

# Dynamic LLR scheme based on EM algorithm for LDPC decoding in NAND flash memory

Junli Peng<sup>1</sup>, Qi Wang<sup>2</sup>, Xiang Fu<sup>2</sup>, and Zongliang Huo<sup>3a)</sup>

<sup>1</sup> University of Chinese Academy of Sciences, Beijing, People's Republic of China

<sup>2</sup> Institute of Microelectronics of Chinese Academy of Sciences,  
Beijing, People's Republic of China

<sup>3</sup> School of Microelectronics, University of Chinese Academy of Sciences,  
Beijing, People's Republic of China

a) [huozongliang@ime.ac.cn](mailto:huozongliang@ime.ac.cn)

**Abstract:** A dynamic log-likelihood ratio (DLLR) scheme based on expectation-maximization (EM) algorithm for the decoding of low-density parity-check (LDPC) codes in NAND flash memory is proposed. When LDPC soft decoding fails, the DLLR scheme employs the EM algorithm to estimate the parameters of the threshold voltage distribution of NAND flash memory, and then recalculates the LLR values for decoding. Simulation results show that the proposed scheme can significantly improve the error correcting performance of LDPC soft decoding in NAND flash memory.

**Keywords:** NAND flash memory, threshold voltage distribution, EM algorithm, LDPC code, soft information

**Classification:** Integrated circuits

## References

- [1] Y. Cai, *et al.*: "Read disturb errors in mlc nand flash memory: Characterization, mitigation, and recovery," 2015 45th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (2015) 438 (DOI: [10.1109/DSN.2015.49](https://doi.org/10.1109/DSN.2015.49)).
- [2] W. Liu, *et al.*: "Low-power high-throughput bch error correction vlsi design for multi-level cell nand flash memories," 2006 IEEE Workshop on Signal Processing Systems Design and Implementation (2006) 303 (DOI: [10.1109/SIPS.2006.352599](https://doi.org/10.1109/SIPS.2006.352599)).
- [3] K. C. Ho, *et al.*: "A 520 k (18900, 17010) array dispersion ldpc decoder architectures for nand flash memory," IEEE Trans. Very Large Scale Integr. (VLSI) Syst. **24** (2016) 1293 (DOI: [10.1109/TVLSI.2015.2464092](https://doi.org/10.1109/TVLSI.2015.2464092)).
- [4] J. Kim and W. Sung: "Rate-0.96 ldpc decoding vlsi for soft-decision error correction of nand flash memory," IEEE Trans. Very Large Scale Integr. (VLSI) Syst. **22** (2014) 1004 (DOI: [10.1109/TVLSI.2013.2265314](https://doi.org/10.1109/TVLSI.2013.2265314)).
- [5] A. P. Dempster, *et al.*: "Maximum likelihood from incomplete data via the em algorithm," J. R. Stat. Soc., Ser. B (Methodological) **39** (1977) 1.
- [6] C. M. Bishop: *Pattern Recognition and Machine Learning* (Springer, New York, 2006) 1st ed. 437–439.

- [7] M. A. T. Figueiredo and A. K. Jain: “Unsupervised learning of finite mixture models,” *IEEE Trans. Pattern Anal. Mach. Intell.* **24** (2002) 381 (DOI: [10.1109/34.990138](https://doi.org/10.1109/34.990138)).
- [8] Y. Cai, *et al.*: “Threshold voltage distribution in mlc nand flash memory: Characterization, analysis, and modeling,” *2013 Design Automation & Test in Europe Conference & Exhibition (DATE)* (2013) 1285 (DOI: [10.7873/DATE.2013.266](https://doi.org/10.7873/DATE.2013.266)).
- [9] C. Kim, *et al.*: “A 21 nm high performance 64 gb mlc nand flash memory with 400 mb/s asynchronous toggle ddr interface,” *IEEE J. Solid-State Circuits* **47** (2012) 981 (DOI: [10.1109/JSSC.2012.2185341](https://doi.org/10.1109/JSSC.2012.2185341)).
- [10] S. Song, *et al.*: “A unified approach to the construction of binary and nonbinary quasi-cyclic ldpc codes based on finite fields,” *IEEE Trans. Commun.* **57** (2009) 84 (DOI: [10.1109/TCOMM.2009.0901.060129](https://doi.org/10.1109/TCOMM.2009.0901.060129)).
- [11] Z. Cui, *et al.*: “Reduced-complexity column-layered decoding and implementation for ldpc codes,” *IET Commun.* **5** (2011) 2177 (DOI: [10.1049/iet-com.2010.1002](https://doi.org/10.1049/iet-com.2010.1002)).

## 1 Introduction

In order to increase the NAND flash storage capacity and lower the cost per gigabyte, multi-level cell (MLC) and triple-level cell (TLC) technology have been employed by the industry and become mainstream. Due to various inevitable reliability issues caused by these technologies, the threshold voltage distribution of NAND flash memory may overlap seriously, leading to error occurrence in read operation [1]. NAND flash memory relies on advanced signal processing and error correcting code (ECC) to overcome the high bit error rate (BER) problem. However, the conventional Bose–Chaudhuri–Hocquenghem (BCH) code has been increasingly inadequate [2]. In order to ensure the integrity of data stored in NAND flash memory, more powerful ECC, such as the LDPC code with soft-decision decoding algorithms, is considered to be the promising candidates [3, 4].

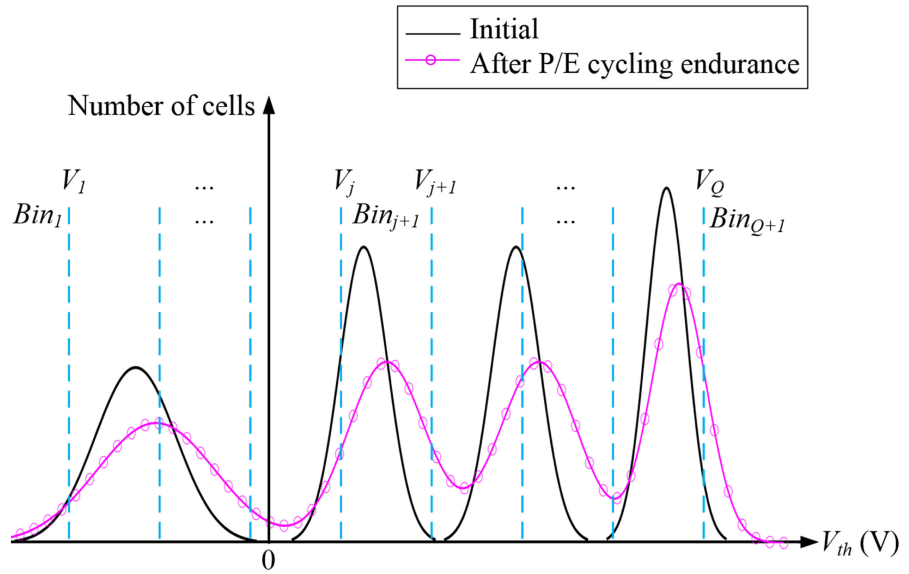
LDPC decoding algorithm demands LLR values as soft information in the decoding process. The accuracy of LLR values seriously affects the performance of LDPC decoding. However, obtaining LLR values needs to know the threshold voltage distribution of NAND flash memory, which is affected by various reliability mechanisms. If we use a series of fixed LLR values during the entire lifespan of NAND flash memory, the performance of LDPC decoding will be greatly reduced. An effective method to mitigate this effect is dynamically updating the LLR values when LDPC soft decoding fails.

In this letter, we propose a scheme for dynamically updating the LLR values based on the EM algorithm [5, 6]. We introduce the EM algorithm to find the parameters of the threshold voltage distribution of MLC NAND flash memory by performing the multiple reads operation with distinct read voltages, and the distribution is modeled as a Gaussian mixture model (GMM) [7] with four components. LLR values are calculated according to the estimated parameters. Simulation results show that the LDPC soft decoding algorithm with the DLLR scheme significantly outperforms the decoding algorithm with fixed LLR values.

## 2 EM algorithm for MLC NAND flash memory

The EM algorithm is an elegant and powerful method in maximum likelihood estimation problems with latent variables involving incomplete data. In order to apply the EM algorithm to MLC NAND Flash memory, read retry feature [8] should be supported by NAND flash memory chips. Consider the threshold voltage distribution of MLC NAND flash memory shown in Fig. 1. The threshold voltage distribution is affected by the program/erase (P/E) cycling endurance [8], and our task is to estimate the parameters of this distribution by the EM algorithm and then update the LLR values for LDPC soft decoding. We first conduct multiple reads operation with  $Q$  distinct read voltages, labeled as  $V_1$  to  $V_Q$ . Then these read voltages can divide the threshold voltage range into  $Q + 1$  bins, labeled as  $Bin_1$  to  $Bin_{Q+1}$ . Let  $N_{Bj}$  denote the number of cells within a NAND flash memory page whose threshold voltage falls into  $Bin_j$  (for  $j = 1, 2, \dots, Q + 1$ ), and assume that the size of a flash memory page is  $N$ , we have

$$N = \sum_{j=1}^{Q+1} N_{Bj}. \quad (1)$$



**Fig. 1.** Obtaining the data for the EM algorithm by conducting the multiple reads operation.

We can arbitrarily select a voltage value in each bin as the threshold voltage  $V_{th}$  of all cells belong to the bin. In this work, we choose the middle value of two adjacent read voltages as the threshold voltage of the corresponding bin. For  $Bin_1$  and  $Bin_{Q+1}$ , we choose  $V_1$  and  $V_Q$  as their threshold voltages, respectively. Let  $V_{Bj}$  denote the threshold voltage of  $Bin_j$ , then  $V_{Bj}$  can be expressed as:

$$V_{Bj} = \begin{cases} V_1, & j = 1 \\ \frac{V_{j-1} + V_j}{2}, & j = 2, 3, \dots, Q \\ V_Q, & j = Q + 1 \end{cases} \quad (2)$$

Through the above method, we can acquire  $Q + 1$  pairs of data that can be represented as  $(V_{Bj}, N_{Bj})$ . These data can be used to estimate the means and standard deviations of the GMM in NAND flash memory by the EM algorithm.

The EM algorithm is an iterative algorithm, each iteration includes two computational processes, called the E-step and the M-step [6]. Each memory cell has its own threshold voltage,  $N$  cells form a data set, denoted by  $S = \{v_1, \dots, v_i, \dots, v_N\}$ , where  $v_i$  is the threshold voltage of  $i$ th cell in a page. At the E-step, we use  $\beta_k(v_i)$  (for  $i = 1, 2, \dots, N$ ) to denote the posterior probability  $p(k|v_i)$  that data  $i$  belong to  $i$ th component of GMM, and assume that  $v_i \in \text{Bin}_j$ , we can obtain

$$\beta_k(v_i) = \frac{\pi_k N(v_i; \mu_k, \sigma_k)}{\sum_{m=1}^4 \pi_m N(v_i; \mu_m, \sigma_m)} = \frac{\pi_k N(V_{Bj}; \mu_k, \sigma_k)}{\sum_{m=1}^4 \pi_m N(V_{Bj}; \mu_m, \sigma_m)} = \beta_k(V_{Bj}), \quad (3)$$

where  $N(V_{Bj}; \mu_k, \sigma_k)$  represents the Gaussian function with mean  $\mu_k$  and standard deviation  $\sigma_k$ ,  $\pi_m$  is called mixing probability, and must satisfy

$$0 \leq \pi_m \leq 1, \text{ and } \sum_{m=1}^4 \pi_m = 1. \quad (4)$$

At the M-step, using (1) and  $(V_{Bj}, N_{Bj})$ , the estimated mean  $\tilde{\mu}_k$ , standard deviation  $\tilde{\sigma}_k$  and mixing probability  $\tilde{\pi}_k$  of the  $k$ th component of the mixture model can be re-estimated as:

$$\tilde{\mu}_k = \frac{\sum_{i=1}^N \beta_k(v_i) v_i}{\sum_{i=1}^N \beta_k(v_i)} = \frac{\sum_{j=1}^{Q+1} N_{Bj} \beta_k(V_{Bj}) V_{Bj}}{\sum_{j=1}^{Q+1} N_{Bj} \beta_k(V_{Bj})}, \quad (5)$$

$$\tilde{\sigma}_k^2 = \frac{\sum_{i=1}^N \beta_k(v_i) (v_i - \tilde{\mu}_k)^2}{\sum_{i=1}^N \beta_k(v_i)} = \frac{\sum_{j=1}^{Q+1} N_{Bj} \beta_k(V_{Bj}) (V_{Bj} - \tilde{\mu}_k)^2}{\sum_{j=1}^{Q+1} N_{Bj} \beta_k(V_{Bj})}, \quad (6)$$

$$\tilde{\pi}_k = \frac{1}{N} \sum_{i=1}^N \beta_k(v_i) = \frac{1}{N} \sum_{j=1}^{Q+1} N_{Bj} \beta_k(V_{Bj}). \quad (7)$$

The EM algorithm checks for convergence in each iteration by evaluating the log-likelihood function, the log-likelihood function can be represented as:

$$\begin{aligned} \log p(S | \tilde{\Theta}) &= \sum_{i=1}^N \log \sum_{m=1}^4 \tilde{\pi}_m N(v_i; \tilde{\mu}_m, \tilde{\sigma}_m) \\ &= \sum_{j=1}^{Q+1} N_{Bj} \cdot \log \sum_{m=1}^4 \tilde{\pi}_m N(V_{Bj}; \tilde{\mu}_m, \tilde{\sigma}_m), \end{aligned} \quad (8)$$

where  $\tilde{\Theta} = \{\tilde{\pi}_1, \dots, \tilde{\pi}_4, \tilde{\mu}_1, \dots, \tilde{\mu}_4, \tilde{\sigma}_1, \dots, \tilde{\sigma}_4\}$  is the estimated parameter set of the GMM model with four components. The E-step and the M-step of the EM algorithm are guaranteed to increase the log-likelihood function. When the variation of the log-likelihood function falls below a certain value, it can be considered

that the EM algorithm is convergent [6]. The proposed EM algorithm for MLC NAND flash memory is listed as Algorithm 1.

---

**Algorithm 1** Proposed EM Algorithm

---

**Input:**  $I_{max}$ : maximum iteration

$\Theta = \{\pi_1, \dots, \pi_4, \mu_1, \dots, \mu_4, \sigma_1, \dots, \sigma_4\}$ : initial parameter set

**Output:**  $\tilde{\Theta} = \{\tilde{\pi}_1, \dots, \tilde{\pi}_4, \tilde{\mu}_1, \dots, \tilde{\mu}_4, \tilde{\sigma}_1, \dots, \tilde{\sigma}_4\}$ : estimated parameter set

1: calculate the initial value of the log-likelihood function using  $\Theta$

2: **for**  $i = 1$  to  $I_{max}$  **do**

3: Evaluate  $\beta_k(V_{Bj})$  using the current parameter set

4: Re-estimate  $\tilde{\mu}_k$ ,  $\tilde{\sigma}_k^2$  and  $\tilde{\pi}_k$  using  $\beta_k(V_{Bj})$

5: Evaluate the log-likelihood function using  $\tilde{\Theta}$

6: **if** convergence criterion is satisfied **then**

7: get  $\tilde{\Theta}$  and go to the end

8: **else**

9:  $i = i + 1$ ,

9: go to line 3;

10: **end if**

11: **end for**

---

### 3 LLR calculation

---

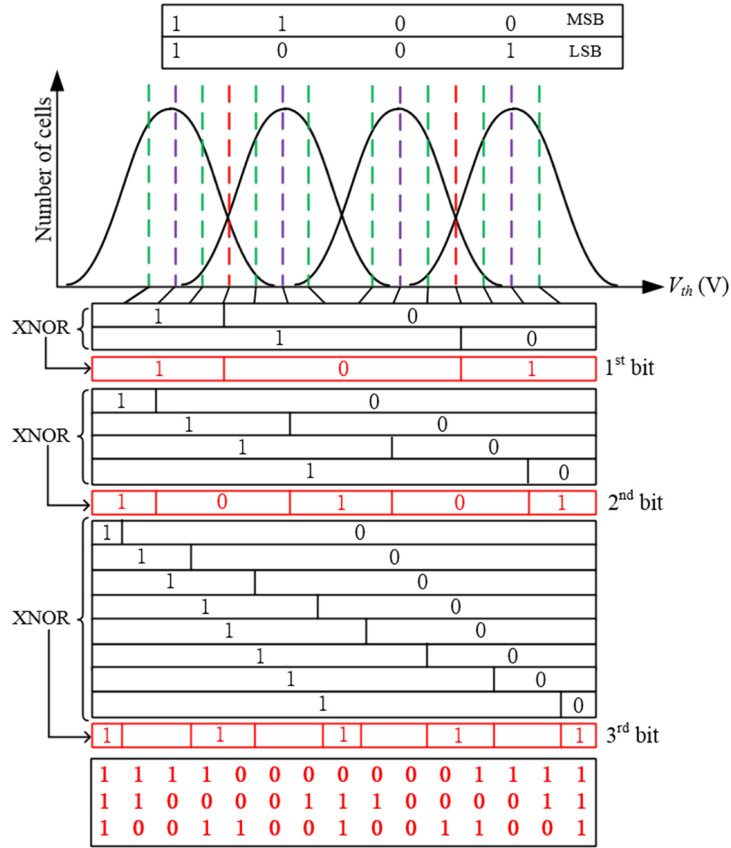
Before employing the LDPC decoding algorithm, we need to obtain the soft decision data [9] by performing the multiple reads operation and other logical operations. These data that imply probabilistic information will be converted into the corresponding LLR value for LDPC decoding using the estimated parameters. After updating the LLR value, these new values will be used for LDPC decoding until the next update.

#### 3.1 Obtaining soft decision data

In order to support LDPC decoding, NAND Flash need to generate the soft decision data that indicates the probability of ‘0’ or ‘1’. Fig. 2 shows an example of obtaining the soft decision data in MLC NAND Flash memory. By performing fourteen times read operations with distinct read voltages and XNOR operations, we can obtain 3-bit soft decision data for least significant bit (LSB) of MLC NAND Flash memory. First we perform the read operation with two read voltages (red dashed lines) to obtain two output, then using the XNOR operation, we can obtain the 1<sup>st</sup> bit of the soft decision data, which indicated the sign of the LLR value. Similarly, we can obtain the 2<sup>nd</sup> bit and 3<sup>rd</sup> bit of the soft decision data by more read voltages. Fourteen read voltages divide the threshold voltage range into fifteen intervals, and there are eight possible cases of the soft decision data output, each case implies a LLR value. For most significant bit (MSB), seven times read operations are needed to obtain the 3-bit soft decision data.

#### 3.2 Calculating the LLR value using the estimated parameters

After obtaining soft decision data output, the next step is translating the information



**Fig. 2.** Obtaining 3-bit soft decision data for LSB of MLC NAND Flash memory by performing fourteen times read operations and XNOR operations.

into LLR values for LDPC decoding. We can arbitrarily select a voltage value in each intervals, and then calculate the LLR value of the corresponding soft decision data using this voltage and the estimated parameters.

Let  $c$  denote the symbol taking on values in the set  $\{0, 1\}$ , representing the information stored in a NAND Flash memory cell. And we assume  $P(c = 0) = P(c = 1) = 0.5$ . Let  $y$  denote the threshold voltage of a cell with symbol  $c$ . According the definition, LLR of the cell can be written as

$$LLR(y) = \log \frac{P(y | c = 0)}{P(y | c = 1)}. \quad (9)$$

Considering the case of SLC NAND flash memory, there are two possible states in a cell, erase state and program state. The threshold voltage distribution of each state is modeled as Gaussian distribution. For the case of  $c = 1$ ,  $y \sim N(\mu_E, \sigma_E^2)$ , otherwise,  $y \sim N(\mu_P, \sigma_P^2)$ , where  $N$  denotes the Gaussian distribution,  $\mu_E$  and  $\sigma_E^2$  are the mean and variance of the erase state,  $\mu_P$  and  $\sigma_P^2$  are the mean and variance of the program state. We can rewritten (9) as

$$LLR(y) = \left( \frac{1}{2\sigma_E^2} - \frac{1}{2\sigma_P^2} \right) y^2 + \left( \frac{\mu_P}{\sigma_P^2} - \frac{\mu_E}{\sigma_E^2} \right) y + \left( \frac{\mu_E^2}{2\sigma_E^2} - \frac{\mu_P^2}{2\sigma_P^2} + \log \sigma_E + \log \sigma_P \right). \quad (10)$$

If  $\sigma_E = \sigma_P = \sigma$ , (10) can be simplified as

$$LLR(y) = \frac{\mu_P - \mu_E}{\sigma^2} y + \frac{\mu_E - \mu_P}{2\sigma^2}. \quad (11)$$

For MLC NAND flash memory, we can still compute the LLR value using (10) or (11), and the difference is that we need to use the estimated parameters of the corresponding states of MLC NAND flash memory.

#### 4 DLLR scheme based on the EM algorithm

Fig. 3 shows the entire DLLR scheme based on the EM algorithm for LDPC soft decoding in NAND flash memory. In the early stage of flash memory lifespan, the threshold voltage distribution is relatively ideal, and LDPC hard decoding algorithm is enough to guarantee low BER. Along with the increase of the number of P/E cycles, BER will also increase, we need to conduct LDPC soft decoding algorithm to mitigate this impact. However, when LDPC soft decoding algorithm fails, the DLLR scheme based on the EM algorithm begins to work, and restarts LDPC soft decoding algorithm. One way to determine the failure of decoding is to set a threshold for BER, when the BER is higher than the threshold, the decoding is deemed to have failed. In this work, we think that the decoding fails as long as the output of the decoding is different from the transmitted data.

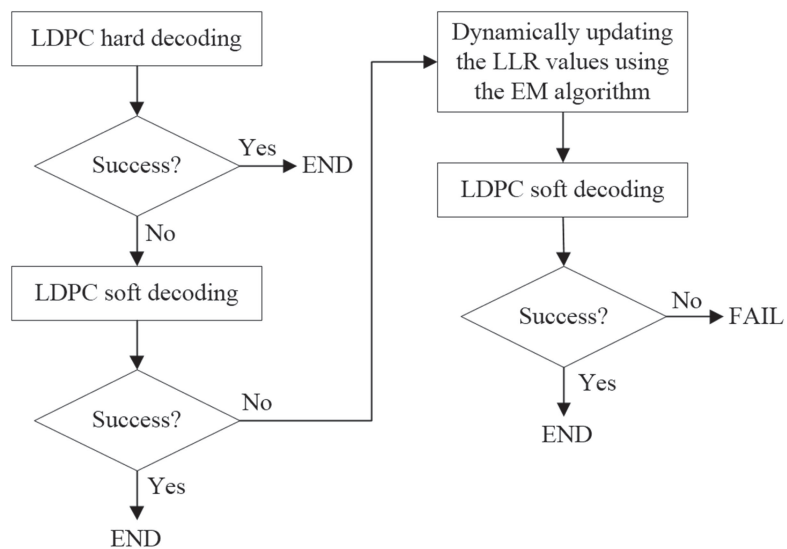


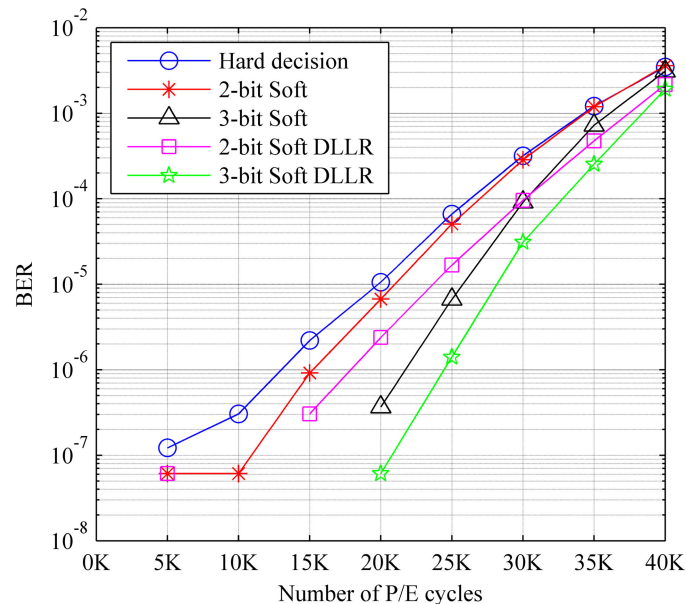
Fig. 3. The entire DLLR scheme.

#### 5 Simulation results

We construct an (8704, 8192) 1 KB quasi-cyclic (QC) LDPC code with code rate of 0.94 by circulant permutation matrix (CPM) [10]. According to the P/E cycling model proposed in [8], the data samples are generated by simulated MLC NAND flash memory model. We adopt layered offset min-sum algorithm [11] for simulation, and the number of decoding iterations is set to 5. Fig. 4 shows the BER performance of the LDPC code with and without the proposed DLLR scheme, and Q is set to 23. The ‘2-bit soft’ and ‘3-bit soft’ case adopt the LDPC soft decoding algorithm with 2-bit and 3-bit soft decision data. We observe that increasing the number of quantization levels of the LLR values can significantly improve the error



correcting performance. When the number of P/E cycles exceeds 30 K, the 2-bit soft decoding algorithm with DLLR scheme can gain lower BER than the 3-bit case with fixed LLR values. What's more, the LDPC soft decoding algorithm with 3-bit DLLR scheme achieves the best error correcting performance and the BER is  $6 \times 10^{-8}$  when the number of P/E cycles is 20 K, while that of 3-bit fixed LLR case is  $3.7 \times 10^{-7}$ .



**Fig. 4.** The BER of MLC NAND flash memory varies with the number of P/E cycles.

## 6 Conclusion

In this letter, we propose a DLLR scheme based on the EM algorithm to improve the error correcting performance of LDPC soft decoding algorithm in NAND flash memory. Our scheme does not need to know the initial threshold voltage distribution of flash memory in advance, and the improved EM algorithm does not require iterative computation. Simulation results have demonstrated that our DLLR scheme can accurately update the LLR values for LDPC soft decoding. Therefore, the proposed scheme is a good solution to reduce the BER of NAND flash memory.

## Acknowledgments

This work was supported by National Natural Science Foundation of China (No. 61474137).