OPTIMAL SELF CORRECTING FAULT FREE ERROR CODING TECHNIQUE IN MEMORY OPERATION

Harikishore .Kakarla[1], Madhavi Latha .M[2] and Habibulla Khan[3]

[1]Asst.Professor, Dept.of ECE, KL University, Vijayawada, AP, India.
kakarla.harikishore@gmail.com
[2] Professor and Head, Dept.of ECE, JNTUH, Hyderabad, AP, India.
[3] Professor and Head, Dept.of ECE, KL University, Vijayawada, AP, India.

ABSTRACT

As with the increase in data density the probability of error happening in data transfer has increased the need for error free coding technique is in greater demand. In this paper an optimal approach for the processing of self correcting logic in faulty condition during coding and decoding operation is developed. The process of digital designing and its evaluation for realization is presented. The implementation of proposed design on Xilinx-FPGA device is presented.

KEYWORDS

Error free coding, Memory operation, self correcting logic, forward error correction

1. INTRODUCTION

Technology scaling, a reduction in operating voltages, and the increase in cache size and circuit complexity have been key enablers to achieving the performance improvement expectation dictated by Moore's Law. The resulting reduction in the node charge of circuit latches and cache cells has resulted in an ever-increasing soft error rate (SER) estimation for logic components. As technology has scaled, clear trends have emerged from one generation to the next that allow for SER estimates of future technologies to be made. While this approach worked well in previous technology generations, device scaling, and the resulting capacitance and power supply reductions are moving technologies into new regimes of SER sensitivity. Nanotechnology provides smaller, faster, and lower energy devices which allow more powerful and compact circuitry; however, these benefits come with a cost-the nanoscale devices may be less reliable. Thermal and shot-noise estimations alone suggest that the transient fault rate of an individual nanoscale device may be orders of magnitude higher than today's devices. As a result, we can expect combinational logic to be susceptible to transient faults in addition to storage cells and communication channels. Therefore, the paradigm of protecting only memory cells and assuming the surrounding circuitries (i.e., encoder and decoder) will never introduce errors is no longer valid.Traditionally, memory cells were the only circuitry susceptible to transient faults, and all the supporting circuitries around the memory (i.e., encoders and decoders) were assumed to be fault-free. As a result most of prior work designs for fault-tolerant memory systems focused on protecting only the memory cells. However, as we continue scaling down feature sizes or use sublithographic devices, the surrounding circuitries of the memory system will also be susceptible to permanent defects and transient faults. One approach to avoid the reliability problem in the surrounding circuitries is to implement these units with more reliable devices. However, from an area, performance, and power consumption point of view it is beneficial to implement encoders and decoders with scaled feature size or nanotechnology devices. Consequently, it is important to remove the reliability barrier for these logic circuits so they can

DOI : 10.5121/ijcsit.2011.3305

be implemented with scaled feature size or nanotechnology devices. Almost all of the proposed fault tolerant encoders and decoders so far, use the conventional fault tolerant scheme e.g., logic replication or concurrent parity prediction to protect the encoder and corrector circuitry. That is, they add additional logic to check the correctness of the circuit calculation. In contrast, the technique introduced in this work exploits the existing structure of the ECC to guarantee the fault-secure property of the detector unit without adding redundant computations.

2. ERROR CODING

Various error coding techniques were proposed in past. Among these coding techniques the forward error coding technique is one of the optimal coding approaches. In forward error coding techniques Reed-Solomon (RS) coding technique is one of the optimal approaches. RS Code is from the family of cyclic check block codes. The instinct behind RS Code is using polynomials over a certain Galois Field (GF) to encode/decode the information symbols. RS code could be non-binary code but we are most interesting about its binary form when it is defined over GF (2^m). In general, RS code over GF (2^m) with length k'= 2^m-1 can reconstruct k=2^m-1-2t original data symbols with up to t errors by using matrix inversions and multiplications on a binary generator matrix of size m (2^m-1-2t)* m (2^m-1). RS Code is widely used in many products such as compact CD-ROM due to its good capability with burst errors. However, the decoding time of RS code increases as a proportion of k^2 so RS code becomes very inefficient as the length of message grows bigger. Thus, RS Code is considered as a kind of small block codes and often used when information has a small size. The modified approach of these coding techniques is the LDPC coding approach. Low-Density Parity-Check (LDPC) codes are simple parity check codes defined by parity-check matrices that are restricted to have a small number of 1's in each row and column as compared to the lengths of the column and row. The term, low-density implies sparse parity-check matrices. An LDPC code defined by an r£n sparse parity-check matrix, H can be also graphically represented by a bipartite graph depicted in Figure.1. On the graph, each column and row of the sparse parity-check matrix are denoted as a variable (or left) and check (or right) nodes and represent a received coded symbol and a parity check, respectively. That is, the n^{th} variable (check) node represents the n^{th} column (row) of H.

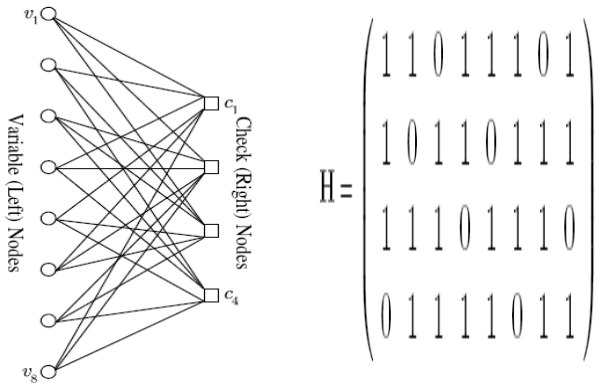

Figure 1. A bipartite graph and parity-check matrix of a (3, 6) regular code.

In a bipartite graph, a connection between a variable node and a check node is called an edge which corresponds to a non-zero term at the position indexed by the column and row. Each node has at least one edge, and the number of edges incident with a node is called the degree of the node. Thus, if a variable (check) node has d edges, the corresponding column (row) has d non-zero terms in H. A cycle is a walk through edges from a node to itself without visiting a node and edge again except the end nodes. The approach of this coding technique based on graph technique is proposed for the realization of a self correcting logic in Data I/O operation with memory interface.

3. SELF CORRECTING DATA INTERFACE

For the development of the suggested approach a self correcting data interface unit is developed. The approach for the realization of the suggested approach is shown in Figure 2. For the processing the information bits are fed into the encoder to encode the information vector, and the fault secure detector of the encoder verifies the validity of the encoded vector. If the detector detects any error, the encoding operation must be redone to generate the correct codeword. The codeword is then stored in the memory.

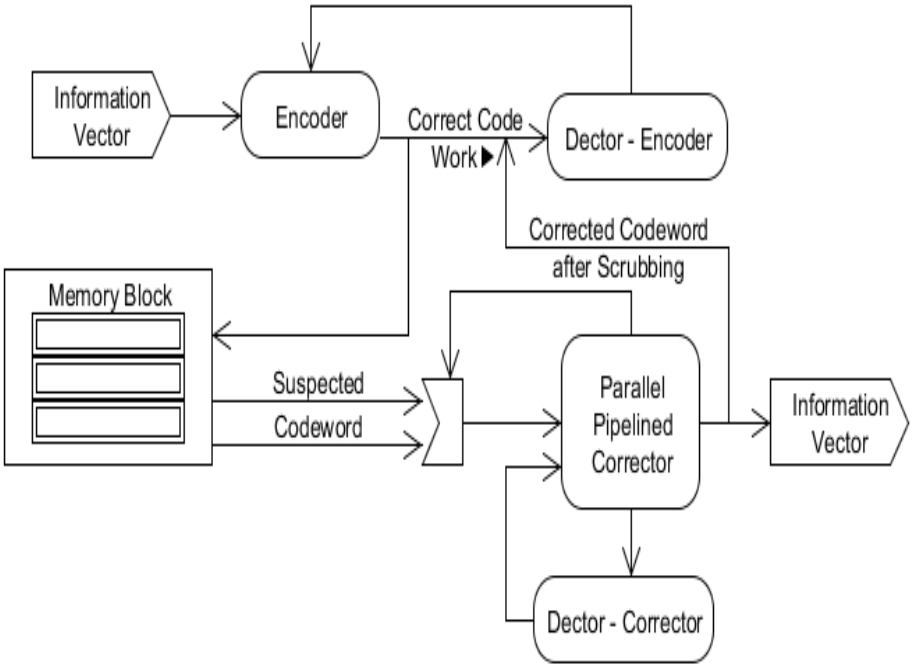

Figure 2. Proposed fault-tolerant memory architecture, with pipelined corrector.

During memory access operation, the stored codewords will be accessed from the memory unit. Codewords are susceptible to transient faults while they are stored in the memory; therefore a corrector unit is designed to correct potential errors in the retrieved codewords. In this design all the memory words pass through the corrector and any potential error in the memory words will be corrected. Similar to the encoder unit, a fault-secure detector monitors the operation of the corrector unit. The Data bits stay in memory for a number of cycles and, during this period, each memory bit can be upset by a transient fault with certain probability. Therefore, transient errors accumulate in the memory words over time. In order to avoid accumulation of too many errors in any memory word that surpasses the code correction capability, the system must perform memory scrubbing. Memory scrubbing is the process of periodically reading memory words from the memory, correcting any potential errors, and writing them back into the memory. To perform the periodic scrubbing operation, the normal memory access operation is stopped and the memory performs the scrub operation. The functional description for the suggested processing unit is as presented below;

3.1. Encoder

Let be a Euclidean Geometry with n points and J lines. EG is a finite geometry that is shown to have the following fundamental structural properties:
1) Every line consists of p points;
2) Any two points are connected by exactly one line;

3) Every point is intersected by y lines;
4) Two lines intersect in exactly one point or they are parallel; i.e., they do not intersect.
Let H be a Jxn binary matrix, whose rows and columns corresponds to lines and points in an Euclidean geometry, respectively, where h_{ij}=1 if and only if the ith line of EG contains the jth point of EG, and h_{ij}=0 otherwise. A row in H displays the points on a specific line of EG and has weight p. A column in H displays the lines that intersect at a specific point in EG and has weight y. The rows of H are called the incidence vectors of the lines in EG, and the columns of H are called the intersecting vectors of the points in EG. Therefore H, is the incidence matrix of the lines in EG over the points in EG. It is shown in that H is a LDPC matrix, and therefore the code is an LDPC code.

3.2. Corrector

1) One-Step Majority-Logic Corrector: One-step majority logic correction is the procedure that identifies the correct value of a each bit in the codeword directly from the received codeword; This method consists of two parts:
1) generating a specific set of linear sums of the received vector bits and
2) finding the majority value of the computed linear sums. The majority value indicates the correctness of the code-bit under consideration; if the majority value is 1, the bit is inverted, otherwise it is kept unchanged.

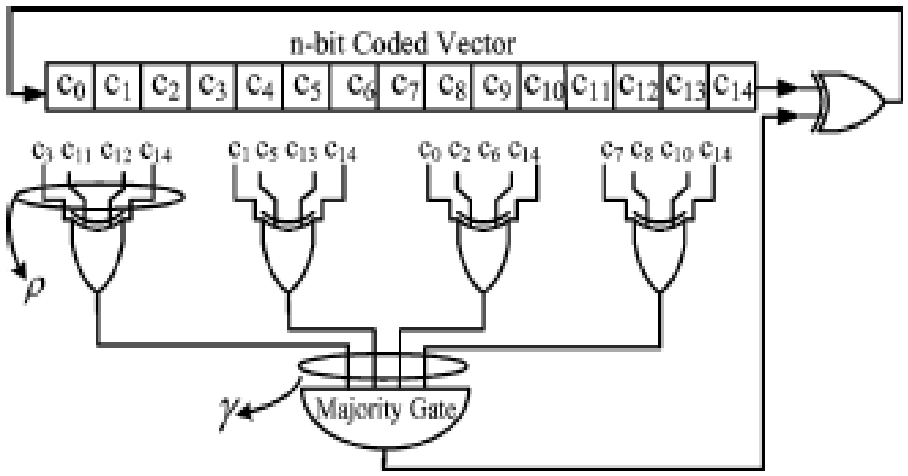

Figure 3. Serial one-step majority logic corrector structure to correct last bit of 15-bit (15, 7, 5) EG-LDPC code.

The circuit implementing a serial one-step majority logic corrector for (15, 7, 5) EG-LDPC code as shown in Figure above. This circuit generates parity-check sums with XOR gates and then computes the majority value of the parity-check sums. Since each parity-check sum is computed using a row of the parity check matrix and the row density of EG-LDPC codes are, each XOR gate that computes the linear sum has inputs. The single XOR gate on the right of Figure 3 corrects the code bit using the output of the majority gate. Once the code bit is corrected the codeword is cyclic shifted and code bit is placed at position and will be corrected. The whole codeword can be corrected in rounds. The proposed design is implemented using VHDL coding for its description. The simulation results obtained and the implementation detail over Xilinx FPGA device is as outlined below.

4. RESULT OBSERVATION

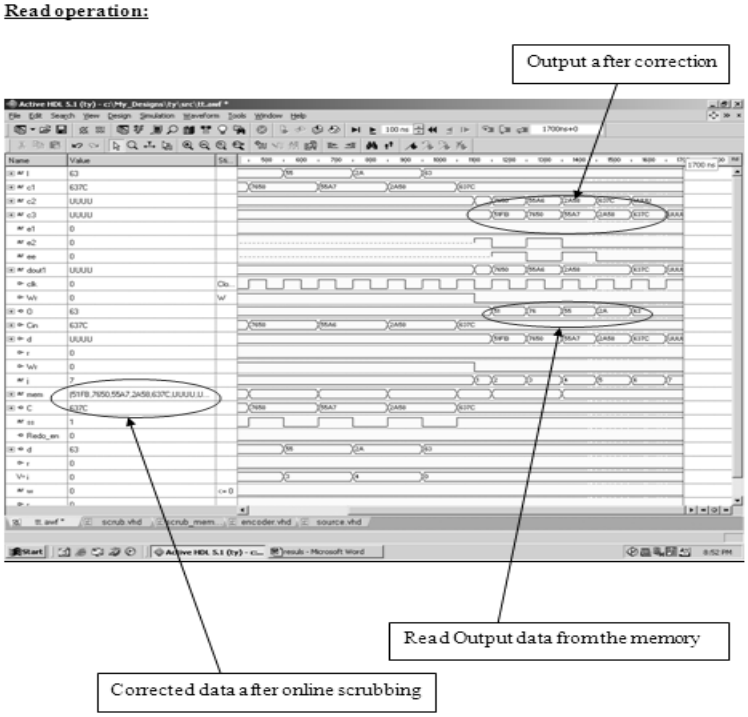

Figure 4. Obtianed timing observation for read operation.

The figure illustrates the write mode opertion for the proposed approach. The design of encoding operation for source reading and the conditions for during-error mode and error free mode is observed. The triggering opertion for redo operation during encoding operation is observed. It is observed that when an error is introduced the redo operation for performng the encoding operation is generated. This results in the elimination of error during encoding operation.

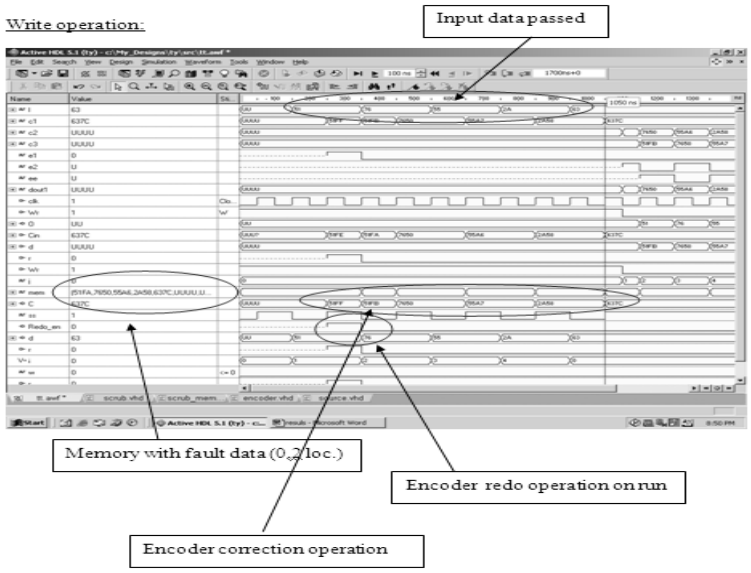

Figure 5. Obtianed timing observation for write operation.

The operation of decoding is observed and the obtained result is as shown above. From the observation it is observed that during writing operation if there is any error introduced either during storage or during reading operation for decoding operation. In case of error coding the decoder triggers the corrector units which intern apply the correction operation on fetched data resulting in error free coding.

The implementation for the proposed system is targeted for the Xilinx-FPGA device and the observations obtained the device utilization summary and the timing summary at the maximum frequency 28.046 MHz for the device obtained are as follows:

```
Design Statistics
# IOs                    : 9

Cell Usage :
# BELS                   : 997
#      GND               : 1
#      INV               : 11
#      LUT1              : 128
#      LUT2              : 26
#      LUT2_D            : 2
#      LUT3              : 141
#      LUT3_D            : 7
#      LUT3_L            : 2
#      LUT4              : 239
#      LUT4_D            : 33
#      LUT4_L            : 16
#      MUXCY             : 187
#      MUXF5             : 64
#      MUXF6             : 15
#      VCC               : 1
#      XORCY             : 124
# Flip-flops/Latches     : 360
#      FD                : 59
#      FDE               : 195
#      FDR               : 1
#      FDRE              : 96
#      FDS               : 1
#      LDC_1             : 1
#      LDE_1             : 7
# Clock Buffers          : 1
#      BUFGP             : 1
# IO Buffers             : 8
#      IBUF              : 1
#      OBUF              : 7
```

```
Device utilization summary:
---------------------------

Selected Device: v1600efg1156-7

 Number of Slices:              380 out of 15552    2%
 Number of Slice Flip Flops:    353 out of 31104    1%
 Number of 4 input LUTs:        605 out of 31104    1%
 Number of IOs:                 9
 Number of bonded IOBs:         9 out of    724     1%
    IOB Flip Flops:             7
 Number of GCLKs:               1 out of      4    25%
```

```
Timing Summary:
---------------
Speed Grade: -7

   Minimum period: 35.656ns (Maximum Frequency: 28.046MHz)
   Minimum input arrival time before clock: 10.958ns
   Maximum output required time after clock: 6.229ns
   Maximum combinational path delay: No path found
```

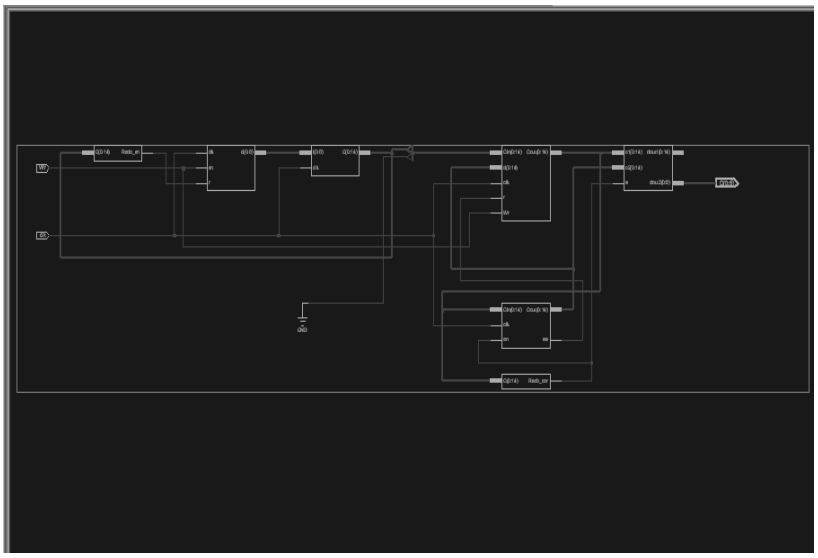

Figure 6. The RTL realization of the developed system on Xilinx V16000efg1156-7 device.

The logical interconnection for the targeted device obtained is as shown above.

The routing interconnection developed for the proposed system is observed as shown below.

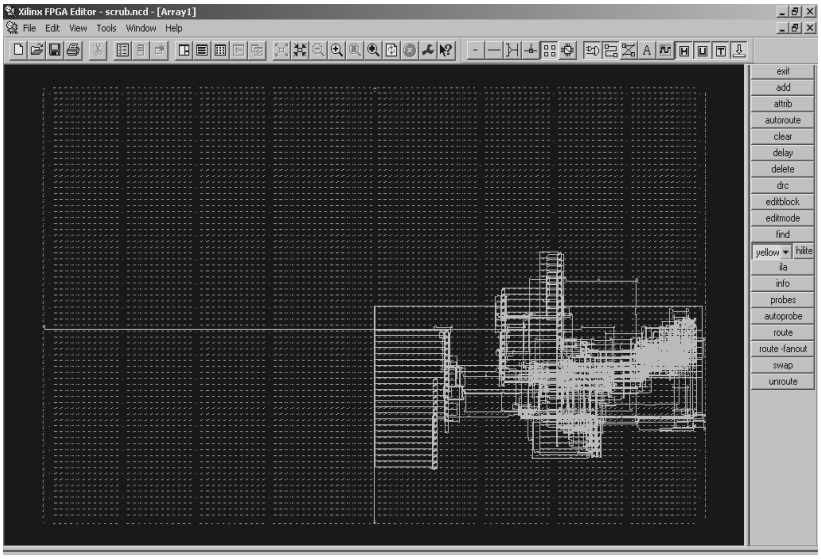

Figure 7. FPGA routing of the implemented system on Xilinx V16000efg1156-7 device.

The targeted device consumes about 1/3rd of the FPGA area for the realization of the suggested logic. The routes for the interconnection of the logical blocks with the port connection is as obtained above.

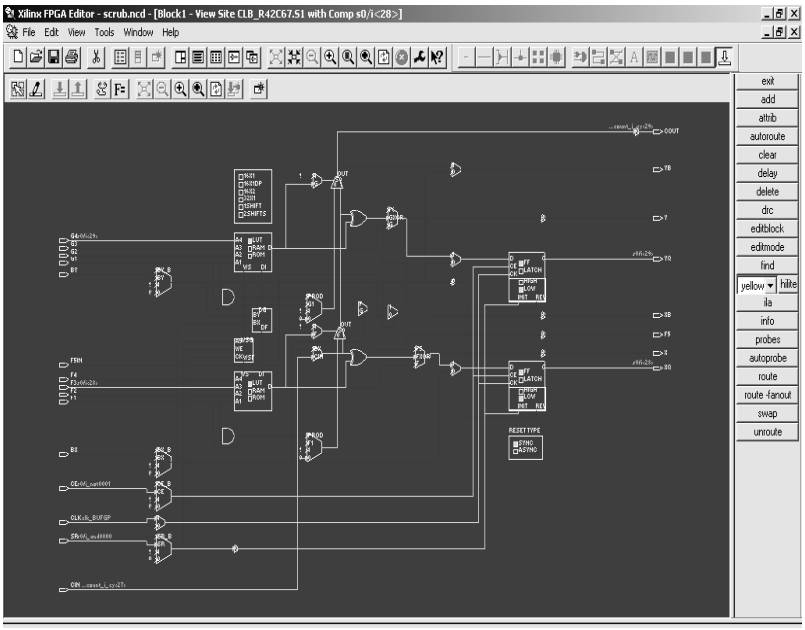

Figure 8. Logical placement for the proposed approach in the FPGA device.

The logical placement for the logical block when implemented onto the targeted device obtained is shown in Figure 8. The device uses the logical interconnections of such configurable block to realize the proposing logic.

```
------------------------------------------------------------------
Release 9.1i - XPower SoftwareVersion: J.30
Part:         v1600efg1156-7
Data version: PRODUCTION, v1.0, 05-28-03

Power summary:                          I(mA)     P(mW)
------------------------------------------------------------------
Total estimated power consumption:                  367
                                  ---
                      Vccint 1.80V:       200       360
                      Vcco33 3.30V:         2         7
                                  ---
                           Clocks:          0         0
                           Inputs:          0         0
                            Logic:          0         0
                          Outputs:
                             Vcco33         0         0
                          Signals:          0         0
                                  ---
           Quiescent Vccint  1.80V:       200       360
           Quiescent Vcco33  3.30V:         2         7

Thermal summary:
------------------------------------------------------------------
   Estimated junction temperature:                  30C
                  Ambient temp:  25C
                     Case temp:  29C
                     Theta J-A:  13C/W

Analysis completed: Mon Jul 06 21:44:37 2009
------------------------------------------------------------------
```

The obtained power consumption report for the targeted device is as shown above. The device for the fault correcting unit consumes about 367mW power operation. The developed system is hence obtained with the specification of operating power of 360mW and with the operating frequency of 28.046MHz frequency.

5. CONCLUSIONS

In this paper, a fully fault-tolerant memory system that is capable of tolerating errors not only in the memory bits but also in the supporting logic including the encoder and corrector is presented. A Euclidean Geometry coding approach for the fault tolerance is developed. Using these FSDs a fault-tolerant encoder and corrector is designed, where the fault-secure detector monitors their operation and provides controlling signal for their operation.A unified approach to tolerate permanent defects and transient faults is also focused. This approach reduces the area overhead. With this technique to tolerate errors in the encoder logic, a reliable encoders and decoder logic is realized.

It is expected that devices become less reliable in smaller feature sizes and experience both more permanent defects due to the imperfect manufacturing process and more transient faults due to the effect of noise. Providing reliability is becoming constantly more challenging due to increase in both the device failure rate and system complexity up to the point that the conventional techniques will not be efficient enough or even capable of tolerating these error rates and complexity for the future generation systems. Hence we are going to provide robust defect- and fault- tolerant designs which include fine grained and global reliability techniques to achieve reliability in the systems.

ACKNOWLEDGMENT

The authors place on record their grateful thanks to the authorities of KL University, Vijayawada, A.P, India,St.John's College of Engineering and Technology,Kurnool,A.P,India and Jawaharlal Nehru Technological University, College of Engineering, Hyderabad, India for providing the facilities.

REFERENCES

[1] M. Forshaw, R. Stadler, D. Crawley, and K. Nikolic´, "*A short review of nanoelectronic architectures"*, Nanotechnology, vol. 15, pages S220–S223, 2004.

[2] Clark, J. and Holton, D. A., "*A First Look at Graph Theory"*, 687 Hartwell Streets, Teaneck, NJ 07666: World Scientific Publishing Co. Pvt. Ltd., 1991.

[3] R. G. Gallager, "*Low-Density Parity-Check Codes*", Cambridge, MA: MIT Press, 1963.

[4] Wicker, S.B., "*Error Control Systems for Digital Communications and Storage"*, Upper Saddle River, New Jersey 07458: Prentice Hall, 1995.

[5] C. E. Shannon, "*A Mathematical Theory of Communication"*, Bell System Technical Journal, vol. 27, pages 379-423, 1948.

[6] E. Berlekamp, R. McEliece, and H. van Tilborg, "*On the inherent intractability of certain coding problems"*, IEEE Trans. Inform. Theory, vol. 24, pages 384-386, 1978.

[7] R. G. Gallager, "*Low Density Parity-Check Codes",* MIT Press, Cambridge, MA, 1963.

[8] J. Pearl, "*Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference"*, Morgan Kaufmann Publishers, Inc., 1988.

[9] M. Luby, M. Mitzenmacher, and A. Shokrollahi, "*Analysis of random processes via and-or tree evaluation"*, in Proceedings of the 9th Annual ACM-SIAM Symposium on Discrete Algorithms, pages 364-373, 1998.

[10] International Technology Roadmap for Semiconductors,",, 2005.

[11] R. Sinnott, A. Asenov, D. Berry, D. Cumming, S. Furber, C. Millar, A. Murray, S. Pickles, S. Roy, A. Tyrrell, and M. Zwolinksi, "Meeting the Design Challenges of VLSI CMOS Electronics: An

Introduction to an Upcoming EPSRC Pilot Project," in Proceedings of the UK e-Science All Hands Meeting, Nottingham, UK, September 2006.

[12] B.P.Wong, A.Mittal, Y.Cao, and G.Starr, VLSI-CMOS Circuit and Physical Design.John Wiley and Sons, Inc., 2005.

Authors

1. **Harikishore.Kakarla** was born in Vijayawada, Krishna (Dist.), AP, India. He received B.Tech. in Electronics & Communication Engineering from St. Johns College of Engg.&Tech,Kurnool(Dist.,),AP, India, M.Tech from G. Pulla Reddy Engineering College, Kurnool, AP, India. He is pursuing Ph.D in the area of VLSI in KL University, Vijayawada, AP, India. He is working as Assistant Professor for Department of Electronics & Communication Engineering, KL University, Vijayawada, AP, India. He has published one National Conference.
E-mail: Kakarla.harikishore@gmail.com

2. **Madhavi Latha. M** was born in Guntur (Dist.,), AP, India. She received B.E in Electronics & Communication Engineering from Achary Nagarjuna University, Guntur (Dist.), AP, India, M.Tech. from JNTUH, Hyderabad, India. She received Ph.D degree in Engineering from JNTUH, Hyderabad, AP, India. Currently, she is working as Professor & Head for Department of Electronics & Communication Engineering, JNTUH, Hyderabad, AP, India and currently working in VLSI field. Her research interest includes design of Low power and mixed signal circuits. She has published 32 publications in various journals and conferences at National and International level and presented papers in conferences held at Lasvegas, Lousiana, USA and Iunstrruck, Austria presently guiding nine students for Ph.D and one student for M.S. She has conducted Ten UGC refresher courses and in DSP, VLSI & Embedded systems, workshops in EDA Tools for VLSI, CMOS & ASIC Designs, MATLAB Programming & Applications. She is the life member of FIETE, and MISTE.

3. **Habibulla Khan** was born in Vijayawada, Krishna (Dist.), AP, India. He received B.Tech in Electronics & Communication Engineering from VR Siddhartha Engineering College, Vijayawada, AP, India, M.Tech. from CIT Engineering College, Coimbatore, Tamilnadu, India. He received Ph.D degree in Engineering from Andhra University, Visakhapatnam, AP, India. Currently, he is working as Professor & Head for Department of Electronics & Communication Engineering, KL University, Vijayawada, AP, India, and currently working in RADAR and MICROWAVE Engineering field. He has published 15 publications in various journals and conferences at National and International level and presented papers in conferences.