International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 11 Issue: 04 | Apr 2024 www.irjet.net p-ISSN: 2395-0072

DESIGN AND SIMULATION OF DATA SEGMENTATION SECTION CODE FOR SPACE COMMUNICATION

K.G.VENKATA KRISHNA[1], B.LAVANYA[2], K.SRAVANI[3], K.JAGADEESH[4]

[1] Assistant Professor , Dept of ECE , Krishna University College of Engg & Tech, Machilipatnam , Krishna-521004 , AP , India

[2,3,4] UG Students , Dept of ECE, Krishna University College of Engg & Tech, Machilipatnam , Krishna-521004 , AP , India

---

Abstract - *In the interest of technological scalability, bit mistakes caused by single occurrences or repeated cell disruptions by environmental elements like cosmic radiation, alpha and neutron particles, or the highest temperature in space can cause on-chip memory in a die to corrupt data. Through a communication channel, error detection and correction algorithms (ECC) identify and fix faulty data. This work proposes an enhanced divide-symbol based error correcting 2-dimensional code to decrease radiation-induced MCUs in memory for space applications. The XOR technique was used to examine the diagonal, parity, and check bits used in the data encoding process. Once more, an XOR operation was carried out between the encoded bits and the recalculated encoded bits in order to retrieve the data. Following analysis, there is a procedure of verification, selection, and correction. Xilinx Vivado, which was built in Verilog HDL, was used to simulate and synthesize the suggested methodology. In contrast to established techniques, this encoding and decoding procedure uses less energy, takes up less space, and requires less time.*

Key Words: **—Error Correction Code, Multiple Cell Upsets, Encoder, Parity, Decoder, Memory**

1.INTRODUCTION

An earlier error detection approach required a parity. Each character was processed by appending an additional bit. Many factors, including the type of parity and the amount of logic-one bits in the data character, make the bit stubborn.

Alliteration code is an additional code structure that provides error detection information. The code doodle displays a bit across channel to achieve error-free communication. Data are separated into data bits and data blocks in a data stream. Each block is sent out for the predetermined amount of times. It is ineffective as parity since the same mistakes cause greater issues. They are straightforward and utilized in the number station transmission process. The process of identifying mistakes and transforming the original data into error-free data is known as error correction. The ability of a system to continue operating as intended even when one or more of its component parts fail is known as fault tolerance. A method called "hamming code" uses a set of ECC to identify and fix bit mistakes in data transmission and storage.

2. LITERATURE SURVEY

2.1 R. C. Baumann (2005), "Soft errors in advanced computer systems," IEEE Des. Test. Comput., vol. 22, no. 3, pp. 258-266.

Due to the reduction in device feature size and supply voltage, achieving soft error reliability in sub-micrometer digital circuits is becoming extremely challenging. We consider the problem of choosing the gate sizes in a combinational logic circuit in order to minimize the soft error rate (SER) of the circuit. This problem can be solved using the heuristic as well as the greedy-based approaches for small-size problems; however, when the circuit size increases, the computational time grows exponentially, and hence, the previous methods become impractical. This paper proposes a novel technique for soft error tolerant design of large-scale combinational circuits using a cone-oriented gate sizing. Circuit partitioning is used to split the circuit into a set of small sub-circuits. The gates of sub-circuits are resized, such that the entire circuit SER is reduced based on a new soft error descriptor metric. The proposed cone-oriented gate sizing framework is used for selective gate sizing, leading up to 31% SER reduction with less than 17% area overhead when applied to large-scale benchmarks. The results also show that the proposed method is 21% more efficient and up to 292 times faster when compared with that obtained using a similar work based on the sensitive-based gate sizing scheme

2.2 C. L. Chen and M. Y. Hsiao (1984), "Error-correcting codes for semiconductor memory applications: A state-of-the-art review," IBM J. Res. Develop., vol.28, no. 2, pp. 124-134.

This paper presents a state-of-the-art review of error-correcting codes for computer semiconductor memory applications. The construction of four classes of error-correcting codes appropriate for semiconductor memory

designs is described, and for each class of codes the number of check bits required for commonly used data lengths is provided. The implementation aspects of error correction and error detection are also discussed, and certain algorithms useful in extending the error-correcting capability for the correction of soft errors such as α-particle-induced errors are examined in some detail.

2.3 E. Ibe, S. Chung, S. Wen, H. Yamaguchi, Y. Yahagi, H. Kameyama, S. Yamamoto and T.Akioka (2006), "Spreading diversity in multi-cell neutron-induced upsets with device scaling," in Proc. IEEE Custom Integrated Circuit Conf., pp. 437-444.

Because of technological scaling, on-chip memories in a die experience bit errors as a result of single events or multiple cell disruptions caused by environmental factors like cosmic radiation, alpha and neutron particles, or the space environment's maximum temperature. This results in data corruption. The damaged data over a communication channel is identified and corrected by error detection and correction techniques (ECC). It is suggested in this research to weaken radiation-induced MCUs in memory for space applications by using an enhanced error correction 2-dimensional coding based on the divide-symbol. The XOR operation was used to assess the diagonal bits, parity bits, and check bits for data encoding. The encoded bits and the recalculated encoded bits were again XOR ed in order to extract the data. Following analysis, there is a procedure of verification, selection, and correction. Xilinx Vivado was used to simulate and synthesis the suggested design, which was then implemented in Verilog HDL. As compared to well-known existing approaches, this encoding-decoding procedure uses less power, takes up less space, and has shorter latency.

2.4 P. Reviriego, J.A. Maestro and C. Cervantes (2007), "Reliability analysis of memories suffering multiple bit upsets," IEEE Trans. Device Mater. Rel., vol. 7, no. 4, pp. 592- 601.

In space missions, boot software is in charge of the initialization sequence of flight computers. The processor module in which it runs has a high tolerance to radiation, although not all devices have the same tolerance level. A boot software design capable of recovering from errors in the most vulnerable devices shall provide greater system reliability. This work has been carried out in the context of the boot software development for the control unit of the Energetic Particle Detector instrument on-board the Solar Orbiter mission. This mission operates close to the Sun where high-energy particles can cause single event effects on electronic devices, especially SDRAM and EEPROM, which show lower radiation tolerance than the other devices.

2.5 G. C. Yang (1995), "Reliability of semiconductor RAMs with soft-error scrubbing techniques," IEEE Proc. Computers and Digital techniques, vol. 142, no. 5, pp. 337- 344.

This article presents an analysis of the reliability of memories protected with Built-in Current Sensors (BICS) and a per-word parity bit when exposed to Single Event Upsets (SEUs). Reliability is characterized by Mean Time to Failure (MTTF) for which two analytic models are proposed. A simple model, similar to the one traditionally used for memories protected with scrubbing, is proposed for the low error rate case. A more complex Markov model is proposed for the high error rate case. The accuracy ofthe models is checked using a wide set of simulations. The results presented in this article allow fast estimation of MTTF enabling design of optimal memory configurations to meet specified MTTF goals at minimum cost. Additionally the power consumption of memories protected with BICS is compared to that of memories using scrubbing in terms of the number of read cycles needed in both configurations.

3. EXISTING WORK

1) A method for determining the many cell disturbances in memory is the Punctured Difference Set (PDS) coding.

2) The interleaving approach, which has been employed to restrict numerous cell disruptions, divides bits by the same logical word into distinct physical words.

3) In order to provide fortification against multiple cell disruptions, built-in current sensors (BICS) are slated for reinforcement on corrective single error detection and double error detection.

4) 2-Dimentational matrix codes, in which a word is split into several rows and columns, have the potential to conductively repair several cell upsets per word with a short decoding time.

3.1 DRAWBACKS

1) PDS codes have higher space, power, and delay overheads due to the complexity of their coding and decoding circuitry.

2) Two reasons why the interleaving approach would not be feasible in content-addressable memory (CAM) are the close connection of comparison circuit topologies and hardware structures from both cells.

3) A word can only have two faults corrected by the BICS approach.

4) In any situation, 2D MC can only fix two mistakes.

The new method known as FUEC-triple adjacent error correction (TAEC) may identify and repair burst mistakes of up to four bits, or it can correct errors in one bit, two adjacent bits, or three adjacent bits. Just one more bit of code will do this. In this instance, the FUEC–TAEC coding requires eight code bits for a 16-bit data word. This code's parity check matrix, H, is shown. Ci are the code bits and Xi are the principal data bits, much like in the FUEC-DAEC. Similar to this, designing the encoder/decoder circuitry is rather simple starting from H. However, this method will be regarded as having less precision and being unable to correct a huge number of data points.

4. PROPOSED WORK

In order to improve memory reliability, we present a novel technique in this study called Data Segmentation Section Code (DSSC), which is based on the divide-symbol. This low-cost implementation approach may be used to identify and rectify numerous transient defects in volatile memory.

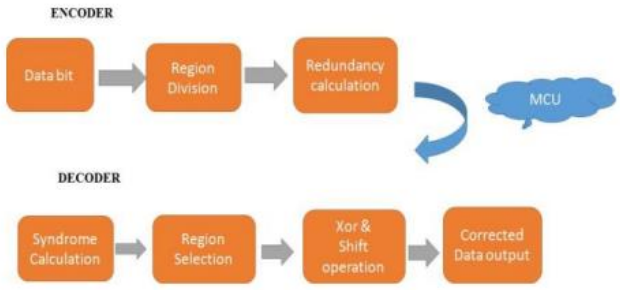

Fig -1: Proposed Work

An error correction code based on a two-dimensional code is found in the Data Segmentation section code. This code converts 16 data bits into 32 bits of information. In order to minimize the area and time conception, only parity bits are employed for data bit encoding.

4.1 Data Segmentation Section Code Encoding Process:

The 32 bits of data that the Data Segmentation Section Code has encoded are shown in Fig. 2. Data bits were represented by the gray pixels, which were separated into four groups (A, B, C, D).

A1	A2	A3	A4	Di1	Di3	CbA13	CbA24
B1	B2	B3	B4	Di2	Di4	CbB13	CbB24
C1	C2	C3	C4	P1	P3	CbC13	CbC24
D1	D2	D3	D4	P2	P4	CbD13	CbD24

Fig -2: Data model encoded with DSSC.

The green cells represent the diagonal bits (Di) that were examined using XOR operations in certain data bits:

$$Di_1 = A_1 \oplus B_2 \oplus C_1 \oplus D_2$$

$$Di_2 = A_2 \oplus B_1 \oplus C_2 \oplus D_1$$

$$Di_3 = A_3 \oplus B_4 \oplus C_3 \oplus D_4$$

$$Di_4 = A_4 \oplus B_3 \oplus C_4 \oplus D_3$$

Parity bits (P) were analyzed using XOR operations in the data bits columns of the cell that was colored blue:

$$P_1 = A_1 \oplus B_1 \oplus C_1 \oplus D_1$$

$$P_2 = A_2 \oplus B_2 \oplus C_2 \oplus D_2$$

$$P_3 = A_3 \oplus B_3 \oplus C_3 \oplus D_3$$

$$P_4 = A_4 \oplus B_4 \oplus C_4 \oplus D_4$$

$$CbA_{13} = A_1 \oplus A_3$$

$$CbA_{24} = A_2 \oplus A_4$$

$$CbB_{13} = B_1 \oplus B_3$$

$$CbB_{24} = B_2 \oplus B_4$$

$$CbC_{13} = C_1 \oplus C_3$$

$$CbC_{24} = C_2 \oplus C_4$$

$$CbD_{13} = D_1 \oplus D_3$$

$$CbD_{24} = D_2 \oplus D_4$$

4.2 The decoding process of DSSC is divided into three steps:

I. Syndrome appraisal of the redundancy bits - The syndrome appraisal consists of a XOR operation between the redundancy data stored and the recalculated redundancy bits (RDi, RP, and RCb). So the values for the Syndrome of Diagonal , Parity and Check bits are estimated by:

$$SDi = Di \oplus RDi$$

$$SP = P \oplus RP$$

$$SCb = Cb \oplus RCb$$

II. Verification of error decoding conditions: One of these two requirements must be met following the analysis of the syndromes in order for the error correction to be executed. (i) At least one value in the SDi and SP vectors is similar to one; (ii) Many SCb values were comparable to one. The criteria alter the method used to determine the error for the area of data bits.

III. The decoding process involves the selection and repair of an incorrect data area. Specifically, a specific region within the data bit is rectified. The regions that make up the region are displayed below.It was revealed that they divided the data bits into three areas in order to choose a specific set of bits for the repair procedure. This lessens the idea of time and space.

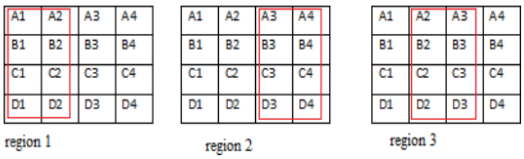

Fig -3: sections of data bits

5. XILINX VIVADO AND VERILOG HDL

5.1 HISTORY OF VERILOG

Verilog was first developed by Gateway Design Automation Inc. in 1984 as a proprietary hardware modeling language. There are rumors that the original language was created by incorporating elements from both conventional computer languages like C and the most well-liked HDL language at the time, HiLo. Verilog was not standardized at the time, and the language changed in practically every update that was released between 1984 and 1990. The Verilog simulator was initially put to use in 1985 and underwent significant expansion until 1987. Gateway's Verilog simulator served as the implementation. The first significant addition was Verilog-XL, which included a few new capabilities including the "XL algorithm," a very effective way to simulate gates at the hardware level.

5.2 DESIGN STYLES

There are two approaches to designing a hardware description language such as Verilog: top-down and bottom-up.

5.3 Bottom-Up Design

The conventional approach to electrical design involves developing from the bottom up, starting with transistors and working up to gates and, eventually, the system. However, when design complexity rises, new structural, hierarchical design techniques must replace conventional bottom-up designs.

5.4 Top-Down Design

It is straightforward and efficient to adapt this design-style for HDL representation. Among its numerous benefits are early testing, fabrication technology independence, structured system design, and more with a true top down design. But adhering to a strictly top-down design is rather challenging. Because of this, the majority of designs combine parts of both design styles and methodologies.

5.5 Features of Verilog HDL

- Verilog is sensitive to case.
- The freedom to combine various degrees of abstraction.
- A single language for all design, verification, and testing needs.
  Lowercase definitions of keywords are used in Verilog.
- The majority of Verilog's syntax is derived from the "C" language.

5.6 VLSI DESIGN FLOW

A formal specification of a VLSI chip is the first step in the VLSI design cycle, which proceeds through several stages to generate a packaged chip.

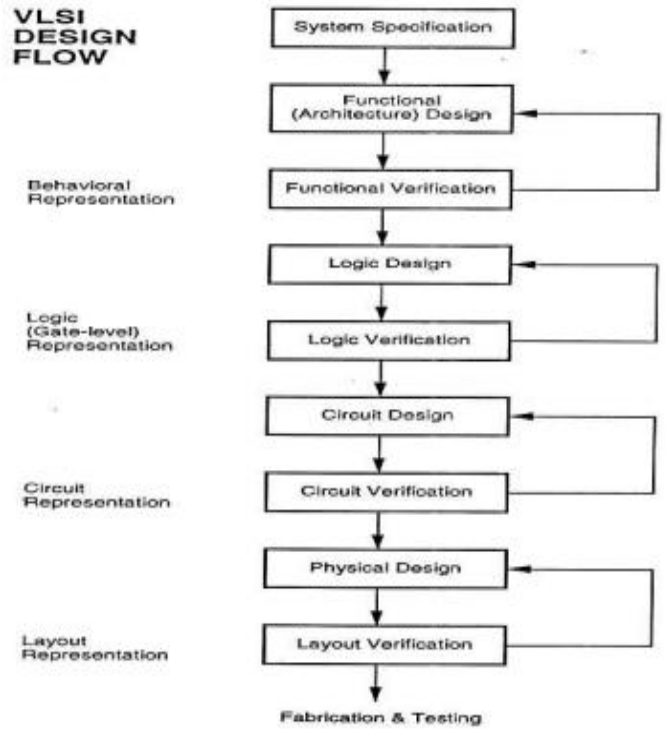

Fig -4: Flow Of Vlsi Design

6. RESULTS

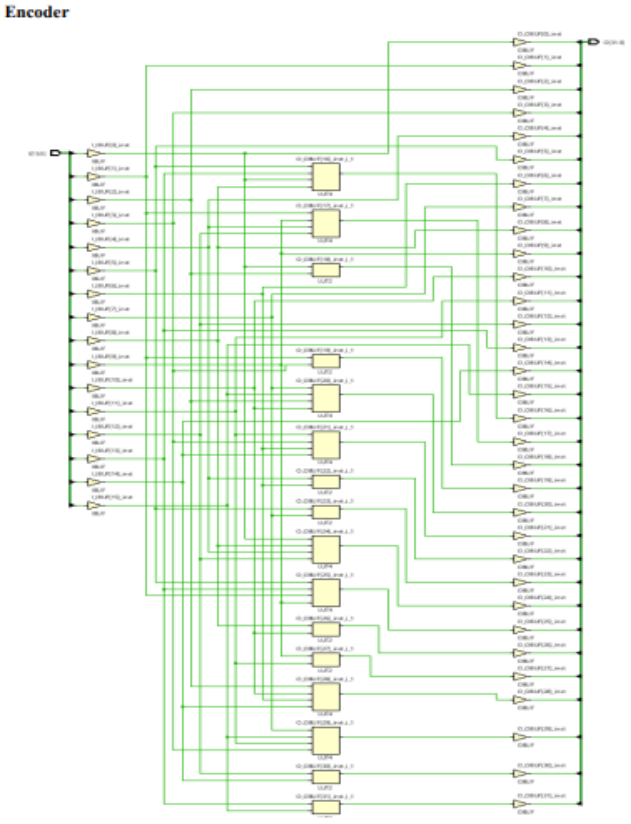

Fig -5: Encoder

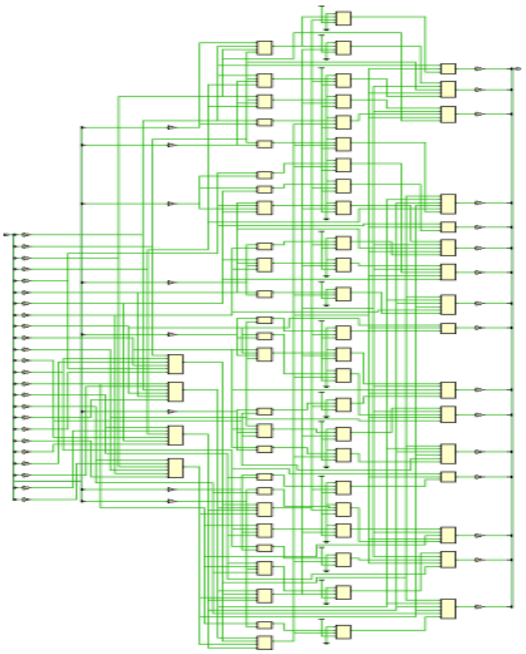

Fig -6: Decoder

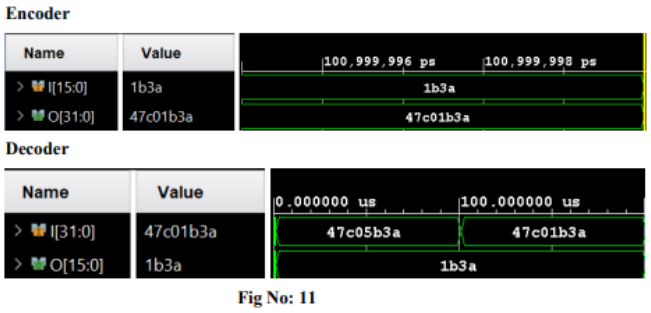

Fig -7: Simulation results for both encoder and decoder

Encoder

Name	Slice LUTs (134600)	Bonded IOB (400)
N Enc	8	48

Decoder

Name	Slice LUTs (134600)	Slice Registers (269200)	Bonded IOB (400)
N Dec	35	24	48

Fig -8: Area results for both encoder and decoder

Encoder

Name	Slack	Levels	High Fanout	From	To	Total Delay	Logic Delay	Net Delay	Requirement	Source Clock
Path 1	∞	3	4	I[1]	O[19]	4.659	3.368	1.291	∞	input port clock
Path 2	∞	3	4	I[5]	O[23]	4.659	3.368	1.291	∞	input port clock
Path 3	∞	3	4	I[4]	O[24]	4.659	3.368	1.291	∞	input port clock
Path 4	∞	3	4	I[8]	O[26]	4.659	3.368	1.291	∞	input port clock
Path 5	∞	3	4	I[6]	O[28]	4.659	3.368	1.291	∞	input port clock
Path 6	∞	3	4	I[11]	O[29]	4.659	3.368	1.291	∞	input port clock
Path 7	∞	3	4	I[12]	O[30]	4.659	3.368	1.291	∞	input port clock
Path 8	∞	3	4	I[13]	O[31]	4.659	3.368	1.291	∞	input port clock
Path 9	∞	3	4	I[8]	O[16]	4.638	3.347	1.291	∞	input port clock
Path 10	∞	3	4	I[12]	O[17]	4.638	3.347	1.291	∞	input port clock

Decoder

Name	Slack	Levels	High Fanout	From	To	Total Delay	Logic Delay	Net Delay	Requirement	Source Clock
Path 1	∞	5	16	I[17]	O[10]	6.068	3.576	2.492	∞	input port clock
Path 2	∞	5	16	I[17]	O[13]	6.068	3.576	2.492	∞	input port clock
Path 3	∞	5	16	I[17]	O[14]	6.068	3.576	2.492	∞	input port clock
Path 4	∞	5	16	I[17]	O[1]	6.068	3.576	2.492	∞	input port clock
Path 5	∞	5	16	I[17]	O[2]	6.068	3.576	2.492	∞	input port clock
Path 6	∞	5	16	I[17]	O[5]	6.068	3.576	2.492	∞	input port clock
Path 7	∞	5	16	I[17]	O[6]	6.068	3.576	2.492	∞	input port clock
Path 8	∞	5	16	I[17]	O[9]	6.068	3.576	2.492	∞	input port clock
Path 9	∞	5	20	I[17]	O[4]	6.052	3.560	2.492	∞	input port clock
Path 10	∞	5	20	I[17]	O[8]	6.052	3.560	2.492	∞	input port clock

Fig -8: Delay results for both encoder and decoder

3. CONCLUSIONS

This work suggests a novel error correction algorithm (ECC) to lessen data corruption in volatile memory systems. Xilinx Vivido, which was built in Verilog HDL, was used to simulate and synthesize the suggested design. The amount of energy used to encode and decode process is, respectively, 0.167W and 0.127W. In contrast to established techniques, this encoding-decoding procedure uses less power and requires less space and time. This method will also be expanded in order to lower the area, latency, and power usage. In comparison to other existing approaches, the decoder area rises since the zones were specifically chosen. Using the advanced region selection criteria further reduces this.

REFERENCES

[1] R. C. Baumann (2005), "Soft errors in advanced computer systems," IEEE Des. Test. Comput., vol. 22, no. 3, pp. 258-266.

[2]] C. L. Chen and M. Y. Hsiao (1984), "Error-correcting codes for semiconductor memory applications: A state-of-the-art review," IBM J. Res. Develop., vol.28, no. 2, pp. 124-134.

[3] E. Ibe, S. Chung, S. Wen, H. Yamaguchi, Y. Yahagi, H. Kameyama, S. Yamamoto and T.Akioka (2006), "Spreading diversity in multi-cell neutron-induced upsets with device scaling," in Proc. IEEE Custom Integrated Circuit Conf., pp. 437-444.

[4] P. Reviriego, J.A. Maestro and C. Cervantes (2007), "Reliability analysis of memories suffering multiple bit upsets," IEEE Trans. Device Mater. Rel., vol. 7, no. 4, pp. 592-601.

[5] G. C. Yang (1995), "Reliability of semiconductor RAMs with soft-error scrubbing techniques," IEEE Proc. Computers and Digital techniques, vol. 142, no. 5, pp. 337-344.

BIOGRAPHIES

K.G.VENKATA KRISHNA , Assistant Professor , Krishna University College of Engineering &Technology , Krishna University, Machilipatanam

BOTCHU.LAVANYA, Student of Department of Electronics and Communication Engineering , Krishna University, Machiliptanam.

KODAVALI.SRAVANI, Student of Department of Electronics and Communication Engineering , Krishna University, Machiliptanam.

KUNCHEPUJAGADEESH, Student of Department of Electronics and Communication Engineering , Krishna University, Machiliptanam.