Digital Finite – Field for Data Coding and Error Correction in GF(2^m)

Mohammad Alipour[#1], Abdollah Saberi Manesh[#2], *Seyed Aliakbar Mousavi[#3], Putra Sumari[#4],
Muhammad Rafie Mohd Arshad[#5]

[#] School of Computer Science, Universiti Sains Malaysia (USM), Malaysia.
[1] alipoures@gmail.com
[2] saberimanesh62@hotmail.com
[3] pouyaye@gmail.com
[4] putras@cs.usm.my
[5] rafie@cs.usm.my

***Abstract*—Data coding and encoding standards require the high-performance error detection and correction algorithms. This paper presents the design error detection and corrects it by Galois Field and binary polynomial and extraction of its root. We start from $GF(2^3)$ and reach to $GF(2^8)$. The Galois Field is explained along with its requirement and relationship between digital and Galois theories. Also Neural Network is used for extraction of some codes those have been explained in another article. Finally all of the coding and encoding are calculated and processed. The results shows successful design error detection and correction by employing Galois Field and binary polynomial and extraction of its root.**

Galois Field, Binary Coefficients, Polynomial, Data Coding and Encoding, Neural Network, ONB, ECC, GEF.

I. INTRODUCTION

Galois field (*GF*) arithmetic is commonly used in encoding and decoding architectures [1]. This way use in communication, electronics, image processing, broadcast science[1] , genetics science and other fields and techniques [2] [3]. Commonly in communication and image processing $GF(2^8)$ is used. A Galois Field is a set of values combined with the operations such as addition, multiplication and inversion, which always produce elements of the field when performed on elements of the field. While these two operations are called addition and multiplication, they are not normal mathematical addition and multiplication except in the infinite Galois Field of all real numbers. ECC is commonly implemented over two major types of fields: prime fields i.e. $GF(Z_P)$ and binary fields i.e. $GF(2^m)$. Prime fields contain all integers between 1 and prime number and binary fields contain all values represented by a set number of bits. For example:

$$y = x^3 + ax + b \quad \bmod p$$

For prime fields i.e. $GF(Z_P)$.

(1)

$$y + yx = x^3 + a\,x^2 + 1$$

For binary fields i.e. $GF(2^m)$.

Of the binary fields, there are two commonly used representations of the values: polynomial basis and normal basis. The Figure 1 shows this taxonomy. In this paper polynomial basis is developed.

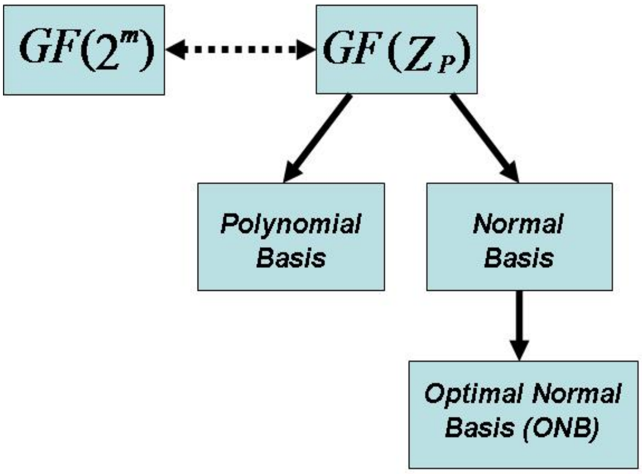

Fig. 1: Taxonomy of *GF*

II. $GF(2^m)$ FAST REVIEW

A $GF(2^m)$ field [4] is a collection of 2^m field elements that each element is m bits wide. When m>1, the $GF(2^m)$ field is an extension of the field $GF(2\ )$, with {0,1}. The mentioned operations defined in $GF(2\ )$ are addition and multiplication, each performed in modulo 2 [4] [5]. A $GF(2^m)$ field is characterized by polynomial like:

$$P(x)=x^m+p_{m-1}x^{m-1}+...+p_1x+p_0 \tag{2}$$

With p_i an element of {0, 1}. For given *m*, there are probably more than one polynomial of degree *m* [4][5]. All 2^m elements in the field can be represented by means of the vector basis $\{\alpha^{m-1},...,\alpha^1,\alpha^0\}$, where α is a root of the polynomial *p(x)* and is called a primitive element of the field . This vector basis allows an element A in $GF(2^m)$ to be expressed as:

$$A = a^{m-1}\alpha^{m-1} + ... + a^1\alpha^1 + a^0\alpha^0 \tag{3}$$

With α_i an element from $GF(2\ )$. $GF(2^m)$ can be associated with polynomials of degree m-1 that have coefficients in binary form . For example, A=(1010) in $GF(2^4)$ is shown polynomial with flow from:

$$1\alpha^3 + 0\alpha^2 + 1\alpha^1 + 0\alpha^0\ .$$

With notation that "addition" in $GF(2^m)$ is shown by bitwise XORing the *m* coefficients of the two polynomials being added. Multiplication is computed as modulo 2 sum of shifted partial products, where the sum is again computed using bitwise XORing. So the result for multiplication in $GF(2^m)$ is *(2m-1)* bits and give a *(2m-2)* degree's polynomial. Also, all $GF(2^m)$ fields are isomorphic for a given *m*.

III. EXTRACTION OF ROOTS

For simplicity, it is started with *m = 3*. Consider G(x) be a 3 degree polynomial binary field:

$$G(x) = x^3 + x + 1 \tag{4}$$

If α become a root of *G(x)* ,then it can be written :

$$\alpha^3 + \alpha + 1 = 0 \Rightarrow \begin{Bmatrix} \alpha^3 + \alpha = 1 \\ \alpha^3 + 1 = \alpha \\ \alpha + 1 = \alpha^3 \end{Bmatrix}$$

If α change to α^2, it can be proved, α^2 can be another root of *G(x)* because, $\alpha^2+\alpha^2=0 \quad and \quad \alpha+\alpha=0$ so $(\alpha^2+1)+(\alpha^2+1)+\alpha+\alpha=0$ and from (5) it can be written :

$$(\alpha^2)^3+(\alpha^2)+1=0 \tag{5}$$

So , α^2 can be one of the *G(x)* roots. It can be proved in the same way that α^4 is a root of *G(x)* too [6]. Now suppose that "B_i"s show an 8 bits block from data, then b_{ij} shows i^{th} bit from the j^{th} block. Also imagine that "C_k"s are the 8 bits blocks to check the sequence of B_i blocks. If any bit of these blocks i.e. b_w faces error, it converts to e_w. If system has had *n* data blocks and m check / control blocks, then it can be written:

$$B_1 \to B_n \qquad Data\ blockes.$$

$$C_1 \to C_m \qquad Check/Control\ blocks.$$

And the sequence of data packet can be demonstrated as figure 2 :

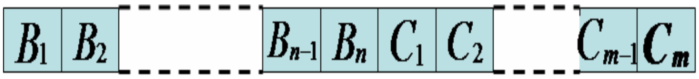

Fig. 2: Data packet sequence.

The aim is binary calculations (modulo – 2) and for $GF(2^8)$. The field is completed with two following operations:

1. Addition: XOR (exclusive – OR) with characteristics: 0+0=0 , 0+1=1 , 1+0=1, 1+1=0 (6)
2. Multiplications: 0*0 = 0, 0*1=0, 1*0=0, 1*1=1 (7)

Let's first start from $GF(2^3)$, because it is simple and extend to $GF(2^8)$ in coming parts. According to what is mentioned for following polynomial their roots are extracted: $P_1(x) = x^3 + x + 1$ (8)

It is shown that if α become one of the roots of the $P_1(x)$ then α^2,α^4 are other roots of the $P_1(x)$. In the same way that is mentioned above it can be proved that $\alpha^3,\alpha^5,\alpha^6$ are roots the $p_2(x)=x^3+x^2+1$ (9) too. And finally for $p_3(x)=0x^3+x+1=x+1$ (10)

$1=\alpha^7=\alpha^0$ is the root of $p_3(x)$.

Now it can be written: $p=p_1*p_2*p_3=(x^3+x+1)(x^3+x^2+1)(x+1)=(x^7+1)$ (11)

It can be seen the roots of *P* are: $\alpha^0,\alpha^1,\alpha^2,\alpha^3,\alpha^4,\alpha^5,\alpha^6$

The above procedure is the same with GEF[1] calculation.

With Matrices expression roots:

$$\alpha^1=\begin{vmatrix}0&0&1\\1&0&1\\0&1&0\end{vmatrix} \quad \alpha^2=\begin{vmatrix}0&1&0\\0&1&1\\1&0&1\end{vmatrix} \quad \alpha^3=\begin{vmatrix}1&0&1\\1&1&1\\0&1&1\end{vmatrix} \quad \alpha^4=\begin{vmatrix}0&1&1\\1&1&0\\1&1&1\end{vmatrix} \quad \alpha^5=\begin{vmatrix}1&1&1\\1&0&0\\1&1&0\end{vmatrix} \tag{12}$$

$$\alpha^6 = \begin{vmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{vmatrix} \alpha^7 = \begin{vmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{vmatrix} = \alpha^0 = 1$$

IV. APPLYING THE ALGORITHM ON A DATA FRAME

In the application discussing, consider a data frame like figure 3. Let's suppose each block has 3 bits.

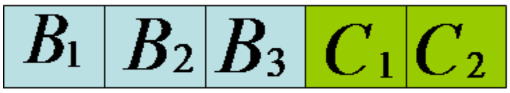

Fig. 3: Simple data packet sequence.

For given "B_i"s blocks, the "C_j"s is programmed in a way that satisfies the following equation:

$$B_1 + B_2 + B_3 + C_1 + C_2 = 0 = S_1 \quad (13)$$

Also equation (14) must satisfied: $\alpha^1 B_1 + \alpha^2 B_2 + \alpha^3 B_3 + \alpha^4 C_1 + \alpha^5 C_2 = S_2 = 0$ (14)

Equation (14) shows a type of polynomial with binary coefficient and application of Galois theory is vivid [7]. For example, consider "B_i"s and "C_j"s have the following values. Of course using computer program one can reach "C_j"s for example with parity role that the even number of "1"s are in each row. It should be mentioned that a new way is invented and applied by the author which uses Neural Network to extract the "C_j"s in coming works and is going to be presented. (Figure 4.)

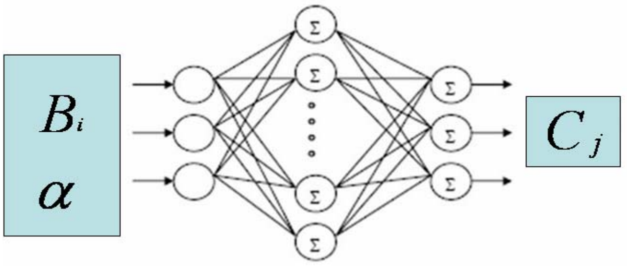

Fig 4 : Using a neural Network for extraction of "C_j"s

Now if an error occurs, for example on B_3 like following table:

TABLE I. Occurred errors

B_1	B_2	B_3	C_1	C_2
1	0	1	1	0
0	1	1	0	1
0	0	0	1	0

Then relation (14) and (15) are obtained: $S_1 = B_1 + B_2 + B_3 + C_1 + C_2 = \begin{vmatrix} 1 \\ 1 \\ 1 \end{vmatrix} \neq 0$ (14)

With uses (12):

$$S_2 = \begin{vmatrix} 0 \\ 1 \\ 0 \end{vmatrix} \neq 0 \quad (15)$$

Relations (14) and (15) shows the occurrence of a data frame error. For extraction of error location it can be written:

$\alpha^1 S_1 = \begin{vmatrix}1\\0\\1\end{vmatrix} \neq S_2$ so in B_1 block it doesn't occur any errors .

$\alpha^2 S_1 = \begin{vmatrix}1\\0\\0\end{vmatrix} \neq S_2$ so in B_2 block it doesn't occur any errors .

$\alpha^3 S_1 = \begin{vmatrix}0\\1\\0\end{vmatrix} = S_2$ so in B_3 block error is occurred, and for correction of "B_3"s error it can be written :

$$B_3 + S_1 = B_1 + B_2 + C_1 + C_2 = B_3 \tag{16}$$

(B_3 without error) . With noting that in $B_3 + S_1$ the error has been omitted because the statement of $B_1 + B_2 + C_1 + C_2$ was with out error .

V. CORRECTION OF ERROR BLOCK

According to what presented in last parts again in this case suppose B_3 block has one block error e.g. *E*.

So when error occurred B_3 block is changed to $B_3 + E$ block. So (14) and (15) equation face these changes:

$$B_1 + B_2 + (B_3 + E) + C_1 + C_2 = S_1 = E(\neq 0) \tag{17}$$

$$\alpha^1 B_1 + \alpha^2 B_2 + \alpha^3 (B_3 + E) + \alpha^4 C_1 + \alpha^5 C_2 = \\ S_2 = \alpha^3 E(\neq 0) \tag{18}$$

Then

$$\alpha S_1 \neq S_2 \\ \alpha^2 S_1 \neq S_2 \\ \alpha^3 S_1 = S_2 \Rightarrow B_3 \ \ block \ \ has \ \ error \tag{19}$$

Next S_1 added to B_3 block, in this case, equation (17) can be written is this form:

$$(B_3 + E) + S_1 = B_3 + E + E = B_3 \tag{20}$$

This means, *E* error block is corrected. One of the ways for calculation of "C_i"s blocks is presented below, an other way that author has used successfully, is Neural Network. From (13) and (14) with assumption of no error in Galois field it can be written: $\alpha^5 * S_1 + S_2 = 0$ (21)

so $(\alpha^5 + \alpha^1) B_1 + (\alpha^5 + \alpha^2) B_2 + (\alpha^5 + \alpha^3) B_3 + (\alpha^5 + \alpha^4) C_1 + (\alpha^5 + \alpha^5) C_2 = 0$ (22)

Because $\alpha^5 + \alpha^5 = 0$ it can be obtained: $(\alpha^5 + \alpha^1) B_1 + (\alpha^5 + \alpha^2) B_2 + (\alpha^5 + \alpha^3) B_3 + (\alpha^5 + \alpha^4) C_1 = 0$ (23)

So from (12) and (23) the following equations can be gained:

$$(\alpha^5+\alpha^1)=\begin{vmatrix}1&1&0\\0&0&1\\1&0&0\end{vmatrix}=\alpha^6$$

$$(\alpha^5+\alpha^2)=\begin{vmatrix}1&0&1\\1&1&1\\0&1&1\end{vmatrix}=\alpha^3 \qquad (24)$$

$$(\alpha^5+\alpha^3)=\begin{vmatrix}0&1&0\\0&1&1\\1&0&1\end{vmatrix}=\alpha^2$$

$$(\alpha^5+\alpha^4)=\begin{vmatrix}1&0&0\\0&1&0\\0&0&1\end{vmatrix}=\alpha^0=1$$

And finally from (23) and (24) it is obtained:

$$C_1=\alpha^6 B_1+\alpha^3 B_2+\alpha^2 B_3 \qquad (25)$$

Also by the save calculation (26) is gained:

$$C_2=\alpha^2 B_1+\alpha^3 B_2+\alpha^2 B_3 \qquad (26)$$

VI. THE DIMENSION EXTINCTION OF DATA BLOCK CAPACITY

N's order primitive polynomial have 2^n-1 roots, and these roots are all different from each other [8]. For example one of 8's order primitive polynomial is $G(x)=x^8+x^4+x^3+x^2+1$ (27)

Let's consider 8 order polynomials in general like (28):

$$P(x)=a_8x^8+a_7x^7+...+a_1x+a_0=\sum_{i=0}^{8}a_ix^i \qquad (28)$$

It can be showed α^1 is one of *P(x)* roots :

$$\alpha^1=\begin{vmatrix}0&0&0&0&0&0&0&a_0\\1&0&0&0&0&0&0&a_1\\0&1&0&0&0&0&0&a_2\\0&0&1&0&0&0&0&a_3\\0&0&0&1&0&0&0&a_4\\0&0&0&0&1&0&0&a_5\\0&0&0&0&0&1&0&a_6\\0&0&0&0&0&0&1&a_7\end{vmatrix} \qquad (29)$$

Now , if $P(x)=x^8+x^4+x^3+x^2+1$ (30)

Coefficients are $a_8,a_4,a_3,a_2,a_0=1$ and $a_7,a_6,a_5,a_1=0$ and α^1 for *P(x)* is :

$$\alpha^1 = \begin{vmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{vmatrix} \tag{31}$$

From α^1 to α^{255}, there are all different matrices use as *P(X)* roots . These matrices can be calculated with computer programming. A suggested algorithm to find for example $\alpha^i B_j$ for building the statement such as S_2, for prototype can be presented with the following method (with noting that α^1 is relative to polynomial (30)).

$$\alpha^1 * B_n =$$

$$\alpha^1 = \begin{vmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{vmatrix} * \begin{vmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{vmatrix} = \begin{vmatrix} b_7 \\ b_0 \\ b_1 + b_7 \\ b_2 + b_7 \\ b_3 + b_7 \\ b_4 \\ b_5 \\ b_6 \end{vmatrix} \tag{32}$$

This is important, because S_2 can be reached from VHDL programming and with implementing them in FPGA hardware [10].

VII. CONCLUSION

The current research illustrated the design error detection and corrects errors by Galois Field (GF) and binary polynomial and extraction of GF root. The process started from third root of GF and reach to eighth root of GF. The Galois Field was comprehensively explained along with it's requirement and relationship between digital and Galois theories. The Neural Network was used for extraction of some codes. Finally, the results shows successful design error detection and correction by employing Galois Field and binary polynomial and extraction of its root.

ACKNOWLEDGEMENT

Special thanks to Professor, Dr. Reza Naghipour and Image Processing Lab. in Iranian National Radio & TV Organization. Also, this research is funded and supported by Universiti of Sains Malaysia (USM). I would like to thank school of computer science to provide the facilities for this research.

REFERENCES

[1] R.Lidl and H.Niedrreiter, " Finite Fields (Encyclopedia of Mathematics and its Applications)", Combridge univ. press , 2008.

[2] Shu Lin, Daniel J. Costello, Jr. , " Error Control Coding: Fundamental and Application) " , Prentice Hall , 1982, 2004 .

[3] Hahn,M. "Channel codec performs versatile error-correction", Computers and Digital Techniques, IEEE Proceedings - Volume 137, Issue 3, 1990, 2005 Page(s): 197-201 ,Digital Object Identifier .

[4] Klappenecker , “Galois Theory and Wavelet Transforms”, Information Theory, 1995 IE International Symposium, pp. 429.

[5] www.maths.warwick.ac.uk .

[6] http://www.math.umn.edu/garrett .

[7] Popovici, E.M. Fitzpatrick, P., "Algorithm and architecture for a Galois field multiplicative arithmetic processor", Information Theory, IEEE Transactions on Volume 49, Issue 12, Dec. 2003 Page(s): 3303-3307Digital Object Identifier 10.1109/TIT.2003.820026.

[8] Guo and C. Wang ,"Systolic Array Implementation of Euclid's Algorithm for Inversion and Division in GF(2^m)", IEEE,ISCAS'96,vol. 2.pp. 481-4,1996.

[9] J.Garcia and M.J.Schulte, "A Combined 16-bit Binary and Dual Galois Field Multiplier",IEEE ISCAS 2002,pp.63-68.

[10] Wolfgang Wilhelm,Andre Kaufmann and Tobias G.Noll "A New Scalable VLSI Architecture for RS Decoders, Chair of Electrical Engineering and Computer Systems University of Technology , RWTH Aachen ,1998 .