

Data Integrity Enhancement with Interleaved Hsiao Code and CRC Error Detection and Correction – A Brief Review

Meghana Das¹, Dr Sudha K L²

"Department of Electronics and Communication Engineering"
"Dayanand Sagar College of Engineering"

DOI: 10.29322/IJSRP.13.06.2023.p13842

<http://dx.doi.org/10.29322/IJSRP.13.06.2023.p13842>

Paper Received Date: 10th May 2023

Paper Acceptance Date: 13th June 2023

Paper Publication Date: 21st June 2023

Abstract- With the continuous miniaturization of CMOS technologies, the upcoming challenges of soft errors and reliability are expected to escalate. In order to address these challenges and bolster circuit reliability, this research paper presents a novel approach involving the design of two interleaved Double-Adjacent-Error-Corrections (DAECs) for Error Detection and Correction (EDAC). The DAECs employ a combination of the Hsiao Code, a modified version of Hamming codes widely utilized in modern systems, and the Cyclic Redundancy Code (CRC). The CRC, a non-secure hash function, is adept at identifying unintentional alterations in digital data within computer networks. It functions by leveraging a generator polynomial as the divisor in a polynomial long division over a finite field, treating the input data as the dividend, and yielding the remainder as the result. Due to its ease of hardware implementation, CRCs are extensively employed. Additionally, this research proposes two storage formats and algorithm designs that enable the efficient management and storage of the 48-bit codeword in memory devices with limited capacity, which is particularly relevant in scenarios such as satellite systems where board space is scarce. To implement and evaluate the designs, Verilog, C++, and ModelSim are employed as the primary tools for design creation and testing. By combining the benefits of the Hsiao Code, CRC, and the optimized storage formats, this research aims to detect and correct errors effectively, ultimately enhancing the reliability of circuits in the face of soft errors. The utilization of Verilog, C++, and ModelSim facilitates the thorough examination and validation of the proposed designs.

Keywords: Hsiao Code, CRC, Hamming Code, EDAC

I. INTRODUCTION

In the rapidly evolving CMOS industry, the reduction in semiconductor size presents a challenge to maintaining the reliability of memory devices, which are susceptible to external factors like electrical surges and ionizing radiation.

This publication is licensed under Creative Commons Attribution CC BY.

<http://dx.doi.org/10.29322/IJSRP.13.06.2023.p13842>

The increasing concern over reliability necessitates the detection and correction of soft errors.

The presence of soft errors in System on Chips (SoC), particularly in embedded memories, is well-documented, resulting from interactions with charged particles or radiation. Soft errors refer to incorrect signals or data, often caused by defects or broken components in design or construction. Cosmic rays can also induce single-event upsets, leading to soft errors.

Single Event Errors (SEU) do not cause permanent damage to circuits but can introduce errors. In space-based microprocessors, the 1st and 2nd-level cache memories are particularly vulnerable due to their small size and high speed, resulting in limited charge capacity. To ensure survivability against SEUs, these caches are often disabled in terrestrial designs.

Various prominent error detection and correction techniques, such as Hamming, Bose-Chaudhuri-Hocquenghem (BCH), and Hsiao Codes, have been proposed as single error correction and double error detection (SEC-DED) Error Detection and Correction (EDAC) methods. Among these, the Hsiao Code EDAC offers superior performance in terms of speed, hardware requirements, and detection rates for three- and four-bit errors, reducing the likelihood of erroneous EDAC correction.

Commonly used error correction codes, such as Hamming or Hsiao codes, provide single-bit error correction and double-bit error detection (SEC-DED) capability. Other codes, like double-bit error correcting and triple-bit error detecting (DEC-TED) codes, single-nibble error-correcting and double-nibble error detecting (SNC-DND) codes, and Reed-Solomon error correction codes, have also been proposed for memory protection. However, in practice, multiple SEC-DED codes are typically interleaved to achieve multi-bit correction.

Early research focused on minimizing the area and delay overheads of Error Correcting Code (ECC) circuits. Hamming

demonstrated the feasibility of SEC-DED codes using a specific check matrix, while Hsiao introduced an alternative matrix with odd-weight columns that offer SEC-DED capability with reduced hardware area and shorter delay compared to traditional Hamming SEC-DED codes. Recent research also aims to minimize power consumption in addition to area and delay optimization. In this study, the design proposes two efficient EDACs capable of correcting and detecting errors. The first EDAC utilizes the Hsiao Code with an encoder matrix, while the second EDAC employs a CRC polynomial with its respective encoder matrix. These two EDAC designs, known as Double-Adjacent-Error-Correction (DAEC) EDACs, are then interleaved. Additionally, the paper presents storage format and algorithm designs to effectively manage and store the 48-bit codeword in memory devices with limited capacity, such as 8-bit and 16-bit formats.

II. LITERATURE REVIEW

In this section, a brief introduction of the work that has been done concerning this field is discussed.

The interleaving of the Hsiao Code and CRC-based EDAC approaches are explained in this paper [1]. It shows promising results with minimum hardware requirements. It also proposes an algorithm for how to manage the 48-bit codeword when board space is scarce. The storage format and algorithm allow us to support this class of EDACs with minimum hardware support.

The research study on two of the most popular linear ECC codes, Hamming and ECC is shown in this paper [2]. It explores the efficiency of their application in embedded systems. It shows the results that Hamming codes are simpler to implement and display better efficiency for smaller data lengths while Hsiao codes are more complex but have better performance, especially for larger data sizes.

In this paper [3], the efficient implementation of error correction code (ECC) processing circuits based on single error correction and double error detection SEC-DED code with check bit pre-computation is proposed for memories. When compared with a conventional implementation utilizing the odd-weight-column code, the implementation based on the proposed SEC-DED code with check bit pre-computation achieves reductions in the number of gates, latency, and power consumption of the ECC processing circuits.

In this paper [4], designs of novel joint crosstalk avoidance and triple-error-correction/quadruple-error-detection codes are proposed, and their performance is evaluated in different NOC fabrics. It is demonstrated that the proposed codes outperform other existing coding schemes in making NOC fabrics reliable and energy efficient, with lower latency.

The author in [5], demonstrated a new way of constructing a class of SEC-DED codes that uses the same number of check bits as the Hamming SEC-DED code but is superior in cost,

performance, and reliability. This also minimizes the hardware, thus tending to lower the failure rate of decoders.

The author in [6], explains the optimal generating algorithm for a matrix of equal weights columns and equal weights of rows. Then modifies some steps and uses the modified algorithm for error correction, it is shown that it is efficient, fast and it's optimal in average cases.

In this paper [7], The Hsiao code is applied to cache memory using FPGA programmable Xilinx circuits. With this implementation, the size of the syndrome generator is reduced and the cost of the error-correcting scheme is also reduced compared to the traditional Hamming code-based solution. This solution using an SEC-DED Hsiao code, increases reliability through fault tolerance, leading to low cost and low memory chip dimension, as this method solves the problem of faults by testing and correcting errors inside the chip.

This paper [8] presents Ultrafast codes, designed for very fast encoding and decoding operations. Ultrafast codes offer high-speed encoder and decoder circuits and interesting error coverages. These codes can also be useful to protect high-speed memories or caches. Firstly, they have summarized Ultrafast SEC, SEC-DED, and SEC-DAEC codes, presented in previous works. Then, new Ultrafast SEC-xAEC-DED codes have been introduced, describing the design methodology and the implementation details. The results confirm that Ultrafast codes achieve very low propagation delays, whereas adding very reasonable increments in the silicon area and the power consumption.

In this paper [9], a new SEC-DED-DAEC code is demonstrated to achieve high-reliability protection against neutron-induced soft errors in on-chip memory systems. It is shown that the proposed code has higher reliability than other SEC-DED-DAEC codes because it addresses the miss-correction problem. It is proved that the proposed code is suitable for a protection scheme against MCU in on-chip memory systems.

The paper [10] proposes the investigation of four SEC-DED codes with different word lengths. The effect of the word length on error control capability is shown. Two different hardware implementations: XOR tree-based and LUT based, for these four Hsiao codes are explored in this paper. The results show that the LUT structure outperforms the standard XOR structure in error correction speed in most cases.

HSIAO CODE

Hsiao codes, also known as single-error-correcting and double-error-detecting (SEC-DED) codes, are a type of error-correcting code used in memory systems. They are designed to detect and correct errors in data that are caused by single-bit errors and detect but not correct double-bit errors. Hsiao codes are based on Hamming code, which is a linear error-correcting code that can detect and correct single-bit errors.

However, unlike the Hamming code, Hsiao codes can also detect double-bit errors by using an additional parity bit.

In Hsiao codes, each data bit is represented by a binary number, and an additional parity is added to the code to detect errors. The number of parity bits added depends on the number of data bits in the code. For example, if there are n data bits, then $\log_2(n+1)$ parity bits are added to the code.

When data is transmitted or stored using Hsiao codes, the receiver or memory system can detect and correct single-bit errors, but not double-bit errors. If a double-bit error is detected, the system will report an error and discard the data.

Overall, Hsiao codes provide a balance between error detection and correction, making them a popular choice for computer memory systems where errors can occur due to various reasons such as electromagnetic interference, cosmic rays, or manufacturing defect.

CYCLIC REDUNDANCY CODE

Cyclic Redundancy Code (CRC) is an error-detection technique used in digital communication networks and storage systems. It is a type of checksum that is used to ensure the integrity of data transmitted over a channel or stored in memory.

CRC works by dividing the data into blocks of a fixed size and adding a checksum to each block. The checksum is computed by performing a polynomial division of the data with a fixed generator polynomial

The remainder of the division is the checksum, which is appended to the data block. When the data is received, the same polynomial division is

performed on the received data, and the resulting remainder is compared to the checksum that was sent. If they match, the data is considered to be error-free.

CRC codes are widely used in communication systems such as Ethernet, Wi-Fi, and Bluetooth, as well as in storage systems such as hard disk drives and optical discs. They are simple and efficient to implement in hardware and software and can detect most errors that occur during transmission or storage. However, they are not perfect and can still fail to detect some types of errors, such as those that occur in bursts.

III. METHODOLOGY

Firstly, the design of EDAC that uses the Hsiao code's encoder matrix is achieved. Then the design of a second EDAC that uses a CRC encoder matrix. Finally, the interleaving of both the EDACs to improve the error detection and correction capabilities of our EDACs is made. Also, a design of storage format and algorithm to manage and store the 48-bit codeword in 8-bit and 16-bit memory devices is proposed.

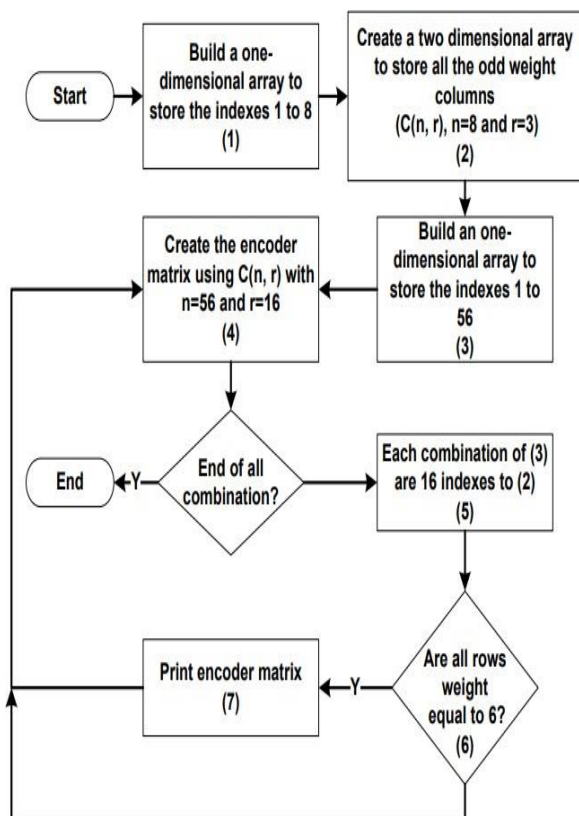
DESIGN OF HSIAO CODE'S ENCODER MATRIX

A C++ program is used to create Hsiao Code's matrix generator, to generate the H-matrices compliant with the three Hsiao Code's rules. Another C++ program is created, a code generator, to take the generated matrix as the input and produce the Verilog code for the EDAC's encoder. The three Hsiao Code rules are:

1. Every column should have an odd number of 1's; i.e., all column vectors are of odd- weights. In the following, n is used to indicate the number of rows in the H-matrix, and r to indicate the number of 1's in each column, which must be an odd number.
 - Use the combination formula $C(n, r) = \frac{n!}{r!(n-r)!}$ to generate all possible odd-weighted columns with $n=8$ and $r=3$. $n=8$ because the matrix has eight rows. $r=3$ because 3 is the smallest odd number next to 1, and 1 is already used in the check-bit columns. Therefore, a total of 56 combinations are obtained.
2. The total number of 1's in the H-matrix should be at a minimum.
 - Compute the 1's in the H-matrix as $(16 * 3) + 8$, or 56 1s' since the first 16 columns of the matrix contain three 1's, and the last eight columns are for check-bits.
3. The number of 1's in each row of the H-matrix should be made equal, or as close as possible, to the average number, i.e., the total number of 1's in the H-matrix divided by the number of rows.
 - Divide the 56 1s' by 8 and get 7 1s' per row. This number includes the check-bit in the last eight columns.

Based on the rules above, the matrix generator in C++ is generated. The figure shows the flowchart of our generator that works as follows:

1. First, build a one-dimensional array to store the row indexes, 1 to 8, since the new matrix has eight rows. Each of the numbers refers to one of the eight rows.
2. A 56x3 array to store all the combinations of the row indexes in the column vectors that each contains a 1. Since each column vector in the H-matrix has three 1's, the combinations of three out of the eight-row indexes generated in step 1 (1 to 8) are taken and stored each combination in a column vector in this 56x3 array. There is a total of 56, or $C(8, 3)$, such combinations stored in this array.
3. Build a one-dimensional array to store the numbers 1 to 56. This is an array of the column indexes of the 56x3 array generated in Step 2.



4. The column indexes to build the H-matrix are as follows. Since the H-matrix has 16 columns, not including the eight columns used for the check-bit, and each column contains three 1's, a combination of 16 out of the 56 column indexes generated in step 3 (1 to 56) is chosen. If all the C (56, 16) combinations are exhausted, then end the program; otherwise, continue to step 5.
5. Use the 16 indexes generated in step 4 to index into the 56x3 array, and obtain a 16x3 sub-array. Use each of the 16 columns to build a 16x8 matrix. For example, if the first column is chosen, then a column vector in the new matrix containing eight rows is built, by storing 1's in rows 1, 2, and 3, and 0's in all remaining rows in this column vector.
6. If all the rows in the matrix generated in step 5 contain 6 1s' (not including the check-bit), go to step 7, otherwise go to step 4.
7. Print the encoder matrix, indicating that the matrix is successfully generated; then go to step 4 to check whether all the C (56, 16) combinations are exhausted.

DESIGN OF CYCLIC REDUNDANCY CODE'S ENCODER MATRIX

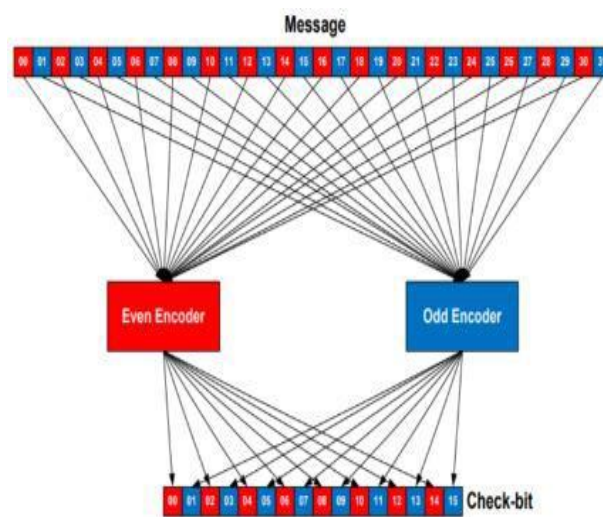
Cyclic Redundancy code is another type of error detection technique used to ensure the integrity of data transmitted over a channel or stored in memory. For networking, the interest is

the Hamming Distance (HD), which is the least possible number of bit inversions in a message that can create an error undetectable by that message's CRC-based Frame Check Sequence.

Here the design of the next EDAC based on a CRC polynomial to leverage the CRC detection capability is done. To increase the information rate, the computation of a 16-bit message with an 8-bit CRC generator is proposed. To reduce the bit weight in the encoder matrix, the CRC's seed value is set to zero. When a two-input XOR gate has an input equal to a logical zero, its output value is equal to the other input's value. Based on this Boolean state, the elimination of the XOR gates is required for the seed value, thus reducing the bit-weight of our CRC matrix. The development of a CRC-based encoder matrix generator in C++ to generate our EDAC encoder matrix is made with these criteria.

DESIGN OF AN INTERLEAVED EDAC

The EDAC uses two identical single error correction and double error detection (SEC-DED) EDACs configured as even and odd encoders, syndromes, and decoders. For example, two Hsiao Code encoder matrices generated, or two CRC encoder matrices generated can be used as the two identical encoders. This configuration improves the error correction and detection rates. For example, if a double-bit error has one error occurring on an even bit, and the other on an odd bit of a codeword, this error can be corrected.



CODEWORD STORAGE FORMAT AND ALGORITHM

A hardware- or software-based EDAC provides methods to ensure data reliability in memory devices. The cost for more reliable data is a lower information rate because an EDAC creates additional bits for error detection and correction. These extra bits are known as the check-bit.

Depending on the EDAC implementation, the number of bits used for the check-bit varies. In this research, a 16-bit check-bit (or two 8-bit check-bits) is considered.

When the EDAC corrects an error, the fault-tolerant memory controller writes the corrected codeword (message and check-bit) back to the same memory location. This process is known as memory scrubbing. A 48-bit codeword would need six 8-bit, three 16-bit, or one 32-bit and one 16-bit memory device to store the message (code or data) and the check-bit. In a scenario where a satellite's board space is scarce, these memory devices' configurations are not desirable. Instead, it is recommended to use an 8-bit or a 16-bit memory device to store the 48-bit codewords.

The algorithm for storing the message and check-bit in an 8-bit or a 16-bit memory device works as follows:

1. The EDAC computes the 16-bit check-bit from the 32-bit message (data or code).
2. The memory controller writes the 32-bit message as follows:
 - 8-bit memory device— writes the 32-bit message in four bytes.
 - 16-bit memory device—writes the 32-bit message in two 16-bit words
3. After writing the fourth byte, the address is 0000_0003h.
4. The controller takes the last address (0000_0003h), shifts the address value to the right by one bit, and inverts the address. The address after inversion is the written address for the check-bit. In this case, the address is FFFF_FFFEh.
5. The memory controller writes the lower byte of the 16-bit check-bit to address FFFF_FFFEh and the higher byte to FFFF_FFFFh.

Address	Data (8-bit)	Assignment	Size
FFFF_FFFF		Check-bit	33.33% for Check-bits
FFFF_FFFE		Check-bit	
FFFF_FFDD		Check-bit	
FFFF_FFDC		Check-bit	
FFFF_FFDB		Check-bit	
FFFF_FFDA		Check-bit	
FFFF_FF99		Check-bit	
FFFF_FF88		Check-bit	
...	
...	
0000_000F		Message	66.66% for Message
0000_000E		Message	
0000_000D		Message	
0000_000C		Message	
0000_000B		Message	
0000_000A		Message	
0000_0009		Message	
0000_0008		Message	
0000_0007		Message	
0000_0006		Message	
0000_0005		Message	
0000_0004		Message	
0000_0003		Message	
0000_0002		Message	
0000_0001		Message	
0000_0000		Message	

: The format for storing codewords in an 8-bit wide memory device.

Address	Data (16-bits)	Assignment	Size
FFFF_FFFE		Check-bit	33.33% for Check-bits
FFFF_FFFC		Check-bit	
FFFF_FFFA		Check-bit	
FFFF_FF88		Check-bit	
...	
...	
0000_000E		Message	66.66% for Message
0000_000C		Message	
0000_000A		Message	
0000_0008		Message	
0000_0006		Message	
0000_0004		Message	
0000_0002		Message	
0000_0000		Message	

: The format for storing codewords in a 16-bit wide memory device.

IV. FUTURE SCOPE

This proposed method shows the implementation of the encoding design with Hsiao & CRC encoding. By replacing this with some other efficient encoding architecture the performance of the EDAC encoding process can be further improvised.

V. CONCLUSION

This paper presents a brief review of two error detection and correction (EDAC) methods, the Hsiao code and the Cyclic Redundancy code. The interleaving of these methods is discussed. This helps EDAC structure helps to correct single-bit-errors, double-adjacent-bit-errors, and double-bit-errors that one error occurs on an even-bit and the other on an odd-bit of a codeword. This paper also shows how to manage the 48-bit codeword when board space is scarce. The storage format and algorithm allow us to support this class of EDACs with minimum hardware support.

VI. REFERENCES

- [1] Mong Tee Sim, Yanyan Zhuang “Design of Two Interleaved Error Detection and Corrections using Hsiao Code and CRC”, 2020, IEEE.
- [2] G. Tshagharyan, G. Harutyunyan, S. Shoukourian, Y. Zorian “Experimental study on Hamming and Hsiao Codes in the Context of Embedded Applications”, 2017, IEEE.
- [3] Sanguhn Cha and Hongil Yoon “Efficient Implementation of Single Error Correction and Double Error Detection Code with Check Bit Pre-computation for Memories”, Journal of Digital

- Semiconductor Technology and Science, Vol.12, No.4, 2012.
- [4] Amlan Ganguly, Partha Pratim Pande, and Benjamin Belzer, "Crosstalk-Aware Channel Coding Schemes for Energy Efficient and Reliable NOC Interconnects", 2009, IEEE.
- [5] M.Y. Hsiao, "A Class of Optimal minimum Odd-Weight-Column SED-DED Codes", 1970.
- [6] Li Chen, "Hsiao-Code Check Matrices and Recursively Balanced Matrices", (Chinese. English summary) [J] J. Nanjing Inst. Technol. 16, No.2, 33-39 (1986). [ISSN 0254-4180],2013.
- [7] NOVAC Ovidiu, SZTRIK Janos, VARI-KAKAS Stefan, KIM Che-Soong "Reliability Increasing Method Using an SEC-DED Hsiao Code to Cache Memories, Implemented with FPGA Circuits", Journal of Computer Science and Control Systems, Vol.4, No.2, 2011.
- [8] Luis-J. Saiz Adalid, Joaquin Gracia-Moran, Daniel Gil-Tomas, J Carlos Baraza-Calvo, and Pedro-J. Gil-Vicente, "Ultrafast Codes for Multiple Adjacent Error Correction and Double Error Detection", 2019, IEEE.
- [9] Ho-yoon Jun and Yong-surk Lee, "Single Error Correction, Double Error Detection, and Double Adjacent Error Correction with No Mis-correction Code", 2013, IEICE Electronics Express.
- [10] Wei Gao, "A Study on Implementation of ECC for Embedded RAM", Queen's University, Kingston, Ontario, Canada, 2003.