LOW POWER CELLULAR AUTOMATA BASED TRIPLE BYTE ERROR CORRECTING CODE

P.Silambarasi*

M.Devanathan**

Abstract

A cellular automaton (CA) has already established its novelty for bits and bytes error correcting codes (ECC). The current work identifies weakness and limitation of existing CA-based byte ECC and proposes an improved CA-based triple byte ECC which overcomes the identified weakness. The code is very much suited from VLSI design viewpoint and requires significantly less hardware and power for decoding compared to the existing techniques employed for Reed–Solomon (RS) Codes. Also it has been shown that the CA-based scheme can easily be extended for correcting more than three byte errors.

Keywords-Byte error correcting code (ECC), Reed–Solomon (RS) code, cellular automata(CA).

* PG Scholar, Arunai College of Engineering, Thiruvannamalai

** M.E(Ph.D). Asst.Prof/Department of ECE, Arunai College of Engineering, Thiruvannamalai

I. INTRODUCTION

Channel coding is used for providing reliable information through the transmission channel to the user. It is an important operation for the digital communication system transmitting digital information over a noisy channel. Forward error correction technique depending on the properties of the system or on the application in which the error correcting is to be introduced. Reed-Solomon (RS) codes have been widely employed for error correction in modern digital communication and data storage systems. For example RS (28, 24) and RS (32, 28) with interleaving are popularly used for multimedia storage CD. RS (255, 239) code having 8 bytes error correcting capability has been recommended as a outer code in WiMax .A 16 bytes error correcting RS (255, 223) code has been used for digital micro-wave radio. To preserve important header information, MB-OFDM UWB adopts RS (23, 17) codes. Also in the recent IEEE 802.20 standard for LAN/MAN—part 20, it is recommended to use RS codes having 1, 2, 3, 4 error correcting capability such as RS (16, 12), RS (16, 14), RS (32, 28), RS (32, 26) as outer code. For portable devices power consumption is of prime concern and at the same time the silicon area should be kept as low as possible. Similar with other forward correction codes (FEC), when using RS codes as channel coding, the errors occurred in transmission procedure are typically divided into random errors and burst errors. Currently, for decoding RS codes with random-error correction, numerous literatures have given extensive studies on theoretical algorithms as well as hardware implementations .In this scenario, our aim to design a low power, high throughput, hardware efficient codec having error correcting capability. The current work identifies weakness and limitation of existing CA-based byte ECC and proposes an improved CA-based double byte ECC which overcomes the identified weakness. The code is very much suited from VLSI design viewpoint and requires significantly less hardware and power for decoding compared to the existing techniques employed for Reed–Solomon (RS) Codes. Also it has been shown that the CA-based scheme can easily be extended for correcting more than two byte errors.

A. Theory behind encoders and Decoders

There are many types of block codes, but the most notable is reed solomon coding, Golay, BCH, Multidimensional parity, and Hamming codes are other examples of block codes. In real world communication, errors are introduced in messages sent from one point to another shown below

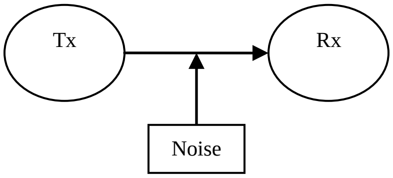


Figure 1. Point to point Communication

Reed Solomon is an error-correcting coding system that was devised to address the issue of correcting multiple errors - especially burst-type errors in mass storage devices (hard disk drives, DVD, barcode tags), wireless and mobile communications units, satellite links, digital TV, digital video broadcasting (DVB), and modern technologies like xDSL ("x" referring to all the existing DSL solutions, whether ADSL, VDSL, SDSL, or HDSL) In order for the transmitted data to be corrected in the event that it acquires errors, it has to be encoded. The receiver uses the appended encoded bits to determine and correct the errors upon reception of the transmitted signal. The number and type of errors that are correctable depend on the specific reed solomon coding scheme used.

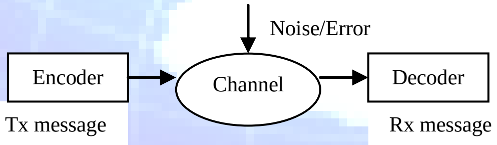


Figure 2. Reed Solomon Codings Scheme.

Reed Solomon Coding SchemeComplexity of RS encoder and decoder increases with the error cor-recting capability of the codes. Hence many researchers have put their effort to minimize the complexity of RS codec. A number of general encoding and decoding schemes of the RS codes may be found in the literature [11]. A high speed systolic architecture for decoding RS code using Berlekamp–Massey (BM) algorithm has been published in[5].

A novel high-speed degree-computation less modified Euclidean algorithm architecture for high-speed RS decoder has been reported in [6]. RS decoding scheme proposed in [5] and [6] are more general in the sense that any arbitrary number of errors can be corrected. But both the schemes require complex modules.

It has been found that these parameters are supported by local neighborhood CA. In [2], CA-based byte error correcting code has been proposed. Another de-sign scheme for CA-based byte error correcting code has been reported in [4]. The proposed design in [2] requires less hardware compared to the existing techniques used for RS code. A design and implementation of CA-based RS (32, 28) encoder and decoder has been presented in [7]. But the modified scheme [7] and the previous scheme [2] can correct t-byte errors ${}_{3} \pm {}_{3} \quad {}_{3} \; {}_{3}$ provided errors are totally confined to information or check byte only. An improved double byte error correcting code using CA has been proposed in [8]. Also a weakness of the schemes [2], [7] has been reported and rectified using modified check symbol expression in [8].

In this work, the design and implementation of an improved double byte error correcting code using CA has been proposed. It also reports the extension of the scheme to detect/correct more than two byte errors. The performance of the double-byte error correcting decoder in terms of decoder error and decoder failure has been described. A comparison between the conventional RS codes and CA-based codes is given. Synthesis result shows that CA-based proposed design requires much less hardware and power. Our proposed scheme is suitable for codes having smaller number of error correction capability and smaller data word length.

The rest of this paper is organized as follows. In Section II, a brief overview of CA-based double byte error correcting code is described. Improved scheme for triple byte error correcting codes and its VLSI architecture are described in Section III. Result is given in Section IV. An extension of double byte error correcting code for more than three byte error correction is given in Section V and finally the paper is concluded in Section VI.

II. EXISTING CA BASED DOUBLE BYTE ERROR CORRECTING CODE.

In this section, we discuss the existing CA based double byte error correcting code which has been proposed in [2] and its weaknesses and limitation reported in [8].

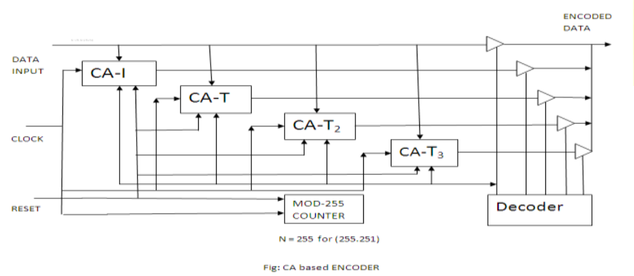


Figure 3. Existing CA Based Encoder

A. OVERVIEW OF EXISTING CA-BASED DOUBLE BYTE ERROR CORRECTING CODE

CA consist of a number of cells arranged in a regular manner, where the state transitions of each cell depends on the state of its neighbors and each cell consists of D flip-flop and a combinational logic imple-menting the next-state function. The next state function is called the rule of the automata. An r cell CA can be characterized by a r * r characteristic matrix T . The detailed theory of CA may be found in [10]. In CA-based double byte error correcting code [2], the four check bytes are generated by running the CA for N cycles, while sequentially feeding the N information bytes (D_m)where $0\leq m\leq(N-1)$. The expression for the bth check byte can be expressed as

$$C_b=D_{N-1}+T^b[D_{N-2}]+\ldots.+T^{b(N-1)}[D_0]$$

B. WEAKNESSES AND LIMITATION OF EXISTING CA-BASED DOUBLE BYTE ERROR CORRECTING CODE

In general, the error correction capability of RS encoders and decoders decrease with increase in the complexity. Also their area as well as power increases. So we need to design a system that is advantageous than above.

In existing CA based codec we have the following points to take care for the problem definition:

Weaknesses*:*

One weakness of the scheme in [2] is that single byte error in m^th^ information byte and double byte errors (one in m^th^ information byte and another in the last information byte) correspond to same equation for error location identification.

Limitation*:*

The scheme in [2] can correct errors provided errors are confined to information or check byte only. The scheme in [2] cannot correct if the errors are distributed both in information and check bytes.

III.ARCHITECTURE OF IMPROVED CA-BASED TRIPLE BYTE ERROR CORRECTING CODEC

This section explains the proposed architecture of encoder.

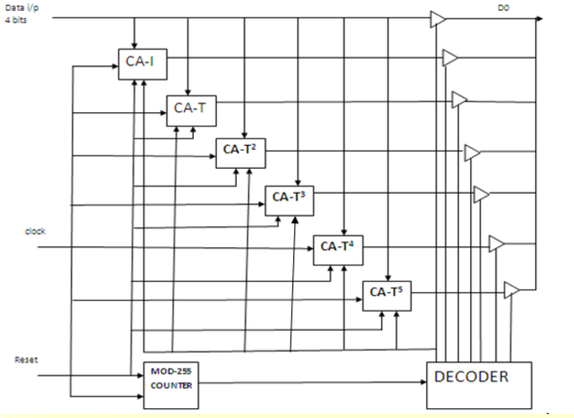


Figure 4. CA Based Encoder

1.Cellular Automata

A Cellular Automata is a discrete model studied in computability theory, mathematics, physics, complexity science, theoretical biology and microstructure modeling. It consists of a regular grid of *cells*, each in one of a finite number of *states*, such as "On" and "Off" (in contrast to a coupled map lattice). The grid can be in any finite number of dimensions. For each cell, a set of cells called its *neighborhood* (usually including the cell itself) is defined relative to the specified cell. For example, the neighborhood of a cell might be defined as the set of cells a distance of 2 or less from the cell. An initial state (time *t*=0) is selected by assigning a state for each cell.

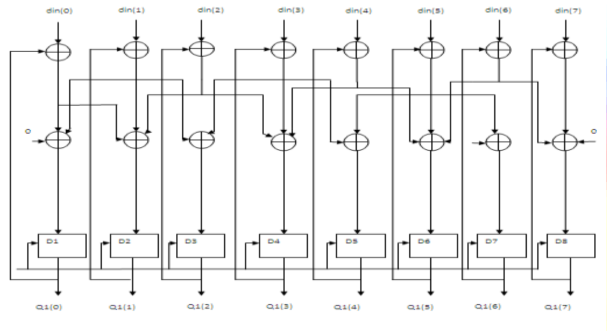


Figure 5. Internal Architecture of CA-T

A new *generation* is created (advancing *t* by 1), according to some fixed rule that determines the new state of each cell in terms of the current state of the cell and the states of the cells in its neighborhood.It consists of two X-OR gate and one D-flip-flop. And the X-OR1 output shifting back once and front once to X-OR 2 input.

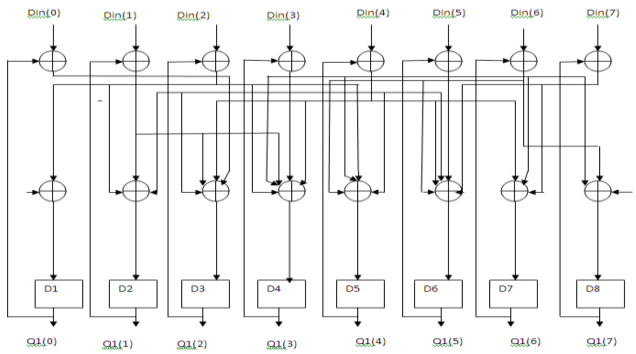


Figure 6. Internal architecture of CA-T^2

It consists of two X-OR gate and one D-flip-flop. All the connections are similar to CA-T RTL. But the X-OR1 output shifting back twice and front twice to X-OR 2 input.

These CA,CA-T,CA-T^2,..CA-T^5 are similar but based on the shifting and internal connection it will get varied.

Mod - N Counter

A synchronous binary counter counts from 0 to 2^N-1, where N is the number of bits/flip-flops in the counter. Each flip-flop is used to represent one bit. The flip-flop in the lowest-order position is complemented/toggled with every clock pulse and a flip-flop in any other position is complemented on the next clock pulse provided all the bits in the lower-order positions are equal to 1. Take for example A_4 A_3 A_2 A_1 = 0011. On the next count, A_4 A_3 A_2 A_1 = 0100. A_1, the lowest-order bit, is always complemented. A_2 is complemented because all the lower-order positions (A_1 only in this case) are 1's. A_3 is also complemented because all the lower-order positions, A_2 and A_1 are 1's. But A_4 is not complemented the lower-order positions, A_3 A_2 A_1 = 011, do not give an all 1 condition. To implement a synchronous counter, we need a flip-flop for every bit and an AND gate for every bit except the first and the last bit. The diagram below shows the implementation of a 4-bit synchronous up-counter. In a binary up counter, a particular bit, except for the first bit, toggles if all the lower-order bits are 1's. The opposite is true for binary down counters. That is, a particular bit toggles if all the lower-order bits are 0's and the first bit toggles on every pulse.

IV.RESULT FOR ENCODER

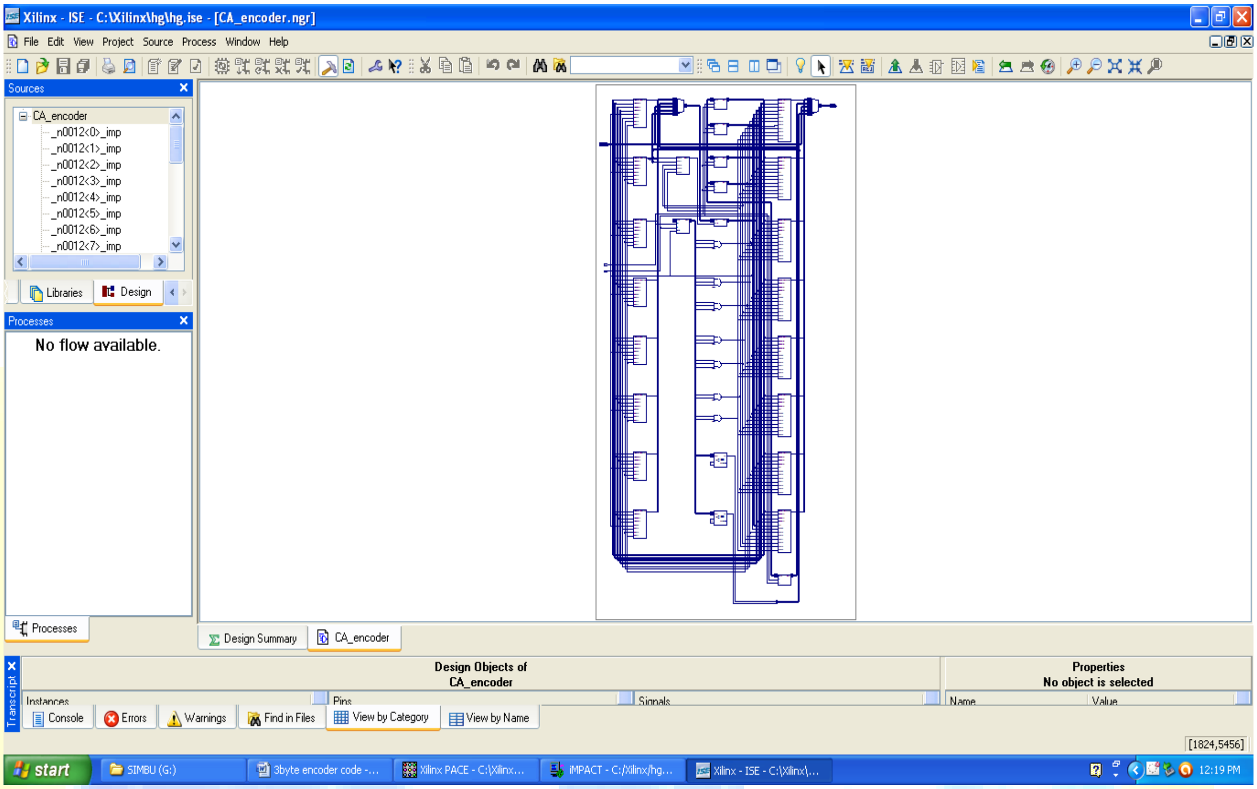
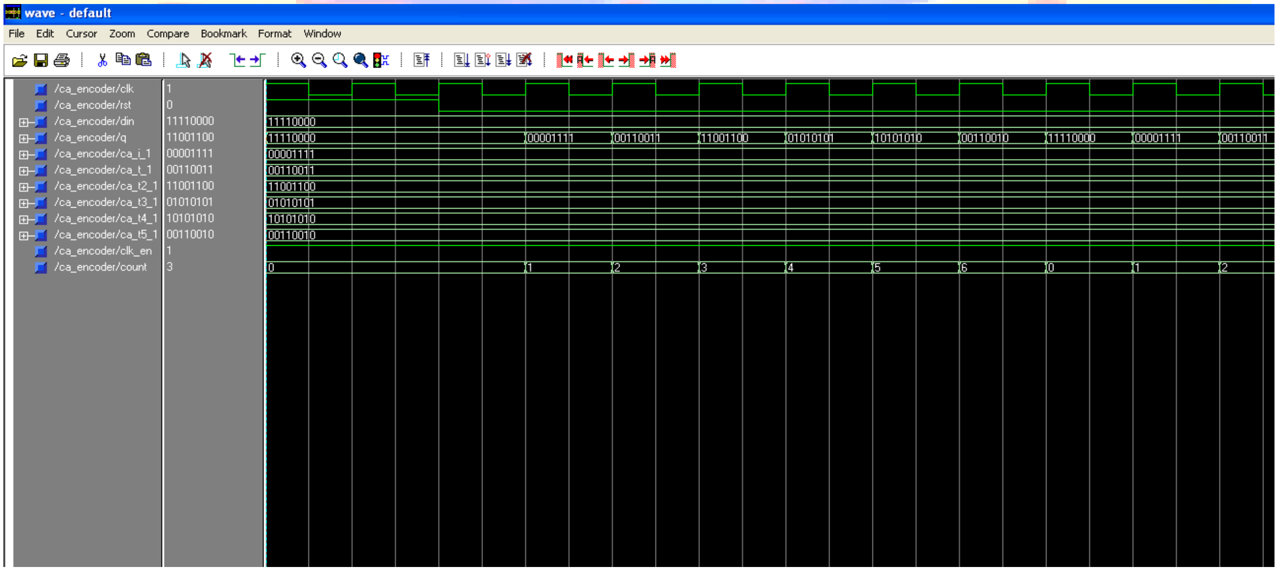


Figure7. Synthesis view for CA encoder



```
/ca_encoder/clk      1
/ca_encoder/rst      0
/ca_encoder/din      11110000   11110000
/ca_encoder/q        11001100   11110000 00001111 00110011 11001100 01010101 10101010 00110010 11110000 00001111 00110011
/ca_encoder/ca_i_1   00001111   00001111
/ca_encoder/ca_t_1   00110011   00110011
/ca_encoder/ca_t2_1  11001100   11001100
/ca_encoder/ca_t3_1  01010101   01010101
/ca_encoder/ca_t4_1  10101010   10101010
/ca_encoder/ca_t5_1  00110010   00110010
/ca_encoder/clk_en   1
/ca_encoder/count    3          0 1 2 3 4 5 6 0 1 2
```

Figure8. Simulaton result for CA encoder

This simulation result shows the function of ca_encoder. It consists of binary input clk, reset, Din,ca_i,ca_t,ca_t2,ca_t3,ca_t4,ca_t5 in and Q is output. At the initial run command the din received at q, next run function ca_i output received .the output progressed sequentially ca_t,ca_t2,ca_t3,ca_t4,ca_t5 and the consecutive output. then function to initial state ,repeat the same sequence of operation.

V.EXTENSION OF THE SCHEME FOR t>3 BYTES ERROR

The CA based scheme can easily be extended for more than three byte errors. The scheme for 4-byte error correcting code is introduced in this section. In four byte error correcting code eight check bytes and six syndrome bytes can be generated using (11) and (12), respectively, in Section III.

VI.CONCLUSION

An improved scheme for the Triple byte error correcting code using CA which overcomes the weakness and limitation of existing scheme. The proposed Triple byte error correcting scheme can generate six check byte and information bytes for determining the error locations which is independent of erroneous byte position, provided number of errors is less than or equal to three. The code is much simpler to design and requires much less hardware and consumes less power for encoding.

The Proposed work is

1) Syndrome generator.
2) Error magnitude and location identifier.
3) Design of decoder.
4) Design of whole system (encoder and decoder).
5) FPGA Implementation.

VII.REFERENCES

[1] Bo Yuan ; Zhongfeng Wang ; Jin Sha ; Hongbing Pan ; Weishan Zheng , "Unified Architecture for Reed-Solomon Decoder Combined With Burst-Error Correction". IEEE Trans on VLSI syst., vol. 20, page: 1346 – 1350.July2012.

[2] Wang, H. ; Pan, B. "Reduced-Complexity LCC ReedSolomon Decoder Based on Unified Syndrome Computation". IEEE Trans on VLSI syst., vol.pp, page:1. 2012 .

[3] Amir Mokhtar Chabi, Samira Sayedsalehi and Keivan Navi, "New Modules for Quatum-dot Cellular Automata AND & OR Gates". Canadian Journal on Electrical and Electronics Engineering., Vol.3, No.5, May 2012.

[4] F.Garcia-Herrero, M.J.Canet, J.Valls, and P.K. Meher, "High-ThroughPut Interpolator Architecture for Low-Complexity Chase Decoding of RS Codes". IEEE Trans. On VLSI syst., vol. 20, No.3, MARCH 2012.

[5] Jaydeb Bhaumik and Dipanwita Roy Chowdhury, "New Architectural Design of CA-Based Codec". IEEE Trans. On VLSI syst., vol. 18, No.7, JULY 2010.

[6] C. L. Chen, "Error-correcting codes for byte-organized memory systems,"*IEEE Trans. Inf. Theory, vol. 32, no. 2, pp. 181–185, Mar. 1986.*

[7] D. R. Chowdhury, I. S. Gupta, and P. P. Chaudhuri, "CA-based byteerror-correcting code," *IEEE Trans. Comput., vol. 44, no. 3, pp.*371–382, Mar. 1995.

[8] T. K. Matsushima, T. Matsushima, and S. Hirasawa, "Parallel architecturefor high-speed Reed-Solomon codec," in *Proc. IEEE Int.Telecommun. Symp., 1998, pp. 468–473.*

[9] S. Nandi, C. Rambabu, and P. P. Chaudhuri, "A VLSI architecturefor cellular automata based Reed-Solomon decoder," in *Proc. I-SPAN,*1999, pp. 158–165.

[10] D. V. Sarwate and N. R. Shanbhag, "High-speed architectures forReed-Solomon decoder," *IEEE Trans. Very Large Scale Integr. (VLSI)Syst., vol. 9, no. 5, pp. 641–655, Oct. 2001.*

[11] S. Lee, H. Lee, J. Shin, and J. Ko, "A high-speed pipelined degreecomputationless modified Euclidean algorithm architecture for Reed-Solomon decoders," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS),*2007, pp. 901–904.

[12] J. Bhaumik, D. R. Chowdhury, and I. Chakrabarti, "Design and implementationof Rs(32, 28) encoder and decoder using cellular automata," in *Proc. 15th Int. Conf. Adv. Comput. Commun. (ADCOM), India,* 2007, pp. 491–496.