ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 10, October 2013

HARDWARE COMPLEXITY REDUCED ECC WITH FSD IN MLC NAND FLASH MEMORIES

Sribarki Srinath[1], Bantupalli Divakar[2]

PG Student, Dept. of ECE, PYDAH College of engineering & Technology, Visakhapatnam, Andhra Pradesh, India[1]

Assistant professor, Dept. of ECE, PYDAH College of engineering & Technology, Visakhapatnam, Andhra Pradesh, India[2]

ABSTRACT— Generally, Memory cells have been protected from soft errors. Particularly in the case of Flash memories, Due to the increase in soft error rate, the encoder and decoder circuitry around the memory blocks have become susceptible to soft errors as well and must also be protected. In order to control soft errors in the memories, Error control coding (ECC) is used. ECC algorithm correction strength (number of bit errors that can be corrected) depends on the ECC algorithm used to correct the soft errors. Simple Hamming codes can only correct single bit errors. Reed-Solomon code can correct more errors but limited to multiple bit errors only. Hence in order to enhance error correction capability and reduced hardware overhead, we are proposing a new design approach based on product code ECC scheme which consists of fault secure encoder and decoder circuitry for memory designs. We are also using Euclidean Geometry Low-Density Parity-Check (EG-LDPC) codes in Fault secure detector (FSD) which can makes design of FSD simple and can also achieve higher reliability and lower area overhead.

Keywords-Hamming Codes, Reed-Solomon codes, EG-LPDC (Euclidean geometry-low density parity check) codes, Error correction codes (ECCs), Fault secure Detector (FSD), and flash memories

I. INTRODUCTION

Flash memory is an electronic non-volatile computer storage device that can be electrically erased and reprogrammed. 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 NAND flash development was to reduce the chip area required to implement a given capacity of flash memory, and thereby to reduce cost per bit and increase maximum chip capacity so that flash memory could compete with magnetic storage devices like hard disks.

But there are some 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. So in order to enhance the reliability of NAND Flash memories and support longer lifetimes, combinations of hardware and software techniques are used.

Here In our paper we are concentrated on the software error control coding techniques which arise, 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.

II.ERROR CORRECTION CODES

Error Correction Codes (ECC) encodes data in such a way that a decoder can identify and correct certain errors in the data. Studies on Error Correction Codes started in the late 1940's with the works of Shannon and Hamming, and since then thousands of papers have been published on the subject. Usually data strings are encoded by adding a number of redundant bits to them. When the original data is reconstructed, a decoder examines the encoded message, to check for any errors.

There are two basic types of Error Correction Codes .

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 10, October 2013

Block Codes, which are referred to as (n, k) codes. A block of k data bits is encoded to become a block of n bits called a code word.

Convolution Codes, where the code words produced depend on both the data message and a given number of previously encoded messages. The encoder changes state with every message processed. The length of the code word is usually constant.

NAND Flash memories typically use Block Codes.

Block Codes

The Block Code family can be divided as in figure 1:

Linear Codes, where every linear combination of valid code words (such as a binary sum) produces another valid code word. Examples of linear codes are:

Cyclic Codes, where every cyclic shift by a valid code word also yields a valid code word

Hamming Codes, which are able to detect three errors and correct one.

Systematic Codes, where each code word includes the exact data bits from the original message of length k, either followed or preceded by a separate group of check bits of length q

In all cases the code words are longer than the data words on which they are based.

Block codes are referred to as (n, k) codes. A block of k data bits is encoded into a block of n bits called a code word. The code takes k data bits and computes (n − k) parity bits from the code generator matrix.

Most Block Codes are systematic, in that the data bits remain unchanged, with the parity bits attached

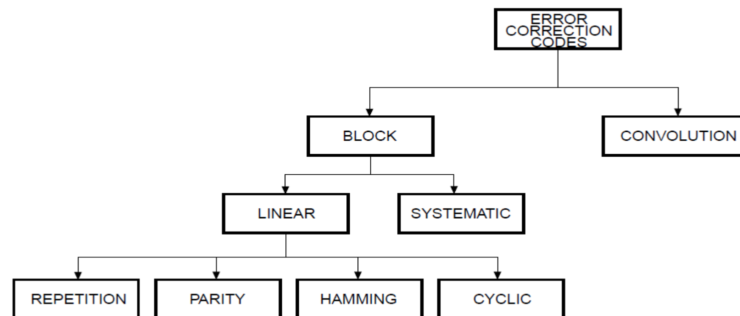

Fig.1 Types of Error Correction Codes

Figure 1 shows the classification of error correction codes and also its sub classes

III. SLC AND MLC

NAND flash memory uses an array of floating gate transistors to store information. In traditional single-level cell (SLC) devices, each bit cell stores only one bit of information. Multilevel cell (MLC) devices can store more than One bit per cell by partially charging the bit cells to encode multiple bit states. Since the underlying bit cells used to construct both SLC and MLC are similar, SLC flash will always be more expensive than MLC flash on a per gigabyte basis. Currently, SLC is about twice as expensive as MLC for low-density flash parts. For this reason, MLC NAND flash has become the dominant form of NAND flash and constitutes about 90% of the flash parts shipped.

Fig. 2 illustrates the distribution of threshold voltages for SLC and MLC (3 bit) storage. As the number of storage levels increase, storage density of one cell improves at an expense of reduction in reliability.

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 10, October 2013

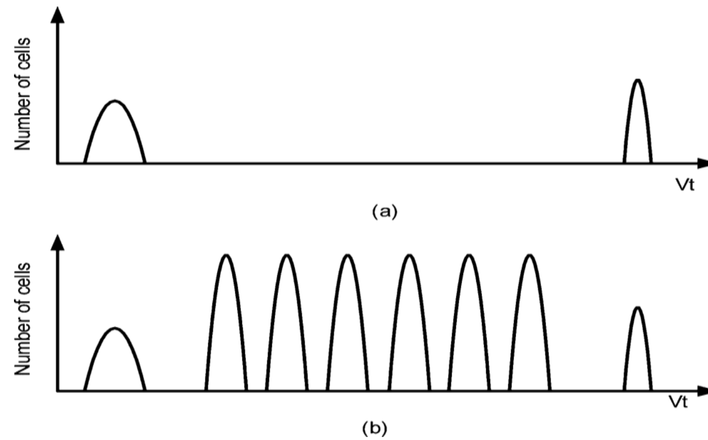

Fig.2 Conceptual representation of threshold voltage distributions for (a) SLC and (b) 3-bit MLC in Flash memory cells.

IV. IMPLEMENTATION OF PRODUCT CODE ECC SCHEMES FOR FLASH MEMORY

In this paper we propose use of product codes which use smaller constituent codes along rows and columns and achieve high error correction capability due to cross parity checking. Such codes have lower hardware overhead and have been successfully used in embedded SRAM caches and interconnection networks.

A. Product Code Scheme:

Basics 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 code words are of low error correction capability.

Let C1 be a (n1, k1) linear code, and let C2 be a (n2, k2) linear code. Then, a (n1n2,k1k2) linear code can be formed where each codeword can be arranged in a rectangular array of n1 columns and n2 rows such that every row is a codeword in C1, and every column is a codeword in C2, as show in Fig 3.

This code array can be formed by first performing row (column) Encoding then column (row) encoding on the data array of size of (k1 X k2). The cross parity block in the bottom right is of size (n1 - k1) X (n2 - k2) and is obtained by encoding the row (column) parity along the other dimension, i.e., column (row).

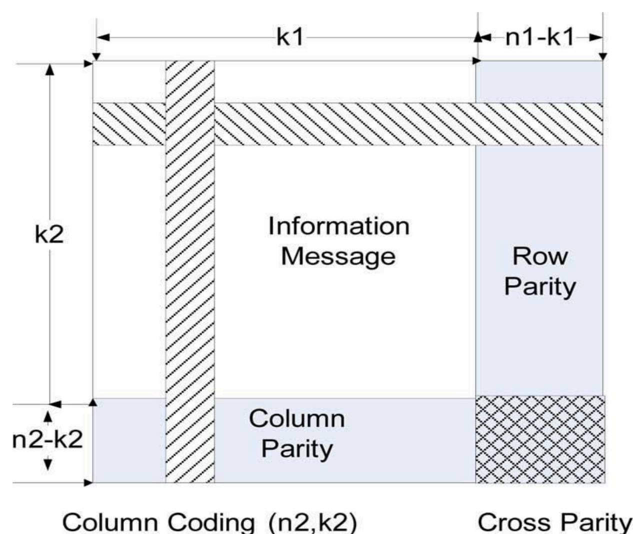

Fig.3 Product code scheme

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 10, October 2013

Figure 3 shows the implementation of product code scheme along the row and column parity for a given information message.

If code C1 has Hamming distance d1 and code C2 h as Hamming Distance d2, the minimum weight of the product code is d1d2 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.

V. ECC WITH FAULT SECURE DETECTOR (FSD)

B. Error-Correcting Code:

Let $i=(i_0,i_1,i_2,........i_{k-1})$ be the k- bit information vector that will be encoded into an n-bit codeword $c=(c_0,c_1,.........c_{n-1})$, for linear codes, the encoding operation essentially performs the following vector-matrix multiplication:

$$c=i.G$$ Where **G** is a **k x n** generator matrix.

The validity of a received encoded vector can be checked with the Parity-Check matrix, which is a **(n-k) x n** binary matrix named **H**. The checking or detecting operation is basically summarized as the following vector-matrix multiplication:

$$s=c.H^T$$

The (n-k) -bit vector s is called the syndrome vector.

A syndrome vector is zero if c is a valid codeword, and nonzero if c is an erroneous codeword. Each code is uniquely specified by its generator matrix or parity-check matrix.

A code is a systematic code if every codeword consists of the original **k**-bit information vector followed by **(n-k)** parity bits. With this definition, the generator matrix of a systematic code must have the following structure:

$$G=[I:X]$$

The minimum distance of an ECC, d is the minimum number of code bits that are different between any two code words. The maximum number of errors that an ECC can detect is **d-1**, and the maximum number that it corrects is **d/2**. Any ECC is represented with a triple **(n,k,d)**, representing code length, information bit length, and minimum distance, respectively.

B.FSD-ECC Definition

The restricted ECC definition which guarantees a FSD-ECC is as follows.

Definition I: Let be an ECC with minimum distance is FSD-ECC if it can detect any combination of overall or fewer errors in the received codeword and in the detector circuitry.

Theorem: Let **c** be an ECC, with minimum distance **d**. **c** is FSD-ECC if any error vector of weight 0 **< e < d-1** has syndrome vector of weight at least **d-e**.

Note: The following proof depends on the fact that any single error in the detector circuitry can corrupt at most one output (one syndrome bit). This can be easily satisfied for any circuit by implementing the circuit in such a way that no logic element is shared among multiple output bits; therefore, any single error in the circuit corrupts at most one output (one syndrome bit).

C. EG-LDPC codes (Euclidean geometry-low density parity check)

In this paper, 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 [10]. 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 generally do not suffer as much from the error-floor problems that plague some randomly constructed LDPC codes.

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 (BERs).

And also 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 and a lower bound (Hamming bound). The EG-LDPC codes are no larger than

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 10, October 2013

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.

VI. FAULT-TOLERENT MEMORY SYSTEM OVERVIEW

We outline our memory system design that can tolerate errors in any part of the system, including the storage unit and encoder and corrector circuits using the fault-secure detector. For a particular ECC used for memory protection, let **E** be the maximum number of error bits that the code can correct and **D** be the maximum number of error bits that it can detect, and in one error combination that strikes the system, let e_e , e_m and e_c be the number of errors in encoder, a memory word, and corrector, and let e_{de} and e_{dc} be the number of errors in the two separate detectors monitoring the encoder and corrector units.

1) Any single error in the encoder or corrector circuitry can at most corrupt a single codeword bit (i.e., no single error can propagate to multiple codeword bits);

2) There is a fault secure detector that can detect any combination of errors in the received codeword along with errors in the detector circuit. This fault-secure detector can verify the correctness of the encoder and corrector operation.

The first property is easily satisfied by preventing logic sharing between the circuits producing each codeword bit or information bit in the encoder and the corrector respectively. We define the requirements for a code to satisfy the second property.

The diagram is described as follows, 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. During memory access operation, the stored code words will be accessed from the memory unit. Code words are susceptible to transient faults while they are stored in the memory. Therefore a corrector unit is designed to correct potential errors in the retrieved code words. In our design (see Fig. 4) all the memory words pass through the corrector and any potential error in the memory words will be corrected. Similar to the encoder unit, a fault-secure detector monitors the operation of the corrector unit. All the units shown in Fig. 6 are implemented in fault-prone the only component which must be implemented in reliable circuitry are two OR gates that Accumulate the syndrome bits for the detectors.

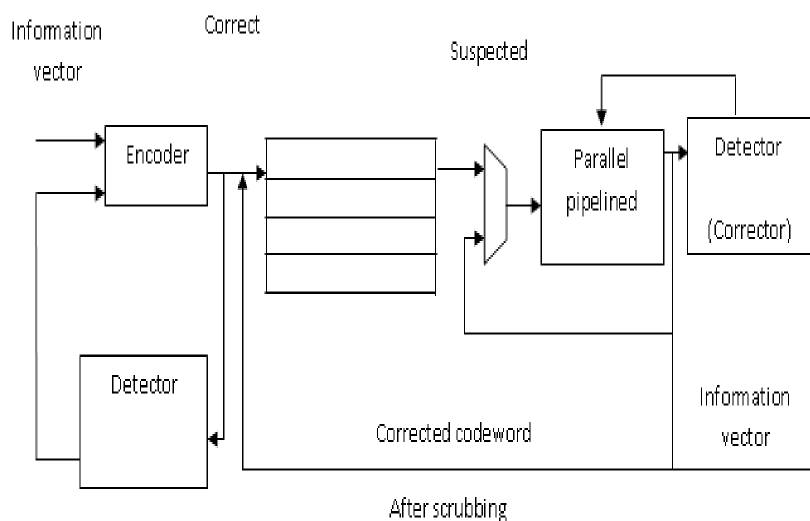

Fig 4: Fault-tolerant memory architecture, with pipelined corrector

An overview of our proposed reliable memory system is shown in Fig.4. Figure.4 shows the Fault-tolerant memory architecture, with pipelined corrector which is going to be used for the design of fault tolerant memory system which can produce a lossless transmission of data.

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 10, October 2013

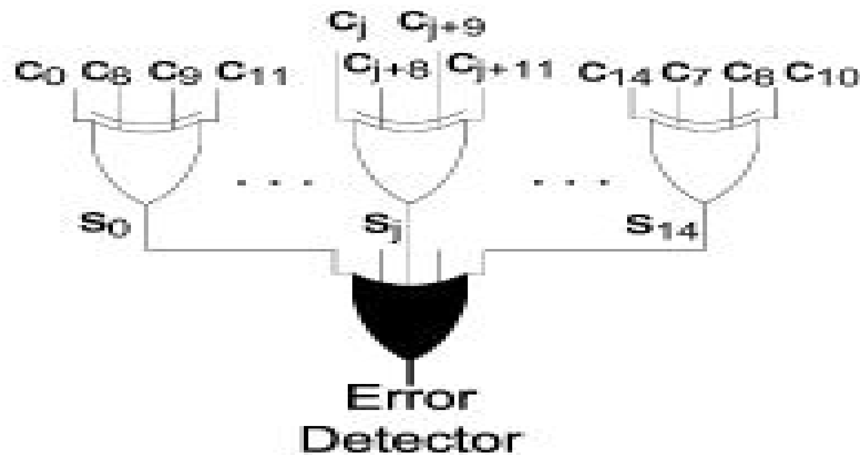

Fig 5: Fault-secure detector for (15, 7, 5) LDPC Code

Figure.5 shows the fault secure detector logic diagram for low density parity check code (15, 7, 5) by which detection an error in the given input code can be determined as well as a secure transmission can be made.

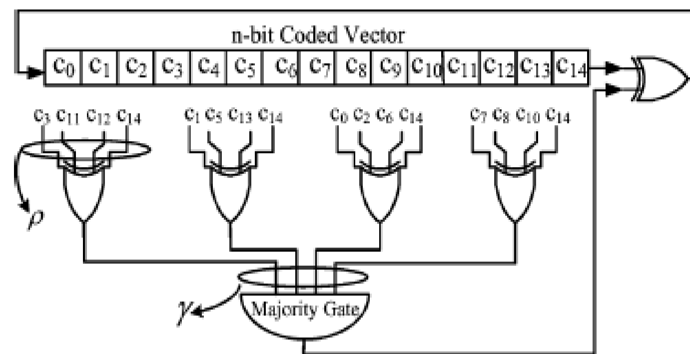

Fig 6: Serial one-step majority logic corrector structure

Fig. 6 corrects the code bit using the output of the majority gate. Once the code bit is corrected the codeword is cyclic shifted and code bit is placed at position and will be corrected. The whole codeword can be corrected in' n' rounds

C. Encoder

An 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. c = i.G.

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 [2]. We used the procedure presented in [2] and [4] to convert the cyclic generator matrices to systematic generator matrices for all the EG-LDPC codes under consideration.

Note the identity matrix in the left columns.

Fig. 7 shows the systematic generator matrix to generate (15, 7, 5) EG-LDPC code. The encoded vector consists of information bits followed by parity bits, where each parity bit is simply an inner product of information vector and a column of X, from **G=[I : X]**.

.

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 10, October 2013

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

Fig 7: Generator matrix for the (15, 7, 5) EG-LDPC in systematic forma

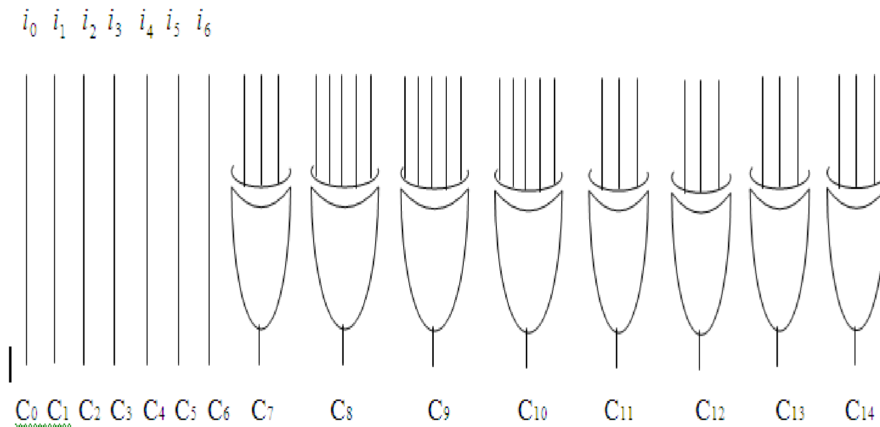

Fig 8: Structure of an encoder circuit for the (15, 7, 5)

Fig. 8 shows the encoder circuit to compute the parity bits of the (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 consists two-input gates then the encoder circuitry takes 22 two-input XOR gates

Table 1 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 Fig. 8. EG-LDPC code; $\mathbf{i_0}$ to $\mathbf{i_6}$ are 7-bit information vector. 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.

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

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

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 10, October 2013

VII. RESULTS

The results are obtained from simulating the verilog programming for each and every block in the fault memory architecture (fig-4) and also from overall hierarchical based design; we had obtained the design of a NAND flash memories.

The results also include wave forms obtained from verilog simulation (fig-9) in XILINX-ise regarding design of NAND flash memories and also it shows synthesis report and RTL Schematic (fig-10).

The macro statistics (table 2) show you the reduction of hardware complexity by using product code schemes and EG-LDPC codes.

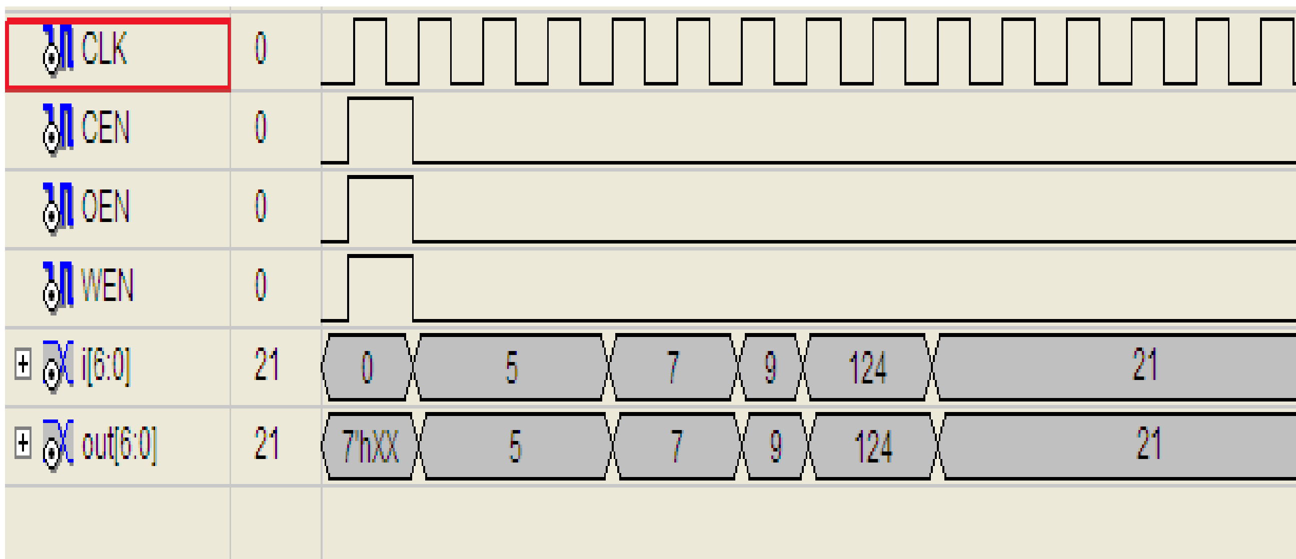

Fig- 9: WAVEFORMs obtained from simulation OF NAND FLASH MEMORY DESIGN

Figure 9 shows the waveforms obtained from simulation which shows that the input given is exactly obtained as an output i.e a lossless transmission of data can be possible with this proposed reliable system.

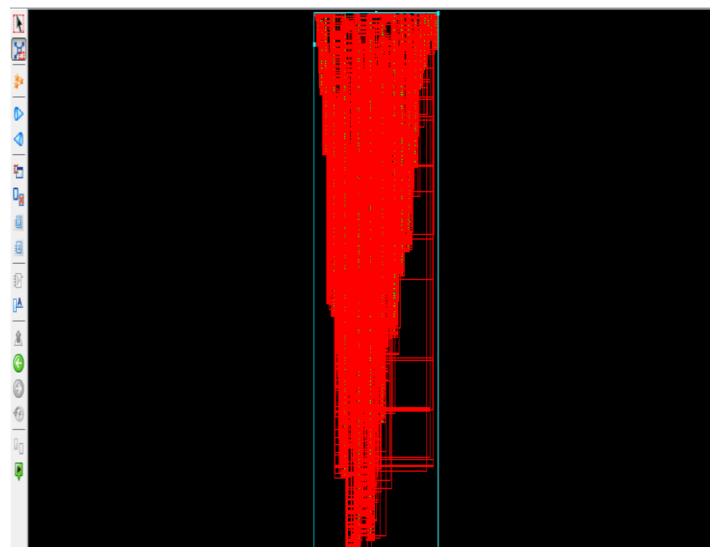

Fig-10: RTL Schematic

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 10, October 2013

# ROMs	1
4x16-bit ROM	1
# Latches	30
1-bit latch	24
16-bit latch	3
7-bit latch	3
# Tristates	16
1-bit tristate buffer	16
# Xors	117
1-bit xor2	23
1-bit xor3	4
1-bit xor4	90

Table-2: Macro Statistics

VIII. CONCLUSION

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.

In this paper, 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. With the enhancement of the comprehensive mission management system the proposed design method will also take fewer amounts of cycles to obtain the transmitted data.

REFERENCES

[1] Chengen Yang, Yunus Emre, and Chaitali Chakrabarti,"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, december 2012

[2] 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.

[3] T. Chen, Y. Hsiao, Y. Hsing, and C.Wu, "An adaptive-rate error correction scheme for NAND flash memory," in Proc. 27th IEEE VLSI Test Symp., 2009, pp. 53–58.

[4] H. Choi, W. Liu, and W. Sung, "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, May 2010.

[5] G. Gibson, D. Dawn ,"Objectionable web content classification using Neural Network," Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955. (references) Rich and Knight, Artificial Intellegence, 2nd ed.,Tata Mc Graw Hill Publication, 2003, pp.68–73.

[6] I. S. Jacobs and C. P. Bean, "Fine particles, thin films and exchange anisotropy," in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.

[7] R.H, Rahmans, "Title of paper with only first word capitalized," J. Name Stand. Abbrev., in press.

[8] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, “Electron spectroscopy studies on magneto-optical media and plastic substrate interface,” IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].

[9] M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.

[10] 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. 41st IEEE/ACM Int. Symp.Microarch. (MICRO), 2009, pp. 24–33.

BIOGRAPHY

SRIBARKI SRINATH received the B.Tech degree in Electronics and Communication engineering from the M.V.G.R College Of Engineering, Vizianagaram, Andhra Pradesh, in 2010 and he is currently pursuing M.Tech from Pydah College of Engineering and technology, Visakhapatnam, and Andhra Pradesh, INDIA.

His research interests include VLSI design especially error control for non-volatile and volatile memories and variation tolerant design techniques, Low power embedded systems design including memory optimization, high level synthesis and compilation, and VLSI architectures.

BANTUPALLI DIVAKAR received the B.Tech degree in the stream of Electronics and Communication engineering in the year 2010. Later he completed his M.tech degree in 2012 and present working as an Assistant Professor in Pydah College of Engineering and technology, Visakhapatnam, and Andhra Pradesh, INDIA.