



GLOBAL JOURNAL OF ADVANCED RESEARCH
(Scholarly Peer Review Publishing System)

A NOVEL ERROR CORRECTING TECHNIQUE IN MLC NAND FLASH MEMORIES FOR DIGITAL APPLICATIONS

J.RAVIBABU

PG scholar, Dept of ECE,
Sreenidhi Institute of Science and Technology,
India.

S.N. CHANDRASEKHAR

Assistant Professor,
Dept of ECE,
Sreenidhi Institute of Science and Technology,
India.

Abstract

In this paper product code based schemes are proposed to support higher error correction capability. Conventional reliable MLC NAND flash memories based on BCH codes or Reed-Solomon (RS) codes have a large number of undetectable and mis corrected errors. Moreover, standard decoders for BCH and RS codes cannot be easily modified to correct errors beyond their error correcting capability. Proposed product codes use Reed-Solomon (RS) codes along rows and Hamming codes along columns and have reduced hardware overhead. The proposed product codes can correct errors in multi level cell (MLC) NAND Flash memories with lower bit error rate (BER). MLC NAND flash memories can reduce the number of errors undetected for all code words to be almost 0 at the cost of less than 20% increase in power and area compared to architectures based on BCH codes and RS codes.

Keyword - BCH codes, Reed-Solomon, MLC NAND flash memories.

1. INTRODUCTION

Flash memory has become the dominant technology for non-volatile memories. It is used in memory cards, USB flash drives, and solid-state drives in application platforms such as personal digital assistants, laptop computers, digital audio players, digital cameras and mobile phones. We focus on NAND Flash memories since they have lower erase times, less chip area per cell which allows greater storage density, and lower cost per bit than NOR Flash memories. Specifically, we focus on multi-level cell (MLC) Flash memories which store two or more bits per cell by supporting four or more voltage states. These have even greater storage density and are the dominant Flash memory technology.



There are some inherent limitations of NAND Flash memories. These include write read disturbs, data retention errors, bad block accumulation, limited number of writes, and stress-induced leakage current. In recent years, due to cell size scaling, these issues have become critical. In particular, reliability of MLC memory significantly degrades due to reduced gap between adjacent threshold levels.

When digital data is stored in a memory, it is crucial to have a mechanism that can detect and correct a certain number of errors. ECC can detect and correct the most common kinds of internal data corruption. Error control coding (ECC) is essential for correcting soft errors in Flash memories. Error correction code (ECC) techniques, which can detect and correct errors by storing and processing extra parity bits, have now become an integral part of Flash memory design.

NAND Flash memory has become the dominant technology for non-volatile memories. NAND Flash memories have lower erase times, less chip area per cell which allows greater storage density, and lower cost per bit than NOR Flash memories. Multi-level cell (MLC) Flash memories can store two or more bits per cell by supporting four or more voltage states.

In this paper product code based schemes are proposed to support higher error correction capability. Conventional reliable MLC NAND flash memories based on BCH codes or Reed-Solomon (RS) codes have a large number of undetectable and mis corrected errors. Moreover, standard decoders for BCH and RS codes cannot be easily modified to correct errors beyond their error correcting capability. Proposed product codes use Reed-Solomon (RS) codes along rows and Hamming codes along columns and have reduced hardware overhead. The proposed product codes can correct errors in multi level cell (MLC) NAND Flash memories with lower bit error rate (BER). MLC NAND flash memories can reduce the number of errors undetected for all code words to be almost 0 at the cost of less than 20% increase in power and area compared to architectures based on BCH codes and RS codes.

While these Flash management techniques increase the life time of Flash memories, they are not good at correcting soft errors. Error correction code (ECC) techniques, which can detect and correct errors by storing and processing extra parity bits, have now become an integral part of Flash memory design. Single error detection/correction codes, such as Hamming codes, used to be sufficient to enhance the reliability of single-level cell (SLC) Flash memory systems. In recent years, long linear block codes with high error correction capability are used because the single error correction capability of Hamming code is no longer sufficient. The Bose-Chaudhuri-Hocquenghem (BCH) code and its subclass Reed-Solomon (RS) code are the best-known linear block codes for memories. Pipelined or bit-parallel BCH code has been used in . Schemes based on concatenation of BCH codes and trellis coding modulation (TCM) have recently been proposed. While they reduce the error correction burden of a single BCH code, they require five (instead of four) threshold states per cell. ECC based on RS codes have been used in several commercial MLC Flash memories. They use plain RS codes and can correct up to 24 errors in 512B, at the cost of larger hardware and coding latency.

2. LITERATURE SURVEY

A. FLASH MEMORY

NAND Flash memories were introduced by Toshiba in 1989. These memories are accessed much like block memory devices such as hard disks or memory cards. A NAND Flash memory bank consists of several blocks, where each block consists of a number of pages. The organization of a NAND Flash memory is shown in Fig. 1. Typical page size for a NAND Flash memory is around 2 to 16 kB (for multiple bit storage devices). For ex-ample, in an 8 kB per page Flash memory, each memory bank consists of 1024 blocks, and each block consists of 64 pages, each of size 8 kB. We assume that each page includes both in-formation bits and parity bits of ECC. Almost all NAND Flash memories rely on ECC to detect and correct errors caused by failures during normal device operation.



The smallest unit that can be programmed or read simultaneously is a page; for erase operation, the smallest unit is a block. A page is formed by memory cells whose gates are connected to the same word line. Each page is independently encoded/de-coded in the ECC block. There is a page buffer located between ECC block and memory that temporarily holds the data. During write, data from I/O bus is serially encoded by ECC, and written to the desired page location from page buffer. During read, ECC block processes data in page buffer serially and transfers it to the I/O bus.

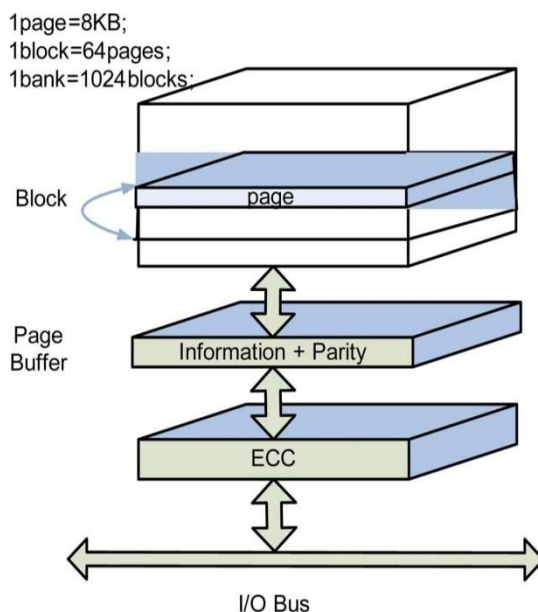


Fig 1. NAND Flash Memory Architecture.

B. Errors In Flash Memory:

Bit errors in Flash memories can be categorized into hard errors and soft errors. Hard errors usually result from oxide break-down due to Flash erase/program limitation and result in permanent failure bits in memory array. Soft errors result from different mechanisms, such as data retention and read/write disturbs, and can be recovered in the next P/E cycle.

Data stored in NAND Flash cells are required to remain valid for a certain period, typically around 10 years. MLC Flash data retention is orders of magnitude lower than SLC Flash. In MLC, the voltage window for threshold of each data state is smaller. All the programmed levels must be allocated in a predetermined sized voltage window. This leads to reduced spacing between adjacent programmed levels, making the MLC memories less reliable. Also, gradual charge leakage from the floating gate results in voltage shift in memory cells, ultimately resulting in a flip in the data stored in these cells.

Furthermore, read/write operations in MLC memory can cause threshold voltage fluctuations, which inadvertently results in errors in consecutive bits. Write operation of a cell consists of several steps in which threshold voltage of the selected transistors are increased. Increased variation in the threshold voltage can result in write errors. Another source of errors during write is the program disturb caused by applying high voltages to non-programmed cells and results in leakage and tunneling from body to floating gate. As a result, blocks that have been erased many times have a shorter data retention life than blocks with lower P/E cycles.



C. Introduction to Reed-Solomon codes:

Reed-Solomon codes are examples of error correcting codes, in which redundant information is added to data so that it can be recovered reliably despite errors in transmission or storage and retrieval. The error correction system used on CD's and DVD's is based on a Reed-Solomon code. These codes are also used on satellite links and other communications systems.

3. PROPOSED DESIGN/ALGORITHM

Most systems require the use of some form of error correction coding ECC in addition to constrained coding of the input signal or symbol sequence. It is therefore natural to investigate the interplay between these two forms of coding and the possibilities for efficiently combining their functions into a single coding operation_ in analogy to the coded modulation techniques now in wide use in data transmission. Proposed product codes use Reed-Solomon RS (255, 247) codes along rows and Hamming (72, 64) codes along columns and have reduced hardware overhead. The proposed product codes can correct errors in multi-level cell (MLC) NAND Flash memories with lower bit error rate (BER).

Product Code Encoding:

Fig.2 shows the encoding flow of the product code scheme, and gives an example of the physical address mapping of RS (255, 247) + Hamming (72, 64) product code when the page buffer size is 16 kB. Note that the physical mapping is different for different product codes. We assume that the Flash controller has the capability to reallocate the storage space to support the different product codes.

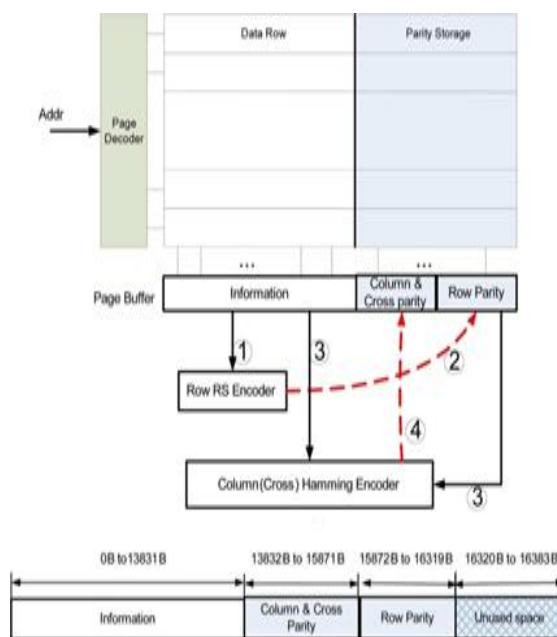


Fig 2. Product code encoding flow



For the RS (255, 247) + Hamming (72, 64) product code, during encoding, the RS encoder reads 247 information bytes at a time and generates 8 bytes or 64 bits corresponding to row parity. The row parity bits are stored in the pre-allocated region in the page buffer. Next, the Hamming encoder operates on the information and row parity bits, and generates the column and cross parity bits. The information bits are read with a stride of 247×8 , and the row parity bits are read with a stride of 8×8 . After column encoding, the column & cross parity bits are stored in the corresponding section of the page buffer. In the allocation, there is 64B unused space which can be used to store the beginning address of the different data regions for the Flash controller.

Product Code Decoding:

The decoding flow of RS + Hamming product codes is illustrated. For column decoding, the information bits in the page buffer are read out with a stride of 247×8 , the column & cross parity bits are read out with a stride of 1 and the row parity bits are read with a stride of 8×8 . The Hamming decoder corrects errors in information bits and row parity bits, and updates these bits in the page buffer.

Decoding involves following steps to decode the given information for detecting and correcting soft errors in MLC NAND Flash Memories.

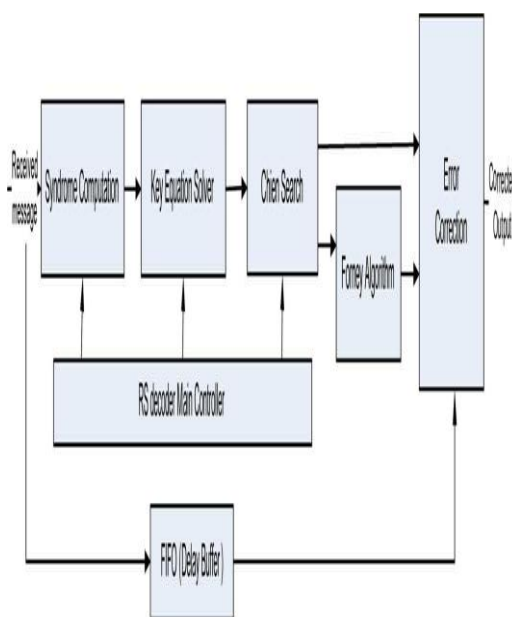


Fig 3. RS decoding Algorithm

- Calculate Syndromes.
- Berlekamp Massey Algorithm Calculates the Error Locator Polynomials.
- Chien Search - Finds the error locations using the Error Locator Polynomial.



- Forney’s Formula - Finds the error magnitudes using the error evaluator polynomial.
- Correct the Errors.

Combinations of RS and Hamming code for 8 kB page, if we use RS (127, 121) along rows, then there are 73 bits in each column. the remaining errors (three per row). In the extreme case, the proposed schemes can correct a very large number of errors. For instance, for a 16 kB page with RS (255, 247) along rows and Hamming (72, 64) along columns, the proposed scheme can correct 3 bytes (24 bits) of errors along each of the 56 rows and an additional $255 \times 8 - 24$ single bit errors along the remaining columns, leading to a total of 3360 errors. However, such a scenario is likely to never exist.

4. SIMULATION RESULTS

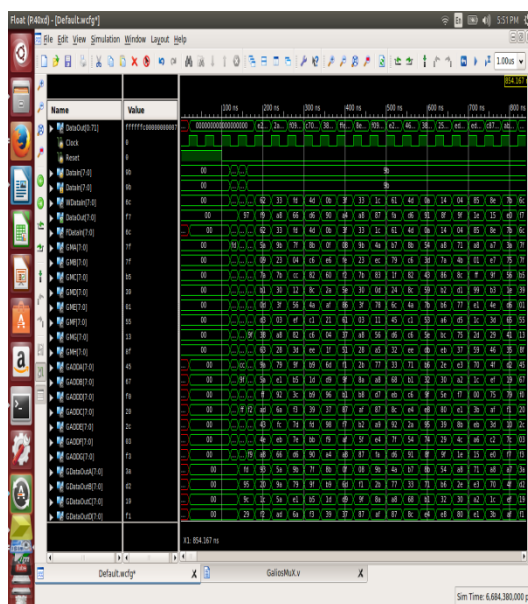


Fig 4. Product Encoder Output

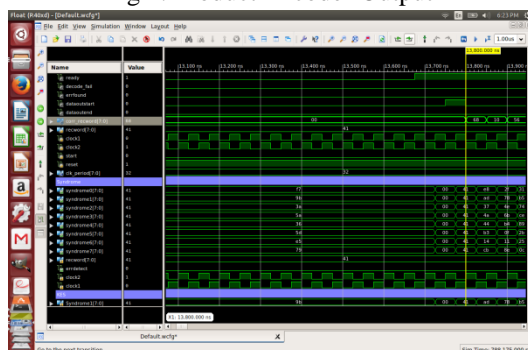


Fig 5. Error Enable Syndrome to the Reed Solomon decoder



GLOBAL JOURNAL OF ADVANCED RESEARCH
(Scholarly Peer Review Publishing System)

- [5] Micron, Boise, ID, "Wear-leveling techniques in NAND flash devices," gate memories?," IEEE Electron Device Lett., vol. 30, no. 2, pp. 178–180.
- [6] R. Micheloni, M. Picca, S. Amato, H. Schwalm, M. Scheppler, and Commodaro, "Non-volatile memories for removable media," Proc. IEEE, vol. 97, no. 1, pp. 148–160, Jan. 2009.
- [7] R. Bez, E. Camerlenghi, A. Modelli, and A. Visconti, "Introduction to flash memory," Proc. IEEE, vol. 91, no. 4, pp. 489–502, Apr. 2003.
- [8] L. M. Grupp, A. M. Caulfield, J. Coburn, S. Swanson, E. Yaakobi, P. Siegel, and J. K. Wolf, "Characterizing flash memory: Anomalies, observations, and applications," in Proc. 41st IEEE/ACM Int. Symp. Microarch. (MICRO), 2009, pp. 24–33.
- [9] N. Mielke, T. Marquart, N. Wu, J. Kessenich, H. Belgal, E. Schares, Truvedi, E. Goodness, and L. R. Nevill, "Bit error rate in NAND flash memories," in Proc. 46th Annu. Int. Reliab. Phys. Symp., 2008, pp. 9–19.