International Journal of Advancements in Electronics and Electrical Engineering– IJAEEE
Volume 4 : Issue 2 [ISSN : 2319-7498]
Publication Date : 30 October, 2015

Implementation of GF (2^{16}) Multiplier Using Combinational Gates

Mohini Sawane, Aaditi Bhoite, Shweta Garthe, P.V.Sriniwas Shastry

***Abstract*— This paper proposes the design and implementation of GF (2^{16}) multiplier using composite field arithmetic. We have introduced an irreducible polynomial $X^2+X+\xi$. This irreducible polynomial is required for transforming Galois field of GF (2^{16}) to composite field of GF $(((2^2)^2)^2)^2$. Our estimation of the value of ξ and subsequently the composite field arithmetic hence forth derived achieved high speed GF (2^{16}) multiplier. The design being purely combinational is a clock free design. We achieved critical path delay of 11.5ns between inputs to output data path. We have used combination of ψ and λ as $\{10\}_2$ and $\{1100\}_2$ respectively. Due to this value of ψ, λ, ξ we achieved fastest implementation, at the cost of few extra gates. The design methodology includes implementation and verification on FPGA using Xilinx ISE and finally the physical layout was designed on ASIC using 90nm CMOS standard cell libraries. Our implementation result shows that without pipelining the hardware core can achieve throughput of 5.39 Mbps on FPGA and we achieved throughput of 5.43Gbps on 90nm ASIC.**

***Keywords*— Galois field, composite field arithmetic, isomorphic mapping.**

I. Introduction

Multiplications are elementary mathematical operations extremely important in signal processing applications.To keep pace with the technology, high speed applications require faster methods of multiplication. Multipliers are the key components of many high performance systems such as FIR filters, microprocessors, and digital signal processors etc.Thecomputational performance of a DSP system is limited by its multiplication performance and since, multiplication dominates the execution time of most DSP algorithms, high-speed multiplier is much desired. Currently, multiplication time is still the dominant Factor in determining the instruction cycle time of a DSP chip.hence, optimizing the speed element and area of the multiplier is a major design issue. The three important considerations for VLSI design are power, area and delay.

Mohini Sawane, Aaditi Bhoite, Shweta Garthe, P V Sriniwas Shastry
Department of Electronics and Telecommunication,
Cummins College of Engineering for Women,
Karvenagar, Pune, India

There are number of techniques to perform binary multiplication. In general, the choice is based upon factors such as latency, throughput, area, and design complexity. Galois field multiplier, Array multiplier, Booth Multiplier and Wallace Tree multiplier are some of the standard approaches to have hardware implementation of multiplier which are suitable for VLSI implementation at CMOS level. Galois field multiplier is fix bit multiplier while others are not. Galois field multipliers are high in performance because of their carry free property. Due to decomposition of Galois field to composite field, complexity is less than Array multiplier, Booth Multiplier and Wallace Tree multiplier. In this paper, high speed GF (2^{16}) multiplier is implemented using tower field decomposition, employing lowest resources.

In the work of [1] presents the design and implementation of substitute Byte process element required in AES encryption. They have used the composite field arithmetic for computing multiplicative inverse. The conversion of GF (2^8) to GF (2^4) and subsequently to GF (2) has reduced the complexity.
Isomorphic Mapping and Inverse Isomorphic Mapping Technique is used for mapping of Galois field to composite field and vice versa [2]. For mapping of GF (2^{16}) to GF (2^8) irreducible polynomial is used which contain constant μ. They performed the multiplication with an assumed value for the constant μ [3]. In the literature published till date, design methodologies of a Galois field multiplier and theory based on pipelining has been presented. A design of Galois field multipliers using a composite field includes designing of lower order Galois field multiplier. For implementation of GF (2^2), GF $(2^2)^2$, GF $((2^2)^2)^2$, GF $(((2^2)^2)^2)^2$ we use irreducible polynomial which has constant ψ, λ, ξ respectively.

The main contribution of this paper is to estimate the value of ξ for Implementation of GF (2^{16}) to GF $(((2^2)^2)^2)$ tower field conversion and also implementing on FPGA as well as on 90nm CMOS technology such that the design consumes low power and area and achieves high speed of operation. The value of ξ which is 8 bit constant required in irreducible polynomial $X^2+X+\xi$.

The rest of the paper is organised in the following manner. Section II explains the fundamentals of Galois field , Section III elaborates our implementations of multiplier. The paper is concluded by Section IV that discusses our results and comparison.

II. Galois field

A Galois field is a field with a finite number of elements. Notations of the finite field are GF (p^m), where the letters

GF stands for "Galois Field". The order of number of elements of a Galois field is of the form p^m where 'p' is Prime number called characteristics of field & 'm' is positive integer called dimensions of the field. The Galois Field operations especially have the advantage of achieving high performance because of its carry free property and low resource requirement. The complete multiplication operation can be realized by using XOR gates only.

The multiplicand and multiplier are expressed in GF where in any number is expressed in a polynomial form. Here a polynomial *f(x)* is a mathematical expression in the form $a_n x^n + a_{n-1}x^{n-1} + ... + a_0$. The highest exponent of *x* is the degree of the polynomial. For example, the degree of $x^5 + 3x^3 + 4$ is 5. In a polynomial, a_n, a_{n-1}... a_0 are called coefficients. If in a polynomial, the coefficients a_n, a_{n-1}... a_1 are all 0, or in other words, the polynomial is in the form of a_0, we call this polynomial a constant. We can add, subtract polynomials by combine the terms in the polynomials with the same powers.

Let

$$f(x) = a_n x^n + a_{n-1}x^{n-1} + ... + a_0 \text{ and}$$
$$g(x) = b_m x^m + b_{m-1}x^{m-1} + ... + b_0$$

be two polynomials over a field *F*, then there is a unique polynomial *r(x)* of degree smaller than *m* and another unique polynomial *h(x)*, both over *F*, such that $f(x) = h(x)*g(x)+r(x)$. The polynomial *r(x)* is called the remainder of *f(x)* modulo *g(x)*. For polynomials *a(x)*, *b(x)* and *g(x)* which are over the same field, we say *a(x)* is congruent to *b(x)* modulo *g(x)* written $a(x) \equiv b(x) \bmod g(x)$.

Example: GF (2^2) is generated by $F(x) = x^2+x+1$

Let $A = (11) = x+1$

$B = (10) = x$

Then $C = AB = (x+1)\, x \bmod F(x)$

$= (x^2+x) \bmod F(x)$

$= (x +x+ 1) +1 \bmod F(x)$

$= 1$

$= (01)$

III. GF multiplier Implementation

The GF (2^{16}) elements can be represented in the polynomial form. For example $\{1010010101001101\}_2$ is represented as $q^{15}+q^{13}+q^{10}+q^8+q^6+q^3+q^2+1$

Polynomial *q* is represented as $q_H x+q_L$

Where q_H is higher bits and q_L lower bits. x is constant number

By using irreducible polynomials as shown below GF (2^8) can be decompose to lower order [2].

GF $((2^2)^2)^2$ to GF $(2^2)^2$: $X^2+X+\lambda$ (1)

GF $(2^2)^2$ to GF (2^2) : $X^2+X+\Psi$ (2)

GF (2^2) to GF (2) : X^2+X+1 (3)

$X^2+X+\lambda$, $X^2+X+\Psi$, X^2+X+1 are irreducible polynomials where λ, Ψ are 4 bit and 2 bit respectively.There are number of combination for Ψ, λ are possible. Combination of Ψ and λ are mentioned in Table I.

TABLE I: COMBINATION OF Ψ AND λ.

Values of Ψ	Values of λ
$\{10\}_2, \{11\}_2$	$\{1000\}_2, \{1100\}_2, \{1001\}_2, \{1101\}_2,$ $\{1010\}_2, \{1110\}_2, \{1011\}_2, \{1111\}_2$

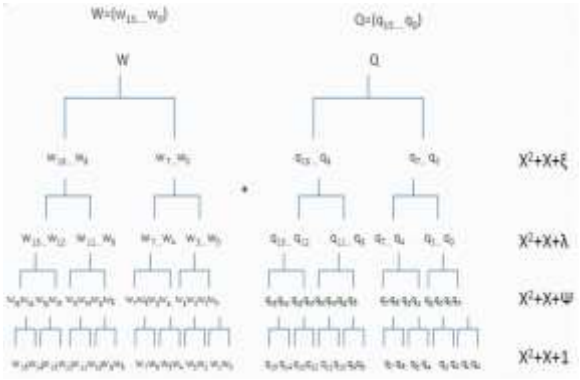

Fig: 1. Representation of decomposition of GF (2^{16}) to GF(2)

Let $K=Q*W$, each a 16-bit finite field and represented as

$K = \{K_{15}, K_{14}, ..., K_0\}$, $Q=\{Q_{15}, Q_{14}, ..., Q_0\}$ And

$W = \{W_{15}, W_{14}, ..., W_0\}$ are elements of GF (2^{16})

Therefore, $K=K_H X+K_L$

Where $K_H = \{K_{15}, K_{14}, ..., K_8\}$ And $K_L=\{K_7,K_6,.....,K_0\}$

Similarly, $Q=Q_H X+Q_L$

Where $Q_H = \{Q_{15}, Q_{14}, ..., Q_8\}$ And $Q_L=\{Q_7,Q_6,.....,Q_0\}$

And $W=W_H X+W_L$

Where $W_H= \{W_{15}, W_{14}, ..., W_8\}$ And $W_L= \{W_7,W_6,.....W_0\}$

$K= K_H X+K_L=(Q_H X+Q_L)*(W_H X+W_L)$

$$K= Q_H*W_H*X^2+Q_H*W_L*X+Q_L*W_H*X+Q_LW_L$$

$$K=Q_H*W_H*X^2+(Q_H*W_L+Q_L*W_H)\,X+Q_L*W_L \quad (4)$$

For decomposition of more complex GF (2^{16}) to lower order GF $((2^2)^2)^2$, GF $(2^2)^2$, GF (2^2) and GF (2^1) irreducible polynomials of (1), (2) and (3) are used. The decomposition of GF $(((2^2)^2)^2)^2$to GF $((2^2)^2)^2$ is done by using irreducible polynomial $X^2+X+\xi$ where ξ is an 8 bit constant.

Substituting the X^2 term with $X^2= X+\xi$ in equation (4) results in (5).

$$K= Q_H*W_H*(X+\xi) + (Q_H*W_L+Q_L*W_H)\,X+Q_L*W_L \quad (5)$$

$K=$ $(Q_H*W_H+Q_H*W_L+Q_L*W_H)$ $X+Q_H*W_H*\xi+Q_L*W_L$ belong to GF (2^8).

The values of ξ may take up many combinations. From Fig.1, the calculation of multiplication in composite field, elements can't apply directly to the GF (2^{16}) elements.It must be mapped into Galois field first. For that purpose isomorphic function δ is used.After performing multiplication,the result will also have to map back from its composite field. For that purpose inverse isomorphic function δ^{-1} is used.Both δ and δ^{-1} can be represented in 16*16 matrix.Let q be the element in GF (2^{16}) then the isomorphic mapping and its inverse can be written as δ*q and δ^{-1}*q.

$$\delta*q = \begin{pmatrix} 1&1&0&1&0&0&0&1&1&0&1&1&0&0&0&0\\ 0&0&0&0&0&0&1&1&0&1&0&1&0&0&1&0\\ 1&0&0&0&1&0&0&1&0&1&0&0&1&0&0&0\\ 0&1&1&1&0&0&0&1&0&0&0&0&0&1&1&1\\ 1&0&0&1&0&1&0&0&1&0&0&0&0&0&1&0\\ 0&0&0&0&0&0&1&0&0&0&0&1&0&1&1&1\\ 0&0&1&0&0&0&1&0&0&0&0&1&1&1&1&0\\ 0&0&1&0&0&0&1&0&0&1&0&0&1&0&1&1\\ 1&0&1&1&0&0&0&0&1&1&0&1&0&0&0&1\\ 0&1&0&1&0&0&1&0&0&0&0&0&0&0&1&1\\ 0&1&0&0&1&0&0&0&1&0&0&0&1&0&0&1\\ 0&0&0&0&0&1&1&1&0&1&1&1&0&0&0&1\\ 1&0&0&0&0&0&1&0&1&0&0&1&0&1&0&0\\ 0&0&0&1&0&1&1&1&0&0&0&0&0&0&1&0\\ 0&0&1&1&1&1&1&0&0&0&1&0&0&0&1&0\\ 0&1&0&0&0&1&0&1&0&0&1&0&0&0&1&0 \end{pmatrix} * \begin{pmatrix} q_{15}\\ q_{14}\\ q_{13}\\ q_{12}\\ q_{11}\\ q_{10}\\ q_9\\ q_8\\ q_7\\ q_6\\ q_5\\ q_4\\ q_3\\ q_2\\ q_1\\ q_0 \end{pmatrix}$$

$$= \begin{pmatrix} q_{15}\oplus q_{14}\oplus q_{12}\oplus q_8\oplus q_7\oplus q_5\oplus q_4\\ q_9\oplus q_8\oplus q_6\oplus q_4\oplus q_1\\ q_{15}\oplus q_{11}\oplus q_8\oplus q_6\oplus q_3\\ q_{14}\oplus q_{13}\oplus q_{12}\oplus q_8\oplus q_2\oplus q_1\oplus q_0\\ q_{15}\oplus q_{12}\oplus q_{10}\oplus q_7\oplus q_1\\ q_9\oplus q_4\oplus q_2\oplus q_1\oplus q_0\\ q_{13}\oplus q_9\oplus q_5\oplus q_4\oplus q_3\oplus q_2\oplus q_1\\ q_{13}\oplus q_9\oplus q_6\oplus q_2\oplus q_0\\ q_{15}\oplus q_{13}\oplus q_{12}\oplus q_7\oplus q_6\oplus q_4\oplus q_0\\ q_{14}\oplus q_{12}\oplus q_9\oplus q_1\oplus q_0\\ q_{14}\oplus q_{11}\oplus q_7\oplus q_3\oplus q_0\\ q_{10}\oplus q_9\oplus q_8\oplus q_6\oplus q_5\oplus q_4\oplus q_0\\ q_{15}\oplus q_9\oplus q_7\oplus q_4\oplus q_2 \end{pmatrix}$$

[6]

$$\delta^{-1}*q = \begin{pmatrix} 1&1&1&0&0&0&1&0&0&0&0&0&0&1&0&1&1\\ 1&1&0&1&1&0&0&0&1&0&0&0&1&0&0&1\\ 1&0&1&0&0&1&0&0&1&1&1&0&1&1&0&1\\ 0&0&0&0&0&0&0&1&1&0&0&0&1&0&0&0\\ 1&0&0&1&1&1&1&0&0&0&0&0&0&0&1&0\\ 0&1&0&0&0&0&0&0&0&1&0&1&0&1&0&1\\ 0&0&1&1&0&1&0&0&1&0&1&0&0&0&0&0\\ 1&1&0&1&0&1&0&1&0&0&1&1&0&1&0&1\\ 0&0&0&0&1&0&1&1&1&1&1&0&0&0&1&0\\ 1&0&0&0&1&0&0&1&1&1&0&1&1&0&0&0\\ 1&1&1&0&1&1&0&1&0&1&0&0&1&0&0&0\\ 1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1\\ 0&0&0&0&0&0&1&0&1&0&0&1&1&1&0&0\\ 0&1&0&1&0&1&0&1&0&0&0&0&0&0&0&0\\ 1&0&1&0&0&0&0&0&0&0&1&1&0&1&0&0\\ 0&1&0&0&1&1&0&1&1&1&0&1&0&1&0&1 \end{pmatrix} * \begin{pmatrix} q_{15}\\ q_{14}\\ q_{13}\\ q_{12}\\ q_{11}\\ q_{10}\\ q_9\\ q_8\\ q_7\\ q_6\\ q_5\\ q_4\\ q_3\\ q_2\\ q_1\\ q_0 \end{pmatrix}$$

$$= \begin{pmatrix} q_{15}\oplus q_{14}\oplus q_{13}\oplus q_9\oplus q_3\oplus q_1\oplus q_0\\ q_{15}\oplus q_{14}\oplus q_{12}\oplus q_{11}\oplus q_7\oplus q_3\oplus q_0\\ q_{15}\oplus q_{13}\oplus q_{10}\oplus q_7\oplus q_6\oplus q_5\oplus q_3\oplus q_2\oplus q_0\\ q_8\oplus q_7\oplus q_3\\ q_{15}\oplus q_{12}\oplus q_{11}\oplus q_{10}\oplus q_1\\ q_{14}\oplus q_6\oplus q_4\oplus q_2\oplus q_0\\ q_{13}\oplus q_{12}\oplus q_{10}\oplus q_7\oplus q_5\\ q_{15}\oplus q_{14}\oplus q_{12}\oplus q_{10}\oplus q_8\oplus q_6\oplus q_3\oplus q_2\oplus q_0\\ q_{11}\oplus q_9\oplus q_8\oplus q_7\oplus q_6\oplus q_5\oplus q_1\\ q_{15}\oplus q_{11}\oplus q_8\oplus q_7\oplus q_6\oplus q_4\oplus q_3\\ q_{13}\oplus q_{14}\oplus q_{13}\oplus q_{11}\oplus q_{10}\oplus q_8\oplus q_7\oplus q_5\oplus q_2\\ q_{15}\oplus q_{11}\oplus q_0\\ q_9\oplus q_7\oplus q_4\oplus q_3\oplus q_2\\ q_{14}\oplus q_{12}\oplus q_{10}\oplus q_8\oplus q_6\\ q_{15}\oplus q_{13}\oplus q_5\oplus q_4\oplus q_2\\ q_{14}\oplus q_{11}\oplus q_{10}\oplus q_8\oplus q_7\oplus q_6\oplus q_4\oplus q_2\oplus q_0 \end{pmatrix}$$

[7]

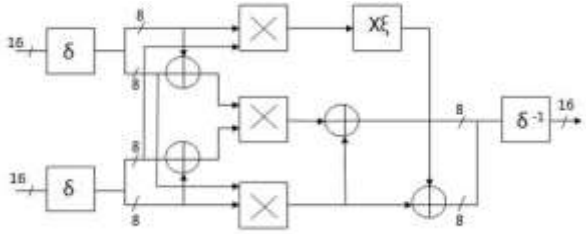

Fig.2. Implementation of GF (2^{16}) multiplier

In order to construct a GF (2^{16}) multiplier, GF (2^8) multiplier implementation is used. For multiplication of 16 bit binary number in Galois field, two 16 bit inputs are given and we get output of 16 bit.

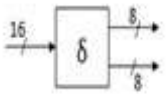

Fig.3. Isomorphic mapping of GF (2^{16}) to GF$((2^8)^2)$

As shown in Fig. 3, the 16 bit binary input given to the delta block transforms it from finite field to composite field [3]. For each input, multiplicand and multiplier, the isomorphic transformation needs to be performed prior to applying to block as shown in Fig.2.

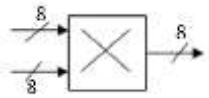

Fig. 4. Hardware block for GF (2^8) multiplier

The block in Fig. 4, represents the 8 bit Galois field multiplier, which has two 8 bit input and 8 bit output. This block can be implemented using combinational gates [2].

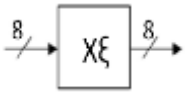

Fig. 5. Hardware block for GF (2^8) multiplier with constant ξ

The block in Fig.5, represent multiplication with constant ξ which is an 8 bit constant used in irreducible polynomial $X^2+X+\xi$ for decomposition of GF $(((2^2)^2)^2)^2$ to GF $(((2)^2)^2)^2$.

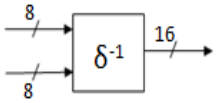

Fig.6. Inverse Isomorphic transformation block

The inverse isomorphic transformation of 16 bit as shown in Fig. 6 maps the polynomials represented in composite field arithmetic to GF(2^{16}) finite field format. It is a 16*16 bit matrix which can be implemented using XOR gates [3].

In the proposed GF (2^{16}) multiplier in Fig.2, the coefficient ξ can take any value out of 128 combinations. The selection of ξ may change the number of gates required while implementing GF (2^{16}) multiplier. The estimation of the constant ξ is a complex and time consuming method and therefore we have followed a methodology to select value of ξ that is explained in the subsequent paragraphs.

TABLE II: COMPOSITE FIELD CONSTRUCTION WITH NORMAL BASIS [3][2]

Galois Field	Normal Basis	Defining Polynomial
$GF(2^{16})\rightarrow GF(((2^2)^2)^2)^2$	$\{\delta,\delta^{256}\}$	$n(X)=X^2+X+\xi$,where $\xi=\beta+\lambda\gamma$
$GF(2^8)\rightarrow GF((2^2)^2)^2$	$\{\gamma,\gamma^{16}\}$	$m(X)=X^2+X+\lambda$, where $\lambda=\psi^2\beta$
$GF(2^4)\rightarrow GF(2^2)^2$	$\{\beta,\beta^4\}$	$l(X)=X^2+X+\psi$
$GF(2^2)\rightarrow GF(2)^2$	$\{\psi,\psi^2\}$	$k(X)=X^2+X+1$

The multiplication with normal bases [3] has been considered for 16-bit multiplication and each internal block of that design was analysed with known inputs and their output products such that value of ξ can be estimated. Subsequently GF(2^2) is constructed by using the irreducible polynomial $k(X)$ over GF(2).Similarly GF$(2^2)^2$ is constructed by using irreducible polynomial $l(X)$,GF$((2^2)^2)^2$ is constructed by using irreducible polynomial m(X) and GF$((2^2)^2)^2)^2$ is constructed by using irreducible polynomial $n(X)$.

The Composite field construction for multiplication with ξ block can be further elaborated only after unfolding the various constituent blocks wherein all are expressed in normal base.

Implementation of $M\psi$ and $M\psi^2$ Multiplier blocks

Let $A=a_0\psi+a_1\psi^2$ and $B=b_0\psi+b_1\psi^2$, *where* $a_0, b_0, a_1, a_0, c_0, c_1 \epsilon GF(2)$.Multiplication by ψ and ψ^2 in GF (2^2) with Normal Bases are shown in (8) and (9).

$$\begin{aligned}\psi A &= \psi(a_0\psi+a_1\psi^2)\\ &= a_0\psi^2+a_1\psi^2\psi\\ &= a_0\psi^2+a_1(\psi+1)\psi\\ &= a_0\psi^2+a_1\psi^2+a_1\psi\\ &= (a_0+a_1)\psi^2+a_1\psi \quad (8)\end{aligned}$$

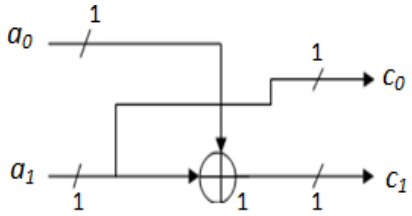

Fig.7.Implementation of ψA for Mψ block

$$\begin{aligned}\psi^2 A &= \psi^2(a_0\psi+a_1\psi^2)\\ &= a_0(\psi+1)\psi+a_1(\psi^2+1)\\ &= a_0\psi^2+a_0\psi+a_1\psi^2+a_1\\ &= a_0\psi^2+a_0\psi+a_1\psi\\ &= a_0\psi^2+(a_0+a_1)\psi \quad (9)\end{aligned}$$

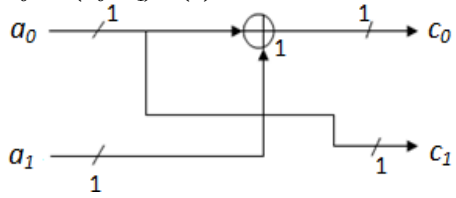

Fig. 8.Implementation of $\psi^2 A$ for Mψ^2block

Implementation of M_{λ^2} block

Let $A=a_0\beta+a_1\beta^4$ and $B=b_0\beta+b_1\beta^4$, where $a_0, b_0, a_1, a_0, c_0, c_1 \epsilon GF(2^2)$.Multiplication of $A \epsilon GF(2^2)^2$ *by* λ^2, β and $\psi\beta$ are computed as follows

$$\begin{aligned}\lambda &= \psi^2\beta\\ \lambda^2 &= (\psi^2\beta)(\psi^2\beta)\\ &= (\psi\beta+\beta)(\psi\beta+\beta)\\ &= \psi^2\beta^2+\psi\beta^2+\psi\beta^2+\beta^2\\ &= \psi\beta^2+\beta^2+\beta^2\\ &= \psi\beta^2 \quad (10)\\ \lambda^2 A &= \psi\beta^2 A\\ &= \psi\beta^2(a_0\beta+a_1\beta^4)\\ &= a_0\psi\beta^3+a_1\psi\beta^6\\ &= a_0\psi(\beta+\psi\beta^4)+a_1\psi^2\beta\\ &= (a_0\psi+a_1\psi^2)\beta+a_0\psi^2\beta^4 \quad (11)\end{aligned}$$

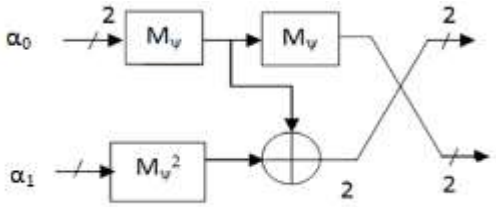

Fig:9 Multiplication of A with λ^2 for $M_{\lambda}{}^2$ block

$$\beta A = \beta (a_0\beta + a_1\beta^4)$$
$$= a_0 [(\psi+1)\beta + \psi\beta^4] + a_1(\psi\beta + \psi\beta^4)$$
$$= [a_0 + (a_0+a_1)\psi]\beta + [(a_0+a_1)\psi]\beta^4 \quad (12)$$

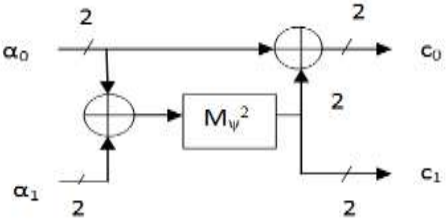

Fig:10 Multiplication of A with β for M_{β} block

$$\psi\beta A = a_0(\beta+\psi^2\beta^4) + a_1(\psi^2\beta+\psi^2\beta^4)$$
$$= (a_0 + a_1\psi^2)\beta + (a_0\psi^2 + a_1\psi^2)\beta^4 \quad (13)$$

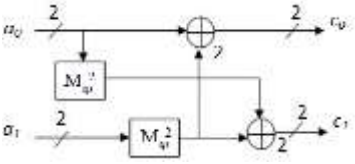

Fig. 11. Multiplication of A with ψ β for $M_{\psi\beta}$ block

Multiplication of A ϵ GF $((2^2)^2)^2$ with ξ, where a_0, $b_0, a_1, a_0, c_0, c_1 \epsilon GF(2^2)$

$$A = (a_0\gamma + a_1\gamma^{16})$$
$$\xi A = (\beta + \lambda\gamma)(a_0\gamma + a_1\gamma^{16})$$
$$= a_0\beta\gamma + a_1\beta\gamma^{16} + a_0\lambda\gamma^2 + a_1\lambda\gamma^{17}$$
$$= [a_0(\psi\beta) + (a_0 + a_1)\lambda^2]\gamma + [a_1\beta + (a_0 + a_1)\lambda^2]\gamma^{16} \quad (14)$$

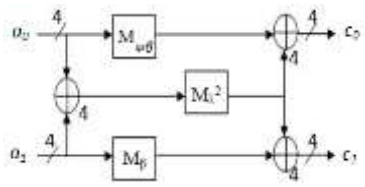

Fig.12. Multiplication of A with ξ

To determine value of ξ

Substitute $A = \{a_1a_0\} = \{0000\ 0001\}$, where $A \epsilon GF((2^2)^2)^2$. Such that ξ A=ξ in the equation (14) which implemented in Fig. 12. The resultant values of nets can be shown in Fig.13.

$a_1 = \{0000\}$ and $a_0 = \{0001\}$

Multiplication of ξ A, where $A = \{0000\ 0001\}_2$

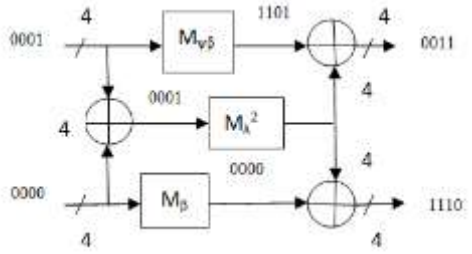

Fig.13. Estimation of $\xi = \{1110\ 0011\}_2$

Multiplication of q with ξ

Let q be any 8 bit binary number

$$k_0 = q(6)+q(5)+q(4)+q(1)+q(0)$$
$$k_1 = q(5)+q(4)+q(0)$$
$$k_2 = q(5)+q(4)+q(3)+q(2)$$
$$k_3 = q(4)+q(2)$$
$$k_4 = q(4)+q(1)$$
$$k_5 = q(7)+q(5)+q(3)+q(1)+q(0)$$
$$k_6 = q(7)+q(5)+q(5)+q(4)+q(2)$$
$$k_7 = q(7)+q(6)+q(4)+q(3) \quad (15)$$
$$k = \{k_7, k_6, k_5, k_4, k_3, k_2, k_1, k_0\}$$

IV. Our Result and Comparison

The implementation of design resulted combinational logic circuit which contain the combination of AND and XOR gates. Our ξ value is $\{11100011\}_2$.The design was implemented on Vertex 4 FPGA using Xilinx ISE tool. In [2] the authors suggested the value of ψ and λ as $\{10\}_2$ and $\{1100\}_2$ respectively. For our implementation, we took the value of ψ, λ and ξ as $\{10\}_2$, $\{1100\}_2$ and $\{11100011\}_2$. We achieved the critical path delay 11.5ns.

Our implementation result shows that without pipelined, we achieved throughput of 5.39Mbps on FPGA and a throughput of 5.43Gbps on 90nm ASIC respectively. While synthesizing and layout design, we have considered TSMC 90nm standard cell libraries. Cadence RTL compiler and Encounter are the tools used for synthesis and physical layout design. The final layout of our implementation is shown in Fig. 14.

The performance of our FPGA implemented design is better than the other two designs as mentioned in Table III. Our implementation on FPGA as well ASIC consumes very low area and power without pipelining the architecture.

TABLE III: COMPARISONS WITH OTHER MULTIPLIER.

Hardware Platform	Implementation	Area		Speed (MHz)	Power (mW)
		# of	# of		
FPGA	WG-29[7]	6,449	-	30	380
	WG-29[8]	4,044	-	34	187
	MOWG-29[8]	5,512	-	35	342
	Ours	272	156	86.3	105
		Area(gates)		Speed	Power
ASIC	WG-29[7]	33,180		144	7.28
	WG-29[8]	19,892		169	4.45
	MOWG-29[8]	26,261		151	5.89
	Ours	2,999		8690	0.27

.

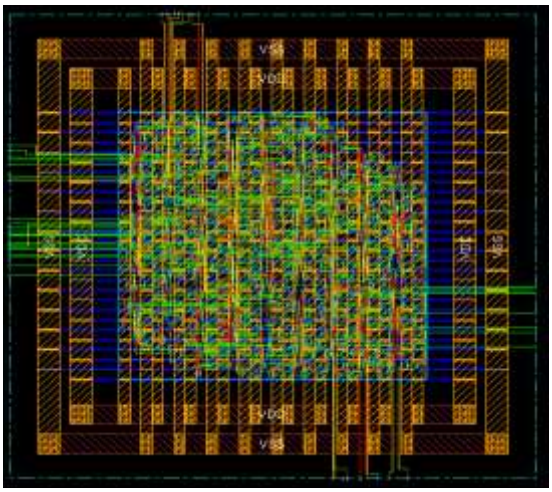

Fig 14: Physical layout of our implementation

:

References

[1] P.V.Sriniwas Shastry, Mukul S. Sutaone, "Multiplicative Inverse in GF (2^8) Using Combinational Logic Circuit", IETE National Journal of Innovation and Research, Vol 1, Issue 1, June 2013.

[2] Edwin NC Mui, "Practical Implementation of Rijndeal S-box Using Combinational Logic" Custom R&D Engineer Texco Enterprise Ptd, Ltd.

[3] Xinxin Fan, Nusa Zidaric ,Mark Aagaard, and Guang Gong , "Efficient Hardware Implementation of the stream cipher WG-16 with Composite Field Arithmetic" Proceedings of 3rd International Workshop on Trustworthy Embedded Devices, 2013, pp 21-34.

[4] Berk Sunar, Erkay Savas, Certin K. Koc,"Constructing Composite Field Representation For Efficient Conversion", IEEE Transactions on Computers, Vol 52, No. 11, pp. 1391-1398.

[5] Jiafeng Xie, Pramod Kumar Meher,Jianjun He, "Low Complexity Multiplier For GF(2^{16}) Based on All One Polynomials" ,IEEE Transactions on VLSI System,vol 21,No 1,1 January 2013.

[6] B.Gashkov, Sergey Igor, "Complexity of computation in finite fields", Journal of Mathematical Science, Vol. 191, No. 5, June 2013, pp.661-685.

[7] Y.Nawaz, G.Gong, "WG A family of stream ciphers with design randomness properties",Information science, vol.178, no.7, 2008, pp.1903-1916.

[8] H.El-Razouk,A.Reyhani-Masoleh, G.Gong, "New Implementations of the WG streamcipher", IEEE Transactions on Very Large Scale Integrated (VLSI) Systems, Vol. 22, No.9, 2013, pp. 1865-1878.

Mohini Siddharth Sawane has completed her Bachelors in Electronics and Telecommunication Engineering in 2015 from Cummins College of Engineering for Women, affiliated to Savitribai Phule University of Pune, India. Her area of interest is VLSI Design

Aaditi Deepak Bhoite, in the year 2015, has post graduated in Electronics and Telecommunication Engineering (Signal Processing) from Cummins College of Engineering, affiliated to Savitribai Phule University of Pune, India. Her area of interest is VLSI Design.

Shweta Satish Garthe has completed her Bachelors in Electronics and Telecommunication Engineering in 2015 from Cummins College of Engineering for Women, affiliated to Savitribai Phule University of Pune, India.

P V Sriniwas Shastry has completed his Masters in Electronics and Telecommunication Engineering in 1993 from Pune University, Pune. He is presently working as Associate Professor at Cummins College of Engineering for Women Pune. He has 25 years of teaching experience and his areas of interest include VLSI Design, VLSI in Digital Signal Processing and Cryptography. He is Member of IEEE and Fellow IETE.