IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

Easing Complication and Error Detection of RS Codes Using CA based Codes

N.Mageswari[*1], Dr.K.Mahadevan[2]

[*1]Assistant Professor, Department of EIE, Bharath Niketan Engineering College, Aundipatti,Tamilnadu, India

[2]Professor,Dept of EEE, PSNA College of Engineering and Technology,Dindigul,Tamilnadu,India

magesmaniengg@gmail.com

Abstract

This paper reports a novel approach for Cellular Automata (CA) based byte error correction codes to extend the integrated architecture of Reed-Solomon (RS) codes. In general, reed Solomon codes are widely used for forward error correction codes in communication system. Burst-error correction is a relatively complicated playing field because of the computational difficulty. In Existing work, they propose a unified design of reed Solomon decoder for correcting the random errors and burst error with fewer complications. In this paper, we propose Cellular Automata based byte error correction by using an adaptive algorithm which can correct the random error as well as burst and byte level of error detection and correction. With these automata, the power and error correction improves for both the encoder and decoder can be efficiently implemented with an array of CA with high throughput. This design is best suited for high speed memory systems built with byte organized RAM chips. This design methodology is so simple which can be implemented in VLSI design.

Keywords: RS codes, Cellular–Automata (CA) code, error correction, VLSI implementation.

Introduction

Error control codes are widely used in communication systems to protect the transmitted data from errors. Error control codes are classified into convolution and block codes. Reed–Solomon (RS) Codes are linear block codes and belong to the class of non binary BCH codes. RS codes [1] providing the capability to efficiently correct burst errors, as well as random errors, have been extensively used in various communications and digital data storage systems, such as Power Line Communications (PLC), Digital Video Broadcasting Terrestrial (DVB-T) system, Vestigial sideband (VSB) system, cable modem, satellite and mobile communications, magnetic recording, etc. The general (n, k, t) RS code defined in the Galois field (GF) has a code length n = 2m- 1, where k denotes the number of m-bit message symbols. The RS code have the error correction capability [t = b (n - k)/2]. RS decoder consists of five blocks, such as the syndrome computation, the key equation solver, the Chien search algorithm, the Forney algorithm, and the error correction. Among these blocks, the key equation solver block requires the highest hardware complexity and many clock cycles to obtain the error locator and the error value polynomials. The Berlekamp–Massey (BM) algorithm, the Euclid algorithm, and the modified Euclid (ME) algorithm [9] have been used to solve the key equation. Various key equation solver blocks have so far been proposed to meet real-time requirement and to reduce hardware complexity. The Euclid algorithm requires inversions, that is, computations of inverse elements to find the greatest common divisor, thus, it requires either a lookup table (LUT) or efficient circuits for the inversion.

For example, RS code has been used for digital micro wave radio; this error correcting scheme leads high power, less hardware efficient, minimum throughput and it also increase the complexity of the system. In this scenario, to design a low power, high throughput, hardware efficient codec having error correcting capability by using technique of Reed Solomon decoder.

In this paper, we propose cellular automata [2] based byte error correction by using an adaptive algorithm which can correct the random errors as well as burst error and byte level of error detection and correction. In this proposed method, it improves the power, high throughput and efficient hardware.

This work is provoked by the following two goals:

1. The first goal is to suggestively can reduce the total gate count over and done with we can get the improved throughput, latency and area.

2. In addition to that our method also provides a good balance between speeds and keeps the extra power consumption low.

The rest of the paper is organized as follows. The Section II Existing Reed Solomon encoder and decoder, Section III for proposed work cellular automata, Section IV for Result and conclusion, Section V for References.

Error Control Coding

Error-control techniques[3],[4] are important in digital communication. In several system applications we find errors both in communication and in storage. Errors in transmission are mainly because of noise, electromagnetic interferences, cross talk, and bandwidth. In storage errors may occur because of increase in magnetic flux as lest of magnetic disc or it can be false change of bits because of electromagnetic interferences as in case of DRAM. So dealing with these errors when they occur is the matter of concern. The primary step is to detect the error and the following step is to correct the error and reconstruction of the original data. Reed Solomon code is one type of error control techniques these are block error correcting codes with wide range of applications in the field of digital communications. These codes are used to correct errors in devices such as CD's, DVD's etc.., wireless communications, many digital subscriber lines.

A set of code words used with an encoder and decoder to detect and correct errors, or both detect and correct error accordingly error-detecting and correcting codes can be commonly well-known between random error detecting and correcting, burst error detecting and some codes can also be suitable for a mixture of random errors and burst errors.

Cellular Automata Techniques

A cellular automaton (CA) is a discrete model. Cellular automata consists of number of cells arranged in a regular manner, where the state transmission of each cell depends on the state of its neighbours and each cell consists of D flip-flop and a combinational logic implementing the next state function is called the rule of automata. In CA[5] there are 256 (rules) distinct next state functions, the next state function of the cell is expressed in the form of a truth table then the decimal equivalent of the output is referred to as "rule" for the cell.

An essential application of single byte error correcting-double byte error detecting codes in the design of memory subsystem has been presented in VLSI .The VLSI system designer always prefers to have simple, regular, modular, and cascade able structure with local interconnection for reliable and high speed operation of the circuit.

Existing Error Correction Method

A Different error correcting techniques have been proposed in error correction method to reduce their cubic computation complexity of the system. However, these error correction techniques are unable to solve the errors that are occurred in the time of processing of signal from encoder to decoder. . The following are the previous methods used for error correcting are Hamming code, Walsh – Hadamard code, Reed –Solomon code.

Existing Reed Solomon Codes

Reed Solomon Encoder. Reed Solomon (RS) codes are viewed as a non-binary cyclic codes and it is a special case of Bose-Chaudhuri-Hocquenghem (BCH) codes, RS codes in the parameter n, k, t and any positive integer m>2, m-bits data symbols exist for all n and k where n is the total number of codeword symbol and k is the number of message symbols, Reed-Solomon codes are mostly useful for burst-error correction they are effective for channels that have memory. Reed Solomon Encoder[1] takes a block of digital data and extra redundancy bits. Encoding is a process of converting a input message into a corresponding codeword polynomial C (z) it operates on multiple bits rather than individual bits. A encoding codeword that are comprised of data symbols followed by parity-check symbols is given by

$$(c_{n-1,}\ c_{n-2,\ldots\ldots}c_1,c_0)= (d_{k-1,}\ d_{k-2,}\ldots\ldots d_{1,}d_0 ,-p_{n-k,}\ -p_{n-k-2\,,\ \ldots..}-p_{1,}-p_0)$$

Reed Solomon Decoder. Let c(z) denotes the transmitted codeword polynomial and let R(z) denotes the received word polynomial .the input to the decoder and it assumes that

$$R\,(z)=C(z)+E(z)$$

where, if errors have occurred during transmission, the error polynomial can be written as

$$E\,(z)=y_1z^{i}_{1+}\ Y_2Z^{i}_{2+\ldots\ldots+}\ Y_e\,Z^{i}_{e}$$

It is conventional to say that the error values Y_1 $Y_{2,\ldots}$ Y_e have occurred at the error location $X_{1=\alpha^{i}_{1},}$ $X_{2=\alpha^{i}_{2},}$ $X_{e=\alpha^{i}_{e..}}$In [6], Li Li proposed a Reformulated Inversion less Burst Error Correction (RIBC) Algorithm and Unified Hybrid Decoding (UHD) Architecture for correcting errors in different work modes and the design has some cubic computation complexity as lower throughput.

Existing CA Byte Error Correction

A Cellular Automata is an array of the same and identical automata or cells, which help with each other. The variable is either a number or a property, which confirms the difference between each cell and neighbourhood represents the cells that exist around the specified cell. The specified cell interacts with its surrounding cells in every successive time interval. The present states of the neighbouring cell taken into account for the next state of each cell. Program is the set of laws

ISSN: 2277-9655
Impact Factor: 1.852

that are incorporated and the result is change in status of the state of the specified cell and the neighbouring cells.

CA can be characterized by a n × n characteristic matrix T. In CA-based double byte error correcting code [7], 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 b^{th} check byte can be expressed as

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

where, $0 \leq b \leq 3$ and T is the characteristic matrix of a n-cell maximum length CA.

In decoder, the syndrome corresponding to the b^{th} check byte, S_b is the computed using the expression

$$S_b = C_b \oplus C'_b: \quad 0 \leq b \leq 3$$

where, C_b is the b^{th} received check byte and C'_b is the b^{th} check byte recomputed from the received information bytes. Assume E_m and E_n are the errors in m^{th} and n^{th} information byte, with $m \neq n$ then corresponding syndrome equations are

$$S_b = T^{b(i)}[E_m] \oplus T^{b(j)}[E_n]$$

The error magnitudes are determined using the following equation

$$E_n = T^z(T^i[S_0] \oplus S_1): \quad E_m = S_0 \oplus E_{n.}$$

Weakness and Limitation. One weakness of the scheme 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. Assume E_m is the error in the m^{th} information byte, and then syndrome equations are

$$S_b = T^{b(i)}[E_m]; \text{ where} 0 \leq b \leq 3.$$

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

Proposed Work

In this paper, we proposed a design of CA Encoder and decoder

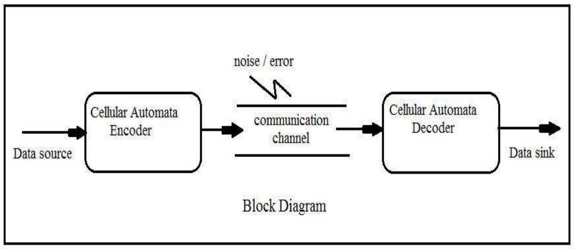


Figure 1. CA Block Diagram

Encoder

An encoder is a device, circuit, transducer, software program or algorithm that converts information from one format to another format or from one type of code to another code, for the purposes of consistency, speed, secrecy, security or saving space by reducing size.

In encoder, check byte C_b is generated by running the CA for N cycles while sequentially feeding the N information bytes using the expression

$$C_b = T^b D_{N-1} \oplus T^{b2}[D_{N-2}] \oplus \ldots\ldots \oplus T^{b(N)}[D_0]$$

Where $0 \leq b \leq 3$ and T is the characteristic matrix of a 8-cell maximum length CA.

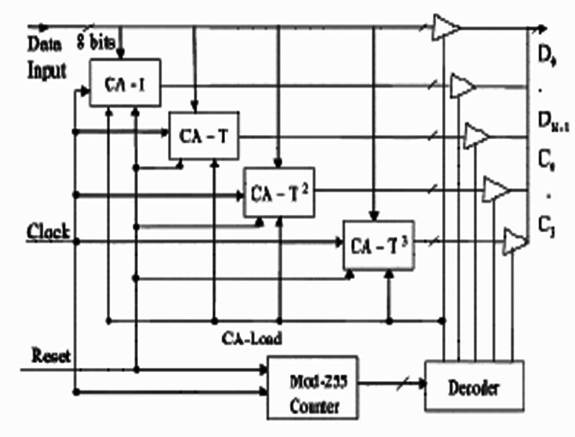


Figure 2. CA encoder architecture

The above figure illustrates the architecture of encoder. The encoder block is using combination of XOR gates and D Flip Flop depends upon the rules of cellular automata. The 8 bit of data is given as an input of encoder and output is taken with the check bytes namely C0, C1, C2, and C3. This check bytes are determined by the Cellular Automata based rules of blocks. The four check bytes are generating by running blocks CA-I , CA-T, CA-T^2,CA- T^3.

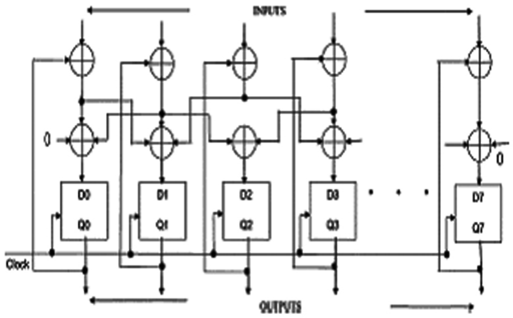


Figure 3. Internal Architecture of CA-T

Fig 2 shows the internal architecture of CA-T having rule vector (150, 150, 90, 150, 90, 150, 90,150). The architecture of CA- T^b is similar to CA-T except the connections are made by considering matrix T^b as a characteristic matrix.

Decoder

The architectural design of Decoder needs four models that are used to recover original message information from encoded stream of data which are sent from the encoder. Those are mentioned below

1. Syndrome generator.
2. Error location identification block.
3. Error magnitude computation block.
4. Error correction block.

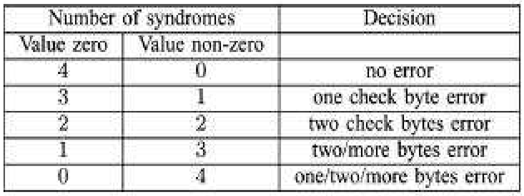
Syndrome Generator. In binary segment the term syndrome means addition of extra bits for future reference, which helps the further more process of decoder. In CA, The syndrome corresponding to the b the check byte S_b is

$$S_b = C_b\,(\text{XOR})\,C'_b\ ;\ 0\leq b \geq 3$$

Where C_b is the b th received check byte and C'_b is the b th check byte recomputed from the received information bytes. These are XOR-ed with the received check bytes C0,C1,C2,C3 respectively, to generate the syndrome bytesS0,S1,S2,S3 Generated four syndromes are stored in four registers, which will be used to locate and correct the errors.

The decoding logic is given in the following table.

TABLE I
DECODING LOGIC



Number of syndromes		Decision
Value zero	Value non-zero	
4	0	no error
3	1	one check byte error
2	2	two check bytes error
1	3	two/more bytes error
0	4	one/two/more bytes error

Error Location Identification Block. It identifies the location of error in the syndrome. The location identifier determined by the following equation. The equations to identify the two error locations and determine two error magnitudes are described as follows. Suppose E_k and E_l are the errors in kth and lth information bytes, then the corresponding syndrome equations are

$$S_{b\,=\,T}{}^{b(i)}[E_k] \oplus T^{b(j)}E_t$$

Where $0 \leq b \leq\ 3$; i +k =N and j + l =N From the four syndrome equations, error locations k= N-i and l=N-j are determined using the following two equations.

$$T^i\,[S_2] \oplus S_3 = T^{2j}(T^i[S_0] \oplus S_1)$$

$$T^{2i}[S_1] \oplus S_3 = T^j(T^{2j}\,[S_0] \oplus S_2)$$

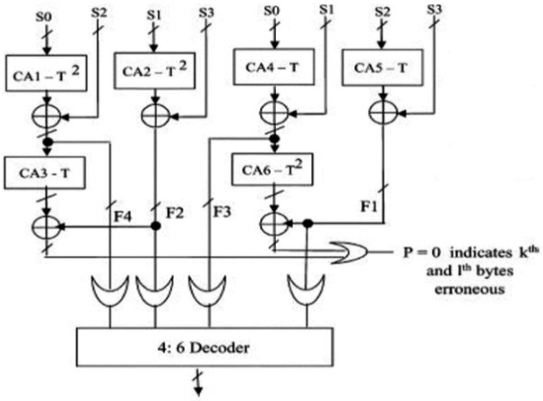
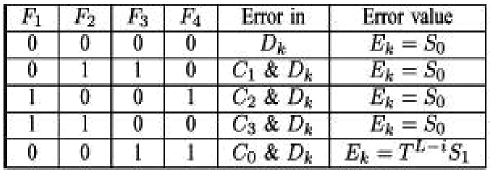


Figure 4. Architecture for Error Location Identifier

The generated syndrome are given as a input of Error Location Identifier and The blocks CA1, CA2, CA4, CA5 are allowed to run till number of clock cycles N=251 to produce the output. In other hand the point P=0 indicates the kth and 1th bytes erroneous by using relationship k=N - i and l=N- j . If the received code word has more than two errors then the output does not go low for the range of i and j.

TABLE II –Error Location Identifier Logic.



F_1	F_2	F_3	F_4	Error in	Error value
0	0	0	0	D_k	$E_k = S_0$
0	1	1	0	C_1 & D_k	$E_k = S_0$
1	0	0	1	C_2 & D_k	$E_k = S_0$
1	1	0	0	C_3 & D_k	$E_k = S_0$
0	0	1	1	C_0 & D_k	$E_k = T^{L-i}S_1$

$$F_1 = S_3 \oplus T^i S_2,\ F_2 = S_3 \oplus T^{2i} S_1$$
$$F_3 = S_1 \oplus T^i S_0,\ F_4 = S_2 \oplus T^{2i} S_0.$$

Magnitude Computation block. The block diagram for error magnitude calculation in case of double information bytes errors

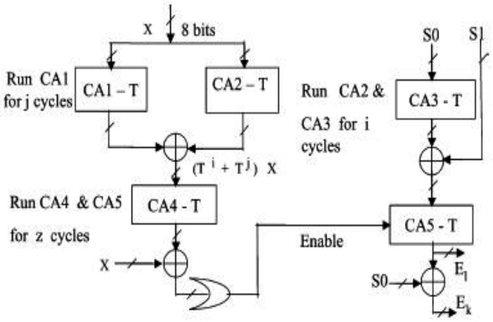


Figure 5. Architecture of Magnitude computation

In above figure .where X is a nonzero 8-bit number. Error correction is done by XOR-ing the erroneous bytes with error vector. From the logical table(TABLE II) of error location identifier, it is noted that information byte error magnitude for the first four types of error is equal to S_0.So it is only required to design a module for error in first check byte and any information byte. The information byte error magnitude for one information byte and first check byte error is determined by running CA-T for N-j cycles with $S_1$is initial seed.

Error Correction Block. This block is used to compute the erroneous bytes in the received bytes and it corrects the errors on it. It uses multiplexers and comparators for correcting the errors and produces the output.

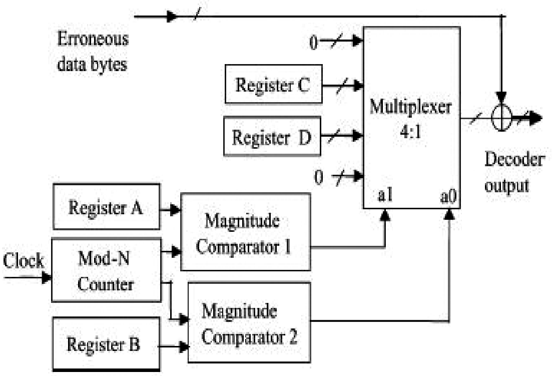


Figure 6. Architecture for Error correction

Result and Comparison

Table 3 Comparison Table

	PROPOSED CA CODES		EXISTNG RS CODES	
Modules	**Encoder**	**Decoder**	**Encoder**	**Decoder**
Power	33.75	33.71	57.59	57.18
GateCount	672	3104	1912	18506
Speed	1.73		2.43	
Throughput	4.0		1.77	

In this section we report a system with required much less power, high throughput, hardware efficiency, and increase the power. Simultaneously it perform a double byte error detection and correction using CA design of the proposed (255,251) byte error correcting codes. CA based encoder and decoder Modules have been implemented in VHDL and simulated using Modelsim 5.5f software.The design has been synthesized by XILINIX 12.1.

Conclusion

This paper presents an improved design for the double byte error correcting code and also it cover different application using CA which over the existing method**.** The CA based encoding and decoding scheme for correcting and detecting double byte errors of such a code is suitable for VLSI design position and smart for its straightforwardness and uniformity. The basic cell of a Double byte Error Location and Double byte Error Correction code has been designed, that are simulated and synthesized by Xilinx tools.The design can be basically extended to correct multiple byte errors. The ability of information bit will be increased from 8 bits into 251 bits by the proposed system of Cellular Automata and also it increases the speed of the system by simultaneous error correcting technique.

References

[1] S.Ardalan, K.Raahemifer, F.Yuan and V.Geurkov "Reed–Solomon Encoder and Decoder Design,Simulation and Synthesis",*IEEE CCECE* 2003,May 2003.

[2] Dipanwita Roy Chowdhury,Indranil Sen Gupta and Parimal Pal Chaudhuri "CA – Based Byte Error-Correcting Code",*IEEE Transactions on Computers*,vol.44,no.3,March 1995.

[3] S.Lin and D.J.Costello "Error Control Coding:Fundamentals and Applications",Prentice- Hall,Engle wood Cliffs,N.J.,1983.

[4] T.R.N.rao and E.Fujiwara "Error Control Coding for Computer Systems",Prentice-Hall,Engle wood Cliffs,N.J.,1989.

[5] A.K.Das and P.P.Chaudhuri, "Efficient characterization of cellular automata",*Proc.IEE(Part E)*,vol.137,no. 1,pp.81-87,Jan.1990.

[6] Li li,bo yuvan (july2012) unified architecture for Reed Solomon decoder combined with burst error correction IEEE trans " *IEEE transactions on very large scale integration (vlsi) systems*, vol. 20, No.7.

[7] Yuichi Saitoh, Hideki Imai.(May 1991), "Multiple Unidirectional Byte Error-Correcting Codes," *IEEE transactions on information theory*, vol. 37, NO. 3.

[8] Wolfgang Wilhelm(March 1999), "A New Scalable VLSI Architecture for Reed-Solomon Decoders," *IEEE journal of solid-state circuits*, vol. 34, No. 3.

[9] Jae H. Baek and Myung H. Sunwoo(August 2006), "New Degree Computation less Modified Euclid Algorithm and Architecture for Reed-Solomon Decoder" *IEEE transactions on very large scale integration (vlsi) systems*, vol. 14, No. 8.

[10] CHIN-LONG CHEN (March 1986), "Error-Correcting Codes for Byte-Organized Memory Systems," *IEEE transactions on information theory*, vol. it-32, No. 2.

[11] Bella Bose and Sulaiman Al-Bassam ,“Byte Unidirectional Error Correcting and Detecting Codes,” *IEEE transactions on computers*, vol. 41, NO. 12.