ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 1, Issue 10, December 2012

ERROR DETECTION AND CORRECTION ENHANCED DECODING OF DIFFERENCE SET CODES FOR MEMORY APPLICATION

S.Baskar[1], M.Saravanan[2]

PG-Scholar, Electronics and communication, SNS College of technology, Coimbatore, India[1]

Assistant professor, Electronics and communication, SNS College of technology, Coimbatore, India[2]

ABSTRACT: **As technology scales, Multiple Cell Upsets (MCUs) become more common and affect a larger number of cells. In order to protect memories against MCUs as well as SEUs is to make use of advanced Error detecting and correcting codes that can correct more than one error per word. A sub-group of the low-density parity checks (LDPC) codes, which belongs to the family of the Majority logic decoding has been recently proposed for memory application and Difference set codes are one example of these codes which contributes for error detection and correction.ML decodable Codes are suitable for memory applications due to their capability to correct a large number of errors. In this paper, the proposed scheme for fault-detection and correction method significantly makes area overhead minimal and to reduce the decoding time through DC codes than the existing technique and it gives promising option for memory applications. HDL implementation and synthesis results are included, showing that the proposed techniques can be efficiently implemented.**

Keywords: Difference set codes, error correction codes, majority logic decoding, memory; multiple cell upsets (MCUs)

I. INTRODUCTION

RADIATION-INDUCED soft errors are a major issue for Memory reliability. To prevent soft errors from causing data corruption, memories are typically protected with error correction codes (ECCs). The most commonly used codes can correct one error and detect two errors per memory word and are known as single-error- correction double-error- detection (SEC-DED) codes. Their main advantages are that they require few additional bits per word and that the decoding process is simple. A SEC-DED code enforces a minimum distance of four between any two coded words By having a distance of four any word that suffers a double error would be in the worst case at a distance of two from any valid coded word. Therefore, it cannot be mistaken for a single error and miscorrected. The same approach is used for codes that can correct two errors; in this case, Double Error Correction Triple Error-Detection (DEC-TED) codes are used. However, this increases the decoder complexity substantially. In a hierarchical approach That combines a Hamming code and a Bose–Chaudhuri–Hocquenghem code was proposed to minimize the Latency. The use of Euclidean geometry (EG) codes has also been considered for memory Protection in particular EG codes studied are one-step Majority logic decodable and therefore, the decoders can be implemented with low cost. Other codes that are one-step majority logic decodable are difference-set (DS) codes. Their use for memory protection has also been studied recently showing that the properties of t he codes can be exploited to Reduce the decoding time significantly.

The combination of a simple decoder and a reduced decoding time makes DS codes an attractive option for memory protection. Among the ECC codes that meet the requirements of higher error correction capability and low decoding complexity, cyclic block codes have been identified as good candidates, due to their property of being majority logic (ML) decodable. A sub-group of the low-density parity check (LDPC) codes, which be-longs to the family of the ML decodable codes, has been re-searched in. In this paper, we will focus on one specific type of LDPC codes, namely the difference-set cyclic codes (DSCCs), which is widely used in the Japanese teletext system or FM multiplex broadcasting systems.

The remainder of this paper is organized as follows. Section II gives an overview of existing ML decoding

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 1, Issue 10, December 2012

solutions; Section III presents the Existing ML difference-set cyclic codes algorithm; Section IV Proposed two dimensional mod-2 addition algorithm Section V the results obtained for the Different versions in respect to effectiveness, performance, and area and power consumption. Finally, Section V discusses conclusions and gives an outlook onto future work

II. PROFILE ABOUT ML DECODING

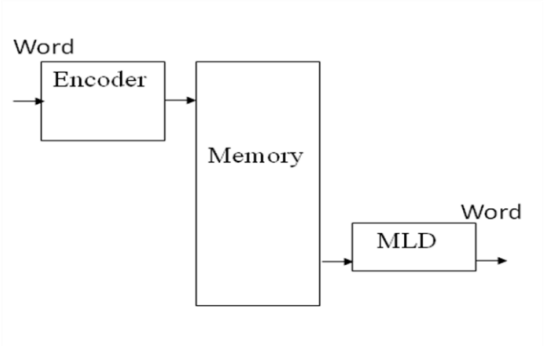

Fig. 1 Memory system schematic with MLD

MLD is based on a number of parity check equations which are orthogonal to each other, so that, at each iteration, each code-word bit only participates in one parity check equation, except the very first bit which contributes to all equations. For this reason, the majority result of these parity check equations decide the correctness of the current bit under decoding. Generic schematic of a memory system is depicted in Fig.1 for the usage of an ML decoder. In this method initially, the data words are encoded and then stored in the memory the resulting sums are then forwarded to the majority gate for evaluating its correctness. If the number of 1's received in is greater than the number of 0's, which would mean that the current bit under decoding is wrong, and a signal to correct it would be triggered. Otherwise, the bit under decoding would be correct and no extra operations would be needed on it and it increases decoder However, they require a large decoding time that impacts memory performance In order to improve the decoder performance, alternative de-signs may be used. One possibility is to add a fault detector by calculating the syndrome, so that only faulty codeword's are decoded. Since most of the codeword's will be error-free, no further correction will be needed,
therefore performance will not be affected. Although the implementation of an SFD reduces the

Average latency of the decoding process, it also adds complexity to the design (as shown in Fig. 2). The SFD is an XOR matrix that calculates the syndrome based on the parity check matrix. Each parity bit results in a syndrome equation. Therefore, the complexity of the syndrome calculator increases with the size of the code.

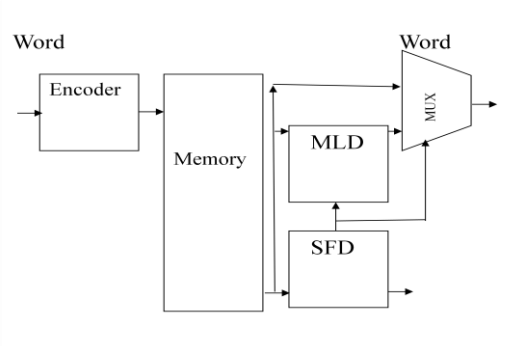

Fig. 2 schematic of an ML decoder with SFD

In order to reduce the size ML decoder that improves the designs presented before. Starting from the original design of the ML decoder introduced in ML detector/decoder (MLDD) has been implemented using the difference-set cyclic codes (DSCCs) this code is part of the LDPC codes (as shown in Fig.3)

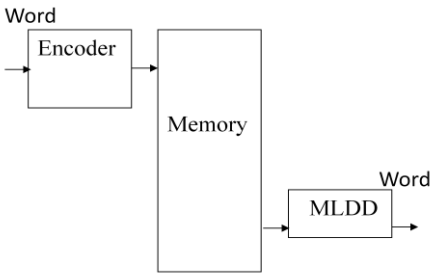

Fig. 3 system schematic of an MLDD

In all above methods though they are efficient in many attributes, there are some drawbacks present, it's due to adding extra bit for parity check equations which in turn gives extra overhead to the entire process. Those methods made use of two dimensional parity check equations for error detection and ML decoders for error correction and this makes a big impact on the performance of the system, depending on the size of the

Code. This explores the idea of using one dimensional Two dimensional mod-2 additions for error detection in order to reduce extra overhead and to improve the performance of the system and for codeword construction I made use of traditional hybrid code technique. And the corresponding DSCC lengths are described in Table.1.

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 1, Issue 10, December 2012

Table. 1 DSCC lengths

Methods	N	Data bits	Parity bits	Segments
1D parity	32	8	45	4
	64	8	90	8
	128	8	180	16
2D-Mod Parity	32	8	40	4
	64	8	72	8
	128	8	136	16

III. EXISTING 1D-ALGORITHM OF MLDD

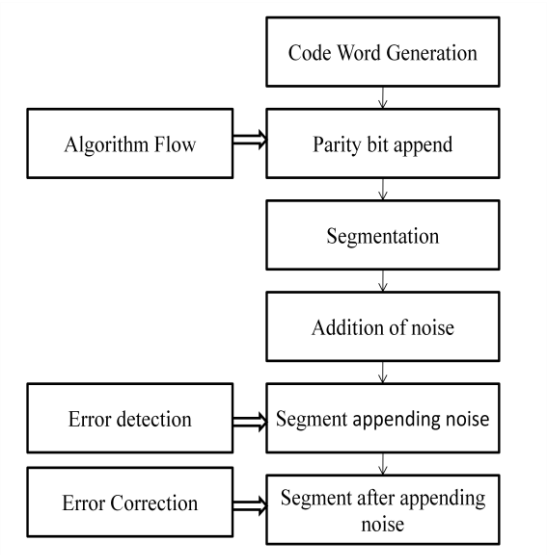

Fig. 4 Flowchart of the DC two dimensional parity flow

Earlier for the code construction it is noted that a generator matrix for a code with parity-check matrix H can be found by performing cyclic factor (shifting) on H to obtain it in the form

H =shifting factor
G= multiplicative factor

The matrix H is called a parity-check matrix. Each row of H corresponds to a parity-check equation and each column of H corresponds to a bit in the codeword. It

Has to be specified to satisfy the above condition to Matches the corresponding parameters of less non-zero bit accumulation of predefined tables.

Initialize H, G
If (H=Satisfied) then
H<=validation
End if
If (G=Satisfied) then
G<=validation
End if
Initialize
If (code=valid)
Then $c*H^t$ terms to zero else
$C*H^t$ terms to be error
End if
For: 1: N do
Parity generation
End for

Algorithm.1 Code construction and parity check equations

Thus for a binary code with m parity-check constraints and length n codeword's the parity-check matrix is an m× n binary matrix (refer Fig.4.). In matrix form a string C= [c1 c2 c3 c4 c5 c6] is a valid codeword for the code with parity-check matrix H if and only if it satisfies the matrix equation $CH^t = 0$.based on parity binary bits are appended with codeword before transmission (As shown in algorithm.1)

IV. PROPOSED TWO DIMENSIONAL BENES NETWORK

H could be found by performing Gauss-Jordan elimination on H to obtain it in the form

H = Data| unity

The generator matrix is then

G= unity| data

The matrix H is called a parity-check matrix.. The segments (S_t) are added using inverted arithmetic to get the sum and corresponding output is processed based on Two dimensional mod-2 addition, and then segmented Two dimensional mod-2 addition is sent along with data segments. All the received segments (S_r) has to be computed. (As shown in algorithm.2) The ML decoding algorithm is a hard-decision message-passing algorithm for LDPC codes. A binary (hard) decision about each received bit is made by the detector and this is passed to the decoder.

Initialize
Validation for code
c=process elements*g
Data segmentation
For: 1: N do
Segment generation
Finished
End for

Algorithm. 2 Code construction and two dimensional mod-2 addition

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 1, Issue 10, December 2012

A bit node sends a message declaring if it is a one or a zero, and each check node sends a message to each connected bit node, declaring what value the bit is based on the information available to the check node. The check node determines that its two dimensional mod-2 addition equation is satisfied if the modulo-2 sum of the incoming bit values is zero. In reality, a communication channel can be quite complex and a model becomes necessary to simplify calculations at decoder side. The model should closely approximate the complex communication channel. Maximum likelihood estimation is a method to determine these unknown parameters associated with the corresponding chosen models of the communication channel. (As shown in algorithm.3)

For i=1 to m do
$S_{t=}S_r$
End for
Repeat
For i=1 to m do
If ($S_{t=}S_r$) then
Finished else
($S_t \neq S_r$)
E_r belongs to S_s
ML E_r corresponds to S_s
Finished
End if
End for

Algorithm. 3 ML decoding

$$P(y \text{ received}| \text{ x send}) = (1-p)^{n-d}.p^d$$

Where d=the hamming distance between the received and the sent codeword's.

n= number of bit sent.
p= error probability of the BSC.
1-p = reliability of BSC.

V. RESULTS

A. *Memory*

The memory read access delay of the plain MLD is directly dependent on the code size, i.e., a code with length 72 needs 72 cycles, etc. Then, two extra cycles need to be added for I/O. On the other hand, the memory read access delay of the proposed MLDD is only dependent on the word error rate (WER). If there are more errors, then more words need to be fully decoded.

B.*Area*

The previous subsection showed that the performance of the proposed design Based on two dimensional MLDD is much faster than the plain MLD version, but slightly

Logic utilization	Existing method	Proposed method	Available resource
Number of Slices	24	13	2400
Number of Slice Flip Flops	3	3	4896
Number of 4 input LUTs	41	22	4896

Table. 2 Logic utilization of device

lower than the design with syndrome calculator (SFD).As mentioned several times, this is compensated with a clear savings in area. The conclusions on the area results are given as follows. The MLDD version has a very similar performance to SFD; however it requires a much lower area overhead, ranging from 10.16% to 0.4 Minimum input arrival time before clock: No path found and Maximum output required time after clock: 5.693ns besides Maximum combinational path delay: 10.381ns.

Thus the simulation result shows the change of segments with respect to noise added in the data. And it has to be analysed via ML decoding.

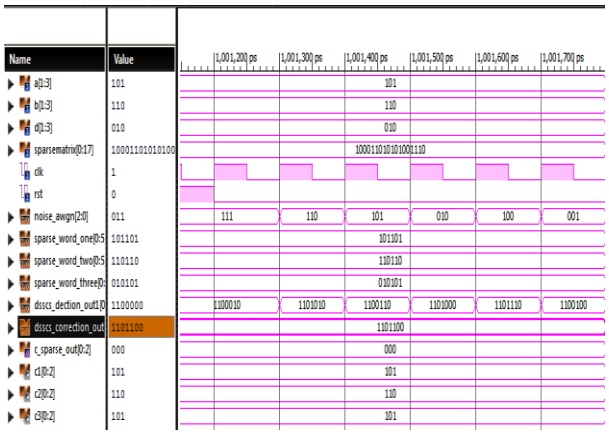

Fig.5.Simulation output of MLDD

VI. CONCLUSION

In this paper, an algorithmic scheme has been presented which effectively helps to correct errors caused by Multiple Cell Upsets (MCUs) as well as SEU in memories. The localization of the errors in an MCU has to analyse by placing data in the memory, thus providing additional error correction capabilities. Modified algorithmic methodology helps to correct burst errors than the existing method. Additionally, it helps to accelerate the decoding and effectively reduced the area and memory occupied by the present MLDD, than the previously proposed algorithms for DS codes. It has been applied to the scheme presented. Thus the results show

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 1, Issue 10, December 2012

that the method is also effective in reducing the decoding time, area, memory and delay when MCUs are present. The proposed scheme has been validated by simulation using a large number of error combinations and implemented to evaluate its cost in terms of circuit area and speed.

ACKNOWLEDGMENT

The authors would like to thank to Anna University and SNS groups of institutions for their Valuable support.

REFERENCES

[1]. I.S.Reed,"A class of multiple-error-correcting codes and the decoding Scheme*," IRE Trans. Inf. Theory"* , vol. IT-4, pp. 38–49, 1954.

[2]. J. L. Massey, Threshold Decoding. Cambridge, MA*: MIT Press*, 1963.

[3]. E.J.Weldon Jr., "Difference-set cyclic codes," *Bell System Tech*. J.vol. 45, pp. 1045–1055, 1966.

[4]. C.W.Slay man, "Cache and memory error detection, correction, and reduction techniques for terrestrial servers and workstations*," IEEE Trans. Device Mater. Reliable*". , vol. 5, no. 3, pp. 397–404, Sep. 2005.

[5]. H.Naeimi and A.DeHon, "Fault secure encoder an decoder for Nano-Memory applications*," IEEE Trans. Very Large Scale Integr.(VLSI) Syst."* , vol. 17, no. 4, pp. 473–486, Apr. 2009.

[6]. G.Torrens, B.Alorda, S.Barceló, J.L.Rosselló, S.A.Bota, and J.Segura, "Design hardening of nanometre SRAMs through transistor width modulation and combination*," IEEE Trans. Circuits Syst. II, Exp. Briefs"*, vol. 57, no. 4, pp. 280–284, Apr. 2010.

[7]. E.Ibe,H.Taniguchi,YYahagi,K.Shimbo,andT.Toba"Impact of scaling on neutron-induced soft error rate in SRAMs from a 250 nm to 22nm design rule*," IEEE Trans.Electron Devices*," vol. 57, no. 7, pp. 1527–1538, Jul. 2010.

[8]. S.Liu, P.Reviriego, J.A.Maestro, "Efficient Majority logic fault detection with difference-set codes f or memory applications," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 20, no. 1, pp. 148–156, Jan. 2012.

[9]. Multiple Cell Upset Correction in Memories Using Difference Set Codes by Pedro Reverie, Mark F. Flanagan, Shih-Fu Liu, and Juan Antonio Maestro, Jun 2012.

Biography

S.Baskar, pursuing final year Master of engineering in VLSI Design from S.N.S College of Technology, Coimbatore, India. He also worked in Ramakrishna College of engineering and technology as a Technical Trainer and Dell portables as a hardware Engineer for more than 2 years. He obtained his Bachelor of engineering in E.C.E from S.N.S college of Technology, Coimbatore, India in 2009. He has published various papers in National, International conferences and International Journal. His areas of interest is on Networking, Memory design, Coding theories, FPGA architecture design, Error detection and correction methodologies.

M.Saravanan, working as Assistant Professor in S.N.S College of Technology, Coimbatore, India. He obtained his Bachelor of engineering from Coimbatore Institute of Technology (CIT) in 2006 and masters of engineering in VLSI Design from Anna University of Technology, Coimbatore, India. He has more than 3 years of teaching and 3 years of Industrial experience. he has published more than 15 research papers in High Impact factor International Journal, National and International conferences. His areas of interest is on Networking, Neural design, Power Electronics, Design of semiconductor memories, Wireless Communication and Embedded system.

.