



AN ENHANCED GALOIS FIELD MULTIPLIER APPROACH FOR LOW AREA AND HIGH SPEED OPERATIONS

P.Shamitha¹, A.Syam Kumar²

¹Pursuing M.tech (VLSI), ²Assistant Professor (ECE),

Nalanda Institute of Engineering and Technology (NIET), Siddharth Nagar, Kantepudi (V),
Sattenpalli (M), Guntur Dist, A.P. (India)

ABSTARCT

GF (2^m) multiplier is one of the arithmetic operation, which have more number of applications and domains, in that cryptography is the main application. One of the most popular cryptographic applications is GF (2^m) multiplier unit. ECC (elliptic curve cryptography) is also one of the cryptographic application it is having finite field extension to perform the operations in a system. In the finite field all the operations are represented in the form of sequences so this effects on the performance of area, speed, efficiency of the ECC system. The proposed paper contains modified structure of large number (160-600) efficient GF (2^m) modular multiplier, which increases the area-speed efficiency of the proposed design. For this we required combination of bit-parallel and bit-serial process if input variables. The proposed design is represented as a modular since no need to design again the required design if any changes in field side. In this we are representing two FPGA architectures, the proposed GF(2^m)architecture modular block and the multiplier article details, which effects on speed, area and how its parameters scale for different types of field sizes.

Keywords: Galois Field Multiplier, Irreducible Polynomial, Finite Field Extension.

I. INTRODUCTION

Finite field multipliers features are advantageous in its various domains: in computational biology and in commutative algebra, in the digital signal processing but mainly they serve on information theory (error correcting codes) and also in cryptography-security. Our attentiveness deceptions in the elliptic curve of security/crypto (ECC) applications and the solutions dedicated to reconfigurable of hardware (FPGA). There are two different types of finite fields used in ECC: prime fields GF(p) and the binary finite fields of extensions GF(2^m). In our research scenario we focus on GF(2^m) as those fields are to seem more suitable for hardware solutions. The Cryptographic applications are the one very demanding. They require the operations on large numbers, and the high efficiency in terms of high speed and area of the operators. In our research, we consider the ECC systems with parameters and defined in cryptographic standards, and issued e.g. by NIST. However, this paper, demonstrate a multiplier architecture, in which the modular can be easily rebuilt for the field sizes other than recommended by the NIST. The smallest all possible GF (2^m) multiplier, which can be built by

basing on the proposition is a 16-bit multiplier. The work of the paper implies some history related on binary finite fields which is regarding construction of multipliers. The multiplication in this area in Section-2. In Section-3 area, Galois Field (2^m), many algorithms on multiplications are modified which we base on the proposed modern architecture of the multiplier are introduced. In Section IV, area we present the detailed description of elaborated architecture and in area Section V. The detail and analyze obtained implementation results are showing how they are scaled for other field sizes.

II. MULTIPLICATION IN THE BINARY FINITE FIELD EXTENSIONS

Multiplication in the finite field is regarded as the complex operation. It requires reduction of modulo specific irreducible polynomial and in the cryptographic applications it requires the operating on large numbers. Then, additionally, due to the fact that we operate in the specific field we can't refer, in order to handle speed of the requirements of Galois field, expert in built-in hardware multiplier units. A complexity of finite field multiplier of architecture depends mostly on parameters of this field. Those parameters are the, representation of the elements of the field on basis, field size m and the irreducible polynomial $f(x)$ generating the field. Hence, the other commonly used bases are the normal bases and the dual basis. Due to the fact that proposed multipliers we should serve ECC applications, and the other two parameters (field size m and the irreducible polynomial $f(x)$), are the chosen regarding ECC standards. The most recent values that are presented in Table I.

field size m	irreducible polynomial $f(x)$
163	$f(x) = x^{163} + x^7 + x^6 + x^3 + 1$
233	$f(x) = x^{233} + x^{74} + 1$
283	$f(x) = x^{283} + x^{12} + x^7 + x^5 + 1$
409	$f(x) = x^{409} + x^{87} + 1$
571	$f(x) = x^{571} + x^{10} + x^5 + x^2 + 1$

Table : NIST recommended $GF(2^m)$ architecture

III. PROPOSED ARCHITECTURE

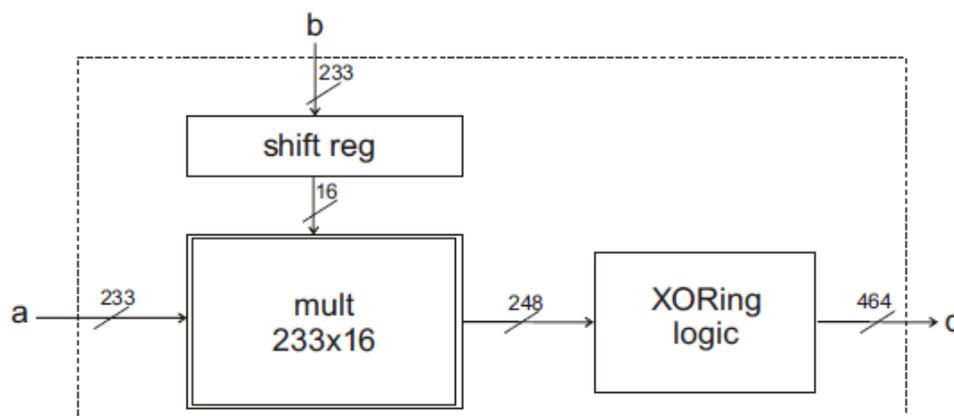


Fig: The proposed architecture of $GF(2^{233})$ multiplier



This paper presents the architecture implementation. This architecture was derived for $m = 233$, the field size we have chosen as an exemplary one. For such m we have assumed partitioning b into 16-bit words. With such partitioning we have to perform $dm/16e$ $ab[i]$ operations before obtaining the final product. Generally the architecture consists of two blocks: 233×16 -bit multiplier block and block containing Exoring logic. The sub-multiplier block performs $ab[i]$ operation. To perform $dm/16e$ operations it requires $dm/16e \times T_{mult} 233 \times 16$ cycles, where $T_{mult} 233 \times 16$ is the delay introduced by the 233×16 multiplier block. The fetching of consecutive words of vector b is controlled by a simple shift register.

The multiplier gives 248-bit partial results ($233 + 16 - 1 = 248$). Each 248-bit output of the multiplier is fetched to a block containing network of the XOR gates, which produces the final 464-bit ($2m-1$) result d . The sub-multiplier comprises AND and the XOR logic unit. The AND logic “multiplies” every bit of vector b by 16-bit word of a and then it fetches the result to a Exoring logic block, this is which is responsible for combining all the partial results into final 30-bit output. So, finally, all 30-bit outputs are to be merged to provide the result of the operation. The given general architecture of 233×16 -bit multiplier block is utilizing combinatorial logic sub blocks is presented in Figure 4. In this architecture, we have process both 16-bit words $a[k]$, $b[i]$ in the bit parallel manner, thus here $T_{mult} 16 \times 16 = 1$, is the means that we have to gain on speed of the solution. Every combinatorial 16×16 -bit sub-multiplier it implements parallels in the following equations:

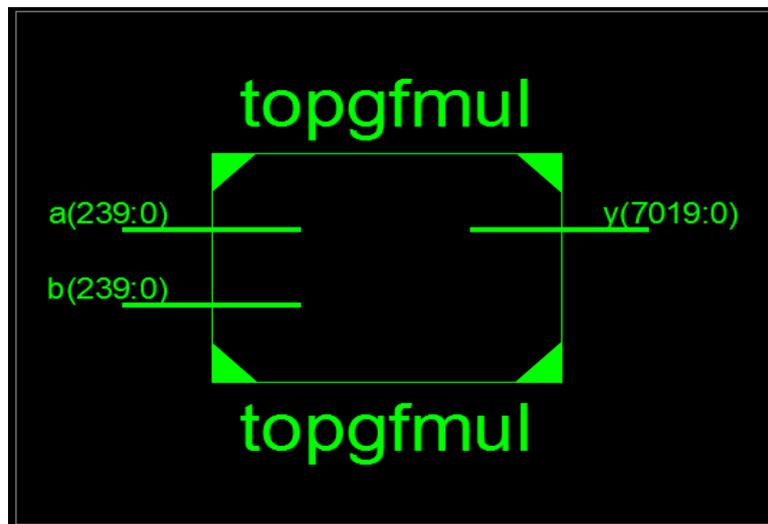
$$\begin{aligned}
 d_0 &= a_0 b_0; \\
 d_1 &= a_1 b_0 \oplus a_0 b_1; \\
 d_2 &= a_2 b_0 \oplus a_1 b_1 \oplus a_0 b_2; \\
 &\vdots \\
 d_7 &= a_7 b_0 \oplus a_6 b_1 \oplus a_5 b_2 \oplus a_4 b_3 \oplus a_3 b_4 \oplus a_2 b_5 \oplus a_1 b_6 \oplus \\
 &\quad a_0 b_7; \\
 &\vdots \\
 d_{15} &= a_{15} b_0 \oplus a_{14} b_1 \oplus a_{13} b_2 \oplus a_{12} b_3 \oplus a_{11} b_4 \oplus a_{10} b_5 \oplus \\
 &\quad a_9 b_6 \oplus a_8 b_7 \oplus a_7 b_8 \oplus a_6 b_9 \oplus a_5 b_{10} \oplus a_4 b_{11} \oplus a_3 b_{12} \oplus \\
 &\quad a_2 b_{13} \oplus a_1 b_{14} \oplus a_0 b_{15}; \\
 &\vdots \\
 d_{22} &= a_{15} b_7 \oplus a_{14} b_8 \oplus a_{13} b_9 \oplus a_{12} b_{10} \oplus a_{11} b_{11} \oplus a_{10} b_{12} \oplus \\
 &\quad a_9 b_{13} \oplus a_8 b_{14} \oplus a_7 b_{15}; \\
 &\vdots \\
 d_{28} &= a_{15} b_{13} \oplus a_{14} b_{14} \oplus a_{13} b_{15}; \\
 d_{29} &= a_{15} b_{14} \oplus a_{14} b_{15}; \\
 d_{30} &= a_{15} b_{15};
 \end{aligned}$$



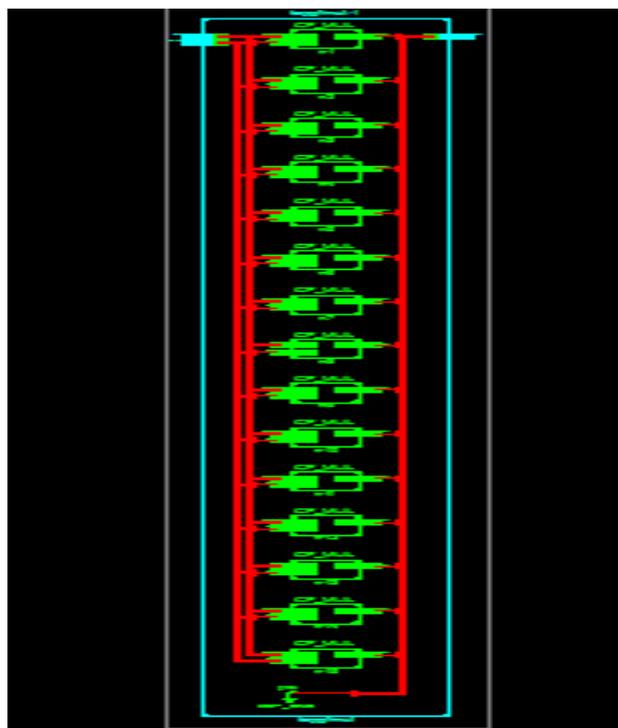
IV. SYNTHESIS AND SIMUALTION RESULTS

In this paper we aimed to design 233*233 multiplier in galios field. The proposed implementation was designed using verilog HDL and it is synthesized in XILINX 13.2 ISE. The synthesis results which are giving fpga mapped results of the multiplier. The synthesis results providing betterment in area propogation delay as compared with the 233*16 multiplier. The 233*16 multiplier will take much number of cycles due to less bitwidth but the propoese design have better performance in output. The synthesize and simualtion results as follows:

Top Level schematic block:



Second level schematic block:





Synthesis results:

topgfmul Project Status (10/30/2015 - 16:38:45)			
Project File:	gfnew.xise	Parser Errors:	No Errors
Module Name:	topgfmul	Implementation State:	Synthesized
Target Device:	xc7a30t-3csg324	• Errors:	
Product Version:	ISE 13.2	• Warnings:	
Design Goal:	Balanced	• Routing Results:	
Design Strategy:	Xilinx Default (unlocked)	• Timing Constraints:	
Environment:	System Settings	• Final Timing Score:	

Device Utilization Summary (estimated values)				
Logic Utilization	Used	Available	Utilization	
Number of Slice LUTs	1624	21000		7%
Number of fully used LUT-FF pairs	0	1624		0%
Number of bonded IOBs	721	210		343%

Detailed Reports					
Report Name	Status	Generated	Errors	Warnings	Infos
Synthesis Report	Current	Tue 3. Nov 17:18:32 2015			
Translation Report					
Map Report					

Timing constraint: Default path analysis
Total number of paths / destination ports: 7679 / 465

Delay: 2.003ns (Levels of Logic = 4)
Source: b<8> (PAD)
Destination: d<43> (PAD)

Data Path: b<8> to d<43>

Cell:in->out	fanout	Gate Delay	Net Delay	Logical Name (Net Name)
IBUF:I->O	237	0.001	0.831	b_8_IBUF (b_8_IBUF)
LUT6:I0->O	2	0.097	0.697	Mxor_d<43>_xo<0>2 (Mxor_d<12>_xo<0>1)
LUT6:I0->O	1	0.097	0.279	Mxor_d<12>_xo<0>5 (d_12_OBUF)
OBUF:I->O		0.000		d_12_OBUF (d<12>)
Total		2.003ns	(0.195ns logic, 1.808ns route)	(9.7% logic, 90.3% route)

Fig: Synthesis results

Simulation results:



Fig: Simulation results



IV. CONCLUSION

In this project we have designed 233×233 Galois field multiplier using with Verilog HDL, synthesis and simulation can be done by using XILINX 13.2 ISE simulator. It have better performance in area speed and power comparing with existing multiplier

REFERENCES

- [1] D. Hankerson Book , A. Menezes, and S. Vanstone, the Guide to Elliptic Curve Cryptography. The Springer, 2004.
- [2] The P. Gallagher, "FIPS PUB 186-3 of Federal data Processing Standards Publication Digitized Signature Standard (DSS)," 2009 and 2010.
- [3] The R. Lidl and H. Niederreiter, Introduction to the Finite Fields and Their Applications, The 2nd ed. Cambridge University Press, 1994 and 95.
- [4] R. McEliece from internet, Finite field for the scientists and engineers. Kluwer Academic of Publishers, 1987.
- [5] The S. S. Erdem, T. Yanik, and C. K. Koc, "Polynomial of Basis Multiplication over $GF(2^m)$," The Acta Applicandae Mathematicae, vol. 93, the no. 1-3, pp. 33–55, Sep. 2006 and 2007.
- [6] A. Reyhani-Masoleh and the M. Hasan, "The Low complexity bit parallel architectures for the polynomial basis multiplication over $GF(2^m)$," The IEEE Trans. Comput., vol. 53, no. 8, The pp. 945–959, Aug. Sep, 2004 .
- [7] "Bit-Parallel Polynomial on Basis Multiplier for the New Classes of Finite Fields," The IEEE Trans. Comput., vol. 57, no. 8, pp. 1023–1031, Aug, Sep 2008.
- [8] The Y. I. Cho, N. S. Chang, C. H. Kim, Y.-H. Park, and Perg. Hong, "New Bit Parallel Multiplier with the Low Space Complexity for All the Irreducible Trinomials Over," The IEEE Trans. VLSI Syst., vol. 20, no. 10, pp. and 1903– 1908, Oct. Nov 2012.
- [9] A. Reyhani-Masoleh, journal "A New Bit-Serial Architecture for the Field Multiplication Using the Polynomial Bases," The CHES 2008, LNCS 5154. Springer, pp. 300–314, Jan. 2009.
- [10] The J. Grossschadl, "A low-power bit-serial bit multiplier for finite fields $GF(2^m)$," in the IEEE International Symposium on the Circuits and Systems (ISCAS) , vol. 5, May, june 2001.
- [11] M. Morales-Sandoval, The C. Feregrino-Uribe, and the P. Kitsos, "Bit-serial and the digit-serial $GF(2^m)$ Montgomery of multipliers using linear feedback shift registers,"

AUTHOR DETAILS

	<p>P.SHAMITHA is pursuing M.tech VLSI system design from Nalanda Institute of Engineering and Technology. She completed her B.tech. Her research of interest includes CMOS DIGITAL, Analog CMOS, VLSI design etc.</p>
	<p>A.SYAM KUMAR is working as assistant professor in Nalanda Institute of Engineering and Technology. He completed his post-graduation in DECS and his area of interest includes Low power VLSI design, CMOS mixed, Digital system design.</p>