2015 Fifth International Conference on Communication Systems and Network Technologies

Reliability-based ECC System for Adaptive Protection of NAND Flash Memories

Liu Yuan, Huaida Liu, Pingui Jia, Yiping Yang
deft. Institute of Automation, Chinese Academy Sciences, Beijing, China.
Email: {liu.yuan, huaida.liu, pingui.jia, yiping.yang}@ia.ac.cn

***Abstract*—In order to make the implementation of the error correcting code(ECC) efficiently, it should be tailored individually to each block, and can be changed over time as the running condition changes. The combined effect of reliability factors makes it difficult to select ECC by considering the effect factors separately. So more appropriate ECC selection method need to be found. Meanwhile, most of the ECCs are designed to adapt to specific hardware, the implementation of the ECC needs to be redesigned when the hardware platform changes. So the introduction of commonly used ECC system can provide a great convenience for error correction. In this paper, we propose an easy-to-use and flexible ECC system which takes advantage of the reliability map to provide adaptive protection. Proposed ECC system comprises ECC selection module and ECC codec module. A method to select appropriate ECC based on reliability map which provides the evaluation of bit error rate is proposed in ECC selection module. ECC encoder and decoder suit are called in ECC codec module to provide various protection. Proposed system can be used in many platforms, such as DSP, ARM, etc. Compared with the early works, the uncorrectable bit error rate(UBER) performance, coding time and redundancy are all optimized by proposed ECC system.**

***Keywords*-Error correcting code(ECC), flash memories, error patterns, reliability, fault tolerant.**

I. Introduction

NAND Flash memories have already been widely used due to their large storage capacity, non-volatile and low power [1]. However, the flash memories are more prone to errors than SDRAM and SRAM, because of the larger storage density [2]. Program/Erase(P/E) cycle count and retention time are the dominant factors that have important influence on bit upset errors [3]. Efforts should be done to reduce the effect of errors. Error correcting code(ECC) is one of the most popular methods to protect the flash memories [4]. The application of ECC can greatly increase the lifetime of memories. Hamming code [5], Reed-Solomon(RS) [6], Bose-Chaudhuri-Hocquenghem (BCH) [7], [8], Low Density Parity Check(LDPC) [9] have entered the mainstream of the fault tolerant designs to provide various levels of protection. However, most of the researches only focus on the the VLSI implementation of the ECC code itself [10], [11]. If the hardware platform changes, the ECC implementation will also need to be changed. The commonly used ECC system which can adapt to most of the hardware platforms is nearly ignored. The finding of easy-to-use, more efficient, convenient and portable ECC system could provide a great convenience for error correction.

We have also found that the decoding time of Hamming was about 10 times less than that of 2-bit BCH(4122,4096,5), and 20 times less than that of 4-bit BCH(4148,4096,9). Meanwhile during the lifetime of NAND Flash, the reliability of each block is changing when the effect factors such as P/E cycle count and retention time change [3], we need to design ECCs with various error correction capability. However, the error correction capability of the regular ECC methods will not change when the reliability changes, more resources will be waste when correcting the same number of errors [4]. The adaptive protection by ECC can greatly improve the coding efficiency, especially under the situation of low bit error rate(BER).

Several efforts have been done to provide unequal protection [11], [12]. But they only concentrate on one kind of ECC code, the efficiency of ECC combination is ignored. Meanwhile, most of the efforts are focus on the implementation of the reconfigurable ECC, but rarely talk about the method to select appropriate ECC. The early work done by us in [4] gives a threshold method to select ECC, but the ECC selection is not so accuracy. Because the ECC selection is applied when the data is stored into the memories. The P/E cycle count can be determined, but we can't make sure the storage time of the data. So there is no way to avoid some incorrect selections because of that the two effect factors are not completely independent. Paper [9] provides a look-up table based method to obtain the combined effect. However, the method can only be applied to the MLC memories. To solve this problem, we propose a new ECC selection method which takes advantage of reliability map to obtain the combined effect of these factors. Through the evaluation of BER for each block, a more suitable ECC selection can be obtained.

Based on the above discussion, we propose a reliability map based ECC system to improve coding efficiency. Proposed ECC system is composed of ECC selection module and ECC codec module. A method to assign appropriate error correction capability to each block based on the combined influence of the effect factors is introduced in ECC selection module. The environmental sensor obtains the effect factors, then the reliability map provides a method

978-1-4799-1797-6/15 $31.00 © 2015 IEEE
DOI 10.1109/CSNT.2015.23

TABLE I
IMPLEMENTATION OF VARIOUS ECC BY ADSP-TS201.

ECC	encoder(μs)	decoder(μs)
Hamming	0.992	1.167
BCH(4122,4096,5)	7.233	10.01
BCH(4148,4096,9)	9.378	21.345

to evaluate the BER for each block based on the effect factors. Then the more accurate ECC selection can be obtained by considering the BER. In order to show the better performance of ECC selection, a more accurate reconfigurable ECC codec based on the scheme in [4] are applied to correct up to 4-bit errors in ECC codec module. When ECC selection module determines ECC for a given block, corresponding ECC encoder and decoder suit are invoked to protect the data. Proposed ECC system can provide rapid and alterable protection, and can be easily transplanted into other hardware platforms.

II. Background and motivation

In this section, a brief introduction of reliability factors is provided first. Errors of NAND Flash can be categorized into hard errors and soft errors [10]. Hard errors are usually caused by oxide breakdown, which can't be corrected by ECC, and the soft errors such as (Single Bit Upset)SEU and MBU which are the main types of errors discussed in this paper can be corrected by ECC [2]. Recently, the higher storage density makes the memories more vulnerable to soft errors. Several mechanisms such as program disturb, read disturb and retention can induce SEU or MBU errors [3]. During the lifetime of NAND Flash memories, read disturb errors make less contribution to total BER, program disturb induced errors and retention induced errors are the major error types [4]. If the retention time is longer than 1 day, retention errors are dominant, otherwise program-disturb(PI) errors are dominant [1]. Researchers have found that the bit error rate was closely related to P/E cycle count and retention time [3].

Firstly, all types of errors especially PI errors are increased with the growth of P/E cycle count, because it influences the range of threshold voltage [2]. Based on the results in [3], the effect factors can be obtained as P/E cycle count and retention time. At the beginning of program, the BER is still very low. Then with the growth of P/E cycle count, BER is slowly increased when retention time is less than 1 day. But the increasing trend is obviously when retention time is larger than 3 day [3]. Secondly, with the passage of time, retention errors show a significant effect on BER. When P/E cycle count is higher, the effect is more obviously. When retention time is increased from 1 day to 1 month, the BER is increased by about 1 order of magnitude with P/E cycle count equal to $10K$ [3]. Finally, the effect factors are not independent. They will altogether affect the reliability

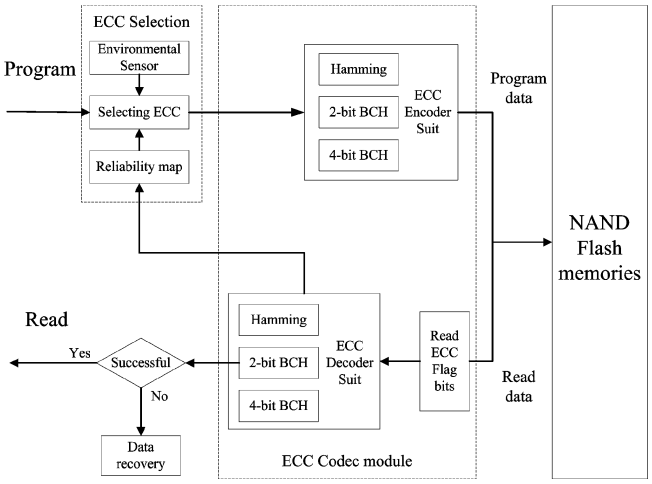

Fig. 1. Structure of Proposed ECC System for Adaptive Protection of Flash Memories.

[4]. The combined effects can provide an evaluation of BER for each memory block. So if we record the BER and the corresponding P/E cycle count and retention time in the reliability map, the BER during the lifetime can be obtained by the real running conditions obtained by the environmental sensor. When we obtain these effect factors, we can select appropriate error correction capability by the evaluated BER for the block at a given time.

Errors in NAND Flash memories are mostly random [10], and can be corrected by ECCs. BCH and Hamming are more suitable to the protection of flash memories because they can correct random errors fast [4]. Table 1 shows the complexity of various ECC. The encoding time of BCH(4148, 4096, 9) is $9.378\mu s$, which is 9.45 times longer than that of Hamming. The decoding time of BCH(4148, 4096, 9) is $21.35\mu s$, which is 2.13 times longer than that of BCH(4122, 4096, 5) and 18.3 times longer than that of Hamming. So the larger error correction capability will result in a large coding time.

Through above analysis, we can draw a conclusion that the selection of appropriate ECC based on reliability has a positive influence on the coding efficiency. Through adaptive ECC protection, we can avoid the waste of error correction capability, and obtain a much lower coding latency.

III. Proposed Reliability-based ECC system

Based on the reliability factors, proposed ECC system provides adaptive protection for flash memories. In this paper, we mainly introduce the structure of ECC system and how to use proposed system to correct errors. A new ECC selection method is also introduced. Fig.1 shows the structure of it. Proposed ECC system has two modules: one is ECC selection module which has environmental sensor and a built-in reliability map, the other one is ECC codec module which comprises ECC encoder suit and ECC

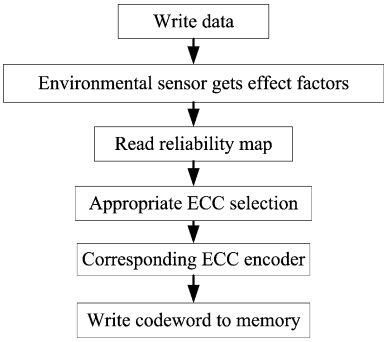

Fig. 2. Programming Flow of Proposed ECC System.

TABLE II
RELIABILITY MAP STORES BER WITH VARIOUS P/E CYCLES N AND RETENTION TIME T.

	N_0	N_1	$\cdots$	N_{n-1}	N_n
T_0	$BER_{0,0}$	$BER_{0,1}$	$\cdots$	$BER_{0,n-1}$	$BER_{0,n}$
T_1	$BER_{1,0}$	$BER_{1,1}$	$\cdots$	$BER_{1,n-1}$	$BER_{1,n}$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
T_{m-1}	$BER_{m-1,0}$	$BER_{m-1,1}$	$\cdots$	$BER_{m-1,n-1}$	$BER_{m-1,n}$
T_m	$BER_{m,0}$	$BER_{m,1}$	$\cdots$	$BER_{m,n-1}$	$BER_{m,n}$

decoder suit. The functions of these modules are as follows:

When there are data to be programmed, ECC selection module provides a method to select suitable ECC for each block. Environmental sensor senses the practical running environment and provides continuously monitoring of P/E cycle count and retention time. Reliability map is prerecorded to evaluate BER by the effect factors. ECC selection module takes advantage of the effect factors and reliability map to determine which level of ECC should be chosen.

ECC codec module comprises ECC encoder suit and ECC decoder suit to provide ECC protection. After receiving the ECC selection result, the reconfigurable ECC protection is presented by the ECC encoder suit and ECC decoder suit.

Next we show how proposed ECC system detects and corrects errors. The programming flow and reading flow are introduced separately.

A. Programming Flow

Fig.2 shows the major steps in programming flow. During programming, the following steps should be done: Firstly, ECC selection module is invoked. In step 1, environmental sensor obtains P/E cycle count and retention time. Environmental sensor is composed of a timer and a lookup table to record P/E cycles and retention time. The P/E cycle count of a given block is recorded in the lookup table. When the block is programmed, the P/E cycle count is increased by 1. The timer is used to record the retention time of the last programming. We have two methods to solve the problem that the storage time of this programming is not sure. One is the periodic refresh, when programming the data, we firstly make sure the period of refreshing, so the retention time can be obtained in advance. The other one assumes that the retention time is equal to the last programming, because the refresh rate between two programming is usually similar. So both of the P/E cycle count and retention time can be obtained.

In step 2, reliability map is read from RAM to obtain the evaluation of BER. Table 2 shows an example of $(m+1)\times(n+1)$ reliability map which stores BER with various P/E cycle count and retention time. The P/E cycle count up to 10K is divided into $n+1$ levels and the retention time up to 1 month is divided into $m+1$ levels based on the relationship with BER. The test results in [2], [3] can provide the BER data with various effect factors. So the reliability map can be generated by the test results that can be obtained. BER is stored with the corresponding P/E cycle count N and retention time T. Next, we show how to evaluate the BER based on the limited BER data in the reliability map. After environmental sensor senses the running environment, we can obtain P/E cycle count as $N_j \le N \le N_{j+1}$, retention time as $T_i \le T \le T_{i+1}$. Then the evaluated BER can be obtained by the bilinear interpolation method:

$$\begin{aligned} BER =& BER_{i,j} \times \frac{(T-T_i)(N-N_j)}{(T_{i+1}-T_i)(N_{j+1}-N_j)} + \\ & BER_{i,j+1} \times \frac{(T-T_i)(N_{j+1}-N)}{(T_{i+1}-T_i)(N_{j+1}-N_j)} + \\ & BER_{i+1,j} \times \frac{(T_{i+1}-T)(N-N_j)}{(T_{i+1}-T_i)(N_{j+1}-N_j)} + \\ & BER_{i+1,j+1} \times \frac{(T_{i+1}-T)(N_{j+1}-N)}{(T_{i+1}-T_i)(N_{j+1}-N_j)} \end{aligned} \quad (1)$$

In step 3, ECC selection module takes advantage of the evaluated BER to determine which ECC should be selected. Fig.3 shows the uncorrectable bit error rate(UBER) as a function of raw BER for different error correction capability. 10^{-15} can be regarded as safe BER for flash memories, there is no need for ECC in the case $BER < 10^{-15}$. When $10^{-15} < BER < 7\times10^{-10}$, 1-bit ECC Hamming is suggested. When $7\times10^{-10} < BER < 7\times10^{-8}$, 2-bit BCH is suggested. When BER is between 7×10^{-8} and 3×10^{-6}, 4-bit BCH should be chosen. Through the relationship between UBER and BER, appropriate ECC can be selected.

After ECC selection, ECC encoder suit in ECC codec module calls corresponding level of ECC to encode the data. ECC encoder suit is composed of Hamming, BCH(4122,4096,5) and BCH(4148,4096,9) encoders. Compared with the early work in [4], the encoders in proposed ECC system use different methods to mark the ECC. We don't use the number of error check bits here, the TMR based ECC flag bits are used to show which ECC is selected. Because the TMR method can correct the errors in the check bits. The codewords generated by these encoders are shown in Fig.4. Hamming codeword which

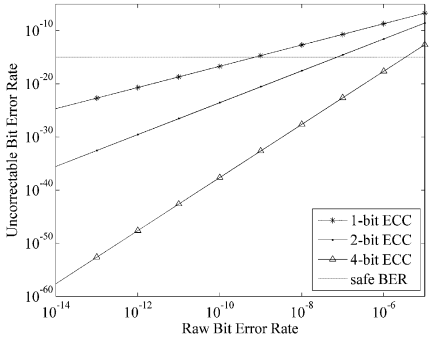

Fig. 3. UBER versus Raw BER for Different Error Correction Capability.

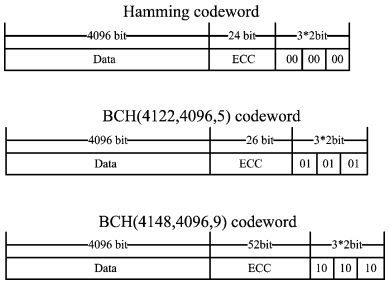

Fig. 4. Structure of Proposed ECC Codewords.

has 512Byte data bits, 3Byte error check bits, and TMR based 2bit ECC flag bits '00' is generated by Hamming encoder. The generated 2-bit BCH codeword is composed of 4096 data bits, 26 error check bits and 3×2 ECC flag bits '010101'. The 4-bit BCH codeword has 52 error check bits and TMR based ECC flag bits '101010'.

Finally, the generated codewords with TMR stored ECC flag bits are written into flash memories.

B. Reading Flow

Fig. 5 gives the reading flow of flash memories in proposed system. During reading, ECC decoder suit which comprises BCH(4148,4096,9) decoder, BCH(4122,4096,5) decoder and Hamming decoder is applied to detect and correct errors. The following steps should be done.

Firstly, TMR based ECC flag bits are voted to determine which level of ECC is selected. In step 2, the selected ECC decoder in ECC decoder suit is invoked to correct errors as in [4]. After the decoding, if the number of errors is not larger than the error correction capability, we can obtain the data after correction. Otherwise we should replace the dirty data with backup data and increase the error correction capability. Next, the reliability map also need to be revised by the error-correction results. BER stored in reliability map will be replaced by the real tested BER. The new BER can be obtained by the operation that divide the number of errors by the number of test data. Then we call environmental sensor to obtain the P/E cycle count and retention time at the given

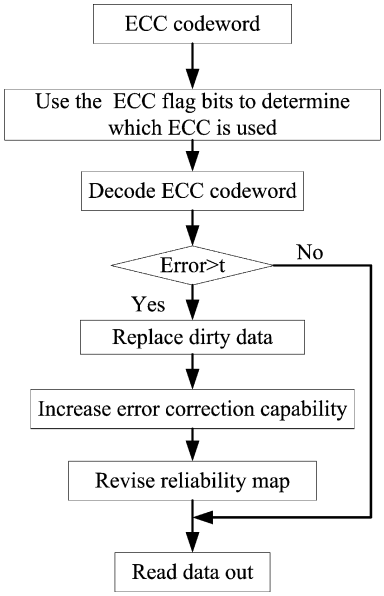

Fig. 5. Reading Flow of Proposed ECC System.

time. The new BER will be stored in the reliability map with the nearest P/E cycle count and retention time. Finally, the corrected data can be read out from the memories.

IV. Implementation Results

In this section, we show the flexible performance of proposed ECC system. A type of NAND Flash is used to evaluate the performance of proposed ECC system. The type of memory has 4096 blocks, a block has 128 pages and one page has 8 sectors. Sector is the unit of ECC protection, which has 4096 data bits and 128 OOB bits. The error check bits are all stored in OOB area. The ADSP-TS201 system is used as one of the hardware platforms to implement the ECC system. Visual DSP++ 5.0 compiler is used to simulate the results.

A. Error Models

As shown in section 2, we found that the running conditions had experienced several stages. So four typical error models are assumed in this paper to be used in fault injection. A sample of 128 blocks are used to simulate the results. Fig.6 shows the distribution percentage of various multiple bit upset (MBU) errors in these blocks. Four error models are assumed by considering the MBU possibility results in [1] as follows:

Model 1: This model can be regarded as the initial phrase of flash memories. Almost all the blocks in the memories have less P/E cycle count and retention time. We assume that in this case the levels of error rate are all level 1 in the codeword and are all independent in the memories. The average BER of these blocks is less than 2.5×10^{-10}.

Model 2: In this case, the memory blocks have already

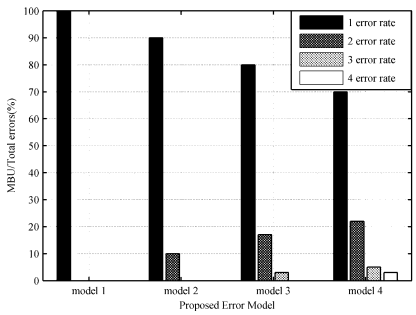

Fig. 6. The Distribution Percentage of Various MBU Errors.

been running for some time. The P/E cycles and the retention time of several blocks are increased. Errors are assumed to have two types: 90% blocks have the error rate of level 1 and 10% blocks have level 2. The average BER of this model is around 3.5×10^{-9}.

Model 3: The P/E cycle count and retention time of memories continue to increase. More blocks have larger P/E cycle count and retention time. The average BER of this model is 1.7×10^{-8}. In this case there are three types of blocks with 80% level 1, 17% level 2 and 3% level 3.

Model 4: Errors in this model have four types: 70% blocks have the error rate of level 1, 22% blocks have level 2, 5% blocks have level 3 and 3% blocks have level 4. In this case, the memories have already worked for a longer time under the situation of higher P/E cycles. The average BER for simulation is 1.0×10^{-7}.

B. Simulation Results

The four typical error models mentioned above will be injected into memories to show the improvement of proposed ECC system. For simplify, we assume that the BER of the memory block is equal to the bit error rate of the codeword and the number of errors in the codeword has reached the maximum bit of the given level. The following performance metrics are widely used to evaluate the performance of proposed ECC system.

1. Average uncorrectable bit error rate(UBER): The average number of bits that can't be corrected by ECC divided by the number of received data bits. If the error correction capability of ECC is t-bit errors per codeword, an uncorrectable error will happen if t+1 or more errors occur [1], [2].

2. Average running time: The average running time to encode or decode a ECC codeword which is obtained by considering the average performance of the memory block.

3. Redundancy bits: The rate of redundancy bits is equal to the number of error check bits divided by the length of ECC codeword [1].

These three performance metrics are used to compare the performance of proposed ECC system with uniform BCH(4148,4096,9) under the typical running conditions as shown in Fig.6. For simplicity, we assume that all the errors are detected and corrected. In the ideal situation, we assume that all the ECC selection is appropriate. The implementation results of proposed ECC system are shown in Table 3 and the results of regular 4-bit BCH are shown in Table 4. The UBER performance is optimized that the UBER can be always around 1×10^{-15} in the design of error models. At the initial phase of lifetime, both the coding time and redundancy are decreased rapidly. The coding time of proposed ECC system can reach about $1\mu s$ which is over 10 times less than that of regular BCH, and the redundancy rate is reduced from 1.25% to 0.72%. At the middle stage of flash memories as model 2,3,4, with the growth of BER, the coding time is a little increased, but the speedup can still be over 3.1x. Even the stage as model 4, the encoding time of proposed ECC system can be decreased to $3\mu s$ and the decoding time is reduced to $4.7\mu s$. The data throughput can reach $871.5Mbps$. Compared with the early work in [4], the reading throughput can be increased by 1.7 times. The redundancy rate can reach 0.8%, without more storage cost, the error check bits can be stored in OOB area.

TABLE III
IMPLEMENTATION RESULTS OF PROPOSED ECC SYSTEM.

Proposed	Encode time(μs)	Decode time(μs)	Redundancy(%)	$UBER$ (bit^{-1})
model 1	1.012	1.173	0.72%	1.20×10^{-16}
model 2	1.598	2.002	0.75%	1.27×10^{-16}
model 3	2.348	3.334	0.78%	1.26×10^{-16}
model 4	3.034	4.699	0.80%	1.26×10^{-16}

Proposed ECC system can reduce the encoding and decoding time obviously, as well as the storage redundancy. The reduction trend of coding time and redundancy will be decreased with the increase of running time. Because of that when the average BER is high, most of the error correction capability has reached to the maximum, the ECC selection results have less influence on the coding efficiency. So proposed ECC system is particulary suitable to the situation of initial and middle phase of the lifetime. When the running of the given flash memories reaches to a specific level, the ECC selection module in the ECC system will be closed, then the ECC protection for all the blocks reaches the maximum level.

V. Conclusion

In this paper, a reliability-based ECC system which comprises ECC selection module and ECC codec module is proposed to provide adaptive protection of flash memories. ECC selection module takes advantage of reliability map to determine which ECC should be chosen. By assigning suitable ECC based on the evaluated BER, proposed ECC system shows an obvious improvement in coding time and

TABLE IV
IMPLEMENTATION RESULTS OF REGULAR 4-BIT BCH.

4-bit BCH	Encode time(μs)	Decode time(μs)	Redundancy(%)	$UBER$ (bit^{-1})
model 1	9.373	14.275	1.25%	2.28×10^{-36}
model 2	9.373	14.389	1.25%	1.15×10^{-26}
model 3	9.373	14.716	1.25%	3.84×10^{-22}
model 4	9.373	15.110	1.25%	3.74×10^{-18}

storage redundancy while keeping the UBER around 10^{-15}. The speedup of coding time can be 3.1x. The redundancy rate can be about 36% lower than that of regular BCH. No limited to the DSP platforms, proposed ECC system can be used in various hardware platforms. Under the situation that the platform has a 600MHz CPU, such as the ADSP-TS201, the data throughput can be increased to about $871.5Mbps$, which is 1.7x larger than the early work. Real-time demands can be satisfied through proposed ECC system. Future investigation is the real application of proposed ECC system in the protection of NAND Flash memories to show the efficiency in the application.

REFERENCES

[1] C. Yang, Y. Emre, and C. Chakrabarti, "Product code schemes for error correction in MLC NAND flash memories", IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 20, no. 12, pp. 2302-2314, Dec. 2012.

[2] N. Mielke, T. Marquart, N. Wu, J. Kessenich, H. Belgal, E. Schares, F. Trivedi, E. Goodness, and L. R. Nevill, "Bit error rate in NAND Flash memories", Proc. IEEE Int. Rel. Phys. Symp., 2008, pp. 9-19.

[3] Y. Cai, E. F. Haratsch, O. Mutlu, and K. Mai, "Error patterns in MLC NAND flash memory: Measurement, characterization, and analysis", Proc. of the Design, Automation & Test in Europe Conference & Exhibition (DATE), pp. 521-526, Aug. 2012.

[4] L. Yuan, H. Liu, P. Jia and Y. Yang, An adaptive ECC scheme for dynamic protection of nand flash memories. accepted by Proc. ICASSP, 2015.

[5] Micron Corp. (2011.). "Error Correction Code (ECC) in SLC NAND Introduction", [Online]. Available: http://www.micron.com/~/media/Documents/Products/Technical%20Note/NAND%20Flash/tn2963_ecc_in_slc_nand.pdf.

[6] B. Chen, X. Zhang, and Z. Wang, "Error correction for multilevel NAND flash memory using Reed-Solomon codes", Proc. 2008 IEEE Workshop on Signal Process. Syst. (SiPS), 2008, pp. 94-99.

[7] W. Liu, J. Rho, and W. Sung, "Low-Power High-Throughput BCH Error Correction VLSI Design for Multi-level Cell NAND Flash Memories", Proc. IEEE Workshop Signal Processing Systems (SiPS), 2006, pp. 248-253.

[8] Micron Corp. (2012.). "Enabling Software BCH ECC on a Linux Platform Introduction", [Online]. Available: http://www.micron.com/~/media/Documents/Products/Technical%20Note/NAND%20Flash/tn2971_software_bch_ecc_on_linux.pdf.

[9] S. Tanakamaru, Y. Yanagihara, and K. Takeuchi, "Error-Prediction LDPC and Error-Recovery Schemes for Highly Reliable Solid-State Drives(SSDs)", IEEE J. Solid-State Circuits, vol. 48, no. 11, pp. 2920-2933, Nov. 2013.

[10] C. Yang, Y. Emre, and C. Chakrabarti, "Product code schemes for error correction in MLC NAND flash memories", IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 20, no. 12, pp. 2302-2314, Dec. 2012.

[11] S. Paul, F. Cai, X. Zhang, and S. Bhunia, "Reliability-driven ECC allocation for multiple bit error resilience in processor cache", IEEE Trans. Comput., vol. 60, no. 1, pp. 20-34, Jan. 2011.

[12] Y. Hu, N. Xiao, and X. Liu, "An elastic error correction code technique for NAND flash-based consumer electronic devices", IEEE Trans. Consumer Electronics, vol. 9, no. 1, pp. 53-58, Apr. 2013.