Implementation of Multi Bit Error Detection and Correction using LDPC Codes

M.SHALINI, Sri.P.JAYARAMI REDDY
M.Tech Student, Assistant Professor
Department Of ECE
Dr.K.V.S.R. Institute Of Technology, Kurnool

ABSTRACT

On behalf of technology scaling, on-chip memories in a die undergoes bit errors because of single events or multiple cell upsets by the ecological factors such as cosmic radiation, alpha, neutron particles or due to maximum temperature in space, leads to data corruption. Error detection and correction techniques (ECC) recognize and rectify the corrupted data over communication channel. In this paper, an advanced error correction 2-dimensional code based on divide-symbol is proposed to weaken radiation-induced MCUs in memory for space applications. For encoding data bits, diagonal bits, parity bits and check bits were analyzed by XOR operation. To recover the data, again XOR operation was performed between the encoded bits and the recalculated encoded bits. After analyzing, verification, selection and correction process takes place.

A typical system-level technique to harden memory against multiple bit upsets (MBUs) would be the use of error correction codes (ECCs) for enhanced correction capabilities. Building updated ECCs with low redundancy and correction of errors however has be a significant issue, especially about adjacent ECCs. Present MBU mitigation codes concentrate primarily on correcting up to 3-bit explosive errors. The amount of impaired bits will quickly extend to even more than 3 bit as that of the software scales as well as the cell interval gap decrease. Consequently, the earlier approaches are not adequate to meet the criterion for durability in harsh conditions. In this article, a technique for 4-bit bursting bug fix (BEC) codes was introduced with a Multiple biterror detection andcorrection(MBEDC)codes. Initially you define the interface principles, then you create a search algorithm for locate the codes that correspond to both the rules. Usable are the 4-bit BEC H matrices with MBEDC codes. Any additional parity check bits were needed compared with such a BEC 3-bit code. That efficiency of 4-bit BEC was also substantially enhanced by adding the latest algorithm with existing 3-bit BEC codes.A project with verilog HDL would be built. The Simulation & Synthesis Xilinx ISE method is used.

I.INTRODUCTION

Error correction codes are widely used to secure and this so-called Soft Error Memory by destroying the circuit that affects the logical significance of memory cells. With technology scales, memory devices develop larger and error correction codes are becoming more efficient. To this end it was recently recommended to use more sophisticated codes.Ses codes may remedy further mistakes, but typically need complicated decoders. The use of one-stage, primarily decodable logic keys was first suggested for memory applications to prevent a large decoding difficulty. With a rather simple circuitry, however long decoding times are required, one-step major decoding can be implemented. In such a memory, the access time will be expanded. There could only be several coding groups decoded by OS-MLD. These comprise a range of DS-LDPC, EG-LDPC and OLS codes.

The usage of OLS codes has become extremely relevant with respect to interconnections, memories and caches. Because of its modularity, their error correcting capabilities will quickly be tailored to the error rate or operating mode. In order to fix the same amount of errors OLS codes usually need more parity bits instead of other codes. That modularity as well as the fast and quick decoding method (since OS-MLD is the OLS code) can compensate in many applications for this disadvantage. A big concern is the probability of errors with the required encoder and decoder circuits (ECCs). An incorrect term can be entered throughout the memory whenever a mistake influences the encoder. An error of the decoder will contribute to the misunderstanding of a proper word or even the misunderstanding of a wrong word as the right word.

A strategy for speeding the serial execution of the encoding of DS-LDPC codes for maximum logic were recently suggested. The goal of the process is to use the first big logic decoding increments to detect whether the decoded word requires explanation. When no errors occur, decoding may be prevented without the remaining iterations, thereby decreasing decoding period considerably. More logic decoding with simple hardware can be performed serially, and that takes a long decoding duration. This raises the memory access time for device applications. The procedure detects whether a word has mistakes during the first major logical decoding iterations and where no errors arise the decoding stops without the remainder of both the iterations done. As most of the terms in a memory are error-free, the mean time for encoding is decreased dramatically.

1.1 Motivation

That soft error rates of complex electronic components differ throughout their lifespan. The "bath curve" for system failures usually starts with a very high child death rate, which smoothens for just a time but instead increases steadily towards the end of the development phase. Moreover, the predicted fault rate (e.g., temperature, radiation levels, local radio transmission) may vary with both the environmental stress. Classique, to accommodate the maximum predicted failure rate throughout the lifetime for a system, one must design error correction codes (CECs) which lead in 'wasteful' sensitivity of 'too many' failures over a life cycle of the product. The goal is to monitor the decrease in failure tolerances during a portion's lifespan at the highest levels of the work discussed here. More specifically, it is our task to allow malfunction and failure tolerance, device power traded, and other key parameters to be established. For eg, they would really like to switch off some power-hungry ECC circuits whether we are able to identify and reset sub-components after such a cycle of high-fault infant death and predicted that the fault response time is considerably decreased over a significant length of time. Then we'd like to turn those ECC circuits on again when the fault rates climbed again.

1.2 Literature Survey

QiujuDiao, Ying Yu Tai, Shu Lin, Khaled Abdel-Ghaffar proposed on “Trapping Set Structure of LDPC Codes on Finite Geometries”

LDPC codes are the most promising coding tool for achieving the Shannon capabilities of different networks at present. They do well with iterative, belief propagation-based decoding. And most LDPC codes have such a general extreme vulnerability, referred to as error level with iterative decoding. Error Floor might prohibit applications that need very low error rates from obtaining LDPC codes. The error level of both the AWGN channel is largely attributed to an unintended layout, known as a trapping package, in the Tanner code graph. This geometric methodology is used to evaluate the trapping set configuration for LDPC codes centred on finite geometer, named LDPC finite geometry (FG). Trapping structures are shown by subgeometries of the geometry over which the code was designed throughout the Tanner Graph of even a FG – LDPC code. This geometrical description can be used to extract and evaluate boundaries and configurations of FG LDPC code trapped sets.

Juntan Zhang, Jonathan S. Yedidia, and Marc P.C. Fossorier suggested the EG LDPC codes low latency encoding method

We define simulated annealing codes to Euclidean Geometrie dependent low-density parity-check codes, which are ideal for use in applications that need very rapid decoders. The decoders are based on mixed and repeated Bit-flitting iterative (BF) & quantized BF structures. That decoders proposed converge faster than regular BF decoders, offering a better efficiency. We present simulations to demonstrate how these decoders function against uncertainty. In certain instances, we can prove that no major mistake occurs by important sampling.

"On the basis of RCD SPC codes for arrays and their equivalence to codes rendered from Euclidia geometries and partial BIBDs" suggested by Amitkumar Mahadevan and Joel Morris.

We present a technique for constructing a Gallager family for low density parity-check codes (LDPC), centered on η ×η sequence in which the η become prime with μm2 diagonal bundles reflecting equations for parity-check. These γ= 3 codes are referred to as the single-parity search (SPC) row-column-diagonal (RCD). The equivalence is provided for the Gallager LDPC codes for¨¨¨ diagonal bundles, in which¯ is primary, as well as the Euclidean geometry (EG) codes in which Gallager LDPC codes were constructed, with parameters m=2, s=1,¨ and LDPC codes constructed throughout the (¯)¨ grating of both the gallager LDPCs, in¯ square arrays. They define each building technique, demonstrating that the three building techniques are similar to instances.

"Radiation-induced soft error in advanced semiconductor technologies" suggested by Robert C. Baumann

The formerly ephemeral soft error triggered by radiation is becoming a significant challenge for sophisticated electronic commercial substances and systems. Left without problems, soft errors have the potential to cause all the other reliability processes to combine the maximum failure rate. These article describes briefly some types to soft error component failures, the 3 predominant radiation mechanisms which produce soft error in terrestrial applications as well as how the set of radiation-induced charge generates these soft mistakes. Applications also are certainly to be mitigated in soft errors, that soft sensitivity as just a result of both the technology scaling with different memory & logic components would then be addressed.

Saad Bin Qaisar suggested "Low Density Parity Verification Codes on Finite and Incomplete Underground Architecture"

II.PROPOSED METHOD

In space, due to high temperatures, electronic circuits, in particular memories subject to soft errors lead to poor reliability. Memory cells are interrupted by neutron or alpha particles from earth’s atmosphere [3]. One way to reduce these errors by maximizing the critical charge at the state

nodes or by well and substrate techniques (process related techniques). Another way is, by applying error correction codes (ECC) on memories with that some errors can be overcome [4]. This is usually performed by single error correction (SEC) code on each memory to deal with independent errors. Scrubbing with SEC increases accuracy.

It reads memory and corrects the single errors time to time; will not accumulate over time [5]. This is not effective for multiple cell upsets (MCUs) because in this MCU two or more bits of same memory gets affected. To overcome the multiple cell upsets in memories, interleaving method is proposed [6]. Many studies have been takes place with this technique to manage multiple cell upsets (MCUs). But this proposed interleaving method increases the system design complexity and shows impact in area and power consumption. So that ECCs is used in case for multiple cell upsets (MCUs). This requires more parity bits, more time for decoding the bits and more complex circuits are needed to perform for the encoding and decoding operations. Various ECCs focused to reduce area, delay and power.

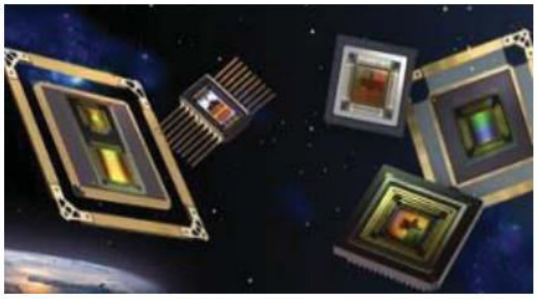


Fig. 1 ICs for space applications

Zhu et al., [7] proposes a new error correction codes for the reduction of radiation exposed multiple bit upsets in memories. This detects and corrects the adjacent double bit errors and also lowers the errors for the non-adjacent double bit errors. The experimental results demonstrate that it reduces 40% hardware redundancy and more efficient compared to other existing ECC codes. Also, this method minimizes the errors for non-adjacent DBE by 12% when compared with the conventional SEC-DED-DAEC codes leads to high reliability memory system design.

Pedro et al., [8] has introduced an efficient single and double adjacent error correcting parallel decoder for the (24, 12) extended Golay code. In this parallel decoder, first a 12 bit OR gate is used for the implementation of foremost and the rest is implemented using 24 bit OR gate. Using HDL, and with the mapping of TSMC 65 nm technology of synopsis design compiler synthesized twice to reduce the area and delay. That means, the reductions are around 45% and 11% for area and the reductions 28% and 21% for delay when relate with existing SEC- DAEC decoder.

A new decimal matrix code ECC is used to enhance the memory reliability in [9]. The encoder and decoder are synthesized in SMIC 0.18 µm technology by synopsis design compiler. Simulation result shows that the MTTF of this technique is 452.9%, 154.6%, 122.6% and the delay is 73.1%, 69.0%, 26.2% for hamming, MC and PDS respectively.

Using different set of codes (cyclic-linear block codes) as ECC, protect the memory from data loss is proposed in [12]. This scheme exploits the localization of MCU errors, also the features of DS codes to enhance error correction possibilities and to reduce the decoding time. This is implemented in HDL and the simulation result indicates that this technique is effective in reducing the decoding time and also the area and power consumption. Reviregо et al., [13] suggested a new code to correct triple adjacent errors (SEC-DAEC-TAEC) and 3-bit burst errors for different data word lengths (16, 32 and 64 data bits).

Two optimization criteria have been used; reducing the total number of ones in the parity check matrix minimizes decoding time and the maximum number of ones in its rows optimizes the speed. Andrew et al., [14] proposed a turbo system and also lowdensity parity-check code for space engineering. This turbo system and LPDC codes are widely used in the data transmission for the aircraft applications. The error correcting code with low power consumption from space has been introduced in an encoding-decoding manner.

Low-density parity-check codes are organized in the form of parity check matrix, where the code rate reduces then this parity check matrix gets increases and therefore, this decoder is more complex. The development of the turbo codes and LPDC codes increases the efficiency and reliability of the system compared to the existing error correction codes in the deep space applications. Alternatively, turbo codes are formed on trellises. There will be one trellis section/information bit, for various code symbols. Therefore, turbo codes are more superior to low-density parity-check codes at low rate. In comparison, with other decoding methods, this iterative decoding of either turbo or LDPC code is more complex.

A new forward error correction code (FEC Encoder) is introduced for DVB S2 system with BCH code and LPDC code and also using QPSK modulation [15]. For FPGA implementation, the code length of 64800 bits and ½ codes normal rate LDPC code is considered. The design is processed

at 122 MHz for timing specifications. Pipelining technology is also included in the design to improve the coding efficiency. In this paper, a new encoding-decoding algorithm is proposed for the error correction and detection in multiple cell upset (MCUs). For encoding, data bits, diagonal bits, parity bits and check bits were examined using the XOR operation. And to recover the original data, again XOR operation was performed between the redundancy bits and the recalculated redundancy bits. After analyzing, verification, selection and correction process takes place in the decoding process. This section gives the introduction and the related works.

ERROR CORRECTION AND DETECTION SCHEME

To enhance memory reliability, a new error correction 2-dimensional code (2D-ECC) is proposed. This algorithm detects and corrects errors effectively when relates with other existing error correction techniques. This performs data region division, redundancy and syndrome calculation, verification and region selection one by one to recover the original data. Boolean XOR operation is performed which is most widely used in cryptography and also in generating parity bits for error checking and fault tolerance. The block diagram of the proposed ECC methodology is shown in fig. 2.

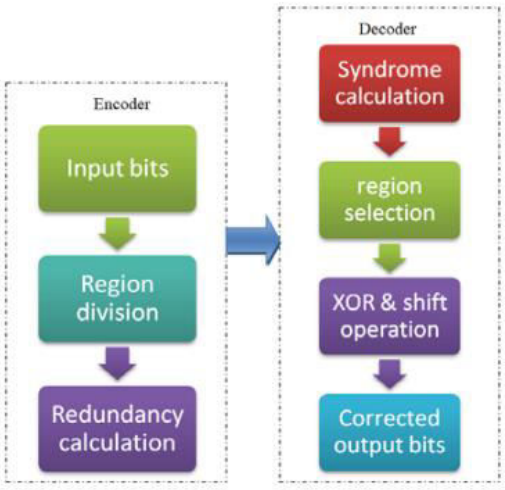


Fig. 2 ECC methodology

This 2-dimensional algorithm performs encoding-decoding process which codifies 16 bit input data into 32 bits in encoding and while decoding again the original 16 bit data is recovered.

Proposed Algorithm

STEP 1: Read the input 16 bit data (A16 – A0)

STEP 2: Divide the input data into 4 groups

X_1	Y_1	Z_1	W_1
X_2	Y_2	Z_2	W_2
X_3	Y_3	Z_3	W_3
X_4	Y_4	Z_4	W_4

STEP 3: Analyze diagonal bits, parity bits and check bits using XOR operation.

i) Diagonal bits (D1, D2, D3, D4) using XOR operation as the 2×2 matrix,

$$D_1 = X_1 \oplus Y_2 \oplus Z_1 \oplus W_2$$
$$D_2 = X_2 \oplus Y_1 \oplus Z_2 \oplus W_1$$

ii) Parity bits (P1, P2, P3, P4) using XOR operation taking the first bits, second bits, third bits and the fourth bits from four groups

$$P_1 = X_1 \oplus Y_1 \oplus Z_1 \oplus W_1$$
$$P_2 = X_2 \oplus Y_2 \oplus Z_2 \oplus W_2$$

iii) Check bits (Cx, Cy, Cz, Cw) using XOR operation by taking the alternative bits

$$Cx_{13} = X_1 \oplus X_3$$
$$Cx_{24} = X_2 \oplus X_4$$
$$Cy_{13} = Y_1 \oplus Y_3$$
$$Cy_{24} = Y_2 \oplus Y_4$$

STEP 4: Calculate the syndrome values for diagonal, parity and check bits by performing XOR operation between the redundancy data stored and the recalculated redundancy bits (RDi, RPi, RCi)

$$SD_i = D_i \oplus RD_i$$
$$SP_i = P_i \oplus RP_i$$
$$SC_i = C_i \oplus RC_i$$

Wherei = 1, 2, 3, 4

STEP 5: Check the following conditions to identify the error that to be satisfied

i) SDi and SPi bits have atleast one value similar to 1

ii) ii) More than one SCi value was similar to 1

STEP 6: Perform region selection and change the erroneous data to get the corrected output.

Process of encoding First, divide the 16 input bits into four groups (Xi, Yi, Zi, Wi). The diagonal bits (Di), parity bits (Pi) and check bits (Ci) are

determined using XOR operation. In the process of encoding, the input 16 bits gets converted into 32 bits (redundancy bits).

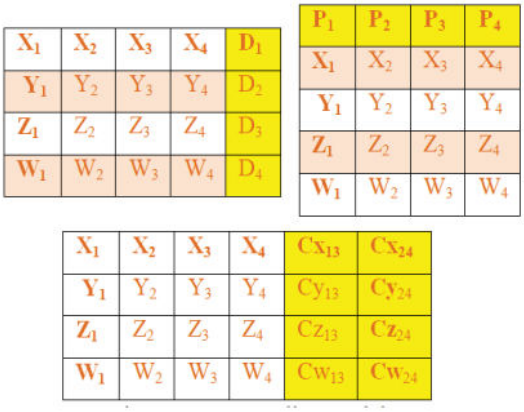


Fig. 3 Encoding model

Process of decoding

In decoding, the syndrome calculation has been analyzed with the encoded data and the recalculated encoded bits (SDi, SPi and SCi). After that, verification, region selection and correction can be performed.

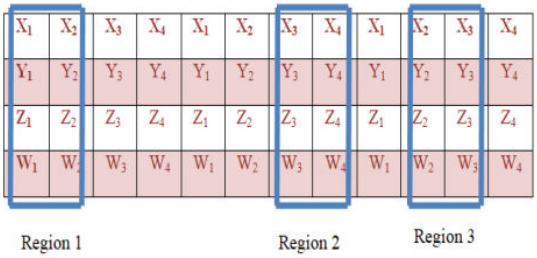


Fig.4 Different regions of data bits

Divide the data bits into regions 1, 2 and 3. This is formed by dividing the data bits by columns (1&2, 3&4, 2&3) to get efficient results.

Region	Selection criteria
Region 1	SD1+SD2+SP1+SP2 > SD3+SD4+SP3+SP4
Region 2	SD1+SD2+SP1+SP2 < SD3+SD4+SP3+SP4
Region 3	SD1+SD2+SP1+SP2 = SD3+SD4+SP3+SP4

Table 1 Region selection criteria

In table 1, the equations derived from SDi and SPi. The region 1 and region 2 with more syndrome bits equal to 1 has the error bits and the region 3 for which both equations equal. This algorithm could be most widely used in space engineering for the error correction and detection of bits during transmission of information.

III.EXTENSION METHOD

For biomedical devices, durability is an essential prerequisite. In electronic devices, memories of data storage elements play an essential function. They are usually shown on a processor and embedded applications throughout the device. Memories reflect a significant part of both the circuit region in these systems. This triggers more spatial radiation to memories than in others. The susceptibility to memory radiation has thus become a crucial concern in maintaining the safety of electronic devices. In contemporary static random access memories (SRAM) two common single event consequences are soft radiation-induced failures in such a single event upset (SEU) and multi-bit upset (MBU). Unless the semi-conductor technology evolves through submicrometer technology towards UDSM, their memory cells have a smaller scale and the radius influenced by a particle has a wider range as seen in Fig.4.1. Their size of both the memory cells becomes smaller.

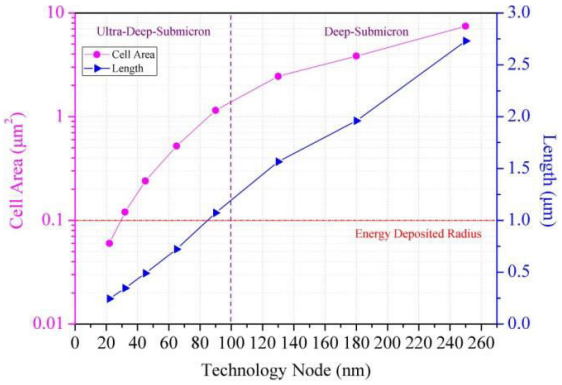


Fig. 4.1 Memory cell area of different technology (cell area shape is simplified to a square and Length is the length of side).

Whenever a cosmic ray particles touches the base memory cell, their electron-hole pairs radially distributed throughout the transport path. These created electron-holes will lead towards soft bugs by modifying the values stored throughout the memory cell that lead towards data damage and device malfunction. The radiation incident only impacts single memory cell with integrated circuits with a broad feature scale, which implies only its SEU exists. Throughout this situation single error-correction (SEC) codes were necessary to shield memory from radiational consequences, such as dual error detection (DED).

The vital load decreases and the region of the memories scales with each successive technology node decreases as the function size reaches into the DSM range. Which makes it easier to impact more synaptic connections by the particle struck in the figure. 4.2. That electron holes created by the substrate will spread to surrounding cells which induce MBUs through CMOS bulk technologies by reducing the cell- to-cell distance..

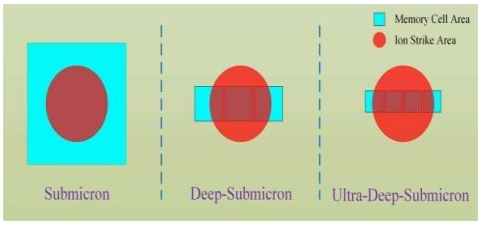


Fig 4.2. Schematic description of memory cells included in the radiation effect with variation of the technology node

This is opposed to FDSOI, which separates transistors but eliminates the multi-collecting process. That multi-collection process is also much more critical for a large-scale technology, and the risk of MBU is greater. ECCs which fix neighbouring bit errors or many bit mistakes are used to safeguard against MBUs.

While several bit error correction codes (ECCs) can fix a number of error detection, the difficulty of both the decoding method and constraint of both the key length limit their usage in either error pattern except restricted to adjacent bits. For the meantime, the MBU form relies on the original angle of incidence but scattering angles of both the beta electrons according to the generation theory of MBU. Therefore, neighbouring bit errors are the prevalent patterns of errors within multiple bit errors. Thus, adjacent bits of ECCs for correction within memory-hard designs becomes common. EtMany codes are suggested as well as the adjacent bits repair potential is primarily based on double adjacent error correction (DAEC), Triple Adjacent error correction(TAEC) & 3-bit Burst (BEC) error correction. A SEC or SEC-DED code paired with an interlocking of both the memory cells is also an alternative to codes that can fix any adjacent errors. It ensures the physical separation of cells which contribute to the very same logical term. This implies that an error on many neighbouring cells affects many terms, each with a single bit error which can be resolved by such an SEC code. When stated in previous research, interleaving complexifies memory interrelationships and routing and can result in an improvement in scope and energy usage as well as in the image resolution limitations. Consequently, it is architecture dependant and the two options are of concern to include an SEC plus interleaving or a code that can fix adjacent errors. As the technology arrives at just the UDSM, the memory cell region begins to decrease and sometimes even atomic transistor memory emerges. Many cell memory throughout word-line direction as seen in Fig 4.2 could be located in the spectrum of ionisation with intensity in the micrometre. Until considered as three bits. This implies that even a memory stability can't be guaranteed by the SEC-DAEC-TAEC codes.

More sophisticated correction codes are requested. Codes with poor redundancy were introduced, such example, for SEC-DAEC-TAEC and for 3-bit BEC. That correction between adjacent bit errors would thus be of importance to be introduced, particularly if it can be achieved without inserting additional parity bits.

In this article, the 3-bit BEC codes have strengthened to include multiple bit error correction (MBEDC codes) as well.

There are two facets of the code architecture technologies for the low redundancy MBEDC codes:

1) Error space satisfiability 2) Unique syndrome satisfiability.

MBEDC codescodes were valid for 16, 32 and 64 data bits. Process models were used to increase the performance of decoders as well as the decoder latency at ECC stage with an integrated circuit architecture perspective:

1) Minimizing the overall parity control matrix number

2) Minimization of the parity control matrix number in the heaviest lines.

In addition, an algorithm with weight limitation feature and documentation of both the past process is built based on the traditional recursive backtracking algorithm. The latest algorithm not only decreases software run-time costs but also greatly enhances 4-bit BEC codes efficiency. In Verilog hardware definition language (HDL), the encoders and decoders for MBEDC codescodes are introduced. The region and latency overhead were moderate to improve the correction capacity in contrast to the previous 3-bit BEC codes.

4.2 Binary Block Linear Codes

Codes have also been suggested in past projects

for SEC-DAEC-DED, SEC-DAEC-TAEC and 3-bit BEC. They are both binary linear codes of both the block. That procedure for developing the codes becomes focused on some principles for the creation of linear block codes. Throughout this article, binary linear block codes are indeed possible and identical building laws are followed. The numbers of databit k, redundancy bits, (n-k) and encoded-word sizes, n, are typically specified in binary data. The converter G matrix or the singular value matrix of H in a (n, k) number is described

$$G = [P_{k\times(n-k)} \cdot I_{k\times k}] \quad H = [P^T \cdot I_{(n-k)}] \qquad (1)$$

Throughout the encoding process, another generator matrix G is used to encode the databits through the procedure into P. Whereby Ik* k is really the defining matrix, P is the substance of k * (n-k) and PT seems to be the transposition of P.

$v = u \cdot G$ (2)

Where you encode u(u0, u1...uk−1) where v(v0, v1...vn– 1, was its encrypted codeword. That parity search matrix H would be used in the decoding phase to decipher the obtained codeword through all the process

$S = r \cdot H^T$(3)

Where even the codeword obtained is r(r0, r1...rn−1, the condition is S(s0, s1...sn−k−1) as well as an essential parameter through error correction. The inserted errors throughout the receiver code can be represented with

r =v +e (e0, e1...en−1) (4)

Where e (e0,e1, ... ,en−1) seems to be the velocity error which really signals the presence of an error throughout the ith bit, ei=1. Whenever a codeword r with both the Error Vector e = (e0, e1...en-001) has several Bit mistakes, the code syndrome can be determined by using the formula in respect of the error vector.

$S = e \cdot H^T$. (5)

The relation among syndrome as well as the corresponding trend is formulated throughout this equation. Taking into account the comprehensive parity matrix H form, the respective syndrome is equivalent to the vector of both the ith column if an error exists throughout the ith bit. If the ith bit as well as the jth bit errors exist, the respective syndrome is equivalent to the xor of both the vector including its Ith-column and even the Jth-column. Therefore it meets the following laws, whether a mistake may be changed or found.

1) Correct Restriction: throughout the syndrome range, the corresponding syndrome variable is special.

2) Detectable restriction: non-zero variable of the related syndrome.

The definition of Latin Squares is the foundation of both the OLS codes. Latin square m size is indeed a m by m matrix which has permutations through its rows and also in its columns of digits 0, 1... m − 1. If any ordered pair of elements occurs just once, two Latin squares are orthogonal. OLS is a product of OLS protocols. Such codes have k = m2 and 2tm known as descriptive, where t has been the error number which can be corrected. The code t = 2 and thus 4 m search bits can be used for double error checking. One benefit of OLS codes though is that structure is modular, as indicated in the description. In other words, clearly 2 m verification bits are applied to the code that really can fix t+1 errors to get a code that really can correct t 0 errors. This could be beneficial when applying adaptive bug fixes. The modular function also helps you to choose the option to repair errors for a specific word size. As already stated, OLS codes could be decoded by using OS-MLD, because each data bit is exactly 2 t and now one of them takes part. This makes for a quick patch whether the amount of bits is t or fewer. The 2 t control parts are recalled and a ballot by a plurality is held. The bit is errored, which should be corrected if another value of something is received. The bit becomes right otherwise. Under the worst case, any residual t −1 errors will impact t −1 check bits, as long as the number of errors is t or less.

$$\mathbf{H}=\begin{bmatrix}
1&0&0&0&0&0&0&1&0&0&1&0&1&0&1&1&1&0&0&0&1&0&0\\
0&1&0&0&0&0&0&0&0&1&0&1&1&0&1&1&0&0&1&0&0&0&1&0\\
0&0&1&0&0&0&0&0&0&0&1&0&0&0&1&0&0&1&1&0&1&0&0&1\\
0&0&0&1&0&0&0&1&0&0&1&0&0&0&0&1&0&1&0&1&1&0&0&0\\
0&0&0&0&1&0&0&0&1&0&0&1&1&1&1&0&0&0&0&0&1&1&0&0\\
0&0&0&0&0&1&0&0&1&0&0&0&1&1&0&0&0&1&0&1&0&0&1\\
0&0&0&0&0&0&1&0&0&1&0&0&1&0&0&1&0&1&1&1&0&1&1
\end{bmatrix}$$

Fig.4.3. Parity check matrix for OLS code with k = 16 and t = 1.

(1)Therefore, also a plurality of t+1 causes an incorrect bit fix. In all events, decoding begins with the singular value bits recalculated and the retained parity checkbits verified. OLS codes have been developed from either the parity search matrix H. This same matrix for k=14 and 8 test bits to fix single mistakes is illustrated in the figure as an illustration. 1. For codes which can repair further mistakes, the modular nature of OLS codes this structure is part of both the H matrix. For instance, seven consecutive rows were introduced into the H matrix and acquire another code that really can fix two errors. That H matrix for both the SEC OLS is developed for an arbitrary value of k = m2 as follows:

$$H = \begin{bmatrix} M_1 \\ M_2 \end{bmatrix} \quad I_{2m}$$

M2 matrices were size m — or otherwise m2 — in which I2 m is identity matrix of 2 m but m1 size. For each row, that matrix M1 have m. R − 1 is in place − m − 1,(r − 1) − m+ 2, ... (r − 2) − m − 1, (r − 1) − m − 1 − m − m − m. For row r − 1) The M2 matrix was designed accordingly:
M2 = [In the. In]. In. (2) Accessible in German.
For m = 4 the Fig. is easily evident between matrices M1 and M2. 1. The encoding matrix G is only the H matrix for extracting the control bits.

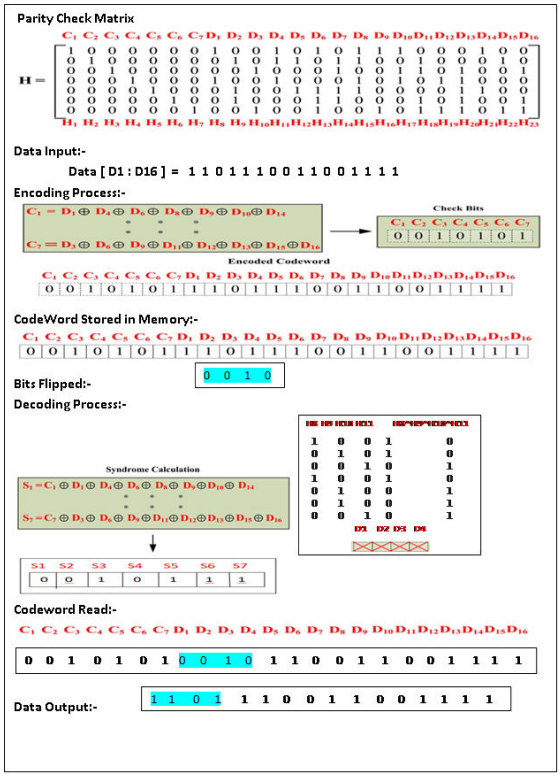
$$G = \begin{bmatrix} M_1 \\ M_2 \end{bmatrix}.$$

. (3)

In short, the encoder uses the G matrix, which would be derived through Latin Squares and also has the following characteristics, to calculate k = m2 data bits (di) for 2tm symmetry test bits (ci).
1) Every data bit is exactly involved in 2 t parity controls.
2) The parity sensors provide a pair with data bits (both bits).

IV. 4PROCEDURE OF ENCODING AND DECODING FOR MBEDC

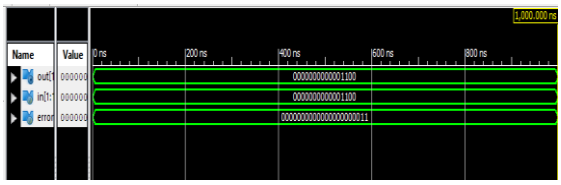
.



That H matrix used throughout Fig is seen in an illustration for 16 transmitted data below. 4.3. The Commission's plan. That check bits were determined either by corresponding transmitted data depending on the structure of the parity check matrix. That latest codeword, verify bits even data bits are saved throughout the memory. That material of damaged memory cells is changed as the particles enter the memory that contributes to MBUs. In the following segment, four times that adjacent bit is flipped for D1, D2, D3 and D4 for the clarification capabilities of MBEDC codescodes. That syndrome becomes computed by the retained check bit through data bit as well as the parity check matrix structure throughout the decryption stage. The flipped parts could be found thanks to the associated association between the syndromes as well as the column XOR outcome. With the reversed bits, errors were easily fixed from the storage level. And that is the whole encoding & decoding process for both the MBEDC codescodes suggested.

V.SIMULATION RESULTS

Proposed results:



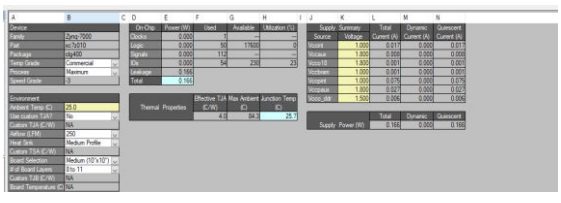
DESIGN SUMMARY:

Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slice LUTs	59	17600	0%
Number of fully used LUT-FF pairs	0	59	0%
Number of bonded IOBs	54	100	54%

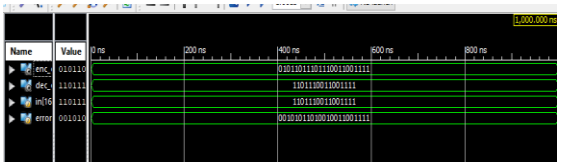
TIME SUMMARY:

```
Data Path: error<20> to U1/D1/OUT_12
                                  Gate     Net
  Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
  ----------------------------------------  ------------
   IBUF:I->O             3   0.000   0.353  error_20_IBUF (error_20_IBUF)
   LUT2:I0->O            2   0.043   0.554  U1/Mxor_err_in_2_xo<0>1 (U1/err_in<20>)
   LUT6:I0->O            1   0.043   0.550  U1/D1/Mxor_S<4>_xo<0>1 (U1/D1/Mxor_S<4>_xo<0>)
   LUT6:I0->O           14   0.043   0.610  U1/D1/Mxor_S<4>_xo<0>4 (U1/D1/S<4>)
   LUT6:I0->O            6   0.043   0.451  U1/D1/S[1]_H21_22[1]_equal_75_o<1>1 (U1/D1/S[1]_H
   LUT6:I3->O            1   0.043   0.000  U1/D1/Mmux_OUT[1]_in[18]_MUX_196_o11 (U1/D1/OUT[1
   LD:D                     -0.035          U1/D1/OUT_12
  ----------------------------------------
  Total                      2.734ns (0.215ns logic, 2.519ns route)
                                     (7.9% logic, 92.1% route)
```

POWER SUMMARY:



EXTENSION RESULTS:



DESIGN SUMMARY:

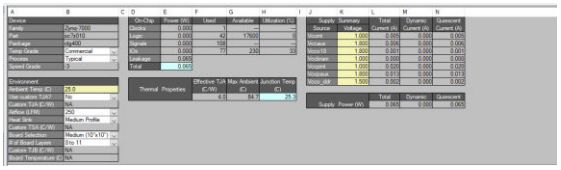
Device Utilization Summary (estimated values)			[-]
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice LUTs	55	17600	0%
Number of fully used LUT-FF pairs	0	55	0%
Number of bonded IOBs	77	100	77%

TIME SUMMARY:

```
Delay:               1.124ns (Levels of Logic = 4)
  Source:            in<13> (PAD)
  Destination:       enc_out<23> (PAD)

  Data Path: in<13> to enc_out<23>
                                Gate     Net
    Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
    ----------------------------------------  ------------
     IBUF:I->O             9   0.000   0.465  in_13_IBUF (in_13_IBUF)
     LUT3:I0->O            1   0.043   0.289  E1/Mxor_C<7>_xo<0>_SW0 (N6)
     LUT6:I5->O            2   0.043   0.283  E1/Mxor_C<7>_xo<0> (enc_out_23_OBUF)
     OBUF:I->O                 0.000          enc_out_23_OBUF (enc_out<23>)
    ----------------------------------------
    Total                      1.124ns (0.086ns logic, 1.038ns route)
                                       (7.6% logic, 92.4% route)
```

POWER SUMMARY:



VI.CONCLUSION AND FUTURE SCOPE

CED technique for OLS code encoders and syndrome computation has been suggested. The suggested methodology used the properties of OLS codes to design a parity prediction system that can be applied effectively and identifies any errors involving a single circuit node. The methodology was tested for various word types, which revealed that the overhead is minimal for big terms. That's also interesting since broad word sizes are used, for example, in caching for which OLS codes have currently been proposed.

The proposed error management scheme involved a considerable delay; nevertheless, its effect on access time can be reduced. This was done by testing in conjunction with the writing of the data in the case of the encoder and in parallel with the majority decision as well as the correction of both the mistake in the situation of the decoder. Throughout the general case, the suggested scheme needed a somewhat greater overhead, since most ECCs did not have the features of OLS codes. This restricted the applicability to OLS codes of both the proposed CED system. Through use of low capital error detecting strategies for encoder and syndrome computing is another justification to recommend the usage of OLS codes throughout high-speed memory and caches.

References

[1] R. C. Baumann, "Radiation-induced soft errors in advanced semiconductor technologies," *IEEE Trans. Device Mater. Reliab.* vol. 5, no. 3, pp. 301–316, Sep. 2005.

[2] M. A. Bajura, Y. Boulghassoul, R. Naseer, S. DasGupta, A. F.Witulski, J. Sondeen, S. D. Stansberry, J. Draper, L. W. Massengill, and J. N. Damoulakis, "Models and algorithmic limits for an ECC-based approach to hardening sub-100-nm SRAMs," *IEEE Trans. Nucl. Sci.*, vol. 54, no. 4, pp. 935–945, Aug. 2007.

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

[4] S. Ghosh and P. D. Lincoln, "Dynamic low-density parity check codes for fault tolerant nano-scale memory," presented at the Foundations Nanosci. (FNANO), Snowbird, Utah, 2007.

[5] S. Ghosh and P. D. Lincoln, "Low-density parity check codes for error correction in nanoscale memory," SRI Computer Science Lab., Menlo Park, CA, Tech. Rep. CSL-0703, 2007.

[6] H. Naeimi and A. DeHon, "Fault secure encoder and decoder for memory applications," in *Proc. IEEE Int. Symp. Defect Fault Toler.VLSI Syst.*, 2007, pp. 409–417.

[7] B. Vasic and S. K. Chilappagari, "An information theoretical framework for analysis and design of nanoscale fault-tolerant memories based on low-density parity-check codes," *IEEE Trans. Circuits Syst.I, Reg. Papers*, vol. 54, no. 11, pp. 2438–2446, Nov. 2007.

[8] H. Naeimi and A. DeHon, "Fault secure encoder and decoder for nanomemory applications," *IEEE Trans. Very Large Scale Integr.(VLSI) Syst.*, vol. 17, no. 4, pp. 473–486, Apr. 2009.

[9] S. Lin and D. J. Costello, *Error Control Coding*, 2nd ed. Englewood Cliffs, NJ: Prentice-Hall, 2004.

[10] S. Liu, P. Reviriego, and J. 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.

[11] H. Tang, J. Xu, S. Lin, and K. A. S. Abdel-Ghaffar, "Codes on finite geometries," *IEEE Trans. Inf. Theory*, vol. 51, no. 2, pp. 572–596, Feb. 2005.