Main Selection Orthogonal Lattice Square Codes Majority Logic decoder Based Double Error Detection And Correction In Communication Systems

S. Sravani*, K. Padma Vasavi **

Department Of Electronics And Communications,

Shri Vishnu Engineering College For Women (Autonomous), Bhimavaram, INDIA

ABSTRACT

In communication Systems information passing through a channel from transmitter to receiver was not completely received by the receiver due to loss of the signal or any noise generation within the network, this may reduce the capacity of the system. Therefore, an error detection and correction system at receiver's end can adjust and helps to reconstruct the signal. In this paper, a two block based 32-bit single error correction- double error detection based double error correction scheme with orthogonal lattice majority logic was compare with our proposing model Main selection Based Orthogonal Lattice Square codes Majority Logic based SEC-DED. Due to the complex nature of the existing circuit, only a limited contribution of error correction will be verified and corrected, therefore to reduce the limitation of latency and power consumption in the existing circuits we proposed MS-OLS-MLD scheme i.e., main selection orthogonal lattice square based majority logic decoder circuit with double error correction mode. Our proposing model results in 25 % efficiency of power, 50 % efficiency in time, compared with the existing scheme.

Key words: Encoder, Decoder, Memories, Double Error Correction Codes, MS-OLD-MLD, SECs.

INTRODUCTION

To detect the errors present in the received data bit stream. These codes contain some bits, which are included appended to the original bit stream. These codes detect the error, if it is occurred during transmission of the original data bit stream is known to be an error detection process. But the codes that are used to correct the errors present in the received data bit stream so that, original data can be retrieved known as error correction codes. Error correction codes also use the similar strategy of error detection codes

To safeguard memories error correction codes (ECC's) are mostly used. In this the errors doesn't corrupt the data they accumulate. These codes uses parity check bits to identify the error and to rectify the error. To correct the errors present in the bits encoding and decoding circuits are needed. Memory access time depends on encoding and decoding circuit so to reduce the access time we are using decoding circuit of less complexity.

Protecting memories against errors, error correction codes (ECCs) are used. It's necessity to use advanced ECCs due to occurrence of multiple error's. The latest ECC's that are used for memory protection are Orthogonal latin squares

(OLS) codes. These codes have less delay because of their ease and modularity of the decoding algorithm . A vital issue is that when ECCs used, the encoder and decoder may suffer errors. during this brief, multiple error correction technique using extended ols codes implemented .Then use of more complex codes which will correct more errors is restricted by their impact on delay and power, which can reduce their applicability to memory designs.To overcome issues, the utilization of codes that are one step majority logic decodable (OS-MLD) has evolved recently. OS-MLD codes can decode with low latency. Therefore they're want to protect memories.

Y. Bonhomme, V.Gherman, S. Evain proposed SEC-DED codes [2]. SEC-DED means single error correction-double error detection .It can correct one error and detect all possible double errors. It was mostly utilized in memories and caches. Now technology was rapidly increasing day by day. So it is necessary to improve the error correction capabilities like DEC codes i.e; Double error correction codes.

In comparision with SEC-DED codes, DEC codes are more difficult in terms of area, delay and power. To overcome this another DEC codes are proposed those are Horizantal Vertical diagonal codes and matrix codes.

S. Ahammed, M. S. Rahman proposed horizontal vertical diagonal codes [6]. In this error detection and correction is at high level. It uses parity check bits in all four directions. So the major limitation is more number of parity check bits are needed.

T.Kocak, C.Argyrides proposed matrix code [7]. Due to the occurance of high defects and multiple bit upsets matrix codes in memory chips are used to increase the realiablity and yield. So that decoding complexity can be reduced but the limitation is number of parity check bits are more so that the memory size becomes larger.

To reduce the parity check bits orthogonal latin square code or difference set codes are proposed but the drawback is 32 bit data width. But by using OLS it takes only certain limited blocks. To overcome that author's using DEC code which is a combination of DS codes with SEC-DED codes. Here we can also reduce the number of parity check bits.

This paper was organized as, Section-1 deals with error correction and detection definitions along with other fellow researchers work (Literature Survey) identity, Section-2 comprises with the system model of existing system, Section-3 helps in understanding the proposing system model with how MS-OLD-MLD process was implemented, Section-4 deals with simulation results of both systems and their effective comparison of performance metrics.

Existing System

It is a combination of DS and SEC-DED codes. It consists of both encoding and decoding process. Fig:1 Encoding consists of 3 blocks. In this 32 bit data words get separated in to these blocks .First and second block consist of 11 bits with SED-DED encoder and the third block consist of only 10 bits with DS encoder. Here the input for (20, 11) DS is xor gate. $s_{1a,}$ $s_{2a,}$ ….. $s_{5a,}$ are the parity check bits of first block and $s_{1b,}$ $s_{2b,}$ s_{5b} are the parity check bits of second block. The 3 rd block consists of a parity check bits $s_{1d,}$ $s_{2d,}$ …….. $s_{9d.}$ Totally it generates 19 check bits. This made increasing in testing portion of iterative error identification of the sytem.

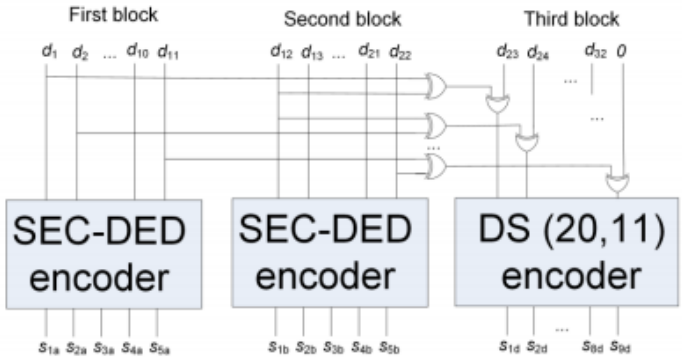

Fig:1 Encoder for the existing scheme

Now the data bits are used for double error correction. Fig:2 In the decoding process the first two blocks are SEC-DED error corrector and the last block is error locator .Firstly it undergoes syndroms and every block generates correction signals like $e_{1a,}$ .....$e_{11a,}$ $e_{2b,}$.....$e_{11b\ and\ e1d}$ .....e11d.

The first 2 blocks undergo SEC and along with correction signals. Double error detected signals are generated. The third block does not take data bits as inputs the input to this is data bits from 3 - blocks are combine in to xor.

If the correction signal is activated then the xor present in third block used to tell that the correction signal is imposed to error among the three data bits

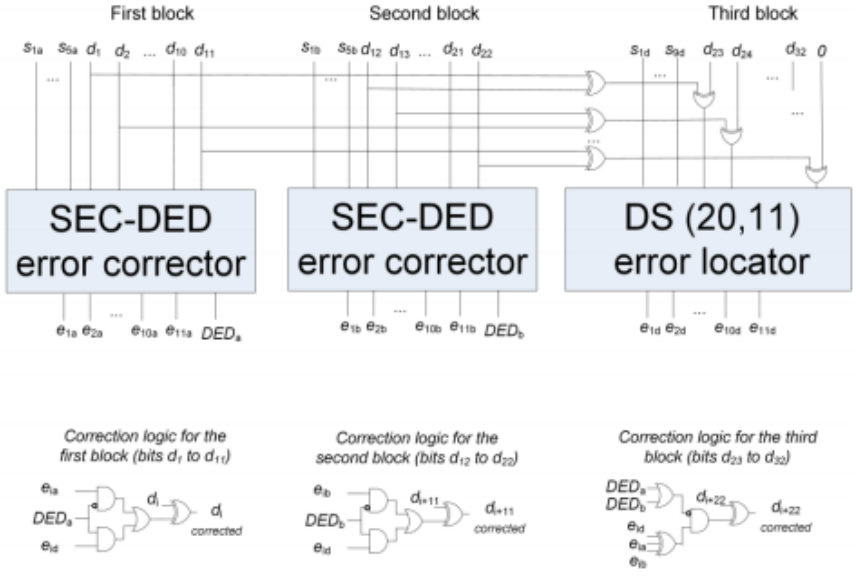

Fig:2 Decoder for the existing scheme

The following steps are required to use DEC for the correction signals because it doesn't use directly.

The following ways are used for correction of the data bits in first and second blocks:

If DED signal in first and second block is absence then SEC-DED code gives correction signal to fulfill correction.

- In this more than one bit error in block needs to be detected, for this reason, SEC for correcting the signal was impemented.
- If DED signal in first and second block is in active state, then DS code corrects the input data to fulfill correction process. In this SEC-DED blocks are not responsible for two errors detection that completely depends on DS block .So DS block will perform correction.
- The three blocks will result double error detection for every 12 bits by accessing the input from all the three blocks majority logic decoder will XOR the bits. Therefore the correction signal on DS encoder gets activated due to errors present in any one of the block. If suppose d_1 is error bit then e_{1d} correction signal gets activated.

Either of the two DED signals is in on then there will be no errors in the third block . Other than this block the correction has to undergo.

The input for the DS- encoder is xor of the two SED-DED blocks correction signal for that bit in DS-block is one if errors present in only third block than ds decoder has to be used . In other case, if errors present in first and second block it will be guarantee that no miscorrection on the third block. Example : If an error occur on bit d1 then e_{1g} and e_{1d} are the activated correction signals.

In this model, based on location of bits the correction was performed. If any error detected in the system, this will be operated using write logic "AND" by using majority logic "OR" operate to correct the data stream . In this 7- bits were considered for parity identifiers and detects any error was exist in the data stream or not, this might be a single or double error based error detection observes the position of the parity bit, but this led to a drawback of the system in terms of delay as this approach needs adjust for all bit verification and this leads to gets increase in the count of iterative statements. This iterative loop statements made the system more complex and power utility was also increased.

PROPOSED SYSTEM

The implementing method is based on the observation that by construction, the groups formed by the m parity bits in each Mi matrix have at most a one in every column of parity matrix. For the example: those groups correspond to bits (or rows) 1– 4, 5–8, 9–12 and 13–16. Therefore, any combination of four bits from one of those groups will at most share a one with the existing columns in a parity matrix. For example, the combination formed by bits 1, 2, 3, and 4 shares only bit 1 with columns 1, 2, 3, and 4. This is the condition needed to enable OS-MLD.

Therefore, combinations of four bits taken all from one of those groups can be used to add data bit columns to the parity matrix. For the code with k=16 and t=2 , we have m=4. Hence, one combination can be formed in each group

by setting all the positions in the group to one. This is shown in Fig. 3, where the columns added are highlighted. In this case, the data bit block is extended from k= 16 to k=20 bits.

Operation for ERROR Detection and Correction using MS-OLD-MLD DEC-DED

1111000000000000|1000|1000000000000000

0000111100000000|1000|0100000000000000

0000000011110000|1000|0010000000000000

0000000000001111|1000|0001000000000000

1000100010001000|0100|0000100000000000

0100010001000100|0100|0000010000000000

0010001000100010|0100|0000001000000000

0001000100010001|0100|0000000100000000

1000010000100001|0010|0000000010000000

0100100000010010|0010|0000000001000000

0010000110000100|0010|0000000000100000

0001001001001000|0010|0000000000010000

1000001000010100|0001|0000000000001000

0100000100101000|0001|0000000000000100

0010100001000001|0001|0000000000000010

[0001010010000010|0001|0000000000000001]

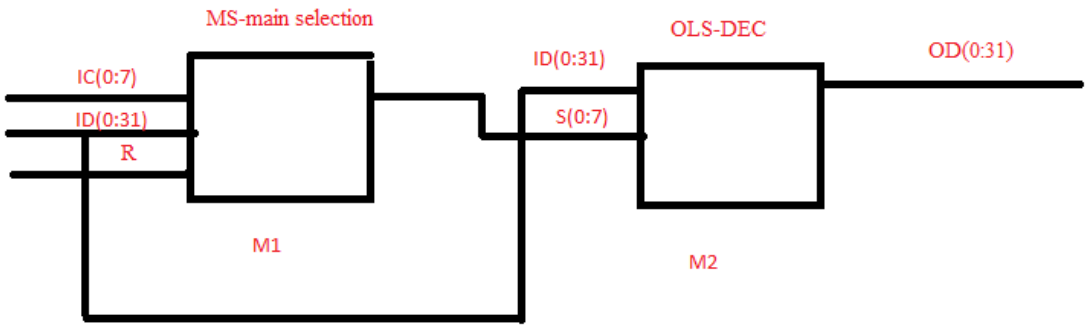

Fig- 3: MS OLS DEC based error correction code

The implementing method first divides the parity check bits into groups of m bits given by the Mi matrices. Then, the second step is for each group to find the combinations of 2t bits such that any pair of them share at most one bit. This second step can be seen as that of constructing an OS-MLD code with m parity check bits. To keep the OS-MLD property for the extended code, the combinations formed for each group have to share at most one bit with the combinations formed in the other $2t - 1$ group. When m is small, such combinations are found easily. When m is larger, several combinations can be formed in each group. This occurs, for example, when m = 8. In this case, the OLS code has a data block size k = 64. With eight positions in each group, now two combinations of four of them that share at most one position can be formed. This means that the extended code will have eight (4×2) additional data bits. As the size of the OLS code becomes larger, the number of combinations in a group also grows. For the case m = 16 and k= 256, each group has 16 elements. Interestingly enough, there are 20 combinations of four elements that share at most one element. Those combinations are obtained using the extended OLS code shown in Fig. 3 in each of the groups. Therefore, in this case, $4 \times 20 = 80$ data bits can be added in the extended code. The parameters of the extended codes are shown in Table I, where $n - k = 2tm$ is the number of parity bits. The data block size for the original OLS codes (k-OLS*)* is also shown for reference.

The proposing approach is a combination of selection of parity bits the correction based on error identified in parity bits. For this approach initially main selection of parity bits were identified.

ALGORITHM 1:Parity Selection

Input: X parity bits and positions.

Step 1: Create an Empty Stack S,
Step 2: Set Parity bits (I) and locations of input and recieved.
Step 3: Randomly Select S∈I
Step 4: Whiles Location and Count do
Step 5: If error detected
Step 6: S=(Location, Bit change)
Step 7: Share change bit
Step 8: else
Step 9: S=S
Step 10: update Count
Step 10: end
Step 11: end

Output: Error Location and bit to be changed.

The method can be applied to the general case of an OLS code with MLD k = m2 that can correct t errors. Such a code has 2tm parity bits that as before are divided into groups of m bits. The code can be extended by selecting combinations of 2t parity bits taken from each of the groups. These combinations can be added to the code as long

as any pair of the new combinations share at most one bit. When m is small, a set of such combinations with maximum size can be easily found.

However, as m grows, finding such a set is far from trivial (as mentioned before, solving that problem is equivalent to designing an OS-MLD code with m parity bits that can correct t errors). An upper bound on the number of possible combinations can be derived by observing that any pair of bits can appear only in one combination. Because each combination has 2t bits, there are (2t 2) pairs in each combination. The number of possible pairs in each group of m bits is m2. Therefore, the number of combinations N.G. in a group of m bits has to be such that.

In this brief, the error correction technique for extended OLS codes is implemented. The extended codes have the same number of parity bits as the original OLS codes but a more significant number of data bits. The data bits in extended OLS codes are 20 bits whereas parity bits are 16 bits. The number of errors extended OLS codes can correct is 2 adjacent errors. Therefore, the relative overhead is smaller. The derived codes can be decoded using OS-MLD as the original OLS codes. The decoding area and delay are also similar. Therefore, the new codes can be an exciting option to reduce the number of parity bits required to implement multiple bit error correction in memories or caches.

RESULTS

A Verilog HDL programming language used in this .Spartan 6family, FGG900 package ,XC6SLX150T device with a speed grade –3 are the design properties and simulated by uing Xillinx ISE 14.7 and ISIM simulator.

Fig:4 Simulation result for encoder

Fig:5 Simulation result for Decoder

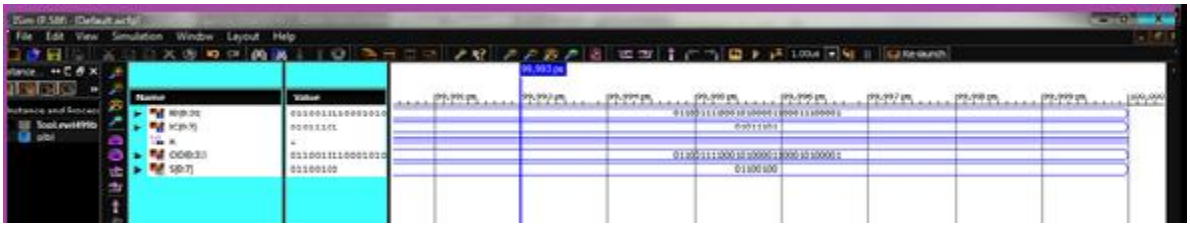

Fig:6 Simulation result for proposed work

In this power gets compared with a double error correction code for 32-bit data stream with efficient decoding and orthogonal lattice square based double error detection and correction.

Table:1 Comparison of power between a double error correction code for 32-bit data words with efficient decoding and MS-OLS-MLD correction code.

Power report of the system	Power in watts
A double error correction code for 32-bit data words with efficient decoding	15.311w
MS-OLS-MLD correction code	11.484w

Table :1shows the power comparison table for a double error correcton code for 32-bit data words with efficient decoding and MS-OLS-MLD correction code.In MS-OLS-MLD correction code It reduces to w then a double error correction code for 32- bit data words with efficient decoding.

Table:2 Performance metrics Comparison between a double error correction code for 32-bit data words with efficient decoding and MS-OLS-MLD correction code

S.NO	Parameters	A double error correction code for 32-bit data words with efficient decoding	MS-OLS-MLDcorrection code
1	No: of slice LUT's	76	**55**
2	No: of occupied slices	43	**26**
3	No: of bonded IOB's in %	22%	**18%**
4	No: of LUT's flipflop pairs used	76	**55**
5	Delay	2.889ns	**2.603ns**
6	Power	15.311w	**11.484w**

In table: 2 various parameters are compared like area, delay, power. In MS-OLS-MLD correction code power and area are reduced when compared with a double error correction code for 32-bit data words with efficient decoding.

CONCLUSION

In this proposing work, MS-OLS-DEC codes have been implemented. In the majority logic based OLS codes with support of SEC implemented for minimising the latency and reducing the power drain of the system. The implemented results were completely relay in ols approach. This results in 50% efficiency in latency and 25% effective in power consumption. In future unequal error protection codes to verify double error for the data in combination with ols needs to be developed for improving (in reduction) of the delay.

REFERENCES

[1] Liyi Xiao, Pedro Reviriego,Jiaqiang proposed "A double error correction code for 32-bit data words with efficient decoding", In Proc.IEEE on DEVICE and Material Reliability.,in 2017.

[2] L. Xiao, A. Sanchez Macian, S. Liu, P. Reviriego, J.A. Maestro, Comments on "Extend orthogonal Latin square codes for 32-bit data protection in memory applications, Microelectronics Reliability, (2016)", Microelectronics Reliability, Elsevier, vol. 69, February 2017, pp. 126-129.

[3] J. A. Maestro, L. Xiao and S. Liu "Reducing the Cost of Triple Adjacent Error Correction in Double Error Correction Orthogonal Latin Square Codes," in IEEE Transactions on Device and Materials Reliability, vol. 16, no. 2, pp. 269-271, June 2016, doi: 10.1109/TDMR.2016.2547187.

[4] S. Ahammed, M.S. Rahman Soft error tolerance using Horizontal-Vertical-Double-Bit Diagonal parity method, International Conference on Electrical Engineering and Information Communication Technology. IEEE, 2015, pp.1-6.

[5] J.A. Maestro, M. Demirci, P. Reviriego, Optimized Parallel Decoding of Difference Set Codes for High Speed Memories, Microelectronics Reliability, vol. 54, No 11, November 2014, pp. 2645-264

[6] Y. Bonhomme, , N. Seymour , S. Evain, and V. Gherman Generalized parity check matrices for SEC-DED codes with fixed parity, in Proc. IEEE On Line Testing Symp., July 2011.

[7] J. Draper and R. Naseer, DEC ECC design to improve memory reliability in sub 100 nm technologies, Proc. IEEE ICECS, pp. 586- 589, 2008.

[8] N. A. Touba and A. Dutta "Multiple bit upset tolerant memory using a selective cycle avoidance based SEC-DED-DAEC code," in Proc. 25th IEEE VLSI Test Symp. May 2007.