Fault Detection with Error Correction Codes for Memory Applications

K. Sushmaja[1] , Dr. Fazal Noorbasha[2]

[1] (M. Tech student, Department of Electronics and Communication Engineering, K L University, Vaddeswaram, Guntur Dt.)

[2](Assoc. Prof, Department of Electronics and Communication Engineering, K L University, Vaddeswaram, Guntur Dt.)

ABSTRACT

The paper deals with error detection method for difference set cyclic codes with majority logic decoding. These are suitable for memory applications due to their capability to correct large number of errors. The proposed Majority Logic decoder itself detects failures which minimize area overhead and power consumption. The memory access time is also reduced when there is no error in the data read. The proposed circuit is simulated in DSCH, Cadence- Virtuoso, and Xilinx. The results obtained in various tools are presented in this paper.

Keywords – Difference set cyclic codes, Error Correction Codes (ECC), Majority Logic decoder

I. INTRODUCTION

Memory applications should be protected from all kinds of faults for reliable performance. Those faults can be detected using Error Correction Codes (ECC). When digital data is stored in a memory, it is crucial to have a mechanism that can detect and correct a certain number of errors. ECC encodes data in such a way that a decoder can identify and correct certain errors in the data. Usually data strings are encoded by adding a number of redundant bits to them. When the original data is reconstructed a decoder examines the encoded message, to check for any errors.

1.1 Error Correction Codes (ECC)

There are 2 basic types of Error Correction Codes:

- Block codes, which are referred to as (n,k) codes.
- Convolution codes, where the code words produced depend on both the data message and a given number of previously encoded messages.

Among the ECC codes that meet the requirements of higher error correction capability and low decoding complexity, cyclic block codes have been identified as good one, due to their property of being Majority Logic (ML) decodable. LDPC (Low Density Parity Check) codes belong to the family of ML decodable codes. Difference Set Cyclic Codes (DSCC), one specific type of LDPC codes are used here. DSCCs are one step ML decodable codes with high error correction capability and are linear cyclic block codes.

1.2 DSCC

DSCCs are one-step ML decodable codes with high error-correction capability and are linear cyclic block codes[5].

1) Perfect Difference Set: DSCCs work on the difference-set concept for which a brief description follows. Given a set and a difference of the elements, we have

$P = \{l_0, l_{1 \dots q\}}$ $(0<= l_1 < l_2 \dots\dots <l_q \ <=(q+1))$

$D = \{l_i - l_j; i=! j\}$

2) DSCC Construction: For a binary code, the perfect difference-set is constructed using the relationship

$q= 2^s \ : s \varepsilon N$

3) DSCC Parameters: Besides from the definitions and equations previously explained, the following parameters completely define the DSCC codes:

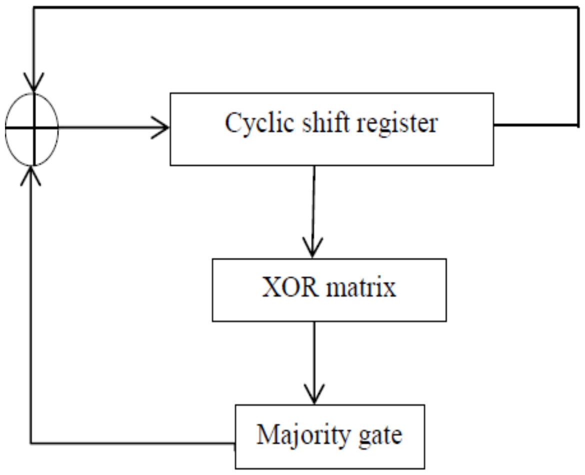
- Code length: $2^{2s} + 2^s + 1$
- Message bits: $2^{2s} + 2^s + 3^s$
- Parity-check bits: $3^s + 1$
- Minimum distance: $d = 2^s + 2$

II. MAJORITY LOGIC DECODER (MLD)

MLD is based on a number of parity check equations which are orthogonal to each other, so that, at each iteration, each code word bit only participates in one parity check equation, except the very first bit which contributes to all equations. For this reason, the majority result of these parity check equations decide the correctness of the current bit under decoding.

ML decoder is a simple and powerful decoder, capable of correcting multiple random bit-flips depending on the number of parity check equations. It consists of four parts:

1) a cyclic shift register
2) an EXOR matrix
3) a majority gate
4) an XOR for correcting the code word bit under decoding.



In this paper, 6 D- Flip flops are considered. Majority gate is used to detect whether if number of 1's > number of 0's are not. XOR gates are used for generating b0, b1, b2. These b0, b1, b2 are inputs to majority gate. If number of 1's> number of 0's, information vector and output vector will vary. So, to correct it, output of majority gate and output of first flip-flop are XORed and it is given as input to multiplexer of first flip-flop.

In this proposed decoder, the majority gate is circuit that calculates carry equation in full adder. So, the majority gate consists of 3 AND gates and 1 EXOR gate. The Proposed circuit is designed in DSCH for verifying functional correctness. The same circuit is implemented using Verilog code in Xilinx. For calculating power dissipation values, cadence tool has been chosen. The schematics have been implemented in cadence – Virtuoso and corresponding waveforms and power dissipation values have been observed. All the simulation results are presented in this paper.

III. SIMULATION RESULTS

3.1 DSCH schematics and results

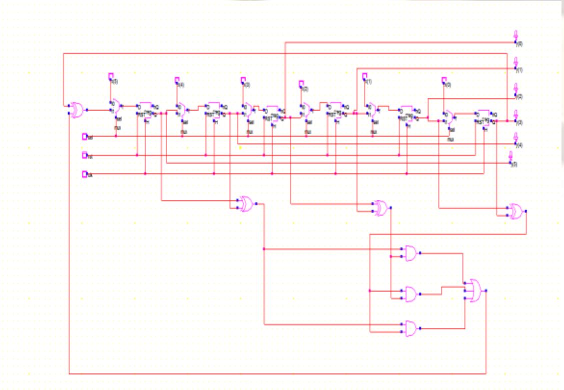


Fig1. ML decoder in DSCH

When input is 110000

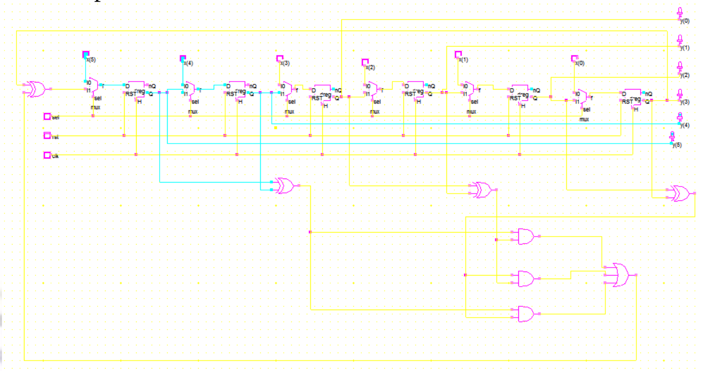


Fig.2. Error free decoded output

When input is 010101

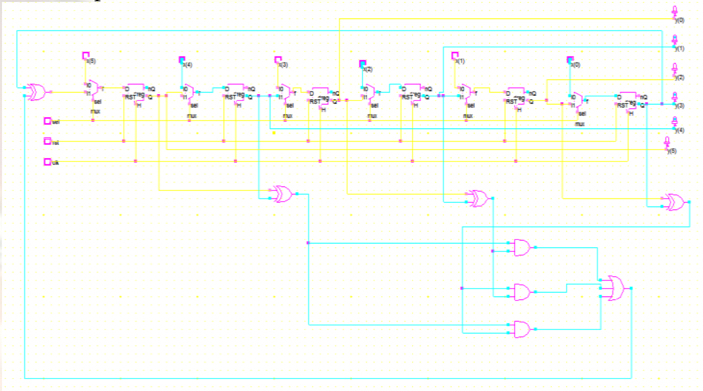


Fig3. Error in decoded output

In this DSCH tool, the correct path is displayed in one colour and the path with fault is displayed in another colour so that we can differentiate faulty paths easily.

3.2 Schematics in Cadence

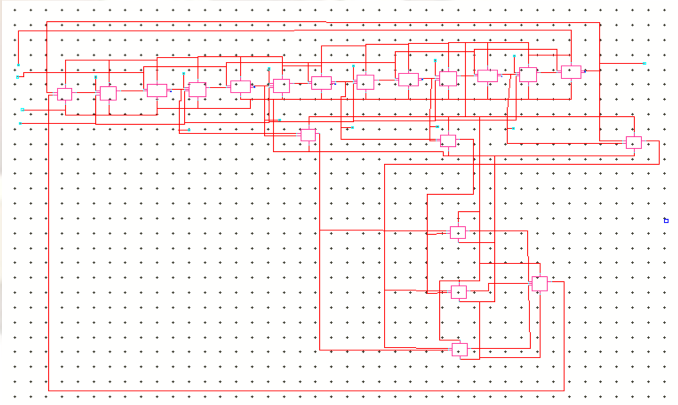


Fig4. ML decoder schematic in Cadence- Virtuoso

In cadence, to create the entire schematic, initially it is necessary to create symbols for each and every block in the circuit. In ML decoder, the main blocks that exist are D- Flip flop, Multiplexer, 2- input XOR gates, 2- input AND gates, 3- input OR gates. So, if we create schematics for those blocks then the respective symbols can be instantiated in the main module.

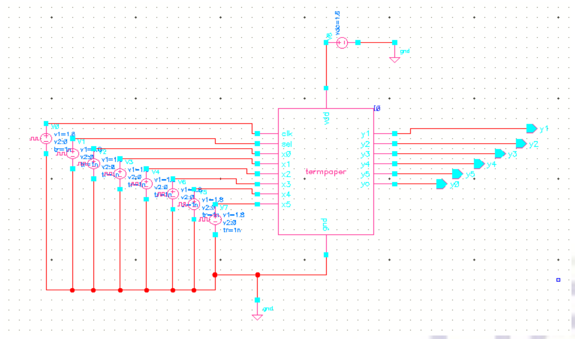


Fig5. ML decoder symbol in Cadence- Virtuoso

The following are the output waveforms in Cadence:

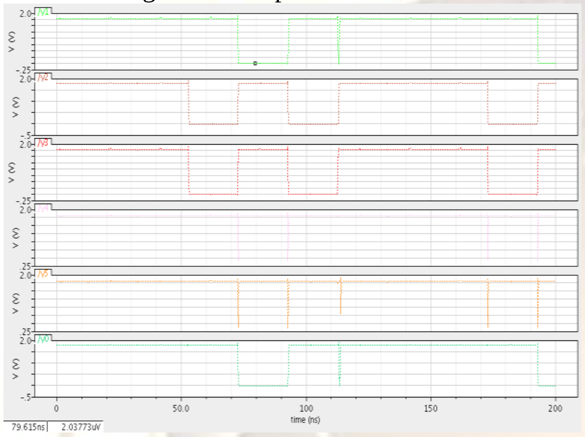


Fig6. Output waveform in cadence

Power Dissipation Values

3- input OR gate	: 32.4249 pW
2-1 Multiplexer	: 1.86006 nW
D- Flip flop	: 935.898 pW
2- input XOR gate	: 1.27603 nW
2- input AND gate	: 653.021 pW
ML decoder circuit	: 1.78214 nW

Delay in ML decoder circuit is 20.41E-9

Power dissipation values for the each module can be known while generating its transient analysis. Log file of the particular module contains its power dissipation value.

3.3 Simulation results of Xilinx

Verilog coding is defined for the proposed circuit. Two different modules i.e., error free and error existing modules are defined in the main module. Results are observed in Xilinx ISE simulator.

When sel=0 and correct data vector

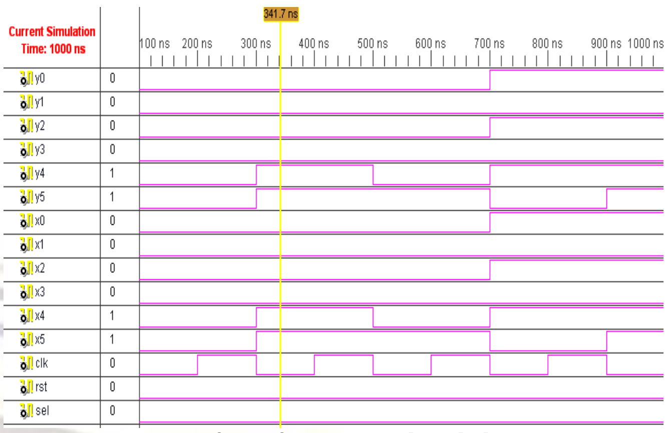


Fig6. Output waveform for correct decoded vector when sel=0

After observing the result for correct decoded vector, error is induced in the decoded data vector and the error can be detected when its circuit is simulated.

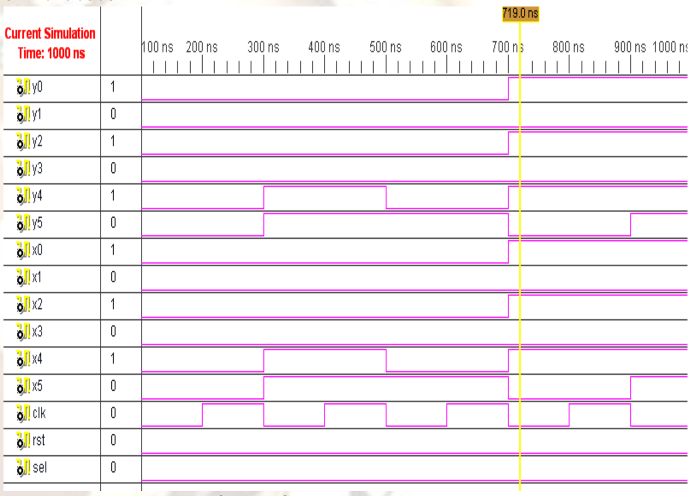


Fig7. Output waveform for wrong decoded vector when sel=0

IV. Conclusion

In this paper, a fault-detection mechanism, MLD, has been presented based on ML decoding using the DSCCs. Using this mechanism, the decoding and detection of data vectors can be done simultaneously. This circuit can be implemented in various tools and compare the results. Power dissipation values of different modules in this circuit can be obtained when it is implemented in Cadence.

Acknowledgements

I am grateful to **Prof. Dr. Fazal Noorbasha, K L University** for his constant encouragement and I whole heartedly thank him for his support in completion of this paper successfully. I would like to express my heartfelt thanks to my beloved parents for their blessings and their wishes that helped me in successful completion of this report.

REFERENCES

[1] Shih-Fu Liu, " Efficient Majority Logic Fault Detection With Difference-Set Codes for Memory Applications", IEEE transactions on very large scale integration (VLSI) systems, vol. 20, no. 1, january 2012

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

[3] S. Reed, "A class of multiple-error-correcting codes and the decoding scheme," *IRE Trans. Inf. Theory*, vol. IT-4, pp. 38–49, 1954.

[4] H. Naeimi and A. DeHon, "Fault secure encoder and decoder for NanoMemory applications," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 17, no. 4, pp. 473–486, Apr. 2009.

[5] Y. Kato and T. Morita, "Error correction circuit using difference-set cyclic code," in *Proc. ASP-DAC*, 2003, pp. 585–586.