and cell-wise encoding using GC codes. We present a performance analysis for TLC and QLC flashes with codes for 2 and 4 kB payloads of data and different code rates. While the analysis of the channel capacities is based on measured data, the analysis of the codes uses a unipolar M-ASK channel model.

2. Basics of NAND Flash Memory

This section describes some basics of NAND flash memory. Floating gate transistors are the main components of flash-based storage. NAND flash memory is organized into multiple blocks and pages [36]. The flash technology defines how many bits are stored per NAND flash cell and how many pages are managed per cell. There are three possible operations on a NAND flash cell: program, erase, and read. Reading and programming are done at the page level, and erasing is done at the block level. In addition to the storage capacity for user data, a flash memory provides a spare area which is used for error correction, meta data, or system pointers [36].

The threshold voltage of the floating-gate transistor depends on a value currently stored on the floating gate. During readout, different voltages are applied to the control gate to infer the charge state of the cell and to read out the stored information.

Figure 1 shows an example of the voltage distribution of a TLC flash memory with possible charge states S_i. A TLC flash cell stores three bits per cell; thus, it differentiates eight possible charge states. They are denoted $S_0, \ldots, S_7$. The x-axis represents the gate voltage and the y-axis the state-dependent probability densities. Such a threshold voltage distribution represents the probability that transistors are activated at a certain voltage over a large number of flash memory cells with the same charge level. The threshold voltage distributions are often assumed to be Gaussian distributions [37]. For simplicity, we have illustrated Gaussian distributions in Figure 1. Later, we will consider measured distributions.

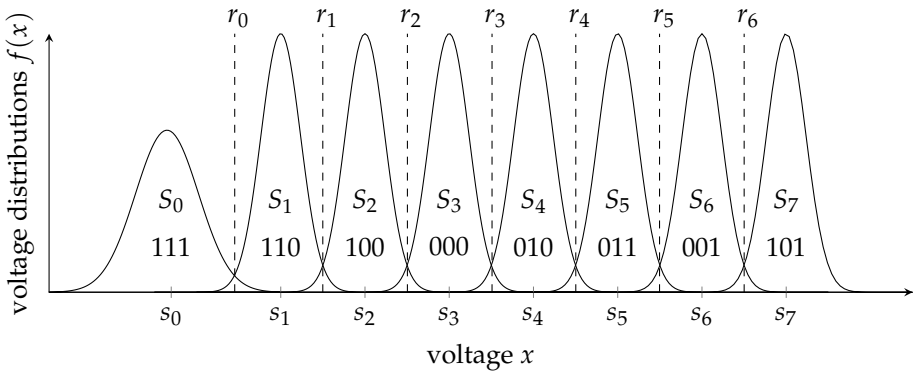

Figure 1. Example of the voltage distribution of a TLC flash memory, with read reference voltages (dashed lines) and bit-labeling with Gray code.

To read the charge state of a floating gate transistor, we distinguish between hard and soft reading. In hard reading, several reference voltages r_i are applied one after the other in order to infer the charge state S_i and the corresponding bit values. The reference voltage r_i is determined by the voltage distribution of the states S_i and S_{i+1}.

The bits of a charge state are associated with different pages, because mostly a page-wise readout of the data is desired to reduce the read latency. Usually, Gray codes are used as bit labeling of the data on the different charge states, e.g., as depicted in Figure 1. The main characteristic of Gray codes is that neighboring bit patterns only differ by one bit.

Each bit is mapped to a page; e.g., the bits of a triple-level flash cell are classified into three different pages. These are called the most significant bit (MSB) page, the center significant bit (CSB) page, and the least significant bit (LSB) page. In Figure 1, the first bit is the MSB and the last bit the LSB. Three examples of possible Gray codes for TLC Flash cells are listed in Table 1.

Table 1. Possible Gray code bit-labeling for a TLC flash cell.

Charge State	S_0	S_1	S_2	S_3	S_4	S_5	S_6	S_7
Gray code 1	111	110	100	000	010	011	001	101
Gray code 2	111	110	100	101	001	000	010	011
Gray code 3	111	101	100	110	010	011	001	000

The soft read provides reliability information about the charge state. Figure 2 shows examples of soft read thresholds between two voltage distributions producing 2 bits of soft information. The soft values are quantized according to the number of additional reference voltages. Here, the reference r_i gives the best distinction between the states S_i and S_{i+1}; and the references $r_i^{(-1)}$ and $r_i^{(1)}$ generate one bit of soft information, and $r_i^{(-2)}$ and $r_i^{(2)}$ a second bit of soft information. Depending on the bit mapping, a different number of reference voltages must be applied per cell to read out one bit of a page. With the additional readout of soft bits, the number of readouts of an entire cell increases enormously, as do the latency and power consumption. However, this increases the information capacity, and decoding failures and loss of saved information are made less probably in the flash memory.

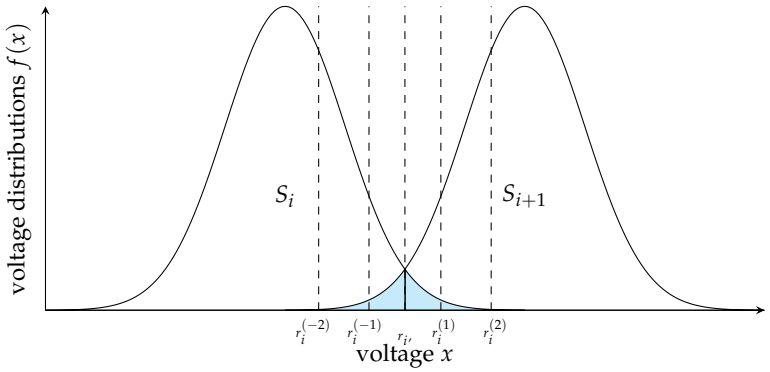

Figure 2. Voltage distributions with soft thresholds.

3. Analysis of the Channel Capacity

The capacity analysis provides an indication of how much information is obtained by reading at a particular reference voltage. Due to aging, the threshold distribution and the channel capacities of a flash memory change with time [38]. In [9,38], the capacities of MLC flash cells over their lifetime are considered. Similarly, reference [23] provides capacity considerations of TLC flash cells at the end of life (EOL) state. EOL is the last state defined by the manufacturer at which the flash memory should be readable. This case is reached when the maximum number of P/E cycles and the maximum data retention time defined by the manufacturer have been reached. The modeling of flash memories is considered in [29–34].

3.1. Measured Voltage Distributions

As in [23], we consider measured voltage distributions at the EOL state and apply a parameter estimation approach to estimate the probability distributions. We investigate the channel capacity of an industrial-grade TLC at EOL which is defined by 3000 P/E cycles and a data retention time of one year. The data retention time is simulated by a baking process.

For instance, Figure 3 shows the measured histogram of state S_7 of a TLC flash memory at EOL together with the fitted model and a Gaussian model. To obtain the required measurements, a flash memory page is read at 256 possible voltage steps for each reference r_i. Figure 3 represents the histogram in a logarithmic scale. The normalized threshold voltage represents the discrete measurement steps.

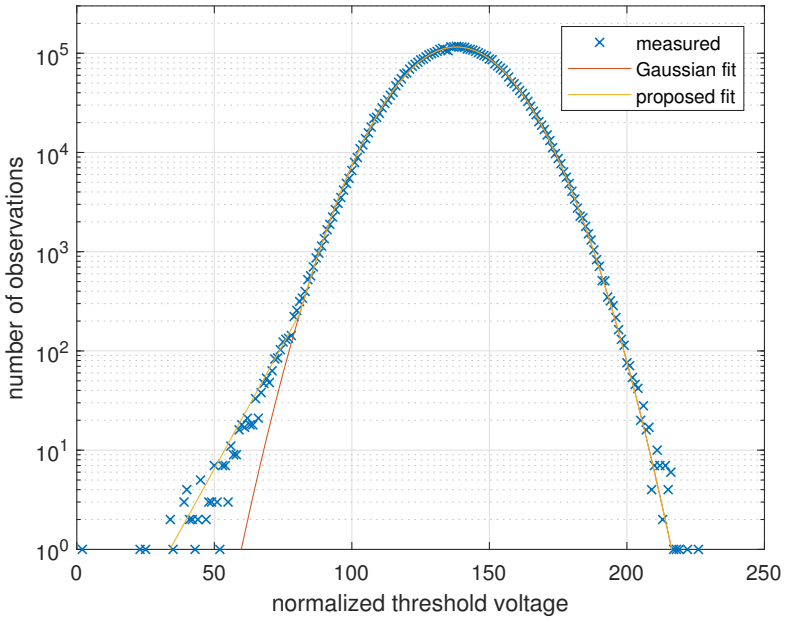

Figure 3. Fitting according to [23] for S_7 at EOL.

With the logarithmic scale, one can see that the parabolic shape of the Gaussian distribution is a suitable approximation for the voltage distribution around the mean value. However, the measure distribution has tails that do not follow the Gaussian distribution. The tail to the left impacts the bit error probability and cannot be neglected. Voltage distributions with exponential tails are also reported in [33,34]. To consider such tails, we use a piecewise function $f(x \mid S_i)$ for each state S_i that is defined in two parts, where the left part corresponds to an exponential distribution and the right part to a Gaussian distribution:

$$f(x \mid S_i) = \frac{1}{n_i} \cdot \begin{cases} c_i \cdot e^{\lambda_i (x - x_i)} & x < x_i \\ \frac{1}{\sqrt{2\pi\sigma_i^2}} e^{\frac{-(x-\mu_i)^2}{2\sigma_i^2}} & x \geq x_i \end{cases} \tag{1}$$

with

$$c_i = \frac{1}{\sqrt{2\pi\sigma_i^2}} e^{\frac{-(x_i-\mu_i)^2}{2\sigma_i^2}}. \tag{2}$$

The parameter n_i is chosen to guarantee the normalization, $\int_{-\infty}^{\infty} f(x|S_i)dx = 1$. This model allows estimating the probabilities for errors in non-adjacent states, e.g., from state S_7 to state S_5. Such errors cannot be reliably estimated from the measurement samples.

Figure 4 shows the fitted distributions $f(x \mid S_i)$ for the measured TLC flash at EOL, where the normalized threshold voltage x represents the discrete measurement steps. The dashed lines indicate the reference voltages.

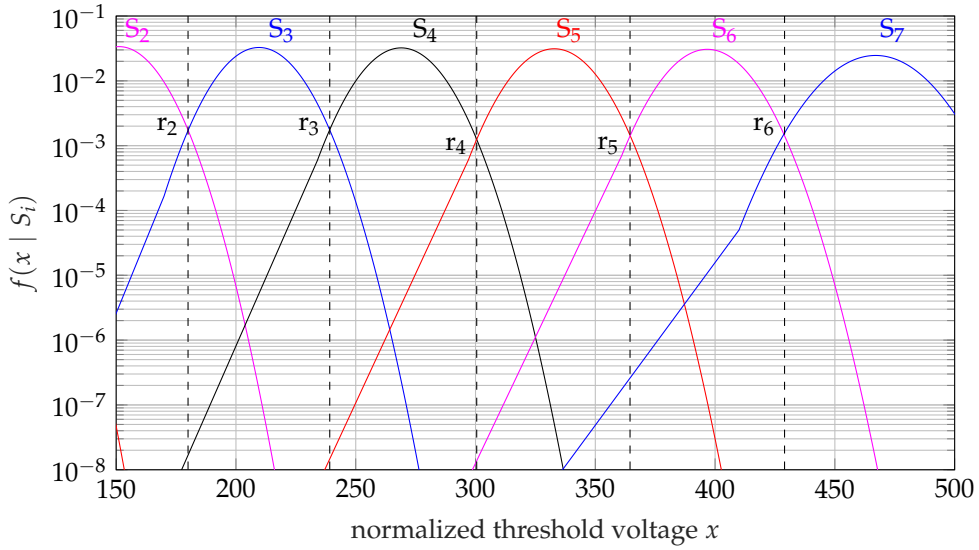

Figure 4. Fitted voltage distributions for a TLC flash at EOL.

3.2. Capacity Consideration

In the following, we consider the channel capacities of the measured TLC flash in the EOL scenario. The code rate of an error correction code is calculated as $R = k/n$, where k is the dimension and n is the code length. The channel capacity C is the supremum of all achievable rates. The channel capacity limits the possible code rate for reliable communication, resulting in $R < C$. The channel capacity is expressed by

$$C = \sup_{f_X(x)} I(X;Y) \tag{3}$$

where $I(X;Y)$ is the mutual information between the random variables X and Y. In our considerations, X is the channel input, i.e., the charge state of the cell. The variable Y denotes the channel output. The supremum is taken over all possible choices of the input distribution $f_X(x)$.

In the case of hard-input decoding and page-wise read operations, Y is a binary random variable, and we can calculate the level capacities C_l for each page. The channel capacity can be split up according to the chain rule of information theory [39]. The sum of the level capacities C_l gives the total capacity:

$$C = \sum_l C_l. \tag{4}$$

When applying a fixed set of reference voltages, the total capacity C does not depend on the selected bit-mapping, but the page capacities do. That is, each page can have a different capacity. It is possible to achieve the total capacity by a water filling or multilevel coding approach [39–41], where the code rates are adjusted to the page capacities; i.e., the code rate of each page l is $R_l = k_l/n_l < C_l$. However, multilevel coding requires shared processing of all pages.

In a practical flash memory, each page has the same number of cells and the same spare area. Hence, page-wise reading typically implies that the same code is used on all pages with $R = R_l < C_{min}$ because the code rate is determined by the smallest page's capacity. Such a page-wise coding limits the achievable total capacity. The capacity of such an independent coding procedure becomes

$$C_{min} = \min_l C_l. \tag{5}$$

Hence, the total achievable capacity of a TLC with page-wise read is $C_p = 3C_{min} < C$, and it is $C_p = 4C_{min} < C$ for QLC flash. This restricts the achievable error correcting performance.

The capacity of the bit-pages may differ significantly. The main cause for these differences is that different numbers of reference voltages are required to read a particular page. To provide some insight into this issue, we consider a simplified model that demonstrates that the bit error probabilities for the pages mainly depend on the number of reference voltages. Consider, for example, Gray code 2 from Table 1. The MSB is one for states S_0 to S_3 and zero for states S_4 to S_7. Hence, only the reference voltage r_3 has to be applied to read the MSB page. Assume that each state has the same probability $p(S_i) = 1/8$. The probability of making an error in the MSB page is therefore

$$p_{e,MSB} = \frac{1}{8}\sum_{i=0}^{3}\int_{r_3}^{\infty} f(x|S_i)dx + \frac{1}{8}\sum_{i=4}^{7}\int_{-\infty}^{r_3} f(x|S_i)dx, \tag{6}$$

where $f(x|S_i)$ is the probability density for the threshold voltage, given state S_i was programmed.

Note that the error probability for each state mainly depends on the distance between the reference r_3 and the mean value μ_i for state S_i. The probability of an error in a non-adjacent states is extremely low, as can be seen in Figure 4. Assuming that non-adjacent states cannot induce an error, we obtain the simplified estimate

$$p_{e,MSB} \approx \frac{1}{8}\int_{r_3}^{\infty} f(x|S_3)dx + \frac{1}{8}\int_{-\infty}^{r_3} f(x|S_4)dx. \tag{7}$$

Neglecting the asymmetry of the voltage distributions, we also assume

$$p_e \approx \int_{r_i}^{\infty} f(x|S_i)dx \approx \int_{-\infty}^{r_i} f(x|S_{i+1})dx, \ \forall i. \tag{8}$$

This leads to the bit error probability $p_{b,MSB} \approx \frac{1}{4}p_e$ for the MSB page. Under this model, the value p_e is the probability that an error occurs at r_i given that the programmed state is an adjacent state, i.e., either S_i or S_{i+1}. The factor $\frac{1}{4}$ is the probability that S_i or S_{i+1} is used. Consequently, the symbol error probability is $p_s \approx \frac{14}{8}p_e$, because the states S_0 and S_7 have only one neighboring reference, whereas all other states have two.

In this simplified model, the bit error probability for a page depends on p_e and the number of states that are adjacent to the required reference voltages. The number of states that are adjacent to a reference voltage is determined by the number of references $N_{R,page}$ for the page. Thus, we have the bit error probability

$$p_{b,page} \approx \frac{2N_{R,page}}{8}p_e. \tag{9}$$

For instance, consider the LSB page for Gray code 2. The relevant references for the LSB page are r_0, r_2, r_4, and r_6. Hence, all states are adjacent to a reference voltage, which leads to the error probability for the LSB page $p_{b,LSB}of \approx p_e$. Accordingly, Gray code 1 has the page error probabilities $p_{b,MSB} = p_{b,LSB} \approx \frac{1}{2}p_e$ and $p_{b,CSB} \approx \frac{3}{4}p_e$ and is therefore better balanced than the other Gray codes from Table 1.

While this model is oversimplified to calculate the exact bit error probabilities, it explains the differences in the page capacities in Table 2. Consider again Gray code 2: the LSB page uses four references, which leads to the highest bit error probability and the lowest page capacity. The CSB page uses two references. The highest capacity is obtained for the MSB page with a single reference voltage.

Table 2. Page capacities of a TLC at EOL depending on the bit-labeling.

	C_0 (MSB Page)	C_1 (CSB Page)	C_2 (LSB Page)	C	C_p	Read Thresholds
Gray code 1	0.967	0.958	0.982	2.907	2.874	7
Gray code 2	0.983	0.975	0.949	2.907	2.847	7
Gray code 3	0.983	0.964	0.959	2.906	2.877	7
Gray code 1 with 1 bit soft	0.981	0.975	0.988	2.944	2.926	21
Gray code 1 with 2 bit soft	0.982	0.978	0.991	2.951	2.934	35

Table 2 provides the corresponding channel capacities C_l for the Gray codes from Table 1 with hard reading and soft reading. The capacities were calculated using the model with the fitted distributions $f(x|S_i)$. The reference voltages were obtained by maximizing the mutual information $I(X;Y)$ for the measured voltage distributions. Note that the page capacities are different for different Gray codes. Similarly, the achievable rate C_p with page-wise encoding depends on the bit labeling. In all cases, the capacity with page-wise encoding is smaller than the total capacity C. The capacity C is achieved when all pages are considered jointly, i.e., with cell-wise reading. We can conclude that the bit labeling impacts the page capacities and consequently the achievable capacity with page-wise read operations. It also determines the latency for the random access performance for the different pages, because a larger number of references requires more read operations [9,10]. Note that it is not possible for a TLC to completely balance the number of reference voltages for all pages.

In addition to the hard-input capacities, the capacities with one and two soft bits for Gray code 1 are given in Table 2. The soft read increases the capacity compared to hard read operations, but requires many more read thresholds. In [23], a multilevel coding approach was proposed to reduce the number of read thresholds compared with soft reading. This coding approach is based on a cell-wise encoding of the bits and uses nine read thresholds. Hence, the latency is increased with respect to hard-input encoding with seven read thresholds.

In this work, we also consider joint processing of all pages, but without increasing the number of read references. This approach was motivated by the results in Table 2. For instance, consider Gray code 1. The loss in terms of the overall achievable channel capacity due to page-wise encoding is $2.907 - 2.874 = 0.033$ bits per cell. The gain of one bit of soft reading is $2.926 - 2.874 = 0.052$, which is only slightly higher than the gain for cell-wise encoding. On the other hand, soft reading increases the number of references significantly. Hence, a cell-wise encoding should also lead to a performance gain compared with page-wise hard-input decoding. Moreover, it should help to postpone the costly soft read operations.

4. Proposed Coding Scheme

In this section, we propose a construct for generalized concatenated codes for a cell-wise encoding of the bits. GC codes are suitable for providing very low residual error rates with low decoding complexity [42]. GC codes also enable efficient soft-input decoding [20,21].

Similarly to coded modulation [39,43], generalized concatenated codes are multilevel codes based on the partitioning of the inner code. The partitioning of a code results in subcodes with smaller cardinalities and higher minimum distances. The proposed construction is similar to the GC codes from [18–21]; i.e., we use outer RS codes and inner BCH codes. The inner BCH codes enable simple partitioning and efficient algebraic decoding [44]. However, other codes, such as polar codes, can also be used as inner codes [45–47].

The main novelty of the proposed construction is the adaptation to cell-wise encoding. Similarly to the approach proposed in Orlitsky [48], we use binary codes which are interpreted over the symbol alphabet of the channel. We assume that m bits of the code are jointly encoded in a flash cell with a Gray code. Hence, for a TLC we have $m = 3$ and for

QLC we have $m = 4$, respectively. Due to the error characteristic of the flash, the probability of multiple bit errors in a single symbol of m bits is extremely low. This property can be used for the code construction. We briefly review the encoding and decoding of GC codes. For more details, we refer to [44,49].

4.1. Encoding

A GC code is constructed from L outer RS codes over the extension field $GF(2^{m_a})$ and L binary inner BCH codes. All outer codes have the length n_a. We denote the parameters of the l-th outer code by $\mathcal{A}^{(l)}(2^{m_a}, n_a, k_a^{(l)}, d_a^{(l)})$, where $k_a^{(l)}$ is the dimension and $d_a^{(l)}$ the minimum Hamming distance. Similarly, the l-th inner code is denoted by $\mathcal{B}^{(l)}(n_b, k_b^{(l)}, d_b^{(l)})$. The inner codes are nested codes of length n_b, i.e., $\mathcal{B}^{L-1} \subset \mathcal{B}^{L-2} \subset \cdots \subset \mathcal{B}^0$. In other words, the code $\mathcal{B}^{(0)}$ is partitioned into sub-codes with smaller dimensions and a higher error-correction capability. The sub-codes of a BCH code can be constructed from the cyclotomic polynomials [44].

The concatenated code has length $N = n_b \cdot n_a$, and each codeword can be represented by an $n_b \times n_a$ binary matrix, as shown in Figure 5. The encoding starts with the outer codes, where $m_a k_a^{(0)}$ bits are mapped to the first outer codeword, $m_a k_a^{(1)}$ to the second, and so on. Hence, the overall dimension is $K = m_a \sum_{l=0}^{L-1} k_a^{(l)}$.

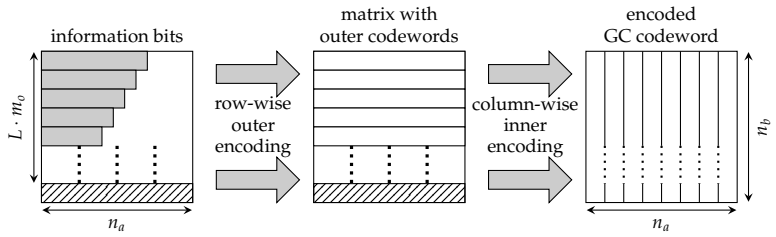

Figure 5. Encoding of a GC code.

Next, the code bits of the outer codes are encoded column-wise with the inner codes. The codeword of the j-th column is the sum of L codewords.

$$\mathbf{b}_j = \sum_{l=0}^{L-1} \mathbf{b}_j^{(l)}. \tag{10}$$

These codewords $\mathbf{b}_j^{(l)}$ are formed by encoding the symbols $a_{j,l}$ of the outer codewords with the corresponding sub-code $\mathcal{B}^{(l)}$. The symbol $a_{j,l}$ is the j-th symbol (m_a bits) of the outer code $\mathcal{A}^{(l)}$. The dimensions of $\mathcal{B}^{(l)}$ are $k_b^{(l)} = (L-l)m_a$. However, only m_a bits are mapped to each codeword $\mathbf{b}_j^{(l)}$. The remaining bits are filled with zero-padding; i.e., $(L-l-1)m_a$ zero bits are prefixed onto the symbol $a_j^{(l)}$ for encoding $\mathbf{b}_j^{(l)}$. Note that the j-th column $\mathbf{b}_j$ is a codeword of $\mathcal{B}^{(0)}$, because of the linearity of the nested codes.

4.2. Decoding

The decoding of a GC code is done level by level, where each level is decoded in the same succession of decoding steps. The decoding starts with the first level, $l = 0$. All columns of the received matrix are decoded with respect to $\mathcal{B}^{(0)}$. The inner decoding results in estimates of the outer code symbols $a_j^{(0)}$ of $\mathcal{A}^{(0)}$, where j is the column index. After decoding all columns with respect to $\mathcal{B}^{(0)}$, the outer RS code $\mathcal{A}^{(0)}$ can be decoded. After successful outer decoding, a partial decoding result $\hat{a}_j$ is available. This result has to be re-encoded using $\mathcal{B}^{(0)}$. The estimated codewords of the inner code $\mathcal{B}^{(0)}$ are subtracted from the received matrix before the next level can be decoded. The second level can be decoded with respect to $\mathcal{B}^{(1)}$. This code has smaller dimensions and greater error-correcting capability than the code $\mathcal{B}^{(0)}$. Consequently,

the error-correction capability of the outer codes can be reduced from level to level. Next, we consider the design of the code parameters for the outer and inner codes.

4.3. Code Design

There exist different design rules for multilevel coding [39]. In order to achieve low residual error rates, we consider a balanced error probability approach for the different levels of the generalized concatenated code. This design aims at balancing the decoding error probability in all levels and is motivated by the error analysis of the GC decoder based on the union bound. We briefly review this concept.

We consider error and erasure decoding for the outer RS codes [44]. The outer RS code $\mathcal{A}^{(l)}(2^{m_a}, n_a, k_a^{(l)}, d_a^{(l)})$ has minimum Hamming distance $d_a^{(l)} = n_a - k_a^{(l)} + 1$. With an odd minimum distance, it can correct up to $t_a^{(l)} = \frac{n_a - k_a^{(l)}}{2}$ errors and up to $n_a - k_a^{(l)}$ erasures. The probability $P_{a^{(l)}}$ of a decoding error with error and erasure decoding at the l-th level can be computed as follows [50].

$$P_{a^{(l)}} = \sum_{j=t_a^{(l)}+1}^{n_a} \sum_{i=d_a^{(l)}-2j}^{n_a} \binom{n_a}{j} \binom{n_a - j}{i} \cdot \rho_{b^{(l)}}^{j} \lambda_{b^{(l)}}^{i} (1 - \rho_{b^{(l)}} - \lambda_{b^{(l)}})^{n_a - j - i}, \tag{11}$$

where $\rho_{b^{(l)}}$ is the error probability for the inner code $\mathcal{B}^{(l)}$ and $\lambda_{b^{(l)}}$ is the erasure probability. Using the union bound, the word error rate P_e of the GC code can be estimated by the sum of the error probability over all levels

$$P_e \leq \sum_{l=0}^{L-1} P_{a^{(l)}}. \tag{12}$$

The sum in Equation (12) is dominated by the largest probability $P_{a^{(l)}}$. Hence, the design aims at balancing these error probabilities over all levels for a target channel error rate. The error and erasure probabilities for the inner codes can be estimated based on a computer simulation or by bounds.

We estimate the error probabilities with cell-wise encoding; i.e., we consider the m bits of a flash cell as a code symbol of the inner code. For instance, consider a code over a 2^m-ary alphabet with an even minimum distance of $d_b^{(l)}$. This code can correct all patterns where the number of errors is at most $t_b^{(l)} = \frac{d_b^{(l)} - 2}{2}$, whereas $t_b^{(l)} + 1$ errors can be detected as uncorrectable and an erasure is declared. Under the mentioned assumptions, the erasure probability is bounded by

$$\lambda_{b^{(l)}} \leq \sum_{j=t_b^{(l)}+1}^{\mathfrak{n}_\mathfrak{b}} \binom{\mathfrak{n}_\mathfrak{b}}{j} p_s^j (1 - p_s)^{\mathfrak{n}_\mathfrak{b} - j}, \tag{13}$$

where p_s denotes the symbol error rate and $\mathfrak{n}_\mathfrak{b}$ is the code length in 2^m-ary symbols. Similarly, the error probability is bounded by

$$\rho_{b^{(l)}} \leq \sum_{j=t_b^{(l)}+2}^{\mathfrak{n}_\mathfrak{b}} \binom{\mathfrak{n}_\mathfrak{b}}{j} p_s^j (1 - p_s)^{\mathfrak{n}_\mathfrak{b} - j}. \tag{14}$$

For the code design, we utilize the Gray labeling and the fact that the probability of an error to non-adjacent states is very small. Using a Gray code, the vast majority of symbol errors result in an error pattern where only a single bit out of the m bits of a cell can be in error. Hence, we use binary inner codes and interpret these codes over a 2^m-ary alphabet, where the number of symbols is $\mathfrak{n}_\mathfrak{b} = n_b/m$. In this case, the bounds

in Equations (13) and (14) are not rigorous, but they can still be used to estimate the error probabilities for the code design and to explain the gain of the cell-wise encoding.

For instance, consider the LSB page for Gray code 2 with the model explained in Section 3. Under the discussed assumptions, the bit error probability for the LSB page $p_{b,LSB}$ is approximately equal to the error rate p_e, and we have the symbol error rate $p_s \approx 1.75 p_e$. By designing a GC code for page-wise encoding, we can use Equations (13) and (14), but we have to substitute the code length $\mathfrak{n}_\mathfrak{b}$ with the length of the binary code n_b and p_s with the bit error rate. This leads to much higher error probabilities for the inner code with page-wise encoding. For instance, consider the binary code $\mathcal{B}(120, 112, 4)$ for $p_{b,LSB} = 0.01$. With cell-wise encoding, we have $\lambda_b \approx 0.13$ and $\rho_b \approx 0.025$ according to Equations (13) and (14), whereas with page-wise encoding we obtain $\lambda_b \approx 0.29$ and $\rho_b \approx 0.095$.

4.4. Code Examples

Finally, we constructed error correction codes with different sizes and rates that meet requirements for flash memories. The parameters of the GC codes were designed to guarantee a specified word error rate (WER) in certain channel conditions. The code parameters were optimized to balance the error rates in Equation (12) based on the bounds Equations (11), (13), and (14). We used GC codes at this point because we could guarantee a very low residual error rate. We designed the codes such that joint decoding guaranteed a target WER. For comparison, the same code was used for pagewise decoding.

For the mapping, the length of the inner code must be divisible by the number m of bits per cell, i.e., $m = 3$ for TLC and $m = 4$ for QLC. We used extended BCH codes for the inner codes. Hence, all inner codes had an even minimum Hamming distance. The additional parity bit was used to detect decoding failures. In the case of a decoding failure, an erasure was declared for the outer RS decoder.

First, we consider a code for a TLC Flash memory with rate $R \approx 0.9$ for a 4 kB payload of data. The detailed code parameters are shown in Table 3. For the second example, we consider a code of rate $R \approx 0.87$ for a 2 kB payload of data. Table 4 shows the corresponding code parameters. Table 5 provides a code for QLC Flash memory for 4 kB payload of data and rate $R \approx 0.9$ and Table 6 provides a code with almost the same code dimension and rate $R \approx 0.85$.

Table 3. Exemplary GC code (N = 36,414, K = 32,768, $R \approx 0.9$) for TLC (designed for WER 10^{-15}).

Level	Inner Code	Outer Code
1	$\mathcal{B}(153, 144, 4)$	$\mathcal{A}(2^8, 238, 148, 91)$
2	$\mathcal{B}(153, 136, 6)$	$\mathcal{A}(2^8, 238, 202, 37)$
3	$\mathcal{B}(153, 128, 8)$	$\mathcal{A}(2^8, 238, 220, 19)$
4	$\mathcal{B}(153, 120, 10)$	$\mathcal{A}(2^8, 238, 226, 13)$
5	$\mathcal{B}(153, 112, 12)$	$\mathcal{A}(2^8, 238, 230, 9)$
6	$\mathcal{B}(153, 104, 14)$	$\mathcal{A}(2^8, 238, 232, 7)$
7	$\mathcal{B}(153, 96, 16)$	$\mathcal{A}(2^8, 238, 234, 5)$
8	$\mathcal{B}(153, 88, 18)$	$\mathcal{A}(2^8, 238, 234, 5)$
9	$\mathcal{B}(149, 80, 20)$	$\mathcal{A}(2^8, 238, 236, 3)$
10	$\mathcal{B}(149, 72, 22)$	$\mathcal{A}(2^8, 238, 236, 3)$
11	$\mathcal{B}(149, 64, 24)$	$\mathcal{A}(2^8, 238, 236, 3)$
12	$\mathcal{B}(149, 56, 26)$	$\mathcal{A}(2^8, 238, 236, 3)$
13	$\mathcal{B}(149, 48, 28)$	$\mathcal{A}(2^8, 238, 236, 3)$
14–18	$\mathcal{B}(149, 40, 30)$	$\mathcal{A}(2^8, 238, 238, 1)$

Table 4. Exemplary GC code (N = 18,810, K = 16,386, $R \approx 0.87$) for TLC (designed for WER 10^{-15}).

Level	Inner Code	Outer Code
1	$\mathcal{B}(114, 106, 4)$	$\mathcal{A}(2^8, 165, 101, 65)$
2	$\mathcal{B}(113, 98, 6)$	$\mathcal{A}(2^8, 165, 137, 29)$
3	$\mathcal{B}(112, 90, 8)$	$\mathcal{A}(2^8, 165, 151, 17)$
4	$\mathcal{B}(111, 82, 10)$	$\mathcal{A}(2^8, 165, 155, 11)$
5	$\mathcal{B}(110, 74, 12)$	$\mathcal{A}(2^8, 165, 157, 9)$
6	$\mathcal{B}(109, 66, 14)$	$\mathcal{A}(2^8, 165, 159, 7)$
7	$\mathcal{B}(108, 58, 16)$	$\mathcal{A}(2^8, 165, 161, 5)$
8	$\mathcal{B}(114, 50, 22)$	$\mathcal{A}(2^8, 165, 163, 3)$
9	$\mathcal{B}(113, 42, 24)$	$\mathcal{A}(2^8, 165, 163, 3)$
10–13	$\mathcal{B}(112, 34, 28)$	$\mathcal{A}(2^8, 165, 165, 1)$

Table 5. Exemplary GC code (N = 36,300, K = 32,783, $R \approx 0.9$) for QLC (designed for WER 10^{-16}).

Level	Inner Code	Outer Code
1	$\mathcal{B}(220, 211, 4)$	$\mathcal{A}(2^8, 165, 51, 15)$
2	$\mathcal{B}(220, 203, 6)$	$\mathcal{A}(2^8, 165, 113, 53)$
3	$\mathcal{B}(220, 195, 8)$	$\mathcal{A}(2^8, 165, 139, 27)$
4	$\mathcal{B}(220, 187, 10)$	$\mathcal{A}(2^8, 165, 149, 17)$
5	$\mathcal{B}(220, 179, 12)$	$\mathcal{A}(2^8, 165, 155, 11)$
6	$\mathcal{B}(220, 171, 14)$	$\mathcal{A}(2^8, 165, 157, 9)$
7	$\mathcal{B}(220, 163, 16)$	$\mathcal{A}(2^8, 165, 159, 7)$
8	$\mathcal{B}(220, 155, 18)$	$\mathcal{A}(2^8, 165, 161, 5)$
9	$\mathcal{B}(216, 147, 20)$	$\mathcal{A}(2^8, 165, 161, 5)$
10	$\mathcal{B}(216, 139, 22)$	$\mathcal{A}(2^8, 165, 161, 5)$
11–15	$\mathcal{B}(216, 131, 24)$	$\mathcal{A}(2^8, 165, 163, 3)$
16–26	$\mathcal{B}(216, 91, 38)$	$\mathcal{A}(2^8, 165, 165, 1)$

Table 6. Exemplary GC code (N = 38,500, K = 32,773, $R \approx 0.85$) for QLC (designed for WER 10^{-16}).

Level	Inner Code	Outer Code
2	$\mathcal{B}(220, 203, 6)$	$\mathcal{A}(2^8, 175, 41, 135)$
3	$\mathcal{B}(220, 195, 8)$	$\mathcal{A}(2^8, 175, 101, 75)$
4	$\mathcal{B}(220, 187, 10)$	$\mathcal{A}(2^8, 175, 135, 41)$
5	$\mathcal{B}(220, 179, 12)$	$\mathcal{A}(2^8, 175, 151, 25)$
6	$\mathcal{B}(220, 171, 14)$	$\mathcal{A}(2^8, 175, 159, 17)$
7	$\mathcal{B}(220, 163, 16)$	$\mathcal{A}(2^8, 175, 163, 13)$
8	$\mathcal{B}(220, 155, 18)$	$\mathcal{A}(2^8, 175, 167, 9)$
9	$\mathcal{B}(216, 147, 20)$	$\mathcal{A}(2^8, 175, 167, 9)$
10	$\mathcal{B}(216, 139, 22)$	$\mathcal{A}(2^8, 175, 169, 7)$
11	$\mathcal{B}(216, 131, 24)$	$\mathcal{A}(2^8, 175, 171, 5)$
12	$\mathcal{B}(216, 123, 26)$	$\mathcal{A}(2^8, 175, 171, 5)$
13	$\mathcal{B}(216, 115, 28)$	$\mathcal{A}(2^8, 175, 171, 5)$
14–18	$\mathcal{B}(216, 107, 30)$	$\mathcal{A}(2^8, 175, 173, 3)$
19–26	$\mathcal{B}(216, 67, 46)$	$\mathcal{A}(2^8, 175, 173, 3)$

5. Performance Analysis

In this section, we present numerical results for the GC codes with hard-input decoding. Each page of the flash cell has a different channel condition depending on the selected bit-labeling. It is common to plot the word error rate (WER) of an error correction code over the bit error probability of the flash memory. However, the different pages have different bit error rates. Hence, we plot the WER results versus the bit error probability averaged over all pages. Regarding the TLC flash memory, we consider Gray code 1 and Gray code 2 from Table 1. Remember that among the presented TLC codes, Gray code 1 has the best error balance over the pages, and it has the best labeling considering the channel capacities for page-wise read.

We compare the joint encoding to the page-wise encoding for the different pages. In the later case, the bits are interleaved over all pages. In contrast to [24], we applied hard-input decoding. Soft-input decoding would probably achieve higher gains but would also require channel estimation in order to determine the log-likelihood ratios for the different bits.

The channel model from Section 3 is based on measurements of a particular NAND flash memory. In order to analyze the codes for different flash types and different code rates, we calculated the error probabilities according to a unipolar amplitude-shift keying with additive Gaussian noise. This channel model approximates an ideal flash channel where all states have the same noise variance. Nevertheless, the different pages had different error rates and channel capacities due to the bit-labeling. The presented numerical results were calculated with the formulas provided in Section 4. For the analysis, we used the same GC code for each page and for the cell-wise approach.

The channel analysis procedure randomly generated binary symbols, and encoded and converted them to m-ary symbols. Then, an average error probability was calculated, the corresponding variance was determined according to the channel model and applied to the m-ary symbol. The symbol with variance was then hard decoded and converted back to a binary vector. This binary vector was subsequently decoded with the corresponding presented GC decoder.

Figure 6 presents results for the GC code from Table 3 at rate $R = 0.9$ using Gray codes 1 and 2. The page-wise encoding in the right figure considers the worst-case page, i.e., the LSB page for Gray code 2. This page has the highest bit error probability, whereas the MSB and CSB pages show relatively good error rates. The curves labeled with joint decoding correspond to a cell-wise encoding where all three pages are used uniformly. Hence, the bit error rate is the average value for all pages.

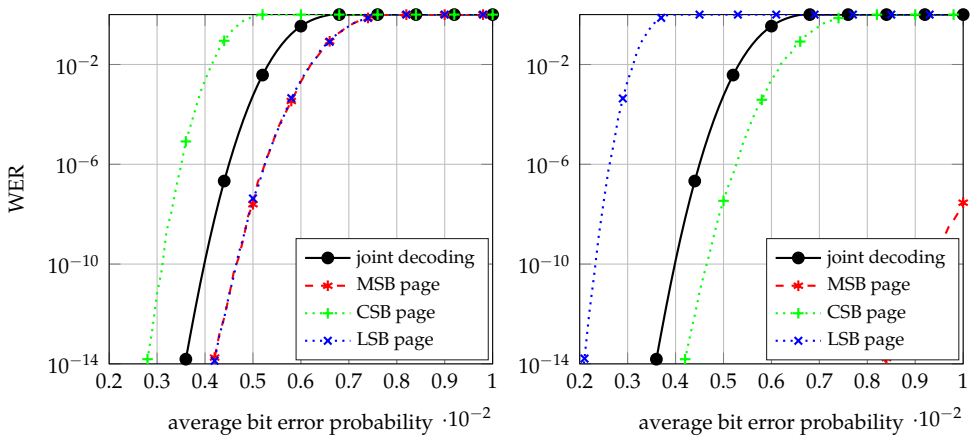

Figure 6. GC codes of rate $R = 0.9$ and 4 kB of payload data for a TLC model using Gray code 1 (**left**) and Gray code 2 (**right**).

Next, we compare the GC code with rate $R = 0.87$ with 2 kB from Table 4 for a TLC model. The left part of Figure 7 presents the word error rate for the different pages and the joint decoding using Gray code 1. The right part of the figure shows the performance with Gray code 2. Note that the MSB page with Gray code 2 achieves a WER of $<10^{-15}$ at the average RBER of 10^{-2} and is therefore not visible in the figure.

Finally, we present results for Gray codes for QLC Flash. A QLC flash has 4=four pages which are sequentially numbered, starting with the MSB page as page 1 and having the LSB page as page 4. Table 7 shows two possible Gray code labels for QLC Flash. Gray code 4 was presented in [51]. For this code, page 1 has three thresholds, and pages 2 to 4 have four thresholds for bit selection. This labeling results in well-balanced error probabilities. The page error probabilities for good channel conditions are $p_{b,page1} \approx \frac{3}{8} p_e$ and $p_{b,page2} = p_{b,page3} = p_{b,page4} \approx \frac{1}{4} p_e$.

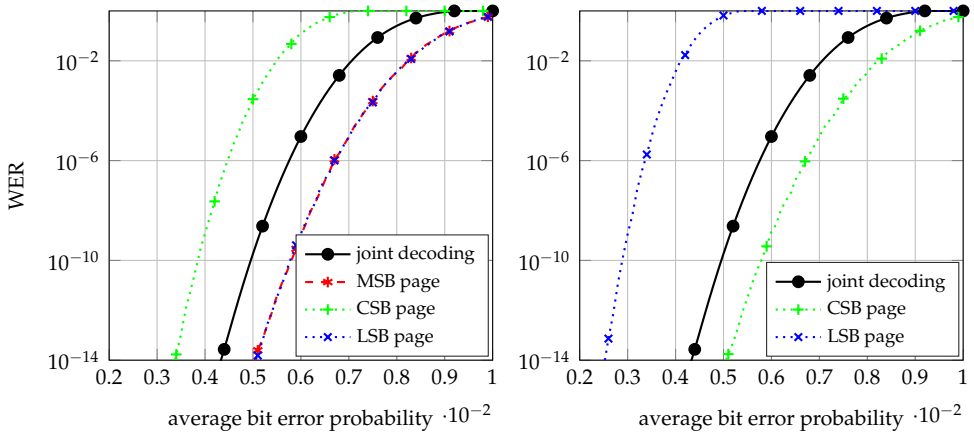

Figure 7. GC codes of rate $R = 0.87$ with 2 kB of payload data code for TLC model with pagewise and joint decoding using Gray code 1 (**left**) and Gray code 2 (**right**).

Table 7. Possible Gray code bit-labeling for a QLC flash cell.

Charge State	Page No.	S_0	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	S_{10}	S_{11}	S_{12}	S_{13}	S_{14}	S_{15}
Gray code 4	1	1	1	0	0	0	1	1	1	1	1	1	0	0	0	0	0
	2	1	0	0	0	0	0	0	1	1	1	0	0	1	1	1	1
	3	1	1	1	0	0	0	0	0	0	1	1	1	1	0	0	1
	4	1	1	1	1	0	0	1	1	0	0	0	0	0	0	1	1
Gray code 5	1	1	1	1	1	0	0	0	0	0	0	0	1	1	1	1	0
	2	1	1	1	0	0	0	0	1	1	1	1	1	0	0	0	0
	3	1	1	0	0	0	0	1	1	1	0	0	0	0	1	1	1
	4	1	0	0	0	0	1	1	1	0	0	1	1	1	1	0	0

The left part of Figure 8 presents numerical results using Gray code 4 and the GC code from Table 5. Pages 2, 3, and 4 provide the same WER, whereas page 1 provides a lower WER than the other pages. For this code, the joint encoding using the GC code results only in a small improvement compared to the performance of the worst-case pages.

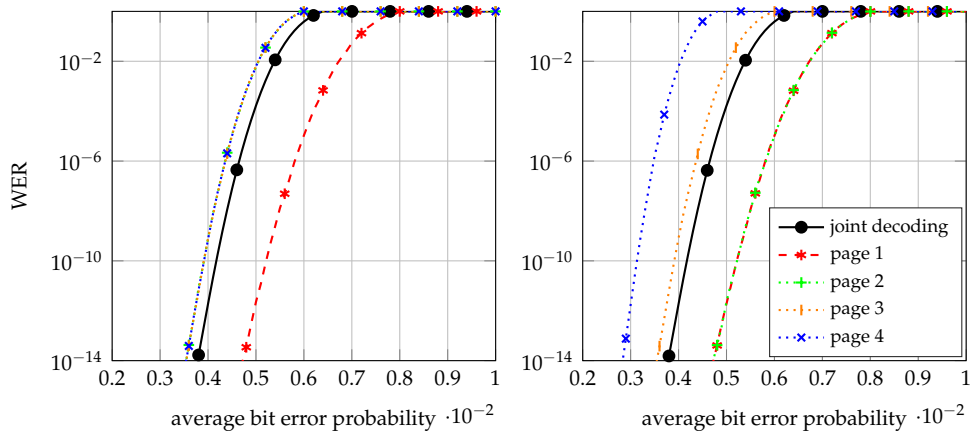

Figure 8. GC codes (from Table 5) of rate $R = 0.9$ and 4 kB of payload data for a QLC model using Gray code 4 (**left**) and Gray code 5 (**right**).

The right part of Figure 8 shows numerical results using Gray code 5 with the GC code from Table 5. Gray code 5 has 3 thresholds for page 1 and page 2, 4 thresholds for page 3, and 5 thresholds for page 4. The page error probabilities for good channel conditions using Gray code 5 are $p_{b,page1} = p_{b,page2} \approx \frac{3}{8} p_e$, $p_{b,page3} \approx \frac{1}{2} p_e$, and $p_{b,page4} \approx \frac{5}{8} p_e$. In this case, the GCC code with joint encoding achieved a significant gain compared to the worst-case pages.

In Figure 9, results are shown for two Gray codes for the GC code from Table 6. Note that in the left figure, the curves for page 2, 3, and 4 are on top of each other. With page-wise encoding, Gray code 5 provides better performance compared with Gray code 4. The joint encoding improves the reliability compared to the worst pages.

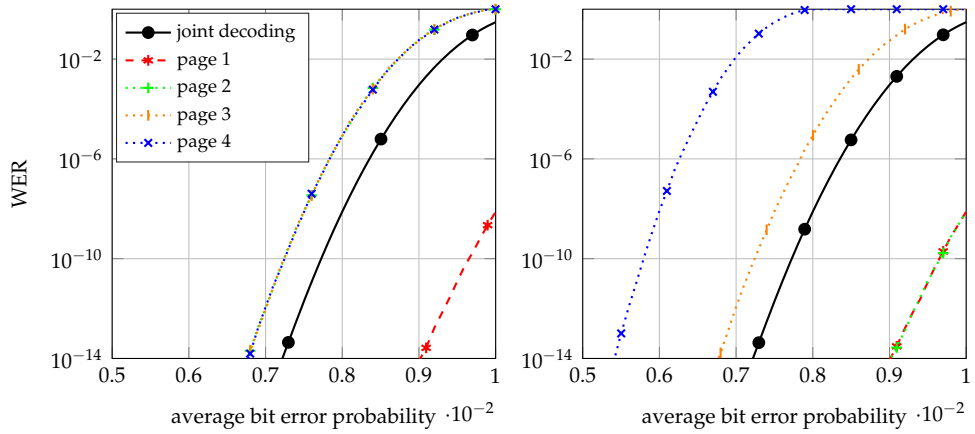

Figure 9. GC codes (from Table 6) of rate $R = 0.85$ and 4 kB of payload data for a QLC model using Gray code 4 (**left**) and Gray code 5 (**right**).

The presented results show that bit-labeling plays a significant role in page-wise decoding approaches. The average error rates of the different Gray codes are dominated by the worst page. The error correction performance can also be improved with larger code lengths. Similarly to the cell-wise encoding, a larger code length also impairs the random access read latency. However, a joint encoding achieves larger gains than longer codes over a single page.

In Figure 10, we compare the rate $R = 0.9$ and 4 kB of payload data codes for the QLC and TLC model with respect to the signal-to-noise ratio measured in terms of the energy per bit to noise power spectral density ratio (E_b/N_0). Storing more information bits per cell requires a higher signal-to-noise ratio in order to distinguish between different charge states.

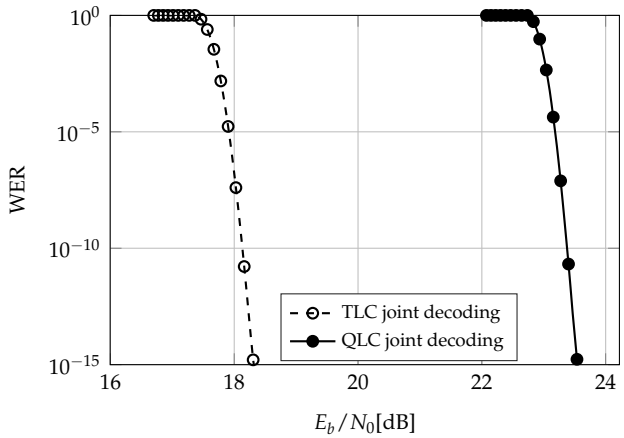

Figure 10. Comparison of TLC and QLC with respect to the signal-to-noise ratio; GC codes with joint coding, rate $R = 0.9$, and 4 kB of payload data.

6. Conclusions

In this work, we have investigated an error correction approach for TLC and QLC flash memories where all bits stored in a cell are jointly encoded using Gray labeling. This cell-wise encoding improves the achievable channel capacity compared with conventional independent page-wise encoding. The objective of the proposed approach is to avoid the costly soft-input decoding as long as possible. The performance of the joint decoding does not depend on the particular Gray code, whereas the error probabilities with page-wise encoding strongly depend on the bit-labeling. The average error probability of all pages is dominated by the page with the highest error probability. The presented results for joint decoding demonstrate a performance gain due to the cell-wise encoding compared to bit-interleaved coded modulation, where the bits are interleaved over all pages. A bit-interleaved coded modulation (BICM) approach similar to that in [24,25] would achieve better performance. However, such a method requires soft-input decoding and channel estimation to calculate log-likelihood ratios for the different bits depending on the estimated charge state.

The considered GC codes were designed to guarantee residual error probabilities of 10^{-15} and 10^{-16}. Such residual error rates are required for industrial applications. Moreover, the results presented in this work are focused on hard-input decoding. On the other hand, the cell-wise encoding also improves the achievable capacity for soft-input decoding. We think that the code design for cell-wise encoding with soft-input decoding is an interesting research direction.

PLC flash memories are already being developed [4]. PLC–NAND enables a large number of possible bit-labeling codes. Due to the larger number of bits per cell, codes that reduce the inter-cell interference might be an interesting approach for this technology [26].

Author Contributions: The research for this article was exclusively undertaken by D.N.B., J.-P.T., and J.F.; conceptualization and investigation, D.N.B., J.-P.T., and J.F.; software and validation, D.N.B. and J.-P.T.; writing—review and editing, D.N.B., J.-P.T., and J.F.; writing—original draft preparation, D.N.B., J.-P.T., and J.F.; supervision, project administration, and funding acquisition J.F. All authors have read and agreed to the published version of the manuscript.

Funding: The German Federal Ministry of Research and Education (BMBF) supported the research for this article (16ES1045) as part of the PENTA project 17013 XSR-FMC.

Acknowledgments: The authors would like to thank Hyperstone GmbH, Konstanz, for supporting the research for this article.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analysis, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Kang, J.; Huang, P.; Han, R.; Xiang, Y.; Cui, X.; Liu, X. Flash-based Computing in-Memory Scheme for IOT. In Proceedings of the 2019 IEEE 13th International Conference on ASIC (ASICON), Chongqing, China, 29 October–1 November 2019; pp. 1–4. [CrossRef]
2. Bennett, S.; Sullivan, J. NAND Flash Memory and Its Place in IoT. In Proceedings of the 2021 32nd Irish Signals and Systems Conference (ISSC), Athlone, Ireland, 10–11 June 2021; pp. 1–6. [CrossRef]
3. Ohshima, S.J. Empowering Next-Generation Applications through FLASH Innovation. In Proceedings of the 2020 IEEE Symposium on VLSI Technology, Honolulu, HI, USA, 16–19 June 2020; pp. 1–4. [CrossRef]
4. Goda, A. Recent Progress on 3D NAND Flash Technologies. *Electronics* **2021**, *10*, 3156. [CrossRef]
5. Taranalli, V.; Uchikawa, H.; Siegel, P.H. Channel Models for Multi-Level Cell Flash Memories Based on Empirical Error Analysis. *IEEE Trans. Commun.* **2016**, *64*, 3169–3181. [CrossRef]
6. Spinelli, A.S.; Compagnoni, C.M.; Lacaita, A.L. Reliability of NAND Flash Memories: Planar Cells and Emerging Issues in 3D Devices. *Computers* **2017**, *6*, 16. [CrossRef]
7. Chen, B.; Zhang, X.; Wang, Z. Error correction for multi-level NAND flash memory using Reed-Solomon codes. In Proceedings of the 2008 IEEE Workshop on Signal Processing Systems, Washington, DC, USA, 8–10 October 2008; pp. 94–99.
8. Freudenberger, J.; Spinner, J. A configurable Bose-Chaudhuri-Hocquenghem codec architecture for flash controller applications. *J. Circuits Syst. Comput.* **2014**, *23*, 1450019. [CrossRef]
9. Dong, G.; Xie, N.; Zhang, T. On the Use of Soft-Decision Error-Correction Codes in NAND Flash Memory. *IEEE Trans. Circuits Syst. Regul. Pap.* **2011**, *58*, 429–439. [CrossRef]
10. Wang, J.; Vakilinia, K.; Chen, T.Y.; Courtade, T.; Dong, G.; Zhang, T.; Shankar, H.; Wesel, R. Enhanced Precision Through Multiple Reads for LDPC Decoding in Flash Memories. *IEEE J. Sel. Areas Commun.* **2014**, *32*, 880–891. [CrossRef]
11. Freudenberger, J.; Rajab, M.; Shavgulidze, S. Estimation of channel state information for non-volatile flash memories. In Proceedings of the IEEE 7th International Conference on Consumer Electronics (ICCE), Berlin, Germany, 3–6 September 2017.
12. Rajab, M.; Thiers, J.; Freudenberger, J. Read Threshold Calibration for Non-Volatile Flash Memories. In Proceedings of the IEEE 9th International Conference on Consumer Electronics (ICCE-Berlin), Berlin, Germany, 8–11 September 2019; pp. 119–123.
13. Zhao, K.; Zhao, W.; Sun, H.; Zhang, X.; Zheng, N.; Zhang, T. LDPC-in-SSD: Making Advanced Error Correction Codes Work Effectively in Solid State Drives. In Proceedings of the 11th USENIX Conference on File and Storage Technologies (FAST 13), San Jose, CA, USA, 12–15 February 2013; pp. 243–256.
14. Lin, W.; Yen, S.W.; Hsu, Y.C.; Lin, Y.H.; Liang, L.C.; Wang, T.C.; Shih, P.Y.; Lai, K.H.; Cheng, K.Y.; Chang, C.Y. A low power and ultra high reliability LDPC error correction engine with Digital Signal Processing for embedded NAND Flash Controller in 40 nm CMOS. In Proceedings of the Symposium on VLSI Circuits Digest of Technical Papers, Honolulu, HI, USA, 10–13 June 2014; pp. 1–2.
15. *IEEE Std 1890-2018*; IEEE Standard for Error Correction Coding of Flash Memory Using Low-Density Parity Check Codes. IEEE: Piscataway, NJ, USA, 2019; pp. 1–51. [CrossRef]
16. Liao, Y.C.; Lin, C.; Chang, H.C.; Lin, S. A (21150, 19050) GC-LDPC Decoder for NAND Flash Applications. *IEEE Trans. Circuits Syst. Regul. Pap.* **2019**, *66*, 1219–1230. [CrossRef]
17. Richardson, T. Error floors of LDPC codes. In Proceedings of the Annual Allerton Conference on Communication Control and Computing, Monticello, IL, USA, 1–3 October 2003; Volume 41, pp. 1426–1435.
18. Spinner, J.; Freudenberger, J. Decoder Architecture for Generalized Concatenated Codes. *IET Circuits Devices Syst.* **2015**, *9*, 328–335. [CrossRef]
19. Zhilin, I.; Kreschuk, A.; Zyablov, V. Generalized concatenated codes with soft decoding of inner and outer codes. In Proceedings of the International Symposium on Information Theory and Its Applications (ISITA), Monterey, CA, USA, 30 October–2 November 2016; pp. 290–294.
20. Spinner, J.; Freudenberger, J.; Shavgulidze, S. A Soft Input Decoding Algorithm for Generalized Concatenated Codes. *IEEE Trans. Commun.* **2016**, *64*, 3585–3595. [CrossRef]
21. Spinner, J.; Rohweder, D.; Freudenberger, J. Soft input decoder for high-rate generalised concatenated codes. *IET Circuits Devices Syst.* **2018**, *12*, 432–438. [CrossRef]
22. Rajab, M.; Shavgulidze, S.; Freudenberger, J. Soft-input bit-flipping decoding of generalised concatenated codes for application in non-volatile flash memories. *IET Commun.* **2019**, *13*, 460–467. [CrossRef]
23. Thiers, J.P.; Bailon, D.N.; Freudenberger, J. Bit-Labeling and Page Capacities of TLC Non-Volatile Flash Memories. In Proceedings of the IEEE 10th International Conference on Consumer Electronics (ICCE-Berlin), Berlin, Germany, 9–11 November 2020; pp. 1–6. [CrossRef]
24. Lee, S.; Kim, D.; Ha, J. A paired-page reading scheme for NAND flash memory. In Proceedings of the 2016 International Conference on Information and Communication Technology Convergence (ICTC), Jeju, Korea, 19–21 October 2016. [CrossRef]
25. Wong, N.; Liang, E.; Wang, H.; Ranganathan, S.V.S.; Wesel, R.D. Decoding Flash Memory with Progressive Reads and Independent vs. Joint Encoding of Bits in a Cell. In Proceedings of the 2019 IEEE Global Communications Conference (GLOBECOM), Waikoloa, HI, USA, 9–13 December 2019. [CrossRef]

26. Hareedy, A.; Dabak, B.; Calderbank, R. Q-ary Asymmetric LOCO Codes: Constrained Codes Supporting Flash Evolution. In Proceedings of the 2020 IEEE International Symposium on Information Theory (ISIT), Los Angeles, CA, USA, 21–26 June 2020. [CrossRef]
27. Gabrys, R.; Dolecek, L. Constructions of Non-Binary WOM-Codes for Multilevel Flash Memories. *IEEE Trans. Inf. Theory* **2015**, *61*, 1905–1919. [CrossRef]
28. Yadgar, G.; Yaakobi, E.; Margaglia, F.; Li, Y.; Yucovich, A.; Bundak, N.; Gilon, L.; Yakovi, N.; Schuster, A.; Brinkmann, A. An Analysis of Flash Page Reuse with WOM Codes. *ACM Trans. Storage* **2018**, *14*, 1–39. [CrossRef]
29. Cai, Y.; Haratsch, E.F.; Mutlu, O.; Mai, K. Threshold voltage distribution in MLC NAND flash memory: Characterization, analysis, and modeling. In Proceedings of the 2013 Design, Automation Test in Europe Conference Exhibition (DATE), Grenoble, France, 18–22 March 2013; pp. 1285–1290. [CrossRef]
30. Amoroso, S.M.; Compagnoni, C.M.; Ghetti, A.; Gerrer, L.; Spinelli, A.S.; Lacaita, A.L.; Asenov, A. Investigation of the RTN Distribution of Nanoscale MOS Devices From Subthreshold to On-State. *IEEE Electron Device Lett.* **2013**, *34*, 683–685. [CrossRef]
31. Lee, D.; Sung, W. Estimation of NAND Flash Memory Threshold Voltage Distribution for Optimum Soft-Decision Error Correction. *IEEE Trans. Signal Process.* **2013**, *61*, 440–449. [CrossRef]
32. Chung, Y.T.; Huang, T.I.; Li, C.W.; Chou, Y.L.; Chiu, J.P.; Wang, T.; Lee, M.Y.; Chen, K.C.; Lu, C.Y. V_t Retention Distribution Tail in a Multitime-Program MLC SONOS Memory Due to a Random-Program-Charge-Induced Current-Path Percolation Effect. *IEEE Trans. Electron Devices* **2012**, *59*, 1371–1376. [CrossRef]
33. Monzio Compagnoni, C.; Ghidotti, M.; Lacaita, A.L.; Spinelli, A.S.; Visconti, A. Random Telegraph Noise Effect on the Programmed Threshold-Voltage Distribution of Flash Memories. *IEEE Electron Device Lett.* **2009**, *30*, 984–986. [CrossRef]
34. Parnell, T.; Papandreou, N.; Mittelholzer, T.; Pozidis, H. Modelling of the threshold voltage distributions of sub-20nm NAND flash memory. In Proceedings of the IEEE Global Communications Conference, Austin, TX, USA, 8–12 December 2014; pp. 2351–2356.
35. Micheloni, R. (Ed.) *3D Flash Memories*; Springer: Dordrecht, The Netherlands, 2016. [CrossRef]
36. Micheloni, R.; Crippa, L.; Marelli, A. *Inside NAND Flash Memories*; Springer: Dordrecht, The Netherlands, 2010. [CrossRef]
37. Cho, S.; Kim, D.; Choi, J.; Ha, J. Block-Wise Concatenated BCH Codes for NAND Flash Memories. *IEEE Trans. Commun.* **2014**, *62*, 1164–1177. [CrossRef]
38. Parnell, T.; Dünner, C.; Mittelholzer, T.; Papandreou, N. Capacity of the MLC NAND Flash Channel. *IEEE J. Sel. Areas Commun.* **2016**, *34*, 2354–2365. [CrossRef]
39. Wachsmann, U.; Fischer, R.; Huber, J. Multilevel codes: Theoretical concepts and practical design rules. *IEEE Trans. Inf. Theory* **1999**, *45*, 1361–1391. [CrossRef]
40. Gallager, R.G. *Information Theory And Reliable Communication*; John Wiley & Sons, Inc.: New York, NY, USA, 1968.
41. Proakis, J.; Salehi, M. *Digital Communications*; McGraw-Hill Science/Engineering/Math: New York, NY, USA, 2007; p. 1168.
42. Zyablov, V.; Shavgulidze, S.; Bossert, M. An Introduction to Generalized Concatenated Codes. *Eur. Trans. Telecommun.* **1999**, *10*, 609–622. [CrossRef]
43. Ungerboeck, G. Channel coding with multilevel/phase signals. *IEEE Trans. Inform. Theory* **1982**, *28*, 55–67. [CrossRef]
44. Bossert, M. *Channel Coding for Telecommunications*; John Wiley & Sons, Inc.: New York, NY, USA, 1999.
45. Trifonov, P.; Semenov, P. Generalized concatenated codes based on polar codes. In Proceedings of the 8th International Symposium on Wireless Communication Systems, Aachen, Germany, 6–9 November 2011; pp. 442–446. [CrossRef]
46. Trifonov, P. Efficient Design and Decoding of Polar Codes. *IEEE Trans. Commun.* **2012**, *60*, 3221–3227. [CrossRef]
47. Goldin, D.; Burshtein, D. Performance Bounds of Concatenated Polar Coding Schemes. *IEEE Trans. Inf. Theory* **2019**, *65*, 7131–7148. [CrossRef]
48. Orlitsky, A. Interactive Communication of Balanced Distributions and of Correlated Files. *SIAM J. Discret. Math.* **1993**, *6*, 548–564. [CrossRef]
49. Spinner, J. Channel Coding for Flash Memories. Ph.D. Thesis, Universität Konstanz, Konstanz, Germany, 2019.
50. Weiburn, L.; Cavers, J. Improved performance of Reed-Solomon decoding with the use of pilot signals for erasure generation. In Proceedings of the VTC 98. 48th IEEE Vehicular Technology Conference, Ottawa, ON, Canada, 21 May 1998; Volume 3, pp. 1930–1934.
51. Liu, S.; Zou, X. QLC NAND study and enhanced Gray coding methods for sixteen-level-based program algorithms. *Microelectron. J.* **2017**, *66*, 58–66. [CrossRef]