EXTENDING 3-BIT BURST ERROR CORRECTION CODES WITH QUADRUPLE ADJUCENT ERROR CORRECTION

[1]Aade Kailas Ukala,[2]U.Alekya,[3]Lingampally Shivprasad,[4]B.Santhosh
[1234]Assistant Professor
Department Of ECE
Kshatriya College of Engineering

ABSTRACT

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 quadruple adjacent error correction (QAEC). 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 QAEC. 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.

1. 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 an 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

Qiuju Diao, 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 throughou 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"

Low Density Parity Control (LDPC) Codes were groups of linear block codes that have almost power on broad data channel set and are simultaneously feasible to achievable decoders. That Gallger suggested LDPC codes for the first time in 1967 which were scarcely shown when Mackay, Luby and others rediscovered them. Many research aims to define and enhance the efficiency of LDPC codes on multiple networks.

Mark F. Flanagan, John Craddock, Colm P. Less and Stephen J. Redmond suggested "Girth-8 Gallager LDPC codes for a Euclidean algebraic building methodology"

In Low Density Parity-Check (LDPC) practice of having on Euclidean niter geometries EG(m; 2s), a construction technique is suggested. The Gallager codes with Tan ner graphs in eight are seen to be normal. This indicates that the minimum distance of these codes is less than 2 m. The codes can be also used in combination with the Turbo Decode Message Transmission (TDMP) algorithms for the LDPC encoding of an efficient partially parallel decoder. An addition, the findings of the simulation indicate that these codes work very well.

1.3 Organization of the Thesis

Chapter 1 gives Introduction about the motivation, Literature survey and Thesis of organization. Chapter 2 deals about existing architecture Euclidean Geometry Low-Density Parity-Check (EG-LDPC) codesimplementation. Chapter 3 deals about Implementation of Proposed architecture. Chapter 4 VLSI Design and classification, summarizes the Verilog, Design issues, design reusability, importance of HDLs, design flow of verilog and Design styles in VLSI. Chapter 5 gives details about Xilinx ,overview, project navigator, creating project, creating source file and about synthesize & simulation. Chapter 6 deals about experimental results both simulation and synthesis and comparison results. Finally, the conclusion and suggestions for further refinement of the proposed solutions are given in chapter 7.

2. Existing Method

2.1 Introduction To Euclidean Geometry LDPC Codes

With technology scales, memory devices develop larger but error correction codes are becoming more efficient. To this purpose, it has only been suggested to use more sophisticated codes. Ses codes may remedy further mistakes, but typically need complicated decoders. The usage of decodable logic codes with one majority phase was first proposed in device applications to prevent a large decoding difficulty. Additional work was then presented on this subject. With rather simple circuitry, however long decoding times are needed, a phase majority of logic decoding could be implemented. Which raises the access period in a memory, a significant parameter of the device. A single-step majority logic decoding is only feasible to decipher a few groups ofcodes. There involve several low-density Euclidean geometry parity control(EG-LDPC) code used low-density parity control (DS-LDPC) codes.This approach was introduced to speed up the decoding of differential logics establishing codes for low density parity. That's also beneficial because decoding for majority logic could be achieved serially with simple hardware, but takes a great deal of decoding time. Which raises the memory access time with memory applications. That procedure detects whether a word has mistakes during the first major logical decoding executions 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. Throughout this brief, they examine the application of even a related methodology to both the low-density parity (EG-LDPC) class of Euclidean geometry codes which is decodable by a majority step logic. The findings indicate that EG-LDPC codes are also reliable. Extensive simulation results are given such that the likelihood of error detection for various code size and number in errors can be correctly calculated.

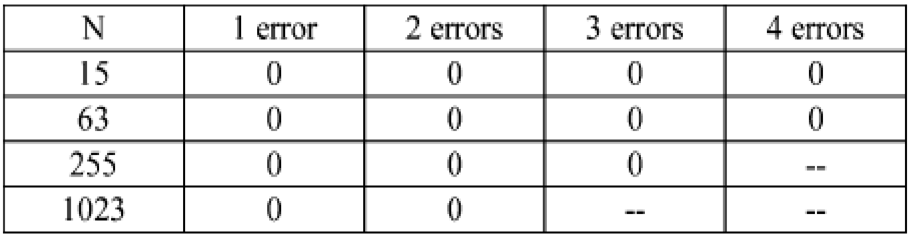
A strategy for speeding the serial execution of the encoding of DS-LDPC codes for majority logic were recently suggested. The goal of the process is to use the first big logic decoding iterations and detect whether the decoded word requires explanation. When no errors occur, decoding may be stopped without the remaining iterations, thereby decreasing decoding period considerably. A majority logic decoding for just a code containing N block length involves N iteration (while applied in series), such that decoding time is allowed as code size increases. Just the three initial iterations are used for the proposed solution to find errors and increase the speed while N is high. For DS-LDPC codes, it was seen that during the first successive sessions all error combinations with up to 5 mistakes can be identified. Furthermore, errors involving more than 5 bits were found with a rather high chance. The risk of error not observed also decreased as the duration of the code block grew. Just a few (or even none) mistakes were not found for a billion error patterns. For some purposes, this might be necessary. Another benefit of this approach is that the circuitrification procedure is often used for the identification of errors and does not require much extra circuits. It was shown, for example, that just about 1 percent for broad word measurements were required to enforce the system.

Implementation

The proposed approach is used in the MLD EG-LDPC class of 1 stage. The outcomes first were described in a theory validated in such a simulation and then in part through a quantitative review throughout the Appendix. The observations are presented. The findings achieved can be summed up in the hypothesis below. "Any errors in only three decoding cyints can be found by the memory reading with a single stage MLD EG-LDPC codes influenced by three to 4 bit flips."

This hypothesis, note, is distinct from the hypothesis for DS-LDPCs in[10], as up to 5 bit errors have often been observed throughout this case. That is because the DS-LDPC and EG-LDPC codes structural variations. The EG-LDPC codes considered were introduced and checked for the validity of the aforementioned hypothesis.

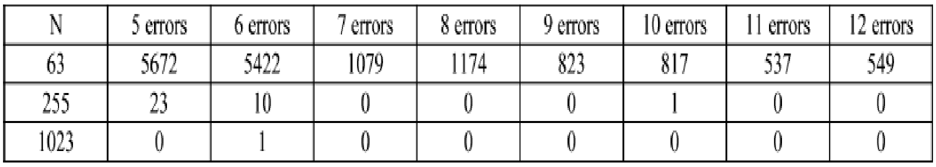
It is easy to create and track all potential error variations of codes tiny terms that are influenced by a small number of bit phrases. With the scale of the code growing and the amount of bit flips can, all potential variations can no longer be thoroughly checked. The simulations are thus carried out in two ways, through a detailed examination among all combinations with errors where possible and by testing random combinations created in other circumstances. Table 2.2 displays the findings for the comprehensive inspections. These findings illustrate the theory for smaller word size codes (15and 63). Two to 3 mistakes were checked extensively for N=255 although only single and double mistake variations were carefully tested with N=1023.



N	1 error	2 errors	3 errors	4 errors
15	0	0	0	0
63	0	0	0	0
255	0	0	0	--
1023	0	0	--	--

Table 2.2 Undetected Errors In Exhaustive Checking

Simulations using unpredictable error rates were used to supplement exhaustive inspection findings for broader codes and multiple mistakes. A billion error variations have been checked in all of the tests. Table 2.3 displays the effects of errors with more than 4 bits as errors with a limit of 4 bits have not been identified. There are a limited amount of error types that do not exist in the first three iterations with errors involving upwards of 4 bits. With sentence size and amount of mistakes this number was reduced.



N	5 errors	6 errors	7 errors	8 errors	9 errors	10 errors	11 errors	12 errors
63	5672	5422	1079	1174	823	817	537	549
255	23	10	0	0	0	1	0	0
1023	0	1	0	0	0	0	0	0

Table 2.3 Undetected Errors With One Billion Random Error Combinations

The reduction throughout the word size may be as continues to follow: the bigger the word size, the smaller the amount with equations for both the MLD scan (see Table I) and, as just a result, errors in the equal equation are less likely to occur. The same is the case for the amount of mistakes: the more errors, the greater the chance that certain least one mistake happens. Lastly, the odds for undetected errors are distinct, then one of the mistakes must arise in a bit not tested for any equivalent amount of errors, as in the above example.

Both errors involving two and five bits are found in the first successive sessions, according to the findings of the simulation. There is a slight chance of failing to identify errors that involve a greater amount of bits. That odds for big word sizes in certain implementations are limited enough to be appropriate. In brief, all errors involving 4 or less bits and virtually all other identifiable mistake influencing more bits are remembered in the first three iterations. The efficiency of this would be marginally lower than with the DS-LDPC file, where 5-bit errors were found. But for EG-LDPC codes, this same majority logically is easier, because the number of equations is a two-powered one or a having to sort network-based approach could be used to decrease the majority vote. Furthermore, EG-LDPC codes has block lengths of about two power, and hence satisfy modern memory device specifications. In certain instances, utilising EG-LDPC and maintaining the clock frequency consistent with current definitions (power of two) could be more practical and using a DS-LDPC code involving an extra document or shorter text form. If word size is really a power of 2, a bit that is not included in the EG-LDPC code is given (see table I). Table I. Each bit may be extended to a parity of all bits which identifies any mistakes that impact a strange amount of bits. In that case any errors involving five or less bits will also be found in the specification utilising that EG-LDPC.

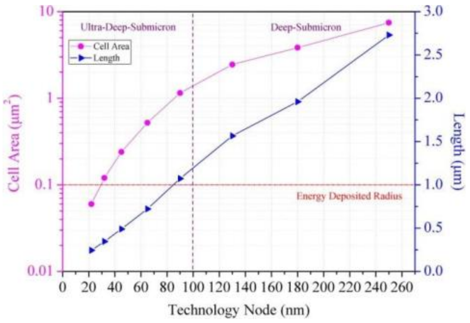
Disadvantages Of Existing System

- ➢ The encoding difficulty of DS-LDPC codes requires constant verification, even though the mistake isn't really present.
- ➢ For N code term, N clock cycles are needed. It is also extremely difficult to decipher.
- ➢ The encoding difficulty of EG-LDPC code is lower and the mistake is found during the first three cycles and that only a bit could be resolved in one loop.
- ➢ The delay rises and the productivity decreases.

3. IMPLEMENTATION OF PROPOSED ARCHITECTURE

3.1 Introduction

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.3.1. Their size of both the memory cells becomes smaller.



Technology Node(nm)

Fig. 3.1 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. 3.2. 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..

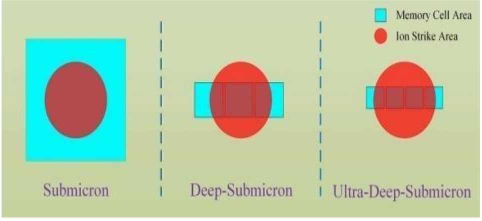


Fig 3.2. Schematic description of memory cells included in the radiation effect with variation of the technology node

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 3.2 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 quadruple adjacent error correction (QAEC) as well.

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

1) Error space satisfiability 2) Unique syndrome satisfiability.

QAEC codes 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

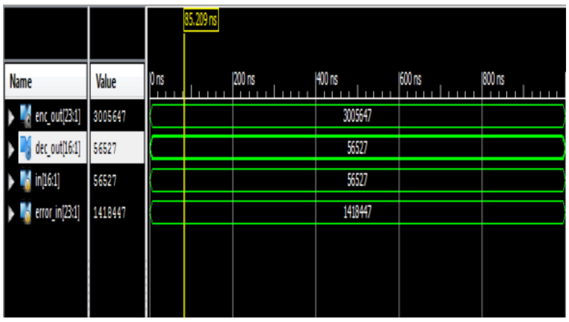
2) Minimization of the parity control matrix number in the heaviest lines.

3.2 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.

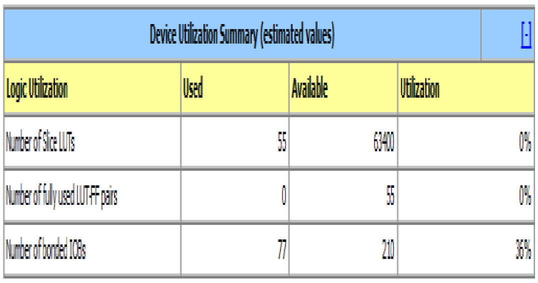
4.RESULTS

4.1 Simulation Result



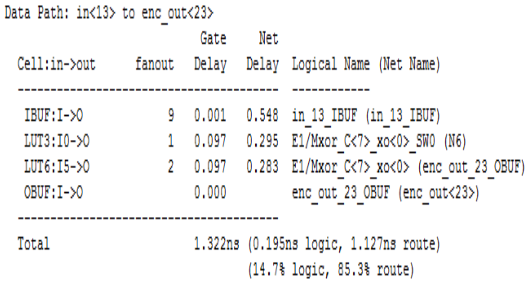
waveform

RTL schematic



Device Utilization Summary (estimated values)			[-]
Logic Utilization	Used	Available	Utilization
Number of Slice LUTs	55	63400	0%
Number of fully used LUT-FF pairs	0	55	0%
Number of bonded IOBs	77	210	36%

DESIGN SUMMARY



```
Data Path: in<13> to enc_out<23>
                              Gate     Net
  Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
  ----------------------------------------  ------------
   IBUF:I->O             9   0.001   0.548  in_13_IBUF (in_13_IBUF)
   LUT3:I0->O            1   0.097   0.295  E1/Mxor_C<7>_xo<0>_SW0 (N6)
   LUT6:I5->O            2   0.097   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                      1.322ns (0.195ns logic, 1.127ns route)
                                     (14.7% logic, 85.3% route)
```

Time summary

5.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.