ISSN 2277-2685

IJESR/Jul-Sep. 2024/ Vol-14/Issue-3/516-523

Ravali Sailla *et. al.,* / International Journal of Engineering & Science Research

DESIGN AND SIMULATION OF CRC ENCODER AND DECODER USING VHDL

1 Ravali Sailla, 2 Dr. L.Jagadeesh Naik

1 Pg Scholar , **Ece,** Holy Mary Institute Of Technology And Science

2 (Associate Professor) , **Ece,** Holy Mary Institute Of Technology And Science

ABSTRACT *Traditionally, memory cells were the only circuitry susceptible to transient faults. The supporting circuitries around the memory were assumed to be fault-free. 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 be protected. In this paper a new approach to design fault-secure encoder and decoder circuitry for memory designs. The key novel contribution of this paper is identifying and defining a new class of error-correcting codes whose redundancy makes the design of fault secure detectors(FSD) particularly simple. We further quantify the importance of protecting encoder and decoder circuitry against transient errors, illustrating a scenario where the system failure rate(FIT) is dominated by the failure rate of the encoder and decoder.*

Prove that Euclidean Geometry Low-Density Parity-Check(EG-LDPC) codes have the fault secure detector capability. Using some of the smaller EG-LDPC Codes, we can tolerate bit or nanowire defect rates of 10% and fault rates of 10^-18 upsets, achieving a FIT rate at or below one for the entire memory system and a memory density of10^11 bit/cm2 with nanowire pitch of 10nm for memory blocks of 10Mb or longer. Larger EG-LDPC codes can achieve even higher reliability and lower area overhead.

1.INTRODUCTION

Memory systems are protected against transient upsets of data bits using ECCs. Hamming codes are often used in today's memory systems to correct single error and detect double errors in any memory word. In these memory architectures, only errors in the memory words are tolerated and there is no preparation to tolerate errors in the supporting logic (i.e. encoder and corrector).

However combinational logic has already started showing susceptibility to soft errors, and therefore the encoder and decoder (corrector) units will no longer be immune from the transient faults. Furthermore, memory system designed with nanotechnology devices are expected to experience even higher transient fault rate therefore, protecting the memory system support logic implemented with nanotechnology devices is even more important. Here we proposed a fault tolerant memory system that tolerates multiple errors in each memory word as well as multiple errors in the encoder and corrector units. We illustrate using Euclidean Geometry codes and Projective Geometry codes to design the above fault-tolerant memory system, due to their well-suited characteristics for this

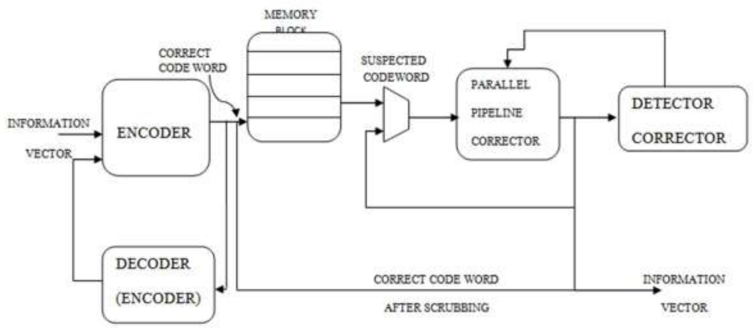
application,
which include:

1) Memory applications require low latency encoders and decoders.

2) These codes allow us to design a fault tolerant error-detector unit that detects any error in the received code-vector despite having faults in the detector circuitry; in other words we
can design a fault-secure detector for these codes

We use the fault secure detector unit to check the output vector of the encoder and corrector circuitry, and if there is any error in the output of either of these units, that unit has to redo the operation to generate the correct output vector. Using this detect-and-repeat technique we can correct potential transient errors in the encoder or corrector output and provide fault-tolerant memory system with fault-tolerant supporting circuitry.

2.PROPOSED SYSTEM:



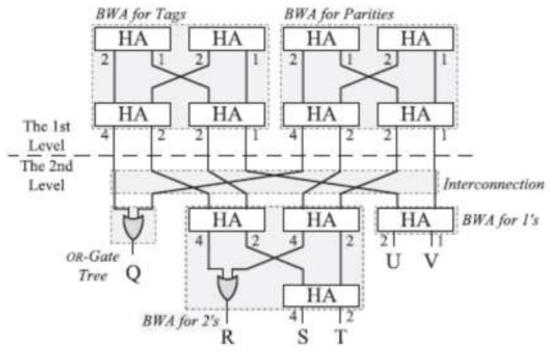
The above shown figure is about fault secure encoder in this fault secure encoder the reception of data is 100% of transmitted data

When we see the block diagram the information vector is firstly given to a encoder that encodes the information and it has a detector circuit which detect for errors and it consist of a memory block for storing of information and it also consist of parallel pipline corrector which is used to transmit the data and the data from it is given to the block of suspectedcode word and the final block is detector block and the output is take from parallel pipe line corrector.

2.2.1.Introduction:

Memory systems are protected against transient upsets of data bits using ECCs. Hammingcodes are often used in today's memory systems to correct single error and detect doubleerrors in any memory word. In these memory architectures, only errors in the memorywords are tolerated and there is no preparation to tolerate errors in the supporting logic(i.e. encoder and corrector).However combinational logic has already started showing susceptibility to soft errors,and therefore the encoder and decoder (corrector) units will no longer be immune fromthe transient faults. Furthermore, memory system designed with nanotechnology devicesare expected to experience even higher transient fault rate [3] [5]; therefore, protectingthe memory system support logic implemented with nanotechnology devices is even moreimportant. Here we proposed a fault tolerant memory system that tolerates multiple errorsin each memory word as well as multiple errors in the encoder and corrector units.

We illustrate using Euclidean Geometry codes and Projective Geometry codes to designthe above fault-tolerant memory system, due to their well-suited characteristics for this application,which include:



1) Memory applications require low latency encoders and decoders.

2) These codes allow us to design a fault tolerant error-detector unit that detects any errorin the received code-vector despite having faults in the detector circuitry; in other words wecan design a fault-secure detector for these codes . We use the fault secure detector unit to check the output vector of the encoder and
corrector circuitry, and if there is any error in the output of either of these units, that unit
has to redo the operation to generate the correct output vector. Using this detect-and-repeatechnique we can correct potential transient errors in the encoder or corrector output andprovide fault-tolerant memory system with fault-tolerant supporting circuitry.

2.2.2.System Overview:

In this section we outline the memory system which can tolerate errors in any part of the system, including the storage unit, encoder, corrector, and detector circuitry. Let E be the maximum number of error bits that the code can correct and D be the maximum number of error bits that it can detect, and in one error combination that strikes the system let ee, em, and ec be the number of errors in encoder, memory word, and corrector. In existing designs, the system would guarantee error correction as long as em _ E and ee = ec = 0. In contrast, here we guarantee that the system can correct any error combination as long as em _ E, ee + ede _ D, and em + ec + edc _ D, where ede and edc are the number of errors in two separate detectors, monitoring the encoder and corrector units. This design is feasible when the following two fundamental properties are satisfied:

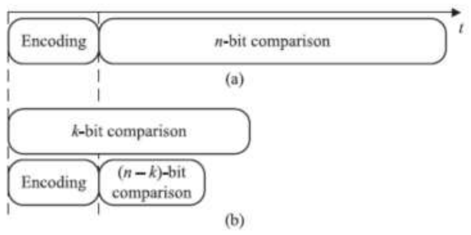
1) Any single error in the encoder or corrector circuitry can only corrupt a single codeword digit (i.e. cannot propagate to multiple codeword digits).

2) There is a fault secure detector that can detect
any combination of errors in the received codeword along with errors in the detector circuit.This fault-secure

detector can verify the correctness of the encoder and corrector operation. An overview of our proposed reliable memory system is shown in Figure 1, and is as described below: The information bits are fed into the encoder to encode the information vector, and the fault secure detector of the encoder verifies the validity of the encoded vector. If the detector detects any error, the encoding operation must be redone to generate the correct codeword. The codeword is then stored in the memory. Later during operation, the stored codeword will be retrieved from the memory unit. Since the codeword is susceptible to transient faults while it is stored in the memory, the retrieved codeword must be fed into the checker to detect any potential error and possibly to the corrector to recover any erroneous bits. The fault secure detector of the corrector unit guarantees the correctness of the corrector unit operations with the detect-and-repeat technique similar to the encoder. Transient errors accumulate in the memory words over time. In order to avoid accumulation of too many errors in the memory words that surpasses the code correction capability, the system has to perform memory scrubbing. Memory scrubbing is periodically reading memory words from the memory, correcting any potential errors and writing them back into the memory .

2.2.3. Linear Block Error Correcting Codes:

This section provides a brief introduction on linear block ECCs. Let i = (i0, i1, ..., ik−1) be k-bit information vector that will be encoded into n-bit codeword, c = (c0, c1, ..., cn−1). For linear codes the encoding operation essentially performs the following vector-matrixmultiplication: c = i × G, where G is a k × n generator matrix. The checking or detecting operation is the following vector-matrix multiplication: s = c×HT , where H is an (n−k)×n Parity-Check matrix, and the (n − k)-bit vector s is called syndrome vector. A syndrome vector is zero if c is a valid codeword, and non-zero if c is an erroneous codeword. A code is a systematic code if any codeword consists of the original k-bit information vector followed by n − k parity-bits [8]. With this definition, the generator matrix of a systematic code must have the following structure: G = [I : X], where I is a k × k identity matrix and X is a k×(n−k) matrix that generates the parity-bits. The advantage of using systematic codes is that there is no need for a decoder circuitry to extract the information bits. The information bits are simply available in the first k bits of any encoded vector. A code is said to be Cyclic code if for any codeword c, all the cyclic shifts of c is still a valid codeword. A code is cyclic iff the rows of its parity-check matrix and generator matrix are the cyclic shifts of their first rows.



Direct compare Method

2.2.4.Encoder :

An -bit codeword , which encodes a -bit informationvector is generated by multiplying the -bit informationvector with a bit generator matrix

Encoder block shown in the above figure is used to encode the information , an encoder is a device circuit, transducer, software program, algorithm or person that converts information from one format or code to another, for the puropses of standardization, speed, security, or saving space by shrinking size.

Ex: A compressor encodes data (e.g audio/video images) into a small form.

2.2.5.Decoder:

A decoder is a device which does a reverse operation of an encoder, undoing the encoding so that the original information can be retrieved. The same method to used to encode is usually reversed in order to decode. It is a combinational ciruit that converts binary information from 'n' input lines to maximum of 2^n unique output lines.

Ex: The example decoder circuit would be an 'and' gate.

2.2.6.Memory Block:

Data bits stay in memory for a number of cycles andduring this period, each memory bit can be upset by atransient fault with certain probability. Therefore, transienterrors accumulate in the memory words over time.In order to avoid accumulation of too many errors inany memory word that surpasses that code correction capability, the system must perform memory scrubbing.Memory scrubbing is the process of periodically readingmemory words from the memory, correcting any potentialerrors and writing them back into the memory. Toperform the periodic scrubbing operation, the normalmemory access operation is stopped and the memoryperforms the scrub operation.

2.2.7.Corrector:

During to transient faults while they are stored in the memory. There for a corrector unit is designed to correctpotential errors in the retrieved code words. In our design,all the memory words pass through corrector andany potential error in the memory words will be corrected.Similar to the encoder unit, a fault secure detectormonitors the operation of the corrector unit.

2.2.8.Operation:

The information vector is given as input to the encoder, if the given input does'nt have any errors then it will give the message bits to the next stage (i.e.,memory block). Otherwise the given inputs have any errors then it will dectects & corrects the message bits and then after it will send to the memory block.

Before the memory block the message bits are combined with the generated matrix(i.e., C=MG) which will give the code word. Then the codeword is passed through memory block, the function of this block is to store the value one-by-one in matrix form. The main aim to use memory block is to store the given input data without any errors.

After the memory block, the data is transfered to next block which is called as 'suspected code word'. This suspected code word is nothing but which will compare the correct code word & expected code word.

From this block it will go to the serial pipeline corrector, which is nothing but passes the data in a serial manner i.e,(bit-by-bit). If any error occurs in the data it will again send back to the suspected code word, from there it will again compares the original data to the erronous data. Then after it will correct and again send to serial pipeline corrector.

If syndrome is equal to zero, then there is a perfect receive of data bits. Otherwise the data is erronous. In the serial pipeline corrector the data, if it is equal to one then the recieved data bit is error. So it will send to the detector (corrector) then it will again compare with the suspected code word if it is equal then the data bit is correct....

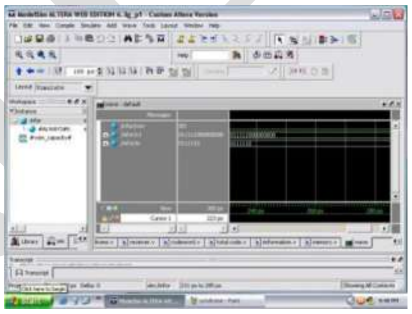
Memory scrubbing is the process of periodically reading memory words from the memory, correcting any potential errors and writing them back into the memory. To perform the periodic scrubbing operation, the normal memory access operation is stopped and the memory performs the scrub operation..

3. RESULT & ANALYSIS

3.1. Snapshots of Output:

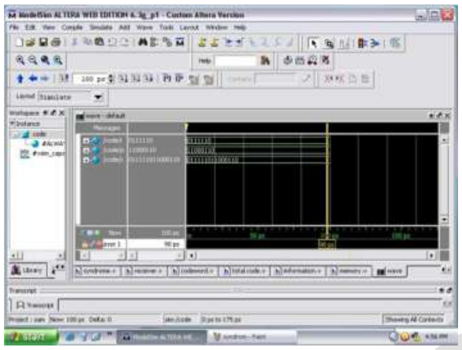
Information vector snapshot:

Here in this information vector input is:001111100000000, inorder to get the output corrector is given as '0', and we get the output as '0111110' because the output is of 7 bit thats why first 7 bits are accepted.



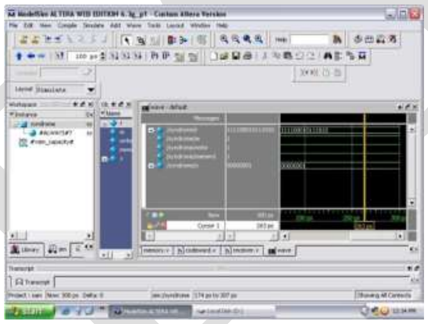
Snapshot of the code word:

The input of codeword is the output of information vector the input is given as '0111110' and the generator matrix is '11000110' the output is the combination of generator matrix and message therefore the output is '011111011000110' the output is of the form (n,k) where n is the message bits.



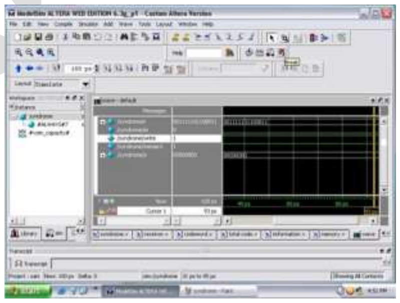
Syndrome with the errors

Here in the syndrome if the original code word matches with the suspected code word the syndrome output is '0', if it does't match the its value is '1', here the original code word is '111100010111011' but the suspected is '111100010111010' that's why the syndrome output is '1'.



syndrome with out errors:

Here the suspected code word is matched with the original code word thats why the output is '0'.



4. CONCLUSION

In this paper, we have developed a memory system that can tolerate and correct errors not only in the storage unit but also in the supporting circuitry. We used Euclidean geometry codes. We prove that these codes are FSDECC. Using these FSD we design a fault tolerant encoder and corrector ,where the fault secure detector monitors the operation.

5. FUTURE SCOPE:

In this project we have studied how the data should be sent in a secure manner and this project can be used in the applications of nanometry, but in the future we can develop a nanometric fault secure encoder which will be more accurate and more fast in speed compared with this project. In this project it tolerates soft errors and sends the data in a secure manner in the nanometric fault secure encoder it also tolerates soft errors and sends data in a more secure manner and with more accurate speed.

6. REFERENCES

[1] Forouzan, A.B., Data Communications &Networking(sie). TataMcGraw-HillEducation,chapter10,p.p265-278,2011

[2] en.wikipedia.org/wiki/Cyclic_redundancy_check4.

[3] I. S. Satran, D. Sheinwald, "Out of Order Incremental CRC Computation", IEEE Trans. Comp., vol. 54, pp. 1178–1181, 2005.

[4] Tam, S., Single error correction and double error detection. Xilinx Application Note, 2006.

[5] E. Stavinov, "A practical Parallel CRC Generation Method", Circuit Cellary, 2010.

[6] William Stallings, Data and Computer Communications, Tenth Edition. Pearson Education, Inc., publishing as Prentice Hall,chapter 6, p.p 194, 2014.

[7] M. Divya, K. Vinodalakshmi, G. Sumalatha, Shahbazsarik, Abhishek Awasthi,' Design and Implementation of Encoder for (15, k) BinaryBCH Code Using VHDLand eliminatingthe fan- out', IOSR Journal of Electronics and Communication Engineering (IOSR-JECE), Volume 7, Issue 6 (Sep. - Oct.2013), PP 01-06.

[8] M.Y. Rhee - "Error Correcting Coding Theory", McGraw-Hill, Singapore, 1989.

[9] Goresky, M. and Klapper, A.M. Fibonacci and Galois representations of feedback-with-carry shift registers, IEEE TransactionsonInformationTheory,Nov2002,Volume:48,On page(s): 2826 –2836.

[10] M.A.; Barbetta, L.; Neri, F.; "A Petri net simulation model of HDLC Marsan",TENCON '89. Fourth IEEE Region 10 International Conference 22-24 Nov. 1989Page(s):240– 247Digital Object Identifier 0.1109/TENCON.1989.176933.

s