IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 16 May 2023, accepted 27 May 2023, date of publication 6 June 2023, date of current version 13 June 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3283445

RESEARCH ARTICLE

Read Reference Voltage Adaptation for NAND Flash Memories With Neural Networks Based on Sparse Histograms

DANIEL NICOLAS BAILON[1], (Graduate Student Member, IEEE), SERGO SHAVGULIDZE[2], (Member, IEEE), AND JÜRGEN FREUDENBERGER[3], (Member, IEEE)

[1]Institute for System Dynamics (ISD), HTWG Konstanz, University of Applied Sciences, 78462 Konstanz, Germany
[2]Faculty of Informatics and Control Systems, Georgian Technical University, 0175 Tbilisi, Georgia
[3]Agentur für Innovation in der Cybersicherheit GmbH (Cyberagentur), 06108 Halle (Saale), Germany

Corresponding author: Jürgen Freudenberger (jfreuden@htwg-konstanz.de)

This work was supported in part by the German Federal Ministry of Research and Education (BMBF) through the Project MEMTONOMY-2 under Grant 16ME0718, and in part by the Shota Rustaveli National Science Foundation under Grant STEM-22-340.

ABSTRACT Non-volatile NAND flash memories store information as an electrical charge. Different read reference voltages are applied to read the data. However, the threshold voltage distributions vary due to aging effects like program erase cycling and data retention time. It is necessary to adapt the read reference voltages for different life-cycle conditions to minimize the error probability during readout. In the past, methods based on pilot data or high-resolution threshold voltage histograms were proposed to estimate the changes in voltage distributions. In this work, we propose a machine learning approach with neural networks to estimate the read reference voltages. The proposed method utilizes sparse histogram data for the threshold voltage distributions. For reading the information from triple-level cell (TLC) memories, several read reference voltages are applied in sequence. We consider two histogram resolutions. The simplest histogram consists of the zero-and-one ratios for the hard decision read operation, whereas a higher resolution is obtained by considering the quantization levels for soft-input decoding. This approach does not require pilot data for the voltage adaptation. Furthermore, only a few measurements of extreme points of the threshold voltage distributions are required as training data. Measurements with different conditions verify the proposed approach. The resulting neural networks perform well under other life-cycle conditions.

INDEX TERMS Non-volatile NAND flash, channel estimation, machine learning, neural network, read reference adjustment.

I. INTRODUCTION

Due to its reliability against mechanical shocks, high storage density, low power consumption, and low-cost production, flash memory is widely used [1], [2], [3], [4]. For applications in safety-critical areas, e.g., Internet of Things, automotive, and medical devices, the storage solution must demonstrate high reliability. However, the reliability of flash memories declines during the lifetime. Advanced signal processing is required to adapt the reading process to the actual life-cycle conditions.

The associate editor coordinating the review of this manuscript and approving it for publication was Li He.

A flash cell stores data as an electrical charge. The threshold voltage is the minimum voltage needed to turn the cell on. The threshold voltage depends on the cell's charge state. To read the data, a certain read reference voltage (RRV) must be applied [5]. The threshold voltage distributions and the RRV change during the lifetime. They depend on the number of program/erase (P/E) cycles and the data-retention time (DRT) [6], [7] as well as on temperature-related charge losses, fluctuations during programming, read and write disturb effects [8], [9], [10], [11], [12]. All these effects influence the error probabilities and the optimal RRVs. To minimize the bit error rates, an adaptation of the RRV is required.

Different methods for RRV adaptation are proposed in the literature [9], [13], [14], [15], [16], [17], [18], [19]. In [14], a Gaussian mixture model (GMM) was used to estimate the threshold voltage distribution (TVD). However, GMM are not suitable to approximate the TVDs of current triple-level cell (TLC) NAND flash memories [19], [20], [21]. A low complexity calibration approach based on the decoding of metadata is proposed in [17] and [19]. This method requires either pilot data or a strong error correction code error-correction code (ECC) for the metadata and estimates the changes of the RRVs based on the observed number of bit errors in the metadata. A possible adaptation algorithm, using an error characteristic aware RRV acquisition scheme, was proposed in [22]. This approach gains the threshold voltage distribution polynomial to calculate the minimum point between two neighboring charge states, resulting in the optimal RRV. However, this method needs a high number of measurements and large latency to calculate a polynomial.

The life-cycle states of flash memories were estimated using machine-learning. An approach to predict process variations of flash devices is being investigated by [23] and [24] characterizes flash memory in different life-cycles to adapt error correction. An machine-learning approach to identify the life state of flash memory was proposed in [25]. Such methods can be trained based on measured data and avoid the difficult task of modeling the threshold distributions.

Methods to estimate the flash channel using neural network (NN) were proposed in [26] and [27]. In [26], a NN approach is proposed based on meta-information and ECC decoding, i.e., the knowledge of the number of P/E cycles, the frame error rate (FER), the average number of iterations of the LDPC decoder, and the bit flip ratio of the decoder are used as features. This method shows good results, but requires metadata and therefore needs additional readouts. Mei et al. recommend a different machine-learning approach using a recurrent NN to dynamically estimate the RRV of multi-level cell (MLC) flash memory [27]. The data for training and verification are generated by a theoretical Gaussian mixture model. The method is based on a deep learning technique in combination with the ECC decoder. The adaption procedure incurs additional read latency and power consumption and is proposed as a background operation when the system is idle.

In order to reduce the average read and decoding latency [28] presented an approach to predict the optimal decoding method called dynamic error recovery flow. This method does not predict optimal RRVs, rather it assumes optimal RRVs. The underlying NN is designed to be reusable for different flash memories with different process variations. The chosen flash model is a Gaussian channel and noise model. As input feature for the neural network the on-cell ratio, P/E cycles, wordline and page index are necessary. The on-cell ratio is the ratio between active and total cells at a fix predefined RRV. Note that for the classification in [28], two on-cell ratios at different RRV are required.

In this work, we use feedforward neural network (FNN) to predict the optimal RRVs, for TLC NAND flash memories with different life-cycle conditions. The features of the NN are sparse histograms. The sparse histogram component is similar to the description of the on-cell ratio from [28]. However, we do not calculate a ratio, instead we count the active cells between adjacent RRVs, and it is not strictly necessary to use predefined RRVs. In contrast to [26], the proposed approach uses only sparse histograms as features. Furthermore, it does not require any information about the current life-cycle condition. Reference [26] describes three different machine-learning approaches based on RRV adaptation: k-nearest neighbors (k-NN), nearest centroids (NC), and polynomial regression (PR). The highest complexity across these approaches is reached with k-NN. The other approaches require a similar storage overhead than the FNN approach presented in this paper for good results. Compared to the approach in [27], the proposed method does not use a Gaussian mixture model for training and verification. The NN network proposal from [27] is specialized for MLC flash memory, but requires much higher complexity than the algorithm used here. One reason is the used RNN, which is much more complex than an FNN. On the other hand, the number of neurons from [27] is considerably higher than the shallow NN approach that we employ in our proposed approach.

With TLC, soft-input decoding techniques are used to improve the ECC performance. For instance, the standard IEEE Std 1890-2018 provides strong LDPC codes for non-volatile memory applications [29]. Generalized concatenated codes are also suitable for flash-based memories that require a low guaranteed residual error rate [30], [31]. In this work, we show that sparse histograms of the RRV thresholds are good features for machine learning and inference of the optimal RRVs. We consider different histogram resolutions. The simplest histogram consists only of the relative frequencies of zeros and ones at the different RRVs. More information on the optimal RRVs is obtained by considering the quantization levels with soft input decoding. This approach requires no pilot data and enables efficient voltage adaptation in the case of a decoding failure, because no additional read operations are required for the adaptation.

For the training, we use measurements of flash memories with different life-cycle conditions. The NN has no knowledge about the life-cycle state of the flash memory and does not depend on any information from the decoder. It is demonstrated that near-optimal RRVs can be obtained with the suggested adaptation approach. The preliminary work in [32] uses high-resolution histograms, whereas the new histogram-based approach works with low resolution applicable in real-life scenarios.

The suggested method may be used in combination with other adaptation techniques. For instance, in [19] a calibration and tracking procedure for the RRVs is presented, where the tracking is required to adjust the prediction to the variations in different pages. This method requires metadata or pilot data

for the estimation. The new method omits the reading of the metadata but requires the histograms of the current RRVs. The first method is suitable for initial threshold calibration. However, a single adaptation step does not guarantee sufficient accuracy. Adaptation errors may lead to decoding failures. Moreover, the RRVs vary from page to page even in the same flash block. The proposed approach enables efficient voltage adaptation in the case of a decoding failure or voltage changes because no additional read operations are required. In the following, we also present a performance comparison between the method from [19] and our NN approach.

The remainder of this paper is structured as follows. First, we consider some basics of flash memories in Section II. Then we describe flash memory aging in more detail and describe the measurement data. Next, we introduce the NN approach in Section IV. In Section V we present numerical results. Finally, we conclude the paper in Section VI.

II. BASICS OF NAND FLASH MEMORY AND MEASUREMENT DATA

This section describes some details about TLC NAND flash memory for this work. More information about the flash architecture can be found in [5].

A flash cell is either a floating gate (FG) or a charge trap (CT) metal-oxide-semiconductor field-effect transistor (MOSFET). An electrical charge is stored corresponding to the data. This charge changes the threshold voltage V_{th} required to switch the transistor on, i.e., to enable a current between source to drain. In this work, we consider TLC NAND flash memories. Therefore, each cell stores three bits and there are eight possible charge states denoted as S_i. Each bit is mapped to a different page. These pages are referred to as most significant bit (MSB), center significant bit (CSB), and least significant bit (LSB) page, depending on the bit associated to the page.

A TVD represents the conditional probability that a cell is activated at a certain voltage given the programmed state. Often the TVDs are considered as Gaussian distributions as shown in Figure 1. The figure illustrates the eight charge states $S_0, \ldots, S_7$. The x-axis describes the threshold voltage, i.e., the gate voltage, and the y-axis the state dependent probability density. The voltages $V_{r0}, \ldots, V_{r6}$ illustrate the optimal RRVs which minimize the readout error probability.

There are different schemes to label the charge states with a binary representation, e.g., [33] presents optimized bit-labeling for TLC flash memory. Gray codes are particularly suitable for pagewise reading because the adjacent bit combination differs by only one position. We take the code in Table 1 from [33] as bit-labeling. As explained before, the charge states are distinguished by the RRVs. The bold lines in the bit pattern of Table 1 represent the decision boundaries between a binary '0' and '1' for the respective page. The MSB and LSB pages require two, and the CSB page needs three readouts to determine the corresponding bit. Note that it is not possible to create a code for TLC NAND flash with the same number of readouts for each page.

TABLE 1. Gray code bit-labeling for TLC flash cell used in this work.

page	S_0	S_1	S_2	S_3	S_4	S_5	S_6	S_7
MSB	1	1	1	0	0	0	0	1
CSB	1	1	0	0	1	1	0	0
LSB	1	0	0	0	0	1	1	1

A. READOUT FOR SOFT-INPUT DECODING

The RRVs are useful for hard-decision (HD) readout. The performance of ECC with HD is limited and a better performance is achieved with soft-decision (SD) decoding. This decoding method needs additional reliability information. For flash memory, a high resolution of the soft information requires more read operations and hence a higher readout latency and power consumption. Furthermore, more read operations increase the read disturb (RD) [8], [34]. Therefore, a small number of readouts is pursued.

Figure 2 shows two exemplary readout methods. SD readout with low resolution for LDPC codes are considered in [35]. Good performance is achieved with two additional readouts per reference, which results in one additional soft bit (SB). We can distinguish four intervals $l_{i,0}, l_{i,1}, l_{i,2}, l_{i,3}$ with this readout method as shown in Figure 2a. In [36] and [30] quantization with four additional readouts are considered which result in two additional SBs. Together with the hard bit, six quantization intervals $l_{i,0}, \ldots, l_{i,5}$ are generated. Figure 2b illustrates this method. In the following the soft readout types are called 1 SB and 2 SB method, respectively. Notice that in perfect conditions, the additional RRVs are chosen such that the mutual information is maximized [35].

III. MEASUREMENT DATA

Numerous physical effects change the TVDs. Repeated programming and erasing damages the tunnel oxide. The number of P/E cycles is a crucial metric of life expectancy [5]. An essential property of flash memory is retaining electrical charge, i.e., information, for a long time. For some applications, the ability to maintain data is more important than the ability to withstand many P/E cycles. The higher the number of P/E cycles, the worse the oxide layer, and the faster the cells can discharge. Therefore, P/E cycling reduces the possible DRT of the charges.

A persistent source of interference in 3D-NAND flash is the RD effect [5]. Reading a page several times without erasing may cause a charge increase in the cells of the pages. The probability of RD increases with the number of P/E cycles and is more likely on already damaged cells. Erasing the cells resets this disturbance, as this procedure discharges the cells completely [5].

The temperature of the flash memory may affect the possible operations on the flash cell. Significant temperature variations can lead to critical errors if the RRVs are not adjusted. This is particularly problematic with cross-temperature effects, i.e., writing at one temperature and reading at another. Especially in the automotive sector, integrated circuits have to pass special stress tests that require a certain

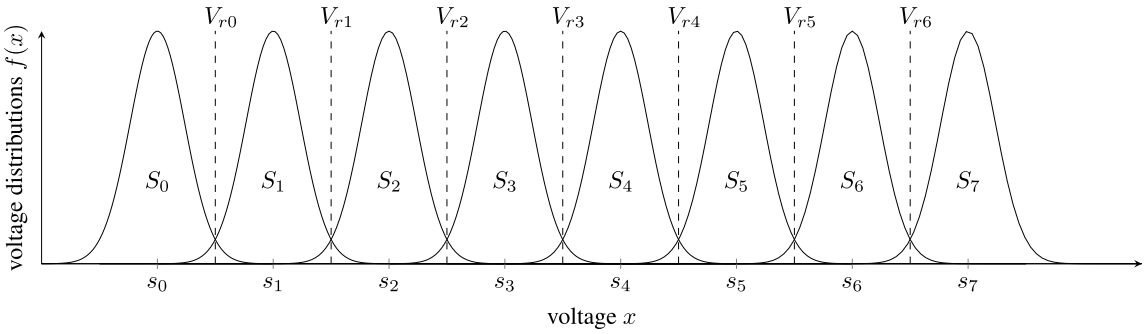

FIGURE 1. Example for the theoretic TVD of a TLC flash memory and optimal RRVs (dashed lines).

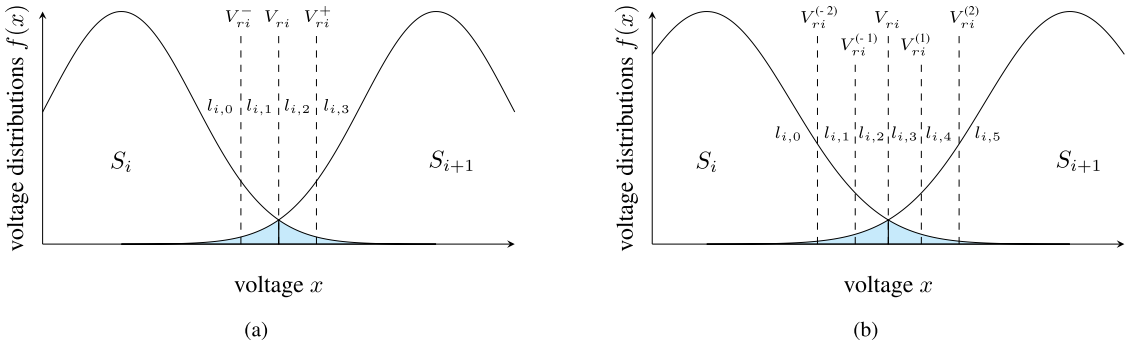

FIGURE 2. TVD of two soft readout possibilities with corresponding RRV. (a) Readout with 2 additional RRV generating 1 SB. (b) Readout with 4 additional RRV generating 2 SB.

operating temperature, e.g., for the AEC-Q100 Grade 3 qualification, the temperature range is −40 °C to 85 °C.

Each life-cycle condition changes the TVD and the associated optimal, i.e., error minimizing, RRV. Due to the complex modeling, measurements are necessary to characterize life-cycle conditions. For the analysis in this research, we conducted measurements under different life-cycle conditions. We consider an industrial BiCS TLC NAND flash memory that has a defined end-of-life (EOL) with 3000 P/E and one year data retention time. For this study, we consider P/E, DRT, RD and cross-temperature effects. The data set consists of 151 measurements of flash blocks in different life-cycles, i.e., with different aging.

One measurement consists of one flash block with 256 word-lines, i.e., 256 pages per block. The data contains 800 to 3000 P/E cycles with 200 P/E steps. DRT is simulated with a baking process where 83 h of baking correspond to one year data retention. We have measurements with 0 h, 13 h, 27 h, 42 h and 55 h baking time, which corresponds to approximately 0 h, 1.4×10^3 h, 2.8×10^3 h, 4.4×10^3 h and 5.4×10^3 h data retention. The temperatures used for cross-temperature measurements are −35 °C, 25 °C and 85 °C, where 25 °C is room temperature. This temperature range is similar to the AEC-Q100 Grade 3 qualification. Most measured sets have a combination of several effects, but there are no measurements with DRT and cross-temperature.

Figure 3 shows the TVD of two life states, showing the different variances and mean values. 200 pages are considered for these TVDs. The x-axis represents a discrete voltage step, and the y-axis represents the number of cells switched on. We use discrete voltage steps because the measurements' exact values are unavailable. Note that S_0 represents the deleted state. The measurement setup does not allow measuring the distribution of S_0.

Figure 4a and 4b show TVDs at reference V_{r6} under different life-cycle conditions. We see the distribution of S_6 and S_7 as well as the sum of both, i.e., the mixture distribution. The written data is generally unknown, and the observer retrieves only the mixture distributions. A complete flash page was read for this measurement, i.e., all cells in a word-line. The distributions are shifted and have different variances. Applying the same fixed RRVs would result in high error probabilities. The measurements for a single page are noisy. Hence, the estimation of the optimal read reference is not trivial.

Table 2 shows the optimal discrete RRVs with different life-cycle conditions. The optimal RRV produces the lowest error probability. Note that RRV V_{r0} is ignored because the measurement setup cannot correctly determine the distribution of S_0. We observe from Table 2 that large changes of the optimal RRVs occur for different life-cycle conditions.

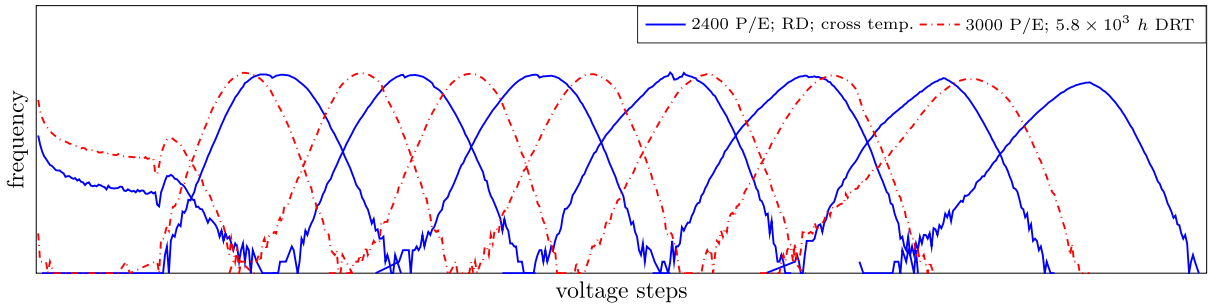

FIGURE 3. TVD for two measurements: blue curves correspond to 2400 PE cycles, no DRT, RD, −35 °C PTMP, 85 °C RTMP; red curves correspond to 3000 PE 5.4 × 10^3 *h* DRT, no RD, 25 °C PTMP, 25 °C RTMP. 200 pages are considered per TVD.

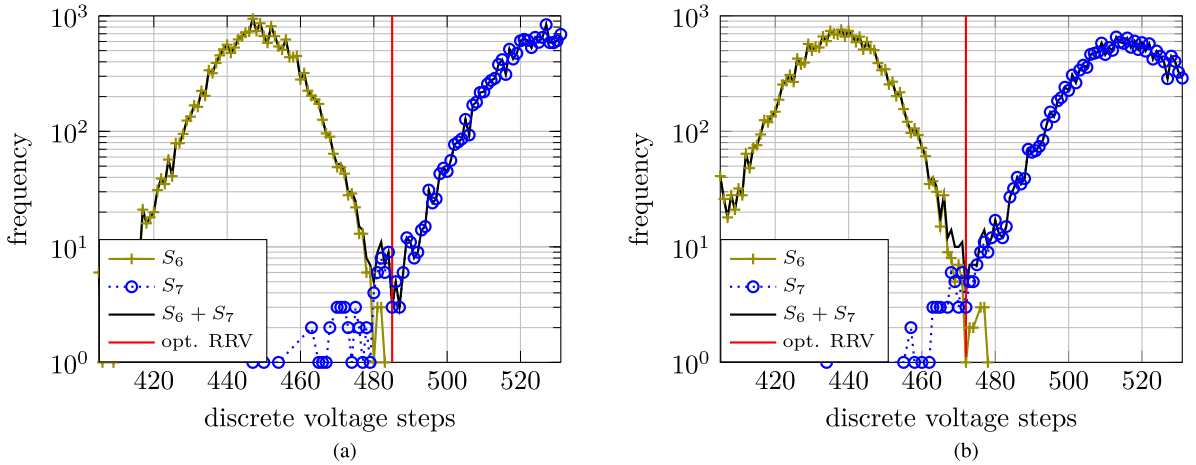

FIGURE 4. TVDs for S_6, S_7, and the sum of both distributions (mixture distribution) for a single page. (a) 800 PE, 0 *h* DRT, no RD, 25 °C PTMP, - 35 °C RTMP. (b) 3000 PE, 5.4×10^3 *h* DRT, no RD, 25 °C PTMP, 25 °C RTMP.

IV. READ REFERENCE VOLTAGE ADAPTATION

In this section, we introduce a machine-learning approach with NNs to adjust the RRVs. The input of the NNs should be sparse to keep the overhead for adaption small. Histograms of the threshold voltages with a high resolution, as considered in [32], cause high latency and power consumption. To reduce this overhead, we consider the sparse histograms of the readout for soft-input decoding. Thus, no additional information is required for the adaption when soft-input decoding is applied. Note that conditional histograms require known data. Hence, we consider the unconditioned histograms of the mixture distributions.

A. HISTOGRAM GENERATION

The sparse histograms are generated such that cells are counted which are turned on by an RRV in the interval between two reference voltages V_{rk} and V_{rk+1}. For instance, to measure the sparse histogram for the MSB page using the Gray code from Table 1 under hard decision readout, we need to apply two reference voltages. These define three voltage intervals. In [28], a similar figure is generated, called the on-cell ratio, where the number of cells turned on at a reference voltage is divided by the total number of cells.

The HD readout quantization levels are performed on standard RRV, and the quantization levels of the SD readout are selected to maximize mutual information. This is the usual approach for the selection of the SD RRVs. To avoid any modeling and consider real-world distributions, the training data for the NN is based on measurements of different life-cycle scenarios.

For the considered TLC flash, the MSB and LSB page require two and the CSB page needs three reference voltages to determine the corresponding bit. For MSB and LSB page, this results in histograms with three intervals for HD, seven intervals for one SB and eleven intervals for two SB. For the CSB page, we have four intervals for HD, ten intervals for one SB and 16 intervals for two SB.

B. INFORMATION ANALYSIS

The input features for the neural network must contain as much information as possible related to the expected output.

To investigate this dependency, we consider an information theoretic measure, i.e., we use the mutual information $I(X;Y)$, which is defined as [37]

$$I(X;Y) = H(X) - H(X|Y) = \sum_{x\in X}\sum_{y\in Y} f_{XY}(x,y)\cdot log_2\left(\frac{f_{X|Y}(x|y)}{f_X(x)}\right). \quad (1)$$

TABLE 2. Optimal discrete RRV for different life-cycle conditions.

	life-cycle condition					optimal read reference voltage step					
test set	P/E	DRT	RD	PTMP	RTMP	V_{r1}	V_{r2}	V_{r3}	V_{r4}	V_{r5}	V_{r6}
1	800	0	no	25 °C	-35 °C	142	202	261	322	384	449
2	800	0	yes	85 °C	-35 °C	142	202	261	322	384	449
3	1000	0	no	-35 °C	85 °C	154	219	286	356	424	489
4	1400	0	yes	-35 °C	85 °C	152	215	280	349	416	481
5	1600	0	yes	-35 °C	25 °C	158	223	290	357	425	490
6	1800	0	no	-35 °C	85 °C	148	210	272	339	405	472
7	2200	0	no	-35 °C	25 °C	151	214	277	342	408	477
8	2400	0	yes	-35 °C	85 °C	156	222	290	360	429	492
9	2600	0	no	85 °C	-35 °C	139	199	261	322	385	449
10	3000	0	no	25 °C	25 °C	142	204	265	331	397	464
11	3000	2.8×10^3 h	no	25 °C	25 °C	142	201	260	321	383	451
12	3000	5.8×10^3 h	no	25 °C	25 °C	136	191	250	310	370	434
	maximal difference					20	31	40	50	59	58

Note that the mutual information is non-negative and the minimum of the entropies of both random variables X and Y defines an upper bound.

The features discussed in [19] for determining the optimal RRV are obtained based on the successful decoding of the metadata. In contrast, we investigate features that rely on reading the payload data without decoding. In the following, we compare the mutual information between the different feature sets. First, we describe the features analyzed in [19], where the RRV are inferred from the number of errors in the metadata. We assume the same parameters and constraints for this analysis as in [19]. The mutual information between the number of errors E and the optimal reference voltage V_{opt} is denoted by $I(E; V_{opt})$. The mutual information is a function of the distance ΔV_{th} between the currently used read voltage and the optimal reference voltage V_{opt}.

Another feature in [19] is the asymmetry A between the error proportions at a reference voltage, where more errors on the right side of the distribution lead to $A = 1$ and $A = 0$ otherwise. The mutual information between the asymmetry and V_{opt} is denoted by $I(A; V_{opt})$.

Next, we describe a feature set that relies on reading the payload data without decoding. The variable B represents the number of active cells within each interval of quantization levels. Specifically, an instance of B is defined for each interval, resulting in two values in the case of a HD. To quantify the relationship between B and the optimal threshold voltage V_{opt}, we define the mutual information $I(B; V_{opt})$.

Figure 5 shows $I(E; V_{opt})$ and $I(A; V_{opt})$ regarding the metadata and $I(B; V_{opt})$ for the payload data, considering the counter for the right interval for HD readout, on the RRVs for the MSB page V_{r2} and V_{r6}. The calculations consider all available measurements and the x-axis of the figure is chosen, such that $\Delta V_{th} = 0$ corresponds to the mean optimum RRV.

The mutual information $I(E; V_{opt})$ shows noticeably higher values compared to $I(A; V_{opt})$, which is why in [19] the number of errors in the metadata was used as the primary feature. $I(B; V_{opt})$ show significantly larger values than $I(E; V_{opt})$ and $I(A; V_{opt})$. The progression shows that an RRV that is far from $\Delta V_{th} = 0$ provides even higher information. Moreover, the range of high mutual information is very large, which favors readout at different threshold voltages. Therefore, the analysis of the variable B with respect to the payload data leads to a larger statistical significance. This analysis supports adopting the number of active cells per interval as a feature over V_{opt}.

For reading out the soft information for SD decoding, the RRV intervals should be chosen to maximize the mutual information on each decision threshold. However, the mutual information depends on the noise level and it is not feasible to adapt the quantization intervals for each life-cycle condition. The investigation of the optimal quantization level by neural networks is not part of this study. Hence, the SD intervals are chosen to maximize mutual information in the EOL case. As shown in Figure 3 and described in [11], [19], [33], and [38], the charge distributions of the flash memory are highly asymmetric with exponential tails. This results in asymmetrical SD intervals.

C. NEURAL NETWORK APPROACH

In the following, we demonstrate that an adaptation with NNs using sparse histograms of the soft values and the used HD RRV can achieve near-optimal readout performance. However, we do not focus on implementing an optimized low-complexity NN approach for this problem. We use a fully-connected FNN in the following to adapt the RRV to different life-cycle conditions. As training function of the NN we use the Levenberg-Marquardt (LM) algorithm [39]. The LM algorithm is a well-suited approach for backpropagation training, that combines the gradient descent and the Gauss-Newton method. It is theoretically justified, easy to

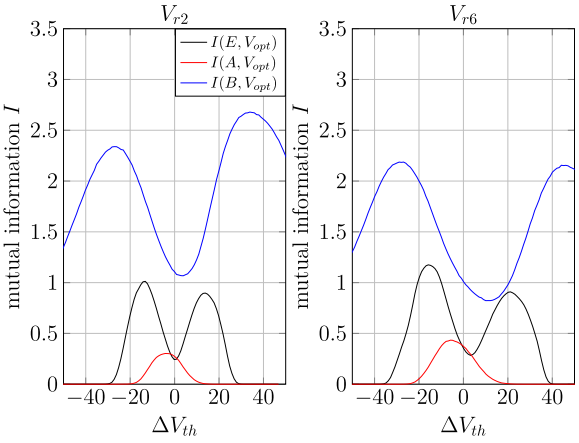

FIGURE 5. Mutual information (MI) on V_{r2} and V_{r6}. $I(E; V_{opt})$ - MI between the number of errors E and V_{opt}. $I(A; V_{opt})$ - MI between error asymmetry A and V_{opt} $I(B; V_{opt})$ - MI between B and V_{opt}.

implement, and has proven robustness and efficiency in practice [40], [41]. Further details about NNs using LM training algorithms are described in [42]. The FNN consists of an input layer with a certain number of inputs, depending on the readout method and the corresponding histogram, one or more hidden layers with a specific number of neurons, and an output layer with one or more outputs. As input we use the current RRVs and the sparse histograms, that are mapped to the range [-1, 1]. This is a common practice to normalize the inputs for sigmoid activation functions, to prevent fast saturation and accelerate the training process. Afterward, each neuron j in the hidden layers manipulates the input vector $\mathbf{x} = [x_1, \ldots, x_n]$ with a weighting factor w_i and a bias term b_i such that $a_j = \sum_{i=1}^{n} w_i \cdot x_i + b_i$. The weighting factors are defined during training with the LM algorithm. We use a supervised learning approach, i.e., input values are associated with target values during training. The weighting factors of the FNN are changed by comparing the target and actual output values of the FNN until the error between these is minimized. Each hidden neuron uses a hyperbolic tangent sigmoid transfer function $z_j = \text{tansig}(a_j) = 2/(1+\exp(-2a_j)) - 1$ as activation function, where z_j is the output value of a neuron. After the hidden layers, the outputs are connected to the output layer. This layer uses a linear transfer function to map the results to the required numeric range. As final output of the FNN, we get the optimal RRVs.

A diagram of such a network with three input neurons, one hidden layer with three neurons, and two output neurons with all connections is shown in Figure 6. In general this FNN approach has a storage overhead of $N(I_n + 1)$, where N is the number of neurons of a layer and I_n is the number of inputs. The specification of the total complexity in terms of floating operation points (FLOPs) depends strongly on the implementation of the mapping, tansig and linear transfer function for the corresponding input, hidden and output layer. Ignoring these special functions, the number of floating point additions and multiplications required corresponds to the storage overhead.

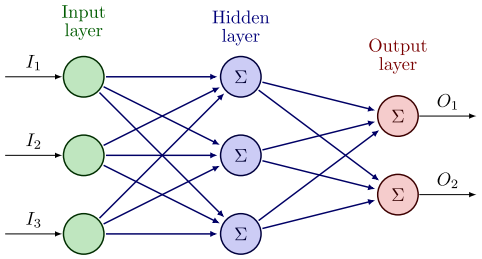

FIGURE 6. Diagram of a FNN with, three input neurons, one hidden layer with three hidden neurons and two output neurons.

The goal is to improve the generalization capability of the NN. If the neural network performs well on data, it has not seen before, it generalizes well. In our case, the trained FNN should predict the RRVs with unknown histograms of life-cycle conditions that are not represented in the training data. On the other hand, a high amount of training data can lead to overfitting. Overfitting causes the NN to fit very well to the training data, but the NN does not work correctly with any other data. One option to counteract overfitting and to improve the generalization is adding noise to the training data. The performance of the NN is tested and verified with independent sets.

Similar to [19], we assume that the data is initially read at a default RRV V_{ri}, i.e., these default RRVs are used to generate the histogram data for the input features. For one of the evaluations, the default RRVs are also used as static references without adaptation. For the default RRVs, we average the optimal RRVs of the measurement data with 800 P/E, 0 h DRT, no RD, 25 °C program temperature (PTMP) and −35 °C read temperature (RTMP). This is a nearly fresh condition. In addition, the SD intervals used to generate one and two SB at each V_{ri} were chosen to maximize the mutual information at EOL state. These default values are used for the training and verification of the NNs. To generalize the NN approach, we compute additional training data generating sparse histograms at different RRVs. This generates some uncertainty in the training process and avoids the use of fixed predefined RRVs.

To avoid overfitting and reduce the requirements for the measurements, we use as little training data as possible to demonstrate that a few specific measurements are a sufficient training basis.

V. PERFORMANCE OF READ REFERENCE VOLTAGE (RRV) ADAPTION

In this section, we present the results of the proposed adaptation method. Furthermore, we compare the results with the read reference calibration approach from [19].

Several NNs have been tested. We observed that a few hidden layers with few neurons are sufficient to solve the estimation problem. For the presented numerical results, we consider two cases:

1) NN with one hidden layer and five neurons (NN1)
2) NN with two hidden layers with five neurons each (NN2).

TABLE 3. Size of the NN (number of weights) in the amount of floating point numbers.

	NN1			NN2		
page	HD	1 SB	2 SB	HD	1 SB	2 SB
MSB/LSB	42	62	72	68	92	112
CSB	58	88	103	88	118	148

Depending on the used readout method and the target page, we get different numbers of input features for the FNN, which results in different storage overheads. Table 3 shows the needed storage for each readout method for the presented NN configurations.

We consider three different training sets (TSs) with different measurements:

1) training set 1 (TS1): 2 data sets, two cases of extreme measurements - consisting of test set 8 and 12.
2) training set 2 (TS2): 4 data sets, two cases of extreme measurements - consisting of test set 8 and 5; two data sets with condition 12.
3) training set 3 (TS3): 6 data sets consisting of TS2 with two additional data sets (test set 10 and 11).

Additionally, we use eight different RRV sets to generate multiple histograms of each training set. These RRV sets are equally distributed between the minimum and maximum of the mean RRVs of the data sets. In total, we use for the training 256 pages per data set. The verification is performed with other data sets but includes similar life-cycle conditions.

To compare the statistical performance of the different training methods we use boxplot figures. The boxplot is constructed such that the top and bottom of each box correspond to 75% and 25% of the samples, respectively. The red line in the box indicates the median. The top and bottom whisker lengths are adjusted to be three times the length of the interquartile range. Values exceeding this range are considered extreme outliers and are not considered in the boxplot figures. One measurement set consists of 256 pages, i.e., the total number of pages for verification is $151 \cdot 256 = 38\ 656$. Each page has a payload of 4 kB with a code rate $R = 0.9$, resulting in $36\ 490$ bits per page.

In addition, we consider the adaptation of the RRV on individual measurement sets. As shown in Figure 3, the TVD of flash memory shifts at different life-cycle conditions, and therefore the optimal RRVs are shifted, too. To emulate other life-cycle conditions, we shift the measurement data and recalculate the optimal RRV. For the numerical results, the results of five NNs were averaged to avoid anomalies.

We concentrate on applying the NN on the MSB page as an example, i.e., we predict V_{r2} and V_{r6}. According to the bit-labeling on Table 1, the charge states S_3 - S_6 are assigned to logical '0', and S_0, S_1, S_2, S_7 are assigned to logical '1'.

The boxplots represent bit error rates. Additionally, we show two residual error probabilities of an LDPC code with a code rate of $R = 0.91$ and payload length of 1 kB from [29]. This code has been standardized for flash memories with a high storage rate and the shortness of the code enables fast processing. It is possible to split a longer code to achieve faster performance through parallelization. At a bit error rate (BER) of 0.011, this code reaches a residual block error probability of about 10^{-6}, wheres for a bit error rate of 0.019, the decoder fails with a probability close to one. This BER range is very narrow and shows how sensitive the error-correcting is to estimation errors. Using the default RRV, approximately 37% of the measured pages cannot be successfully decoded. In contrast, using the optimal RRVs, 98% of the pages reach a block error rate below 10^{-6}. However, some extreme outliers (0.2% of all codewords) are not decodable even with perfect RRV adaptation.

First, we assume the case that the readings are taken with the default RRV. We compare all the results with the RRV calibration algorithm based on metadata from [19] denoted as RRV calibration. Figure 7a shows numerical results using histograms with one SB for the MSB page. This method results in seven histogram values, i.e., the input features consist of these seven values plus two values for the HD RRVs, resulting in nine input features for the FNN. For the small training sets (TS1), the NN with two hidden layers (NN2) outperforms the NN with one hidden layer (NN1). With the more extensive data set (TS3), both NNs achieve close to optimal performance. The predicted RRVs from the NNs provide in all cases a lower error rate than the static default RRVs. The static RRVs lead to three times more errors on average.

The NNs using two SBs readout show similar results as using one SB. We observe, that the additional histogram values do not significantly improve the resulting mean error rate. Therefore, we omit numerical results.

In the following, we consider numerical results using a HD readout for the histograms as input to the NN. This means that we generate only five input values for the NN, specifically three values as histogram data and two values for the RRVs for the MSB page. As can be seen in Figure 7b the estimation based on the HD readout always leads to some performance loss for small sets of training data. Depending on the selected neural network and training method, the hard decision approach achieves similar results as the RRV calibration from [19].

Next, we consider a two-step process, first performing the RRV calibration of [19] and then using the presented FNN approach to improve the adaptation. This scenario is more realistic since an SD readout would only be used if the calibration is not successful. The histograms are generated with the RRV from the calibration result, i.e., for each page we use the RRV suggested by the calibration. For this analysis, we use the same NN and data sets as before. We analyze the method where the calibration is performed first, followed by an SD readout. Figure 8a shows the results using 1 SB. The 2 SB approach show almost the same results in terms of error rate and therefore we omit numerical results at this point.

It is possible to generate more training data by data augmentation (DA). In our case, shifting the measured histogram

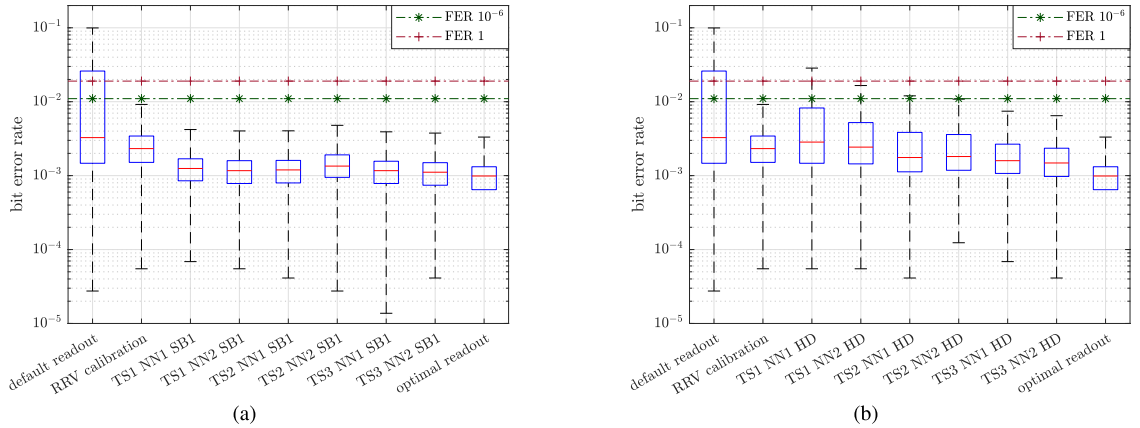

FIGURE 7. NN based on default readout against RRV calibration: Bit error rates for the MSB page with different readout methods and training strategies compared to RRV calibration strategy based on metadata from [19]. (a) Quantization with one SB SD readout. (b) Quantization based on HD readout.

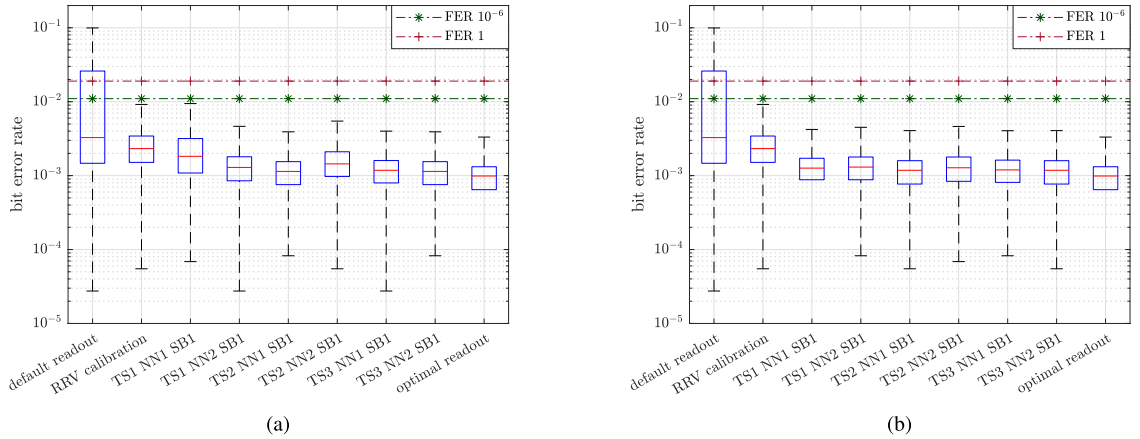

FIGURE 8. Two-step process against RRV calibration: Bit error rates for the MSB page on NN with different training strategies using one SB SD readout after using RRV calibration compared to RRV calibration strategy based on metadata from [19]. (a) Training without DA. (b) Training with DA.

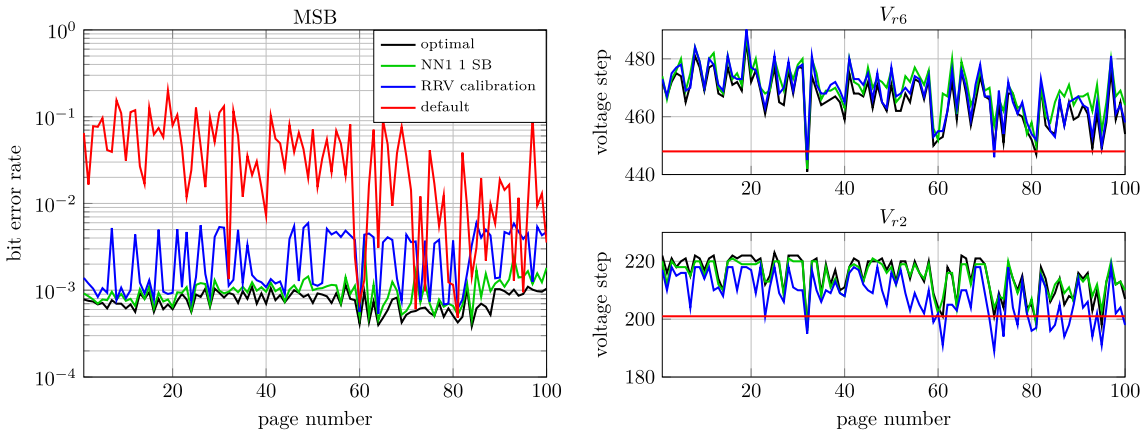

FIGURE 9. Results with 1 SB SD readout on RRV of the calibration method (two-step process) for using NN1 using training set 2 (TS2) against RRV calibration on pages with 3000 P/E, 1.4 × 10^3 *h* DRT, no RD, 25 °C PTMP and 25 °C RTMP.

sets can lead to emulating other life-cycle conditions. This is motivated by the observation that small changes in the life-cycle conditions typically lead only to changes in the mean value, but do not affect the shape of the distribution.

DA allows reducing the number of life-cycle conditions used for training and improves the generalization capability of the NNs. We limit the use of DA in this paper to doubling the data in the TSs. In Figure 8b we apply DA to enhance the training sets. In this case, DA only leads to a significant improvement in performance for TS1, i.e., only in cases where little training data is available.

The boxplot graphs give an insight into the resulting error rates using different NNs. In the following, we discuss the performance of one exemplary life-cycle condition measurement considering the variations over pages. One measurement consists of 256 consecutive pages, and though these are noisy measurements, we only show partial sections of a measurement.

Figure 9 shows an example of the two-step process, consisting of applying the RRV calibration from [19] first and afterward the neural network approach in case of a decoding failure. The default readout would lead to a high error rate, which is an extensive problem in many scenarios. Therefore, an adjustment of the RRV would be necessary in any case. We show here results for NN1 with TS2. This example shows that the two step-process enhances the performance significantly, such that near-optimum results are achievable. Note that in this case, both NN approaches reach almost the same performance and therefore numerical results of NN2 are not presented.

VI. CONCLUSION

The RRVs of non-volatile NAND flash memories are important concerning achievable storage reliability. The optimal RRVs depend on many aging effects, such as the number of P/E cycles, DRT, RD, and others. Using fixed default RRVs leads to high bit error rates and will eventually cause decoding failures.

In this work, we have addressed the issue of RRV prediction. The proposed prediction method is based on shallow FNNs, which utilize histograms of the soft-readout values as features. By using the training sets for different life-cycle conditions, we can show that only a few measurements cover the essential aspects of the TVD variation. The NNs approach achieves a near-optimum readout performance with a small data set. We have shown, that DA improves the results when very little data is used in the training sets. We have shown that an NN with an HD readout achieves similar results to conventional adaptation algorithms that employ pilot data algorithms. Compared to the calibration method from [19], HD readout achieves similar performance, while SD readout yields even better results. We also have demonstrated that the histogram generation for the input features of the NN with HD and SD readout does not require fixed predefined RRVs, enabling sequential application with other calibration methods.

The question arises at which point an adaptation of the RRV is necessary. If the HD decoding is not successful, the SD is triggered afterward. At this point, the NN can estimate the optimal RRV so that in case of continued decoding failure, the readout is done with an improved RRV. When reading again, the residual error rate could be reduced due to the adjusted RRV that HD decoding is successful.

A data-driven estimation method cannot consider all possible life-cycle conditions. Future research should consider other approaches for the DA, e.g., generating training data based on more accurate probability density functions for different life-cycle conditions.

ACKNOWLEDGMENT

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

REFERENCES

[1] H. Kim, S. Ahn, Y. G. Shin, K. Lee, and E. Jung, "Evolution of NAND flash memory: From 2D to 3D as a storage market leader," in *Proc. IEEE Int. Memory Workshop (IMW)*, May 2017, pp. 1–4.
[2] R. Godse, A. McPadden, V. Patel, and J. Yoon, "Memory technology enabling the next artificial intelligence revolution," in *Proc. IEEE Nanotechnol. Symp. (ANTS)*, Nov. 2018, pp. 1–4.
[3] J. F. Kang, P. Huang, R. Z. Han, Y. C. Xiang, X. L. Cui, and X. Y. Liu, "Flash-based computing in-memory scheme for IoT," in *Proc. IEEE 13th Int. Conf. ASIC (ASICON)*, Oct. 2019, pp. 1–4.
[4] S. Bennett and J. Sullivan, "NAND flash memory and its place in IoT," in *Proc. 32nd Irish Signals Syst. Conf. (ISSC)*, Jun. 2021, pp. 1–6.
[5] R. Micheloni, *3D Flash Memories*. Dordrecht, The Netherlands: Springer, 2016.
[6] Y. Cai, Y. Luo, E. F. Haratsch, K. Mai, and O. Mutlu, "Data retention in MLC NAND flash memory: Characterization, optimization, and recovery," in *Proc. IEEE 21st Int. Symp. High Perform. Comput. Archit. (HPCA)*, Feb. 2015, pp. 551–563.
[7] V. Taranalli, H. Uchikawa, and P. H. Siegel, "Channel models for multi-level cell flash memories based on empirical error analysis," *IEEE Trans. Commun.*, vol. 64, no. 8, pp. 3169–3181, Aug. 2016.
[8] H. Kaneko, "Error control coding for flash memory," in *Flash Memories*. Rijeka, Croatia: InTech, Sep. 2011.
[9] F. Sala, R. Gabrys, and L. Dolecek, "Dynamic threshold schemes for multi-level non-volatile memories," *IEEE Trans. Commun.*, vol. 61, no. 7, pp. 2624–2634, Jul. 2013.
[10] S. Qi, D. Feng, and J. Liu, "Optimal voltage signal sensing of NAND flash memmory for LDPC code," in *Proc. IEEE Workshop Signal Process. Syst. (SiPS)*, Oct. 2014, pp. 1–6.
[11] C. Cao and I. Fair, "Mitigation of inter-cell interference in flash memory with capacity-approaching variable-length constrained sequence codes," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 9, pp. 2366–2377, Sep. 2016.
[12] H. Yassine, J. Coon, M. Ismail, and H. Fletcher, "Towards an analytical model of NAND flash memory and the impact on channel decoding," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2016, pp. 1–6.
[13] H. Zhou, A. Jiang, and J. Bruck, "Error-correcting schemes with dynamic thresholds in nonvolatile memories," in *Proc. IEEE Int. Symp. Inf. Theory Proc.*, Jul. 2011, pp. 2143–2147.
[14] D. Lee and W. Sung, "Estimation of NAND flash memory threshold voltage distribution for optimum soft-decision error correction," *IEEE Trans. Signal Process.*, vol. 61, no. 2, pp. 440–449, Jan. 2013.
[15] D. Lee and W. Sung, "Decision directed estimation of threshold voltage distribution in NAND flash memory," *IEEE Trans. Signal Process.*, vol. 62, no. 4, pp. 919–927, Feb. 2014.
[16] B. Peleato, R. Agarwal, J. M. Cioffi, M. Qin, and P. H. Siegel, "Adaptive read thresholds for NAND flash," *IEEE Trans. Commun.*, vol. 63, no. 9, pp. 3069–3081, Sep. 2015.
[17] M. Rajab, J. Thiers, and J. Freudenberger, "Read threshold calibration for non-volatile flash memories," in *Proc. IEEE 9th Int. Conf. Consum. Electron. (ICCE-Berlin)*, Sep. 2019, pp. 109–113.
[18] L. Yang, Q. Wang, Q. Li, X. Yu, J. He, and Z. Huo, "Gradual channel estimation method for TLC NAND flash memory," *IEEE Embedded Syst. Lett.*, vol. 14, no. 1, pp. 7–10, Mar. 2022.
[19] J.-P. Thiers, D. N. Bailon, J. Freudenberger, and J. Lu, "Read reference calibration and tracking for non-volatile flash memories," *Electronics*, vol. 10, no. 18, p. 2306, Sep. 2021.

[20] A. Spinelli, C. Compagnoni, and A. Lacaita, "Reliability of NAND flash memories: Planar cells and emerging issues in 3D devices," *Computers*, vol. 6, no. 2, p. 16, Apr. 2017. [Online]. Available: http://www.mdpi.com/2073-431X/6/2/16

[21] J. Freudenberger, M. Rajab, and S. Shavgulidze, "A source and channel coding approach for improving flash memory endurance," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 26, no. 5, pp. 981–990, May 2018.

[22] M. Zhang, F. Wu, Q. Yu, W. Liu, Y. Wang, and C. Xie, "Exploiting error characteristic to optimize read voltage for 3-D NAND flash memory," *IEEE Trans. Electron Devices*, vol. 67, no. 12, pp. 5490–5496, Dec. 2020.

[23] J. K. Lee, K. Ko, and H. Shin, "Analysis on process variation effect of 3D NAND flash memory cell through machine learning model," in *Proc. 4th IEEE Electron Devices Technol. Manuf. Conf. (EDTM)*, Apr. 2020, pp. 1–4.

[24] C. Zambelli, G. Cancelliere, F. Riguzzi, E. Lamma, P. Olivo, A. Marelli, and R. Micheloni, "Characterization of TLC 3D-NAND flash endurance through machine learning for LDPC code rate optimization," in *Proc. IEEE Int. Memory Workshop (IMW)*, May 2017, pp. 1–4.

[25] S. Chattopadhyay, P. Kumari, B. Ray, and R. S. Chakraborty, "Machine learning assisted accurate estimation of usage duration and manufacturer for recycled and counterfeit flash memory detection," in *Proc. IEEE 28th Asian Test Symp. (ATS)*, Dec. 2019, pp. 1–6.

[26] H. Choe, J. Jee, S. Lim, S. M. Joe, I. H. Park, and H. Park, "Machine-Learning-Based read reference voltage estimation for NAND flash memory systems without knowledge of retention time," *IEEE Access*, vol. 8, pp. 176416–176429, 2020.

[27] Z. Mei, K. Cai, and X. He, "Deep learning-aided dynamic read thresholds design for multi-level-cell flash memories," *IEEE Trans. Commun.*, vol. 68, no. 5, pp. 2850–2862, May 2020.

[28] M. Hwang, J. Jee, J. Kang, H. Park, S. Lee, and J. Kim, "Dynamic error recovery flow prediction based on reusable machine learning for low latency NAND flash memory under process variation," *IEEE Access*, vol. 10, pp. 117715–117731, 2022.

[29] *IEEE Standard for Error Correction Coding of Flash Memory Using Low-Density Parity Check Codes*, IEEE Standard 1890–2018, Feb. 2019, pp. 1–51.

[30] J. Spinner, J. Freudenberger, and S. Shavgulidze, "A soft input decoding algorithm for generalized concatenated codes," *IEEE Trans. Commun.*, vol. 64, no. 9, pp. 3585–3595, Sep. 2016.

[31] M. Rajab, S. Shavgulidze, and J. Freudenberger, "Soft-input bit-flipping decoding of generalised concatenated codes for application in non-volatile flash memories," *IET Commun.*, vol. 13, no. 4, pp. 460–467, Mar. 2019.

[32] D. N. Bailon, G. Taburet, S. Shavgulidze, and J. Freudenberger, "Neural network aided reference voltage adaptation for NAND flash memory," in *Proc. IEEE 12th Int. Conf. Consum. Electron. (ICCE-Berlin)*, Sep. 2022, pp. 1–5.

[33] J. Thiers, D. N. Bailon, and J. Freudenberger, "Bit-labeling and page capacities of TLC non-volatile flash memories," in *Proc. IEEE 10th Int. Conf. Consum. Electron. (ICCE-Berlin)*, Nov. 2020, pp. 1–6.

[34] Y. Zhang, L. Jin, D. Jiang, X. Zou, H. Liu, and Z. Huo, "A novel read scheme for read disturbance suppression in 3D NAND flash memory," *IEEE Electron Device Lett.*, vol. 38, no. 12, pp. 1669–1672, Dec. 2017.

[35] J. Wang, T. Courtade, H. Shankar, and R. D. Wesel, "Soft information for LDPC decoding in flash: Mutual-information optimized quantization," in *Proc. IEEE Global Telecommun. Conf. (GLOBECOM)*, Dec. 2011, pp. 1–6.

[36] J. Freudenberger, M. Rajab, and S. Shavgulidze, "Estimation of channel state information for non-volatile flash memories," in *Proc. IEEE 7th Int. Conf. Consum. Electron. Berlin (ICCE-Berlin)*, Sep. 2017, pp. 69–73.

[37] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. Hoboken, NJ, USA: Wiley, 1991.

[38] Y. Cai, E. F. Haratsch, O. Mutlu, and K. Mai, "Threshold voltage distribution in MLC NAND flash memory: Characterization, analysis, and modeling," in *Proc. Design, Autom. Test Eur. Conf. Exhib. (DATE)*, Mar. 2013, pp. 1285–1290.

[39] D. W. Marquardt, "An algorithm for least-squares estimation of nonlinear parameters," *J. Soc. Ind. Appl. Math.*, vol. 11, no. 2, pp. 431–441, Jun. 1963.

[40] J. J. Moré, "The Levenberg–Marquardt algorithm: Implementation and theory," in *Numerical Analysis* (Lecture Notes in Mathematics), vol. 630, G. A. Watson, Eds. Berlin, Germany: Springer, 1978, pp. 105–116.

[41] J. Nocedal and S. J. Wright, *Numerical Optimization*, 2nd ed. New York, NY, USA: Springer, 2006, pp. 245–269.

[42] M. T. Hagan, H. B. Demuth, M. H. Beale, and O. D. Jesús, *Neural Network Design*, 2nd ed. Stillwater, OK, USA: M. Hagan, 2009.

DANIEL NICOLAS BAILON (Graduate Student Member, IEEE) received the B.Eng. degree in electrical engineering and information technology and the M.Eng. degree in electrical systems engineering from the HTWG Konstanz, University of Applied Sciences, Germany, in 2017 and 2019, respectively. He is currently pursuing the Ph.D. degree with the University of Ulm, Germany. In 2019, he joined the Institute of System Dynamics (ISD), HTWG Konstanz, University of Applied Sciences, working on coding theory and signal processing.

SERGO SHAVGULIDZE (Member, IEEE) received the Diploma degree in communication engineering from Georgian Technical University, Tbilisi, Georgia, in 1980, the Candidate of Techn. Science degree from the Institute for Control Problems, Moscow, Russia, in 1984, and the Doctor of Techn. Science degree from the Institute for Information Transmission Problems, Moscow, in 1991. Since 1980, he has been with Georgian Technical University, where he is currently a Professor with the Department of Telecommunications. Since 2004, he has also been with Georgian National Communications Commission. From 2007 to 2008, he was an Associate Member of the Information and Communications Security Panel under the NATO Science for Peace and Security Program. He headed his country's delegation to RA-2007, WRC-2007, RA-2015, WRC-2015, and RA-2019, and was elected as the Vice Chairperson of ITU-R Study Group 5 (Terrestrial Services) at RA-2015 and RA-2019. On leave from Georgian Technical University, he held different research positions with Linköping University and Lund University, Sweden; Darmstadt Technical University, Konstanz University of Applied Sciences, and Ulm University, Germany; Technical University of Denmark, Lyngby, Denmark; and Lancaster University and HW Communications Ltd., U.K. He has authored over 180 papers on these subjects. He has coauthored the book titled *Generalized Concatenated Constructions on a Base of Convolutional Codes* (Moscow, Russia: Nauka, 1991) with V. Zyablov. His research interests include coding theory and communication systems with a special emphasis on (generalized) concatenated codes, woven codes, coded continuous phase modulation, spatial modulation, and space-time coding.

JÜRGEN FREUDENBERGER (Member, IEEE) received the Dipl. (Ing.) and Ph.D. degrees in electrical engineering from the University of Ulm, Germany, in 1999 and 2004, respectively. In 2003, he joined the Daimler-Chrysler Research Center, Ulm, working on signal processing for automatic speech recognition. Since 2006, he has been a Professor with the HTWG Konstanz, University of Applied Sciences, Germany. He is currently on academic leave serving as the Head of the Key Technology Department, Cyberagentur, Halle (Saale), Germany. He has authored or coauthored over 150 papers and coauthored (with A. Neubauer and V.Kühn) the book *Coding Theory: Algorithms, Architectures and Applications* (John Wiley and Sons, 2007). His research interests include embedded system design, coding theory for communication and storage systems, and code-based cryptography.

He is a member of the German Association for Electrical, Electronic & Information Technologies (VDE/ITG) and received the ITG Award for his dissertation, in 2005.