International Journal of Ethics in Engineering & Management Education
Website: www.ijeee.in (ISSN: 2348-4748, Volume 2, Issue 5, May 2015)

Majority Logic Fault Detection with Difference Set Cyclic Code and LDPC Codes for Memory Applications

KUBRA BEGUM[1]
M.Tech student, VLSI Design & Embedded Systems
APPA Institute of Engineering & Technology
Gulbarga, Karnataka, India.
kubrafshaik@gmail.com

MAHESH R.K[2]
Assistant Professor, Dept. of E&CE, VLSI Design & Embedded System APPA Institute of Engineering & Technology Gulbarga, Karnataka, India.
rkmahesh10@gmail.com

***ABSTRACT*- Due to higher integration densities, parameter variations and technology scaling the failures to performance may occur for every practical applications. this must be protected with effective error correction codes. An advanced error correction codes are used when an additional protection is needed. The project work deals with the idea of majority logic fault detection and correction technique using DS-LDPC codes with the application focused on memories. the majority logic decoder/detector codes is used because of their capability to correct large number of soft errors. Even though MLD consumes large time, this can be overcome by the proposed method which detects the errors in less cycle time. The proposed technique significantly reduces memory access time and also it takes three iterations instead of N iterations when there is no error in the data read and the MLD itself use as a fault detector which improves the performance and also achieves high data rate while minimizing the area of the majority gate using sorting network.**

Index Terms: ECC, majority logic decoder/detector (MLDD), difference set cyclic codes, memory, sorting network, soft errors.

I. INTRODUCTION

In the modern digital system design memories are considers as an essential parts due to their reliability and security. To protect memories from so–called soft errors[1]. the error correction codes are commonly used. Soft error is caused due to radiation event without damaging the circuit/device, which causes enough disturbance of charge to change the logic value of memory cell[2].the soft error is also called as single event upsets(SEUs).A multi bit upsets(MBU) is created when high energy of radiation event may be affects more than a single bit. Memories are processed by which information to be encoded, stored and retrieved. While retrieving encoded information should be uncorrupted. So it is important to prevent memory against soft-error.

A) Memory Mitigation Technique:

1) Triple Modular Redundancy(TMR)

TMR is a special case of von-neumann method. It consist of mainly three versions of the design in parallel, by selecting the correct output using a majority voter[3]. The complexity overhead would be three times adds the complexity of the majority voter and thus increasing the power consumption.

2) Error Correction Codes(ECC)

ECC codes are the good way to mitigate memory soft errors. when there is a low soft error rate for terrestrial radiation environments, the best solution is to use codes like SEC-DED[3] because of their low encoding and decoding complexity. However as a consequence of higher integration densities, there is an increase in number of soft errors, which causes the need for higher error correction capabilities [4].

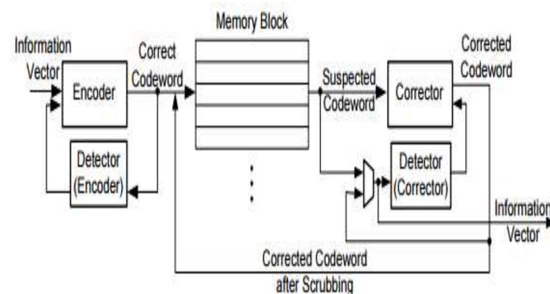

Figure1.overview of fault corrector architecture for the proposed technique

II PREVIOUS WORK

Majority logic decoding(MLD) consist of number of parity check equations and are orthogonal to each other ,so that, each codeword participates in only one parity equation for each iteration leaving very first bit which adds to all equations. Because of these reason ,the outcome of majority of these parity check equations defines the correctness of the current bit under decoding. Two type of decoder is implemented as Type I ML decoder & Type II ML decoder.

A) plain ML decoder

The ML decoder is a simple, power full decoder and able to correct multiple random bit flips based on the number

IJEEE

International Journal of Ethics in Engineering & Management Education

Website: www.ijeee.in (ISSN: 2348-4748, Volume 2, Issue 5, May 2015)

of parity checking equations[1]. it is mainly consist of four parts: a cyclic shift register, an XOR matrix, a majority gate, and an XOR for correcting the codeword bit under decoding. Initially the input signal is stored into cyclic shift register and then shifted through all the N taps, by taking intermediate values in each tap are used to find the results{Bj} of the check sum equations from the XOR matrix. In the Nth decoding cycle, the output signal is produced which is decoded version of the input, after the result reached the final tap.

B) Plain MLD with SFD

An alternative designs may be used to improve the decoder performance .there is one possibility is to add a fault detector[1] to decode only faulty code words by calculating the syndrome. the implementation of an SFD reduces the average latency of the decoding process, it also adds design complexity as shown in the figure2.

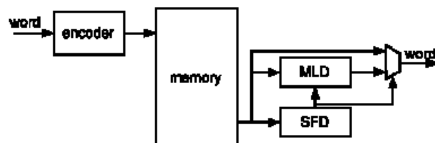

Figure 2.Schematic of an ML decoder with SFD

III. PROPOSED WORK

Here in the present paper a serial corrector and fault detector is implemented. By designing a majority gate using sorting network we have proved that the fault detector designed with serial corrector provides error detection in the three decoding cycles. By using a The following hypothesis[7] is made for the proposed technique: "Given a word that reads by a memory protected with DSCC codes and it will be moved upto five bit flips and all errors can be detected with three decoding cycles only". This is a big advancement over the uncomplicated case where N decoding cycles are required to ensure that the errors are detected. Figure3 shows the general memory schematic of MLDD. Which is consist of an encoder, memory and MLDD.

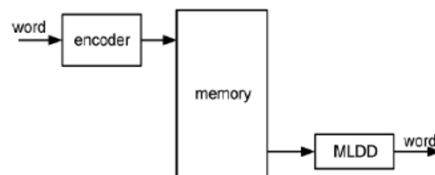

Figure3.A generic memory schematic of MLDD.

1) The Encoder

Figure 4 shows an example for encoder design for (15,7,5)EG-LDPC code. The information bits are forward to the encoder to encode the information vector. This section gives a brief introduction on linear block ECC's. The encoder circuitry consist of (n-k) XOR gates. An n bit codeword c which is used to encode a k bit information vector I is produced by multiplying the k bit information memory. The encoding operation for linear codes essentially performs the vector-matrix multiplication. C = [I x G] where G is a k x n generator matrix. The encoder vector includes two parts, the first one is information bits memory and second is the parity bits. Whereas each parity bit is made of an inner product of information vector and a column of X, from G = [I:X] where I is a k x k identity matrix and X is a k x (n-k) matrix that generates the parity bits. The encoder circuit[1] is used to compute the parity bits of the (63,37,26) EG-LDPC. The encoder vector c are (c0,c1,c2,…c25) bits are copied from the information vectors I are (i0,i1,i2,…i25)bits. The remaining of the encoded vector(c27,c28,…c62) is the parity bits and are linear sums (XOR) of the information bits.

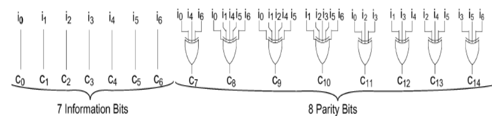

Figure4. structure of an encoder circuit for the (15,7,5)EGLDPC code

2) Fault Secure Detector:

The detector operation is used to generate the syndrome vector and the checking or detecting operation is given by $S = C.H^T$. The syndrome vector S is an (n-k) bit vector. every bit of the syndrome vector is obtained by the product of C with each row of H . if the syndrome S is zero then the C is a valid codeword else the C is errorneous [5]. An XOR gate is used to implement the binary sum of this product. the fault secure detector[6] design example for an (15,7,5) EG-LDPC code is shown in figure 5.

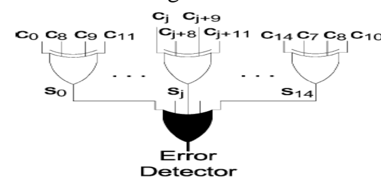

Figure 5. structure of an fault secure detector circuit for the (15,7,5)EG-LDPC

3) Memory Block

In order to prevent accumulation of too many errors in any memory word that surpasses the code correction capability and the system must performing memory scrubbing[3]. The process of periodically reading memory words from the memory block and correcting any potential faults[5] and finally write them back to the memory is the function of memory scrubbing. The normal memory access operation is kept constant for the periodic scrubbing operation to be carried out.

4) Serial Corrector

Figure6 shows the One step majority logic decoder or corrector .The serial majority takes n cycles to correct the codeword which contain errors. When there is a low fault rate found then the corrector block is used infrequently. the normal memory reads path will be placed off by the serial corrector. This is shown in the figure1 .the memory words are retrieved from the memory block and checked by the detector unit. When the detector detects an error the memory word is fed to the corrector block to be corrected that has a latency of the detector plus the n round of the corrector[4].

In this section we present the overall MLDD system in our pr

Figure6. one step serial corrector(15,7)

5) MLDD Design

Figure7 shows the schematic of the proposed MLDD for N-bit codeword .The extra hardware needs to perform the error detection is illustrated in the figure9 that is The control unit & output tri-state buffers.

Figure7.Schematic of proposed MLDD for N bit

Figure8. flow diagram of the MLDD algorithm

Figure9.Schematic of MLDD control unit

Figure8. shows the flowchart of MLDD algorithm. and the steps are given as Step1: initially the input signal is stored into the cyclic shift register and shifted throughout all the taps. Step2: the intermediate values in each tap are then used to find the results {Bj} of the checksum equation from the XOR matrix. Step3: the resulting sums {Bj} are then send to the majority gate(MG) for calculating its correctness. If the received number of 1's in {Bj} is larger than the number of zero's, that means the current bit under decoding is wrong and a signal triggered to correct it. Otherwise, the BUD [4]would be correct and there are no extra functions would be needed on it. Step4: in the next step, the contents of the shift registers are rotated and the above procedure is repeated till all codeword bits have been processed[6].

6) Majority Gate(MG)

The responsibility of the majority gate is to compute the parity checksums and to take decision about the correctness of the current bit under decoding(BUD). Conventionally, the implementation of the MG grows exponentially with the no of inputs added[4]. To avoid this complexity issue, producing a majority gate based on sorting networks.

7) Sorting Network

Figure10(a) & Figure10(b) shows the two bit sorter network. A sorting network consist of wires and comparators that will sort all possible inputs into ascending order in correct manner. So that it is used to reducing the gates and their interconnections of the majority gate.

Figure10 a) Comparature structure b) 2 bit sorter.

IV. SIMULATION RESULTS

The proposed designs are simulated in Xilinx 14.7 and the outputs are shown below. Experimental results shows the power, area and decoding time has been reduced .

Figure11. simulation waveform for encoder

Figure12. simulation waveform for1 step MLD(63,37,26)

IJEEE

International Journal of Ethics in Engineering & Management Education

Website: www.ijeee.in (ISSN: 2348-4748, Volume 2, Issue 5, May 2015)

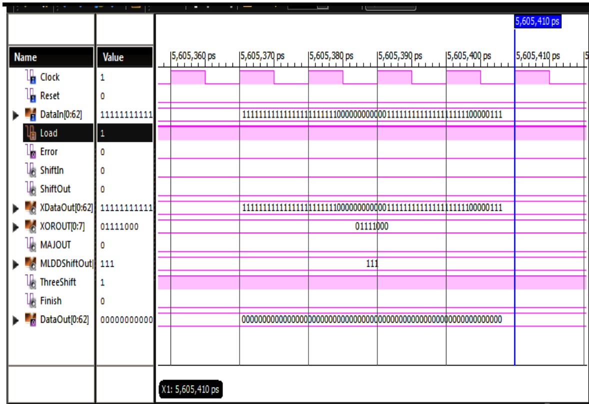

Figure13. Simulation result of MLDD(63,37,26).

V .RESULTS AND COMPARISON

Technique	Number of cycles needed			
	I/O	Error Detection N	Cycle at which the output obtained after detection process	
			Without error	With error
Plain MLD(Existing)	2	N (E.g.N=15)	N+2 (E.g.N=15+2)	N+2 (E.g.N=15+2)
Proposed MLDD	2	3	3+2=5	N+5

Table I comparison of no. of cycles needed for proposed and existing designs

Technique	Time at which the o/p obtained for error free codeword(ns)	Delay(ns)			Total power consumption(mw)
		Gate delay	Net delay	Total delay	
Existing MLD	3033	3.836	9.17	14.006	49
Proposed MLDD	1934	3.983	8.595	12.578	47

Table II comparison of speedup, delay and total estimated power consumption.

VI. CONCLUSION AND FUTURE WORK

In this paper the MLDD fault detector module has been designed in a way that is independent of the code size and This makes its area overhead less and power consumption is quite reduced compared with other approaches by the use of sorting network. And the extension to this work is done by implementing the majority logic decoder/detector using additional error detection logic such as BIST technique. Future work can be extended by replacing the conventional gates with reversible logic gates and use scrubbing method in order to reduce the conventional power consumption and message bits.

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, Sep2005.

[2]. M. A. Bajura, Y. Boulghassoul, R. Naseer, S. DasGupta, A. F.Witulski, J. Sondeen, S. D. Stansberry, J. Draper, L. W. Massengill, and J. N. Damoulakis, "Models and algorithmic limits for an ECC-based approach to hardening sub-100-nm SRAMs," IEEE Trans. Nucl. Sci., vol. 54, no. 4, pp. 935–945, Aug. 2007.

[3]. R. Naseer and J. Draper, "DEC ECC design to improve memory reliability in sub-100 nm technologies," Proc. IEEE ICECS, pp. 586–589, 2008.

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

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

[6]. H. Naeimi and A. DeHon, "Fault secure encoder and decoder for memory applications," in Proc. IEEE Int. Symp. Defect Fault Toler. VLSI Syst., 2007, pp. 409–417.

[7]. B. Vasic and S. K. Chilappagari, "An information theoretical framework for analysis and design of nanoscale fault-tolerant memories based on low-density parity-check codes," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 54, no. 11, pp. 2438–2446, Nov. 2007.