International Journal of Engineering & Technology, 7 (2.12) (2018) 386-389

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research Paper

Implementation of reconfigurable galois field multipliers over2m using primitive polynomials

B. Raj Narain [1]*, Dr. T. Sasilatha [2]

[1] Research scholar, Anna University,Chennai, India
[2] Dean, Department of Electrical and Electonics (Marine), AMET university, Chennai
**Corresponding author E-mail: rajnarainb@gmail.com*

Abstract

The Galois field multiplier finds extensive use in cryptographic solutions and applications. The Galois field multiplier can be implemented as fixed bitwise or reconfigurable. For fixed length, the data is restricted to the fixed length. But in reconfigurable GF multipliers, the bit length of the multiplier is flexible and is independent of hardware architecture. This paper proposes a method to implement a reconfigurable GF multiplier for various bit values from 8 to 128 bits. This paper compares the area complexity of various bit size in Xilinx Spartan 3E family FPGA and estimates the resources required for the implementation.

Keywords: *Galois Field; Reconfigurable; Primitive Polynomial*

1. Introduction

The study of Galois Field which was entitled later as Evariste Galois, and also recognised as finite field, denotes to an arena in which there happens finitely various components. It is predominantly expedient in interpreting computer data as they are characterised in binary logic. That is, computer data comprise of blend of two numbers, 0 and 1, which are the modules in Galois field whose number of components is two. Demonstrating data as a vector in a Galois Field permits mathematical procedures to ascent data effortlessly and efficiently.

This Galois Field (GF) is extensively castoff in digital signal processing, cryptography and channel coding. There have been various works on planning cost effective encryption hardware used in battery based applications. Most work semphasis on area reduction and propagation delay or critical path. In [1],the writersadvise a low cost technique to device limited 8 bit multipliers. In [2], a completely rolled pipelined architecture on basis exchange un it was focused. A technique consuming composite field arithmetic to moderate hardware complexity in modular transposal above GF(2^m) was planned in [3]. In [4], a scalable multiplier involving of several processing components connected in a parallel manner was planned, where each PE comprises two w-bit Carry save adders. Its performance is reliant on tractability of performance /area trade-off in numerous applications There are two bases used to represent the finite field elements over GF (p^m), namely normal basis and polynomial basis.

2. Previous works

In the collected works, numerous algorithms and hardware architectures for the regularsource multiplication over GF (2^m) are offered.The hardware execution of finite field multipliers expending normal basis can be categorized into three strategies.

The first technique is established on bit level execution [5]–[8]. In [5], Massey and Omura conceived a bit-level GF (2^m) standard basis multiplier. In that circumstance, a bit-level multiplier receipts m clock cycles to compute one multiplication above a binary field of size m. In [6], Gao et al. offereddevelopments over the Massey-Omura multiplier and condensed the area and power depletion. In [7], Beth et al. offeredapproaches for VLSI execution of public key algorithms and finite field arithmetic. In [8], Agnew et al. offered an execution of the exponentiation in GF (2^m), where its foremost disadvantage is the implementationperiod to achieve the multiplication for huge field proportions. Nearly some of the workings [9] are executed in bit serial or bit parallel to condense area intricacies reliant on the hardware.

The secondary one is grounded on a parallel-level execution [9]–[12]. In [9], [11] and [12], it is instigated a similarform of the Massey-Omura multiplier by confiscating the dismissal. In [9], Sunar and Koçplanned a regular basis multiplier created on the canonical basis multiplier. In [9], theperiod complexity is a lesser amount of than those produced by All One Polynomials (AOP), and the quantity of XOR and AND gates is aroundpartialthan that of theMassey-Omura multiplier over GF (2^m). The bit serial multiplication erected on polynomial basis was scheduled in [20] [21]. Bit parallel multipliers are completed in [22], [23] and [26].

The bit-serial-multipliers prompt the outcome of two ‘m’ bit operands subsequently ‘m’ clock cycles (latency) [27] and [28] .The bit-serial multipliers deal a benefit of reduced hardware means of intakes. The detriment is the greater quantity of essential clock cycles (latency) which consequences lower presentation. Owing to the less significant hardware operation, bit-serial multipliers are often applied for inhibited requests, where low extent is the important prerequisite.

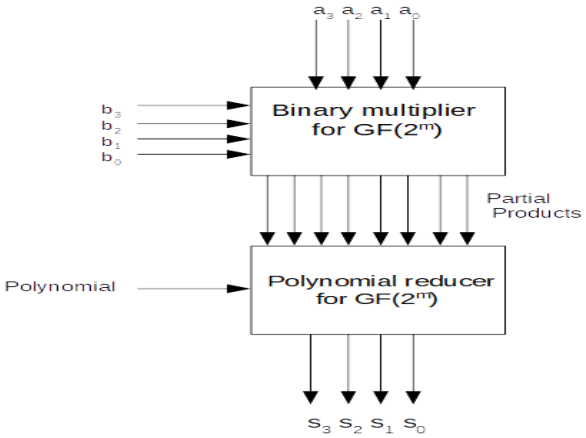

Fig. 1: Structure of a 4 Bit Galois Field Multiplier and Polynomial Reducer.

Byemploying bit-parallel multipliers, multiplication of two 'm' bit operands are considered at the similarperiod for application and the outcome is instantaneously computed [29], [30], [31], [32], [33] and [34].
This requires a supplementary clock base when associated to parallel multipliers subsequently the technique is approved ready on a bit by bit foundation. Numerous procedures are approved out in parallel in bit parallel procedure multipliers and this declines the latency above. Henceforward to progress the rapidity of the multiplier, a multimode multiplier is strategic which attains more than two procedures and this increases the quickness of procedure. Therefore an exact circuit is planned discussing to different GF lengths.
The thirdlymethod is created on a word-level execution, which is the furthermostnormally used [13]–[18]. In this method, a word-level multiplier receipts w (or d) clock cycles, where $1 \leq w \leq m$, to estimate one multiplication process over a binary field of size m. The assessment of w can be designated to attain the best trade-off amid area and time. Though, all the overhead works are very exact designs over GF (2^m), that is, if the m size variations aninnovativestrategy is necessary.
In interpretation of the varied field demos of GF (2^m), incalculable bit serial and digit serial multipliers has stayedstrategic and they have to conciliation in exhibition or over positioned as well as in theregion.
In order to abbreviate the time of forecasting but increase applicability, this paper proposes the Reconfigurable Galois Field multiplier which is relevant in manysolicitations.

3. Galois fields

A field is a customary of elements with two practicedemarcated arithmetic procedures: furthermostusually, addition and multiplication. Thus theelements of the field are called as an additive abelian assembly, and the non-zero elements of the field are called as a multiplicative abelian assembly. These resources that all elements of the field have an additive inverse and all non-zero elements have a multiplicative inverse. A field is also called finite if it certainly has a finite quantity of elements.
The elements of Galois Field GF (p^n) is defined as

GF (p^n) = (0, 1, 2,……., p-1) U(p, p+1, p+2,……., p+p - 1) U

(p^2, p^2 + 1, p^2 + 2,……., p^2+p-1)U..…. U (p^{n-1}; p^{n-1} + 1; p^{n-1} + 2,……, p^{n-1}+p-1)

Where the p∈P and n∈Z+. The order of the field is given by p^n while p is called the characteristic of the field. On the other hand, gf, as one may have guessed it, stands for Galois Field. Also note that the degree of polynomial of each element is at most n - 1.
4. Galois Field Multiplication
A circuit effects regular multiplication of two field elements in a Galois field GF (2^m). Each of the field elements is expressed by an m-bit binary number. The two field elements are applied to a binary multiplier array which generates 2m-1 bit partial products. The partial products are divided by a generator polynomial of the Galois field to produce final m-bit binary products.
Let a(x) and b(x) be two field elements and s(x) be their product. Then,

$$s(x) = a(x)b(x) \bmod p(x) \quad (1)$$

Thus, the polynomial basis multiplication involves two steps: polynomial multiplication and reduction modulo an irreducible polynomial as shown in figure 1 for a 4 bit multiplier. The product d(x) of the polynomials representing the field elements a(x) and b(x), d(x) = a(x) b(x), is a degree 2m – 2 polynomial. In the modular reduction s(x) = d(x) mod p(x), the degree 2m – 2 polynomial d(x) is reduced by the degree m irreducible polynomial p(x) iteratively. The choice of the irreducible polynomial p(x) may ease the modular reduction. Sparse irreducible polynomials having fewer nonzero terms are usually preferred for efficiency.
Example for Galois field Multiplication:

If A = x^3+x^2+1 (1101_2 in binary) and B =x^2+x (0110_2 in binary) are two polynomials, then A.B is called polynomial multiplication which returns $x^5+x^3+x^2+x$, if m = 4. The result should be reduced to a degree less than 4 by irreducible polynomial x^4+x+1.
Therefore

A.B mod P = $x^5+x^3+x^2+x$ (mod p(x))

= $(x^4+x+1)\, x+ x^5+x^3+x^2+x$

= $2x^5+x^3+2x^2+2x$

= x^3 (after reducing the coefficient on mod 2)

A = 1101_2

B = 0110_2

A*B = 1000_2

For implementing the modular multiplication the following irreducible or primitive polynomials are considered.

Table 1: Primitive Polynomials

m	Polynomial p(x)
4	$1+x+x^4$
8	$1+x^2+x^3+x^4+x^8$

This paper proposes that Reconfigurable Galois field multiplier which is applicable at for length GF multiplication by changing primitive polynomial for different GF. Actually, the multiplier consists of AND gates and special cells as shown in figure 2, whichconsist of product and reduction. The distinct cell estimates partial product and partial reduction at the sametime which is similar to binary product and division according toa(x).b(x) mod p(x). Adistinct cell that receives 2 multipliers, precedent result (pr), primitive polynomial (pp), quotient (q), and produces an output

$$S= a.b \oplus pp.pr. \quad (2)$$

For GF (2^4) themultiplier cellconsists of 4 AND gates and 12 special cells, which has rows similar to Linear Polynomial Reduction and the dotted line is quotient to use in the next row. Distinct cells are according to a relationship of wiring, for example equation 2 can be written as below:

$$s_{ij} = a_{m-j} \,.\, b_{m-i} \oplus p_{m-j} \,.\, s_{i-1,1} \oplus s_{i-1,j+1} \quad (3)$$

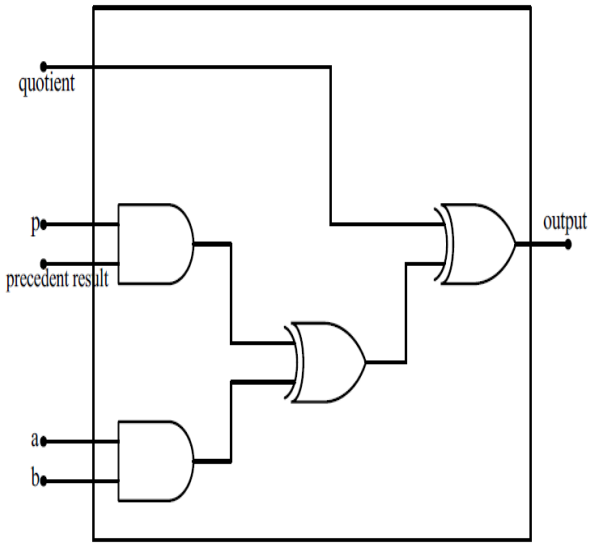

Fig. 2: Special Cell of Reconfigurable Galois Field Architecture.

4. Reconfigurable galois field multipliers

In order to randomly calculate different Galois Field multiplication, one of the methods is to include an additional Galois Field length controller. The Galois Field length controller makes GF multiplier architecture to receive different irreducible polynomials. The GF length controller shown in figure 3 consists of memory and multiplexer which receives a control signal and gives the primitive polynomial.

The memory of GF length controller initialises the primitive polynomials and connects multiplexer to choose another primitive polynomial. The control signal is the selecting of multiplexer which connects the input to the output. Galois Field multiplier arrangement modifies the data path allowing to primitive polynomial. The user adds the primitive polynomial to the memory rendering to the need of GF length, while the Galois Field multiplier arrangementcovers the largest Galois Field length. For example when the length of the Galois Field is m = 8 it can determine any Galois Field multiplication for m ≤ 8. The memory initialises primitive polynomial for m=4 to m=8 and The Galois Field multiplier structure expands to 8x8.

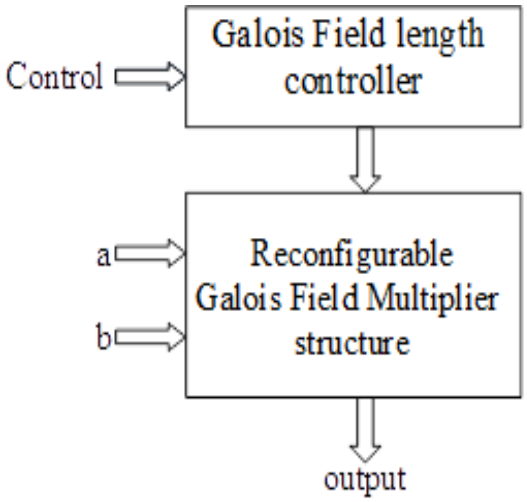

Fig. 3: Reconfigurable Galois Field Multiplier with Length Controller.

5. Simulation result

The multiplier has been implemented for 4 and 8 bits for the polynomial$1+x^2+x^3+x^4+x^8$.Even though various polynomials are used for reducing the result of the Galois field multiplication, the output result may vary but is limited to the m value. The multipliers are implemented in the Xilinx IDE for the devices Spartan 3E. The results shown in figure 4 and 5 were obtained using Xilinx ISE tool by synthesising for the Spartan 3E family's device XC3S500E. The synthesis results in table 2 shows the number of Lookuptablesand slices required for applyingthe 8, 16, 32 and 64 bit multipliers over Galois Field of 2^m. The table 3 compares the number of slices required to implement 8 and 16 bit multipliers of this work with [25] and [26] .Thus we can observe that the number of look up tables and slices required for different m values are increasing.

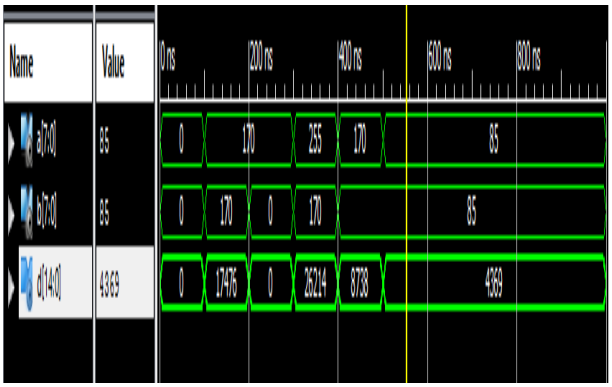

Fig. 4: Parallel Multiplication without Reducing.

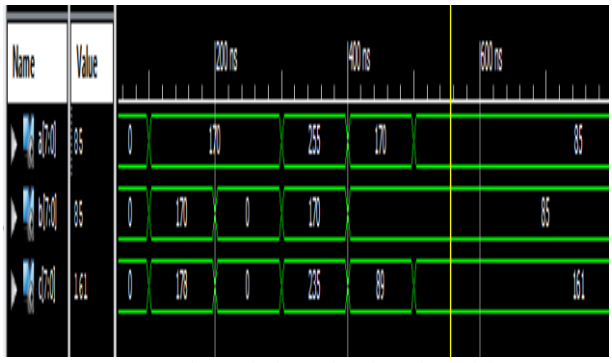

Fig. 5: Multiplication after Reducing Using the Polynomial "00011011".

Table 2: Comparison of Slices and LUT for Various Values of M in Galois Field Multiplier

S. No.	Bit size	Slices	LUT	Estimated Power (mW)	Combinational path delay (ns)
1	8	28	53	79	10.215
2	16	101	197	76	10.669
3	32	407	800	196	13.007
4	64	1584	3139	196	14.538

Table 3: Comparison of Slices for Various Values of M in Galois Field Multiplier

S. No.	Bit size	[25]	[26]	This work
Device		Xilinx-Virtex -V100FG256	Xilinx-Virtex -V100FG256	Xilinx – XC3S500E
1	8	30	43	28
2	16	105	153	101
3	32	-	-	407
4	64	-	-	1584
5	128	-	-	9168

6. Conclusion

The synthesis is completed for8 to 64 bits using the same algorithm .The power is also estimated at an ambient temperature of 25ºC. It is observed that if the temperature of the atmosphere increases, the leakage current and subsequently the power consumed also increases. In addition to that, due to the absence of clock input in the synthesis, the delay occurring due to clock input is reduced considerably. Since the output is unique for different polynomials, it can be used as a key in cryptographic applications without much change in the hardware .The reconfigurable Galois field multiplier

can also be replaced with other methods such as Karatsuba – Ofman algorithm, Montgomery multiplier for further reduced hardware and low power consumption when more number of bitsis used.

References

[1] Chelton.W and Benaissa.M,"Concurrent Error detection in GF (2m) multiplication and its application in elliptic curve cryptography" IET Circuits Devices Systems,vol.2, No.3, pp.289-297,2008.

[2] Chiou .C.W. , Lee C.Y., Deng. A.W. and Lin J.M, "Concurrent error detection in montgomery multiplication over GF (2m)," IEICE Trans.fund., vol.E89A no.2 ,pp.566-574, 2006.

[3] Harris.D, Krishnamurthy.R, Anders.M,Mathew.S and Hsu .S, "An improved unified scalable radix-2 montgomery multiplier", in Proc.17th IEEE Symp. Computer Arithmetic, pp.172-178 ,2005.

[4] Huang. K.H. and Abraham.J.A.,"Algorithm based fault tolerance for matrix operations", IEEE Transactions on Computers,.vol.33, no.6,pp 518-522, 1984.

[5] J. L. Massey and J. K. Omura, "Computational method and apparatus for finite field arithmetic," U.S. Patent 4 587 627, May 6, 1986.

[6] L. Gao and G. E. Sobelman, "Improved vlsi designs for multiplication and inversion in GF (2M) over normal bases," in Proc. 13th Annu. IEEE Int. ASIC/SOC Conf., Sep. 2000, pp. 97–101.

[7] T. Beth and D. Gollman, "Algorithm engineering for public key algorithms," IEEE J. Sel. Areas Commun., vol. 7, no. 4, pp. 458–466, May 1989.

[8] G. Agnew, R. Mullin, I. Onyszchuk, and S. Vanstone, "An implementation for a fast public-key cryptosystem," J. Cryptol., vol. 3,no. 2, pp. 63–79, 1991.

[9] A. Reyhani-Masoleh and M. Hasan, "A new construction of Massey-Omura parallel multiplier over GF(2M)," IEEE Transactions on Computers ,vol. 51, no. 5, pp. 511–520, May 2002.

[10] C. K. Koc and B. Sunar, "Low-complexity bit-parallel canonical and normal basis multipliers for a class of finite fields," IEEE Transactions on Computers, vol. 47, no. 3, pp. 353–356, Mar. 1998

[11] M. A. Hasan, M. Z. Wang, and V. K. Bhargava, "A modified Massey-Omura parallel multiplier for a class of finite fields," IEEE Transactions on Computers, vol. 42, no. 10, pp. 1278–1280, Oct. 1993.

[12] H. Wu and M. A. Hasan, "Low complexity bit-parallel multipliers for a class of finite fields," IEEE Transactions on Computers, vol. 47, no. 8, pp. 883–887, Aug. 1998.

[13] A. Reyhani-Masoleh and M. A. Hasan, "Efficient digit-serial normal basis multipliers over binary extension fields," Trans. Embedded Comput. Syst., vol. 3, no. 3, pp. 575–592, Aug. 2004.

[14] A. H. Namin, H. Wu, and M. Ahmadi, "Comb architectures for finite field multiplication in F(2M)," IEEE Transactions on Computers, vol. 56, no. 7, pp. 909–916, Jul. 2007.

[15] A. H. Namin, H. Wu, and M. Ahmadi, "A new finite-field multiplier using redundant representation," IEEE Transactions on Computers, vol. 57, no. 5, pp. 716–720, May 2008.

[16] A. Reyhani-Masoleh and M. A. Hasan, "Low complexity word-level sequential normal basis multipliers," IEEE Transactions on Computers, vol. 54, no. 2, pp. 98–110, Feb. 2005.

[17] A. Reyhani-Masoleh, "Efficient algorithms and architectures for field multiplication using Gaussian normal bases," IEEE Transactions on Computers, vol. 55, no. 1, pp. 34–47, Jan. 2006.

[18] Y. S. Cho and J. Y. Choi, "A new world-parallel bit-serial normal basis multiplier over GF(2m)," Sci. Eng. Res. Support Soc., vol. 6, no. 3, pp. 209–216, Jun. 2013.

[19] Li. H. and Li .J, "A new compact architecture for AES with optimized shift rows operation," in Proceedings IEEE ISCAS, pp 1851-1854, 2007.

[20] L. Song and K.K. Parhi, "Efficient finite field serial/parallel multiplication," Proc. of International Conf. on Application Specific Systems, Architectures and Processors, Chicago, pp. 72-82, 1996.

[21] C.W. Chiou, C.Y. Lee and J. M. Lin, " Finite field polynomial multiplier with linear feedback shift register,"J. Sci. Eng., 2007, 10, (3), pp. 253–264.

[22] D. Pamula and A. Tisserand, " GF(2m) finite-field multipliers with reduced activity variations," In 4th International Workshop on the Arithmetic of Finite Fields, volume 7369 of LNCS, pp. 152-167, Bochum, Germany, July 2012.

[23] N. Iliev, J.E. Stine and N. Jachimiec, "Parallel Programmable Finite Field GF(2m) Multipliers," In Proc. IEEE Computational society Annual Symp. VLSI Emerging Trends (ISVLSI'04), pp. 299-302, Feb. 2004.

[24] M. Matsumoto and K. Murase, "Multiplier in a galois field, " U.S. Patent,No. US4918638 A, 1990.

[25] N. Iliev, J.E. Stine and N. Jachimiec, "Parallel Programmable Finite Field GF(2m) Multipliers," In Proc. IEEE Comp. Soc. Annual Symp. VLSI Emerging Trends (ISVLSI'04), pp. 299-302, Feb. 2004.

[26] H. Yi, S. Tang, and L. Xu, "A Versatile Multi-Input Multiplier over Finite Fields." IACR Cryptology ePrint Archive 2012: 545.

[27] H. El-Razouk and A. Reyhani-Masoleh, "New Bit-Level Serial GF(2m) Multiplication Using Polynomial Bases," 22nd IEEE Symposium on Computer Arithmetic, 2015, pp. 129–136.

[28] A. Reyhani-Masoleh, "A New Bit-Serial Architecture for Field Multiplication Using Polynomial Bases," In Cryptographic Hardware and Embedded Systems-(CHES 2008), vol. 5154 of the series Lecture Notes in Computer Science (LNCS), E. Oswald and P. Rohatgi (Eds). Springer Berlin Heidelberg, 2008, pp. 300–314.

[29] A. Chatterjee, I. Sengupta, "High-Speed Unified Elliptic Curve Cryptosystem on FPGAs Using Binary Huff Curves," Progress in VLSI Design and Test (VDAT), vol. 7373 of the series Lecture Notes in Computer Science (LNCS), pp 243–251, 2012.

[30] S. Ghosh, A. Kumar, A. Das and I. Verbauwhede, "On the Implementation of Unified Arithmetic on Binary Huff Curves," Cryptographic Hardware and Embedded Systems-(CHES 2013), vol. 8086 of the series Lecture Notes in Computer Science (LNCS), pp 349–364, 2013.

[31] M. Imran, M. Kashif and M. Rashid, "Hardware Design and Implementation of Scalar Multiplication in Elliptic Curve Cryptography (ECC) over GF (2163) on FPGAs," International Conference on Information and Communication Technologies (ICICT), Karachi, Pakistan, 2015, pp. 1–4.

[32] A. Sghaier, M. Zeghid, B. Bouallegue, A. Baganne, and M. Machhout, "Area–Time Efficient Hardware Implementation of Elliptic Curve Cryptosystem," International Journal of Computer Science and Information Security, vol. 14, no. 4, pp. 1–7, 2016

[33] L. Batina, J. Hogenboom, N. Mentens, J. Moelans, and J. Vliegen, "Side-channel Evaluation of FPGA Implementations of Binary Edwards Curves," Proceedings of the 17th IEEE International Conference on Electronics Circuits and Systems, 2010, pp. 1248– 1251.

[34] A. Chatterjee and I. Sengupta, "Performance Modelling and Acceleration of Binary Edwards Curve Processor on FPGAs," International Journal of Electronics and Information Engineering, vol. 2, no. 2, pp. 80–93, 2015.