ISSN No :- 2456- 2165

Detecting and Correcting Multiple Cell Upsets With 64-Bit Hamming Code in Memories

Nimita Sharma
M.Tech Scholar
EC-VLSI
Jagannath University , Jaipur(Raj),India
nimitaskit21@gmail.com

Nidhish tiwari
Assistant professor
Jagannath National Institute of Technology(JNIT),Jaipur(Raj),India
Nidhishtiwari@gmail.Com

Abstract :- The Transient MCUs (Multiple Cell Upsets) are becoming the big issues in memory reliability that exposed to environmental radiation. In order to prevent the MCUs are causing data-corruption. The more complex ECCs (error correction codes) is widely been used-to protect a memory; but the problem is which they required higher delay overhead. In order to gain the maximum capability of error detection; decimal algorithm is used by DMC. And again ERT (Encoder Reuse Technique) is given to extra circuit's area overhead minimization without disturbing of entire encoding and decoding process. The DMC is used by DMC encoder to be a decoder part.In this paper , MCs (Matrix Codes) - which based on HCs (Hamming Codes), have proposed for protection of memory. An important issue is that they're DEC (double error correction) codes. And for MCU (32-bit), HC is capable of correction of 16-bits in single step. Here we worked on MCUs based on 32 and 64 bits. In 32 bits redundant bits are 23. And proposed HC (Hamming Code) is able to minimize LUTs and delay for 32-bits and 64-bits.

Keyword:- *ECCs (error correction codes), MTTF (mean time to failure), memory, MCUs (multiple cells upsets), Decimal algorithm, etc.*

I. INTRODUCTION

A generalized schema for attaining error detection & correction is used for invading some redundancy (which is some supplementary information) into a message that can be used by receiver for evaluating consistency of delivered messages & selecting data assumed to be corrupted. Error detection & correction schema can be either of systematic or non-systematic form. In systematic schema, transmitter transmits unique data & links some definite check bits (or parity data), that are obtained from data bits with help of a deterministic algorithm. If there is requirement of error detection only, then a receiver can implemented that algorithm for receiving data bits & comparing its produced outcomes to received check bits. If there is some difference in values then there is some error took ace during transmission. Error correction codes are applied in lower layered communication & also for reliable storage components like hard disks, RAM, CDs & DVDs etc.

In a system having non-systematic code, unique message needs to be converted to en encoded message which is having minimum of bits that are in that unique message. This error detection & correction code is concentrated over providing support confronting soft errors which distinct it as bit flips in memory. Various schemes are applied to midi gate upsets in the memory. As an illustration, Bose-Chaudhuri-Hocquenghem codes [8], Reed-Solomon codes [9], PDS (punctured difference set) code [10] & matrix codes are implemented for communicating with MCUs in memories. Though, these codes need more amount of power, area & delay overheads as encoding & decoding circuitries are having complicated layout in such codes. Reed-Muller code [14] is another type of protection code that has ability of recognizing & correcting substantial errors than a Hamming code. The disadvantage related to these protection codes is that they are having higher power & area requirements. Hamming codes are majorly applied for correcting SEUs (single error upsets) in memory because of their capability of correcting single errors by minimized performance & area overheads [13]. As they are efficient in correcting single errors in a data word, they are not able to correct double bit errors through single event upset. An extension to basic SEC-DED Hamming code is proposed for producing a special category of codes called Hsiao codes for improving speed, reliability & cost of decoding logic [14].

Another class of SEC-DED codes called single error correcting; double error detecting, single byte error detecting SEC-DED codes are proposed for recognizing any amount of errors that disturbs a single byte. Such codes are more accurate than traditional SEC-DED codes for securing byte arranged memories [15] [16]. Even though they work over lessen overhead & are suitable for detecting several errors, they are not capable of correcting several errors. Some extra codes like single byte error correcting, double byte error detecting (SBC-DBD), DEC-TED (double error correcting, triple error detecting) codes are capable of correcting several errors as described in [10].

This single error correcting, double error detecting & double adjacent error correcting (SEC-DED-DAEC) codes furnishes ECC techniques with minimal costing for correction of relative errors as proposed in [12]. The main disadvantage of this code is that there is some chance of miss correction in small subset having too many errors. As the technology of CMOS is brought down to nano scale & memories are interlinked by increased electronic components, soft error rate in memory cells is increased at a rapid speed particularly in that case where memories work in space environments because of ionizing impact of alpha particles, neutron & cosmic rays [21],[22],[23].

Interleaving schema is applied for restraining of MCUs that helps in rearrangement of cells in physical layout to distinguish the bits in similarly logical word in distinctive physical words. Though, interleaving techniques cannot be efficiently applied to CAM (content addressable memory), as it has strict coupling of hardware design from both of the cells & copying the circuit design. BICS (built in current sensors) are proposed for supporting single error correction & double error detection codes for providing security against MCUs. Though, this technology is capable of correcting two errors in one word. Recently, in a 2-D matrix codes (MCs) are proposed for correction of MCUs for every word in an efficient manner having a lower decoding delay, where a single word is divided to several rows & columns in logical manner. Bits for each row are secure thorough Hamming codes where parity code is invaded into every column. For the MCs constituted over

ISSN No :- 2456- 2165

Hamming, as Hamming recognize two errors, vertical syndrome bits are triggered in order to correct those two errors. As an outcome, MC has ability off correcting two errors in every scenario. A technique that combines decimal algorithm & hamming code are obtained for implemented over software scale. It makes use of addition of integer values for recognizing & correcting software errors.

A novel DMC (decimal matrix code) constituted over divide symbol is proposed for furnishing improved memory reliability. This proposed DMC makes use of decimal algorithm (decimal integer addition & decimal integer subtraction) for recognizing errors. The superiority of implementing such decimal algorithm is its error reorganization ability is maximized in order to enhance the memory reliability. Apart from ERT (encoder-reuse technique) is proposed for reducing area overhead of additional circuitries (encoder & decoder) without creating any disturbance in whole of encoding &decoding process as ERT makes use of DMC encoder by to be counted as a portion of decoder.

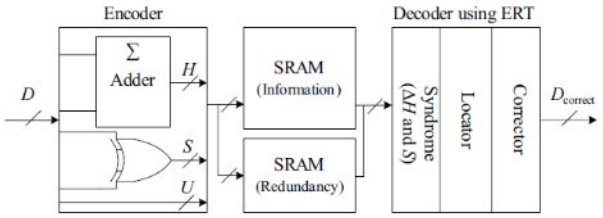


Fig. 1. Proposed schematic of fault-tolerant memory protected with DMC [25].

II. DMC(DECIMAL MATRIX CODE)

DMC is implemented for ensuring reliability where MCUs are present by minimized performance overheads & an example of 64-bit decoding & encoding is considered over this proposed scheme.

A. Fault-Tolerant Memory

A fault tolerant memory is presented in the figure. Initially, in the encoding (write) process, information bits D are provided to DMC encoder & horizontal redundant bits & vertical redundant bits V are fetched though DMC encoder. As process of encoding is completed, attained DMC codeword is accumulated n memory. If MCUs takes place in memory, such errors are made accurate in decoding (read) technique. Because of superiorities of decimal algorithm, proposed DMC possess higher fault tolerant ability with decreased performance overheads. In fault tolerant memory, ERT techniques are applied for minimizing area overhead of additional circuitries & will be describe in the below portions.

B. DMC Encoder

In first step of DMC, place-matrix & divide symbol techniques are implemented i.e. N-bit word is divided in k number of symbols with m bits (N = k x m), and these obtained symbols are placed in k1 × k2 2-D matrix (k = k1 × k2, in which k1 & k2 present number of rows & columns in logical matrix). Secondly, horizontal redundant bits termed as H are generated by applying decimal integer addition of chosen symbols in every row. Here, every symbol is referred as decimal integer. Third, vertical redundant bits V are fetched through binary functions in the bits for every column. It must be noted that both of the arrange-matrix & divide symbol are applied in logical manner alternatively physically. Hence, there is no such need to vary physical layout by the proposed DMC.

For explaining the proposed schema of DMC, we have considered a 64-bit word as demonstrated in figure 2. Cells from D0 – D63 are referred as information bits. The 64 bit word splinted to 16 symbols of 4-bit.k1 = 2 & k2 = 4 are chosen respectively for this. H0-H39 are referred as horizontal check bits; while V0-V31 are respective vertical check bits. Though, it is described that both the terms of maximum correction capability (i.e. maximum size of MCUs can be corrected) & number of redundant bits are different from each other when different values are taken for m & k respectively. Hence, adjustment of m & k should be made cautiously for minimizing the correction ability & number of redundant bits. As an illustration, in this scenario, k = 2 x 2 & m = 8, it is possible to correct 1-bit error only & number of redundant bits will be 80. As k = 4x4 & m = 2, it is possible to correct 3-bit errors & number of redundant bits will be 32. Therefore, as k = 2 x 4 & m = 4, maximum correction ability can be up to 5 bits & number of redundant bits will be 72. In this paper, memory reliability is enhanced through improving error correction capability that is initially computed. Therefore, k = 2 x 8 & m = 4 are used for generating DMC. The decimal integer addition is used for obtaining horizontal redundant bits 'H' as below:

$$H4H3H2H1H0 = D3D2D1D0+D19D18D17D16 \quad .....(1)$$
$$H9H8H7H6H5= D7D6D5D4 + D23D22D21D20 \quad .....(2)$$

And in a similar fashion for horizontal redundant bits H14H13H12H11H10, H19H18H17H16H15H16, H24H23H22H21H20, H29H28H27H26H25, H34H33H32H31H30 & H39H38H37H36H35 in which "+" presents decimal integer addition. For vertical redundant bits V, we are having

$$V0 = D0 \wedge D31 \quad .....(3)$$
$$V1 = D1 \wedge D32 \quad .....(4)$$

And parallel to remaining vertical redundant bits. The encoding process can be implemented through binary & decimal addition function from (1) to (4). The encoder computing redundant bits makes use of multi-bit adders are & XOR gates as demonstrated in figure. In this figure, H39-H0 are horizontal redundant bits, V31-V0 are vertical redundant bits & remaining U63-U0 are referred as information bits that are straightly copied from D31-D0 [27][28].

C. DMC Decoder

For obtaining a corrected word, decoding procedure is needed. As an illustration, initially, the redundant bits that are received are H4H3H2H1H0' are formed by information bits D' that are received. Secondly, horizontal syndrome bits ΔH4H3H2H1H0 & vertical syndrome bits S3 − S0 are computed as below:

$$\Delta H4H3H2H1H0 = H4H3H2H1H0' - H4H3H2H1H0 \quad .....(5)$$
$$S0=V0'\wedge V0 \quad .....(6)$$

And dissimilar to remaining vertical syndrome bits in which "-" stands for decimal integer subtraction. As ΔH4H3H2H1H0 & S3 − S0 equates to 0, accumulated codeword will be possessing real information bits in symbol0 having no error. As ΔH4H3H2H1H0 & S3 − S0 are nonzero values, persuaded errors (the quantity of errors is 4 in this case) are recognized & located in symbol 0 & following equation is used to correct those errors.

$$D0correct = D0 \wedge S0 \quad .....(7)$$

ISSN No :- 2456- 2165

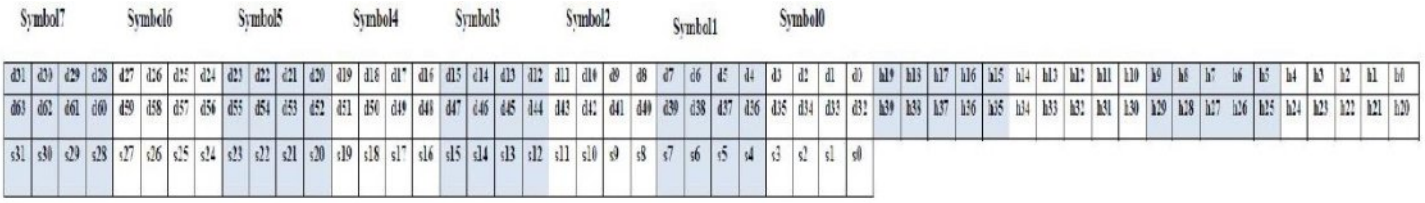


Fig 2: 64-bits DMC logical organization (k = 2 × 8 and m = 4). Here, each symbol is regarded as a decimal integer

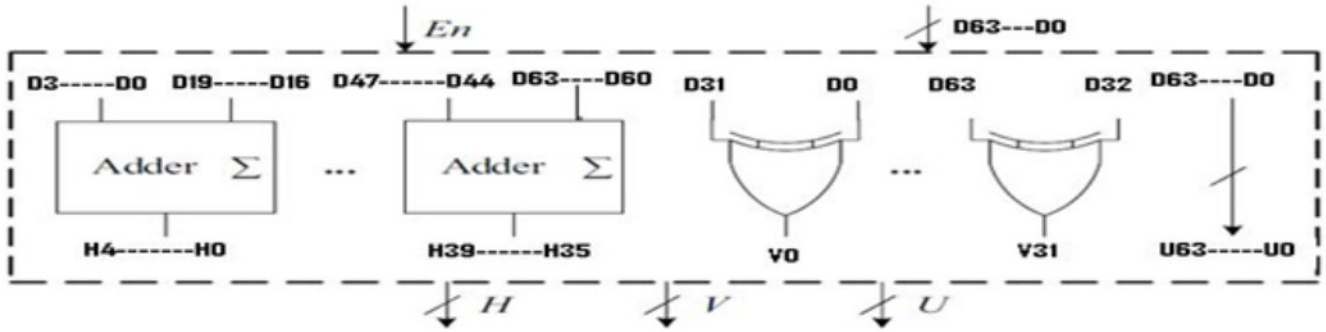
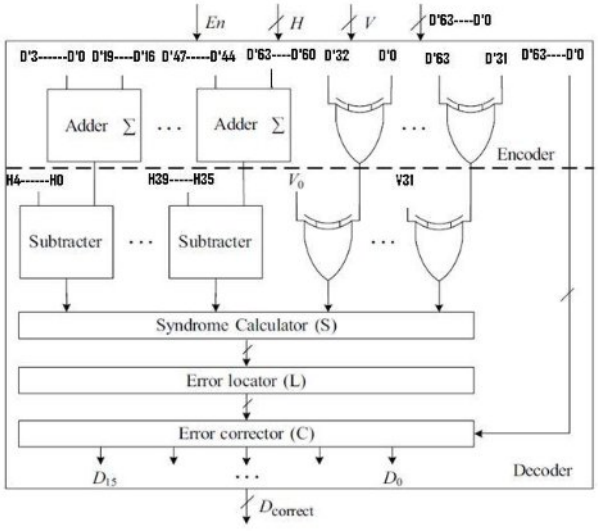


Fig 3: 64-bit DMC encoder structure using multi bit adders and XOR gates

The figure presents a DMC decoder that is produced for given sub modules & implements a précised job in the process of decoding: locating error, syndrome calculation & error correction. It is clearly observed from figure that there is a need to re-compute the redundant bits from received information bits D' & it is compared to original set of bits for acquiring syndrome bits ΔH & S. the error locator requires ΔH & S for detecting & locating those bits having some errors in it. Lastly, in error corrector, such errors can be corrected through inversion of values of error bits [30][31].



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

Fig 4: 64-bit DMC decoder structure using ERT

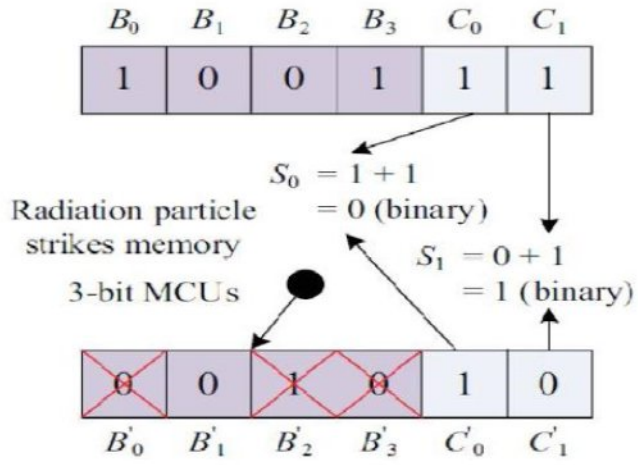


Fig 5 : limits of binary error detection in simple binary operations

Area of the circuitry in DMC is reduced by re-usage of encoder. This is termed as calling of ERT. This ERT is capable of minimizing area overhead of DMC without producing any variations in the process of encoding & decoding. It is seen from the figure that DMC encoder is reused as well for attaining syndrome bits in DMC encoder. Hence, whole circuit area of DMC can be reduced though present encoder circuits. Apart from this, figure presents the proposed decoder along allow signal En to decoder if encoder should be incorporated with decoder or not. In different wordings, En signal is required for determining a difference in encoder & decoder and it is managed by read & write signals. Hence, in encoding (write) mode, DMC encoder is just an encoder for implementing encoding functions. Though, it decoding (read) mode, encoder is used for computation of syndrome bits in decoder. This consecutively presents the way area overhead of additional circuitries is minimized.

ISSN No :- 2456- 2165

III. PROBLEM STATEMENT

The transient multiple cell upsets (MCUs) are seen as a major flaw in reliability of memories when revealed to radiant conditions. Complicated ECCs (error correction codes) along DMC (decimal matrix code) is used for eliminating the condition of data corruption in MCUs & protecting the memory. The issue related to them is that they need overhead with high delay. Lately, MCs (matrix codes) constituted over Hamming codes are proposed for protecting the memory.

A. Issue 1 :- High Complexity and Delay

Here DMC code is proposed in this paper. While on the other side it is also observed that consumption of power & complexity in circuitry is high. The flow of data & consumption of power will be more because of higher complexity. Therefore, we are working over minimizing the delay & complexity of circuitry in order to reduce the area as well.

B. Issue 2 :- High Number of LUTS(Look Up Table)

As observed from the session of outcomes, for the existed DMC code there are much LUTS. The proposed methodology is used in order to minimize the quantity of LUTS. The complexity & areas of system will be minimized by reducing the LUTS in a system.

IV. PROPOSED METHODOLOGY

Performance of the system can be enhanced by minimizing area & delay of a system in order to decrease the complexity. Hamming code is applied to serve this purpose. We focus over 32 bit & 64 bit hamming code in MCU. Hamming code generates redundant bits horizontally to improve the performance. The size of horizontal syndrome bit is 7 for proposed Hamming code constituted over DMC.

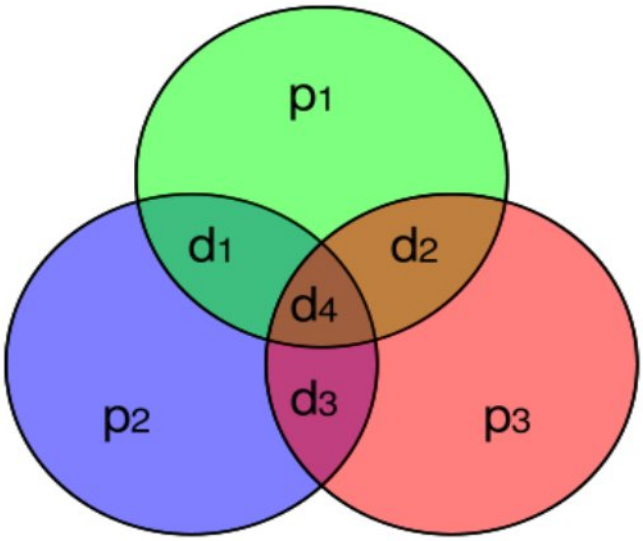


Fig 6 :- Hamming code detection

p1:=d(0) xor d(3) xor d(1) .....(8)

p2:=d(0) xor d(3) xor d(2) .....(9)

p3:=d(1) xor d(3) xor d(2) .....(10)

ha<=(p1 & p2 & p3 & d(3 downto 0)) .....(11)

V. RESULTS

A. 32 Bit ECC By Hamming Code

In this we design the 32 bit ECC(Error Correction Code) by Hamming code. The capability of the error correction is 16 bits in a one time . In the 32 bit ECC 23 bits are redundant bits which are using as a external bits for correct the code . For correct the error in MCU (Multiple Cell Upsets) , Hamming code are using . In this 7 bits are using for the horizontal syndrome bits(H1-H7) and 16 vertical bits (V1-V16) and these bits are known as redundant bits .

For the input we give the original address of the memory then the fault of the MCU . The original bit is 11110101101011111111011010101100 and fault bit is 11110101101011111111011110101111.

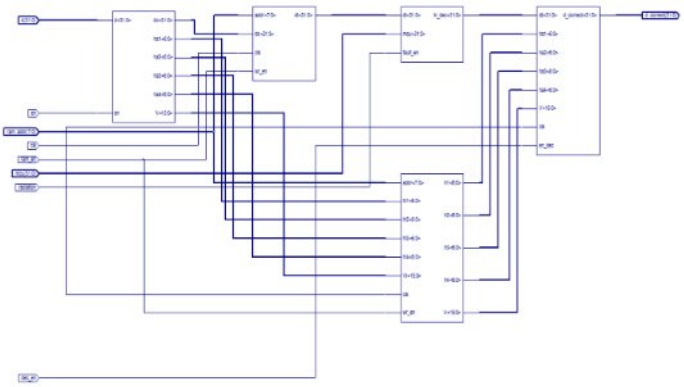


Fig 7 :- RTL design for the 32 Bit ECC by Hamming code

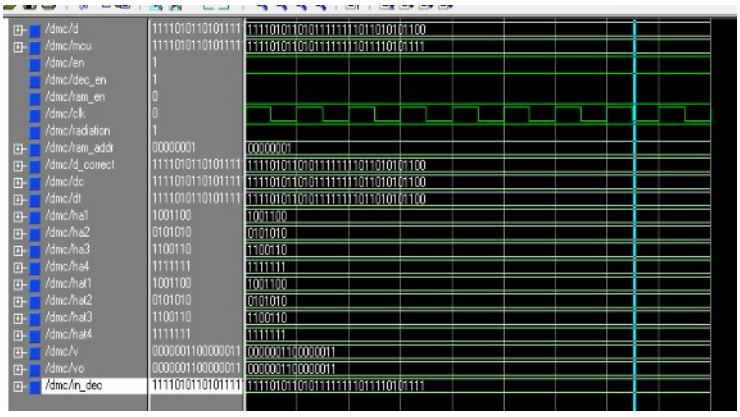


Fig 8 :- Output Waveform for the 32 bit ECC by Hamming Code

From figure 6.8 , we can check that as we give the input of D which is 11110101101011111111011010101100 and MCU fault is 11110101101011111111011110101111 .After the correction of the bits output is 11110101101011111111011010101100. As we can check from the waveform output is same as the input of the D.

ISSN No :- 2456- 2165

B. 64 Bit ECC By Hamming Code

In this we design the 64 bit ECC(Error Correction Code) by Hamming code. The capability of the error correction is 16 bits in a one time . In the 64 bit ECC 39 bits are redundant bits which are using as a external bits for correct the code . For correct the error in MCU (Multiple Cell Upsets) , Hamming code are using . In this 7 bits are using for the horizontal syndrome bits(H1-H7) and 32 vertical bits (V1-V32) and these bits are known as redundant bits .

For the input we give the original address of the memory then the fault of the MCU . The original bit is 1111111111111111111111111111111111111101011010111111111011010 101100 and fault bit is 1111111111111111111111111111111111111101011010111111111011110 101111.

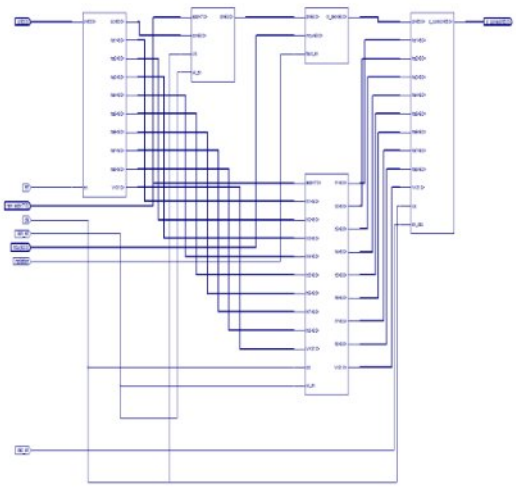


Fig 9 :- RTL design for the 64 Bit ECC by Hamming code

Figure 9 is showing the RTL design for the 64 bit ECC by Hamming code. In this RTL 64 bit is given as a D and MCU.

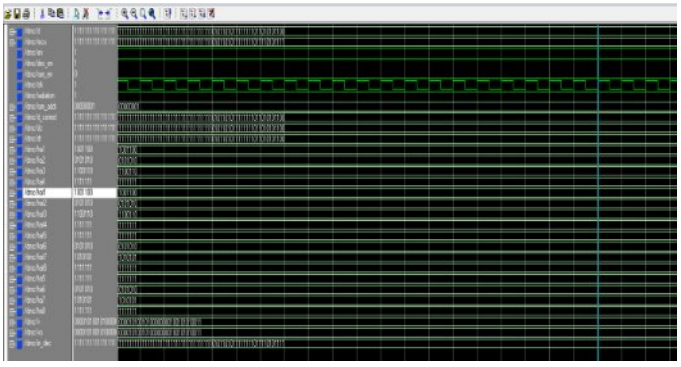


Fig 10 :- Output Waveform for the 32 bit ECC by Hamming Code

From figure 10, we can check that as we give the input of D which is 1111111111111111111111111111111111111101011010111111111011010 101100 and MCU fault is 1111111111111111111111111111111111111101011010111111111011110 101111 .After the correction of the bits output is 1111111111111111111111111111111111111101011010111111111011010 101100. As we can check from the waveform output is same as the input of the D.

C. Comparison Table For 32 Bit ECC In MCU

	Delay	Number Of Slices
32 Bit ECC By DMC	13.342 ns	180
32 Bit ECC by Hamming Code	9.193 ns	136

Table 1:- Comparison Table for 32 bit ECC in MCU

D. Comparison Table For 64 Bit ECC In MCU

	Delay	Number Of Slices
64 Bit ECC By DMC	13.342 ns	362
64 Bit ECC by Hamming Code	12.192 ns	322

Table 2:- Comparison Table for 64 bit ECC in MCU

VI. CONCLUSION AND FUTURE SCOPE

A. Conclusion

In the present scenario, there is a high need to secure memory cells through protection codes in order to stabilize reliability level. Several types of error detection & correction techniques are applied for this. Complicated ECCs (error correction codes) along DMC (decimal matrix code) is used for eliminating the condition of data corruption in MCUs & protecting the memory. The issue related to them is that they need overhead with high delay. Lately, MCs (matrix codes) constituted over Hamming codes are proposed for protecting the memory.

In the research, we proposed novel ECC in the MCU along Hamming code for ensuring reliability of memory. An algorithm based over Hamming code was incorporated in proposed protection code in order to recognize more number of errors & correcting them. So obtained outcomes reveal that proposed schema gains a higher level of security in contrast to DMC.

As observed from session of outcomes, number of slices & delay in DMC is higher contrasting to proposed layout. As observed value of delay for a 64 bit ECC by DMC is 13.342 whereas number of slices are 362. For a 64 bit ECC by Hamming code, value of delay is 12.192 where number of slices are 322.

B. Future Scope

There is a further scope of improvisation of system by combination of Hamming code & DMC. In this Hamming code will tend to minimize the delay & area of circuitry. In the mentioned structure, there are 39 redundant bits are used which can leads to reduction in further improvisations.

References

[1] Jing Guo, Liyi Xiao, Member, IEEE, Zhigang Mao, Member, IEEE, and QiangZhao,"Enhanced memory reliability against multiple cell upsets using Decimal Matrix Code" IEEE Trans. Very Large ScaleIntegr. (VLSI) Syst., vol. 22, no. 1, pp.127-135, Mar 2013.

[2] D. Radaelli, H. Puchner, S. Wong, and S. Daniel, "Investigation of multi-bit upsets in a 150 nm technology SRAM device," IEEE Trans.Nucl. Sci., vol. 52, no. 6, pp. 2433–2437, Dec. 2005.
[3] 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.
[4] C. Argyrides and D. K. Pradhan, "Improved decoding algorithm for high reliable reed muller coding," in Proc. IEEE Int. Syst. On Chip Conf., Sep. 2007, pp. 95–98.
[5] A. Sanchez-Macian, P. Reviriego, and J. A. Maestro, "Hamming SEC-DAED and extended hamming SEC-DED-TAED codes through selective shortening and bit placement," IEEE Trans. Device Mater. Rel., to be published.
[6] S. Liu, P. Reviriego, and J. A. Maestro, "Efficient majority logic fault detection with difference-set codes for memory applications," IEEETrans. Very Large Scale Integr. (VLSI) Syst., vol. 20, no. 1, pp. 148–156, Jan. 2012.
[7] M. Zhu, L. Y. Xiao, L. L. Song, Y. J. Zhang, and H. W. Luo, "New mix codes for multiple bit upsets mitigation in fault-secure memories," Microelectron. J., vol. 42, no. 3, pp. 553–561, Mar. 2011.
[8] R. Naseer and J. Draper, "Parallel double error correcting code design to mitigate multi-bit upsets in SRAMs," in Proc. 34th Eur. Solid-StateCircuits, Sep. 2008, pp. 222–225.
[9] G. Neuberger, D. L. Kastensmidt, and R. Reis, "An automatic technique for optimizing Reed-Solomon codes to improve fault tolerance in memories," IEEE Design Test Comput., vol. 22, no. 1, pp. 50–58, Jan.–Feb. 2005.
[10] 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.
[11] S. Baeg, S. Wen, and R. Wong, "Interleaving distance selection with a soft error failure model," IEEE Trans. Nucl. Sci., vol. 56, no. 4, pp. 2111– 2118, Aug. 2009.
[12] K. Pagiamtzis and A. Sheikholeslami, "Content addressable memory (CAM) circuits and architectures: A tutorial and survey," IEEE J.Solid-State Circuits, vol. 41, no. 3, pp. 712–727, Mar. 2003.
[13] D. Radaelli, H. Puchner, S. Wong, and S. Daniel, "Investigation of multi-bit upsets in a 150 nm technology SRAM device," IEEE Trans. Nucl. Sci., vol. 52, no. 6, pp. 2433–2437, Dec. 2005.
[14] 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.
[15] C. Argyrides and D. K. Pradhan, "Improved decoding algorithm for high reliable reed muller coding," in Proc. IEEE Int. Syst. On Chip Conf., Sep. 2007, pp. 95–98.
[16] A. Sanchez-Macian, P. Reviriego, and J. A. Maestro, "Hamming SEC-DAED and extended hamming SEC-DED-TAED codes through selective shortening and bit placement," IEEE Trans. Device Mater. Rel., to be published.
[17] S. Liu, P. Reviriego, and J. A. Maestro, "Efficient majority logic fault detection with difference-set codes for memory applications," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 20, no. 1, pp. 148–156, Jan. 2012.
[18] M. Zhu, L. Y. Xiao, L. L. Song, Y. J. Zhang, and H. W. Luo, "New mix codes for multiple bit upsets mitigation in fault-secure memories," Microelectron. J., vol. 42, no. 3, pp. 553–561, Mar. 2011.
[19] R. Naseer and J. Draper, "Parallel double error correcting code design to mitigate multi-bit upsets in SRAMs," in Proc. 34th Eur. Solid-State Circuits, Sep. 2008, pp. 222–225.
[20] G. Neuberger, D. L. Kastensmidt, and R. Reis, "An automatic technique for optimizing Reed-Solomon codes to improve fault tolerance in memories," IEEE Design Test Comput., vol. 22, no. 1, pp. 50–58, Jan.–Feb. 2005.
[21] 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.
[22] S. Baeg, S. Wen, and R. Wong, "Interleaving distance selection with a soft error failure model," IEEE Trans. Nucl. Sci., vol. 56, no. 4, pp. 2111– 2118, Aug. 2009.
[23] Y. Yahagi, H. Yamaguchi, E. Ibe, H. Kameyama, M. Sato, T. Akioka, and S. Yamamoto, "A novel feature of neutron-induced multi-cell upsets in 130 and 180 nm SRAMs," *IEEE Trans. Nucl. Sci.*, vol. 54, no. 4, pp. 1030–1036, Aug. 2007.
[24] N. N. Mahatme, B. L. Bhuva, Y. P. Fang, and A. S. Oates, "Impact of strained-Si PMOS transistors on SRAM soft error rates," *IEEE Trans. Nucl. Sci.*, vol. 59, no. 4, pp. 845–850, Aug. 2012.
[25] C. A. Argyrides, P. Reviriego, D. K. Pradhan, and J. A. Maestro, "Matrix-based codes for adjacent error correction," *IEEE Trans. Nucl. Sci.*, vol. 57, no. 4, pp. 2106–2111, Aug. 2010.
[26] F. 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.
[27] Costas Argyrides, IEEE. Member, ‖Using Single Error Correction Codes To Protect Against Isolated Defects and Soft Errors‖, IEEE Transactions on Reliability, VOL 62, NO.1,March 2013.
[28] D. Bhavsar, ―An algorithm for row–column self-repair of RAMs and its implementation in the Alpha 21264,‖ in Proc. Int. Test Conf., 1999, pp. 311–318.
[29] I. Kim, Y. Zorian, G. Komoriya, H. Pham, F. Higgins, and J. Lewandowski, ―Built-in self repair for embedded high density SRAM,‖ in Proc. Int. Test Conf., 1998, pp. 1112–1119.
[30] M. Horiguchi, J. Etoh, M. Aoki, K. Itoh, and T.Matsumoto, ―A flexible redundancy technique for high-density DRAMs,‖ IEEE J. Solid-State Circuits, vol. 26, no. 1, pp. 12–17, Jan. 1991
[31] A. Dutta and N. Touba, ―Multiple bit upset tolerant memory using a selective cycle avoidance based SEC-DED-DAEC code,‖ in *Proc. 25th* IEEE VLSI Test Symp., May 2007, pp. 349–354.