Survey on Error Detection and Correction Schemes for Memory Applications

Kavya B S
Student(M.tech)
VLSI design and Embedded system
Sapthagiri college of engineering
Bangalore
Kavybs_sushma@yahoo.in

Mrs.B.N.Shobha
Associate professor
Dept. of ECE
Sapthagiri college of engineering
Bangalore
b_n_shobha67@yahoo.com

Abstract-Memory is responsible for digital circuit for storing as well as retrieving any digital data that are needed at particular time. Encoding and Decoding are the two basic operations that are responsible for reading and writing. Due to environmental interference and physical defects in the communication medium can cause random bit errors during data transmission from the source to a receiver. Also due to technology scaling and higher integration densities there are variations in parameters and noise levels which will lead to larger error rates at various levels of the computations. Therefore error detection techniques allow detecting such errors, while error correction enables reconstruction of the original data in many cases. This paper describes error detection/correction mechanisms that can be utilized within system to protect applications against various types of bit errors. These detection/correction mechanisms have different overhead costs in terms of energy, performance, and area, and also differ in their error coverage, complexity, and programmer effort. In order to achieve the highest efficiency in designing and running a computer system, to understand the trade-offs among the aforementioned metrics for each detection/correction mechanism and choose the most efficient option for a given running environment.To accomplish such a goal, this paper enumerate many error detection/correction techniques that enable reliable delivery of digital data over unreliable communication channels

IndexTerms- Embedded memorys, Types of errors, error detection/correction techniques**.**

I. INTRODUCTION

Memories are the most important component which have been protected from soft errors. Types of embedded memory such as ROM, SRAM, DRAM, flash memory etc are seen in all system chips. The memory failure rates are increasing due to the impact of technology scaling, smaller dimensions, high integration densities, lower operating voltages, etc. A soft error occurs when a radiation event causes enough of a charge disturbance to reverse or flip the data state of a memory cell, register, latch, or flip-flop. The error is "soft" because the circuit/device itself is not permanently damaged by the radiation. If new data are written to the bit, the device will store it correctly. The soft error is also often referred to as a single event upset. If the radiation event is of a very high energy, more than a single bit will affected, creating a multibit upset as opposed to the single bit [1], [2].

- *Kavya B S is currently pursuing M.Tech in VLSI DESIGN AND EMBEDDED SYSTEM in VISVESVARAYA TECHNOLOGICAL UNIVERSITY, India, PH-9986098618. E-mail:kavyabs_sushma@yahoo.in*

Types of errors

If the signal is carrying binary encoded data, such changes can alter the meaning of the data.These errors can be divided into two types: Single-bit error and Burst error.

Single-bit Error:-The term single-bit error means that only one bit of given data unit (such as a byte, character, or data unit) is changed from 1 to 0 or from 0 to 1 as shown in Figure 1.

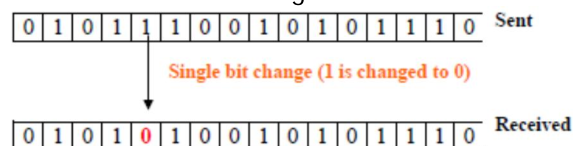

Figure 1 :- Single Bit Error

Burst Error:-The term burst error means that two or more bits in the data unit have changed from 0 to 1 or vice-versa. Note that burst error doesn't necessary means that error occurs in consecutive bits. The length of the burst error is measured from the first corrupted bit to the last corrupted bit.

Some bits in between may not be corrupted as shown in Figure 2.

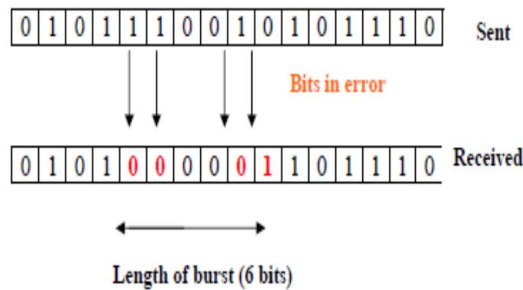

Figure 2 :- Burst Error

Some commonly used error detecting techniques are Triple Modular Redundancy and Error Correction Codes. The TMR triplicates all the memory parts of the system and to choose the correct data using a voter. This method have disadvantage of large area and complexity overhead of three times. TMR is a special case of the von Neumann method consisting of three versions of the design in parallel, with a majority voter selecting the correct output. As the method intimate, the complexity overhead would be three times plus the complexity of the majority voter and thus increasing the power consumption. For memories, it turned out that ECC codes are the best way to mitigate memory soft errors. Therefore the ECC became the best way to mitigate soft errors in memory [4].

The most commonly used ECC codes are Single Error Correction codes that can correct one bit error in a memory word. Due to consequence of augmenting integration densities, there is an increase in soft errors which points the need for higher error correction capabilities. SRAM reliability faces serious challenges from radiation induced soft errors (transient faults induced by ionizing radiation) and process-variation-induced defects in sub-100nm technologies. SRAM cells are designed with minimum geometry devices to increase density and performance, resulting in reduced critical charge to upset cells and more pronounced effects from process variations. Therefore, it has become conventional to protect memories with the application of error correcting codes such as single-error correcting [1], [3]. Exceeding advanced ECCs have been proposed for memory applications but even Double Error Correction codes with a parallel implementation incur in a significant power consumption penalty. The usual multierror correction codes, such as Reed–Solomon (RS) or Bose Chaudhuri–Hocquenghem are not suitable for this task due to complex decoding algorithm. The binary Bose–Chaudhuri–Hocquenghem codes, an optimized scheme is introduced which combines a multibit error-correcting BCH code with Hamming codes in a hierarchical manner to give an average latency as low as that of the single-bit correcting Hamming decoder. A Hamming algorithm with 2-b error-correcting capacity for small block sizes is another low-latency multibit ECC algorithm.

II. Comparison between some Forward Error Detection/Correction Techniques

The error detection and correction codes are explained and compared in this section.

• Hamming Code - In a hamming encoder parity bits are inserted into the message bits. These parity bits are decided so as to impose a fixed parity on different combinations of data and parity bits. In decoder those combinations are checked for that fixed parity. Accordingly decoder parity bits are set.Binary equivalent of this combination decides the location of the error. Then that particular bit is flipped to correct the data. Hamming code is a single error correction code. Double errors can be detected if no correction is attempted.

• Berger Code - Berger code is a unidirectional error detection code. It means it can only detect error either '1' flipped to '0' or '0' flipped to '1' but not both in a single code. If designed for detecting errors with '1' flipped to '0' then binary equivalent of number of 0s in the message is sent along with the message. Similarly when design for detecting '0' flipped to '1' error binary equivalent of the number of 1s in the message are sent along with the message. Decoder compares the number of 0s or 1s as per the design with the binary equivalent received. Mismatch between the two indicates the error. It can be used where error is expected to be unidirectional.

• Constant weight code -In this code a valid code word always have a constant weight. It means number of 1s in a valid code word is fixed. Hence any variation in this is an indication of error. It is simple but not efficient way of encoding as multiple errors can cancel out each other.

• M out of N code - In an M out of N encoder message is mapped to a N bit code word having M number of 1s in it. The N-M bits of message are appended with additional M number of bits which are used to adjust the number of 1s in the code. If the message consists of no 1s in it then all the M bits are set to '1'. It is also not an efficient code in terms of coding rate.

• Erasure code - Erasure means error when its location is known in advance from previous experience. Erasure code is able to correct such errors. In this type of code the decoder circuit does not need an error locator as it is already known. Hence only error magnitude is calculated by the decoder to correct the erasure.

• Low Density Parity check code - Low density parity check code is a linear block code. The message block is transformed into a code block by multiplying it with a transform matrix. Low density in the name implies low density of the transform matrix. That means number of 1s in the transform matrix is less. It is the best code as far as the coding gain is concerned but encoder and decoder design is complex. Mainly used for memory applications.

• Turbo Code - It is a convolutional code. Encoding is simple convolutional encoding. It is defined by (n, k, l) turbo code where n is the number of input bits, k is the number of output bits and l is the memory of the encoder. Decoding is done in two stages. First one is soft decoding stage then a hard decoding stage. It has very good error correcting capability i.e. coding gain. The main drawback is that it has low coding rate and high latency. Hence it is not suitable for many applications. But in case of satellite communication as the latency due to the distance itself is so high this additional latency is negligible. Hence it is used mainly in satellite communication.

• Reed Solomon Code - Reed Solomon code is a linear cyclic systematic non-binary block code. In the encoder Redundant symbols are generated using a generator polynomial and appended to the message symbols. In decoder error location and magnitude are calculated using the same generator polynomial. Then the correction is applied on the received code. Reed Solomon code has less coding gain as compared to LDPC and turbo codes. But it has very high coding rate and low complexity. Hence it is suitable for many applications including storage and transmission.

III. LITERATURE SURVEY

In literature work carried out on Majority Logic Fault Detector/Decoder and soft errors are presented.

R.C.Baumann, [1] this paper briefly reviews the types of failure modes for soft errors, the three dominant radiation mechanisms responsible for creating soft errors in terrestrial applications, and how these soft errors are generated by the collection of radiation-induced charge. The soft error sensitivity as a function of technology scaling for various memory and logic components is then presented with a consideration of which applications are majority likely to require soft error mitigation.

C.W.Slayma, [2] in their paper describes that as the size of the SRAM cache and DRAM memory grows in servers and workstations, cosmic-ray errors are becoming a dominant for systems designers and end users. Several techniques exist to detect and mitigate the occurrence of cosmic-ray upset, such as error detection, error correction, cache scrubbing, and array interleaving. This paper covers the tradeoff of these techniques in terms of area, power, and performance penalties versus increased reliability. In majority system applications, a combination of several techniques is required to meet the necessary reliability and data-integrity targets.

M.A.Bajura et al., [3] in their paper describes that a mathematical bit error rate model for upsets in memories protected by error-correcting codes and scrubbing is derived. This model is compared with expected upset rates for sub-100-nm SRAM memories in space environments. Because sub-100-nm SRAM memory cells can be upset by a critical charge of 1.1 fC or fewer, they exhibit significantly higher upset rates than those reported in earlier technologies. Because of this, single-bit-correcting ECCs become impractical due to memory scrubbing rate limitations. The overhead needed for protecting memories with a triple-bit-correcting ECC is examined relative to an approximate 2X "process generation" scaling penalty in area, speed, and power.

R. Naseer et al., [4] in their paper describes the concept of soft errors and the double error

correcting and Bose-Chaudhuri-Hocquenghem codes have not found favorable application in SRAMs due to non-alignment of their block sizes to typical memory word widths and particularly due to the large multi-cycle latency of traditional iterative decoding algorithms.

IV. CONCLUSION

Coding theory develops different methods to protect information against a noise for memory.

Error detection and correction mechanisms are vital and numerous techniques exist for reducing the effect of bit-errors and trying to ensure that the receiver eventually gets an error free version of the packet.An error-detecting/correcting code is an algorithm for expressing a sequence of numbers such that any errors which are introduced can be detected and corrected based on the remaining numbers.The error detection mechanisms are further classified based on their redundancy type,placement in the system hierarchy, and error type coverage. As a qualitative trade-off analysis,techniques in each category are explained in detail and compared to one another where applicable.It is shown that different techniques have different trade-offs in terms of performance, energy and area.All error detection, correction controlling mechanisms has been studied.

ACKNOWLEDGMENT

I express my sincere gratitude to my guide Mrs. B. N.Shobha, Associate Professor Dept. of ECE for guiding my research work throughout the process and I extend my thanks to Sapthagiri college of Engineering.

REFERENCES

[1] R.C.Baumann, Radiation-induced soft errors in advanced semiconductor technologies, IEEE Trans, vol 5, Sep 2005, pp 301–316.

[2] C. W. Slayman, Cache and memory error detection, correction, and reduction techniques for terrestrial servers and workstations, IEEE Trans, vol 5, Sep 2005, pp 397–404.

[3] M.A. Bajura, Models and algorithmic limits for an ECC- based approach to hardening sub-100-nm SRAMs, IEEE Trans, vol 54, Aug 2007 , pp 935–945.

[4] R. Naseer and J. Draper, DEC ECC design to improve memory reliability in sub-100 nm technologies, IEEE ICECS, pp 586–589.

[5] Shih-Fu Liu, Pedro Revingo, and Juan Antonio Meastro, Efficient majority fault detection with difference set codes for memory applications, IEEE Trans, vol 20, Jan 2012, pp 148–156.

[6] H. Naeimi and A. DeHon, Fault secure encoder and decoder for NanoMemory applications, IEEE Trans, vol 17, Apr 2009, pp 473–486.

[7] B. Vasic and S. K. Chilappagari , An information theoretical frame work for analysis and design of nanoscale fault-tolerant memories based on low-density parity-check codes, IEEE Trans, vol 54, Nov 2007 , pp 2438–2446.

[8] Pedro Reviriego,Juan A. Maestro, and Mark F. Flanagan, Error Detection in Majority Logic Decoding of Euclidean Geometry Low Density Parity Check (EG-LDPC) Codes, IEEE Transactions, vol 21, Jan 2013, pp 156-159.

[9] P .Ankolekar, S. Rosner, R. Isaac, and J. Bredow, Multi-bit error correction methods for latency-contrained flash memory systems, IEEE Trans, vol 10, Mar 2010, pp 33–39.

[10] M.A Jayarani, Dr.M.Jagadeeswari, Design and implementation of an efficient majority logic fault detector/decoder, IEEE Trans, vol 21, jan 2013 pp 201-212 .

IJSER