An Elementary Proposal on Fault Tolerant Devices for Memory Scenario

P.Lavanya , N.S.Murti Sarma and Ch.S.V.Maruthi Rao[1]
Sreenidhi Institute of Science and Technology,
Yamnampet, Medchal, Telangana District,501301-India
[1]Sreyas Institute of Engineering and Technology,
Bandlaguda, Nagole, Hyderabad.

Abstract :- The paper aims to propose as elementary work on a reliable memory system that can tolerate multiple transient errors in the memory words as well as multiple errors in the encoder and decoder (corrector) circuitry using one class of Error Correcting Codes i.e. type I 2-dimensional Euclidean Geometry Low-Density Parity-Check (EG-LDPC) codes and to quantify the importance of protecting encoder and corrector circuitry.

Keywords:*Memory system, Transient errors, decoder, Euclidean Geometry, Low Density Parity Check,VHDL 5.1, Xilink implementation.*

*****

1. Introduction

Memory cells have been protected from soft errors for more than a decade; due to the increase in soft error rate in logic circuits, the encoder and decoder circuitry around the memory blocks have become susceptible to soft errors as well and must also be protected. We introduce a new approach to design fault-tolerant encoder and decoder circuitry for memory systems using a new class of error-correcting codes whose redundancy makes the design of detectors particularly simple.

In recent years, the complexity of digital systems has increased dramatically. Although semiconductor manufacturers try to ensure that their products are reliable, it is almost impossible not to have faults somewhere in a system at any given time. As a result, providing reliability is becoming constantly more challenging due to increases in both the device failure rate and system complexity, and the reliability has become a topic of major concern to both system designers and users in particular in data storage applications where high reliability is required. Fault tolerance plays an important role in improving the reliability of the information stored in semiconductor memories. The reliability can be further improved by using more complex error-detection techniques. One potential approach is using error-correcting codes. Hamming codes are often used in memory systems to correct single error and detect double errors in any memory word. Traditionally, memory cells were the only susceptible part to transient faults. However, the supporting logic of memory system is also expected to be affected by transient faults as well. Consequently, developing fault-tolerant encoders, correctors, and detectors for memory system attracted considerable attentions recently. Almost all of the techniques use the conventional fault-tolerant schemes (e.g., parity prediction) to protect the encoder and corrector circuitry similar to other general purpose fault-tolerant schemes. In contrast the technique introduced in this work does not use the conventional fault-tolerant schemes.

A. Research Questions

This work is part of an undergraduate dissertation carried out at JNTUH, Hyderabad. This proposal aimed towards the implementation of a fault-tolerant memory system with fault-tolerant supporting circuitry. The proposed project improves the reliability of a fault-tolerant memory system further using one class of ECC's i.e. Euclidean geometry codes. This work is expected to answer the following research questions.

1. Understanding the concepts of fault-tolerance & different ECC's.
2. Implementing the fault-tolerant memory system using Hamming code for the error-correction capability comparison to the proposed system.
3. Implementing the fault-tolerant memory system using EG-LDPC codes without protecting supporting logic circuitry for the performance comparison to the proposed system.
4. Development of fault-tolerant memory system with protecting supporting logic.
5. Implementation of the proposed system and comparing with the existing system.
6. Implementation of on-run scrubbing technique.

B. Status of the work

To fulfill the objectives of the work, understanding the concepts of the reliability and fault tolerance is very

essential. The fundamentals of the fault tolerance & reliability are studied from [1, 2]. One more prerequisite to fulfill the objectives is to understand the use of error correcting codes, encoding and decoding procedures of a few well known error correcting codes like Block codes etc. These basics are presented in [3, 1] respectively. The problem of improving the reliability for memory applications has been attacked from several different angles in recent Literature. Novel approaches for implementing fault tolerant memory systems using error correcting codes were presented in [4]. These schemes were developed for improving the reliability of data storage systems. Fault tolerant memory systems with fault tolerant supporting logic circuitry are presented in [5, 16]. One class of error correcting codes used in our proposed system i.e. type I 2-dimensional Euclidean geometry codes which offer higher minimum distances is described in [11,15]. A novel Technique for constructing these codes is presented in [11, 2]. An approach called "cyclic to systematic matrix conversion for reducing the complexity of decoder" is presented in [2,7]. A technique called "one-step majority logic decoding" for decoding Euclidean geometry codes is presented in [13, 6, 8]. For implementing the idea of scrubbing technique [9, 10, 11,17], is referred.

C. Research Methodology

Huge data system applications require storage of large volumes of data set, and the number of such applications is constantly increasing as the use of computers extends to new disciplines. At the same time, the error rate in semiconductor memories is increasing with time. Furthermore, combinational logic has already started showing susceptibility to soft errors and therefore the encoder and decoder (corrector) units will no longer be immune from the transient faults; therefore, protecting the memory system support logic is even more important. Here, a fault tolerant memory system that tolerates multiple transient errors in each memory word as well as multiple errors in the encoder and corrector units is proposed. The proposed work is implemented using VHDL language and simulated on Active HDL 5.1 for its functional verification[18]. The implemented design is synthesized using Xilinx 10.1 V implementation tool.

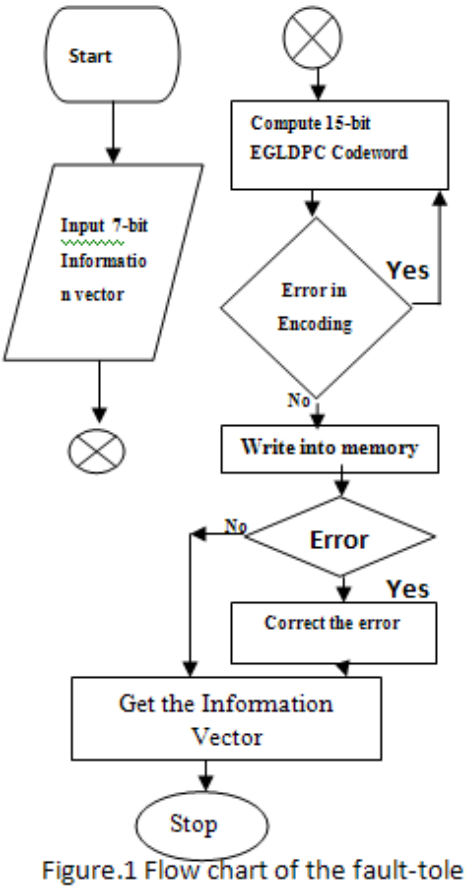

Figure.1 Flow chart of the fault-tolerant memory system

2. Results and discussion

The Fig. 1 shows the flow chart of the proposed fault-tolerant memory system with fault-tolerant supporting logic i.e. encoder and corrector. It provides the direction of flow that makes the writing code easy. The same steps can be put in algorithmic form also. Note that the on-run scrubbing technique is not shown in the figure. The figure emphasizes the function of encoder-detector and corrector-detectors only.

Table 1: Dataset

1 0 1 0 0 0 1
1 0 1 0 1 0 1
1 1 1 0 1 1 0
0 1 0 1 0 1 0
1 1 0 0 0 1 1

A. Input Consideration

The proposed design is tested considering a memory of data, each of 7-bit long as given in Tab.1.

Table.2. Original And Encoded (Using Eg-Ldpc Code) Data Set

S.No.	Data in Binary form	Hexa-Decimal notation	Encoded Data in Binary form using EG-LDPC code	Hexa-Decimal notation
1	1010001	51	1 0 1 0 0 0 1 1 1 1 1 1 0 1 1	51FB
2	1010101	55	1 0 1 0 1 0 1 1 0 1 0 0 1 1 1	55A7
3	1110110	76	1 1 1 0 1 1 0 0 1 0 1 0 0 0 0	7650
4	0101010	2A	0 1 0 1 0 1 0 0 1 0 1 1 0 0 0	2A58
5	1100011	63	1 1 0 0 0 1 1 0 1 1 1 1 1 0 0	637C

This section illustrates the statistical encoding for the given set of data. To encode the given data, it is multiplied by 7x15 generator matrix, note that it is vector-matrix multiplication. The length of the code word is fixed and it is equal to 15-bit in our case.

Table.3. Original And Encoded (Using Hamming Code) Data

S.No.	Data in Binary form	Hexa-Decimal notation	Encoded Data in Binary form using Hamming code	Hexa-Decimal notation
1	1 0 1 0 0 0 1	51	0 1 1 1 0 1 0 1 0 0 1 0	752

Tab.2 shows the data set both in binary and hexa-decimal form and the corresponding encoded vectors are also shown in both binary and hexa-decimal form. Whereas the Tab.3 shows the one element of the data set encoded using Hamming code. Here also the length of the codeword is fixed and is 12-bit long.

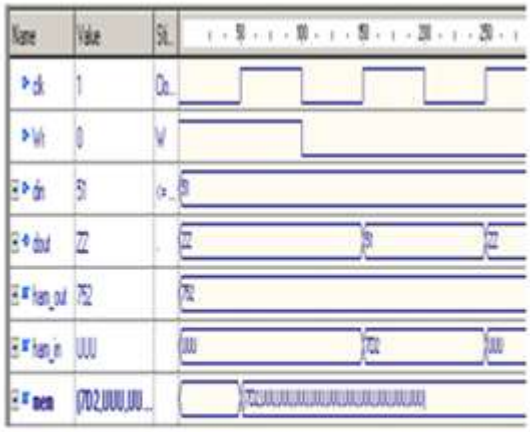

Figure2. Simulation waveform for fault-tolerant system using Hamming code with single fault in memory

Following simulation results show the data encoding and decoding with and without faults with Hamming and EG-LDPC codes for comparison of error correction capability of Hamming and EG-LDPC codes. And, finally the simulation results of proposed system are shown.

C. Simulation Results For Fault-Tolerant Memory Using Hamming Code

The simulation results for the implemented fault–tolerant design using Hamming code with single and double faults in memory unit Fig..2 and.3. In the simulation result shown above signal ‘dout’ indicates output generated at the output of the hamming code top module. Signal ‘din’ shows the unique information vector passed to the hamming code generator unit and signal ‘ham_out’ shows the corresponding codeword for the concerned unique word.

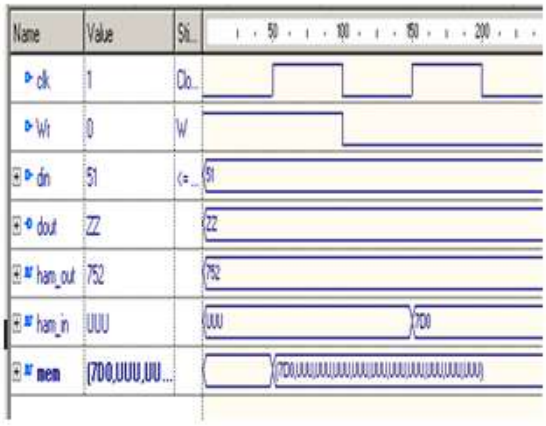

Figure.3. Simulation waveform for fault-tolerant system using Hamming code with double faults in memory

 ISSN: 2321-8169
56 – 61

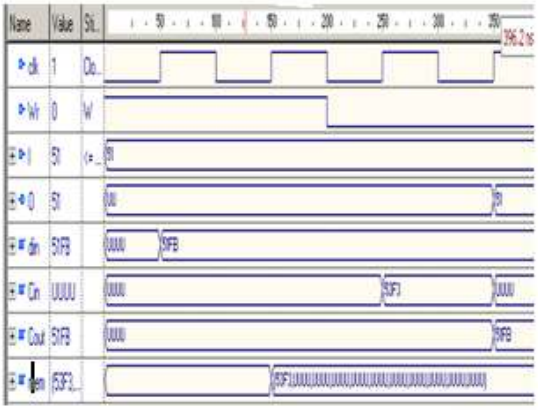

Figure.4. Fault-tolerant memory with fault free encoder and corrector but the memory unit with double faults.

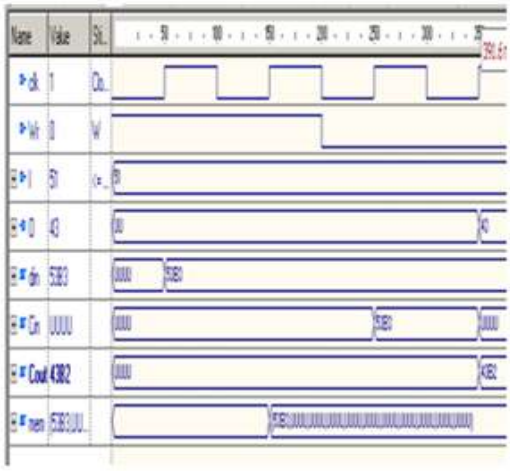

Figure.5. Fault-tolerant memory with faulty encoder.

Signal 'ham_in' shows the input which is the output from memory unit for hamming code corrector module. Signal 'clk' indicate the system clock passed to the designed module for simulation. Signal 'Wr' stands for the write signal used for controlling the read/write operations of the memory. Signal 'mem' shows the memory location for storing the encoded data set. Note that the output from the top module is the corrected correct data in single fault case shown in Fg.2. Whereas from the Fig.3 it is shown that the output of the top module is at its high impedance state as the corrector is unable to correct the double faults.

D. Simulation Results For Fault-Tolerant Memory System Without Protecting Supporting Logic

The simulation results for the implemented fault–tolerant design using EG-LDPC code with double faults, triple faults, and double faults in memory unit, encoder unit and corrector unit respectively. In the simulation result shown above signal 'dout' indicates output of the corrector module appears in Fig. 4, Fig.5 and Fig.6. Signal 'I' shows the unique information vector passed to the EG-LDPC encoder unit and signal 'Cin' shows the corresponding codeword for the concerned unique word i.e. output from the encoder.

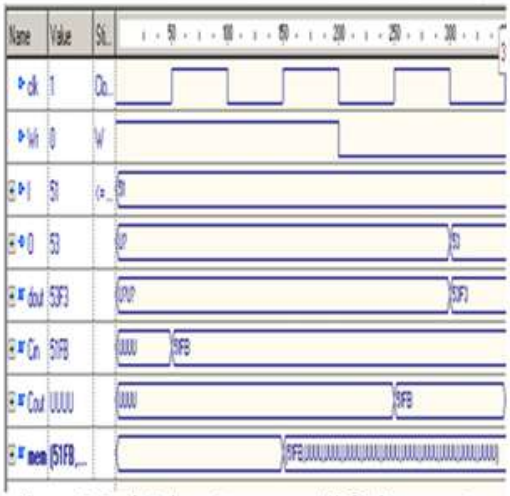

Figure.6. Fault-tolerant memory with faulty corrector.

Signal 'Cout' shows the input which is the output from memory unit for corrector module. Signal 'O' indicate the output from the fault-tolerant memory system with protection free encoder and decoder circuitry. Signal 'clk' indicate the system clock passed to the designed module for simulation. Signal 'Wr' stands for the write signal used for controlling the read/write operations of the memory. Signal 'mem' shows the memory location for storing the encoded data set. Note that the output from the top module is the corrected correct data in case of double faults in memory unit shown in Fig 4. Whereas from the Fig. .5 and Fig..6 it is shown that the output of the top module is wrong when there are errors in encoder and corrector units respectively though the memory unit is fault free.

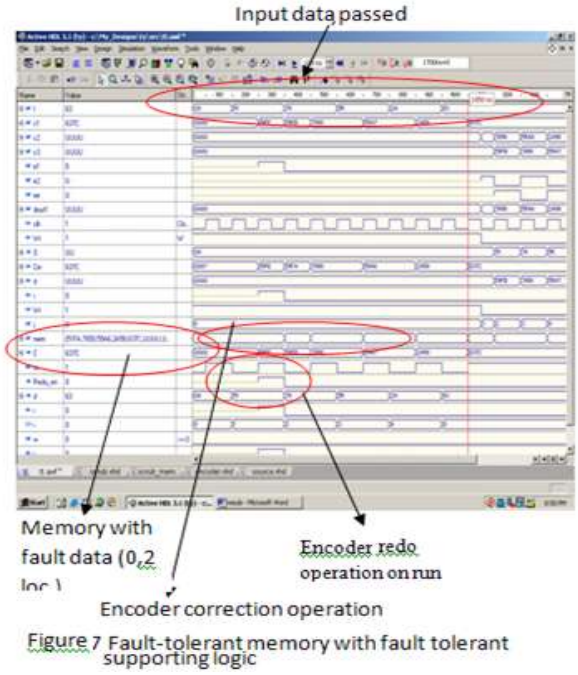

Figure 7 Fault-tolerant memory with fault tolerant supporting logic

E.Simulation Results For Fault-Tolerant Memory System With Fault-Tolerant Supporting Logic ***At Write and read operations***

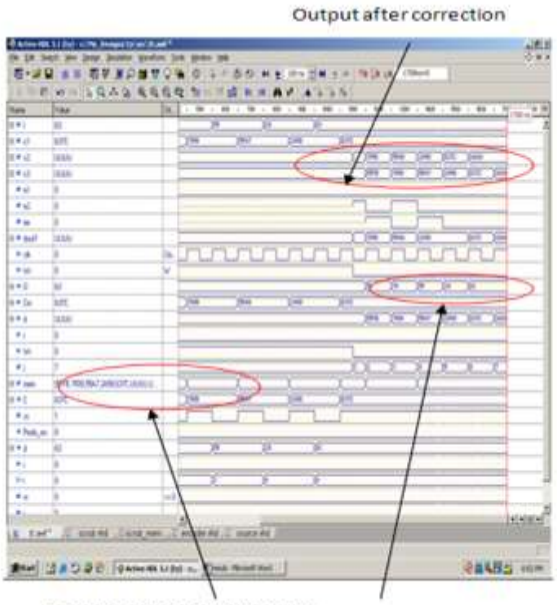

Figure.8. Fault-tolerant memory with fault tolerant supporting logic

Figures 7 and.8 show the simulation results for the implemented fault–tolerant design with fault-tolerant encoder and decoder (corrector) circuitry using EG-LDPC code. In the simulation result shown above signal 'c3' indicates output of the corrector module. Signal 'I' shows the unique information vector passed to the EG-LDPC encoder unit and signal 'C' shows the corresponding codeword for the concerned unique word i.e. output from the encoder. Signal 'c2' shows the input which is the output from memory unit for corrector module. Signal 'O' indicate the output from the fault-tolerant memory system with protected encoder and decoder circuitry. Signal 'Redo_en' indicate the output of the encoder-detector circuit, it goes high only when there is error in the encoded codeword. Signal 'clk' indicate the system clock passed to the designed module for simulation. Signal 'Wr' stands for the write signal used for controlling the read/write operations of the memory. Signal 'mem' shows the memory location for storing the encoded data set.

Note that in the Fig. 7, the purpose of the encoder-detector is shown clearly. When there is a fault in encoder, it produces an incorrect codeword. Here that is dectected by the encoder-detector circuit and a control signal generated by it, is sent back to the encoder to redo the previous encoding. The redo operation and the corrected encoded data is shown in the fig. 7 with red color circle.

In the Fig.8, the importance has been given to the technique called 'on-run scrubbing'. The memory unit is written by faulty data at some memory locations. When that faulty data are read by corrector-detector, it generates a high control signal for the one-step majority corrector to initiate the serial correction. The corrected output data are sent to both the control logic circuit which gives the output to external world and memory unit to replace the faulty codeword by corrected codeword. As the control signal from the corrector circuit high the corrected codeword is written back into memory. This is what called 'on-run' scrubbing.

3. Conclusions

Huge data system applications require storage of large volumes of data set, and the number of such applications is constantly increasing as the use of computers extends to new disciplines. At the same time, the error rate in semiconductor memories is increasing with time. Furthermore, combinational logic has already started showing susceptibility to soft errors and therefore the encoder and decoder (corrector) units will no longer be immune from the transient faults; therefore, protecting the memory system support logic is even more important.

Here an efficient and simple method for protecting the supporting logic is proposed. It is observed that the error correcting codes with higher minimum distances can correct more errors i.e. as higher the minimum distance higher the error correction capability. The proposed system uses the one class of error correcting codes which has higher minimum distances and employs easier hardware implementation.

The paper implements fault tolerant memory system with fault tolerant supporting logic for improving the reliability of memory system. From the obtained simulation results it can been very clearly seen that the proposed system increases the reliability of memory systems by correcting multiple transient errors not only in the memory unit but also in the supporting logic circuitry i.e. encoder and decoder (corrector), and the comparative results of fault-tolerant memory system using Hamming code and fault-tolerant memory system using EG-LDPC code shown that how the EG-LDPC codes are superior to Hamming codes with respect to error correction capability. And, finally it can be observed form the simulation results that the proposed on run scrubbing technique improves the reliability of the system further. The existing memory systems without preparation for protecting supporting logic cannot tolerate if any errors are there in either of the units i.e. encoder and decoder. The implemented system can tolerate for more number of errors than the existing system. From these observations it can be concluded that the proposed system can give higher reliability compared to the existing fault-tolerant system.

As, the content are research questions framed are elementary basis no solid findings were expected after completion of the work. To the level of the first author work is after the graduation, Novelty lies implementation of the base literature thrived itsef.

A. Recommondations

The proposal is expemplified with system developed that was verified for its functionality. From the result its found that the reliability of the proposed system is higher than the existing fault tolerant memory systems. Also its presumed that this may be further extended to improve the reliability and to reduce the area overhead for the implemented design. The implemented system is fully synthesized targeting FPGA (xcv1600e-7-fg1156). And can be targeted on to a FPGA for real time application.

Acknowledgements

Dr.Asha Rani, Professor of ECE, deserves high degree acknowledgement as one of the assessee committee member of the proposal along with the First author. Dr.P.Narasimha Reddy, Director of Sreenidhi Institute of science and Technology, Yamnampet of Medchal, to be honoured to the maximum degree of his co-operation and encouragement. Last, but not least Dr.Suresh, Akella, Principal of Sreyas Institute of engineering and technology deserves due acknowledgements for his esteemed interst in the encouraging the publication.

References

[1] Parag K. Lala. “Fault Tolerant and Fault Testable Hardware Design”. PrinticHall,1985,ACM digtal library, 2017.

[2] C. M. Krishna. & Kang G. Shin, “Real-Time Systems”, McGRAW-Hill International Edition, 1997, ACM digtal library, 2017.

[3] K. Sam Shanmugam, “Digital and Analog ommuni- cation Systems”. John Wiley Sons. 2008.

[4] http://resolver.caltech.edu/CaltechETD:etd,0124200812650 -- for reliable systems with nanoscale devices., Last accessed in Jan2017.

[5] Helia Naeimi & Andre DeHon. “Fault Secure Encoder and Decoder for Memory Applications”. IEEE International Symposium on Defect & Fault Tolerance in VLSI Systems (DFTS 2007), Sept. 26-28, 2007.

[6] Helia Naeimi & Andre DeHon. “Fault Secure Encoder and Decoder for NanoMemory Applications”. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol. 17, No. 4, April, 2009.

[7] Shu Lin and Daniel J. Costello. “Error Control Coding”.Prentice Hall, second edition, 2004.

[8] R. J. McEliece. “The Theory of Information and Coding”. Cambridge University Press, 2002.

[9] S. Lin and D. J. Costello, Jr., “Error Control Coding: Fundamentals and Applications”, Prentice Hall, Englewood Cliffs, New Jersey, 2004.

[10] A.V.Balakrishanan,,”Advances in communication systems”, Academic Press, U.K,2014.

[11] A. Saleh, J. Serrano, and J. Patel, “Reliability of Scrubbing Recovery-Techniques for Memory Systems,” IEEE Transaction on Reliability, vol. 39, no. 1, pp. 114–122, 1996.

[12] D. E. Knuth, The Art of Computer Programming, 2nd ed. Addison Wesley, 2000.

[13] Y. Kou, S. Lin, and M .P. C. Fossorier, “Low Density Parity Check Codes Based on Finite Geometries: A Rediscovery and New Results,” IEEE Trans. Information Theory, Nov. 2001.

[14] Heng Tang et al. “Codes on finite geometries”. IEEE Transaction on Information Theory, 51(2):572- 596, 2005.

[15] J. Bhaskar., “A VHDL Primer” Addision Wesly, 1999.

[16] G. C. Cardarilli, “Concurrent Error Detection in Reed-Solomon Encoders and coders,” IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 15, pp. 842–826, 2007.

[17] Ying Wang, Yinhe Han, Huawei Li, Xiaowei Li, "VANUCA: Enabling Near-Threshold Voltage Operation in Large-Capacity Cache", Very Large Scale Integration (VLSI) Systems IEEE Transactions, vol. 24, pp. 858-870, 2016.

[18] Fernanda Kastensmidt, Paolo Rech, “Soft errors and fault tolerance design”, Springer international, 2015.