RESEARCH ARTICLE OPEN ACCESS

FPGA Implementation of Majority Logic Fault Detection and Correction for Memory Application

Jayasri.T.S
Department of Electronics & Communication Engineering
Government Women's Polytechnic College
Kottakkal, Indiajayasrideepam@gmail.com

***Abstract*— As far as memory applications are concerned the soft errors are always a problem. This paper mainly focuses on the design of an efficient Majority Logic Detector / Decoder (MLDD) for fault detection along with correction off ault for memory application. The error detection and correction method is done by one step majority logic decoding and is made effective for Euclidean Geometry Low Density parity check codes (EG-LDPC). The proposed fault detection method can detect the faultin less decoding cycles. The technique keeps area minimal and power consumption low for large code word sizes.**

***Keywords*- Error Correcting Codes, Euclidean geometry low density parity check codes, ML codes, FPGA, Verilog.**

I. Introduction

Memory cellsaresusceptibletosofterrors.Toprotectmemory cells from soft errors encoder and decoder circuits areused.

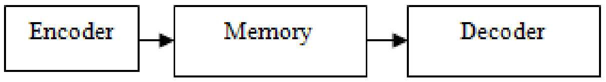


Figure1.BasicBlockdiagram

Theencoderencodestheinformationbitsusingerrorcorrection codes and this encoded bit is stored in the memory.Thedifferenttypesoferror correctioncodesare

- SER(SingleErrorCorrection)
- SEC-DED(SingleErrorCorrection–DoubleErrorDetection)
- RS(reedSolomon)
- BCH(BoseChaudhuriHoequenghem)
- CyclicCodes

AmongtheECCcodes,cycliccodesarebestsuitedbecause of their highererror correction capability andlowdecodingcomplexity.[2] [3].

All other codes are not suitable because they have morecomplexdecodingalgorithmsandincreasecomputationalcosts[1].Cycliccodeshaveapropertyof Majoritylogicdecodable(MLD).Inthispaperonespecificonespecifictype Lowdensityparitycheckcode(LDPC)calledEuclidianGeometrycyclic codes(EG-LDPC) are used.

EG-LDPC codes are low density parity check codes .Theseare majority logic decodable. This type of code uses the checksum algorithm.Thechecksumalgorithmisnothingbutanumerical value is associatedwith the code word which is tobe transmitted..At the receiver end the codeword receivedhassomenumericalvalue.Theexistingmethodisimplementedusingbasichardware.

TableI.EuclideanGeometryLDPCCodes

Cordwordbits	Informationbits	Paritybits	Checksum
15	7	8	4
63	37	26	8
255	175	80	16
1023	781	242	32

The MLD technique uses Serial One Step Majority LogicDecoderisusedtodetecttheerrorsserially.Theserialonestep majority logic decoder algorithm for error detection andcorrectionisexposedinFigure2.

TheMLdecoder consistsofmainlytwosteps.

1. Generatingthechecksumequations usingXORmatrix
2. Determining the majority value ofthe computedlinearsums

In this decoder 15 bit data is first stored in the cyclic shiftregister.Thentheinputsaregiven totheXORgates.TheXOR gates required are four because the input is a 15 bit data.The bit to be detected should be given as one of the inputs forall the XOR gates. The XOR gates outputs are the check sumequationswithsomenumericaldata‟s.Thechecksumequations consist of 0‟s and 1‟s that are binary datas. Then theMajority circuit outputs the data which is in majority numberof 1‟s.If theoutputoftheonestepmajoritycircuit ismajority numberof„1" then thecorrespondingbithas theerrorelsethebitiserror free.

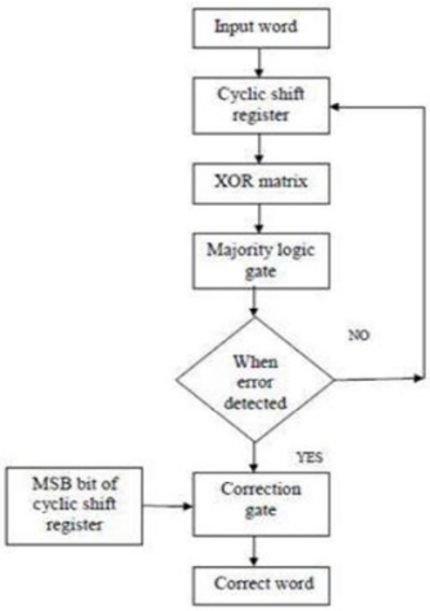


Figure2.Theserialonestepmajoritylogicdecoderalgorithm

The output of the Majority circuit is given as one of theinput to the correction gate. The bit which is under test is theother input to the correction gate. The corrected bit is storedinto the shift register after the first cyclic shift. The entireprocess is called single iteration. Similarly three iterations areprocessed. First three iterations are required to detect all theerrorsofanynumber.

II. EG-LDPCEncoder Structure

The systematic generator matrix to generate (15, 7, 5)EG-LDPC code is shown in Figure 3 [6]. The encoded vectormainlyconsistsoftwoparts,thefirstpartconsistofinformation bits and second part is the parity bits, where eachparity bit is simply an inner product of information vector andacolumnofX , fromG=[I:X].

The encoder circuit [6] to compute the parity bits of the(15, 7, 5) EG-LDPC code is shown in Figure 4. In this figure,the information vectors are (i0,….i6) and will be copied to(c0,..,c6) bits of the encoded vector, c. The rest of encodedvector (c7…c14), that is the parity bits are the linear sums(XOR)ofthe informationbits.

$$\begin{array}{c|ccccccccccccccc}
 & C_0 & C_1 & C_2 & C_3 & C_4 & C_5 & C_6 & C_7 & C_8 & C_9 & C_{10} & C_{11} & C_{12} & C_{13} & C_{14} \\
\hline
i_0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 \\
i_1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 \\
i_2 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\
i_3 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\
i_4 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\
i_5 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 \\
i_6 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\
\hline
 & \multicolumn{7}{c}{\mathrm{I}} & \multicolumn{8}{c}{\mathrm{X}}
\end{array}$$

Figure3.Generatormatrixforthe(15,7,5)EG-LDPCcode

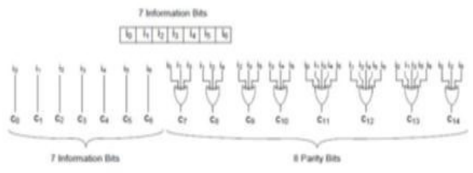


Figure 4. Structure of an encoder circuit for the (15, 7, 5) EG-LDPC code

III. MLDDStructure

MLDDstructure thesame decoding algorithm as the onein Figure 2. The advantage is that, proposed method stopsintermediately in the third cycle when there is no error in dataread, [2] as illustrated in Figure.56, instead of decoding it forthe whole codeword size of N. The xor matrix is evaluated forthe first three cycles of the decoding process, and when all theoutputs{Bj}is“0,”thecodewordisdeterminedtobeerrorfree and forwarded directly to the output. On other hand, theproposed method would continue the whole decoding processto eliminate the errors [2] if the {Bj} contain at least a “1” inanyofthe three cycles.

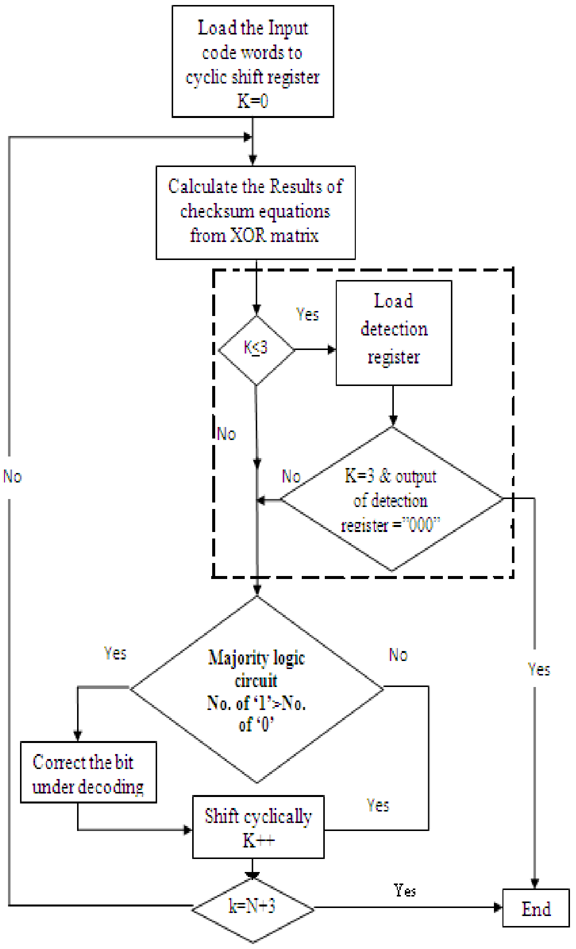

Load the Input code words to cyclic shift register K=0
Calculate the Results of checksum equations from XOR matrix
K≤3
Yes
Load detection register
No
No
K=3 & output of detection register ="000"
No
Yes
Majority logic circuit No. of '1'>No. of '0'
No
Yes
Correct the bit under decoding
Shift cyclically K++
Yes
k=N+3
Yes
End

Figure5. FlowdiagramoftheMLDDalgorithm

A detailedschematicof theproposeddesign for15bitcodewordisshowninFigure 6.

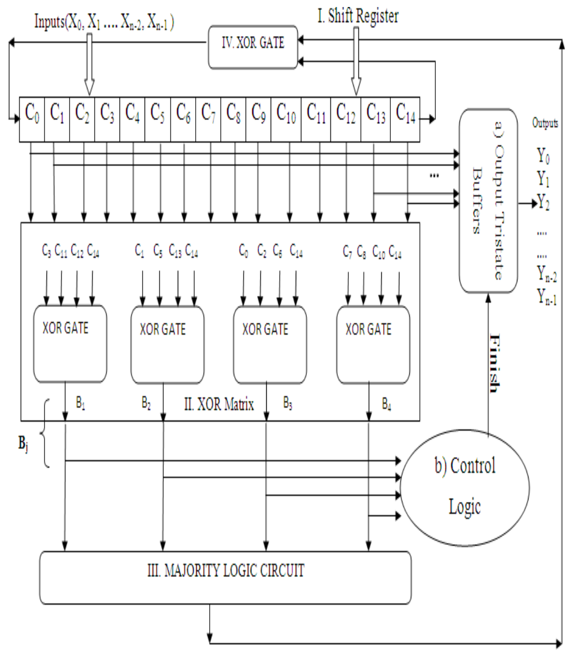

Inputs(X0, X1 .... Xn-2, Xn-1)
I. Shift Register
IV. XOR GATE
C0 C1 C2 C3 C4 C5 C6 C7 C8 C9 C10 C11 C12 C13 C14
a) Output Tristate Buffers
Outputs
Y0 Y1 Y2 Yn-2 Yn-1
C3 C11 C12 C14
C1 C5 C13 C14
C0 C2 C6 C14
C7 C9 C10 C14
XOR GATE
B1 B2 II. XOR Matrix B3 B4
Bj
Finish
b) Control Logic
III. MAJORITY LOGIC CIRCUIT

Figure6.SchematicoftheproposedMLDDfor15bitcodeword

Adetailedschematicoftheproposeddesignfor15bitcodeword is shown in Figure 6. The figure shows the basic MLdecoder with a 15-tap shift register, an XOR array to calculatethe orthogonal parity check sums and a majority logic circuitwhich will decide whether the current bit under decoding iserroneousandtheneedforitsinversion.TheplainMLdecoder [2] showninFigure2isalsohavingthesameschematicstructure up to this stage. The additional hardware [2] intendedfor fault detection illustrated in Figure 6 are: a) the controllogic unit and b) the output tristate buffers. The control unittriggersafinishflagwhenthereisnoerrorsaredetectedindatare ad. The output tristate buffers are always in high impedancestate until the control unit sends the finish signal so that thecurrent values are forwarded to the output y from the shiftregister.

The control logic schematic [2] is illustrated in Figure 7.The detection process is managed by the control unit[7]. Fordistinguishing the first three iterations of the ML decoding, acounter is used here which counts up to three cycles. Thecontrol unit evaluates the output from xor matrix Bj by givingit as input to the OR 1 gate. This output value is fed to twoshift registerswhichhastheresults of the previousstagesstoredinit.Thevaluesareshiftedaccordingly.Thet hirdcominginputisdirectlyforwardedtotheOR2gateandfinally all are evaluated in the third cycle in the OR 2 gate. Ifthe result is "0," a finish signal is send by the FSM whichindicatesthattheprocessedwordiserror-free.TheMLdecodingprocessrunsuntiltheend, ifthe resultis "1".

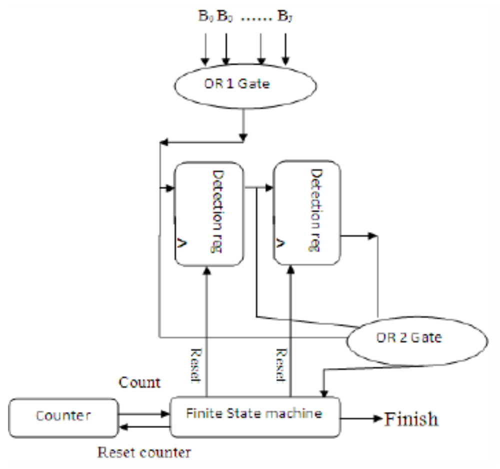


Figure7.Controllogic

The majority logic gate decoding is implemented by usingverilog. That is two level logic [6]. If during the memory readaccess an error is detected, the XOR gate will correct it, byinverting the current bit under decoding. The EG LDPC codeused here is only for 15 bits, it have only outputs four outputsfromxormatrix.TheorthogonalmajorityparametersB1,B 2,…BNareconstructedusingsortingnetworks.Thisclearly provides a performance improvement respect to thetraditional method .The proposed method mostly would onlytake three cycles for decoding. Since most of the words wouldbe error free and would need to perform the whole decodingprocessonlyforthosewordswitherrors.

IV. EXPERIMENTALRESULTS

InthissectionthesimulationsresultsoftheproposedMajorityL ogicDecoder/Detectorandtheencoderispresented.Thefrontendd esignofthearchitecture,itssimulation, synthesis and comparison are done using XILINXISE.DesignSuite7.1.ThetargetdeviceisSpartan3E-XC3S400. The designs are coded in Verilog HDL language. Acodewordofsize15ischosenherefordesigning.Theproposed majority logic decoder and encoder techniques aresimulated both in XILINX and FPGA for both error free anderroneous conditions and the results are shown below in figure8, 9,10,11,12and13.

V. SIMULATIONRESULTSUSINGXILINX

A. *Simulationresultofencoder*

Figure 8 shows the simulation result of an Encoder. Thedata input to given to the memory is encoded first through thisencoderblock.Theinputtotheencoderisthe7bitinformationan d outputisthe15 bitinformation.

B. *Simulationresultof MLDDwithouterror*

Figure 9 shows the Majority logic decoder without error.Theinputcisthe15bitinformationandtheclockgiven.The controloutputislwhichisalsothe15bitinformation.Theoutputisobtainedafter thirdclock.

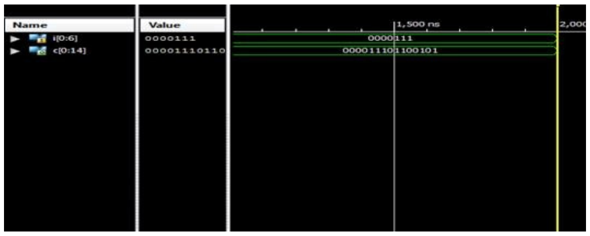


Figure8.Simulationresultoftheencoder

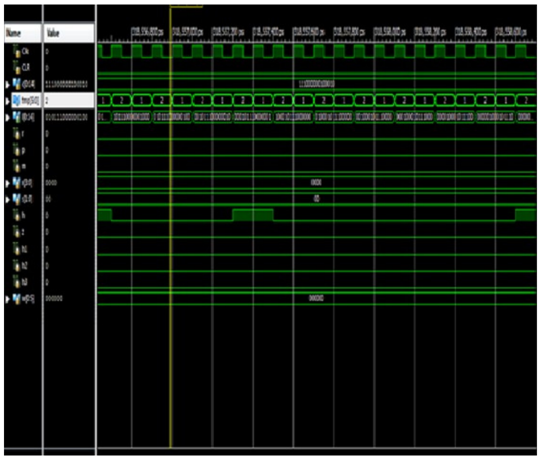


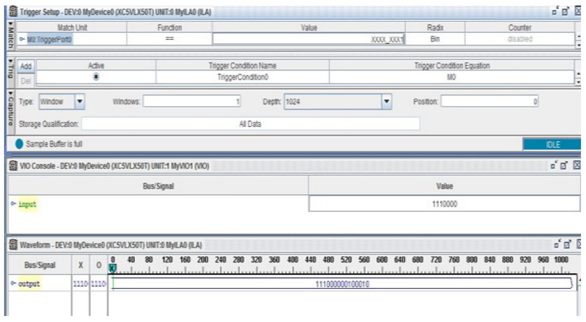
Figure9.SimulationResultofMLDDwithouterror

C. SimulationresultofMLDDwith error

Figure 9 shows the Majority logic decoder without error.The input c is the 15 bit information and the clock given. Thecontrol output is l which is also the 15 bit information. Theoutputisobtainedafter18clocks.

VI. RESULTS OFFPGAIMPLEMENTATION

A. FPGASimulationresultofencoder



Trigger Setup - DEV:0 MyDevice0 (XC5VLX50T) UNIT:0 MyILA0 (ILA)

Match Unit	Function	Value	Radix	Counter
M0:TriggerPort0	==	XXXX_XXX1	Bin	disabled

Active	Trigger Condition Name	Trigger Condition Equation
●	TriggerCondition0	M0

Type: Window Windows: 1 Depth: 1024 Position: 0
Storage Qualification: All Data
Sample Buffer is full IDLE

VIO Console - DEV:0 MyDevice0 (XC5VLX50T) UNIT:1 MyVIO1 (VIO)

Bus/Signal	Value
input	1110000

Waveform - DEV:0 MyDevice0 (XC5VLX50T) UNIT:0 MyILA0 (ILA)

Bus/Signal	X	O	
output	1110	1110	111000000100010

Figure10.FPGASimulationresultofEncoder

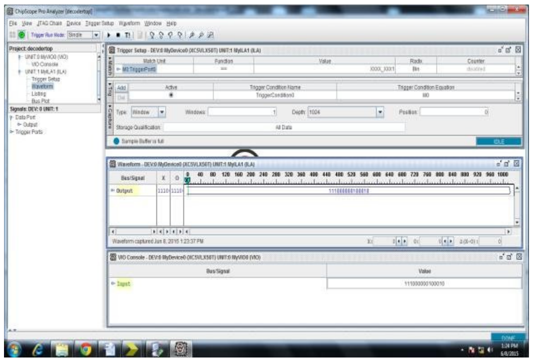

ChipScope Pro Analyzer [decodertop]
Trigger Setup - DEV:0 MyDevice0 (XC5VLX50T) UNIT:1 MyILA1 (ILA)
Waveform - DEV:0 MyDevice0 (XC5VLX50T) UNIT:1 MyILA1 (ILA)
output
111000000100010
VIO Console - DEV:0 MyDevice0 (XC5VLX50T) UNIT:0 MyVIO0 (VIO)
input
111000000100010

Figure11.FPGAsimulationresultofMLDDwithouterror

B. FPGAsimulation resultofMLDDwitherror

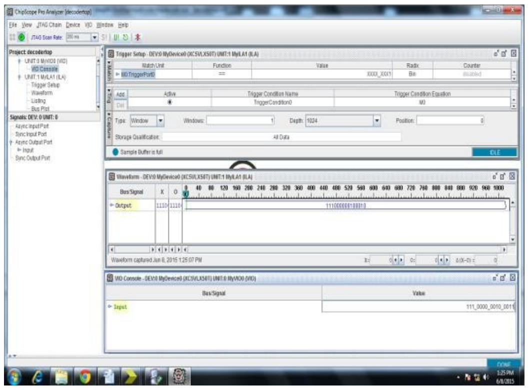

ChipScope Pro Analyzer [decodertop]
Trigger Setup - DEV:0 MyDevice0 (XC5VLX50T) UNIT:1 MyILA1 (ILA)
Waveform - DEV:0 MyDevice0 (XC5VLX50T) UNIT:1 MyILA1 (ILA)
output
111000000100010
VIO Console - DEV:0 MyDevice0 (XC5VLX50T) UNIT:0 MyVIO0 (VIO)
input
111_0000_0010_0011

Figure12.FPGAsimulationresultofMLDDwitherror

VII. CONCLUSION

ThepaperfocusesonthedesignofaMajorityLogicDecoder/Detector(MLDD)forfaultdetectionalongwithcorrectionoffault,suitableformemoryapplications,withreduced faultdetectiontime.

From the simulation results, (A codeword of size 15 ischosen here for designing), when compared to the existingMLD,TheproposedMLDDhascomparativelylessdelayof
12.578 ns and can detect the presence of errors in just 3 cycleseven formultiple bitflips.

It has found that for error detection and correction (forcodeword of 15), when comparing to the existing technique, aspeed up of about 1100 ns is obtained when there is no errorsin data read access. It‟s because the fault detection needs onlythree cycles and after the detection of an error free condition,the codeword is issed to the output without further corrections.This is a great saving of time since most of the situations thememory read access does not make errors. Therefore there is aconsiderablereductioninthememoryaccesstime.

TheproposedMLDDhaveabout4%lowpowerconsumptionthantheexistingMLDtechnique,sincetheproposeddesigndetectsthefaultsinjustthreecycles.

Therefore a large no. of clock cycles (here 12 clock cycles) aresavedandhenceconsiderablereductioninpower isachieved.

MLDDerrordetectorisdesignedasitisindependentofthe code word size and inference about area is that for largevalues of code word size, the area overhead of the MLDDactually decreases with respect to the plain MLD technique.i.e.,forlargevaluesofcodewordsizebothareasarepracticallythesame.ThereforetheproposedMLDDwillbeanefficientdesignforfaultdetectionand correction.

REFERENCES

[1] PedroReviriego,JuanA.Maestro,andMarkF.Flanagan,” Error Detection in Majority Logic Decodingof Euclidean Geometry Low Density Parity Check (EG-LDPC) Codes” IEEE Trans. Very Large Scale Integration(VLSI) Systems, Vol.21, No.1,January2013.

[2] R.C.Baumann,“Radiation-inducedsofterrorsinadvancedsemiconductortechnologies,” IEEETrans.Device Mater. Reliab., vol. 5, no. 3, pp. 301–316, Sep.2005.

[3] M.A.Bajura,Y.Boulghassoul,R.Naseer,S.DasGupta, A.F.Witulski,J.Sondeen,S.D.Stansberry,J.Draper,L.W.Massengill,andJ.N.Damoulakis,“Modelsandalgorithmiclimitsforan ECC-basedapproachtohardening sub-100-nm

[4] SRAMs,” IEEE Trans. Nucl. Sci., vol. 54, no. 4, pp. 935–945,Aug. 2007.

[5] R. Naseer and J. Draper, “DEC ECC design to improvememory reliability in sub-100 nm Technologies,” Proc.IEEEICECS, pp. 586–589,2008.

[6] S..GhoshandP.D.Lincoln,“Dynamiclow-densityparity check codes for fault-tolerant nano-scale memory,”presentedattheFoundationsNanosci.(FNANO),Snowbird,Utah, 2007.

[7] S. Ghosh and P. D. Lincoln, “Low-density parity checkcodesforerrorcorrectioninnanoscalememory,”SRIComputerScienceLab.,MenloPark,CA,Tech.Rep.CSL-0703,2007.

[8] H.NaeimiandA.DeHon,“Faultsecureencoderanddecoderformemoryapplications,”inProc.IEEEInt.Symp.DefectFault Toler.VLSISyst.,2007,pp.409–417.

[9] B. Vasic and S. K. Chilappagari, information theoreticalframeworkforanalysisanddesignofnanoscalefault-tolerantmemoriesbasedonlow-densityparity-checkcodes,” IEEE Trans. CircuitsSyst. I, Reg. Papers, vol. 54,no.11, Nov. 2007.

[10] H.NaeimiandA.DeHon,“Faultsecureencoderanddecoder for nanomemory applications,” IEEE Trans. VeryLarge Scale Integr. (VLSI) Syst., vol. 17, no. 4, pp. 473–486,Apr. 2009.

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

[12] S. Liu, P. Reviriego, and J. Maestro, “Efficient majoritylogic fault detection with difference-set codes for memoryapplications,”IEEETrans.VeryLargeScaleIntegr.(VLSI) Syst., vol. 20,no.1, pp.148–156,Jan. 2012.