How to Cite:
Neelima, K., & Subhas, C. (2022). Half diagonal matrix codes for reliable embedded memories. *International Journal of Health Sciences*, *6*(S2), 11664–11677. https://doi.org/10.53730/ijhs.v6nS2.8117

Half diagonal matrix codes for reliable embedded memories

Ms. Neelima K
Research Scholar, Department of Electronics and Communication Engineering, Jawaharlal Nehru Technological University Ananthapur, Ananthapuramu, India – 515002
Corresponding Author email: neelumtech17@gmail.com

Dr. C. Subhas
Professor and Head of ECE Department, Vice-Principal, Jawaharlal Nehru Technological University Ananthapur College of Engineering, Kalikiri, India – 517234
Email: schennapalli@gmail.com

Abstract---Reliability of embedded memories is dependent on at-speed rated correction capability of error detection and correction codes. Many critical applications like medical databases, rocket launch details stored in memories require accurate data and which cause major human losses or economy losses even if a 1% of data is compromised. This paper focuses on developing improved reliable error detection and correction codes with minimum redundant bits and maximum code rate. The decoder uses the concept of Encoder Reuse Technique along with the Modulo-2 addition to calculate syndrome, error location and correction of data bits when data is read from memory. The codes are represented in verilog hardware description language and simulated in the Tool Xilinx ISE 14.5 for XC7Z020-1CLG484 FPGA. The codes are evaluated for their performance in terms of technology parameters like area, delay, power, power – delay product, etc. Also for other parameters like bit overhead, code rate, code efficiency, correction coverage, correction efficiency, etc and correction coverage per cost for combined evaluation. The implemented half diagonal code is more effective as compared to existing matrix codes, it reduces bit overhead at least by 5.5% to 46.96% and coding rate increased at least by 16.74% to 23.85%.

Keywords---adjacent multi-bit errors, bit overhead, code rate, error detecting, correcting codes, redundant bits.

Introduction

Usually as the data read from memory has faults due to radiation among the transistors in high density chip. Hence, there is a need for coding which offers reliability by error detection and correction. Coding is a technique by which the characteristics of information are tailored so as to make ready for the intended application. Coding schemes depend on safety requirement, transmission medium, tolerance level and need for standardization. Decoding is a technique of restoring source information from the encoded information received. It can be more complicated as compared to coding if one has little knowledge of various coding schemes. The errors are grouped as in figure 1. Among these this paper focuses on adjacent error correction.

Fig.1. Error classification

The reliability and efficient error detecting and correcting codes (EDAC) are necessary to deal with single or multiple event upsets create soft errors in data stored in memories. These EDAC Codes use the block diagram as shown in figure 2.

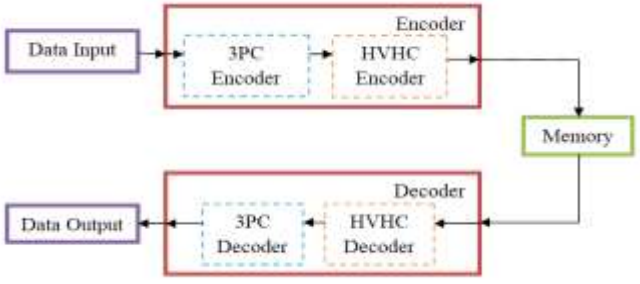

Fig.2. Encoder and Decoder Blocks. [1]

The generic block requires a knowledge which is encoded in required format before writing into the memory and later after read operation, the info is decoded and verified with parity bits for correcting possible errors [1]. Here 3PC represents the three dimensional redundant check code where the 3D redundant check codes (i.e., Horizontal, Vertical and Diagonal Parity Codes) are encoded and these are again encoded in HVHC i.e., Horizontal Vector Hamming Code encoder which again calculates the hamming code parity bits. Then the encoded data is now stored in memory. During the read operation, this encoded data is read by the decoder where the info bits and therefore the parity bits are separated within the sort of MSBs and LSBs

respectively. The parity bits are then decoded using HVHC decoder and are sent alongside the info bits to the 3PC decoder block where errors in erroneous data bits are corrected and are obtained finally as output [13][15][16]. Majorly the changes are done only in 3PC block to reach efficient codes. In this paper, Section II gives a summary of current codes and Section III discusses about new codes. The results are discussed in Section IV, and then paper is finished.

Existing Matrix Codes

In literature, numerous EDAC codes are developed for correcting erroneous data in memories embedded into the system on chip devices. Among them, matrix codes have gained popularity where the data is arranged in matrix form and the redundant bits are calculated. Practically the data along with parity bits called as code word is written into memory and the error evaluation is performed for data read from memory. One among them is 3D parity check codes [1] whose usual way of data representation is indicated in figure 3 for representation of matrix [17][18][19][20], where the H-Bits, V-Bits and D-Bits indicate the Horizontal, Vertical and the Diagonal Parity Bits respectively.

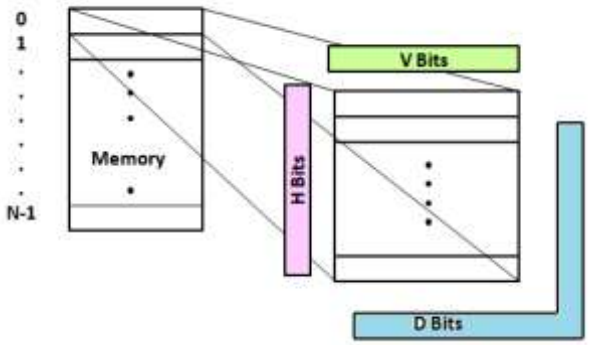

Fig.3. Parity Bits Generation Process from Data Bits

Assume the memory has 64-Bits data storage capacity at each address location. Figure 4 shows the data being represented in matrix form containing 8 rows and 8 columns, so that the parity bits extracted will have 8-H Bits, 8-V Bits and 15-D Bits, a total of 31-Redundant Bits. These parity bits are XORed form of data bits. In decoder, if an error exists, then the combination of all H, V and D Bits indicate the location where the data bit XORed with syndrome calculated is flipped for corrected data. So the decoder uses a more complex circuit than the encoder. Among the Parity Bits, the changes in H-Bits indicate the existence of Errors in Data read from memory, V-Bits indicate the hamming distance and the number of adjacent erroneous bits that can be corrected and the D-Bits indicate the exact error location of the data represented in matrix form.

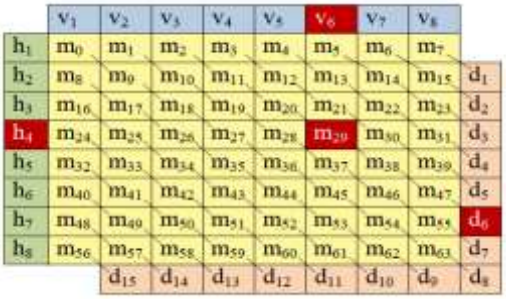

Fig.4. 3D Parity Check Code using 8x8 Matrix Representation

Say if m29 is erroneous, then the change is observed in h4, v6 and d6 parity bits which are evaluated in comparison with the encoded parity bits, then m29 is flipped for corrected data. In [2], a similar concept is used but the diagonal bits are modified into Forward and Backward Diagonal Redundant Bits as shown in figure 4. Hence the number of Parity bits increases by one more diagonal parity bits which necessitate the requirement of a complex decoder to identify candidate erroneous bit.

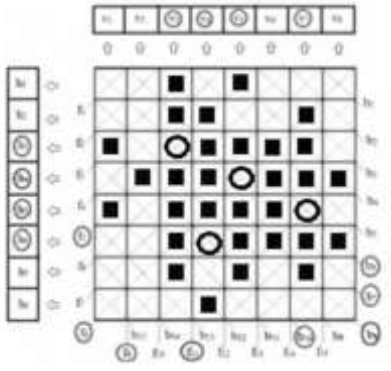

Fig.5. HVD Code with error representation [2]

Also as shown in Figure 5, not only the candidate bits highlighted by circular locations but black squares also represent the erroneous bits instead of being correct [15][16]. Then decoder complexity is increased further due to exact identification of errors in the data contained in the matrix and becomes a time consuming process. The intersection of all four parity bits represents the erroneous bits location i.e., candidate bit which has to be flipped for corrected data.

In [3], Multidirectional Parity Code is also based on Hamming code with minimum bit overhead and maximum code rate shown in figure 6.

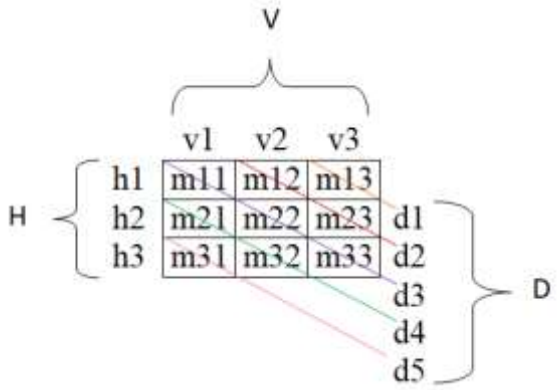

Fig.6. Multidirectional Parity Check Codes [3]

Suppose if m22 has an error, then the h2, v2 and d3 parity bits will be changed and the data has to be flipped for correct data output. Hence, the complexity of decoder is reduced drastically.

In [4], as shown in figure 7, Horizontal Vertical Double Bit Diagonal Parity Code uses two horizontal bits then a h-bit in the next row and so on. Some additional refinement steps are added to identify candidate bits, so decoding process is still complicated.

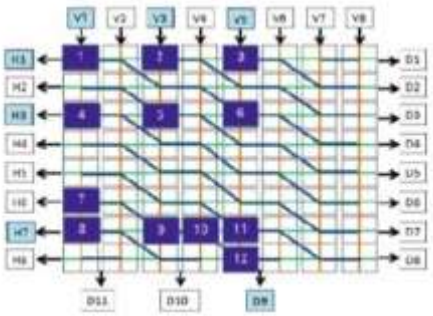

Fig.7. HVDD Codes Candidate Bits [4]

In Figure 7, the diagonal bits take two horizontal bits at a time that reduces complexity in identification of candidate bit as shown in figure 8, which adds to decoder complexity [21].

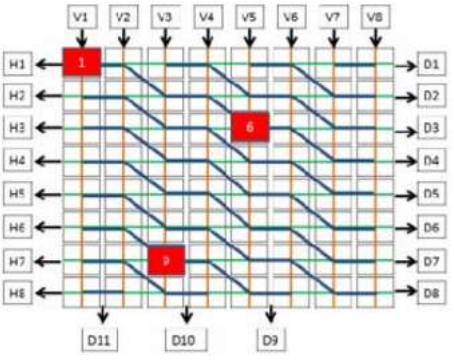

Fig.8. Refined candidate bits obtained from the final step [4]

In [1], the same matrix in 8x8 data representation is modified into a 4x16 matrix [22]. It uses 39 parity bits i.e., 4 H bits, 16 V bits and 19 D bits.

	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9	v_{10}	v_{11}	v_{12}	v_{13}	v_{14}	v_{15}	v_{16}	
h_1	m_0	m_1	m_2	m_3	m_4	m_5	m_6	m_7	m_8	m_9	m_{10}	m_{11}	m_{12}	m_{13}	m_{14}	m_{15}	
h_2	m_{16}	m_{17}	m_{18}	m_{19}	m_{20}	m_{21}	m_{22}	m_{23}	m_{24}	m_{25}	m_{26}	m_{27}	m_{28}	m_{29}	m_{30}	m_{31}	d_1
h_3	m_{32}	m_{33}	m_{34}	m_{35}	m_{36}	m_{37}	m_{38}	m_{39}	m_{40}	m_{41}	m_{42}	m_{43}	m_{44}	m_{45}	m_{46}	m_{47}	d_2
h_4	m_{48}	m_{49}	m_{50}	m_{51}	m_{52}	m_{53}	m_{54}	m_{55}	m_{56}	m_{57}	m_{58}	m_{59}	m_{60}	m_{61}	m_{62}	m_{63}	d_3
	d_{19}	d_{18}	d_{17}	d_{16}	d_{15}	d_{14}	d_{13}	d_{12}	d_{11}	d_{10}	d_9	d_8	d_7	d_6	d_5	d_4	

Fig.9. 4x16 3D Parity Check Code

From Fig.9, the corresponding redundant bits are encoded as

$$h_{ei} = m_i \oplus m_{i+1} \oplus \cdots \oplus m_{i+15}$$
$$v_{ei} = m_i \oplus m_{i+16} \oplus m_{i+32} \oplus m_{i+48}$$
$$d_{ei} = m_i \oplus m_{i+17} \oplus m_{i+34} \oplus m_{i+51} \quad \text{..... (1)}$$

and are decoded as

$$h_{di} = m_i \oplus m_{i+1} \oplus \cdots \oplus m_{i+15}$$
$$v_{di} = m_i \oplus m_{i+16} \oplus m_{i+32} \oplus m_{i+48}$$
$$d_{di} = m_i \oplus m_{i+17} \oplus m_{i+34} \oplus m_{i+51}$$

$$sh_i = h_{ei} \oplus h_{di}$$
$$sv_i = v_{ei} \oplus v_{di}$$
$$sd_i = d_{ei} \oplus d_{di}$$
$$m_{1i} = sh_i \oplus sv_i \oplus sd_i$$
$$m_{oi} = m_i \oplus m_{1i} \quad \text{.... (2)}$$

Say if m29 is erroneous, then it is reflected in flipped h2, v14 and d4 parity bits which after comparison with the encoded redundant bits for correcting erroneous data.

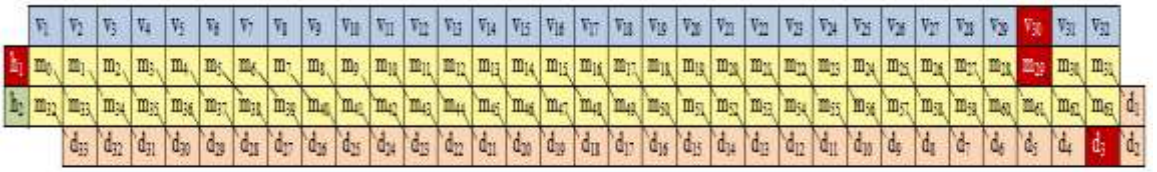

Fig.10. 2x32 3D Parity Check Code

Consider figure 10, the data in 8x8 matrix form is modified into a 2x32 matrix [22]. It uses 67 parity bits are required i.e., 2 H bits, 32 V bits and 33 D bits.

In figure 10, the redundant bits are encoded as

$$\begin{aligned} h_{ei} &= m_i \oplus m_{i+1} \oplus \cdots \oplus m_{i+31} \\ v_{ei} &= m_i \oplus m_{i+32} \\ d_{ei} &= m_i \oplus m_{i+33} \end{aligned} \quad \text{.... (3)}$$

and are decoded as

$$\begin{aligned} h_{di} &= m_i \oplus m_{i+1} \oplus \cdots \oplus m_{i+31} \\ v_{di} &= m_i \oplus m_{i+32} \\ d_{di} &= m_i \oplus m_{i+33} \\ sh_i &= h_{ei} \oplus h_{di} \\ sv_i &= v_{ei} \oplus v_{di} \\ sd_i &= d_{ei} \oplus d_{di} \\ m_{1i} &= sh_i \oplus sv_i \oplus sd_i \\ m_{oi} &= m_i \oplus m_{1i} \end{aligned} \quad \text{.... (4)}$$

where the h_e, v_e and d_e represent the encoder parity bits and h_d, v_d and d_d for decoder parity bits. The s_h, s_v and s_d represent the syndrome bits used for indication of error. The m_{1i} and m_{oi} represent the located erroneous data bit and the recovered data bit respectively. These 3D Parity Check Codes use ultrafast decoding scheme to reduce delay [14]. Say if m29 is erroneous, then it is reflected in h1, v30 and d3 bits which after comparison with the encoded parity bits, will be negated for correct output.

In [24], the Decimal Matrix Codes are developed as shown in figure 10. It uses H and V as parity bits and U-Bits for redundancy which are generated from data input bits. The encoder outputs are calculated as shown in equations below for i = 0 to 31

$$H[8:0] = Data\_in[23:16] + Data\_in\,[7:0]$$

$$H[17:9] = Data\_in\,[31:24] + Data\_in\,[15:8]$$

$$H[26:18] = Data\_in\,[55:48] + Data\_in\,[39:32]$$

$$H[35:27] = Data\_in\,[63:56] + Data\_in\,[47:40]$$

$$V[i] = Data\_in\,[i] \oplus Data\_in\,[i+32]$$

$$U = Data\_in$$

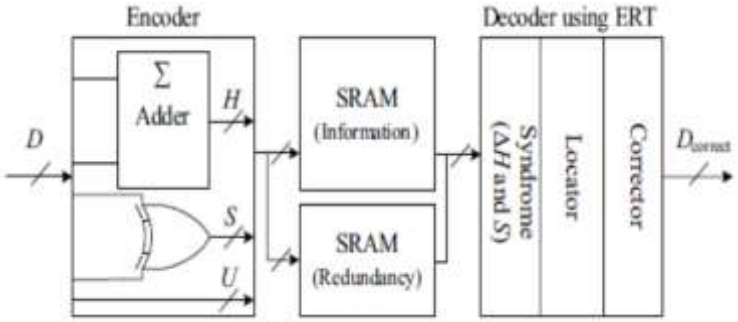

Fig. 11. Block Diagram of Decimal Matrix Code

The decoder works as shown in figure 11, In Encoder Reuse Technique, let Data_read be the input to decoder along with H, V and U encoded bits, for i = 0 to 31

$$H'[8:0] = Data\_read[23:16] + Data\_read\,[7:0]$$

$$H'[17:9] = Data\_read\,[31:24] + Data\_read\,[15:8]$$

$$H'[26:18] = Data\_read\,[55:48] + Data\_read\,[39:32]$$

$$H'[35:27] = Data\_read\,[63:56] + Data\_read\,[47:40]$$

$$V'[i] = Data\_read\,[i] \oplus Data\_read\,[i+32]$$

$$U' = Data\_read$$

For syndrome calculation,

$$\Delta H = H' - H$$

$$S = V' \oplus V$$

For Error Location and Correction,

$$If\ (\Delta H = 0\ and\ S = 0)$$
$$then$$
$$Data\_out = Data\_read$$
$$else$$
$$Data\_out = Data\_read \oplus S$$

Thus in encoder only four 16-bit adders, XOR Gates and Buffers are used and the total number of parity bits used are 68 Bits. If the number of errors are more than 32 Adjacent Bits, then the redundant memory bits i.e., U-Bits are used as Data_out Bits.

In [25], Modified Decimal matrix Codes uses two 32-bit adders instead of four 16-bit adders for encoder. i.e.,

$$H[16:0] = Data\_in[31:16] + Data\_in[15:0]$$

$$H[33:17] = Data\_in[63:48] + Data\_in[47:32]$$

So a total of 66 parity bits used. The remaining process is same as that of Decimal Matrix Codes.

In [26][27][28], Parity Matrix Codes use H Bits as parity encoded bits for 32-bit data i.e., 6-Bits + 6-Bits = 12 Bits in total. So the total of 48 parity bits is used with the remaining process being same.

The other codes that are investigated are from [5-12]. In [5], the Reusable Matrix Codes are developed. In [6], a hybrid architecture uses reconfiguration techniques using compressed redundant information that is non-vulnerable to radiation, called as hardwired seed bits (HSB) along with interleaving for correcting multiple bit upsets.

In [7], a two-level low-density parity-check (LDPC) code is proposed for non-volatile memories. With negligible delay and a small overhead, a auxiliary codeword is used to encode a group of bits from primary data packets of NVM memory. In [8], an efficient 4-bit adjacent error correction codes using less number of parity bits and reduced decoder complexity is developed.

In [9], parity check matrix or H – matrix is developed by using simplified expressions for SEC-DED codes. These are compact and consume less power. In [10], a double asymmetric error correction scheme based on proactive correction coset decoding is used for STT-MRAM, operated at two basic levels, i.e., for first error correction, a proactive correction level (PCL) and for correcting second error, the asymmetric correction level (ACL) are used.

In [11], the transposable retransmissions are adopted for different error rates to improve forward error correction in enhanced Parity Product Code (PPC). In [12], a novel Matrix decoding algorithm is developed which is capable of correcting 9 errors in 16-bit data and 11 errors in 32-bit data. It increases the memory yield in the presence of adjacent errors.

Proposed Matrix Code

The basic mechanism of encoding that is used is as depicted in fig.12 i.e., Consider an 8-bit data being arranged. Here V [3…0] are the vertical encoded parity bits and H [1…0] are the half diagonal parity bits.

D7	D6	D5	D4	H2
D3	D2	D1	D0	H1
V3	V2	V1	V0	H0

Fig.12. Encoding Mechanism

The vertical parity bits are calculated as
$V_0 = D_0 \oplus D_4$
$V_1 = D_1 \oplus D_5$
$V_2 = D_2 \oplus D_6$
$V_3 = D_3 \oplus D_7$
The half diagonal parity bits are calculated as
$H_0 = D_3 \oplus D_2$
$H_1 = D_7 \oplus D_6 \oplus D_1 \oplus D_0$
$H_2 = D_5 \oplus D_4$

The decoding mechanism is same as that of DMC and MDMC except that the $\triangle H$ is calculated from modulo-2 addition. Also the error is detected only if $\triangle H$ and S are nonzero numbers. Similarly the concept can be extended for higher order data bits.

Results and Discussion

For modeling verilog hardware description language was utilized and the Codes are functionally tested in Xilinx ISE 14.5 Tool for XC7Z020-1CLG484 FPGA.

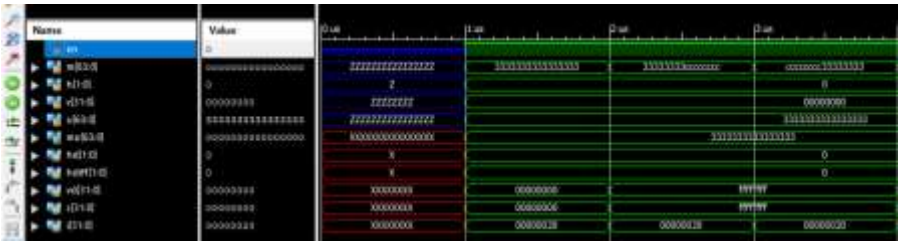

Fig.13. Simulation Result of Matrix Decoders

The simulation result of proposed design is shown in figure 13 emphasizes correction of a maximum of 64-bits for a 64-bit data. For example consider input data for 2 x 32 Horizontal Vertical Parity Check Code Encoder as 64'h3333333333333333 when enable=1, then h=2'b0 and v=32'b0. For its Decoder, if the errors are induced as 32 bits say 64'h33333333cccccccc then the output produced is corrected to 64 bits i.e., 64'h3333333333333333 same as original input data. Hence the correction capability is N/2 erroneous bits in N data bits.

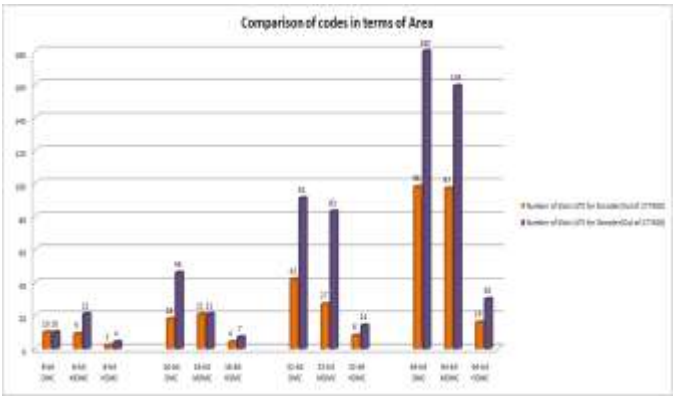

Fig.14. Evaluation of Matrix Codes for Area

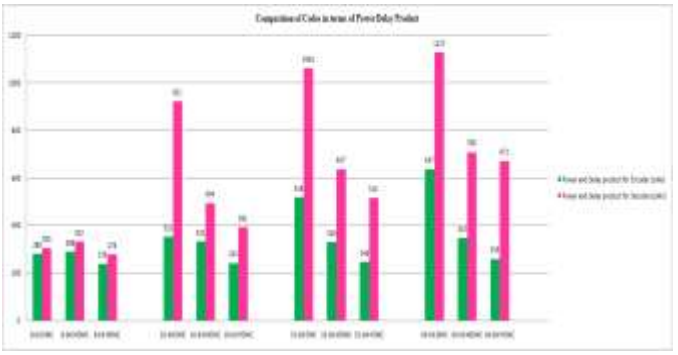

Fig.15. Evaluation of Matrix Codes for Power Delay Product

The HDMC Code uses less area by at least 70.37% to 83.5% for encoder and 66.67% to 83.13% for decoder. Also the power delay product is minimized at least by 14.64 to 25.36% for encoder and 5.2% to 20.85%, as shown in figures 14 and 15. Hence HDMC Code proves to be efficient.

Table.1. Comparison of various codes with varying data lengths for various parameters

Parameters	DMC	MDMC	HDMC	DMC	MDMC	HDMC	DMC	MDMC	HDMC	DMC	MDMC	HDMC
# Data Bits, k	8	8	8	16	16	16	32	32	32	64	64	64
# Parity Bits, r	12	10	7	20	18	11	36	34	19	68	66	35
# Code Word, n=r+k	20	18	15	36	34	27	68	66	51	132	130	99
Bit Overhead, r/k in %	150	125	87.5	125	112.5	68.75	112.5	106.25	59.37	106.25	103.125	54.69
Code Rate, k/n in %	40	44.4	53.33	44.44	47.06	59.26	47.06	48.48	62.74	48.48	49.23	64.65
Code Efficiency, r/n in %	60	55.56	46.67	55.56	52.94	40.74	52.94	51.51	37.25	51.51	50.77	35.35

The table 1 compares different codes for parameters such as bit overhead, code rate and code efficiency. The HDMC code reduces the bit overhead by at least 5.5% to 46.96%, code rate is improved at least by 16.74% to 23.85% and the code efficiency is found to vary from 16.67% to 30.37%.

Conclusion

As the technology scales down, the soft errors are caused in memories due to radiation effects. The errors occur in data when stored and retrieved from memories which can be overcome by EDAC Codes like Matrix codes. This paper mainly concentrates on reducing bit overhead along with reliable data. The designs are modelled in Verilog HDL and are verified in Xilinx ISE 14.5 Tool for 28nm Zynq FPGA with part number XC7Z100-2FFG1156. The assessment of these methods is done for 8 – bit, 16-bit, 32-bit and 64-bit Data. The proposed HDMC Code proves to be more reliable which is capable of correcting half the adjacent errors either in lower half or upper half of data enabling it to be used in reliable memory applications. The use of odd/ even columns in H-Matrix can improve code rate and correction capability. As the number of data bits increase, the bit overhead is reduced by 5.5% to 46.96% and the code rate is improved by 16.74% to 23.85%. Also as the results confirm that the HDMC Code proves to be better in terms of area and PDP when compared with existing decimal and modified decimal matrix codes.

References

[1] Shivani Tambatkar, Siddharth Narayana Menon, Sudarshan.V, M.Vinodhini and N.S.Murty, “Error Detection and Correction in Semiconductor Memories using 3D Parity Check Code with Hamming Code” International Conference on Communication and Signal Processing, April 6-8, 2017, India, pp. 0974-0978.

[2] S. Sharma and P. Vijayakumar, "An HVD based error detection and correction of soft errors in semiconductor memories used for space applications," 2012 International Conference on Devices, Circuits and Systems (ICDCS), Coimbatore, 2012, pp. 563-567.

[3] Vishal Badole, Amit Udawat, “Implementation of Multidirectional Parity Check Code Using Hamming Code for Error Detection and Correction,” International Journal of Research in Advent Technology, Vol.2, No.5, May 2014, E-ISSN: 2321-9637, pp. 1-6.

[4] Md. Shamimur Rahman, Muhammad Sheikh Sadi, Sakib Ahammed and Jan Jurjens, ”Soft Error Tolerance using Horizontal – Vertical Double – Bit Diagonal Parity Method” , 2nd IEEE International Conference on Electrical Engineering and Information and Communication Technology (ICEEICT) , 21-23 May 2015.

[5] J. Athira and B. Yamuna, "FPGA Implementation of an Area Efficient Matrix Code with Encoder Reuse Method," *2018 International Conference on Communication and Signal Processing (ICCSP)*, Chennai, 2018, pp. 0254-0257,

[6] A. J. Olazábal and J. Pleite Guerra, "Multiple Cell Upsets Inside Aircrafts. New Fault-Tolerant Architecture," in *IEEE Transactions on Aerospace and Electronic Systems*, vol. 55, no. 1, pp. 332-342, Feb. 2019.

[7] IEEE Standard for Error Correction Coding of Flash Memory Using Low-Density Parity Check Codes," in IEEE Std 1890-2018 , vol., no., pp.1-51, 28 Feb. 2019.

[8] Li, Jiaqiang & Xiao, Li & Reviriego, Pedro & Zhang, Rongsheng. (2018). Efficient Implementations of 4-Bit Burst Error Correction for Memories. IEEE

Transactions on Circuits and Systems II: Express Briefs. PP. 1-1. 10.1109/TCSII.2018.2817390.

[9] Samanta, J., Bhaumik, J. & Barman, S. “Compact and power efficient SEC-DED codec for computer memory”. Microsyst Technol (2019). https://doi.org/10.1007/s00542-019-04366-7.

[10] Liwen Liu, Yiqi Zhuang, Li Zhang, Hualian Tang, Siwan Dong, “Proactive correction coset decoding scheme based on SEC-DED code for multibit asymmetric errors in STT-MRAM”, Microelectron. J. 82: 92-100 (2018)

[11] Dang, Khanh & Tran, Xuan-Tu. (2019). An Adaptive and High Coding Rate Soft Error Correction Method in Network-on-Chips. VNU Journal of Science: Computer Science and Communication Engineering. 35. 10.25073/2588-1086/vnucsce.218.

[12] M.s, Sunita. (2013). Error Detection and Correction in Embedded Memories Using Cyclic Code. 10.1007/978-81-322-1524-0_16.

[13] Alfonso Sanchez-Macian, Pedro Reviriego, Juan Antonio Maestro, “Hamming SEC-DAED and Extended Hamming SEC-DED-TAED Codes Through Selective Shortening and Bit Placement”, IEEE Transactions on Device and Materials Reliability, Vol. 14, No. 1, pp. 574-576, March 2014.

[14] Luis-J. Saiz-Adalid, Pedro Gil, Juan-Carlos Ruiz, Joaquin Gracia-Moran, Daniel Gil-Tomas, J.-Carlos Baraza-Calvo, “Ultrafast Error Correction Codes for Double Error Detection/ Correction”, 12th European Dependable Computing Conference, pp. 108-116, 2016.

[15] Pedro Reviriego, Jorge Martinez, Salvatore Pontarelli and Juan Antonio Maestro.”A method to Design SEC-DED-DAEC Codes with optimized decoding”, IEEE Transactions on Device and Materials Reliability, Vol. 14, No. 3, pp. 884-889, September, 2014.

[16] Vijay Tawar, Rajani Gupta, “A 4-Dimensional Parity based Data Decoding Scheme for EDAC in Communication Systems”, Internal Journal for Research in Applied Science and Engineering Technology (IJRASET), Vol.3, Issue. IV, pp. 183-191, April, 2015.

[17] Matrix Code Based Multiple Error Correction Technique for N-Bit Memory Data”, International journal of VLSI Design & Communication Systems (VLSICS), Vol. 4, No.1, pp. 30-37, February 2013.

[18] Argryides et al., “Matrix Codes for Reliable and Cost Efficient Memory Chips”, IEEE Transactions on VLSI Systems, Vol. 19, pp.420-428, March 2011.

[19] T. Maheswari, Mr. P. Sukumar, “Error Detection and Correction in SRAM Cell using Decimal Matrix Code”, IOSR Journal of VLSI and Signal Processing (IOSR - JVSP), Vol. 5, Issue 1, pp. 9-14, January – February, 2015.

[20] Avijit Dutta and Nur A. Touba, “Multiple Bit Upset Tolerant Memory Using a Selective Cycle Avoidance Based SEC-DED-DAEC Code”, 25th IEEE VLSI Test Symmposium (VTS’07), 0-7695-2812-0/07, IEEE, 2007.

[21] Ms. Neelima K, Dr. C. Subhas, “Multiple Adjacent Bit Error Detection and Correction Codes for Reliable Memories: A Review”, First International Conference on Communications, Signal Processing and VLSI (IC2SV2019), National Institute of Technology, Warangal, October 26-27, 2019.

[22] Ms. Neelima K, Dr. C. Subhas, “Efficient Adjacent 3D Parity Error Detection and Correction Codes for Embedded Memories”, International Conference on Electronics, Computing and Communication Technologies, CONECCT 2020 organized by IEEE Bangalore Section, India during, July 2-4, 2020.

[23] Shanshan Liu, Liyi Xiao, Jie Li, Yihan Zhou, Zhigang Mao, “Low – Redundancy Matrix-Based Codes for Adjacent Error Correction with Sharing”, 978-1-5090-5404-6/17/$31.00© 2017, IEEE 2017.
[24] T. E. Santhia, R. H. R. Bharathi and M. Revathy, "Error detection and correction using decimal matrix code: Survey," *2017 IEEE International Conference on Electrical, Instrumentation and Communication Engineering (ICEICE)*, Karur, 2017, pp. 1-5, doi: 10.1109/ICEICE.2017.8191867.
[25] Ahilan A. and Deepa P., "Modified Decimal Matrix Codes in FPGA configuration memory for multiple bit upsets," *2015 International Conference on Computer Communication and Informatics (ICCCI)*, Coimbatore, 2015, pp. 1-5, doi: 10.1109/ICCCI.2015.7218146.
[26] Sabeen, Samia and Nussaïbah B Raja. “Enhanced Memory Reliability Using Parity Matrix Code” 2014.
[27] S. Manoj and C. Babu, "Improved error detection and correction for memory reliability against multiple cell upsets using DMC & PMC," *2016 IEEE Annual India Conference (INDICON)*, Bangalore, 2016, pp. 1-6, doi: 10.1109/INDICON.2016.7839094.
[28] Neelima Koppala, Chennapalli Subhas, “Low Overhead Optimal Parity Codes”, Telkomnika (Telecommunication Computing Electronics and Control), Vol 20, No. 3, June 2022, pp. 501-509.