Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 8, No. 2, August : 2023

A 1.8V, ENHANCED DECIMAL MATRIX CODE FOR IMPROVED ERROR DETECTION AND AMELIORATION OF MEMORY RELIABILITY AGAINST MULTIPLE CELL UPSETS

M. Madhusudhan Reddy, Research Scholar, in ECE, J.S University, Shikohabad, U.P.
Dr. Ramgopal, Department of Electronics communication engineering, JS university, UP

Motivation:

Portability has been playing a pivotal role in electronic devices. Low throughput is considered as fundamental requirement once is a myth. Portable devices has became hand held multimedia terminals with audio and video processing at higher rates, motion gesture, facial recognition system and handwriting capabilities. These applications must be capable of operating at mega and giga hertz rate and high throughput is fundamental requirement..Portable devices always embed with lowpower techniques for consumption of low power. The methodology suggested in this paper is a novel decimal matrix code. The suggested code is based on the divide-symbol to increase memory reliability.The suggested DMC employs decimal integer addition (decimal algorithm) on binary code separated symbols. This has been shown in works including the destruction of Java Virtual Machines, cryptographic protocols, and smart cards. This paper presents enhanced decimal matrix code (DMC0 for improved error detection and amelioration of memory reliability against multiple cell upsets (MCU). The decimal algorithm improves the efficiency of the code's error detecting capabilities. To improve memory efficiency, a novel decimal matrix code (DMC) based on the divide-symbol is suggested in this thesis. To identify mistakes, the suggested DMC employs a decimal algorithm (decimal integer addition and decimal integer subtraction). The benefit of using a decimal algorithm is that the error correction capacity is maximized, increasing memory reliability. Furthermore, the encoder-reuse strategy (ERT) is suggested to reduce the region overhead of extra circuits (encoder and decoder) without interfering with the whole encoding and decoding processes, since ERT uses the DMC encoder as part of the decoder.Xilinx Vivado design suite was used for simulation and synthesis. 1.8V Low Voltage CMOS (LVCMOS) Virtex 7 FPGA was used for testing the code.

Keywords: Soft memory glitch,Portable,Lowvoltage,Modified-DMC,FPGA

Introduction:

Modern memory chips may be made with smaller and more sophisticated circuits that operate at low voltage levels and have high storage capacities. These advances also raised the likelihood of soft memory glitches,in which unpredictable bits flip and corrupt the contents of the affected memory cells[5,6]. Soft memory losses may be triggered by power outages, gamma rays, and manufacturing flaws. Soft memory errors will become more common in the future. As the amount of cosmic rays rises with height, soft memory errors may trigger serious problems in avionics and space science. Memory cell corruption may have serious implications for algorithms.A single error in a sorted sequence, for example, will cause a regular binary search to finish up (n) cells away from the exact location. Soft memory failures may be used to compromise the security of information programmes.

Proposed Method:

In the proposed work, we are using carry look ahead [7]as the primary component for the CSA building. The carry pick adder is described by a static and portable multi output CLA.

Area-Efficient Carry Select adder:

In the design of very large scale integrated (VLSI) circuits, power consumption is a critical reliability factor. Furthermore, with the exponential growth of VLSI technology, the need for and success of handheld devices has pushed designers to aim for smaller silicon areas. Adder is the name

of the central electronic circuit that is used for extension. Adders are essential components of a broad range of digital systems. There are many adders, however quick adding with low area and power is still difficult. There are various kinds of adders, such as the ripple carry adder (RCA), the carry skip adder (CSKA), the carry look ahead adder (CLA), the carry save adder (CSLA), and so on one of them RCA has a smaller design, but their computing period is longer. It has the slowest speed of all adders since it has a long propagation delay but takes up less space. Then, in CLA, the effect is quick, but it leads to an increase in region; among these adders, CSLA has a small area, but the delay is increased due to the ripple carry adder. A carry-select adder is usually made up of two ripple carry adders and a multiplexer. Adding two n-bit numbers with a carry-select adder needs two adders (thus two ripple carry adders) to execute the calculation twice[8], once with the assumption of the carry being zero and once with the assumption of the carry being one. After calculating the two results, the correct amounts as well as the correct carry are chosen with the multiplexer until the correct carry is determined. Each carry select block may have a fixed or variable number of bits. When the block size is variable, it can have a delay from extension inputs A and B to the carry out equivalent to the delay of the multiplexer chain preceding it, such that the carry out is measured just in time.

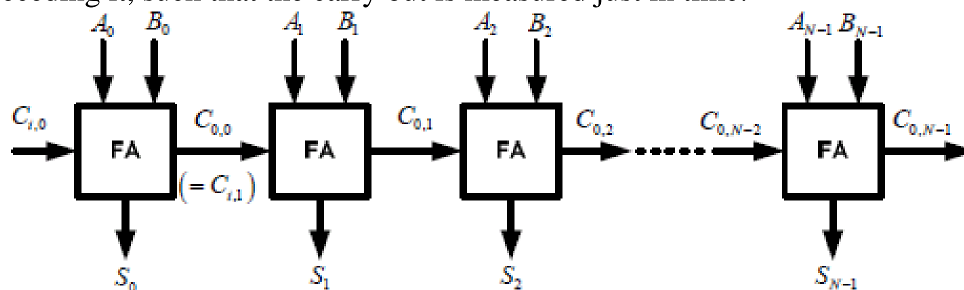


Fig 1: The N-bit carry ripple adder constructed by N set single bit full-adder

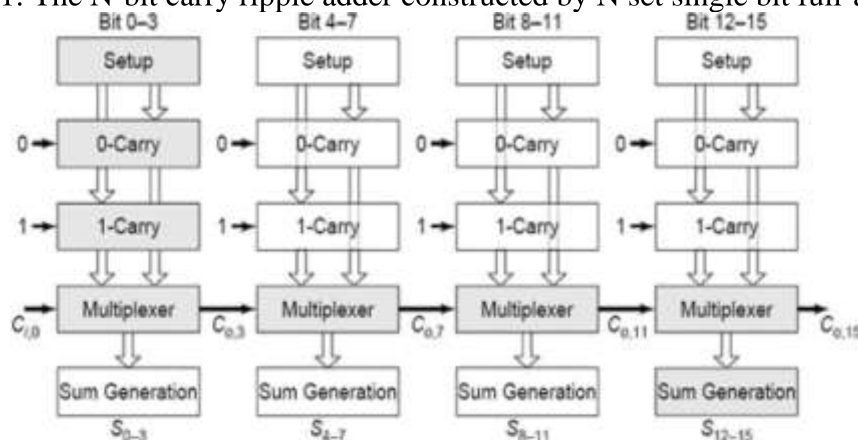


Fig 2:Selecting a 16 Piece Snake to Divide a Patent into 4 Pieces

Binary Error Detection:

The key concept of this paper, as mentioned above, is to use transistor level changed BEC instead of ordinary BEC with cin=1 in order to reduce the area and power consumption of the CSLA. An n-bit transistor level changed BEC is needed to substitute the n-bit ordinary BEC. Table I portrays the feature table of a 3-b BEC. The basic feature of the CSLA is shown in Fig.1 by utilising the 4-bit BEC in conjunction with the mux. The BEC output is fed into one of the 8:4 mux's inputs (B3, B2, B1, and B0), and the mux's other input is the BEC output. The two potential partial effects are generated in parallel, and the mux is used to pick either the BEC output or the direct inputs based on the control signal Cin[10,11]. The value of BEC logic derives from the significant silicon area reduction when designing a CSLA with a large number of pieces.

INPUT[0:3]	OUTPUT[0:3]
000	001
001	010
010	011
011	100
100	101
101	110
110	111
111	000

Table 1: FUNCTION TABLE OF THE 3-B BEC

Limits of Simple Binary Error Detection:

The chance of detection error is reduced in the binary error detection system presented, despite the fact that it demands a small number. The predominant explanation for this is that the error detection mechanism is designed on two processes. Assume that the reductions B3, B2, B1, and B0 are the surplus's original data, and that the reductions C0 and C1 are the additional fractions shown in the figure. A binary algorithm is used to produce C0 and C1 reductions. The findings reveal that errors B2 and B0 are determined wrongly, as they were at the start, and thus all reductions are incorrectly corrected.

Advantage of Decimal Error Detection:

It was discovered in the preceding debate that algorithm tracking-based errors can only identify a small number of errors. These errors, though, may be observed using an error detection algorithm in order to prevent coding errors. This is due to the fact that the A decimal algorithm's operating system differs from a Binary system[14,15]. Fig 3 depicts the image used to identify computational errors in the proposed scheme.

First, the small fractions obtained from the original dataset of symbols 0 and 2 correspond to H4H3H2H1H0

H4H3H2H1H0= D11D10D9D8+ D3D2D1D0

=0111+1111

=10110

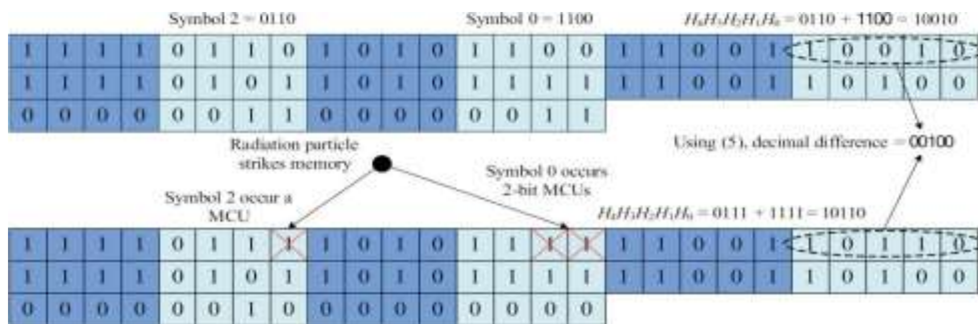


Fig 3: Advantage of Recognizing a Value Error Using a Small Number of Values

If the MCU consists of 0 characters and 2 characters, that is, 0 characters from "1100" to "1111" is bad ("D3D2D1D0 '= 1111), the remaining 2 characters are" 0110 "(D11D10D9D8' =) "0111 "- Size, H4H3H2H1H0 'obtained by scanning, calculates the following.

H4H3H2H1H0= D3D2D1D0+ D11D10D9D8

=1100+0111

=10010

The physical rating of H4H3H2H1H0 can be obtained using strong numbers.

Δ H4H3H2H1H0= H4H3H2H1H0’ -H4H3H2H1H0
=10110-10010
=00100

The decimal value is not "0," indicating that the mistake is between 0 and 2. The precise location of the short traps can be calculated using the vertical symbols S3 - S0 and S11 - S8. Finally, in Section, both of these errors may be reversed (7). As a result, decimal values provide more stability in protecting the proposed MCU memory. As a consequence, regardless of the detected, each single and double error in each column, as well as multiple forms [21,22,23]of faults, may be resolved according to the guidelines.

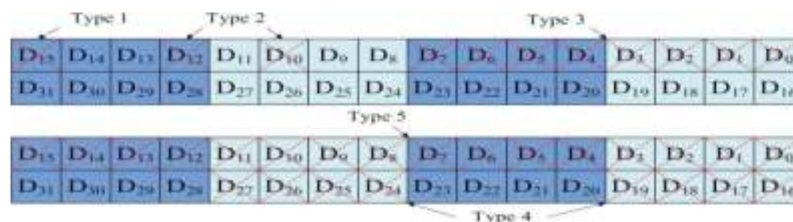


Fig 4: The image version of the MCU may be compatible with the proposed DMC

Since the key elements of the DMC are one-way and multi-error correction using two-dimensional labels, the proposed DMC will address problems of types 1, 2, and 3. Figure 4.6 depicts the error variations 4 and 5. As seen in the diagram. Each age has a range of transformational weaknesses. Indicate that the suggested approach would shield the memory from several MCUs in an interesting way. However, it is necessary to remember that for angles of types 4 and 5, errors will occur if the following components are present at the same time (these errors are still different).

1) The product of the integers 0 and 2 is 2 m -1.

2) Both variants of markers 0 and 2 are unlucky.

More information is shown in the Fig 4 Since these two components are designed according to the DMC setup and default processes, H4H3H2H1H0 and H4H3H2H1H0 'are calculated as follows.

H4H3H2H1H0= D3D2D1D0+ D11D10D9D8
= 0110+1001
=01111

H4H3H2H1H0’= D3D2D1D0’+ D11D10D9D8’
=1001+0110
=01111

H4H3H2H1H horizontal symptoms may present

Δ H4H3H2H1H0= H4H3H2H1H0’ -H4H3H2H1H0
=0111-0111
=0000

This result means that at points 0 and 2 nothing went wrong and the memory failed. However, this case is rare. For example, if m = 4, the probability of a correction is incorrect.

$$P_{\Delta H=0} = 4 \times \left(1/2^4\right)^2 \times P_{MCU8} \approx 0.001.$$

If $m = 8$

$$P_{\Delta H=0} = 4 \times \left(1/2^8\right)^2 \times P_{MCU16} \approx 0.0000011.$$

Fig 5: Improved performance of DMC

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 8, No. 2, August : 2023

Many variables affect the tempo at which a device executes. Output can be improved by utilizing quicker circuit technologies to construct the processor and main memory[12,13]. Another option is to arrange the hardware such that many operations can be done at the same time. As a result, the amount of operations conducted per second increases even though the time required to execute any one operation remains constant. This is the fundamental idea behind machine pipelining.

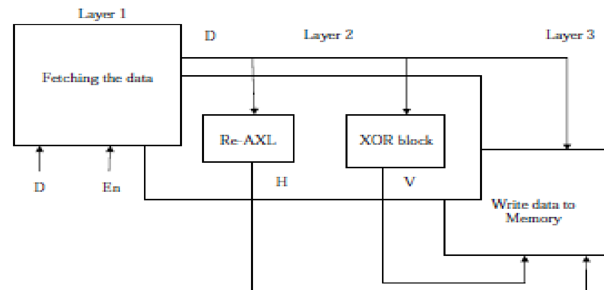


Fig 6: Pipelined MDMC Encoder

The Pipelined MDMC divides the entire DMC into small units that operate concurrently asshown in Fig 6 and Fig 7. The DMC system[35,36] is made up of read and write operations. The inputted data must first be processed in order to calculate horizontal and vertical redundant bits. The data, as well as the vertical and horizontal redundant bits, will then be stored in memory. This is referred to as the write operation. The written data, as well as the calculated vertical and horizontal redundant bits, should then be retrieved.

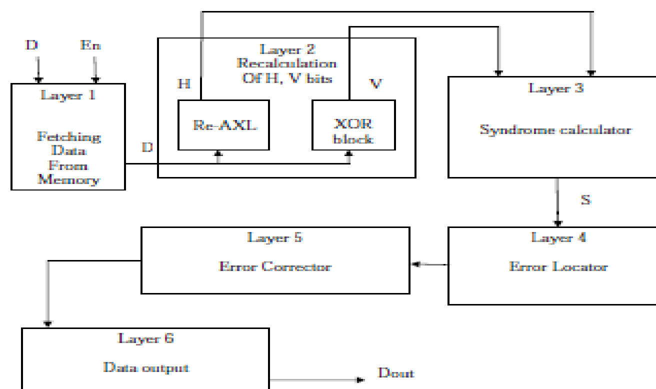


Fig 7: Pipelined MDMC Decoder

The presence of different layers is the main reason for this. The Re-AXL blocks are used in this pipelined structured MDMC to calculate the horizontal redundant bits. This is far superior to the adder blocks used in DMC systems. This replacement aids in reducing the number of redundant bits. This also has another benefit. The rippling in the calculation of the horizontal bits will be reduced by replacing the adder with the Re-AXL block. This will also help to reduce the amount of time spent waiting. When the rippling is reduced, so will the power consumption. The Pipelined MDMC also makes use of the Encoder Reuse Technique (ERT)[6,7,8]. This is due to the fact that either encoding or decoding is performed at the same time, and thus the encoder can be reused in the decoding as well. The complexity of the pipelined structure can be reduced by modifying the ERT. However, by modifying this method, the system can function as both an encoder and a decoder at the same time[29,30]. An En (enable) signal is used to determine the mode of operation of the encoder.

Proposed Modified-DMC:

To improve memory efficiency, modified-DMC is proposed in this portion. This method employs Hamming codes and a decimal algorithm, which improves error identification and correction.

Block diagram of proposed modified-DMC:

Fig 8 illustrates the proposed block diagram of modified-DMC. In this case, the encoding method may be thought of as writing data into SRAM cells and the decoding process as interpreting data from SRAM cells. The knowledge bits D are fed into the DMC encoder, which generates the horizontal and vertical search bits H and V.

When the encoding step is over, the modified-DMC code is placed in SRAM cells. In the register U, a duplicate of knowledge bits D is placed alongside these bits. MCUs can now arise when these cells are subjected to radiation. During the decoding phase, these MCUs are corrected. To identify mistakes, the suggested security code used a decimal algorithm. Since the decoder employs the ERT method, the field overhead of additional circuits is greatly decreased.

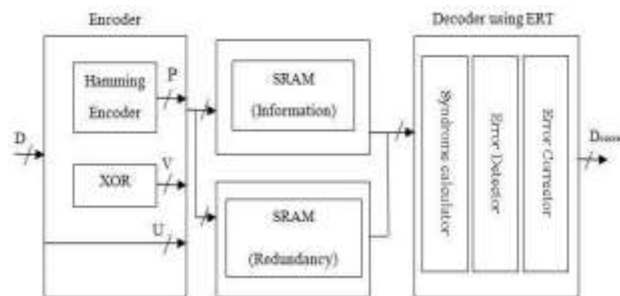
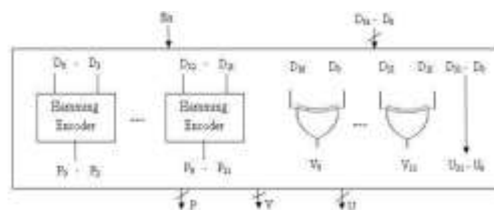


Fig.8: Block diagram of modified-DMC

Encoder of modified-DMC:

The proposed modified-DMC encoder is identical to the current DMC encoder except that it uses Hamming codes to calculate horizontal redundant bits. Figure 4 depicts an encoder circuit that employs a Hamming encoder and XOR gates. To start, an N-bit information word is used as the source data. This N-bit term is broken down into k-symbols of m-bits each, yielding N = k*m. These k-symbols are grouped in a k1xk2 2-D matrix structure in such a way that k = k1xk2. Here, k1 denotes the number of rows and k2 denotes the number of columns. Second, for each symbol in the first row, the horizontal search bits are computed using Hamming codes[39]. Each symbol is regarded as a decimal integer in this context. Third, for each column, the vertical search bits are calculated using a binary exclusive OR operation.

To understand the proposed DMC scheme, implementation of a 32-bit DMC code is explained as an example in Fig 8. The cells numbered from D0 to D31 are information bits. This 32-bit word (N = 32) is divided into eight symbols (k = 8) of 4 bits (m = 4) each. The cells numbered from P0 to P11 are horizontal check bits and from V0 to V15 are vertical check bits. The maximum correction capability and the number of redundant bits are different when different values for k and m are chosen. For example, if k = 2x4 and m = 4, it can correct maximum of 16- bit errors and the number of redundant bits are 28. Therefore, the values of k and m should be selected carefully.



Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 8, No. 2, August : 2023

Fig 9: Modified-DMC encoder

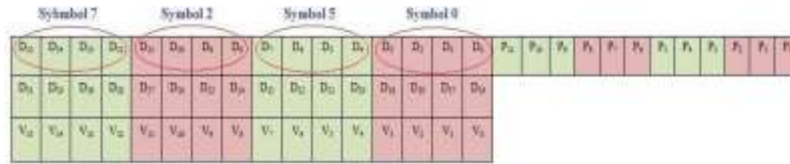


Fig 10: Logical representation of 32-bit modified-DMC

The horizontal redundant bits are calculated for all the symbols in only one row. The following equations represent the Hamming equations to calculate the horizontal check bits.

P0 = D3 ^ D1 ^ D0
P1 = D3 ^ D2 ^ D0
P2 = D3 ^ D2 ^ D1
…….and so on Ti the following equations are used to compute the vertical check bits.
V0 = D0 ^ D16
V1 = D1 ^ D17 ……… and so on the symbol "^ "indicates Binary Exclusive-OR operation.

Decoder of modified-DMC:

The decoding process is a step-by-step process that includes a syndrome converter, error locator, and error corrector[40].

Horizontal syndrome bits are computed using decimal integer subtraction, and vertical syndrome bits are computed using the exclusive OR service. Non-zero horizontal syndrome bits signify error recognition, whereas non-zero vertical syndrome[42] bits indicate error spot. The error corrector block uses XOR operations to correct these errors.

The horizontal syndrome pieces are acquired in the following manner:

▲ P0P1P2D3D2D1D0 = P0P1P2D3D2D1D0' – P0P1P2D3D2D1D0
▲ P3P4P5D7D6D5D4= P3P4P5D7D6D5D4' – P3P4P5D7D6D5D4….and so on

The vertical syndrome bits are obtained as follows:
S0 = V0' ^ V0
S1 = V1' ^ V1 and so on
The errors can be corrected by D0 correct = D0' ^ S0 … and so on

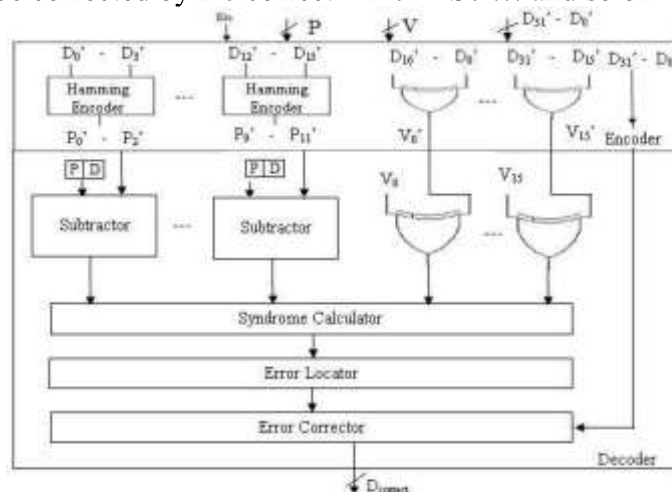


Fig 11: Modified-DMC decoder

The roles of the En signal are shown in table II. The allow signal En may be used to decide if the encoder must be included in the decoder. As a result, the En signal separates the encoder from the decoder, and it is regulated by the write and read signals in memory.

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 8, No. 2, August : 2023

Extra circuit	En signal		Function
	Read signal	Write signal	
Encoder	0	1	Encoding
	1	0	Compute syndrome bits

Table II: Function of EN signal

Algorithm,Simulation and Synthesis Results:

Algorithm programming sequentially executes standardized instructions to solve a query. The DMC algorithm with 32 bits. Code was written in hardware Description Language, Simulated and synthesized using Xilinx Vivado 2019.2

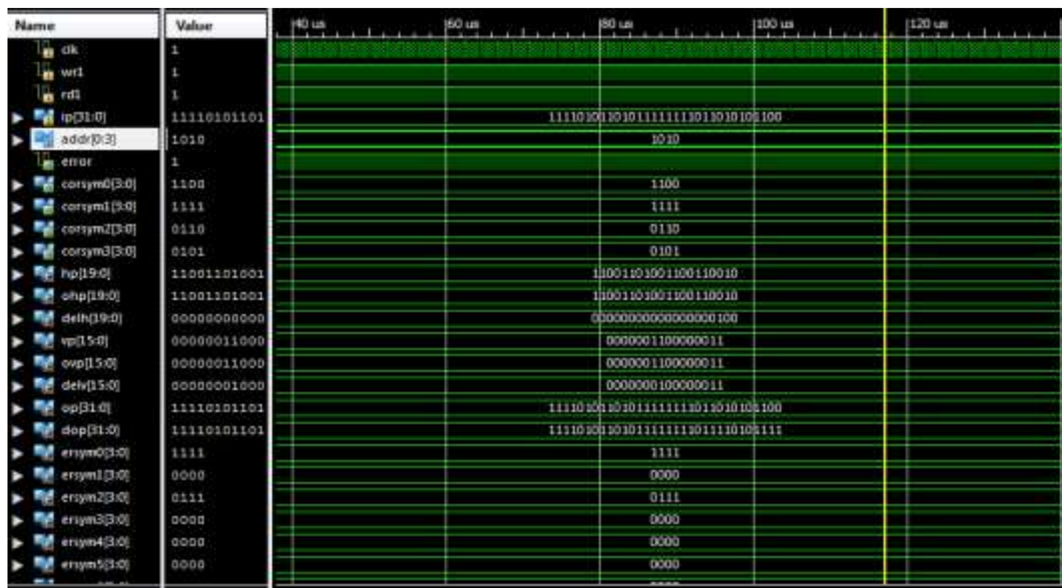


Fig 12: Simulation Results from Previous System

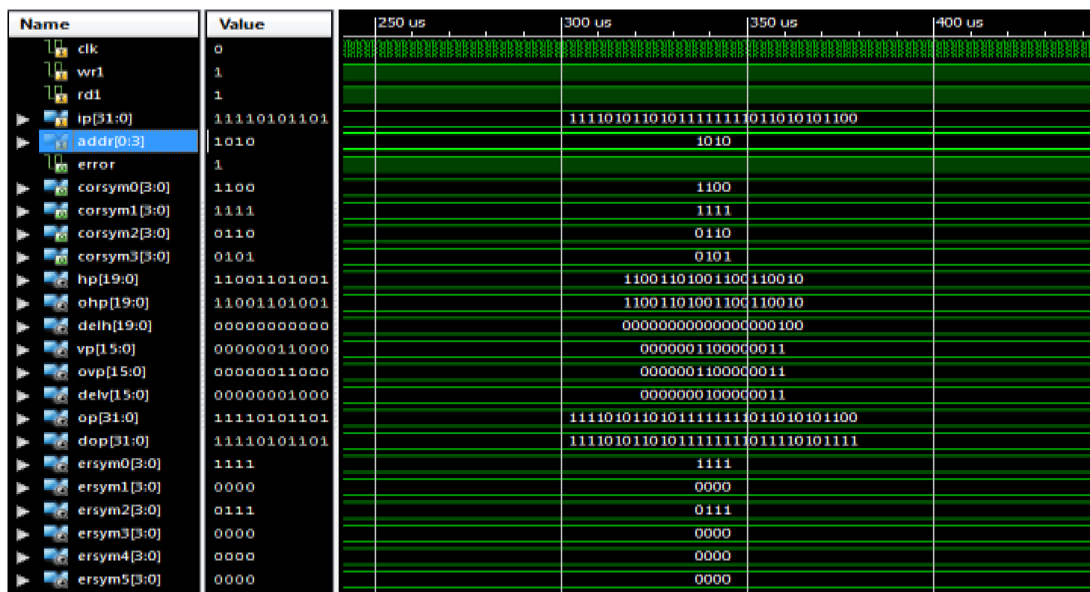


Fig 13: Simulation Results from Proposed System

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 8, No. 2, August : 2023

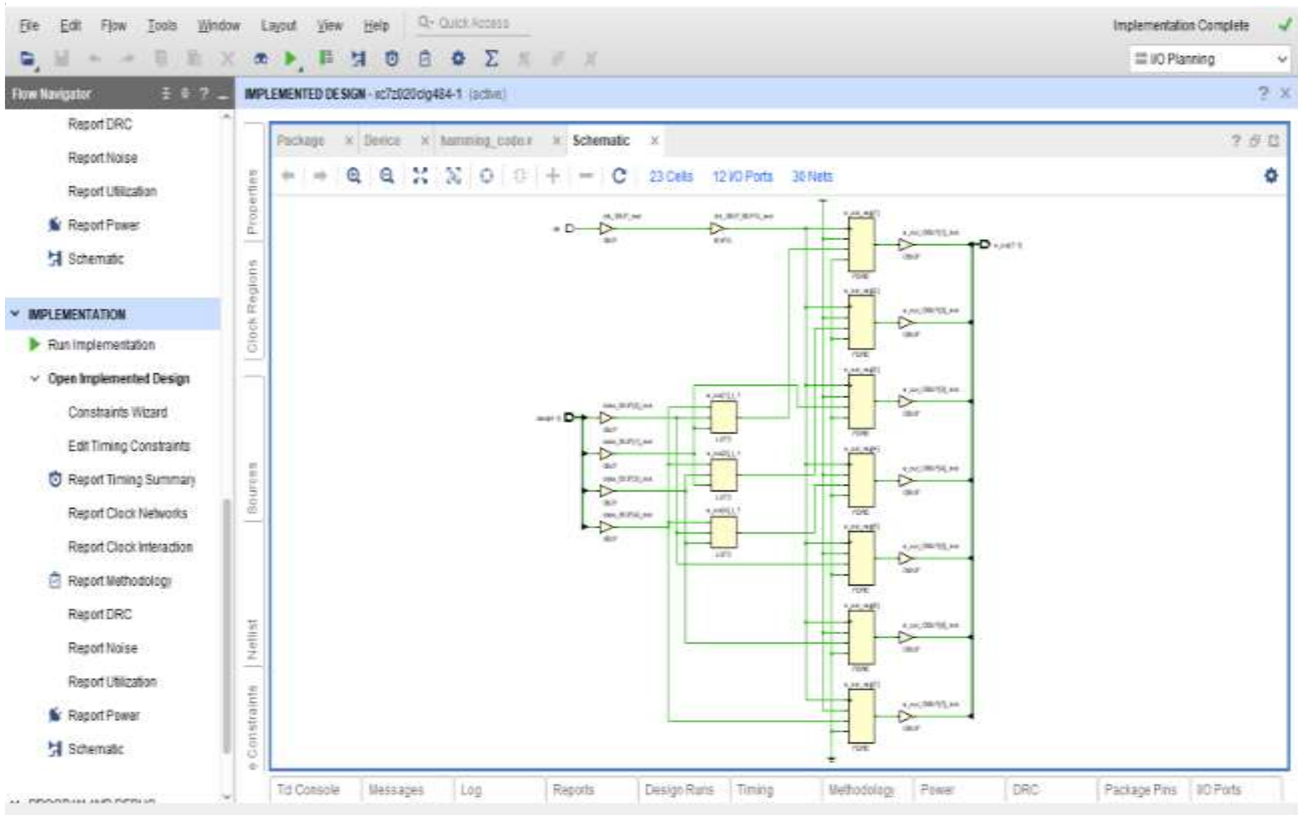


Fig 14:Hamming Code Implemented using Virtex 7

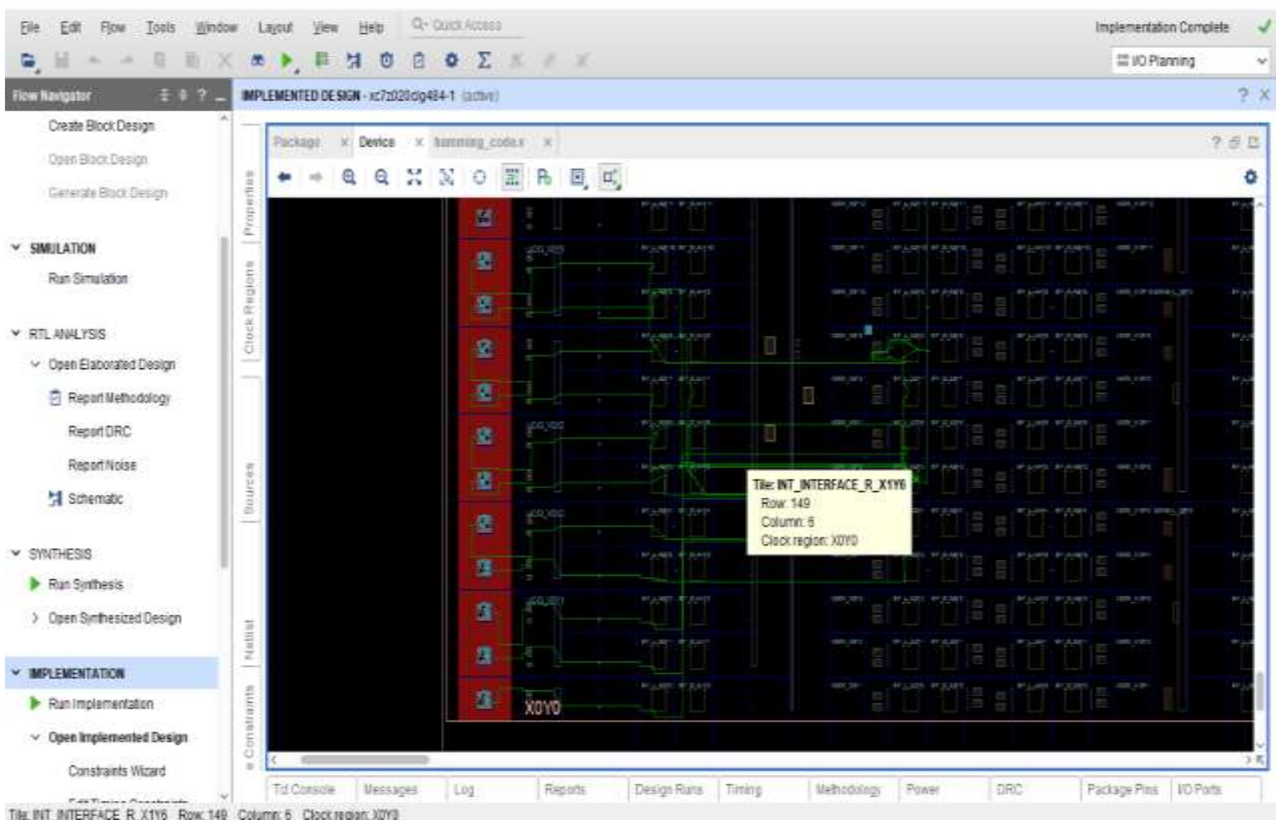


Fig 14: Implemented design in Virtex 7

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 8, No. 2, August : 2023

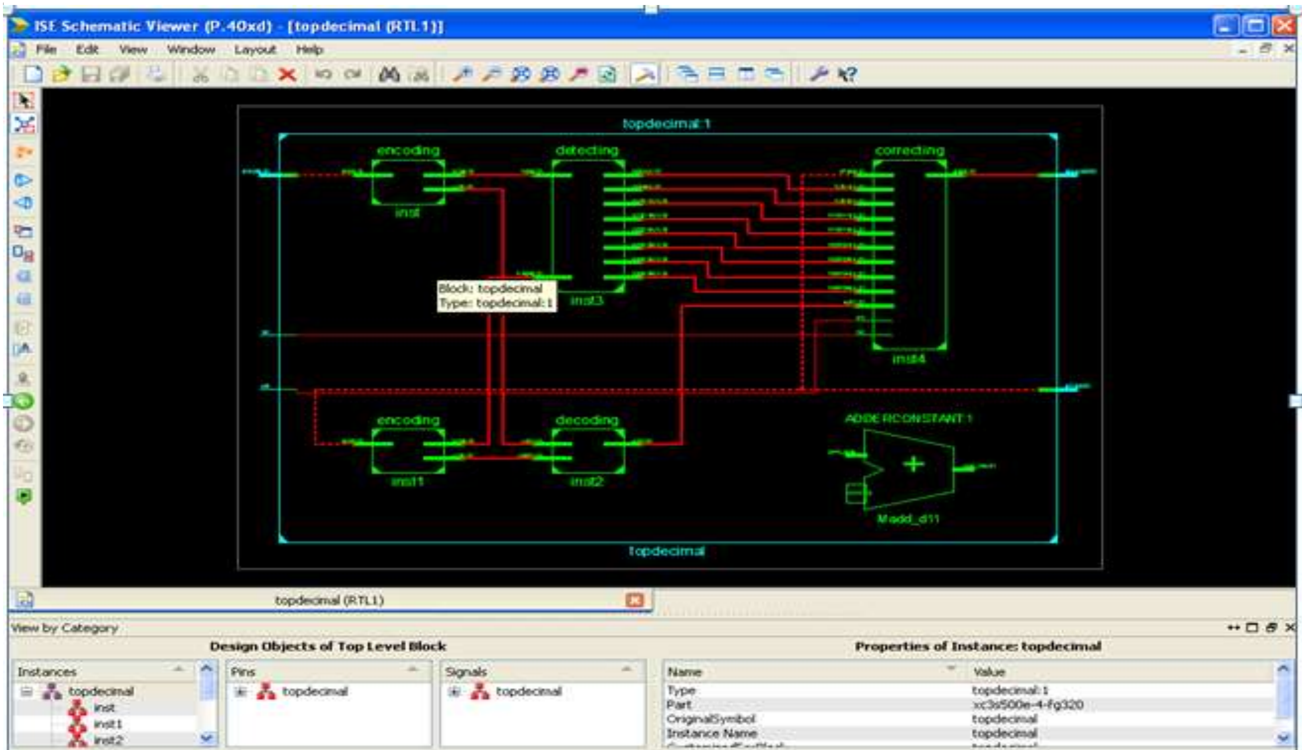


Fig 15: RTL Schematic of 32 bit DMC

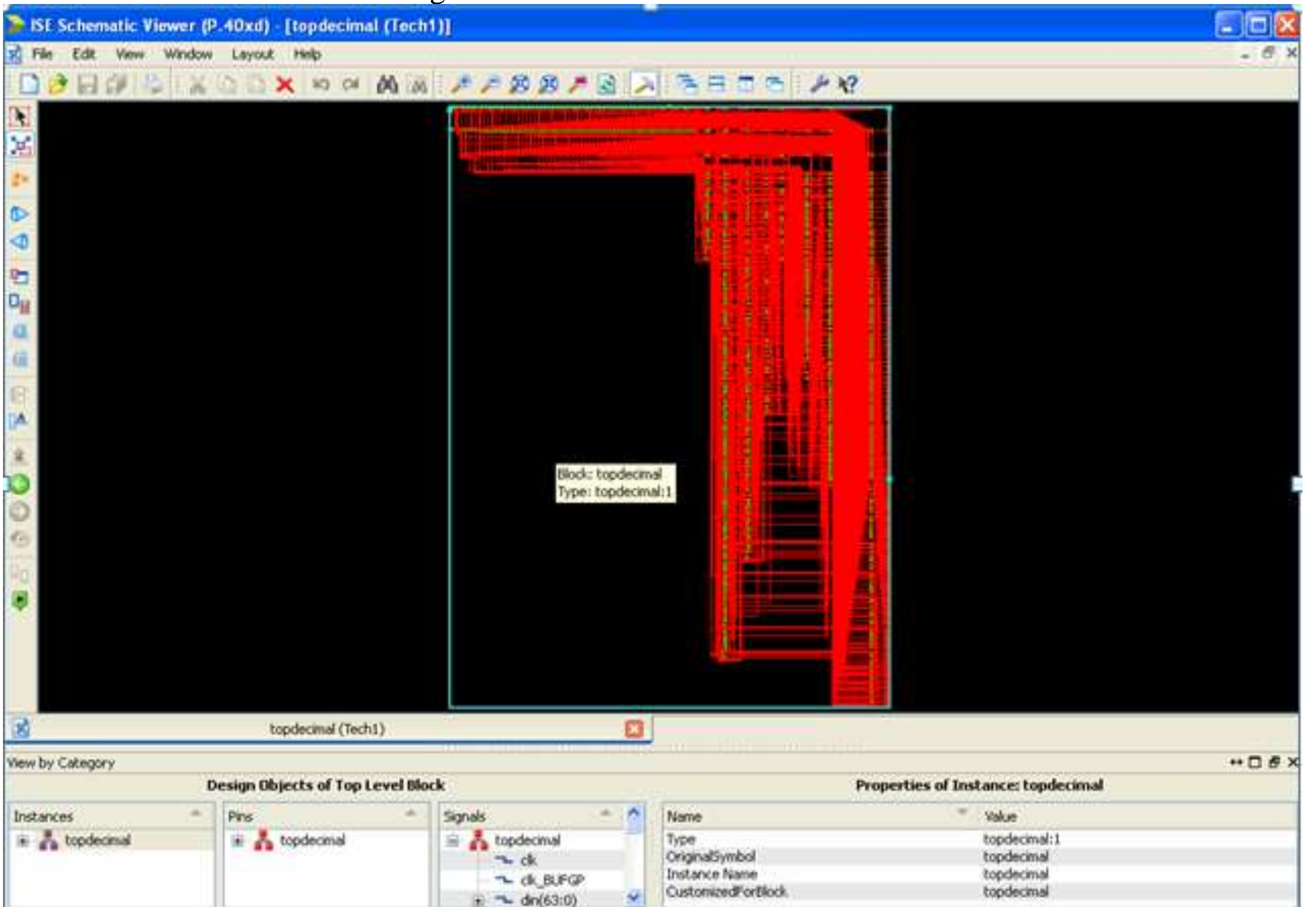


Fig 16: Technology Schematic of 32 bit DMC

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 8, No. 2, August : 2023

Results Comparision for Existing Methods and Proposed Method:

ECC Codes	Area		Power			Delay
	μm2	%	mw	%	ns	%
DMC	41572.6	100	10.8	100	4.9	100
PDS	486778.1	1170.9	2211.1	2047.2	18.7	381.6
MC	77933.7	187.5	24.7	228.7	7.1	144.9
Hamming	58409.4	140.5	20.5	189.8	6.7	136.7

Table III: Area, delay, power analysis of DMC

PARAMETRES	EXISTING METHOD	PROPOSED METHOD
Latency	3.158ns	2.179ns
Frequency	316.676MHz	458.905MHz
Power	0.08067mW	0.01850mW

Table IV: Comparison of Latency, Frequency and Power

Conclusion:

The majority of error correction codes (ECCs) are used to shield the memory in the MCU from corruption, but the key issue is that they must wake up and function. The matrix code (MC) for the encryption code for Hamming memory was recently revised. To maintain continuity, a new DMC name has been added. The security code employs a computer to identify, enable for the identification of, and fix errors. The findings demonstrated that the machines used had adequate security for big MCUs in memory. Furthermore, the error detecting techniques used in accessing MCU CAM are a smart idea since it can be implemented into the CSA to include protection controls further Comparison results clearly shows reduction in latency, Frequency enhancement and Low power. Simulation and Synthesis results shows significant decreasing in area and latency, Furthermore LVCMOS used is 1.8V in Vivado for Virtex-7.

References:

1. D. Radley, H. Puchner, S. Wong et al. nuclear. C, MMP. 52, No. 6, pp. 2433-2437, December2005.
2. H. Hostility, h. Taniguchi, Y. Yahagi, K. Shimbo, and T. Electronic Applications, Vol. 57, no. 7, pp. 1527–1538, July2010.
3. C. Argyroid and d. K "Advanced Encryption Algorithm for Reliable Reed-Muller Code" in Proudhon. IEEE Int. Chip Explosion Crisis, September. 2007, p.95-98.
4. Sh. Liu, pp. Rivirego and J. of. The numbers are huge. (Welsh) Cyst. volume. 20, no. 1, pp. 148–156, January2012.
5. M. Shaw, L. Y. Xiao, L.; L. Song, J. Zhang, et al. WW J., MP. 42, no. 3, pp. 553- 561, March2011.
6. R. Nasir and J. "Improving Code Design to Reduce Multidisciplinary Downloads in SRAM are Two Similar Errors" in Dropper, Proc. Skin 34. -Politicians and Governors, September 2008, pp.222-225.
7. G Neuberger, d. L. Castensmid et al. Rice, "Automated Code Optimization Technology to Improve Error Memory for Errors," Computer IEEE Design Test, Vol. 22, no. 1, pp. 50-58, January-February.2005.
8. Nkiruka, M. Flanagan, and J. his. Maestro, "Trinity Repair Code (32,45) for Application Storage," IEEE Trans. Ahh. Rel., Vol. I 12, no. 1, pp. 101-106, March 2012.
9. S. Begt, S. Wen and R. Wong, "Connection Options and a Soft Forgiveness Model," IEEE Trans. nuclear. C, MMP. 56, No. 4, pp. 2111–2118, August2009.
10. K. Pagiamatsis et al. 41, no. 3, pp. 712-727, March2003.

11. Sh. Bagrio, S. Van, and R. Circle engaged. M, m. Books, Vol. 57, no. 4, pp. 814- 822, April2010.

12. Page Rivirego and J. it. Maestro, "SRAM and BIX Multi-Bit Correction Error Codes," ACM Trans. Design automatically. Electronics for Chemistry. Sist. Vol. 14, no. 1, pp. 18: 1–18: 10, 2009January.C. Argyroid, r. Chipna, P. Vargas and. K Rel., Vol. 60, no. 3, pp. 528-537, September 2011.

13. Old man, d. K Pradhan and T. Kokak, "Matrix Codes for Reliability and Efficiency in Mutual Memory", IEEE Trans. Large numbers. (Welsh) Cyst. volume. 19, no. 3, pp. 420-428, March2011.

14. C. Argyroid, C.S.. Lisbon, d. K Pradhan and L. Caro, "Thorough correction of classification with a minimal delay algorithm." IEEE Latin Amar. Trial, March. 2009, pp.652-657.

15. Yahagi, H .; Yamaguchi, E .. Hostility, h .; Kamayama, M. Sato, T .; IEEE Trans Nuclear C., Vol. 54, No. 4, p. 1030-1036, August 2007.

16. c. Argyroid, p. Ribirgo, d. K Pradhan and J. of. Maestro, "Thesis-Based Approach to Correcting Adjacent Errors," IEEE Trans. nuclear. C, MMP 57, No. 4, pp. 2106- 2111, August2010.

17 Proc IEEE Int. Ulsarani and T. Chen, "Extra-Large TEC-QED ECC on a Chip for Single-Chip Memory Systems". Accelerate. calculate. Design, the number is very large. (Wales) Cyst. calculate. Laws., October 1994, pp.132-137.

Y. He, C. H. Chang, and J. Gu, "An area efficient 64-bit square root carry-select adder for low power applications," in Proc. IEEE Int. Symp. Circuits Syst., 2005, vol. 4, pp. 4082–4085.

17. A.M. Saleh, J.J. Serrano, and J.H Patel, "Reliability of Scrubbing Recovery Techniques for Memory Systems," IEEE Trans. on Reliability, Vol. 39, NO. 1, pp. 114-122, 1990.

18. C. Argyrides, H.R. Zarandi, D.K. Pradhan, "Multiple Upsets Tolerance in SRAM Memory," ISCAS07, New Orleans, USA, 27-30 May, 2007.

19. C.S Wallace, "A Suggestion for a Fast Multiplier" in IEEE Trans. On Electronic Computer, Volume EC-13, Issue 1,Page(s):14 17 Feb. 1964

20. D.K Prahdan, S.M. Reddy "Error-Control Techniques for Logic Processors", in IEEE Trans. on Comp. Vol C-21, No12, Dec 1972

21. D.Rossi, C. Metra "Error correcting strategy for high speed and density reliable flash memories" in IEEE J. Electronic Testing, Theory and applications. Vol. 19, No.5. pp 511-521 Nov 2003

22. K. Kimura et al, "Power reduction in megabit DRAM's, in IEEE Journal of Solid State Circuits, Vol SSC-21, pp. 381-389, June 1986.

23. M. Margala and N.G. Durdle, "Noncomplementaty BiCMOS logic and CMOS logic styles for low-voltage operation- A comprehensive study". In IEEE J. Solid State Circuits, vol 33, pp1580-1585, Oct 1998.

24. M. Nicolaidis "Time Redundancy Based Soft-Error Tolerance to Rescue Nanometer Technologies" in 17thIEEE VLSI Test Symposium. p. 86, 1999

25. P. Hazucha et al. "Measurements and Analysis of SER- Tolerant Latch in a 90-nm Dual-VT CMOS Process" in IEEE Journal of Solid-State Circ., Vol. 39, No. 9, Sept 2004

26. S. Mitra et al., "Robust System Design with Built-In Soft- Error Resilience" , in IEEE computer society, Vol. 38, No.2 pp. 43-52 , Feb. 2005

27. S. Bhattacharjee and D. K. Pradhan, "LPRAM A novel Low-Power High- Performance RAM design with testability and Scalability" in IEEE Trans. on Computers , Vol. 23, No. 5, May 2004.

28. S. Roman, "Coding and information Theory", Graduate Texts in Mathematics, 1992.

29. S. Mitra, N. Seifert, M. Zhang, Q. Shi, and K. S. Kim, "Robust systemdesign with built-in soft-error resilience," IEEE Trans. Computers, vol. 38, no. 2, pp. 43–52, Feb. 2005.

30. E. Ibe, H. Taniguchi, Y. Yahagi, K. Shimbo, and T. Toba, "Impact of scaling on neutron induced soft error in SRAMs from an 250 nm to a 22 nm design rule," IEEE Trans. Electron Devices, vol. 57, no. 7, pp. 1527–1538, Jul. 2010.

31. P. Reviriego, M. Flanagan, and J. A. Maestro, “A (64,45) triple error correction code for memory applications,” IEEE Trans. Device Mater. Rel., vol. 12, no. 1, pp. 101–106, Mar. 2012.

32. Alzahrani, and T. Chen, “On-chip TEC-QED ECC for ultra-large, single-chip memory systems,” in Proc. IEEE Int. Conf. Comput. Design Design, Very- Large-Scale Integr. (VLSI) Syst. Comput. Process.,Oct. 1994, pp. 132–137.

33. P. Reviriego, M. Flanagan, and J. A. Maestro, “A (64,45) triple error correction code for memory applications,” IEEE Trans. Device Mater. Rel., vol. 12, no. 1, pp. 101–106, Mar. 2012.

34. D. Radley, H. Puchner, S. Wong et al. nuclear. C, MMP. 52, No. 6, pp. 2433-2437, December2005.

35. H. Hostility, h. Taniguchi, Y. Yahagi, K. Shimbo, and T. Electronic Applications, Vol. 57, no. 7, pp. 1527–1538, July2010.

36. C. Argyroid and d. K "Advanced Encryption Algorithm for Reliable Reed-Muller Code" in Proudhon. IEEE Int. Chip Explosion Crisis, September. 2007, p.95-98.

37. Sh. Liu, pp. Rivirego and J. of. The numbers are huge. (Welsh) Cyst. volume. 20, no. 1, pp. 148–156, January2012.

38. M. Shaw, L. Y. Xiao, L.; L. Song, J. Zhang, et al. WW J., MP. 42, no. 3, pp. 553- 561, March2011.

39. R. Nasir and J. "Improving Code Design to Reduce Multidisciplinary Downloads in SRAM are Two Similar Errors" in Dropper, Proc. Skin 34. -Politicians and Governors, September 2008, pp.222-225.

40. G Neuberger, d. L. Castensmid et al. Rice, "Automated Code Optimization Technology to Improve Error Memory for Errors," Computer IEEE Design Test, Vol. 22, no. 1, pp. 50-58, January-February.2005.

41. Nkiruka, M. Flanagan, and J. his. Maestro, "Trinity Repair Code (32,45) for Application Storage," IEEE Trans. Ahh. Rel., Vol. I 12, no. 1, pp. 101-106, March 2012.

42. S. Begt, S. Wen and R. Wong, "Connection Options and a Soft Forgiveness Model," IEEE Trans. nuclear. C, MMP. 56, No. 4, pp. 2111–2118, August2009.

43. K. Pagiamatsis et al. 41, no. 3, pp. 712-727, March2003.

44. Sh. Bagrio, S. Van, and R. Circle engaged. M, m. Books, Vol. 57, no. 4, pp. 814- 822, April2010.

45. Page Rivirego and J. it. Maestro, "SRAM and BIX Multi-Bit Correction Error Codes," ACM Trans. Design automatically. Electronics for Chemistry. Sist. Vol. 14, no. 1, pp. 18: 1–18: 10, 2009January.C. Argyroid, r. Chipna, P. Vargas and. K Rel., Vol. 60, no. 3, pp. 528-537, September 2011.

46. Old man, d. K Pradhan and T. Kokak, "Matrix Codes for Reliability and Efficiency in Mutual Memory", IEEE Trans. Large numbers. (Welsh) Cyst. volume. 19, no. 3, pp. 420-428, March2011.

47. C. Argyroid, C.S.. Lisbon, d. K Pradhan and L. Caro, "Thorough correction of classification with a minimal delay algorithm." IEEE Latin Amar. Trial, March. 2009, pp.652-657.

48. Yahagi, H .; Yamaguchi, E .. Hostility, h .; Kamayama, M. Sato, T .; IEEE Trans Nuclear C., Vol. 54, No. 4, p. 1030-1036, August 2007.

49. c. Argyroid, p. Ribirgo, d. K Pradhan and J. of. Maestro, "Thesis-Based Approach to Correcting Adjacent Errors," IEEE Trans. nuclear. C, MMP 57, No. 4, pp. 2106- 2111, August2010.

50. F. of Proc IEEE Int. Ulsarani and T. Chen, "Extra-Large TEC-QED ECC on a Chip for Single-Chip Memory Systems". Accelerate. calculate. Design, the number is very large. (Wales) Cyst. calculate. Laws., October 1994, pp.132-137.