Architecture with reduced Latency And Complexity For Matching of Data Encoded With Hard Systematic Error- Correcting Codes

Josef Patrico Rajakumar ,P.G Scholar
Electronics Communication Engineering
Sri Shakthi Institute of Engineering
Coimbatore

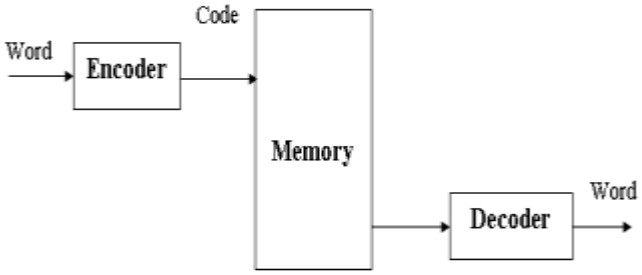
Dinesh.S , Assistant Professor
Electronic Communication Engineering
Sri Shakthi Institute of Engineering
Coimbatore

Abstract

Most of the Memory cells have been protected from soft errors; due to the increase in soft error rate in logic circuits, the encoder and decoder circuitry around the memory blocks have become susceptible to soft errors as well and must also be protected with effective error correction codes. The reliability and security of systems protected by linear codes largely depend on the accuracy of the expected error model . To avoid a high decoding complexity, the use of one step majority logic decodable codes was first proposed for memory applications. Majority logic decodable (MLD) codes are suitable for memory applications because of their capability to correct large number of errors. Here Euclidean Geometry Low-Density Parity-Check (EG-LDPC) codes are used for error correction, because of their fault-secure detector capability. The proposed Majority Logic Decoder/Detector (MLDD) 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 algorithm is coded in VHDL and simulated using ModelSim and Xilinx ISE 8.1isimulator. The results obtained are compared with the existing version of the technique.

I.INTRODUCTION

Memories are critical elements in today's digital systems. Various types of memories are widely used in many different reliable and secure applications and appear in nearly all digital devices. As the technologies move into nano realm, the reliability of memories is largely compromised by the increased rate of multi-bit errors. In secure applications, the security of memories and the whole system is threatened by side-channel attacks such as fault injection attacks. Here Euclidean Geometry Low-Density Parity-Check (EG-LDPC) codes are used for error correction, because of their fault-secure detector capability, which concentrate their error detection and correction capabilities against certain types of errors, e.g. errors with small multiplicities. The reliability and security of systems protected by linear codes largely depend on the accuracy of the expected error model. For applications where the error model is hard to predict or the multi-bit error rate is high, traditional protection methods based on linear codes are not sufficient.



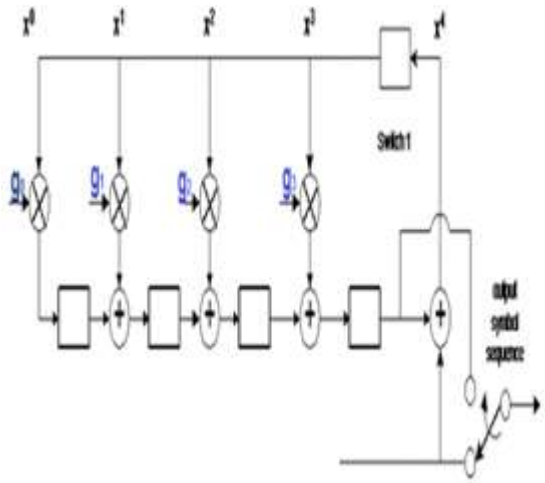
Cyclic block codes have the property of being majority logic (ML) decodable. Therefore cyclic block codes have been identified as more suitable among the ECC codes that meet the requirements of higher error correction capability and low decoding complexity.

A. SOFT ERRORS

The reliability and security of memories are essential considerations in the modern digital system design . Soft error occurs when a radiation event causes enough of a charge disturbance to reverse or flip the data state of a memory cell, register. As technology scales, memory devices become larger and more powerful error correction codes are needed to protect memories from soft errors. The error is 'soft' because it will change the logic value of memory cells without damaging the circuit/device.

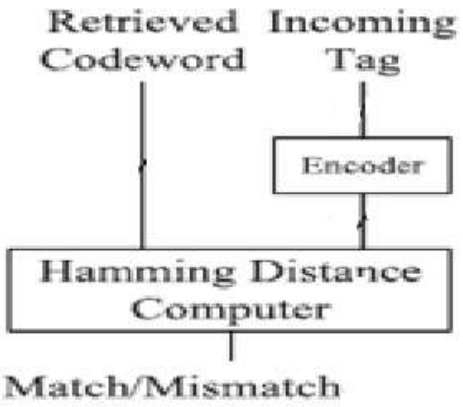
B. BCH CODE:

The generator polynomial of the code is specified in terms of its roots over the Galois field GF (2m). Let α be a primitive element in GF (2m).The generator polynomial g(x) of the code is the lowest degree polynomial over GF (2). Let mi(x) be the minimum polynomials of αi, then generator polynomial G(x) . In this work n=15, k=7 and t=2 is considered. Hence the generator Polynomial with α, α2,α3,α4 as the roots is obtained by multiplying the following minimal polynomials.



C.LOW DENSITY PARITY CHECK CODES (LDPC):

A low-density parity-check (LDPC) code is a linear error correcting code, a method of transmitting a message over a noisy transmission channel, and is constructed using a sparse bipartite graph. Low-density parity-check (LDPC) codes are forward error-correction codes. LDPC codes are capacity-approaching codes, which means that practical constructions exist that allow the noise threshold to be set very close (or even arbitrarily close on the BEC) to the theoretical maximum (the Shannon limit) for a symmetric memory-less channel.



Assuming that the incoming address has no errors, we can regard the two tags as matched if *d* is in either the first or the second ranges. In this way, while maintaining the error-correcting capability, the architecture can remove the decoder

from its critical path at the cost of an encoder being newly introduced. Note that the encoder is; in general, much simpler than the decoder, and thus the encoding cost is significantly less than the decoding cost. Since the above method needs to compute the Hamming distance, presented a circuit dedicated for the computation. The circuit first performs XOR operations for every pair of bits in *X* and *Y* so as to generate a vector representing the bitwise difference of the two codewords.

$$S = \begin{bmatrix} 0000010001011110 \end{bmatrix} \begin{bmatrix} 10001011 \\ 11001110 \\ 01100111 \\ 10111000 \\ 01011100 \\ 00101110 \\ 00010111 \\ 10000000 \\ 01000000 \\ 00100000 \\ 00010000 \\ 00001000 \\ 00000100 \\ 00000010 \\ 00000001 \end{bmatrix}$$

S = 0000 0000 .

Hence received codeword is valid code word.

Original Codeword = (000 0010 0010 1110)

i For 1st cycle, Error Codeword = (000 0011 0010 1111),

ii 1st xor gate inputs: 0111,Output: 1

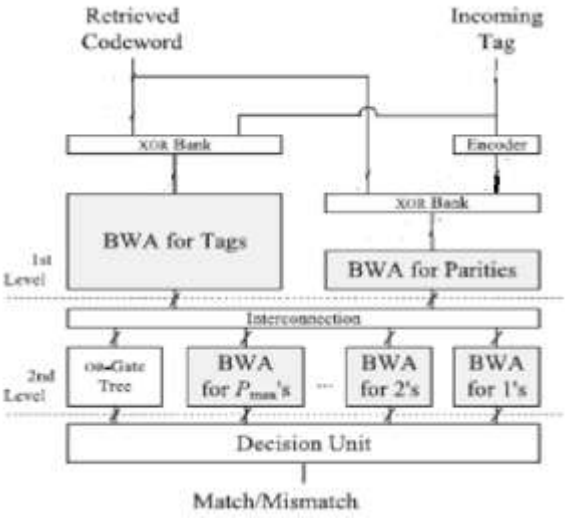
iii 2nd xor gate inputs: 0111,Output:1

iv 3rd xor gate inputs: 0011, Output: 0

v 4th xor gate inputs: 0001,Output:1

II.ARCHITECTURE FOR COMPUTING THE HAMMING DISTANCE

The proposed architecture grounded on the data-path design s. It contains multiple butterfly-formed weight accumulator (BWAs) proposed to improve the latency and complexity of the Hamming distance computation. The basic function of the BWA is to count the number of 1's among its input bits. It consists of multiple stages of HA, where each output bit of a HA is associated with a weight.



In other words, both inputs of a HA in a stage, except the first stage, are

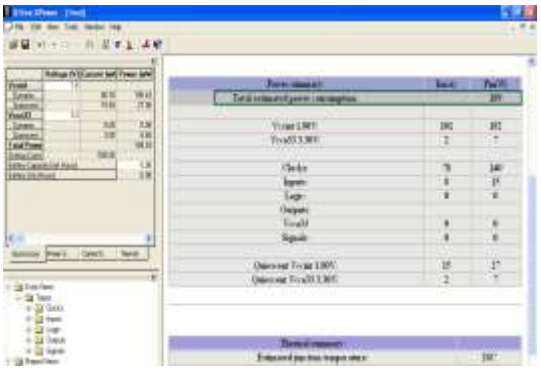
either carry bits or sum bits computed in the upper stage. This connection method leads to a property that if an output bit of a HA is set, the number of 1's among the bits in the paths reaching the HA is equal to the weight of the output bit, for example, if the carry bit of the gray-colored HA is set, the number of 1's among the associated input bits, i.e., A, B, C, and D, is 2. At the last stage, the number of 1's among the input bits

A.EG-LDPC CODES

Euclidean Geometry codes based on the lines and points of the corresponding finite geometries. Euclidean Geometry codes are also called EG-LDPC codes based on the fact that they are low-density parity-check (LDPC) codes. LDPC codes have a limited number of 1's in each row and column of the matrix; this limit guarantees limited complexity in their associated detectors and correctors making them fast and light weight

B.RESULT ANALYSIS OF BWA ALGORITHM:

The existing one step MLD technique is simulated for both error free and erroneous conditions during memory access, and the results for error free condition

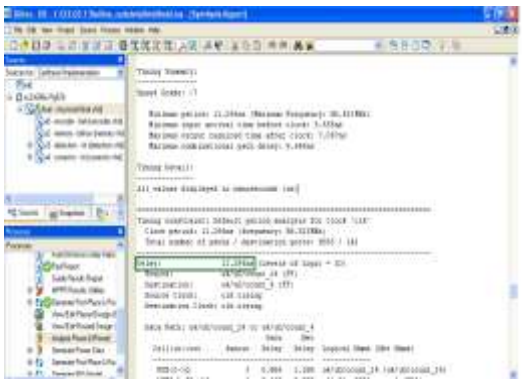


The proposed method BWA designs is been compared with the existent method plain ML decoder (BWA). For the detection of errors, MLD always needs N+2 cycles in all cases. The proposed design just requires three cycles to detect any error (plus two of I/O).

C.SYNTHESIS REPORT FOR BWA :

The proposed MLDD design is

simulated using ModelSim and Xilinx. The power consumed and memory read access delay is calculated. Synthesis report for delay measurement . Proposed BWA have less delay when compared to existing MLD, since the proposed design detects the faults in just three cycles.



Proposed BWA have less delay when compared to existing MLD, since the proposed design detects the faults in just three cycles of the delay comparison. When the code word does not suffer from errors, it can come out in the next 4th cycle itself without further shifting. Therefore this is a great advantage for BWA in terms of delay and performance

III. CONCLUSION

Memories are the most universal component today. They are prone to errors like soft and transient errors. The proposed work focuses on the design of a Majority Logic Decoder/Detector (MLDD) for fault detection along with correction of fault, suitable for memory applications, with reduced fault detection time, when comparing to the existing technique, memory access delay is reduced; when there is no errors in data read access. It's because the fault detection needs only three cycles and after the detection of an error free condition, the codeword is passed to the output without further corrections. The proposed MLDD have less power consumption than the existing MLD technique, since the proposed design detects the faults in just three cycles. Therefore a large no. of clock cycles (here 12 clock cycles) are saved and hence considerable reduction in power is achieved.

REFERENCES

[1] 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.

[2] J. D. Warnock, Y.-H. Chan, S. M. Carey, H. Wen, P. J. Meaney, G. Gerwig,

H. H. Smith, Y. H. Chan, J. Davis, P. Bunce, A. Pelella, D. Rodko,

P. Patel, T. Strach, D. Malone, F. Malgioglio, J. Neves, D. L. Rude,

and W. V. Huott "Circuit and physical design implementation of the

microprocessor chip for the zEnterprise system," IEEE J. Solid-State

Circuits, vol. 47, no. 1, pp. 151–163, Jan. 2012.

[3] AMD Inc. (2010). Family 10h AMD Opteron Processor

Product Data Sheet, Sunnyvale, CA, USA [Online].

[4] Gerwig,H. H. Smith, Y. H. Chan, J. Davis, P. Bunce, A. Pelella, D. Rodko,

P. Patel, T. Strach, D. Malone, F. Malgioglio, J. Neves, D. L. Rude,

and W. V. Huott "Circuit and physical design implementation of the

microprocessor chip for the zEnterprise system," *IEEE J. Solid-State*

Circuits, vol. 47, no. 1, pp. 151–163, Jan. 2012.

[5] H. Ando, Y. Yoshida, A. Inoue, I. Sugiyama, T. Asakawa, K. Morita,

T. Muta, and T. Motokurumada, S. Okada, H. Yamashita, and Y. Satsukawa,

"A 1.3 GHz fifth generation SPARC64 microprocessor," in *IEEE*

ISSCC. Dig. Tech. Papers, Feb. 2003, pp. 246–247.

[6] M. Tremblay and S. Chaudhry, "A third-generation 65nm 16-core

32-thread plus 32-scout-thread CMT SPARC processor," in *ISSCC. Dig.*

Tech. Papers, Feb. 2008, pp. 82–83.

[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.

[8] 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.

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

[10] S. Liu, P. Reviriego, and J. Maestro, "Efficient majority logic fault detection with difference-set codes for memory applications," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 20, no. 1, pp. 148–156, Jan. 2012.

[11] H. Tang, J. Xu, S. Lin, and K. A. S. Abdel-Ghaffar, "Codes on finite geometries," *IEEE Trans. Inf. Theory*, vol. 51, no. 2, pp. 572–596, Feb. 2005.