American International Journal of Research in Science, Technology, Engineering & Mathematics

Available online at http://www.iasir.net

ISSN (Print): 2328-3491, ISSN (Online): 2328-3580, ISSN (CD-ROM): 2328-3629

AIJRSTEM is a refereed, indexed, peer-reviewed, multidisciplinary and open access journal published by International Association of Scientific Innovation and Research (IASIR), USA

(An Association Unifying the Sciences, Engineering, and Applied Research)

A Review on Various Error Detection and Correction Methods Used in Communication

[1]Varinder Singh, [2] Narinder Sharma

[1]Student, Dept. of ECE, Amritsar College of Engineering and Technology, Amritsar-143001, Punjab, INDIA

[2]HOD, Dept. of EEE, Amritsar College of Engineering and Technology, Amritsar-143001, Punjab, INDIA

Abstract*: In today's world of wireless communication, the basic need of any communication system is to transmit and receive the error free data through any noisy channel. Due to the advancement in the data transmission the sources of noise and interference has also increased. Many efforts have been made by engineers to meet the demand for more reliable and efficient techniques for error detection and correction in the received data. To detect and correct the errors in the data transmission various techniques are used . This review paper delivers numerous error detection and correction techniques being used since last few decades.*

Keywords: Error *detection and correction (EDAC), Triple Modular Redundancy (TMR), Cyclic Redundancy Check (CRC), Parity check method, Horizontal Vertical Diagonal(HVD), Hamming method.*

I. INTRODUCTION

EDAC methods are used to find that the data is error free or is not corrupted, either by noisy channel, by hardware failure or during read-write operation in the memory segment. Various error detection methods exist in the communication system. One method currently utilized to produce reliable memory is the use of Error Correction Codes (ECC) to encode data before it is stored in the memory. Error correction codes take a set of information bits at the producer of the information and create a set of redundant bits based on the information bits. These redundant bits are sent or stored with the original set of information bits. The consumer of the information then uses the redundant bits to determine if any errors have occurred in transmission or storage. In the case of memory, the redundant bits are calculated and stored along with the original bits and then when they are read from the memory they are examined to determine if any errors have occurred between the time the information was stored and the time it was retrieved. The most common error detecting and correcting scheme being employed are parity bit, CRC, HVD and Hamming codes. All these methods are implemented on the second layer of OSI model at Data link layer. The upper layers work on some generalized view of network architecture and are not aware of actual hardware data processing. Therefore, the upper layers require error-free transmission between two systems. Almost every application did not work if it receiver data with errors. Applications like voice and video may not get that much affected and may still function well with some error.Data-link layer uses some error control mechanism to ensure that data bit streams are transmitted with certain level of accuracy. But to recognize how errors can be controlled, it is important to know what types of errors may occur.

A. Hardware Redundancy Vs Software EDAC

In order to protect semiconductor memories, software EDAC or redundancy can be used. Redundancy can either be hardware redundancy that is provided by extra components or time redundancy that is provided by extra execution time or by different moment of storage or can be a combination of both the hardware and time redundancies. To allow redundancy to detect permanent faults, the repeated computations are performed differently. TMR (Triple Modular Redundancy) is a suitable technique for SRAM-based FPGAs because of its full hardware redundancy property in the combinational and sequential logic. One solution for the protection of memories is use of hardware redundancy techniques, but they are too costly. When hardware redundancy is not possible, we have to go for software solutions. By using software Error Detection and Correction, transient faults in the combinational logic will never be stored in the storage cells, and bit flips in the storage cells will never occur or will be immediately corrected. For applications where read and write operations are done in blocks of words, such as secondary storage systems made of solid-state memories (RAM discs), software-implemented EDAC could be a better choice than hardware EDAC, because it can be used with a simple memory system and it provides the flexibility of implementing more complex coding schemes. With software EDAC, the data that is

read from main memory may be erroneous, if the error occurs after the last scrub operation and before the time of reading. In other words, single-bit errors may cause failures. In contrast, hardware EDAC checks all the data that is read from memory, and corrects single-bit errors. Therefore, hardware EDAC provides improved reliability and when feasible should be the first choice for protecting the main memory. When hardware EDAC is not available or affordable, software EDAC can be used as a low cost solution for enhancing the reliability of systems. For cases where data is read and written in blocks of words rather than individual words, software EDAC may be a better choice than hardware EDAC. The paper is organized as, the related work that has been done previously is given in section 3, conclusion is given in section 5 and in the end of the paper, references are included in the section 6.

II. TYPES OF ERRORS

There may be three types of errors:

1. Single bit error: In this a frame consist of only one bit corrupted anywhere thoughout.

2. Multiple bit errors: In this a frame is received with more than one bits in corrupted state.

3. Burst bit errors: In this a frame contains more than one consecutive bits corrupted or more than one bit flips.

III. LITERATURE SURVEY

Various error detection and correction methods are being used to maintain good level of reliability, to protect memory cells using protection codes. The method used in [3], is based on the hardware and time redundancy, although this technique reduces the *number of input and output pins of the combinational logic; it requires additional encoding/decoding circuitry. The reliability* issue can be solved, but the hardware redundancy schemes like duplication or triple modular redundancies are expensive. In [5], the encoder and the decoder can use any error detection and correction code. But the data is only coded in write operations, and decoded in read operations. So, the gathering of upsets is likely to occur and it depends on the reading and writing application request frequency. In order to avoid this accumulation of upsets, it is necessary to use an extra logic which is able to constantly detect and correct upsets in all coded data. The EDAC method given in [11] is again based on TMR, so increases the density as it is a hardware redundancy method. The method given in [4], perform memory error correction code which reduces power consumption in single-error correcting and double error-detecting checker circuits. This method can be employed to solve the non linear power optimization problem but it involves tedious computation of H- matrix. The method in [6], is called HVD, provides eminent detection coverage rate that can correct up to three upsets in a data array. It make use of parity codes in four directions in a data part to satisfy the reliability of memories and it can detect and correct the errors in the actual data bits. If the parity bits are itself erroneous, then those errors are detected by generating the parity bits for parities that is syndrome bits, but this is a complicated process. An easy way to find the errors in parity bits is presented in this paper. For this, we can take data bits and parity bits as a whole word. These words can be viewed as an m x n array. The hamming code will be used for the error detection and correction for this whole codeword containing both the data bits and the parity bits throughout the length of an array. After detecting the error, it can discover whether it is a data bit or a parity bit. The method used in [9], it shows that All of multiple error bit flips can be detected and 3-bit errors can be corrected, based on the experimental results. But it can correct only three bit error in a 8×8 matrix. The method used in[8], it shows that it can detect and correct up to 4 bits. With this method a large combination of multiple faults can be corrected which depend upon the length of the coded word array.

IV. VARIOUS METHODS FOR EDAC

A. Type of Error Control

The information of data is transfer from one hop to another hop. In TCP/IP model, the physical layer and the final layer of TCP/IP model transforms the data into stream of bits and transfers them into a signal toward the receiver device. Meanwhile those bits flow from one hop to another, they are exposing to channels interference, for example electrical interference or thermal noise that subject to unpredictable change. These channel interferences can change the shape of the transmitted signal leading into errors in the signal. There are two kinds of error single-bit error and burst error.

B. Parity Check

The most common method for detecting bits error with asynchronous character and character- oriented synchronous transmission is parity bit method. There are two types of parity check schemes: even and odd parity checks. With the even parity check, the redundant bit is chosen so that an even number of bits are set to one in the transmitted bit string of N+r bits, where r is the bit that used to be the even parity check and N is the bit that is transmitted by the transmitter of the network. The receiver re-computes the parity of each received bits from the transmitter and discard the strings with the invalid parity. The parity scheme is always used if 7-bits character is exchanged. If there are 7-bits that are transmitted by the transmitter and parity check are used to detect the error and often the eighth bit is the parity bit.

C. Cyclic Redundancy Check (CRC)

The second method in error detection in data link layer is cyclic redundancy check. As the parity check which is based on the submission of the binary the cyclic redundancy check is based on the binary division. In CRC, rather than adding bits to attain a desired parity, a series of redundant bits, called the CRC remainder, is attached to the end of a data unit so that the resulting data unit becomes exactly divisible by a second. On the receiver side, the incoming binary data bits are divided by the same number to be compared on the transmitter side. Implies that, if the remainder of the division is identical to the value that added to CRC when the data was transmitted, the data will be accepted, otherwise the unmatched reminder produced on the destination after the CRC is indicates the data unit has been damage during the transmission of data. The redundancy bits used by CRC are derived by dividing the data unit by a predetermined divisor; the remainder is the CRC. To be valid, a CRC must satisfy two conditions: It must have exactly one less bit than the divisor and appending it to the end of the data string must make the resulting bit sequence exactly divisible by the divisor. Operations of CRC described by the figure shown below:

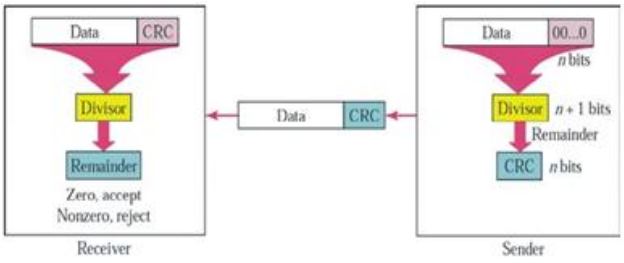


Fig 1. CRC Operation

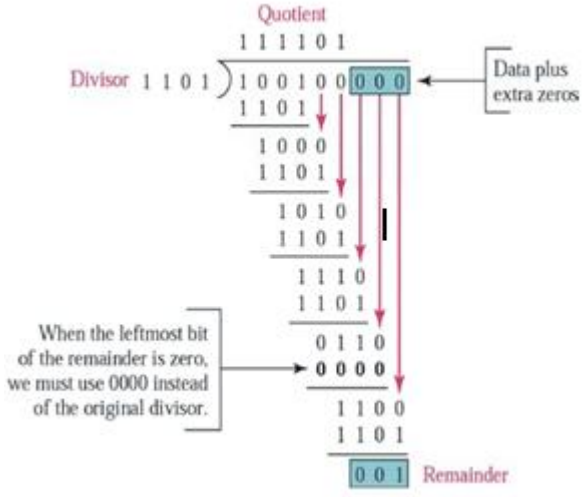


Fig 2. CRC in sender side

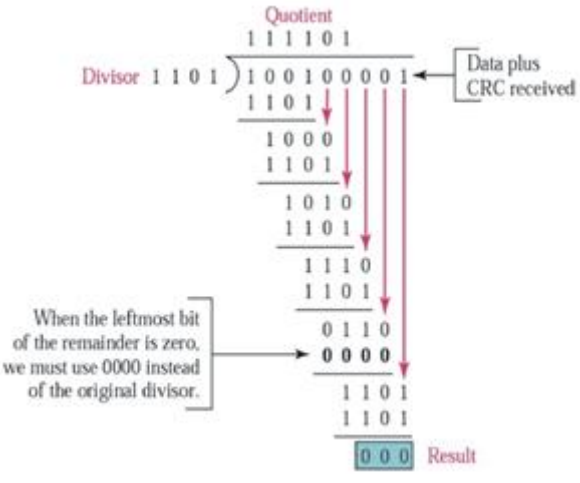
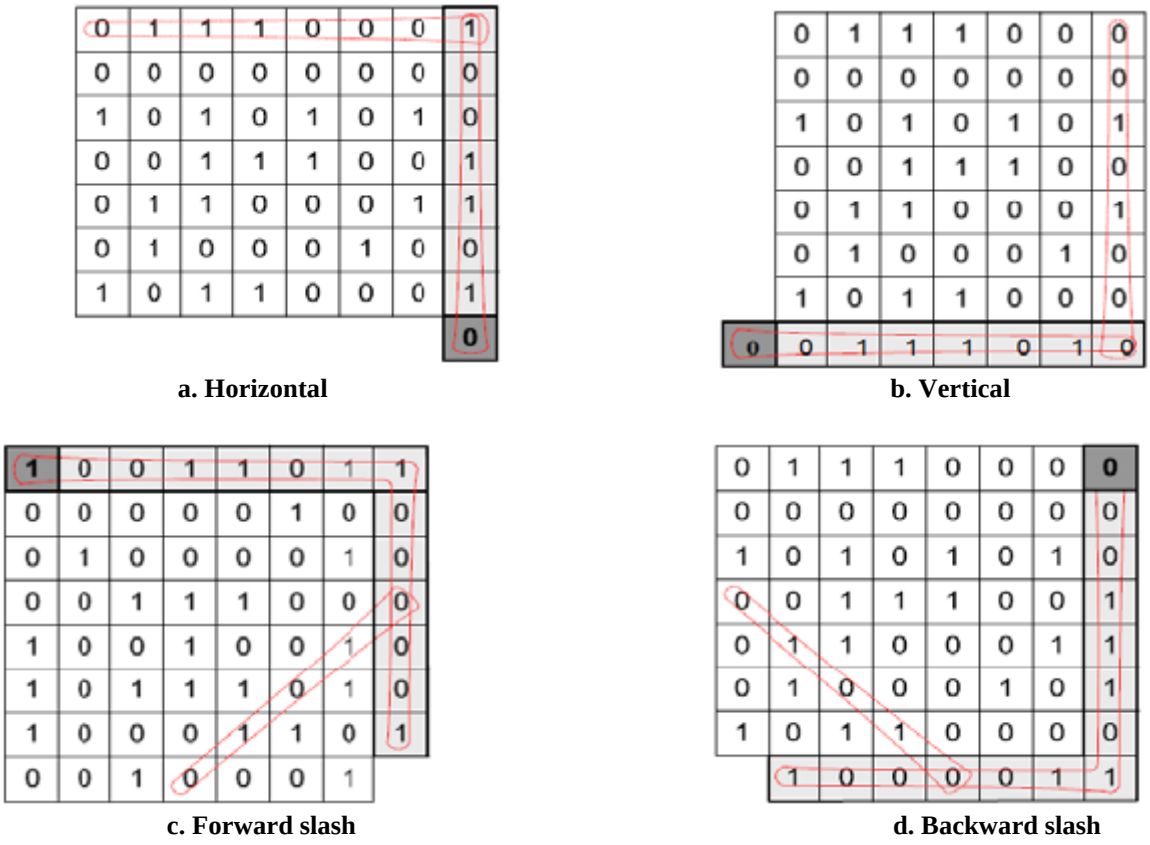


Fig 3. CRC in receiver side

D. HVD Method

Another method used for error detection and correction is called as HVD code. As the parity bits are applied at the three directions i.e row, column, forward slash and backward slash diagonals on a data part. Moreover, two horizontal (H), Vertical (V), parity bits, the diagonal (D), parity in both the directions can be used as shown in figures below a, b, c, d respectively:



a. Horizontal **b. Vertical**

c. Forward slash **d. Backward slash**

Fig 4. Parity bits position

To improve the ability of detection an additional parity bit is calculated on the basis of manipulated parity bits in all dimensions. In HVD code implementation H , V, D and D' represents the number of errors in the horizontal, vertical, forward slash and backward slash lines resp. Whereas H1,V1, F1 and B1 represents the position of first error in the horizontal , vertical and in both the diagonal parity lines resp.

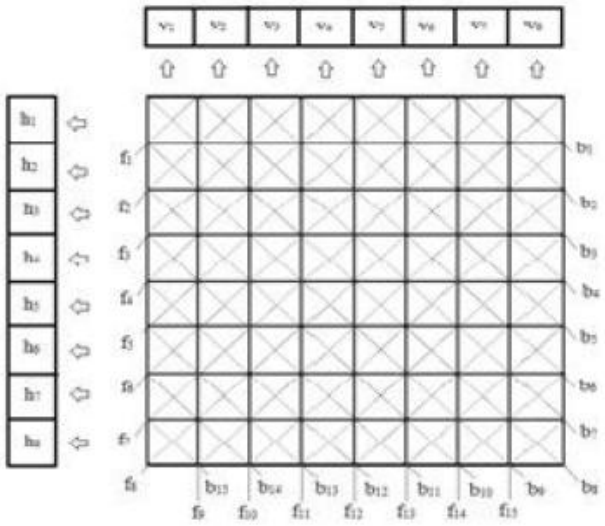


Fig 5. Two dimensional coded array

This method is based on 2-dimensional parity. Parities are calculated in all the directions that are horizontal, vertical, forward slash and backward slash. The 8x8 matrix is shown in figure 5 where the symbol h, v, f and b denotes the parity bits in horizontal, vertical, forward slash and backward slash lines respectively and subscripts represents the position of parity.

D. Hamming Method

The small size of the transistors or capacitors, along with cosmic ray effects, produces instant errors in stored information in large, solid RAM chips, especially that those are influential. These errors can be detected and

corrected by using error detecting and correcting codes in the RAM's. The frequent error detection scheme is the parity bit. The parity bit is generated and then stored in the memory along with the data word. The parity of the word is checked after reading the word from memory. The word is accepted if the parity of the bits read out is correct. If the parity of the bits read is incorrect, an error is detected, but it cannot be corrected. An error-correcting code uses multiple parity check bits that are stored with the data word in memory. Each check bit is a parity bit for a group of bits in the data word. When the word is read from memory, the parity of each group, including the check bit, is calculated. If the parity is correct for all groups, it signifies that no detectable error has occurred. If one or more of the newly generated parity values is incorrect, a unique pattern called a syndrome results that may be able to identify which bit is in error. A single error occurs when a bit changes in value from 1 to 0 or from 0 to 1 while stored or if it erroneously changes during a write or read operation. If the specific bit in error is identified, then the error can be corrected by complementing the erroneous bit.

Hamming Codes

The most common types of error-correcting codes used in RAM are based on the codes devised by R. W. Hamming. In the Hamming code, *k* parity bits are added to an *n*-bit data word, forming a new word of *n* _ *k* bits. The bit positions are numbered in pattern from 1 to *n* _ *k*. Those positions numbered with powers of two are reserved for the parity bits. The remaining bits are the data bits. The code can be used with words of any length. Before giving the general characteristics of the Hamming code, we will illustrate its operation with a data word of eight bits. Consider, for example, the 8-bit data word 11000100. We include four parity bits with this word and arrange the 12 bits as follows: The 4 parity bits *P*1 through *P*8 are in positions 1, 2, 4, and 8, respectively. The 8 bits of the data word are in the remaining positions. Each parity bit is calculated as follows:

*P*1 _ XOR of bits (3, 5, 7, 9, 11) _
*P*2 _ XOR of bits (3, 6, 7, 10, 11) _
*P*4 _ XOR of bits (5, 6, 7, 12) _
*P*8 _ XOR of bits (9, 10, 11, 12) _

Recall that the exclusive-OR operation performs the odd function. It is equal to 1 for an odd number of 1's among the variables and to 0 for an even number of 1's. Thus, each parity bit is set so that the total number of 1's in the checked positions, including the parity bit, is always even. The 8-bit data word is written into the memory together with the 4 parity bits as a 12-bit composite word. Substituting the 4 parity bits in their proper positions, we obtain the 12-bit composite word written into memory: When the 12 bits are read from memory, they are checked again for errors. The parity of the word is checked over the same groups of bits, including their parity bits. The four check bits are evaluated as follows:

*C*1 _ XOR of bits (1, 3, 5, 7, 9, 11)
*C*2 _ XOR of bits (2, 3, 6, 7, 10,11)
*C*4 _ XOR of bits (4, 5, 6, 7, 12)
*C*8 _ XOR of bits (8, 9, 10, 11, 12)
Bit position 1 2 3 4 5 6 7 8 9 10 11 12
*P*1 *P*2 1 *P*4 1 0 0 *P*8 0 1 0 0
Bit position 1 2 3 4 5 6 7 8 9 10 11 12
0 0 1 1 1 0 0 1 0 1 0 0
1⊕1⊕0⊕0⊕0 = 0
1⊕0⊕0⊕1⊕0 = 0
1⊕0⊕0⊕0 = 1
0⊕1⊕0⊕0 = 1

A 0 check bit designates an even parity over the checked bits, and a 1 designates an odd parity. Since the bits were written with even parity, the result, *C* _ *C*8*C*4*C*2*C*1 _0000, indicates that no error has occurred. However, if, the 4-bit binary number formed by the check bits gives the position of the erroneous bit if only a single bit is in error.

V. CONCLUSION

This paper presents different techniques which are used for detection and correction of single bit error and burst errors with improvement in efficiency and reliability of data transmission. All above mentioned techniques can detect and correct errors in data bits along with the parity bits without any extra calculations. It is observed that the Hamming method can correct and detect more number of errors as compared to other techniques mentioned above. Any other correction code can be used with hamming method to further increase the number of error detection and correction which results in enhancing code rate and reducing bit overhead.

REFERENCES

[1] Anlei Wang, Naima Kaabouch, "FPGA Based Design of a Novel Enhanced Error Detection and Correction Technique, IEEE, Vol. 3, Issue No. 5, March 2008, pp 25-29.
[2] Behrouz A. Forouzan "Data Communication and networking" 2nd edit. Tata McGraw Hill.

[3] Fernanda Lima, Luigi Carro, Ricardo Reis "Designing Fault Tolerant Systems into SRAM-based FPGAs" Anaheim,Vol. 3, June 2003,pp 312-318.

[4] Heesung Lee, Joonkyung Sung, and Euntai Kim, "Reducing Power in Error Correcting Code using Genetic Algorithm", World Academy of Science, Engineering and Technology 25 2007

[5] M. Imran , Z. Al-Ars, G. N. Gaydadjiev, "Improving Soft Error Correction Capability of 4-D Parity Codes", IEEE transaction, Vol. 4, Issue 2, January 2009, pp 233-238.

[6] M.KISHANI, H. R. ZARANDI, H. PEDRAM, A TAJARY, M. RAJI, B. GHAVAMI, "HORIZONTAL-VERTICAL DIAGONAL ERROR DETECTING AND CORRECTING CODE TO PROTECT AGAINST WITH SOFT ERRORS" SPRINGER SCIENCE, VOL. 15, ISSUE NO. 3-4, MAY 2011, PP 289-310.

[7] NARINDER PAL SINGH, SUKHJIT SINGH, VIKRANT SHARMA, AMANDEEP SEHMBY, "RAM ERROR DETECTION AND CORRECTION USING HVD IMPLEMENTATION" EUROPEAN SCIENTIFIC JOURNAL,VOL. 9, ISSUE NO.33, NOVEMBER 2013,PP 424-435**.**

[8] S. Sharma, Vijay Kumar, " An HVD Based Error Detection and Correction of Soft Errors in Semiconductor Memories Used for Space Application", International conference on devices, circuits and systems (ICDCS), March 2012, pp. 563-56.

[9] Shubham Fadnavis, "An HVD Based Error Detection And Correction Code In HDLC Protocol Used For Communication", International Journal of Advanced Research in Computer and Communication Engineering, Vol. 2, Issue No.6, June 2013, pp 2349-2353.

[10] T. Kasarni , A Kitai and S. Lin "On the Undetected Error Probability for Shortened Hamming Codes", IEEE Transaction Communication , Vol.33, Issue No. 2, 1985, pp 570 -574.

[11] Y. Bentoutou, "Program Memories Error Detection and Correction On- Board Earth Observation Satellites", World Academy of sciece, Engineering and Technology 66 2010.