International Journal of
Scientific Engineering and Technology Research

www.ijsetr.com

ISSN 2319-8885
Vol.04,Issue.01
January-2015,
Pages:0130-0134

# An Efficient Decoding Architecture with Improved Error Correcting Technique for NAND Flash Memory

S.NARKISH HASHMA[1], S.SARANYA DEVI[2]

[1]PG Scholar, Dept of ECE, UCETW, Madurai, TamilNadu, India, E-mail: narkishhasma@gmail.com.

[2]Asst Prof, Dept of CSE ECE, UCETW, Madurai, TamilNadu, India.

**Abstract:** In this project, Due to higher integration densities, technology scaling and variation in parameters, the performance failures may occur for every application. The memory applications are also prone to single event upsets and transient errors which may lead to malfunctions. This paper proposed a novel error detection and correction method using EG-LDPC. This is useful as majority logic decoding can be implemented serially with simple hardware but requires a large decoding time. For memory applications, this increases the memory access time. The method detects whether a word has errors in the first iterations of majority logic decoding, and when there are no errors the decoding ends without completing the rest of the iterations. Also, errors affecting more than five bits were detected with a probability very close to one. Error commonly occurs in the Flash memory while employing LDPC decoding. The SRMMU actually suggests to use a the VTVI design by introducing the Context Number register, however also a PTPI or VTPI design could be implemented that complies to the SRMMU standard. The VTVI design with a physical write buffer and a combined I/D Cache TLB is the most simple design to implement. This will give error correction in minimum cyclic period using LDPC method. In this work, we proposed CLC combine with FAB and BPLRU to further improve the overall system performance by working with virtual memory management collaboratively. Here we proposed new management schemes for VM as well as VIVT cooperatively for flash memory based systems to reduce write activities and to improve I/O performance. In this architecture, we implement WBC-LRU and PCLRU for VM Management and VIVT management respectively with LDPC method. And to reduce the complexity compare to the existing architecture.

**Keywords:** Bose–Chaudhuri–Hocquenghem (BCH), Bit-Error Rate (BER), Cell-To-Cell Interference (CCI), Multilevel Cell (MLC).

## I. INTRODUCTION

Error correction codes are commonly used to protect memories from so called soft errors, which change the logical value of memory cells without damaging the circuit. As technology scales, memory devices become larger and more powerful error correction codes are needed. To this end, the use of more advanced codes has been recently proposed. These codes can correct a larger number of errors, but generally require complex decoders. To avoid a high decoding complexity, the use of one step majority logic decodable codes was first proposed in for memory application. One step majority logic decoding can be implemented serially with very simple circuitry, but requires long decoding times. In a memory, this would increase the access time which is an important system parameter. Only a few classes of codes can be decoded using one step majority logic decoding. Among those are some Euclidean geometry low density parity check (EG-LDPC) and difference set low density parity check (DS-LDPC) codes.

A method was recently proposed to accelerate a serial implementation of majority logic decoding of DS-LDPC codes. The idea behind the method is to use the first iterations of majority logic decoding to detect if the word being decoded contains errors. If there are no errors, then decoding can be stopped without completing the remaining iterations, therefore greatly reducing the decoding time. For a code with block length, majority logic decoding (when implemented serially) requires iterations, so that as the code size grows, so does the decoding time. In the proposed approach, only the first three iterations are used to detect errors, thereby achieving a large speed increase when is large. NAND Flash memory stores the information by changing the threshold voltage of floating gate transistors. Most of today's high-density NAND Flash memory devices store multiple (usually two) bits in a cell, and this multilevel cell (MLC) technology significantly increases the bit-error rate (BER).

## II. EXISTING SYSTEM

### A. Introduction

NAND Flash memory stores the information by changing the threshold voltage of floating gate transistors. Most of today's high-density NAND Flash memory devices store multi- ple (usually two) bits in a cell, and this multilevel cell (MLC) technology significantly increases the bit-error rate (BER). Moreover, as the feature size of NAND Flash memory shrinks, the number of electrons in the floating gate of a transistor also decreases, and as a result, the memory is very prone to charge loss caused by long data retention. The cell-to-cell interference (CCI) also increasingly deteriorates the reliability of information stored at the floating gates. It is also wellknown that SSD applications usually demand high program- and-erase cycles, which greatly affects the reliability of NAND Flash memory. NAND Flash memory devices have a spare region at each page, where parity bits for error correction can be stored.

Hard-decision decoding algorithms, such as Ham- ming and Bose–Chaudhuri–Hocquenghem (BCH) codes, have been widely used for NAND Flash memory error correction. However, as the process technology scales down continuously, more advanced error-correcting codes are needed to keep NAND Flash memory reliable. It is especially important to employ error-correcting algorithms that show good performance with a limited parity data ratio. Soft-decision error- correcting methods that sense the threshold voltage of a memcell in multiple-precision can increase the error-correcting performance because the reliability of stored information can also be utilized. In exsiting, we develop a NAND Flash memory error correction circuit using a rate-0.96 (68254, 65536) shortened Euclidean geometry (EG) LDPC code that can decode one page (8kB) of NAND Flash memory data at a time. This code employs the quasi-cyclic (QC) structure for the parity-check matrix to reduce the hardware complexity. This paper extends our early work in [20] that employs the normalized MS (NMS) algorithm.

In existing, the decoding hardware employs the normalized a posteriori probability (APP)-based algorithm that not only simplifies both the variable and check node operations but also shows good error-correcting performance for geometric LDPC codes. A conditional variable node update scheme is proposed to make the normalized APP-based algorithm resilient to decoding failure as well as to reduce circuit switching activities in the node processing units (NPUs). The error-correcting performance of the developed algorithm is also studied when varying the precision of the Flash memory output. To increase the decoding throughput while minimizing the chip area, the decoder adopts five-stage pipelined eight-way parallel architecture and employs three memory size reduction techniques: optimizing the word-length of extrinsic information, compressing the extrinsic information, and approximating the second minimum magnitudes. MLC Flash memory is introduced in order to model the non quantization scheme.

Bose–Chaudhuri–Hocquenghem (BCH) codes were proposed that have been widely used for NAND Flash memory error correction methodology. Soft-decision error correcting methods that sense the threshold voltage of a memory cell were proposed which is introduced for error correction methodology. A mutual information based memory quantizer as well as a (9118, 8225) irregular LDPC code is introduced in order to introduce and optimize the error methodology. The existing methods fails in the error correction methodologies since they not completely employed all the optimization process. The interconnection complexity is not discussed in all the existing works ZigBee technology is emerging following the Bluetooth. It is short-range, low power, low cost and low complexity of wireless communications technology. The technology is applies value in the home automation, building automation, industrial control and industrial areas of logistics.

## B. Low-Density Parity-Check Codes

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 memoryless 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 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. For large block sizes, LDPC codes are commonly constructed by first studying the behaviour of decoders. As the block size tends to infinity, LDPC decoders can be shown to have a noise threshold below which decoding is reliably achieved, and above which decoding is not achieved. This threshold can be optimised by finding the best proportion of arcs from check nodes and arcs from variable nodes. An approximate graphical approach to visualising this threshold is an EXIT chart.

The construction of a specific LDPC code after this optimization falls into two main types of techniques:

- Pseudorandom approaches
- Combinatorial approaches

Construction by a pseudo-random approach builds on theoretical results that, for large block size, a random construction gives good decoding performance. In general, pseudorandom codes have complex encoders, but pseudorandom codes with the best decoders can have simple encoders. Various constraints are often applied to help ensure that the desired properties expected at the theoretical limit of infinite block size occur at a finite block size.

Combinatorial approaches can be used to optimize the properties of small block-size LDPC codes or to create codes with simple encoders. Some LDPC codes are based on Reed-Solomon codes, such as the RS-LDPC code used in the 10-gigabit Ethernet standard. Compared to randomly generated LDPC codes, structured LDPC codes.

## III. PROPOSED SYSTEM

### A. Introduction

In this cooperative management schemes for virtual memory and write buffer to maximize the performance of Flash-memory-based systems was proposed. In this work, i proposed CLC combine with FAB and BPLRU to further improve the overall system performance by working with virtual memory management collaboratively. Here we proposed new management schemes for VM as well as VIVT cooperatively for flash memory based systems to reduce write activities and to improve I/O performance. .In my proposed work, i implement ML algorithm for checking error in a word with the combination of MLD EG-LDPC. This provides controlling mechanism when the bit from word was extracted. The method proposed in relies on the properties of DS-LDPC codes and therefore it is not directly applicable to other code classes. In the following, a similar approach for EG-LDPC codes is presented. A general solution to that problem four orders of magnitude better reliability compared to conventional SEC-DED codes for typical soft error rates. The NAND type is primarily used in main memory, memory cards, USB flash drives, solid-state drives, and similar products, for general storage and transfer of data. Flash memory is an electronic non-volatile computer storage medium that can be electrically erased and reprogrammed. Therefore, it is especially helpful for memories severely affected by increased error rates in scaled technologies, either due to soft errors or errors occurring due to process variations.

### B. Error Correction

Automatic Repeat reQuest (ARQ) is an error control method for data transmission that makes use of error-detection codes, acknowledgment and/or negative acknowledgment messages, and timeouts to achieve reliable data transmission. An acknowledgment is a message sent by the receiver to indicate that it has correctly received a data frame. Usually, when the transmitter does not receive the acknowledgment before the timeout occurs (i.e., within a reasonable amount of time after sending the data frame), it retransmits the frame until it is either correctly received or the error persists beyond a predetermined number of retransmissions. Three types of ARQ protocols are Stop-and-wait ARQ, Go-Back-N ARQ, and Selective Repeat ARQ.ARQ is appropriate if the communication channel has varying or unknown capacity, such as is the case on the Internet. However, ARQ requires the availability of a back channel, results in possibly increased latency due to retransmissions, and requires the maintenance of buffers and timers for retransmissions, which in the case of network congestion can put a strain on the server and overall network capacity.

### C. Hybrid Schemes

Hybrid ARQ is a combination of ARQ and forward error correction. There are two basic approaches:

- Messages are always transmitted with FEC parity data (and error-detection redundancy). A receiver decodes a message using the parity information, and requests retransmission using ARQ only if the parity data was not sufficient for successful decoding (identified through a failed integrity check).
- Messages are transmitted without parity data (only with error-detection information). If a receiver detects an error, it requests FEC information from the transmitter using ARQ, and uses it to reconstruct the original message.

The latter approach is particularly attractive on an erasure channel when using a rateless erasure code. The real benefits for NAND are faster program and erase times, as NAND provides over 5 Mbytes/s of sustained write performance. NAND-flash occupies smaller chip area per cell. This maker NAND available in greater storage densities and at lower costs per bit than NOR-flash. a new write buffer management scheme called Block Padding Least Recently Used (BPLRU) to enhance the random write performance of flash storage. BPLRU considers the common FTL characteristics and attempts to establish a desirable write pattern with RAM buffering. More specifically, BPLRU uses three key techniques, block-level LRU, page padding, and LRU compensation. Block-level LRU updates the LRU list considering the size of the erasable block to minimize the number of merge operations in the FTL. Page padding changes the fragmented write patterns to sequential ones to reduce the buffer flushing cost. LRU compensation adjusts the LRU list to use RAM for random writes more effectively. Using write traces from several typical tasks and three different file systems, we show that BPLRU is much more effective than previously proposed schemes, both in simulation and in experiments with a real prototype.

- Here in this process we implement WBC-LRU and PCLRU for VM Management and VIVT management respectively with LDPC method.
- This process to reduce the circuit complexity. And also reduce the power consumption.

## IV. BLOCK DIAGRAM OF PROPOSED ARCHITECTURE

**Modules and Modules Description:**

- NAND- flash memory
- NPU(Node Processing Unit)
- APP MEMORY
- VIVT design management

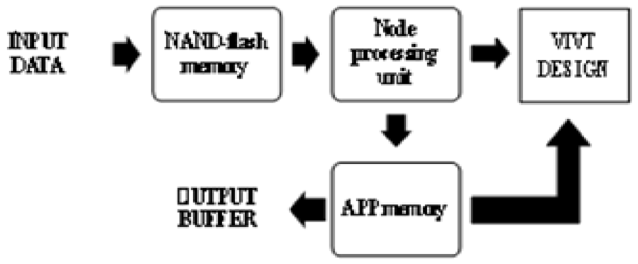

**Fig.1. Block diagram.**

### A. NAND- Flash Memory

The input data to apply the NAND flash memory architecture. And the input data to be stored the correct address bit position. And the data bit first to be write given specified location and then read the exact memory location. The NAND flash memory to be consider the array of memory location based on the NAND based storage process. NAND Flash is used in a wide variety of industrial, communication, automotive and consumer devices requiring non-volatile storage for code, data or content. To respond to market demands for lower costs and increased storage capacity, fabrication technology is advancing quickly to each successive new process node. The NAND Flash memory is composed of the blocks of pages, one block is usually composed of 16, 32 or 64 pages. For most NAND Flash devices there are 512 bytes / 256 words in the "Cell Array" page area (also called "data area") and an extra 16 bytes / 8 words in the "Spare Cell Array" page area (also called "spare area"). There are total 528 bytes / 264 words per page (see Fig.1) and such page is called "small page". For huge capacities (usually 1 Gbit and more) "large page" is used (see Fig.2). It contains 2048 bytes / 1024 words of data area and 64 bytes / 32 words of spare area (2112 bytes / 1056 words total).

### B. NPU (Node Processing Unit)

The node processing unit to find the NAND flash memory data and to applied the node processing architecture. The node processing unit, to find the memory data and to modify the input memory value and to improve node processing level. The min selector to find the magnitude values to be selected by node processing unit. This unit to find the memory input decoding data.

### C. APP Memory

The decoding hardware employs the normalized a posteriori probability (APP)-based algorithm that not only simplifies both the variable and check node operations but also shows good error-correcting performance for geometric LDPC codes. The APP memory to be modified based on the NPU values and to apply the input selection process. So we use the shift registers and the parity check process in APP memory architecture.

### D. VIVT Design Management

The VIVT design process used to correct the error in APP memory architecture. And this technique to improve the error correction process. Because VTVI design to contains the shift register and parity check process for APP architecture. This technique to increase the accuracy level compare to the LDPC methodology. This architecture to find required output for the decoding values in VIVT design management. The flash aware buffer policy (FAB) [10] is another buffer cache management policy used for flash memory. In FAB, the buffers that belong to the same erasable block of flash memory are grouped together, and the groups are maintained in LRU order: a group is moved to the beginning of the list when a buffer in the group is read or updated, or a new buffer is added to the group. When all buffers are full, a group that has the largest number of buffers is selected as victim. If more than one group has the same largest number of buffers, the least recently used of them is selected as a victim. All the dirty buffers in the victim group are flushed, and all the clean buffers in it are discarded. The main use of FAB is in portable media player applications in which the majority of write requests are sequential. FAB is very effective compared to LRU. Note that BPLRU targets random write patterns.

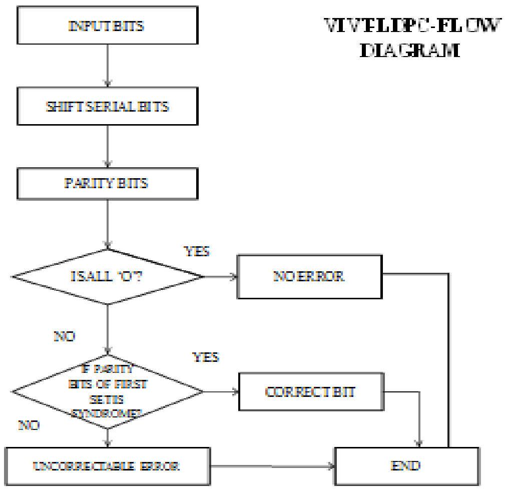

**Fig.2. Flow chart.**

## VI. RESULTS OF EXPERIMENT

Experiment results of this paper is shown in bellow Figs. 3, 4 and 5.

### A. Performance Chart

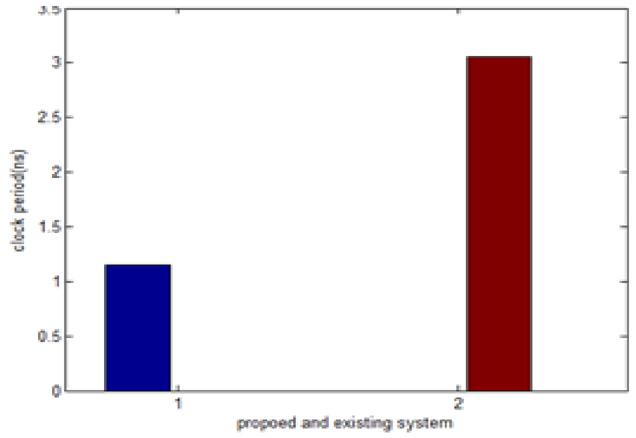

**Fig.3. Power Consumption.**

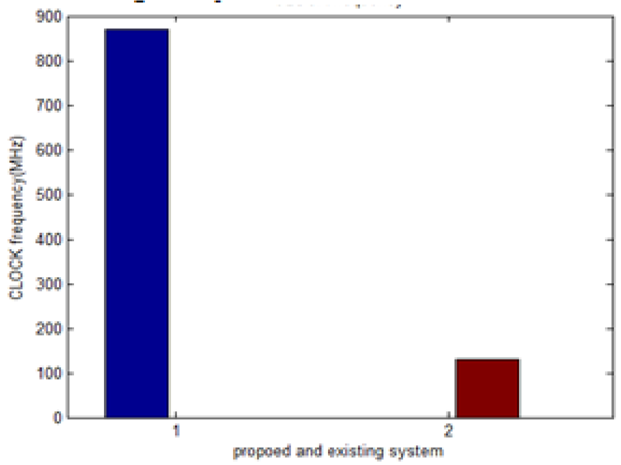

**Fig.4. Clock Frequency.**

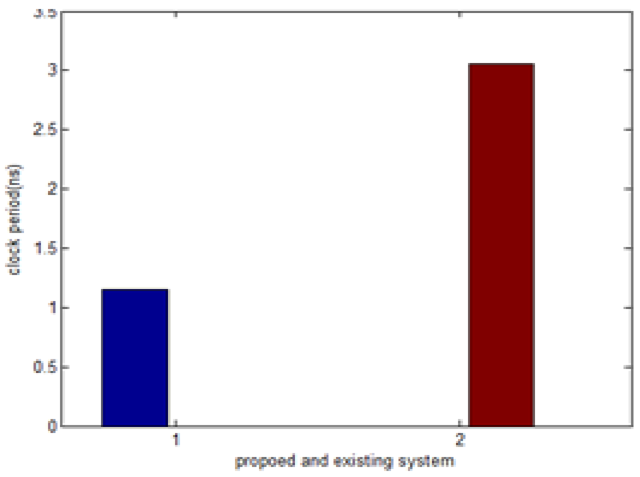

**Fig.5. Clock Period.**

## VII. CONCLUSION

In this project, the proposed ML algorithm is used for error detection and correction in MLD EG-LDPC. This was made by providing a controlling mechanism in the stage of parity decoding. Output shows the performance of our proposed system. In our work, we use a property of EG-LDPC codes that allows simpler decoding logic (involving majority gates) to be used, which can be implemented in nano hardware using different techniques. The simulation results show that all tested combinations of errors affecting up to four bits are detected in the first three iterations of decoding.

## VIII. REFERENCES

[1] K. Kim, “Technology for sub-50nm DRAM and NAND flash manufacturing,”in IEDM Tech. Dig., Dec. 2005, pp. 323–326.

[2] G.Dong, S. Li, and T. Zhang, “Using data postcompensationandpredistortion to tolerate cell-to-cell interference in MLC NAND flash memory,” IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 57, no. 10,pp. 2718–2728, Oct. 2010.

[3] L. M. Grupp, A. M. Caulfield, J. Coburn, S. Swanson, E. Yaakobi,P. H. Siegel, and J. K. Wolf, “Characterizing flash memory: Anomalies,observations, and applications,” in Proc. 42nd Ann. IEEE/ACM Int.Symp. Microarchitecture, Dec. 2009, pp. 24–33.

[4] S. Gregori, A. Cabrini, O. Khouri, and G. Torelli, “On-chip error correcting techniques for new-generation flash memories,” Proc. IEEE,vol. 91, no. 4, pp. 602–616, Apr. 2003.

[5] W. Liu, J. Rho, and W. Sung, “Low-power high-throughput BCH errocorrection VLSI design for multi-level cell NAND flash memories,”in Proc. IEEE Workshop Signal Process. Syst. Design Implement.,
Oct. 2006, pp. 303–308.

[6] R. Micheloni, R. Ravasio, A. Marelli, E. Alice, V. Altieri, A. Bovino,L. Crippa, E. Di Martino, L. D’Onofrio, A. Gambardella, E. Grillea,G. Guerra, D. Kim, C. Missiroli, I. Motta, A. Prisco, G. Ragone,M. Romano, M. Sangalli, P. Sauro, M. Scotti, and S. Won, “A 4 Gb 2b/cell NAND flash memory with embedded 5b BCH ECC for 36MB/s system read throughput,” in IEEE Int. Solid-State Circuits Conf. Dig.Tech. Papers, Feb. 2006, pp. 497–506.

[7] F. Sun, S. Devarajan, K. Rose, and T. Zhang, “Design of on-chip errorcorrection systems for multilevel NOR and NAND flash memories,”IET Circuits, Devices Syst., vol. 1, no. 3, pp. 241–249, Jun. 2007.

[8] T.-H. Chen, Y.-Y.Hsiao, Y.-T.Hsing, and C.-W. Wu, “An adaptive-rateerror correction scheme for NAND flash memory,” in Proc. 27th IEEEVLSI Test Symp., May 2009, pp. 53–58.

[9] R. G. Gallager, “Low-density parity-check codes,” IRE Trans. Inf.Theory, vol. 8, no. 1, pp. 21–28, Jan. 1962.

[10] R. G. Gallager, Low-Density Parity-Check Codes. Cambridge, MA,USA: MIT Press, 1963.

[11] Digital Video Broadcasting (DVB): Second Generation System forBroadcasting, Interactive Services, News Gathering and Other BroadbandSatellite Applications, ETSI Standard 302 307, 2005.

[12] IEEE Standard for Information Technology-Telecommunications andInformation Exchange between Systems-Local and Metropolitan AreaNetworks-Specific Requirements Part 3: Carrier Sense Multiple Accesswith Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications, IEEE Standard 802.3an, 2006.

[13] G. Dong, N. Xie, and T. Zhang, “On the use of soft-decision error-correction codes in NAND flash memory,” IEEE Trans. Circuits Syst.I,Reg. Papers, vol. 58, no. 2, pp. 429–439, Feb. 2011.

[14] G. Dong, Y. Pan, N. Xie, C. Varanasi, and T. Zhang, “Estimating information-theoretical NAND flash memory storage capacity and its implication to memory system design space exploration,” IEEE Trans.Very Large Scale Integr. (VLSI) Syst., vol. 20, no. 9, pp. 1705–1714,Sep. 2012.

[15] J. Wang, T. Courtade, H. Shankar, and R. Wesel, “Soft informationfor LDPC decoding in flash: Mutual-information optimized quantization,” in Proc. IEEE Global Telecommun. Conf., Dec. 2011.