International Journal of Engineering Research and Science & Technology

ISSN : 2319-5991
Vol. 3, No. 4
November 2014

www.ijerst.com

Email: editorijerst@gmail.com or editor@ijerst.com

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com
Vol. 3, No. 4, November 2014

Research Paper

IMPLEMENTATION OF LOW-COMPLEXITY LDPC CODES FOR LOSSLESS APPLICATIONS

Nellipudi Surekha[1]* and V Srinivas[2]

Corresponding Author: **Nellipudi Surekha ✉ nellipudi.surekha@gmail.com*

This paper work presents the Error control coding for Flash memories by using Fault Tolerant Memory system architecture based on product code schemes. Error detection and correction or error control is techniques that enable reliable delivery of digital data over unreliable communication channels or storage medium. Error detection is the detection of errors caused by noise or other impairments during transmission from the transmitter to the receiver. Error correction is the detection of errors and reconstruction of the original, error-free data. The efficient error control schemes used which can handle both random and MBU errors. The Designed FTM architecture is modeled for each block in the Verilog HDL using XILINX ISE and the synthesis results prove that the reduction of hardware complexity in the implementation of proposed ECC in MLC NAND Flash memories.

Keywords: ECC, MLC, LDPC, Memories

INTRODUCTION

Flash memory is an electronic non-volatile computer storage device that can be electrically erased and reprogrammed. 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 which include read /write disturbs, data retention errors, bad block accumulation. 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

[1] M.Tech. Student, Department of ECE, Chirala Engineering College, Chirala 523155, Prakasam Dt., AP.
[2] Assistant Professor, Department of ECE, Chirala Engineering College, Chirala 523155, Prakasam Dt., AP.

threshold levels. To enhance the reliability of MLC NAND Flash memories and support longer lifetimes, soft errors should be reduced.Hence inorder to reduce the soft errors, Error control coding (ECC) is essential 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. ECC based on RS codes have been used in several commercial MLC Flash memories.They use plain RS codes and can correct up to minimum number of errors , at the cost of larger hardware and coding latency.

Due to the increase in error models in the flash memories, plain linear codes are no longer sufficient for the efficient error control coding. Hence product code schemes are proposed that have better ECC performance, lower area and smaller latency than plain codes with comparable

error correction capability.In this work, product code scheme having RS+Hamming codes is used. By using RS codes along rows and by using Hamming codes along columns, the remaining random errors can be corrected with very small overhead. The proposed RS+Hamming product code scheme has an additional advantage. It can be used to derive a flexible ECC scheme where the error correction capability increases to compensate for the larger number of errors caused by the increase in number of program/ erase cycles.

The key novel contribution of this project is identifying and defining a new class of error-correcting codes whose redundancy makes the design of fault-secure detectors (FSD) particularly simple. The parity-check Matrix of an FSD-ECC (fault secure detector - error correcting code) has a particular structure that the decoder circuit, generated from the parity-check Matrix, is Fault-Secure.Hence inorder to implement such a Fault Secure Detector- error control coding(FSD-ECC) using product code schemes, Fault tolerant memory(FTM) architecture is proposed and designed which use smaller constituent codes along rows and columns and achieve higher ECC due to cross parity checking. Such codes have less hardware overhead and have been successfully used in many embedded applications and interconnection networks

ERROR CORRECTION CODES

In information theory and coding theory with applications in computer science and telecommunication, **error detection and correction** or **error control** are techniques that enable reliable delivery of digital data over unreliable communication channels. Many communication channels are subject to channel noise, and thus errors may be introduced during transmission from the source to a receiver. Error detection techniques allow detecting such errors, while error correction enables reconstruction of the original data.The general idea for achieving error detection and correction is to add some redundancy (i.e., some extra data) to a message, which receivers can use to check consistency of the delivered message, and to recover data determined to be corrupted. Error-detection and correction schemes can be either systematic or non-systematic.In a systematic scheme, the transmitter sends the original data, and attaches a fixed number of check bits (or parity data), which are derived from the data bits by some

deterministic algorithm. If only error detection is required, a receiver can simply apply the same algorithm to the received data bits and compare its output with the received check bits; if the values do not match, an error has occurred at some point during the transmission. In a system that uses a non-systematic code, the original message is transformed into an encoded message that has at least as many bits as the original message.

Good error control performance requires the scheme to be selected based on the characteristics of the communication channel. Common channel models include memory-less models where errors occur randomly and with a certain probability, and dynamic models where errors occur primarily in bursts. Consequently, error-detecting and correcting codes can be generally distinguished between random-error-detecting/correcting and burst-error-detecting/correcting. Some codes can also be suitable for a mixture of random errors and burst errors.

LOW-DENSITY PARITY-CHECK (LDPC) CODES

Low-density parity-check (LDPC) codes are a class of linear block codes which provide a near capacity performance on a large collection of data transmission and storage channels while simultaneously admitting implementable decoders.In information theory, a low-density parity-check (LDPC) code is a linear error correcting code, a method of transmitting a message over a noisy transmission channel, and is constructed using a sparse bipartite graph. LDPC codes are capacity-approaching codes, which means that practical constructions exist that allow the noise threshold to be set very close (or even *arbitrarily* close on the BEC) to the theoretical maximum (the Shannon limit) for a symmetric memory-less channel. The noise threshold defines an upper bound for the channel noise, up to which the probability of lost information can be made as small as desired.Using iterative belief propagation techniques, LDPC codes can be decoded in time linear to their block length.

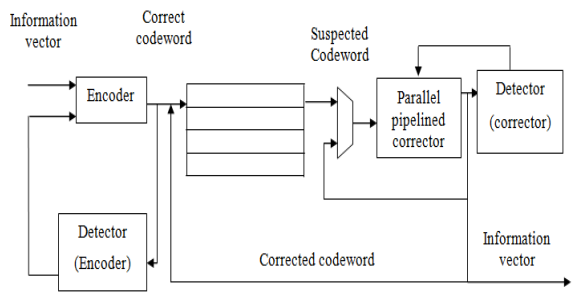
LDPC codes are finding increasing use in applications requiring reliable and highly efficient information transfer over bandwidth or return channel–constrained links in the presence of data-corrupting noise. Although implementation of LDPC codes has lagged behind that of other codes, notably turbo codes, the absence of encumbering software patents has made LDPC attractive to some. LDPC codes are also known as Gallager codes, in honor of Robert G. Gallager, who developed the LDPC concept in his doctoral dissertation at MIT in 1960. It has been shown that these codes achieve a remarkable performance with iterative decoding that is very close to the Shannon limit. Consequently, these codes have become strong competitors to turbo codes for error control in many communication and digital storage systems where high reliability is required.LDPC codes can be constructed using random or deterministic approaches. In this report, we focus on a class of LDPC codes known as Euclidean Geometric (EG) LDPC codes, which are constructed deterministically using the points and lines of a Euclidean geometry. The EG LDPC codes that we consider are cyclic and consequently their encoding can be efficiently implemented with linear shift registers. Minimum distances for EG codes are also reasonably good and can be derived analytically. Iteratively decoded EG LDPC codes also seem to not have the serious error- floors that plague randomly-constructed LDPC codes;

this fact can be explained by the observation made in that EG LDPC codes do not have pseudo-code words of weight smaller than their minimum distance. For these reasons, EG LDPC codes are good candidates for use in applications like optical communications that require very fast encoders and decoders and very low bit error-rates.

FAULT TOLERANT SYSTEM

Fault-tolerance or graceful degradation is the property that enables a system (often computer-based) to continue operating properly in the event of the failure of (or one or more faults within) some of its components. If its operating quality decreases at all, the decrease is proportional to the severity of the failure, as compared to a naïvely-designed system in which even a small failure can cause total breakdown. Fault-tolerance is particularly sought-after in high-availability or life-critical systems. Fault-tolerance is not just a property of individual machines; it may also characterize the rules by which they interact.

Figure 1: Proposed Fault Tolerant Memory Architecture



The information bits are fed into the encoder to encode the information vector, and the fault secure detector of the encoder verifies the validity of the encoded vector. If the detector detects any error, the encoding operation must be redone to generate the correct codeword. The codeword is then stored in the memory. Later during operation, the stored codeword will be retrieved from the memory unit. Since the codeword is susceptible to transient faults while it is stored in the memory, the retrieved codeword must be fed into the checker to detect any potential error and possibly to the corrector to recover any erroneous bits. The fault secure detector of the corrector unit guarantees the correctness of the corrector unit operations with the detect-and-repeat technique similar to the encoder.

Transient errors accumulate in the memory words over time. In order to avoid accumulation of too many errors in the memory words that surpasses the code correction capability, the system has to perform memory scrubbing. Memory scrubbing is periodically reading memory words from the memory, correcting any potential errors and writing them back into the memory. The effect of memory scrubbing and the details of each of the above units that make the fault tolerant memory structure, will be explain in the next chapter.

ECC WITH FAULT SECURE DETECTOR

In this work the encoder is protected with parity-prediction and paritychecker. The decoder is protected by adding a code checker (detector) block and a hammingdistance counter block to count the number of error bits at the output of the decoder. If the code checker detects a non-codeword, then the error in the decoder is detected. If the code checker detects a codeword but the hamming-distance counter indicate a non-zero error, then an error is also detected.Here we propose a multiple-error fault tolerant decoder and encoder that is general enough for any decoder and encoder implementation and for any

kind of ECC that satisfies the restricted ECC definition.

It is important to compare the rate of the EG-LDPC code with other codes to understand if the interesting properties of low-density and FSD-ECC come at the expense of lower code rates. We compare the code rates of the EG-LDPC codes that we use here with an achievable code rate upper bound (Gilbert- Varshamov bound) and a lower bound (Hamming bound). Table I shows the upper and lower bounds on the code overhead, for each of the used EG-LDPC. The EG-LDPC codes are no larger than the achievable Gilbert bound for the same *k* and *d* value, and they are not much larger than the Hamming bounds. Consequently, we see that we achieve the FSD property without sacrificing code compactness.

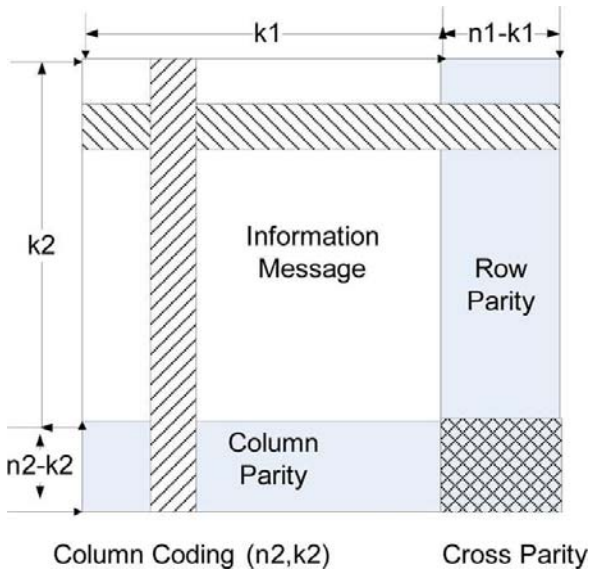
PRODUCT CODE SCHEMES

Product code is a technique to form a long length code with higher ECC capabilities using small length constituent codes. Compared to plain long length codes, it has high performance from cross parity check, and low circuitry overhead since the constituent codewords are of low error correction capability.

Let C_1 be a (n_1, k_1) linear code, and let C_2 be a (n_2, k_2) linear code. Then, a (n_1n_2,k_1k_2) linear code can be formed where each codeword can be arranged in a rectangular array of n_1 columns and n_2 rows such that every row is a codeword in C_1, and every column is a codeword in C_2, as shown in Figure 3.1

This code array can be formed by first performing row (column) Encoding then column (row) encoding on the data array of size of (k_1 X k2) . The cross parity block in the bottom right is of size (n_1 - k_1) X (n_2 - k_2) and is obtained by

Figure 2: Product Code Scheme General Architecture



encoding the row (column) parity along the other dimension, i.e., column (row).

If code C_1 has Hamming distance d_1 and code C_2 has Hamming Distance d_2 , the minimum weight of the product code is d_1d_2 exactly. Thus increasing the minimum weight of each code enhances the number of error patterns which can be corrected in the code array.

In order to provide for high error correction capability in Flash memories, we propose to use a strong code with multiple error correction capability along at least one of the dimensions. Since data is stored along rows in memory, we propose to use Stronger ECC along rows so that both random and burst errors can be dealt with efficiently. Furthermore, we choose a long codeword along this dimension to provide good coding performance.

ENCODER

A n-bit codeword *c* , which encodes a *k* -bit information vector *i* is generated by multiplying

the *k* -bit information vector with a *k X n* bit generator matrix *G* ; i.e.,

Figure 3: Generator Matrix for the (15, 7, 5) in Systematic Format

$$\begin{pmatrix} 1&0&0&0&0&0&0&1&0&0&1&1&1&0&1 \\ 0&1&0&0&0&0&0&1&1&0&0&1&1&1&0 \\ 0&0&1&0&0&0&0&0&1&1&1&0&0&0&1 \\ 0&0&0&1&0&0&0&1&0&1&1&1&0&0&0 \\ 0&0&0&0&1&0&0&0&1&0&1&1&1&0&0 \\ 0&0&0&0&0&1&0&0&0&1&0&1&1&1&0 \\ 0&0&0&0&0&0&1&0&0&0&1&0&1&1&1 \end{pmatrix}$$

EG-LDPC codes are not systematic and the information bits must be decoded from the encoded vector, which is not desirable for our fault-tolerant approach due to the further complication and delay that it adds to the operation. However, these codes are cyclic codes. Figure 4.3 shows the systematic generator matrix to generate (15, 7, 5) EG-LDPC code. In this figure *i* (i_0 ,i_1,i_2 ,.....i_6) is the information vector and will be copied to (c_0 ,.....c_6) bits of the encoded vector, *c* , and the rest of encoded vector, the parity bits, are linear sums (XOR) of the information bits. If the building block is two-inputs gates then the encoder circuitry takes 22 two-input XOR gates. Table 2 shows the area of the encoder circuits for each EG-LDPC codes under consideration based on their generator matrices. Once the XOR functions are known, the encoder structure is very similar to the detector structure shown in Figure 4.7.

Each of the XOR gates generates one parity bit of the encoded vector. The codeword consists of seven information bits followed by eight parity bits.

Figure 4: Structure of an Encoder Circuit for the (15, 7, 5) code

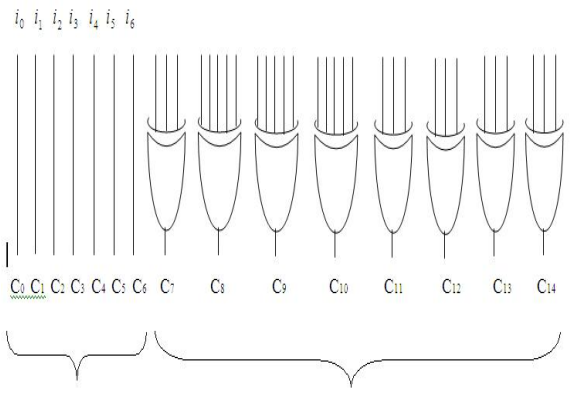


Figure 5: Simple Encoder Circuit

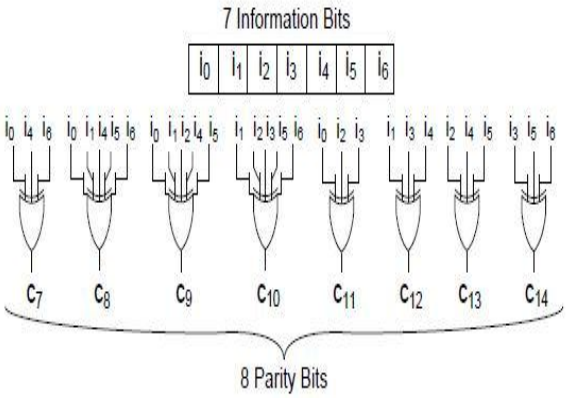


Table 1 shows the area of the encoder circuits for each EG-LDPC codes under consideration based on their generator matrices.

Table 1: Detector, Encoder, and Corrector circuit area in the number of 2-i/p gates

Code	(15,7,5)	(63,37,9)	(275,175,17)
encoder	45	501	3825
Decoder	22	355	6577
Serial connector	19	83	331
Parallel connector	285	5229	84405

RESULTS

Figure 6: RTL Schematic

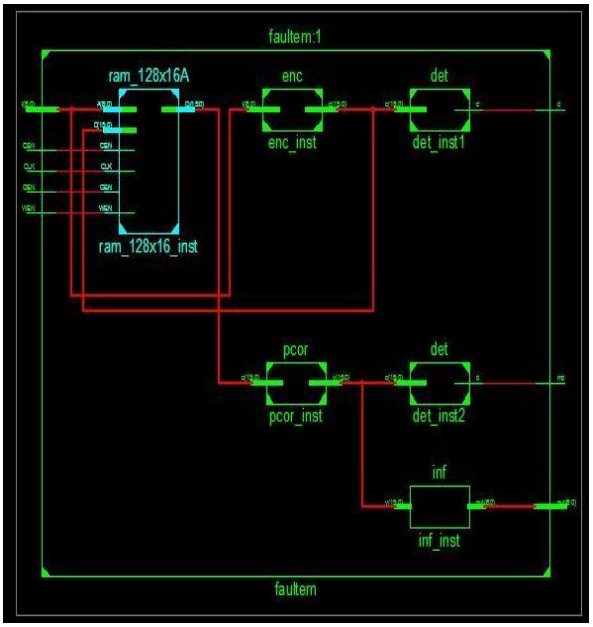


Figure 7: Technology Schematic

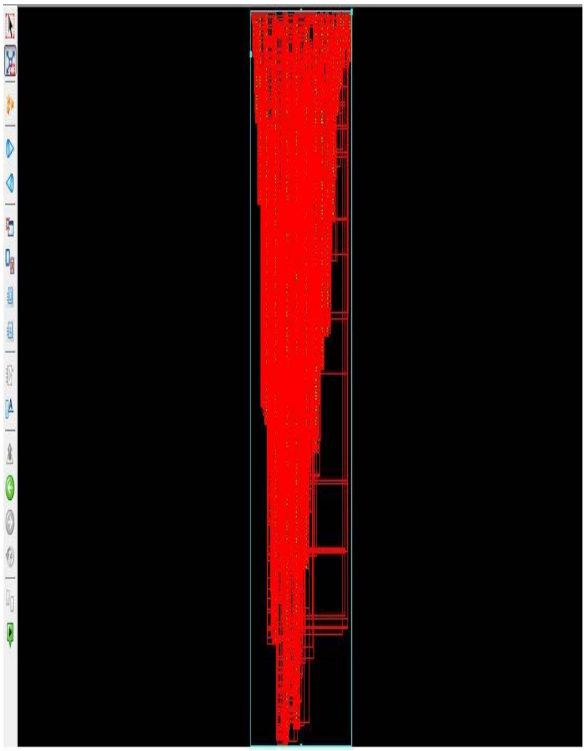
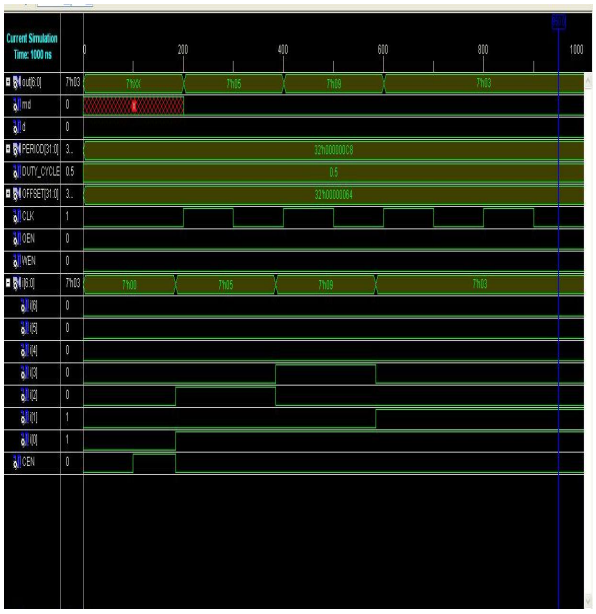


Figure 8: Waveform of FTS



CONCLUSION

In the proposed Error control coding(ECC) based on product code schemes, the no of hardware resources require to implement is also very much reduced. In this report, a fully fault-tolerant memory system that is capable of tolerating errors not only in the memory but also in the supporting logic is designed, Loss Less transmission of data is possible.

The designed ECC is flexible, safe and secure way for the data transmission in real time applications of Flash memories. To support the above statements, Various scenarios for each block of the Fault tolerant memory system is verified effectively during the simulation with respect to its behavior, which proves that There is no loss of data or control information in MLC NAND Flash Memories which can be made possible designing a 128 x 16 RAM.

REFERENCES

1. Chen T, Hsiao Y, Hsing Y and Wu C (2009), "An Adaptive-Rate Error Correction Scheme for NAND Flash Memory", *in Proc. 27th IEEE VLSI Test Symp*., pp. 53–58.
2. Chengen Yang, Yunus Emre, and Chaitali Chakrabarti (2012), "Product Code Schemes for Error Correction in MLC NAND Flash Memories", *IEEE, IEEE transactions on very large scale integration (vlsi) systems*, Vol. 20, No. 12.
3. Choi H, Liu W and Sung W (2010), "VLSI Implementation of BCH Error Correction for Multilevel Cell NAND Flash Memory", *IEEE Trans. Very Large Scale Integr. (VLSI) Syst*., Vol. 18, No. 5, pp. 843–847.
4. Gregori S, Cabrini A, Khouri O and Torelli G (2003), "On-chip Error Correcting Techniques for New-generation Flash Memories", *Proc. IEEE*, Vol. 91, No. 4, pp. 602–616.
5. Grupp L M, Caulfield A M, Coburn J, Swanson S, Yaakobi E, Siegel P H and Wolf J K (2009), "Characterizing Flash Memory: Anomalies, Observations, and Applications", *in Proc. 41st IEEE/ACM Int. Symp. Microarch. (MICRO)*, pp. 24–33.
6. Jianzhong T, Qinglin Q, Feng B, Jinye R and Yongxiang Z (2010), "Study and Design on High Reliability Mass Capacity Memory", *in Proc. IEEE Int. Conf. Softw. Eng. Service Sci.*, pp. 701–704.
7. Sribarki Srinath, Divakar B (2013), "Hardware Complexity Reduced Ecc With Fsd in MLC Nand Flash Memories", *(IJAREEIE) International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, Vol. 2, Issue 10, pp. 2278 –8875.
8. Young M (1989), *The Technical Writer's Handbook*, Mill Valley, CA, University Science, 1989.

International Journal of Engineering Research and Science & Technology
Hyderabad, INDIA. Ph: +91-09441351700, 09059645577
E-mail: editorijlerst@gmail.com or editor@ijerst.com
Website: www.ijerst.com