IOSR Journal of Engineering (IOSRJEN) **www.iosrjen.org**
ISSN (e): 2250-3021, ISSN (p): 2278-8719
Vol. 08, Issue 7 (July. 2018), ||V (V) || 15-20

Comparative Analysis & Implementation of Galois Field Multiplier using Binary & One Hot Technique

Rajnish Ramesh Mishra
Electronics Engineering, B.D.C.E., Sevagram, Wardha
Bhalchandra Hardas
Electronics Engineering, R.C.O.E.M, Nagpur
Corresponding Author: Rajnish Ramesh Mishra

Abstract: In this paper we propose the Galois filed multiplier using Binary encoding technique & One hot Encoding technique. A Galois field multiplication method enables for an arithmetical operations including addition a deduction a multiplication and a multiplier utilizing the multiplication method. The Galois field multiplication method easily realizes various field multipliers by ANDing respective items of multiplier factor in a stepwise manner rotating left values resulted from the AND operation at the previous step Exclusively ORing the respective values resulted from the rotation with respective corresponding values resulted from AND operation at the current step and operating on the highest polynomial term generated at the previous step in accordance with a generated polynomial. This approach of Galois field can be used for designing the encoder and decoder section for the security purposes using the irreducible polynomial based on the NIST standard.

Date of Submission: 13-07-2018 Date of acceptance: 28-07-2018

I. INTRODUCTION

For every prime number p, there exists a Galois field, also known as the finite field, over the set GF(p) having p elements with special elements 0 and 1 as the additive and multiplicative identities, respectively. It is possible to extend the fields over GF(p) to a field that consists of pm elements, where m is a nonzero positive integer. This extended field over the set GF(pm) is known as the extension of the field over GF(p). Let "+" and "_" represent the addition and multiplication operations on the field elements. Then GF(pm) forms a finite field if it forms a commutative ring with identity over these two operations. The finite fields over GF(2) and their extensions over GF(2m) are used in digital logic owing to the field elements 0 and 1 only as well as their carry free logic and ease of implementation in hardware. The finite fields over GF(2m) can be generated with monic irreducible polynomials of the form:P(x) = xm+Σ∑m-1i=0 ci:xi, where ci 2 GF(2) . Other than elements 0 and 1 the field consists of primitive elements that are multiples of the element _, where _ is the root of P(x) i.e., P(_) = 0, and P(x) is the primitive polynomial of the field. To ensure that the operations over the fields are finite, any element in the field having power > (m 1) is reduced to an element with power < (m 1) by reducing it with P(x). The set of elements f0; 1; _; _2; : : : ; _m1g forms elements of the polynomial basis (PB) over a certain primitive polynomial. Any element A 2 GF(2m) is represented using the elements in PB. The polynomial basis multiplication of A(x) and B(x) over GF(2m) is defined using the following expression: C(x) = A(x) _ B(x) mod P(x) where A;B 2 GF(2m).

II. FIELDS

A field is an algebraic structure in which the operations of addition, subtraction, multiplication, and division (except by zero) can be performed, and satisfy the usual rules. More precisely, a field is a set F with two binary operations + (addition) and - (multiplication) are defined, in which the following laws hold:

(A1) a+(b+c) = (a+b)+c (associative law for addition)

(A2) a+b = b+a (commutative law for addition)
(A3) There is an element 0 (zero) such that a+0 = a for all a.
(A4) For any a, there is an element -a such that a+(-a) = 0.
(M1) a *(b * c) = (a * b) * c (associative law for multiplication)
(M2) a * b = b * a (commutative law for multiplication)
(M3) There is an element 1 (not equal to 0) such that a _ 1 = a for all a.
(M4) For any a ≠ 0, there is an element a-1 such that a*a-1= 1.

(D) a * (b+c) = (a * b)+(a * c)   (distributive law)
Using the notion of a group, we can condense these nine axioms into just three:

The elements of F form an Abelian group with the operation + (called the additive group of F). The non-zero elements of F form an Abelian group under the operation * (called the multiplicative group of F). Multiplication by any non-zero element is an auto morphism of the additive group. We usually write x * y simply as xy. Many other familiar arithmetic properties can be proved from the axioms: for example, 0x = 0 for any x. Familiar examples of fields are found among the number systems (the rational numbers, the real numbers, and the complex numbers are all fields). There are many others. For example, if p is a prime number, then the integers mod p forma field: its elements are the congruence classes of integers mod p, with addition and multiplication induced from the usual integer operations. For example, here are the addition and multiplication tables for the integers mod 3.
(We use 0;1;2 as representatives of the congruence classes.)

+	0	1	2
0	0	1	2
1	1	2	0
2	2	0	1

*	0	1	2
0	0	0	0
1	0	1	2
2	0	2	1

III. APPLICATION OF GALOIS FIELD

The implementation and examination of erasure codes in disk arrays, distributed storage systems and content distribution systems has been a common area of research within the systems community over the past few years. Most work is concerned with fault tolerant properties of codes, performance implications of codes, or both. Most of the erasure codes used in storage systems are XOR-based and generally provide limited levels of fault tolerance; a flood of special-purpose, XOR-based codes is the result of a performance-oriented push from the systems side [4, 2, 19]. While these codes perform all encoding and decoding using the XOR operator, they either lack flexibility in the number of tolerated failures or are not maximum distance separable (MDS) and may require additional program complexity. Linear erasure codes, such as Reed-Solomon [12], are MDS. As a result, Reed-Solomon codes provide flexibility and optimal storage efficiency. Unfortunately, Reed-Solomon codes are generally regarded as inefficient because encoding and decoding require Galois field arithmetic. Some effort has gone into alternative representations of Reed-Solomon codes. Arithmetic over field elements GF(2l) may be transformed into operations in GF(2), where multiplication is the bit-AND operation [3, 14]. While multiplication in a binary extension field is avoided, performance is heavily dependent on the choice of the code's generator matrix and the alternative representation results in additional program code complexity. Furthermore, the benefits of the XOR-based Reed-Solomon codes are generally effective when encoding large pieces of data. Compared to XOR-based Reed-Solomon and other special-purpose coding techniques, we believe that the use of Galois field arithmetic in linear codes leads to simple, generalized implementations. Threshold cryptography algorithms, such as Shamir's secret sharing algorithm [17], also rely on Galois fields for encoding and decoding. A random k-degree polynomial over a Galois field is chosen, where the zeroth coefficient is a secret to be shared among n participants. The polynomial is evaluated over n coordinates (shares), distributed among the participants. Polynomial interpolation is used to reconstruct the zeroth coefficient from any k+1 unique shares. The construction, evaluation and interpolation of the polynomial may also be done over Zp for some prime number p. Unfortunately, when dealing with large fields, the use of a suitable prime number may result in field elements that are not byte-aligned. Using Galois fields allows all of the field elements to be byte aligned. Another class of algorithms that use Galois field arithmetic is algebraic signatures [16]. Algebraic signatures are Rabinesque because of the similarity between signature calculation and the hash function used in the Rabin-Karp string matching algorithm [5]. The algebraic signature of a string s0, s1, . . . , sn−1 is the sum Σn−1i=0 siαi, where α and the elements of the string are members of the same Galois field. Algebraic signatures are typically used across RAID stripes, where the signature of a parity disk equals the parity of the signatures of the data disks. This property makes the signatures well-suited for efficient, remote data verification and data integrity in distributed storage systems. All of these applications make extensive use of Galois field multiplication, which is generally second to disk access as a performance bottleneck in a storage system that uses Galois fields. We describe methods aimed at improving general multiplication performance in the next two sections.

A. Abbreviations and Acronyms

GF : Galois Field
FPGA : Field Programmable Gate Array
MOSFET :Metal Oxide Semiconductor Field Effect Transistor
PMOS : Positive channel Metal Oxide Semiconductor
NMOS : Negative channel Metal Oxide Semiconductor
FIR : Finite Impulse Response
OHE : One Hot Encoding
MDS : Maximum Distance Separable
DSP : Digital Signal Processing
VHDL : VHSIC Hardware Description Language
VHSIC : Very High Speed Integrated Circuit
HPC : High Performance Computing

IV. CIRCUIT DESIGN OF GALOIS FILED MULTIPLIER USING BINARY AND ONE HOT TECHNIQUE

A. GF multiplier using Binary Technique

All the designs are made in S- Edit tool of Tanner EDA, In making of all the circuits we had used the MOSFETS (i.e. PMOS & NMOS).

To design multiplier using binary technique we need total 240 nos. of MOSFET's.

Below is the input given to design circuit
VA0 a0 Gnd 5
VA1 a1 Gnd 0
VA2 a2 Gnd 5
VA3 a3 Gnd 0
i.e. VA : 0 1 0 1 (5)

VB0 b0 Gnd 0
VB1 b1 Gnd 5
VB2 b2 Gnd 0
VB3 b3 Gnd 0
i.e. VB : 0 0 1 0 (2)

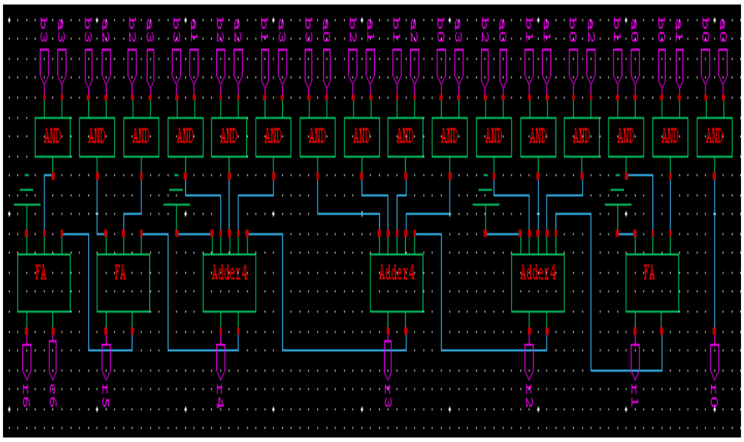

Fig1: GF multiplier using Binary encoding technique

B. GF multiplier using One Hot technique

All the designs are made in S- Edit tool of Tanner EDA, In making of all the circuits we had used the MOSFETS (i.e. PMOS & NMOS).

To design multiplier using binary technique we need total 712 nos. of MOSFET's.

Below is the input given to design circuit
VA0 a0 Gnd 5
VA1 a1 Gnd 5

VA2 a2 Gnd 0
VA3 a3 Gnd 0
i.e. VA: 0 0 1 1 (3)
VB0 b0 Gnd 0
VB1 b1 Gnd 0
VB2 b2 Gnd 5
VB3 b3 Gnd 0
i.e. VB: 0 1 0 0 (4)

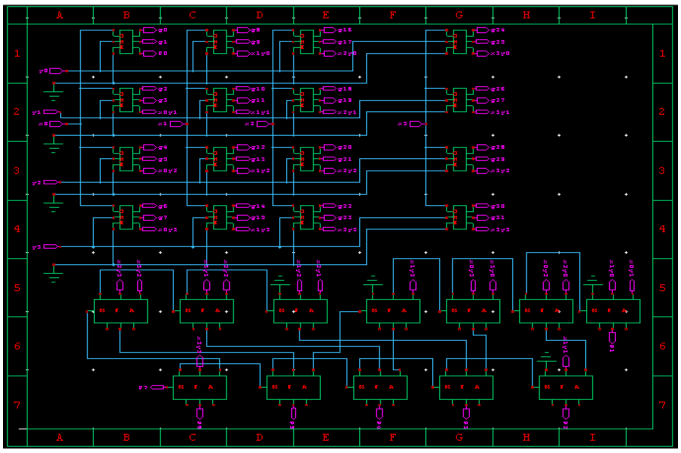

Fig2: GF multiplier using One hot Encoding Technique

C. Output Waveform of multiplier using Binary Technique

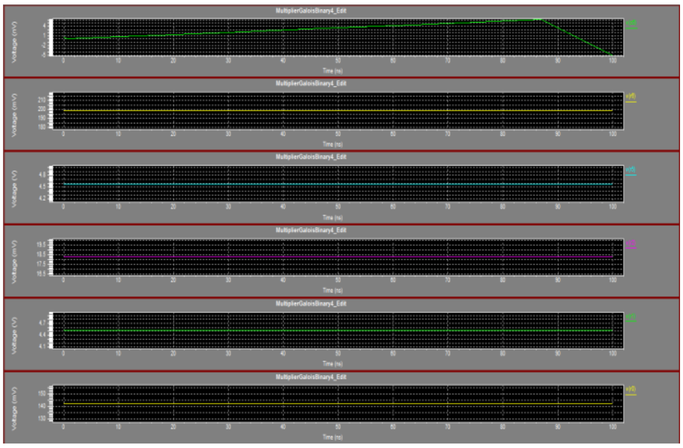

Fig3: output waveform of multiplier using Binary technique

Now the output for $V_{(c6)}$ $V_{(r6)}$ $V_{(r5)}$ $V_{(r2)}$ $V_{(r1)}$ $V_{(r0)}$ is 0 0 1 0 1 0. So that last four digit is giving our result i.e. 1 0 1 0 (10).

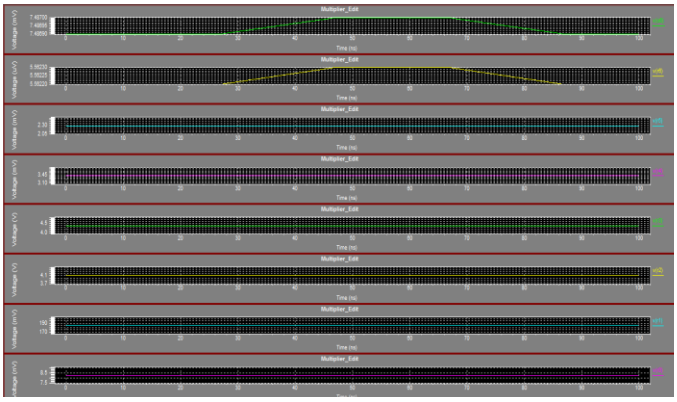

Fig4: output waveform of binary multiplier using One hot technique

The output here for multiplier $V_{(c6)}$ $V_{(r6)}$ $V_{(r5)}$ $V_{(r4)}$ $V_{(r3)}V_{(r2)}$ $V_{(r1)}$ $V_{(r0)}$ is 0 0 0 0 1 1 0 0 . The last four digit output is 1 1 0 0 (12).

V. COMPARISON OF RESULT BETWEEN GF MULTIPLIER USING BINARY & ONE HOT TECHNIQUE

In Multiplier using Binary technique we had used the total 240 MOFEST's i.e. 120 NMOS & 120 PMOS.

AREA = Nos.of PMOS x W x L x 3+ Nos. of NMOS x W x L
= 120 x 1800 x 180 x 3+ 120x1800x180
= 116640000 + 38880000
= 50520000 Nm^2

So the total area is 50520000Nm^2.

In Multiplier using One Hot technique we had used the total 712 MOFEST's i.e. 356 NMOS & 356 PMOS.

AREA = Nos.of PMOS x W x L x 3+ Nos. of NMOS x W x L
= 356 x 1800 x 180 x 3+ 356 x1800 x 180
= 346032000 + 115344000
= 461376000 Nm^2

Parameters	**Multiplier using Binary technique**	**Multiplier using One Hot Technique**
Avg. Power required (mWatt)	2.6	19.45
Area required (nM^2)	50520000	461376000
Delay in response time (nSec)	12.77	20.25

Table1: Comparison of multiplier using binary & One Hot Technique

VI. ACKNOWLEDGMENT

With all my heart I thank my supervisors Professor Bhalchandra Hardas for introducing me to this wonderful world of Finite field (Galois Field) and more so for making amply available their immense support, advise and encouragements in a number of ways. Along with them I thank Dr. M.A. Gaikwad, Principal of B.D.C.E., Sevagram, Wardha, Professor G. Wajurkar, Professor V.R. Ingle, Head of Electronics Department, & Professor K.D. Patil, Dean of R & D department of B.D.C.E. for his continuous support in my research carried out since 2013.

My father Ramesh Mishra has always been my greatest inspiration. I owe him my heart filled benediction for guiding me through a path of knowledge and truth, for being the strongest support in the journey towards my dreams and aspirations.

I thank my mother for her unconditional consecration in making a happy and adorable home, and for keeping me in a healthy mental and physical state. I thanks to my little sister for his love & I also thankful to all of my relatives & friends for their love and support.

Finally, I thank my wife. I thank you dear for your immutable patience and love, for carving a blissful end to each of my tiring days. Your emotional and moral support pulled me through this journey. I am grateful to God for having you by my side forever.

REFERENCES

[1]. K. Truong, I. S Reed and M. T. Shih published paper n Efficient multiplication algorithm over finite fields GF(qm)

[2]. L.K. Hua, "Introduction to Number Theory", Berlin, Heidelberg and New York; Springer-Verlag, 1982

[3]. R. Lidli and H. Niederreiter, published book on "Introduction to finite Fields and Their Applications: Cambridge Univ. Press, New York, USA, 1986

[4]. R. Conway, T. Conway and J. Nelson published paper on "New one- hot RNS structures for high-speed signal processing," Proc. SPIE Int. Symp. Optical Science and Technology, Seattle, WA, 2002.

[5]. FPGA Implementation of an Efficient Multiplier over Finite Fields GF(2m) Proceedings of the 2005 International Conference on Reconfigurable Computing and FPGAs(ReConFig 2005)0-7695-2456-7/05 © 2005 IEEE

[6]. P. Kitsos, G. Theodoridis y O. Koufopavlou. "Anefficient reconfigurable multiplier for Galois field GF (2m)". Microelectronics Journal. Vol. 34, Pags.975-980, 2003.

[7]. A. Daly y W. Marnane. "Efficient Architectures for implementing Montgomery Modular Multiplication and RSA Modular Exponentiation on Reconfigurable Logic".FPGA´02. Monterey, Ca. USA, 2002.

[8]. G. Bertoni, J. Guajardo, S. Kumar, G. Orlando, C. Paar y T. Wollinger. "Efficient GF(pm)Arithmetic Architectures for Cryptographic Applications". In Marc Joyce (Ed.): The Cryptographers´ Track at the RSA Conference CT-RSA 2003, volume LNCS 2616, pp. 158-175, San Francisco, CA, USA, April 2003.

[9]. G.C. Ahlquist, B. Nelson y M. Rice. "Optimal Finite Field Multipliers for FPGA´s". In P. Lysaght, J.Irvine, R. Hartenstein (Eds.): Field Programmable Logic and Applications. 9th International Workshop,FPL´99,volume LNCS 1673, pp. 51-60, Glasgow, UK,August/September 1999.

[10]. S. Lin y D.J. Castello. "Error Control Coding, Fundamentals and Applications", Prentice Hall, New Jersey, 1983.

[11]. F.J. Mac Williams, N.J.A. Sloane, "The theory of error correcting codes", North-Holand, 1977.

[12]. L. Song y K.K. Parhi, "Efficient Finite Field Serial/Parallel Multiplication", Proc. of International Conf.On Application Specific Systems, Architectures and Processors, pp. 72-82, Chicago, USA, 1996.

[13]. Recommended Elliptic Curves for Federal Government Use. ://csrc.nist.gov/CryptoToolkit/dss/ecdsa/NISTR eCur.pdf

[14]. J.Lópezand R. Dahab. "Fast Multiplication on Elliptic Curves over GF (2m) without Precomputation". In C.K. Kocand C. Paar (Eds.): Cryptography Hardware and Embedded Systems, CHES 1999, LNCS, Springer-Verlag, pp. 316-327, 1999.

[15]. E. Savas, A.F. Tenca and C.K. Koc. "A Scalable and Unified Multiplier Architecture for Finite Fields GF (p) and GF (2m)". In C.K. Koc and C. Paar (Eds.): Cryptography Hardware and Embedded Systems, CHES 2000, LNCS, Springer-Verlag, pp. 277-292, 2000.

[16]. F. Rodríguez-Henríquez. New Algorithms and Architectures for Arithmetic in GF (2m) Suitable for Elliptic Curve Cryptography. PhD Thesis, Oregon State University, 2000 Proceedings.

[17]. Lawrence C. Washington, Elliptic Curves: Number Theory and Cryptography, Chapman & Hall/CRC, New York, 2003.

[18]. A. J. Menezes, I. F. Blake, X. Gao, R. C. Mullin, S. A. Vanstone, and T. Yaghoobian, Applications of Finite Fields, Kluwer Academic, 1993.

[19]. C. K. Koc and T. Acar, "Montgomery Multiplication in GF(2k)", in Proc. Of Third Annual Workshop on Selected Areas in Cryptography, Ontario Canada, August 1996, pp. 95 – 106.

[20]. Keshab Parhi, VLSI Signal Processing Systems: Design and Implementation, John Wiley & Sons, 1999.

[21]. Huapeng Wu, "Montgomery Multiplier and Squarer for a class of Finite Fields", IEEE Transactions on Computers,, vol. 51, no. 5, pp 521 – 529, May 2002.

[22]. B. Sunar and C. K. Koc, "Mastrovito Multiplier for All Trinomials", IEEE Transactions on Computers, vol. 48, no. 5, pp 522 – 527, May 1999.

[23]. C. Grabbe, M. Bednara, J. Shokrollahi, J. Teich and J. Von ZurGathen, " FPGA Designs of parallel high performance GF(2233) Multipliers", In Proc. of the IEEE International Symposium on Circuits and Systems (ISCAS-03), volume II, pp 268 271.

[24]. Al-Somani, T.F., M.K. Ibrahim and A. Gutub, 2006. Highly efficient elliptic curve crypto-processor with parallel GF (2m) field multipliers. J. Comput. Sci., 2: 395-400.

[25]. K. K. Parhi, VLSI Digital Signal Processing System. New York: John Willey & Sons. 1999.
[26]. G. Ahlquist, B. Nelson, and M. Rice, "Optimal Finite Field Multipliers for FPGAs", Proceedings of the 9th International Workshop on Field-Programmable Logic and Applications, Lecture Notes In Computer Science, volume 1673, pp.51-60, Springer-Verlag, London, UK, 1999.
[27]. T. Akutsu, S. Miyano, and S. Kuhara. "Identification of Genetic Networks from a Small Number of Gene Expression Patterns Under the Boolean Network Model",In Pacific Symposium on Biocomputing, Volume 4, pp. 17-28, 1999.

Rajnish Ramesh Mishra."Comparative Analysis & Implementation of Galois Field Multiplier using Binary & One Hot Technique" IOSR Journal of Engineering (IOSRJEN), vol. 08, no. 7, 2018, pp. 15-20.

IOSR Journal of Engineering (IOSRJEN) is UGC approved Journal with Sl. No. 3240, Journal no. 48995.

Rajnish Ramesh Mishra "Comparative Analysis & Implementation of Galois Field Multiplier using Binary & One Hot Technique." IOSR Journal of Engineering (IOSRJEN), vol. 08, no. 2, 2018, pp. 00-00.