ERROR DETECTION IN MAJORITY LOGIC DECODING OF EUCLIDEAN GEOMETRY LOW DENSITY PARITY CHECK (EG-LDPC) CODES

D. Indrasena Reddy[1], D. Suneel[2]

[1] Pursuing M.tech from Nalanda Institute of Engineering and Technology(NIET), Siddharth Nagar, Kantepudi village, Sattenepalli Mandal Guntur Dist.,A.P. INDIA

[2] Working as Asst.professor (ECE) from Nalanda Institute of Engineering and Technology(NIET), Siddharth Nagar, Kantepudi village, Sattenepalli Mandal Guntur Dist.,A.P. INDIA

ABSTRACT

The fault secure memory system consist encoder, decoder, detector and parallel pipelined corrector. In this the data should be transferred through the encoder and then it will be decoded, the decoder consist the detector and corrector if the data have any errors that should be detected by detector and then that data will be corrected by the parallel pipelined corrector. So it has large decoding time to correct the data by using parallel pipeline corrector. To reduce this one here we are using majority logic decoding with EGLDPC (Euclidean Geometry Low Density Parity Check) codes. In this method the system was check whether the data has any errors in the first iterations of the majority logic decoding if it does not has any errors the decoding part will be ended without completing the rest of iterations. With this most of the words have no errors in the memory and the average time is also reduced. The proposed design is implemented with the Verilog hdl and simulation results are represented from the XILINX ISE 13.2 simulator.

Keywords: Error correction codes, detector, (EG-LDPC) codes, majority logic decoding, memory, corrector.

I. INTRODUCTION

Error correction codes are normally used to secure memories from so these are called as soft Errors, which change the legical estimation of memory cells without harming the circuit. As innovation scales, memory gadgets get to be bigger and all the more capable error adjustment codes are required. To this end, the utilization of more propelled codes has been as of late proposed. These codes can remedy a bigger number of Errors, yet for the most part require complex decoders. To evade a high interpreting multifaceted nature, the utilization of one stage larger part rationale decidable codes was initially proposed in for memory applications. Further work on this point was then exhibited.One stage majority logic interpreting can be actualized serially with exceptionally straightforward hardware, be that as it may, requires long interpreting times. In a memory, this would increment the entrance time which is a vital framework parameter. Just a few classes of codes can be decoded utilizing one stage lion's share rationale interpreting. Among those are some Euclidean geometry low thickness equality check (EG-LDPC) codes which were utilized as a part, and contrast set low thickness equality check (DS-LDPC) codes.

A strategy was as of late proposed to quicken a serial usage of majority logic disentangling of DS-LDPC codes. The thought behind the strategy is to utilize the first cycles of majority logic decoding to distinguish if the word being decoded contains Errors. In the event that there are no Errors, at that point interpreting can be ceased without finishing the remaining emphases, thusly significantly lessening the decoding time.

For a code with square length N, majority logic decoding(when actualized serially) requiresN cycles, so that as the code size develops, so does the decoding time. In the proposed approach, just the initial three cycles are utilized to recognize Errors, in this way accomplishing an expansive velocity increment at the point when N is substantial. It was demonstrated that for DS-LDPC codes, all error blends of up to five Errors can be identified in the firstthree cycles. Additionally, Errors influencing more than five bits were distinguished with a likelihood near one. The likelihood of undetected Errors was likewise found to diminish as the code square length expanded. For a billion error designs just a couple of Errors (or once in a while none) were undetected. This may be adequate for a few applications.

Another point of interest of the proposed strategy is that it requires exceptionally minimal extra hardware as the deciphering hardware is likewise utilized for error recognition. For instance, it was appeared that the extra territory required to actualize the plan was just around 1% for vast word sizes.

The strategy depends on the properties of DS-LDPC codes and in this way it is not straightforwardly pertinent to other code classes. In the accompanying, a comparable methodology for EG-LDPC codes is displayed.

Nano technology gives littler, speedier, and lower vitality gadgets which permit all the more intense and minimized hardware; on the other hand, these advantages accompany an expense—the nanoscale gadgets may be less dependable. Warm and shot-commotion estimations alone propose that the transient shortcoming rate of an individual nanoscale gadget (e.g., transistor or nanowire) may be requests of size higher than today's gadgets. As a result, we can anticipate that combinational rationale will be helpless to transient flaws notwithstanding capacity cells and correspondence channels. In this manner, the worldview of ensuring just memory cells and accepting the encompassing hardware (i.e., encoder what's more, decoder) will never present mistakes is no more substantial.

In this paper, we present a deficiency tolerant nano scale memory construction modelling which endures transient issues both in the capacity unit and in the supporting rationale (i.e., encoder, decoder (corrector), what's more, finder hardware).Especially, we recognize a class of mistake redressing codes (ECCs) that ensures the presence of a straightforward issue tolerant finder outline. This class fulfils another, confined definition for ECCs which ensures that the ECC codeword has a suitable repetition structure such that it can identify numerous mistakes happening in both the put away codeword in memory and the encompassing circuitries. We call this kind of mistake remedying codes, shortcoming secure finder proficient ECCs (FSD-ECC). The equality check Matrix of a FSD-ECC has a specific structure that the decoder circuit, produced from the equality check

Matrix, is Fault-Secure. The ECCs we distinguish in this class are near ideal in rate and separation, proposing we can accomplish this property without giving up customary ECC measurements.

We utilize the shortcoming secure discovery unit to outline a deficiency tolerant encoder and corrector by observing their yields. On the off chance that a locator identifies a mistake in both of these units, that unit must rehash the operation to create the right yield vector. Utilizing this retry strategy, we can remedy potential transient mistakes in the encoder and corrector yields and give a completely blame tolerant memory system.

II. PROJECT DESCRIPTION

In this area, we give the configuration structure of the encoder, corrector, and identifier units of our proposed flaw tolerant memory framework. We additionally display the execution of these units on a sub-lithographic, nanowire-based substrate. Some time as of late going into the design structure purposes of interest we start with a brief framework of the sub-lithographic memory auxiliary building model.

In this paper, one particular kind of low thickness equality check codes, to be specific Euclidean Geometry-LDPC codes are utilized because of their shortcoming secure locator Capacity, higher relentless quality and lower expand overhead. The EG-LDPC codes which are one step larger part rationale decoder.

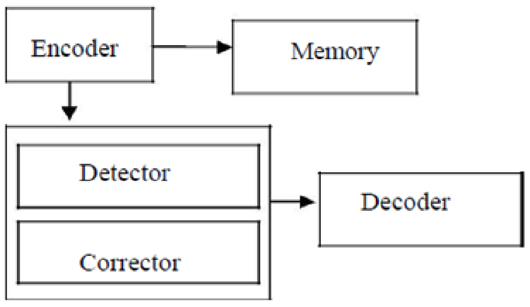


Fig Detector and corrector in EG geometry

Particular blunder revelation frameworks are utilized to maintain a strategic distance from the delicate blunder. One of the timetables is larger part rationale decoder which used to get and right the blunder in crucial way. This framework utilizes the first emphasis of larger part rationale interpreting to recognize the blunder present in the translation. In the event that there are no blunders, then the deciphering methodology may be halted without finishing the remaining emphases. The key explanation for utilizing Majority Logic Decoding (MLD) is that it is not hard to finish and has a low versatile quality. As portrayed in previous, Majority logic decoder is a crucial and plausible decoder arranged for inspecting distinctive piece flips depending upon the measure of correspondence check total examinations. It contains four portions: 1) a cyclic development enrols; 2) a XOR system; 3) a bigger part passage; 4) an EXOR gateway for mistake amendment, as sketched out in figure 2.

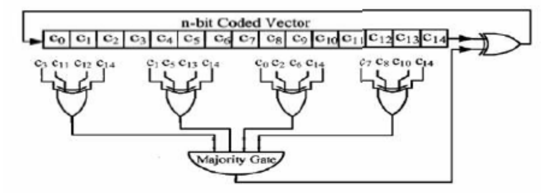


Fig 2: Decoder of One step Majority Logic for (15, 7) G-LDPC Codes

In one stage dominant part rationale deciphering, from the get go the code word is stacked into the cyclic movement register. By then the check correlations are enrolled. The accompanying results are then sent to the larger part door for assessing its exactness. In the event that the measure of 1"s got in is more critical than the measure of 0"s which hint that the present piece under disentangling isn't right, and a sign to alter it would be actuated. Generally the bit under deciphering is right and no additional operations would be required on it. In next, the substance of the registers are pivoted and the above framework is repeated until code word bits have been prepared. At last, the equality check entire would be zero in the event that the code word has been flawlessly decoded. In this procedure, every piece may be revaluated basically once.

Henceforth, the interpreting hardware is crucial, yet it obliges a long deciphering time if the code word is significant. Thusly, by one greater part rationale deciphering, the code is readied for investigating any blunder plot with two or less mistakes. For instance, for a code outpouring of 15-bits, the unravelling would take 15 cycles, which would be intemperate for all things considered applications. The lion's share rationale decoders itself go about as an issue finder.

MLD disentangling all code word bits, the MLD procedure stops transitionally in the third cycle, which can fit to discover up to five piece flips in three disentangling cycles. So the measure of unravelling cycles could be lessened to get updated execution. The schematic of lion's share rationale decoder/finder is spoken to in figure3.

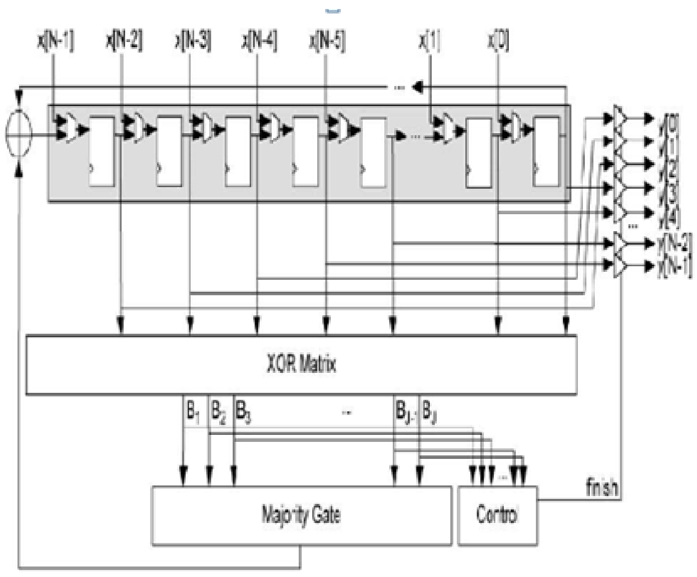


Fig 3: Schematic of Majority Logic Decoder/Detector (MLD)

At the beginning the code word is secured into the cyclic movement register and it traveled through all the taps. The broadly engaging qualities in each one tap is given to the XOR framework to perform the check entirety examinations. The subsequent wholes are then sent to the dominant part section for assessing its rightness. In the event that the measure of 1‟ s got is more unmistakable than the measure of 0‟ s which would recommend that the current bit under interpreting isn't right, so it push forward the interpreting system. Generally, the bit under interpreting would be right and no additional operations would be required on it. Interpreting technique including the operation of the substance of the registers is turned and the above structure is repeated and it stops transitionally in the third emphasis. If to start with three cycles of the interpreting process, the evaluation of the XOR framework for all is "0," the code word is resolved to be sans botch and sent especially to the yield. In case the error contains in any of the three cycles at any rate a "1," it would push forward with the entirety isentangling system to execute the mistakes.

At long last, the equality check aggregates ought to be zero on the off chance that the code word has been precisely decoded. Considering everything the MLD framework is utilized to perceive the five piece mistakes and right four piece blunders viably. In the event that the code word contains more than five piece mistake, it makes the yield at any rate it didn't appear the mistakes showed in the information. This sort of blunder is known as the noiseless data mistake. Downside of this technique is did not distinguishing the quiet data mistake and it gobbling up the area of the lion's share part entryway. The schematic for this memory structure is exhibited in figure 5. It is truly obviously proportionate to the one appeared in fig. 1; 1moreover the control unit was consolidated the MLD module to deal with the translating system (to identify the mistake). The two fundamental steps included in computing in identified code word are as per the following:

1) Create equality check wholes by enlisting the inner delayed consequence of the got vector and the right lines of equality check network.

2) The register wholes are made do with a larger part entryway. The yield of the greater part door adjusts the bit by bothering the quality if the yield of greater part entryway is "1".

The serial greater part corrector takes cycles to change an off kilter codeword. In the event that the defect rate is low, the corrector piece is utilized on occasion; subsequent to the customary case is slip by free code words, the stillness of the corrector won't genuine affect the run of the mill memory perused dormancy.

The one stage greater part rationale identifier may be utilized to change all the n bits of the got codeword of a cyclic code. To recognize each one code bit, the got encoded vector is cyclic moved and supported into the XOR doors. The data bits are reinforced into the encoder to encode the data vector. The inadequacy secure finder of the encoder checks the credibility of the encoded vector. If that the finder distinguishes any bumble the encoding operation must be fixed up to convey the privilege codeword. The codeword is then secured in the memory. In the midst of memory access operation, the set away code words will be gotten to from the memory unit.

A corrector unit is proposed to amend potential blunders in the recovered codeword's. MLD is engaged around distinctive equality check mathematical statements which are orthogonal to one another, so that at every cycle each codeword bit offers in one helps all mathematical statements. For this reasons the bigger part possible result of these equality check mathematical statements pick the rightness of the right piece under unraveling. Non specific schematic of a memory system is depicted for the usage of a ML decoder.

In this technique at the beginning, the information words are encoded and after that set away in the memory the taking after aggregates are then sent to the dominant part rationale door for reviewing its rightness. On the off chance that the measure of 1‟ s got in is more basic than the measure of 0‟ s, which would suggest that the present piece under deciphering isn't right and a sign to review it would be authorized. In general, the bit under disentangling would be right and no additional operations would be required on it and it manufactures decoder. Notwithstanding they oblige a far coming to disentangling time that effects memory execution
To enhance the decoder execution, decision plots may be utilized. One probability is to join a weakness finder by figuring the issue, so just faulty code words are decoded. Since the lion's share rationale of the code word will be sans mistake, no further revision will be obliged subsequently execution won't be influenced. Despite the way that the use of a SFD decreases the commonplace of the deciphering strategy.

2.1 Fault Secure Detector

The center of the indicator operation is to create the disorder vector, which is fundamentally executing the accompanying vector-framework augmentation on the got encoded vector what's more, equality check grid Along these lines every piece of the disorder vector is the result of with one column of the equality check grid. This item is a direct twofold whole over digits of where the relating digit in the framework line is 1

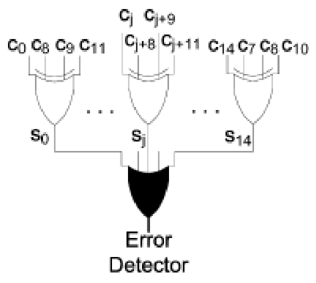


Fig. 3. Fault-secure detector for (15, 7, 5) EG-LDPC code. All the gates except the last OR gate are implemeneted with fault-prone nanoscale circuitry. The last OR gate is implemented with more reliable lithography technique.

src

This paired whole is executed with an XOR door. Fig. 3 demonstrates the identifier circuit for the (15, 7, 5) EG-LDPC code. Since the column weight of the equality check grid is , to produce one digit of the disorder vector we require a - info XOR entryway, or 2-information XOR doors. For the entirety locator, it takes 2-information XOR entryways. Table II delineates this amount for a percentage of the littler EG-LDPC codes. Note that executing every disorder piece with a different XOR entryway fulfills the suspicion of Theorem I of no rationale partaking in identifier circuit usage. A mistake is distinguished if any of the disorder bits has a nonzero esteem. The last blunder location sign is actualized by an OR capacity of all the disorder bits. The yield of this -information OR entryway is the blunder finder sign (see Fig. 3). So as to maintain a strategic distance from a solitary purpose of disappointment, we must execute the OR entryway with a dependable substrate (e.g., in a framework with sub-lithographic nanowire substrate, the OR door is actualized with dependable lithographic innovation—i.e., lithographic-scaled wire-OR). This - information wire-OR is much littler than actualizing the whole 2-info XORs at the lithographic scale. The range of every identifier is processed utilizing the model of NanoPLA furthermore, NanoMemory structure representing all the supporting lithographic wires and reported in Table III.

2.2 Encoder

A - bit codeword , which encodes a - bit data vector is created by duplicating the - bit data vector with a bit generator network ; i.e., . EG-LDPC codes are not efficient and the data bits must be decoded from the encoded vector, which is not attractive for our shortcoming tolerant methodology because of the further complexity what's more, postpone that it adds to the operation. Be that as it may, these codes are cyclic codes. We utilized the strategy introduced as a part of and to change over the cyclic generator frameworks to deliberate generator lattices for all the EG-LDPC codes under thought.

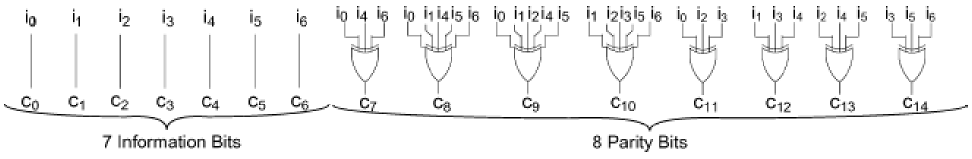

i_0 i_1 i_2 i_3 i_4 i_5 i_6
i_0 i_4 i_6
i_0 i_1 i_4 i_5 i_6
i_0 i_1 i_2 i_4 i_5
i_1 i_2 i_3 i_5 i_6
i_0 i_2 i_3
i_1 i_3 i_4
i_2 i_4 i_5
i_3 i_5 i_6
C_0 C_1 C_2 C_3 C_4 C_5 C_6 C_7 C_8 C_9 C_{10} C_{11} C_{12} C_{13} C_{14}
7 Information Bits
8 Parity Bits

Fig. 6. Structure of an encoder circuit for the (15, 7, 5) EG-LDPC code; i_0 to i_6 are 7-bit information vector. Each of the XOR gates generate one parity bit of the encoded vector. The codeword consists of seven information bits followed by eight parity bits.

Above figure demonstrates the precise generator framework to create (15, 7, 5) EG-LDPC code. The encoded vector comprises of data bits took after by equality bits, where every equality bit is just an inward result of data vector and a segment of , from . Fig. 6 demonstrates the encoder circuit to figure the equality bits of the (15, 7, 5) EG-LDPC code. In this figure is the data vector and will be replicated to bits of the encoded vector, , and whatever remains of encoded vector, the equality bits, are direct aggregates (XOR) of the data bits. On the off chance that the building square is two-information entryways then the encoder hardware takes 22 two-info XOR doors. Table II demonstrates the zone of the encoder circuits for each EG-LDPC codes under thought in light of their generator networks. Once the XOR capacities are known, the encoder structure is fundamentally the same to the indicator structure appeared in Fig. 4, aside from it comprises of XOR entryways of fluctuating quantities of inputs. Each nanowire-based XOR entryway has structure like the XOR tree appeared.

2.3 Corrector

One-stage greater part rationale remedy is a quick and moderately minimal blunder adjusting procedure. There is a constrained class of ECCs that are one-stage greater part correctable which incorporate sort I two-dimensional EG-LDPC. In this segment, we exhibit a brief survey of this adjusting procedure. At that point we demonstrate the one-stage larger part rationale corrector for EG-LDPC codes. 1) One-Step Majority-Logic Corrector: One-stage majoritylogic amendment is the methodology that recognizes the right esteem of an every piece in the codeword specifically from the got codeword; this is as opposed to the general message-passing errorcorrection procedure which may request numerous emphases of blunder finding and trial rectification. Keeping away from cycle makes the remedy dormancy both little and deterministic. This strategy can be actualized serially to give a minimized execution on the other hand in parallel to minimize remedy inactivity. This system comprises of two sections: 1) creating a particular set of straight totals of the got vector bits and 2) discovering the greater part estimation of the figured direct totals. The greater part esteem demonstrates the accuracy of the code-bit under thought; if the greater part esteem is 1, the bit is reversed, else it is kept unaltered. The hypothesis behind the one-stage greater part corrector and the evidence that EG-LDPC codes have this property are accessible. Here we outline the structure of such correctors for EG-LDPC code.

Limited geometries have been utilized to infer numerous blunder redressing codes. One case are EG-LDPC codes which are based on the structure of Euclidean geometries over a Galois field. Among EG-LDPC codes there is a subclass of codes that is one stage larger part rationale decodable (MLD). Codes in this subclass are additionally cyclic. The parameters for some of these codes are given in Table I, where the piece size, the quantity of data bits, the quantity of MLD check mathematical statements and the quantity of mistakes that the code can revise utilizing one stage MLD.

One stage MLD can be actualized serially utilizing the plan as a part of Fig. which compares to the decoder for the EG-LDPC code with. To begin with the information square is stacked into the registers. At that point the check mathematical statements are processed and if a larger part of them has an estimation of one, the last piece is transformed. At that point all bits are consistently moved. This arrangement of operations constitutes a solitary cycle: after emphases, the bits are similarly situated in which they were stacked. All the while, every piece may be adjusted just once. As can be seen, the disentangling hardware is basic, however it requires a long unravelling time if is extensive.

The check mathematical statements must have the accompanying properties (see for more subtle elements).

Table ii

Undetected Errors in Exhaustive Checking

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	--	--

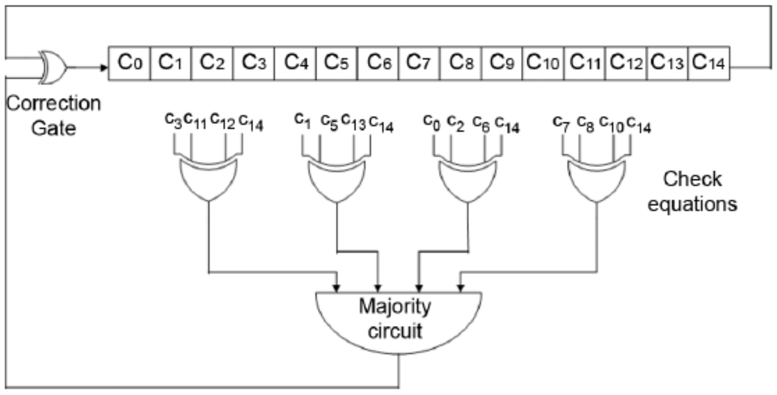
1) All mathematical statements incorporate the variable whose quality is put away in the last register (the one stamped as

2) Whatever is left of the registers are incorporated into at most one of the check
Comparisons.

In the event that blunders can be identified in the initial couple of emphases of MLD, then at whatever point no blunders are distinguished in those cycles, the disentangling can be halted without finishing whatever is left of the cycles. In the first cycle, blunders will be distinguished when no less than one of the check mathematical statements is influenced by an odd number of bits in blunder. In the second emphasis, as bits are consistently moved by one position, blunders will influence other comparisons such that a few blunders undetected in the first cycle will be identified. As cycles propel, every single perceivable mistake will in the long run be distinguished.

TABLE I
ONE STEP MLD EG-LDPC CODES

N	K	J	t_{ML}
15	7	4	2
63	37	8	4
255	175	16	8
1023	781	32	16



In it was demonstrated that for DS-LDPC codes most blunders can be distinguished in the initial three cycles of MLD. Taking into account re-enactment results what's more, on a hypothetical confirmation for the instance of two blunders, the accompanying theory was made.

"Given a word read from a memory ensured with DS-LDPC codes, what's more, influenced by up to five piece flips, all mistakes can be recognized in just three unravelling cycles". At that point the proposed procedure was executed in VHDL and blended, demonstrating that for codes with vast square sizes the overhead is low. This is on account of the current lion's share rationale interpreting hardware is reused to perform mistake location and just some additional control rationale is needed.

2.4 Majority Circuit Implementation

Here we introduce a minimal execution for the larger part door utilizing Sorting Networks. The larger part entryway has application in numerous other blunder rectifying codes, and this conservative usage can enhance numerous different applications. A larger part capacity of double digits is just the middle of the digits (where we characterize the middle of a considerably number of digits as the littlest digit). To locate the middle of the inputs, we do the accompanying:

1) separate the inputs into two parts with size ;

2) sort each of the parts;

3) the middle is 1 if for the th component of

one half and the th component of the other half are both 1.

We utilize double Sorting Networks to do the sort operation of the second step effectively. A - data sorting system is the structure that sorts an arrangement of bits, utilizing a 2-bit sorter building pieces. demonstrates a 4-information sorting system. Each of the vertical lines speaks to one comparator which looks at two bits and allots the bigger one to the top yield and the littler one to the base The four-information sorting system, has five

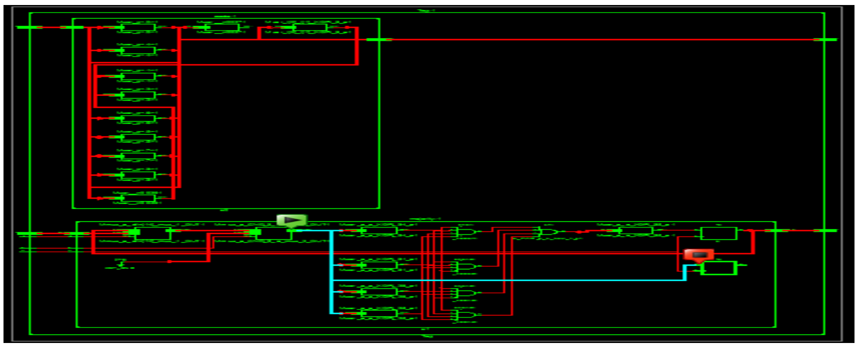
ijates
ISSN 2348 - 7550

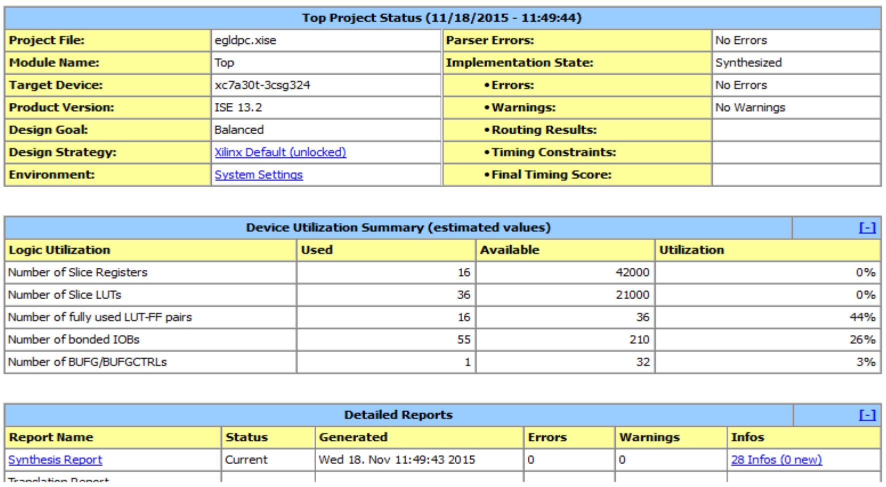
comparator pieces, where every square comprises of two-information entryways; general the four-data sorting system comprises of ten two-information entryways altogether.

To check the condition in the third step, we utilize two-information What's more, entryways took after by a -information OR door. demonstrates the circuit executing the above strategy to locate the middle estimation of 8 bits. It has two - information (four-data) sorting systems taken after by combinational hardware, comprising of four two-info AND entryways and a four-information OR door, which can be executed with three two-data OR doors. In this manner altogether an eight-info larger part door actualized with sorting systems take 27 two-information doors; interestingly, the two-level usage of this greater part door takes five-info AND entryways what's more, one 56-info

III. SYNTHESIS AND SIMULATION RESULTS

This project was implemented with Verilog HDL using Xilinx ISE simulator, and synthesis and simulation results are shown in the below.

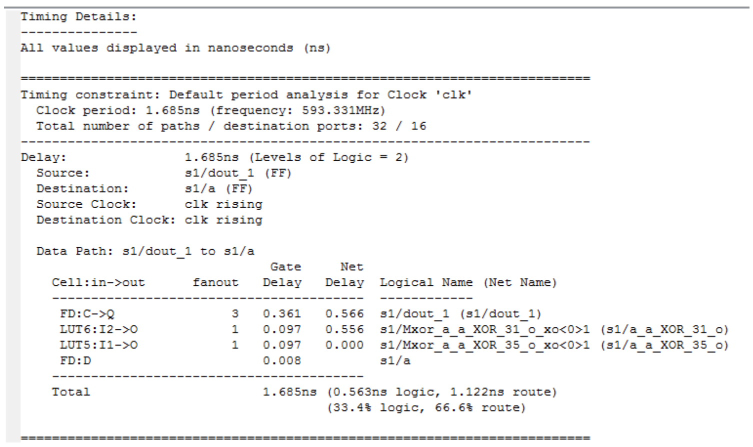




Top Project Status (11/18/2015 - 11:49:44)			
Project File:	egldpc.xise	**Parser Errors:**	No Errors
Module Name:	Top	**Implementation State:**	Synthesized
Target Device:	xc7a30t-3csg324	• **Errors:**	No Errors
Product Version:	ISE 13.2	• **Warnings:**	No Warnings
Design Goal:	Balanced	• **Routing Results:**	
Design Strategy:	Xilinx Default (unlocked)	• **Timing Constraints:**	
Environment:	System Settings	• **Final Timing Score:**	

Device Utilization Summary (estimated values) [-]			
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice Registers	16	42000	0%
Number of Slice LUTs	36	21000	0%
Number of fully used LUT-FF pairs	16	36	44%
Number of bonded IOBs	55	210	26%
Number of BUFG/BUFGCTRLs	1	32	3%

Detailed Reports [-]					
Report Name	**Status**	**Generated**	**Errors**	**Warnings**	**Infos**
Synthesis Report	Current	Wed 18. Nov 11:49:43 2015	0	0	28 Infos (0 new)
Translation Report					



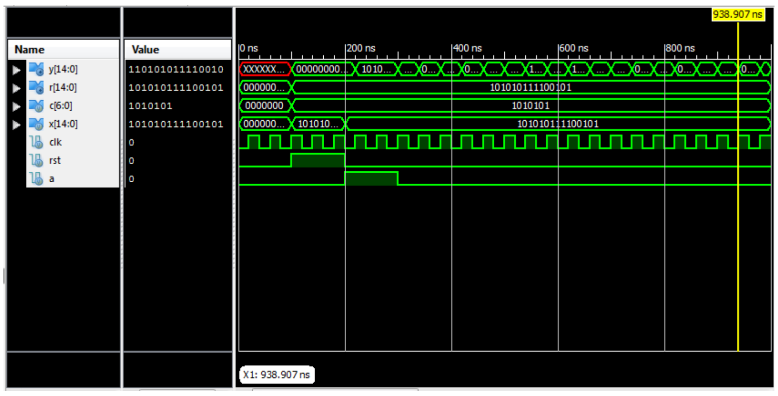
```
Timing Details:
---------------
All values displayed in nanoseconds (ns)

=========================================================================
Timing constraint: Default period analysis for Clock 'clk'
  Clock period: 1.685ns (frequency: 593.331MHz)
  Total number of paths / destination ports: 32 / 16
-------------------------------------------------------------------------
Delay:               1.685ns (Levels of Logic = 2)
  Source:            s1/dout_1 (FF)
  Destination:       s1/a (FF)
  Source Clock:      clk rising
  Destination Clock: clk rising

  Data Path: s1/dout_1 to s1/a
                                Gate     Net
    Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
    ----------------------------------------  ------------
     FD:C->Q               3   0.361   0.566  s1/dout_1 (s1/dout_1)
     LUT6:I2->O            1   0.097   0.556  s1/Mxor_a_a_XOR_31_o_xo<0>1 (s1/a_a_XOR_31_o)
     LUT5:I1->O            1   0.097   0.000  s1/Mxor_a_a_XOR_35_o_xo<0>1 (s1/a_a_XOR_35_o)
     FD:D                      0.008          s1/a
    ----------------------------------------
    Total                      1.685ns (0.563ns logic, 1.122ns route)
                                       (33.4% logic, 66.6% route)

=========================================================================
```

Simulation result:



```
Name        Value
y[14:0]     110101011110010
r[14:0]     101010111100101
c[6:0]      1010101
x[14:0]     101010111100101
clk         0
rst         0
a           0
```

VII. CONCLUSION

In this paper we have designed serial one step majority logic decoder for EGLDPC codes to detect the errors in data. The proposed design is mostly decrease the decoding time in the decoding process for error detection. The proposed simulation results can shows that the combination detection of errors for the iterations of EG LDPC codes. This design also used to detect the all errors effecting five or fewer bits.

REFERENCES

[1] R. C. Baumann, "Radiation-induced soft errors in advanced semiconductortechnologies," 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 approachto 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 reliabilityin sub-100 nm technologies,"Proc. IEEE ICECS, pp. 586–589,2008.

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

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

[6] H. Naeimi and A. DeHon, "Fault secure encoder and decoder formemory 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 memoriesbased 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 fornanomemory 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. EnglewoodCliffs, NJ: Prentice-Hall, 2004.

[10] S. Liu, P. Reviriego, and J. Maestro, "Efficient majority logic faultdetection 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 finitegeometries," IEEE Trans. Inf. Theory, vol. 51, no. 2, pp. 572–596, Feb.2005.

AUTHOR DETAILS

D.INRASENA REDDY, Pursuing M.techin Nalanda Institute of Engineering and Technology, He completed his B.tech Electronics and Communication Engineering. His research of interest includes communication systems, Digital communications, Satellite communication etc.

D.SUNEEL , Working as assistant professors in Nalanda Institute of Engineering and Technology, He completed his Post Graduation, He has four years of teaching experience. Her research of interest includes communication systems, Digital communications, Satellite communication etc.