ISSN: 2278 – 909X
International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE)
Volume 7, Issue 4, April 2018

COMPARATIVE ANALYSIS OF ERROR CORRECTION TECHNIQUES FOR INCREASING RELIABILITY OF EMBEDDED MEMORIES

Indulakshmi.V [(1)],Prajith Prakash Nair[(2)], Anitha.A[(3)],S.Mohan[(4)]
(1) P.G Scholar
(2) (3)(4) Assistant professor /ECE Department
Nehru Institute Of Technology,Coimbatore-641105

***Abstract* - Among the different components in a system memories are one of the parts most sensitive to soft errors. Not only soft errors but also permanent errors effect memories. Error correction code(ECC) and Built in self repair(BISR) have been widely used for improving the reliability of embedded memories. We propose an ECC-enhanced BISR technique which uses ECC to repair single permanent faults first and spares for the remaining faults. Techniques are proposed to maintain reliability in the production stage as well as the online field stage. The error correction coding mechanisms proposed here are the SEC-DED method and the DEC method. The experiment results shows the comparison of both ECC mechanisms and also the proposed scheme can improve the reliability of embedded memories.**

***IndexTerms*: Built In Self Repair(BISR), reliability,ErrorCorrectionCode(ECC),Permanet faults, Soft Errors**

I.INTRODUCTION

The probability of memory faults has increased as the capacity and density of memories have rapidly increased with the technological development of semiconductor manufacturing, This causes reliability drops and quality degradation. There are many proposed fault-tolerance techniques used to boost the reliability of the embedded memory. In the past, hard repair techniques [1-9] by using redundancies are widely used for repairing permanent faulty memory cells. That is, spare rows (SRs) and spare columns (SCs) are used to replace faulty memory cells. Error correction code (ECC) and built-in self repair (BISR) techniques[1] by using redundancies have been widely used for improving the reliability of embedded memories. The target faults of these two schemes are soft errors[11] and permanent (hard) faults, respectively. There are many online concurrent repair techniques for protecting memories from soft errors. The most popularly used technique is by using the error correction codes (ECCs). We can add additional parity bits to the information bits to detect and correct the faulty bits in a codeword, which consists of the check bits and the information bits.

Hamming code and Hsiao code can correct single errors and detect double ones in a given codeword, known as the single-error-correction double-error-detection (SEC-DED)codes. For correcting multiple cell faults[1] the Bose–Chaudhuri–Hocquenghen and Reed–Solomon codes can be used. Conventionally, the soft errors and permanent faults are considered separately. For today's advanced memories, we can deck out them with both ECC and BISR. If the protection capabilities of these two schemes are properly integrated, the reliability could be improved further. This EBISR technique mainly consists of two stages Production stage and Field test and repair stage. The Production stage uses the SEC-DED code or DEC[1] code to repair permanent faults first .The main target is the permanent faults. During the Field and Repair stage, the main targets are soft errors and unintentional permanent faults due to aging and process variations.

Thereafter, the second fault/error can be corrected by the fault identification and correction (FIC) phase in this stage. Based on the first and second read out code words, EBISR [2-7]can discriminate the fault categories (permanent faults or soft errors) and then correct the second faults .Here in ECC we are using two types of error correcting

codes Hamming code and Reed Solomon code[1]. Hamming code can identify double errors and correct single cell fault where as Reed Solomon code can correct double errors. Further set of errors can be handled by spares. Here we are doing a comparison between these two types of error correction codes along with EBISR error correction technique

. The rest of this paper is organized as follows. The basic Production stage and Field and repair stages are given in Section II and sectionIII respectively . Experiment Results are given in section IV. conclusions are given in Section V.

II.PRODUCTION STAGE OF EBISR

The test and repair flow of the Production stage is shown in Fig. 1.1 and Fig 1.2. It mainly consists of the fault detection phase, the ECC repair phase, and the hard repair phase. In the fault detection/identification phase, the BIST module executes the adopted March test algorithms[7] to test the memory array. If no faults are detected, the memories can be shipped or used for normal access. If there are any faults detected, the BIST operations will be suspended, and the fault information including the fault addresses and the fault syndromes, should be stored.The fault syndromes[13] can be used to identify if the faulty words contain SCFs or multiple cell faults. After storing the fault information, the BIST operations are resumed to perform the remaining tests. When the entire memory is completely tested, we then enter the ECC repair phase. The ECC repair phase is of two types, the SEC-DEC ECC [3-5]phase and DEC ECC phase. In order to do the comparison process we have to pass through both the ECC phases.

The SEC-DEC ECC code is used to repair all permanent SCFs. Since the SCFs occupy the largest proportion of all the possible fault types, correcting them by ECC can increase the efficiency of spare usage. The DEC ECC code is used to repair burst faults and also it can correct multiple errors or double errors at a time and the rest is for spares and so the reliability can be increased.

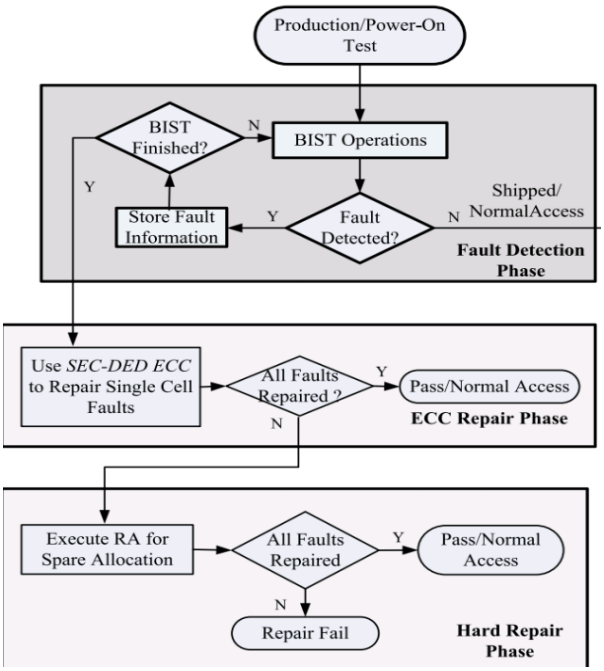

Fig 1.1 Test and repair flow of the production stage using SEC-DED ECC

The test and repair flow of the production stage includes SEC-DED ECC and DEC ECC .The memory to be tested ,should pass through both the phase and results are then compared. Only the ECC method is different, the rest of the procedure is same for both. Fig 1.1 and Fig 1.2 elucidates the procedure in detail.

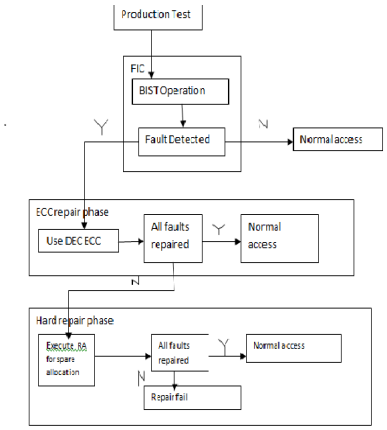

Fig 1.2 Test and repair flow of Production stage using DEC ECC

In the case of SEC-DED ECC phase all the permanent faults array are SCF, then the memory array can be repaired successfully without using redundancies. The memory can be shipped for normal access. If there are other types of faults (e.g., FRs, FCs, and cluster faults), we should enter the hard repair phase to deal with these faults. According to the analysis results, SRs/SCs can be used to replace FRs/FCs.

The addresses of FRs/FCs are stored for address remapping. If the incorporated redundancies are sufficient to repair all FRs/FCs, the memory array [6] can be repaired successfully. That is, the memory array[8] can be shipped or subjected to normal access. However, if all spares are used and there are still faulty memory rows/columns, the memory cannot then be repaired successfully and should be discarded directly. Since most of the SCFs are repaired.In the case of DEC ECC phase ,If the faults are multiple cell faults for eg (FR, FC,,burst faults, cluster faults) .The double errors or we can say two errors are identified and corrected simultaneously and if errors still exist, then enter the hard repair phase and follows the same procedure of the previous one.

III.FIELD AND REPAIR STAGE

The test and repair flow of the Field and Repair stage is shown in Fig. 2. It includes the concurrent fault detection (CFD)phase, the fault deactivation (FD) phase, and the FDC phase. In the CFD phase, single errors[14] (either permanent faults or soft errors) can be corrected, and double errors can be detected by the inherent correction/detection feature of the SEC-DEC ECC. However, if the ECC detects double errors, we should enter the FD phase[15]. In this phase, we complement the erroneous codeword and write it into the memory array again. Thereafter, we read the complemented codeword again and compare it with the original erroneous codeword. Based on the results of the comparison operation, we then enter the FDC phase. In this phase, the error types are identified first. If the readout complemented codeword still contains double errors, we can conclude that there are two permanent faults. Based on the comparison results, we can just complement the two bits, which are identical in the original erroneous codeword and the complemented codeword to correct the double errors. The DEC ECC method can be adopted in the Field and Repair stage also with slight changes in the previous mechanism.

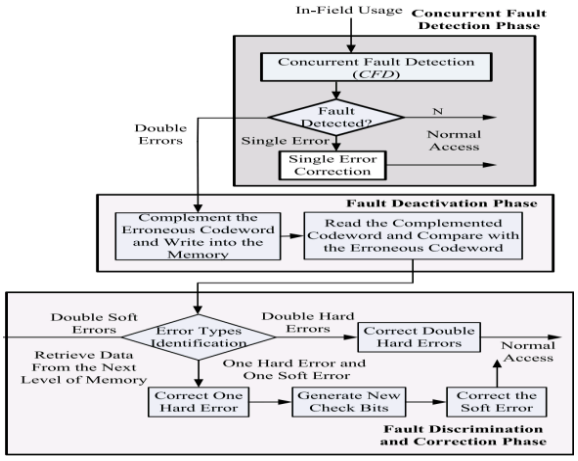

Fig 2 Test and repair flow of Field and Repair stage.

IV.EXPERIMENTAL RESULTS

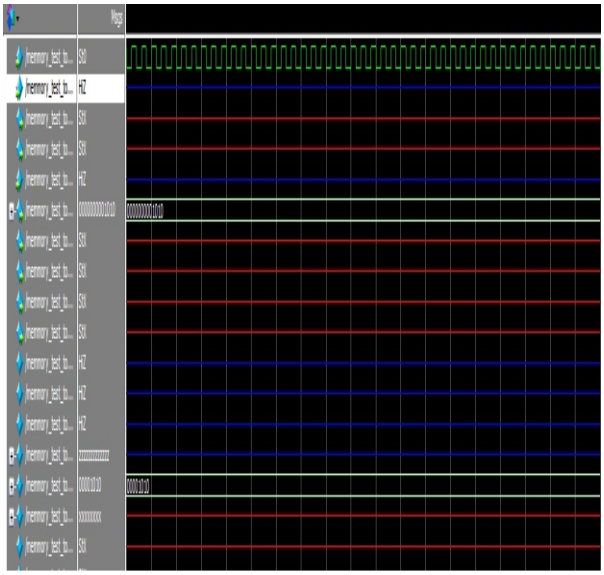

Fig4.1:Memory without error

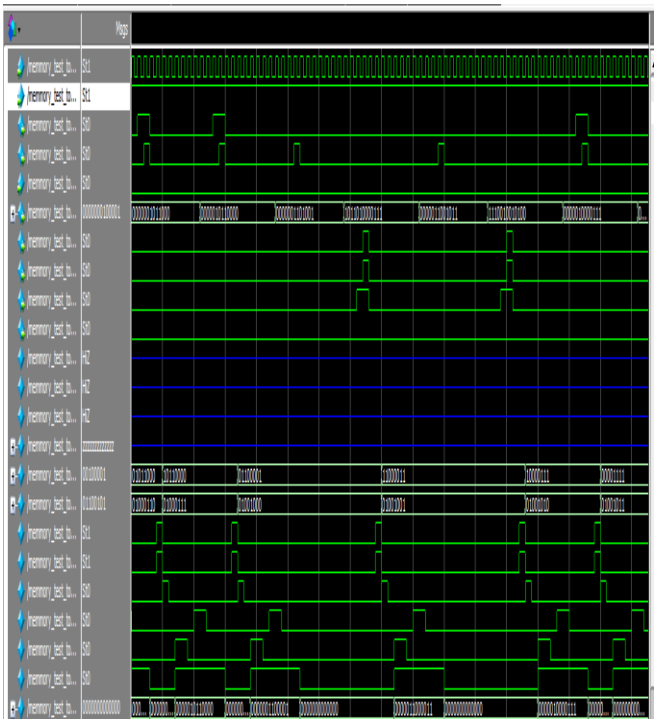

Fig 4.2: Memory with single error corrected

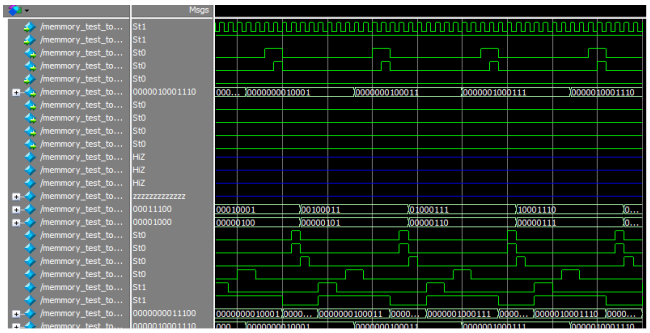

Fig 4.3:Memory with multiple error corrected

Fig 4.4: working of syndrome, example for Field and Repair stage

```
# R
# E
# A
# D
# K
# B
# D
# B
# A
# 1
# Break in Module field_tb at C:/Users/user/Desk
/SIM 4> quit -sim
ModelSim> vsim -gui -novopt work.field_tb
# vsim -gui -novopt work.field_tb
# Loading work.field_tb
# Loading work.memmory_field_top
# Loading work.remap_block
# Loading work.memmory_fault2
# Loading work.controller
# Loading work.parity_code
# Loading work.hamming_code
# Loading work.hamming_correct
# Loading work.parity_checker
# Loading work.UART_TX
# Loading work.UART_RX
# Loading work.BAUD_GEN
/SIM 6> run -all
# R
# E
# A
# D
# K
# B
# D
# B
# A
```

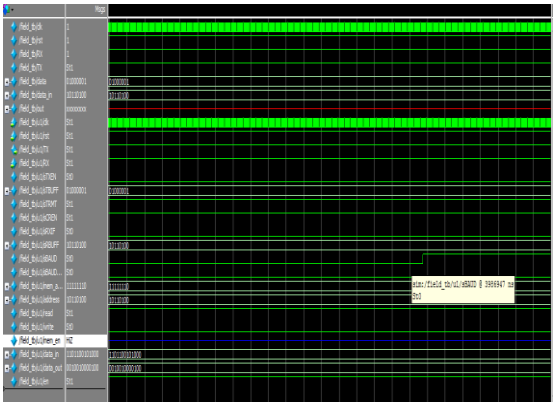

Fig4.5 : Simulation result of Field and Repair stage

Fig 4.6 :Simulation Result of EBISR using ReedSolomon ECC

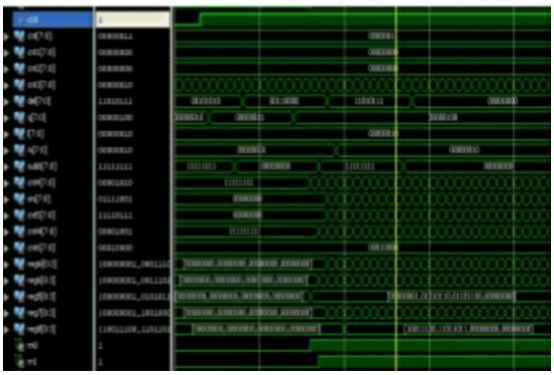

Fig4.7 Syndrome output using Reed Solomon

V.CONCLUSION

EBISR technique is used, integration of ECC and hard repair technique. Two types of ECC are used, Hamming code and Reed Solomon code. Hamming code can detect double error and correct single error. Reed Solomon code can detect and correct double errors. This technique is used in the Production stage and Field and Repair stage. The proposed technique can be easily integrated into the conventional BISR architecture.

REFERENCES

[1] Shyue Kung Lu. and Cheng J u Tsai. (2016)‘ Enhanced Built-In Self-Repair Techniques for improving Fabrication Yield and Reliability of Embedded Memories’IEEE Trans on very large scale integration systems VOL. 24, pp 795-800.

[2] W. Kang. and H. Cho, J. Lee and S.Kang.(2014) ‘A BIRA for memories with an optimal repair rate using spare memories for area reduction,’ IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 22, no. 11,pp. 2336–234

[3] C. Argyrides, P. Reviriego, and J. A. Maestro,(2013). ‘Using single error correction codes to protect against isolated defects and soft errors’ IEEE Trans. Rel., vol. 62, no. 1, pp. 238–243.

[4] Shyue-Kung Lu. Zhen Yu Wang, Yi Ming Tsai. and Jiann Liang Chen (2012)‘Efficient Built-In Self-Repair Techniques for Multiple Repairable Embedded RAMs’. IEEE Transactions On Computer-Aided Design Of Integrated Circuits and Systems, VOL. 31, NO 4.

[5] T.H. Wu (2012) ‘A memory yield improvement scheme combining built-in self-repair and error correction codes’ in Proc. IEEE Int. Test Conf. (ITC) pp. 1–9.

[6] B.Y. Lin and M. Lee, and C.W. Wu.(2012),‘A memory failure pattern analyzer for memory diagnosis and repair’ in Proc. IEEE 30th VLSI Test Symp. (VTS), pp. 234–239.

[7] D. Rossi. and N. Timoncini, M. Spica. and C. Metra.(2011), ‘Error correcting code analysis for cache memory high reliability and performance,’ in Proc. Design, Autom. Test Eur. Conf. Exhibit.pp. 1–6.

[8] J.-A. Maestro and P. Reviriego, (2009),‘Reliability of single-error correction

protected memories' IEEE Trans. Rel., vol. 58, no. 1, pp. 193–201.

[9] L.-M. Denq and Y.T. Hsing, and C.W. Wu. (2009), 'Hybrid BIST scheme for multiple heterogeneous embedded memories' IEEE Des. Test Comput.,vol. 26, no. 2, pp. 64–73.

[10] S.K. Lu. And C.H. Hsu, Y.C. Tsai, K.-H. Wang. and C.W. Wu. (2006), 'Efficient built-in redundancy analysis for embedded memories with 2-D redundancy' IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 14, no. 1,pp. 34–42.

[11] R. Baumann. (2005), 'Soft errors in advancedcomputersystems,'IEEEDes.Test Comput., vol. 22, no. 3, pp. 258–266.

[12] N. Derhacobian. and V. A. Vardanian. and Y. Zorian. (2004),'Embedded memory reliability: The SER challenge.' in Proc. IEEE Int. Workshop Memory Technol., Design Test. (MTDT). pp. 104–110.

[13] X. Du, S. M. Reddy.and W.T. Cheng, J. Rayhawk and N. Mukherjee. (2004), 'At-speed built-in self-repair analyzer for embedded word-oriented memories' Proc. IEEE Int. Conf. VLSI Design, pp. 895–900.

[14] Satoru Tanoi. and S. Tanoi, J. Dreibelbis. (1998), 'BIST: Required for Embedded DRAM' ISSCC 98 Digest., pp. 74-75.

[15] Robert Treur and Vinod.K.Agarwal. (1993), 'Built In Self Diagnosis for Repairable Embedded RAM.' Doctoral thesis, Dept. EE, McGill University. Vol. 42, No.5, pp.421-425.