ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

Fault Secure Encoder and Decoder For Memory Applications

V.Jeevitha[1]

ME-VLSI Design, Shreenivasa Engineering College, B.Pallipatti, Dharmapuri, Tamilnadu, India[1]

Abstract: In a recent paper, a method was proposed to accelerate the majority logic decoding of difference set low density parity check codes. This is useful as majority logic decoding can be implemented serially with simple hardware but requires a large decoding time. For memory applications, this increases the memory access time. The method detects whether a word has errors in the first iterations of majority logic decoding, and when there are no errors the decoding ends without completing the rest of the iterations. Since most words in a memory will be error-free, the averaged encoding time is greatly reduced. In this brief, we study the application of a similar technique to a class of Euclidean geometry low density parity check (EG-LDPC) codes that are one step majority logic decodable. The results obtained show that the method is also effective for EG-LDPC codes.

Key Words:-Error correction codes, Euclidean geometry low-densityparity check (EG-LDPC) codes, majority logic decoding, memory**.**

I. INTRODUCTION

Error correction codes are commonly used to protect memoriesfrom so-called soft errors, which change the logical value of memorycells without damaging the circuit. As technology scales, memory devices become larger and more powerful error correction codes are needed. To this end, the use of more advanced codes has been recently proposed. These codes can correct a larger number of errors, but generally require complex decoders. To avoid a highdecoding complexity, the use of one step majority logic decodable codes was first proposed in for memory applications. One step majority logicdecoding can be implemented serially with very simple circuitry, but requires long decoding times. In a memory, this would increasethe access time which is an important system parameter. Only afew classes of codes can be decoded using one step majority logic decoding. Among those are some Euclidean geometry low densityParitycheck (EG-LDPC) codes which were used in, and differenceset low density parity check (DS-LDPC) codes. A method was recently proposed in to accelerate a serial implementationof majority logic decoding of DS-LDPC codes. The idea behindthe method is to use the first iterations of majority logic decodingto detect if the word being decoded contains errors. If there are no errors,then decoding can be stopped without completing the remainingiterations, therefore greatly reducing the decoding time.

II. EXISTING SYSTEM

The information bits arefed into the encoder to encode the information vector, and thefault secure detector of the encoder verifies the validity of theencoded vector. If the detector detects any error, the encodingoperation must be redone to generate the correct codeword. Thecodeword is then stored in the memory. During memory accessoperation, the stored codewords will be accessed from thememory unit. Codewords are susceptible to transient faultswhile they are stored in the memory; therefore a corrector unit isdesigned to correct potential errors in the retrieved codewords.

In Existing design all the memory words pass through thecorrector and any potential error in the memory words will becorrected. Similar to the encoder unit, a fault-secure detectormonitors the operation of the corrector unit. All the units showing Fig 2 are implemented in fault-prone, nano scale circuitry; the only component which must be implemented in reliable circuitryare two OR gates that accumulate the syndrome bits for the detectors (shown in Fig 2).

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

Data bits stay in memory for anumber of cycles and, during this period, each memory bitcan be upset by a transient fault with certainprobability. Therefore, transient errors accumulate in thememory words over time. In order to avoid accumulation of toomany errors in any memory word that surpasses the codecorrection capability, the system must perform memoryscrubbing. Memory scrubbing is the process of periodically

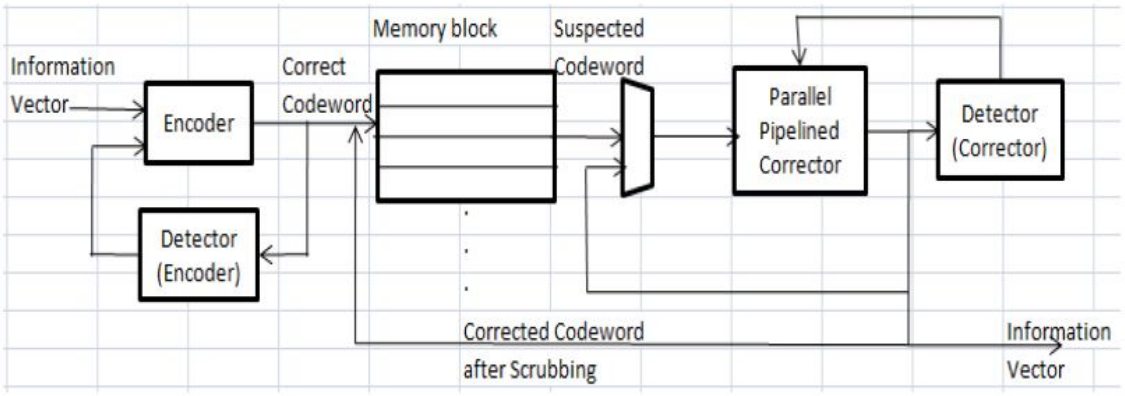

Fig:1 Existing EG LDPC

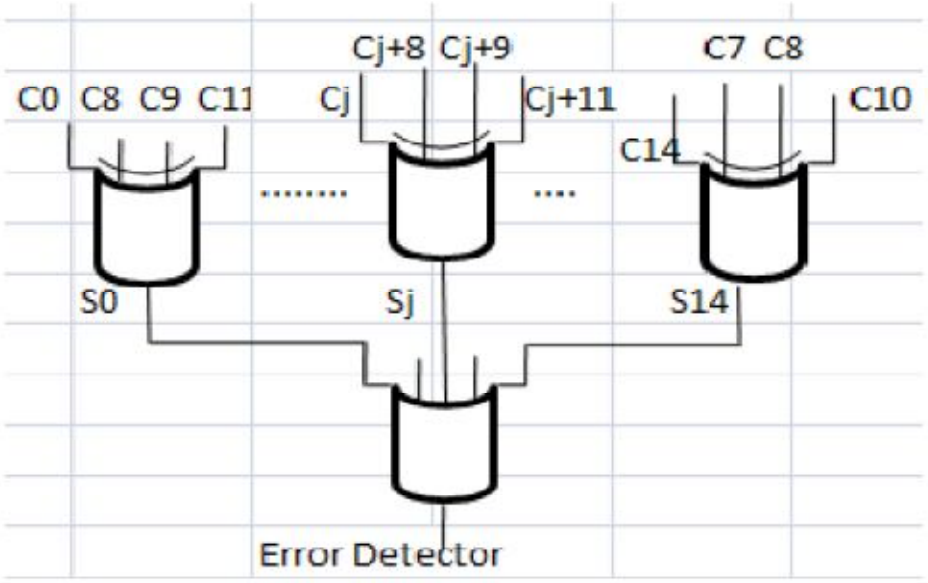

Fig :2fault-secure detector for(15,7,5)EG-LDPC code

For a code with block length N, majority logic decoding (when implemented serially) requires N iterations, so that as the code size grows, so does the decoding time.

An n-bit codeword, which encodes a n-bit information Vector I is generated by multiplying the n-bit information vector with akXn bit generator matrix; i.e., c=i.G. EG-LDPC codes are notsystematic and the information bits must be decoded from theencoded vector, which is not desirable for our fault-tolerantapproach due to the further complication and

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

delay that it addsto the operation. However, these codes are cyclic codes.

Weused the procedure to convert the cyclic generator matrices tosystematic generator matrices for all the EG-LDPC codes underconsideration shown in Fig.3.The encoded vector consists ofinformation bits followed by parity bits, where each parity bit issimply an inner product of information vector and a column ofX, from G=[I:X]note the identity matrix in the left columns.

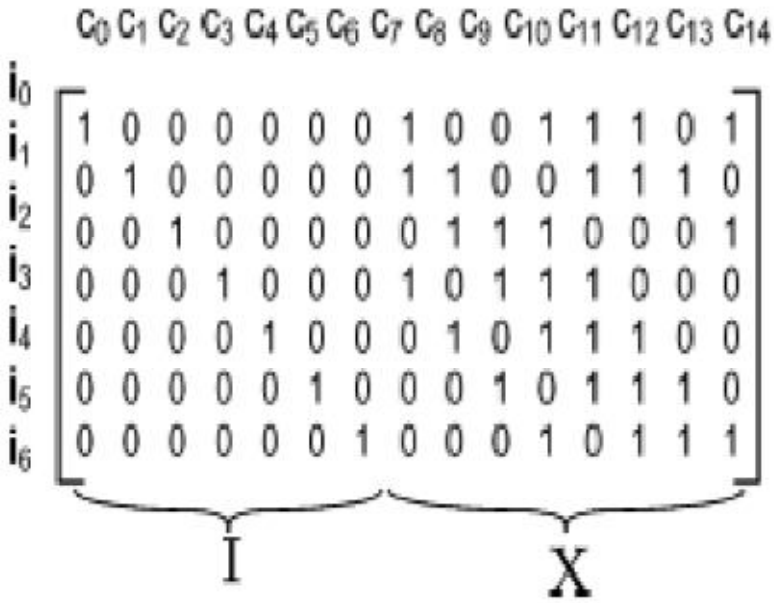

Fig: 3. Generator matrix for the (15, 7, 5) EG-LDPC insystematic format;

2.1 PLAIN MLD WITH SYNDROME FAULT DETECTOR (SFD)

The SFD is an XOR matrix that calculates the syndrome based on the parity check matrix. Each parity bit results in a syndrome equation.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

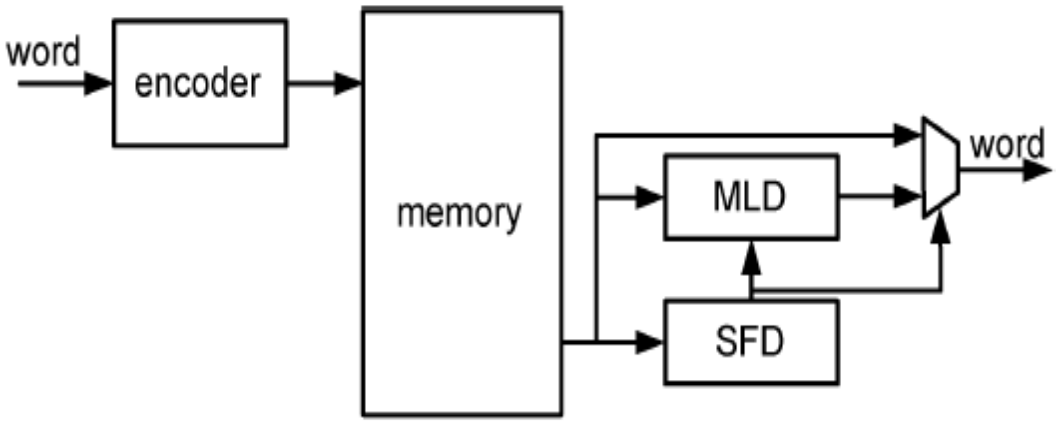

Fig: 4. Schematic of an ML with SFD

Therefore, the complexity of the syndrome calculator increases with the size of the code. A faulty codeword is detected when at least one of the syndrome bits is "1." This triggers the MLD to start the decoding, as explained before. On the other hand, if the codeword is error-free, it is forwarded directly to the output, thus saving the correction cycles. In this way, the performance is improved in exchange of an additional module in the memory system: a matrix of XOR gates to resolve the parity check matrix, where each check bit results into a syndrome equation. This finally results in a quite complex module, with a large amount of additional hardware and powerconsumption in the system.

III. PROPOSED SYSTEM

3.1 MAJORITY LOGIC DETECTOR/ DECODER (MLDD)

If errors can be detected in the first few iterations of MLD, thenwhenever no errors are detected in those iterations, the decoding canbe stopped without completing the rest of the iterations. In the first iteration, errors will be detected when at least one of the check equationsis affected by an odd number of bits in error. In the second iteration, as bits are cyclically shifted by one position, errors will affect otherequations such that some errors undetected in the first iteration will bedetected. As iterations advance, all detectable errors will eventually bedetected.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

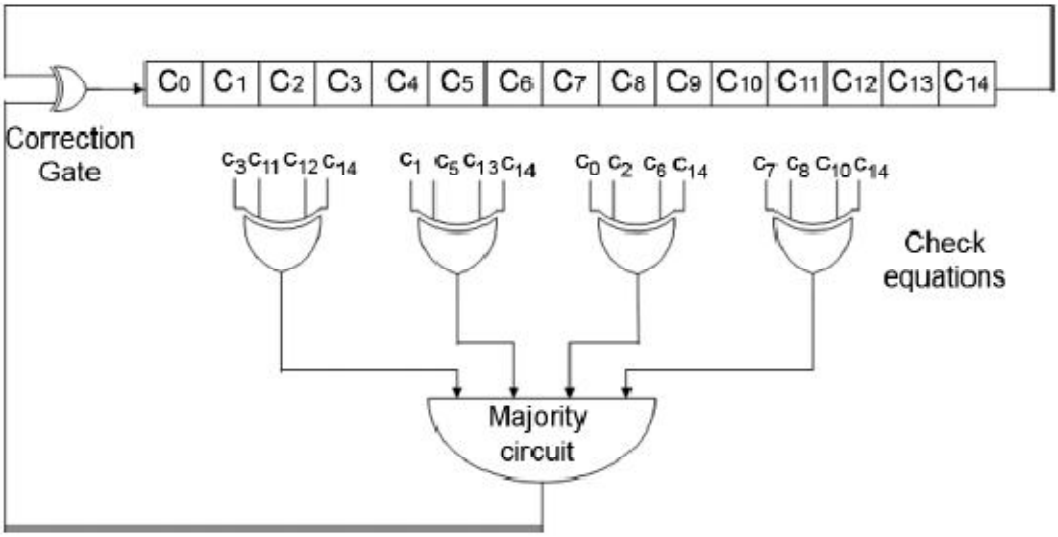

Fig: 5 serial one-step majority logic decoder for the(15,7)EG-LDPC code

ADVANTAGES

1. In the proposed approach, only the first threeiterations are used to detect errors, thereby achieving a large speed increasewhen N is large. In [10] it was shown that for DS-LDPC codes, all error combinations of up to five errors can be detected in the firstthree iterations.
2. Errors affecting more than five bits were detectedwith a probability very close to one.
3. The probability of undetected errorswas also found to decrease as the code block length increased. Fora billion error patterns only a few errors (or sometimes none) were undetected. This may be sufficient for some applications.
4. Another advantage of the proposed method is that it requires verylittle additional circuitry as the decoding circuitry is also used for errordetection. The additional arearequired to implement the scheme was only around 1% for large wordsizes.

3.2CORRECTOR

One-step majority-logic correction is a fast and relatively compact error-correcting technique. There is a limited classof ECCs that are one-step-majority correctable which includetype-I two-dimensional EG-LDPC.

ONE-STEP MAJORITY-LOGIC CORRECTOR

One-step majority logic correction is the procedure thatidentifies the correct value of an each bit in the codeword directlyfrom the received codeword; this is in contrast to the generalmessage-passing error correction strategy which maydemand multiple iterations of error diagnosis and trialcorrection. Avoiding iteration makes the correction latency bothsmall and deterministic. This technique can be implementedserially to provide a compact implementation or in parallel tominimize correction latency. This method consists of two parts:

1) Generating a specific set of linear sums of the received vectorbits
2) Finding the majority value of the computed linearsums.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

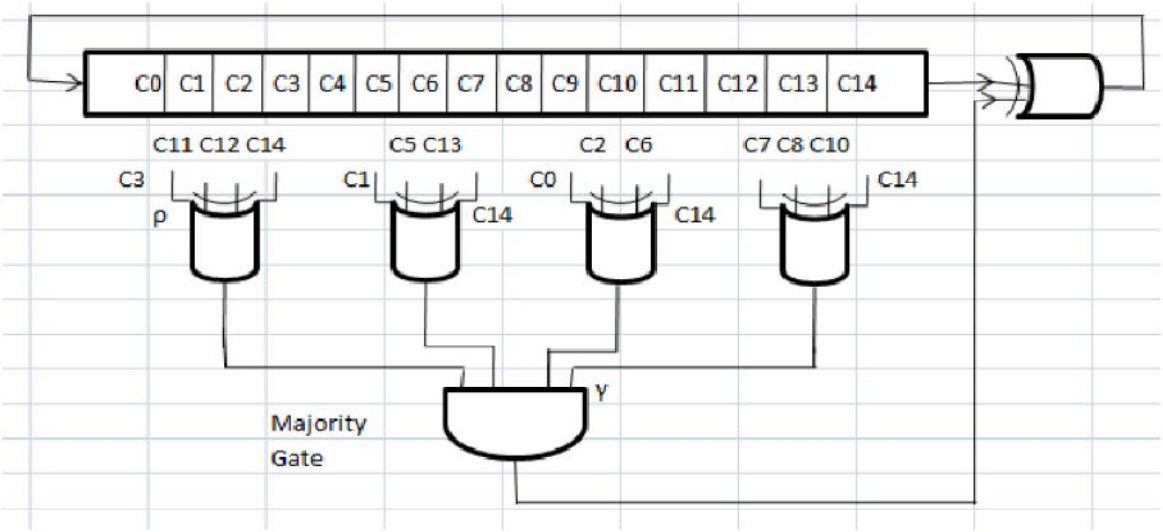

Fig: 6 Serial one-step majority logic corrector structure to correct last bit (bit 14th) of 15-bit (15, 7, 5) EG-LDPC code.

The majority value indicates the correctness of the codebitunder consideration; if the majority value is 1, the bit isinverted, otherwise it is kept unchanged. The circuitimplementing a serial one-step majority logic corrector for (15,7, 5) EG-LDPC code is shown in Fig 6

3.3 MAJORITY LOGIC DETECTOR/ DECODING ALGORITHM

The Modified MLDD algorithm performs the decoding as in the MLDD with some modifications..Modified MLDD algorithm requires additional logic compared to the MLDD algorithm. The corrections are performed during the first n iterations. A counter is used to determine if there have been more than t errors in those iterations and based on the result the corrected or the original register is send to the output. If the majority gate detect any error in codeword the iterations take place depends upon the length of the codeword's.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

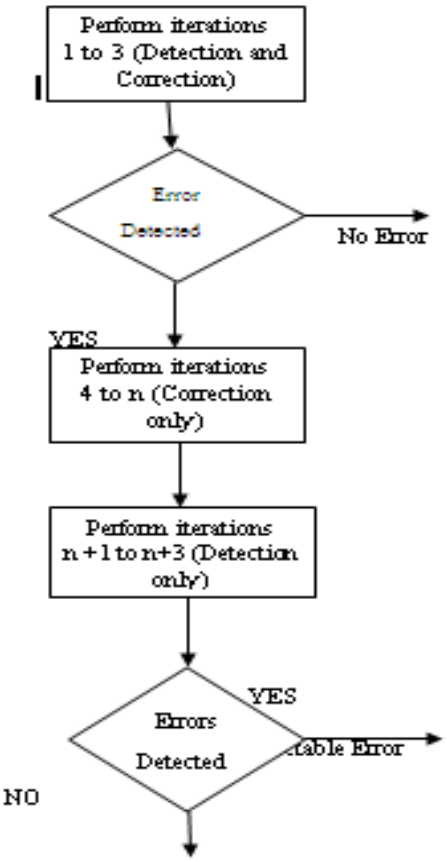

Fig: 7. Flow diagram of the MLDD algorithm.

Modified MLDD detect more than five bit-flips and high efficiency compare to Majority logic decoder. The control unit manages the detection process. It uses a counter that counts up to three, which distinguishes the first three iterations of the ML decoding. 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.
The process stops after three iteration if the codeword length less than ten and six iteration for codeword less than twenty, if the codeword greater than 20 then nine iterations are performed.
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.

IV. CONCLUSION

In this brief, the detection of errors during the first iterations of serialone step Majority Logic Decoding of EG-LDPC codes has beenstudied. The objective was to reduce the decoding time by stopping thedecoding process when no errors are detected. The simulation resultsshow that all tested combinations of errors affecting up to four bits aredetected in the first three iterations of decoding. These results extendthe ones recently presented for DS-LDPC codes, making the modifiedone step majority logic decoding more attractive for memory applications. The designer now has a larger choice of word lengths and error correction capabilities. Future work includes extending the theoretical analysis to the casesof three

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

and four errors. More generally, determining the requirednumber of iterations to detect errors affecting a given number of bitsseems to be an interesting problem. A general solution to that problemwould enables a fine-grained tradeoff between decoding time and errordetection capability.

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 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 frameworkfor 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.