I.J. Image, Graphics and Signal Processing, **2016, 10, 35-41**
Published Online October 2016 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijigsp.2016.10.05

Design of Fast Fourier Transform Architecture for GF(2^4) with Reduced Operational Complexity

Tejaswini P. Deshmukh and Pooja R. Pawar
[1]Electronics department, YCCE, Nagpur, India
Email: {tejaswini.deshmukh, poojapawar501}@gmail.com

P. R. Deshmukh[2] and P. K. Dakhole[1]
[2]Electronics department, Cipna College of Engineering, Amravati, India
Email: prdeshmukh@ieee.org, pravin_dakhole@yahoo.com

Abstract**—**In this paper, the architecture for Fast Fourier Transform over Galois Field (2^4) is described. The method used is cyclotomic decomposition. The Cyclotomic Fast Fourier Transforms (CFFTs) are preferred due to low multiplicative complexity. The approach used is the decomposition of the arbitrary polynomial into a sum of linearized polynomials. Also, Common Subexpression Elimination (CSE) algorithm is used to reduce the additive complexity of the architecture. By using CSE algorithm, the design with reduced operational complexity has been described.

Index Terms**—**Cyclotomic, Fourier Transform, Galois Field.

I. Introduction

Fourier analysis is useful in converting a signal from its original domain into frequency domain and vice versa. The Fast Fourier Transform (FFT) algorithm designed for the complex field is not well-suited for the finite field. The FFT in the complex field has applications throughout the subject of signal processing [3]. Whereas, the FFT over the finite field have been widely used in cryptography and have applications in error correcting codes. The method for Fast Fourier Transform over the finite field (i.e. Galois Field) [2] [3] along with the architecture has been suggested in the paper. Galois Field is a field that contains a number of finite elements.

The suggested method consists of decomposing an original polynomial into a sum of linearized polynomials and evaluating them at a set of basis points [2]. The architecture designed in this paper is for GF(2^4). The Cyclotomic Fast Fourier Transform (CFFT) is useful in RS i.e. Reed-Solomon decoders to reduce the complexity of the decoder [4]. Because Reed Solomon code is cyclic in nature [15]. The CFFT proposed in [2] has low multiplicative complexity but they have high additive complexities. The FFT suggested in this paper can be used to perform the RS decoding which involves two time-consuming steps (Syndrome computation and Chien search). Chien search is a fast algorithm used in determining roots of polynomials defined over a finite field. The RS codes are capable of correcting random errors and multiple burst errors. This architecture can also be used to implement the Gao algorithm [14] which includes operations based on Fourier transform.

The design of architecture follows several steps which have been explained in a simplified manner in this paper. The architecture is designed in 4 stages. The architecture so designed is modified by applying Common Subexpression Elimination (CSE) Algorithm. CSE algorithm reduces the additive complexity of the architecture. The language used for design is Verilog and has been implemented in the Xilinx ISE Design Suite.

The paper proceeds as follows. Section II covers basic notions and definitions of the Fourier transform and the method to determine cyclotomic cosets, along with the basic theory of Galois Field. Section III focuses on linearized polynomials and generation of the matrix. Section IV describes hardware architecture. The Common Sub-expression elimination Algorithm has been explained in section V. Section VI illustrates the architecture after applying CSE. And the paper concludes with the comparison between two architectures in section VII.

II. Definitions

The Fourier transform of a polynomial is the collection of elements.

The Fourier transform can be generated using [2] [10]:

$$f(x) = \sum_{i=0}^{n-1} f_i x^i \qquad (1)$$

is of degree f(x) = n-1 and n | (2^m-1).

2.2. The elements can be estimated through:

$$f(\alpha^j) = \sum_{i=0}^{n-1} f_i \alpha^{ij} \qquad (2)$$

Here, j Є |0, n-1|.

2.3. The cyclotomic cosets C_k over modulo $n=2^m - 1$ for GF(2^m) is calculated as:

$$C_0 = \{0\} ,$$
$$C_{k1} = \{ k_1 , k_1 2, k_1 2^2, ..., k_1 2^{m_1 -1}\},$$
$$..........,$$
$$C_{kl} = \{ k_l , k_l 2, k_l 2^2, ..., k_l 2^{m_l -1}\},$$

where

$$k_s \equiv k_s 2^{ms} \text{modn}. \quad (3)$$

2.4. A linearized polynomial over GF(2^m) is a polynomial represented as:

$$L(x) = \sum l_i x^{2i} \quad (4)$$

It can be shown that L(x) satisfies L(a+b) = L(a) + L(b).

One useful representation of elements in Galois Field is m-tuple representation. Let $\alpha_0 + \alpha_1\alpha + \alpha_2\alpha + ... + \alpha_{m-1} \alpha^{m-1}$ be the polynomial representation of a field element β. Then, β can be represented by an ordered sequence of m components called an m-tuple, as follows [7]:

$(\alpha_0, \alpha_1, \alpha_2, . . . , \alpha_{m-1})$

Where, the m components are simply the m coefficients of the polynomial representation of β.

For GF(2^4) the 4-tuple representation generated by $p(X) = X^4+X+1$ is:

Table 1. 4-Tuple representation

Power representation	Polynomial representation	4-Tuple representation
0	0	(0 0 0 0)
1	1	(0 0 0 1)
α	α	(0 0 1 0)
α^2	α^2	(0 1 0 0)
α^3	α^3	(1 0 0 0)
α^4	$\alpha+1$	(0 0 1 1)
α^5	$\alpha^2+ \alpha$	(0 1 1 0)
α^6	$\alpha^3+ \alpha^2$	(1 1 0 0)
α^7	$\alpha^3+\alpha+1$	(1 0 1 1)
α^8	α^2+1	(0 1 0 1)
α^9	$\alpha^3+\alpha$	(1 0 1 0)
α^{10}	$\alpha^2+\alpha+1$	(0 1 1 1)
α^{11}	$\alpha^3+\alpha^2+\alpha$	(1 1 1 0)
α^{12}	$\alpha^3+\alpha^2+\alpha+1$	(1 1 1 1)
α^{13}	$\alpha^3+\alpha^2+1$	(1 1 0 1)
α^{14}	α^3+1	(1 0 0 1)

The elements of GF(2^m) forms all the roots of $X^{2^\wedge m}$ + X. Let Ø(X) be the polynomial of the smallest degree over GF(2^m). This polynomial Ø(X) is called the minimal polynomial of β. Ø(X) must be irreducible. Minimal polynomials of the elements in GF(2^4) generated by $p(X) = X^4+X+1$ are:

Table 2. Minimal polynomials

Conjugate roots	Minimal polynomials
0	X
1	X+1
$\alpha, \alpha^2, \alpha^4, \alpha^8$	X^4+X+1
$\alpha^3, \alpha^6, \alpha^9, \alpha^{12}$	$X^4+X^3+X^2+X+1$
α^5, α^{10}	X^2+X+1
$\alpha^7, \alpha^{11}, \alpha^{13}, \alpha^{14}$	X^4+X^3+1

III. Cyclotomic Fast Fourier Transform

Based on the formula mentioned in (3) in section II, the cyclotomic cosets so formed for GF(2^4) after substituting m=4 i.e. n=15 are as follows:

$$C_0 = \{0\}$$
$$C_1 = C_2 = C_4 = C_8 = \{1,2,4,8\}$$
$$C_3 = C_6 = C_9 = C_{12} = \{3,6,9,12\}$$
$$C_5 = C_{10} = \{5,10\}$$
$$C_7 = C_{11} = C_{13} = C_{14} = \{7,11,13,14\}$$

An irreducible polynomial p(X) of degree m is said to be primitive if the smallest positive integer n for which p(X) divides X^n+1 is $n = 2^m-1$. $p(X) = X^4+X+1$ divides $X^{15}+1$ but does not divide any X^n+1 for $1 \leq n \leq 15$. Hence, X^4+X+1 is a primitive polynomial for GF(2^4). Let α be the root of this polynomial.

f(α^i) can be developed using:

$$f(\alpha^i) = \sum_{i=0}^{l} (L_i(\alpha^{jki})) \quad (5)$$

These coefficients α_{ijs} are used to form the matrix A. For example, in GF(2^4) l=4, k_0=0, k_1=1, k_2=3, k_3=5, k_4=7. So,

$$f(\alpha^1) = L_0(\alpha^0)+L_1(\alpha)+L_2(\alpha^3)+L_3(\alpha^5)+L_4(\alpha^7)$$
$$= L_0(1) + L_1(\beta) + L_1(\beta^8) + L_2(\beta) + L_3(\gamma) + L_4(\beta) + L_4(\beta^2) + L_4(\beta^4)$$

Here, the basis for C_1, C_3, C_7 is $(\beta, \beta^2, \beta^4, \beta^8)$, where $\beta = \alpha^3$ and α is an element of GF(2^4) as $\alpha = \beta + \beta^8$. For C_5 the basis is (γ, γ^2) where $\gamma = \alpha^5$.

The coefficients of $f(\alpha^1)$ are deduced as

$$\alpha_{ijs} = [1\ 1\ 0\ 0\ 1\ 1\ 0\ 0\ 0\ 1\ 0\ 1\ 1\ 1\ 0]$$

The rest of the equations are developed as follows:

$$f(\alpha^0) = L_0(\alpha^0) + L_1(\alpha^0) + L_2(\alpha^0) + L_3(\alpha^0) + L_4(\alpha^0)$$
$$f(\alpha^1) = L_0(\alpha^0) + L_1(\alpha) + L_2(\alpha^3) + L_3(\alpha^5) + L_4(\alpha^7)$$
$$f(\alpha^2) = L_0(\alpha^0) + L_1(\alpha^2) + L_2(\alpha^6) + L_3(\alpha^{10}) + L_4(\alpha^{14})$$
$$....$$
$$....$$
$$....$$
$$f(\alpha^{13}) = L_0(\alpha^0) + L_1(\alpha^{13}) + L_2(\alpha^{39}) + L_3(\alpha^{65}) + L_4(\alpha^{91})$$
$$f(\alpha^{14}) = L_0(\alpha^0) + L_1(\alpha^{14}) + L_2(\alpha^{42}) + L_3(\alpha^{70}) + L_4(\alpha^{98})$$

Based on the above equations, the matrix A so formed is –

$$A = \begin{bmatrix} 1&1&1&1&1&1&1&1&1&1&1&1&1&1&1 \\ 1&1&0&0&1&1&0&0&0&1&0&1&1&1&0 \\ 1&1&1&0&0&0&1&0&0&0&1&0&1&1&1 \\ 1&1&0&0&0&0&0&0&1&1&1&0&1&0&0 \\ 1&0&1&1&0&0&0&1&0&1&0&1&0&1&1 \\ 1&0&1&0&1&1&1&1&1&0&1&0&1&0&1 \\ 1&0&1&0&0&1&0&0&0&1&1&0&0&1&0 \\ 1&1&1&1&0&0&1&0&0&1&0&0&1&1&0 \\ 1&0&0&1&1&0&0&0&1&0&1&1&1&0&1 \\ 1&0&0&0&1&0&0&1&0&1&1&1&0&0&0 \\ 1&1&0&1&0&1&1&1&1&1&0&1&0&1&0 \\ 1&1&1&0&1&1&0&0&0&0&1&1&1&0&0 \\ 1&0&0&1&0&0&1&0&0&1&1&0&0&0&1 \\ 1&1&0&1&1&0&0&0&1&1&0&1&0&0&1 \\ 1&0&1&1&1&0&0&1&0&0&1&0&0&1&1 \end{bmatrix}$$

This matrix A is multiplied by the vector. The vector is denoted as Lf, where f is the original vector and L is the block diagonal matrix formed by elements β. The matrix L is represented as –

$$L = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & L1 & 0 & 0 & 0 \\ 0 & 0 & L2 & 0 & 0 \\ 0 & 0 & 0 & L3 & 0 \\ 0 & 0 & 0 & 0 & L4 \end{bmatrix}$$

where

$$L1 = L2 = L4 = \begin{bmatrix} \beta & \beta^2 & \beta^4 & \beta^8 \\ \beta^2 & \beta^4 & \beta^8 & \beta \\ \beta^4 & \beta^8 & \beta & \beta^2 \\ \beta^8 & \beta & \beta^2 & \beta^4 \end{bmatrix}$$

and

$$L3 = \begin{bmatrix} \gamma & \gamma^2 \\ \gamma^2 & \gamma \end{bmatrix}$$

Further, applying cyclic convolution between the normal basis and f_i i.e. Lf [2][5][6] is rewritten as [3]
$B = Lf = Q\,((R\,\beta_i^T).(S\,f_i)) =$

$$\begin{bmatrix} 101100001 \\ 101010010 \\ 110100100 \\ 110011000 \end{bmatrix} \begin{bmatrix} 1111 \\ 1100 \\ 0011 \\ 0101 \\ 1010 \\ 1000 \\ 0100 \\ 0010 \\ 0001 \end{bmatrix} \begin{bmatrix} \beta \\ \beta^8 \\ \beta^4 \\ \beta \end{bmatrix} \begin{bmatrix} 1000 \\ 1010 \\ 1010 \\ 1001 \\ 1100 \\ 1111 \\ 1111 \\ 1111 \\ 1111 \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ f_4 \\ f_8 \end{bmatrix}$$

This four point cyclic convolution [6] is obtained for cosets C_1, C_3, $C_{7.}$ Whereas, the same interpretation for coset C_5 gives a two-point cyclic convolution.

$$\begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \gamma \\ \gamma^2 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} f_5 \\ f_{10} \end{bmatrix}$$

The complete architecture can be computed as [2]:

$$F = AQ(C\,.\,(Pf)) \qquad (6)$$

Where, Q is the binary block diagonal matrix, C is the combined vector of constants, and P is the binary block diagonal matrix of combined pre-additions.

IV. Hardware Architecture

The architecture design of FFT starts with the designing of GF multiplier. The GF multiplier is used for multiplication of polynomials [7].

Consider 2 polynomials ($a_0\alpha^3 + a_1\alpha^2 + a_2\alpha + a_3$) and ($b_0\alpha^3 + b_1\alpha^2 + b_2\alpha + b_3$).

The multiplication of these polynomials results in

$$c_0\alpha^3+c_1\alpha^2+c_2\alpha+c_3 = (a_0b_0+a_3b_0+a_0b_3+a_2b_1+a_1b_2)\,\alpha^3 + (a_0b_0 +a_1b_0 +a_0b_1 +a_3b_1 +a_1b_3 +a_2b_2)\,\alpha^2 +(a_1b_1 +a_0b_1 +a_2b_0 +a_0b_2 +a_1b_1 +a_3b_2+a_2b_3)\,\alpha + (a_3b_3 +a_2b_0 +a_0b_2 +a_1b_1)$$

The GF multiplier design can be represented as –

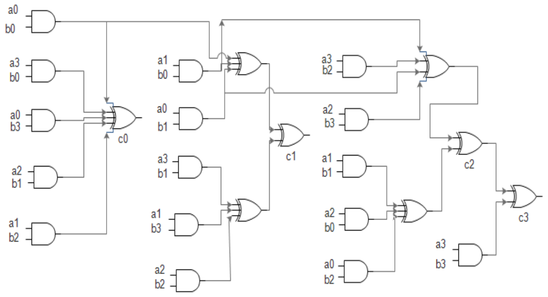

Fig.1. GF Multiplier

The above design requires 16 AND gates and 8 XOR gates.

The complete FFT architecture [3] can be represented as follows:

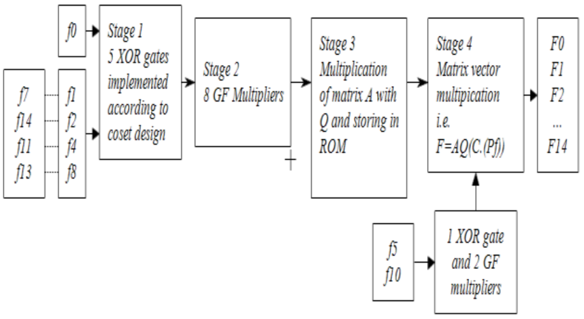

Fig.2. Architecture design

The cosets form the stage 1 and 2 of the architecture. The design for C_1, C_3, C_7 is same, whereas the design for C_5 is different. Cosets C_1, C_3, C_7 requires 5 XOR gates and 8 GF multipliers, whereas C_5 requires only 1 XOR gate and 2 GF multipliers. The design for cosets is:

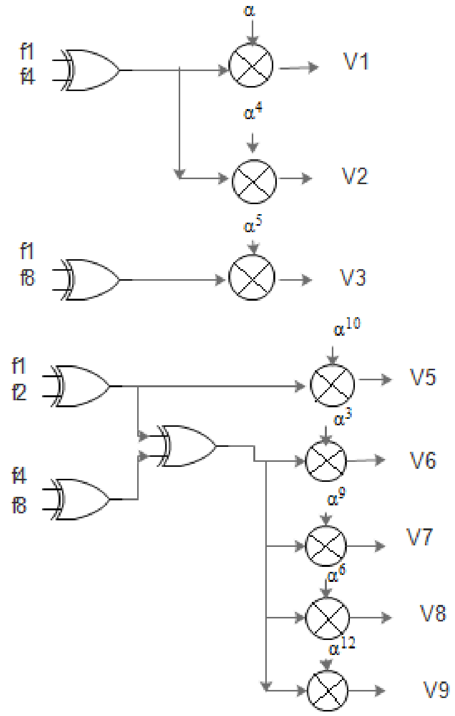

Fig.3. Coset design for C_1, C_3, C_7.

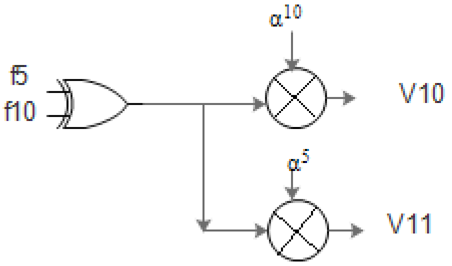

Fig.4. Coset design for C_5.

In stage 3, the matrix multiplication is performed. The matrices A and Q are multiplied and thereby stored in ROM. The resultant matrix size is 15 x 31. Finally, in stage 4 the matrix and vector are multiplied. Stage 1 and 2 results in a vector, whereas stage 3 results in a matrix. The output of stage 4 is the FFT of the input.

V. Common Sub-Expression Elimination Algorithm

In many Digital Signal Processing applications, multiple constant multiplication is widely used. In VLSI design for high-level synthesis, proper optimization of multiple constant multiplication is effective in improving parameters like area and power consumption. To optimize multiple constant multiplications, the Common Sub-Expression Elimination (CSE) algorithm is used in this paper [5]. The approach used in CSE is initially to identify the identical terms i.e. the common sub-expressions present in the equations and then to replace them with a single variable. Thus, by computing the terms only once, results are significantly being reduced in the hardware architecture in VLSI design.

In Galois field, matrix multiplication is performed. Here, the addition is performed via XOR-ing but there are several methods to perform multiplication. In this paper, the multiplication is a linear transform of the form C = AB, where C and B are *m*- and *n*- dimensional column vectors, respectively, and A is an m x n constant binary matrix. Here, B represents input variable and C represents output variables. According to this paper, the B column vector is *(C . (Pf))* and the matrix A is the resultant matrix of *AQ* (Refer (6)). So, the CSE algorithm is applied to this matrix – vector multiplication.

Some general steps are involved in carrying out the CSE algorithm. These steps are as follows [5]:

1. To identify common patterns present in the transformation.
2. Select an appropriate pattern for elimination.
3. Compute the pattern only once.
4. Eliminate the occurrences of the computed pattern
5. Repeat steps 1 to 4 to cover every pattern.

So, by applying CSE algorithm to matrix-vector multiplication there is a significant reduction in a number of XOR gates.

The method suggested in [8] reduces the additive complexities of Cyclotomic Fast Fourier Transform using a weighted sum of the numbers of multiplications and additions. [12] focuses on both area and delay optimization in hardware implementations over GF(2^m).

VI. Reduced FFT Architecture

To reduce the additive complexity of the FFT architecture, in this paper, the CSE algorithm mentioned in section V has been used. The basic stages of architecture mentioned in section IV remain the same, with the only difference in being applying the CSE algorithm. The CSE algorithm is applied to stage 4. After applying the CSE algorithm to the matrix, the matrix size increases from 15 x 31 to 47 x 63. Due to this, the number of LUTs eventually increases in the final architecture but the additive complexity i.e. the number of XOR gates are reduced significantly.

We have written a synthesizable Verilog code for the different stages of the architecture. First 3 stages of the architecture remains the same, whereas, the architecture design changes at stage 4. Based on the appropriate changes the two architectures i.e. without CSE and with CSE can be compared in terms of LUTs, the number of XOR gates required and delay.

VII. RESULTS

The complexity of the proposed architecture has been evaluated considering the synthesis report generated in Xilinx ISE Design Suite. The synthesis report has been generated for two FPGA devices, namely- Spartan 6 and Virtex 5. The results of the GF multiplier used in the architecture are- for Spartan 6 the LUTs required are 9 whereas for Virtex 5 the LUTs required are 8. Number of IOBs remains the same for both FPGA kits.

Table 3. Synthesis Report – GF Multiplier

XOR gates	LUTs required	IOBs	Max Combinational path delay
Spartan 6			
4	9	12	6.781ns
Virtex 5			
6	8	12	5.115ns

The maximum combinational path delay is the maximum delay that would occur for the complete architecture. For cosets C_1, C_3, C_7 since the design is the same the implementation results so generated are same. But, for coset C_5 the results are different. For cosets C_1, C_3, C_7 te results are:

Table 4. Synthesis Report – Cosets C_1, C_3, C_7

XOR gates	LUTs required	IOBs	Max Combinational path delay
Spartan 6			
37	88	80	8.379ns
Virtex 5			
52	80	80	5.969ns

For cosets C_1, C_3, C_7 the XOR gates required for Spartan 6 are 37 and for Virtex 5 are 52.

Table 5. Synthesis Report – Coset C_5

XOR gates	LUTs required	IOBs	Max Combinational path delay
Spartan 6			
9	22	24	7.791ns
Virtex 5			
13	20	24	5.715ns

The stage 3 of the architecture is formed by the multiplication of matrix A with Q. The simulation results are:

Table 6. Synthesis Report – Matrix A*Q

LUTs required	IOBs	Max Combinational path delay
Spartan 6		
31	35	6.1ns
Virtex 5		
31	35	3.979ns

In the stage 3 of architecture, since only matrix multiplication is involved, no XOR gates are required in this stage.

Finally, the results for the architecture without applying CSE are:

Table 7. Synthesis Report – Architecture without CSE

XOR gates	LUTs required	IOBs	Max Combinational path delay
Spartan 6			
121	360	100	10.893ns
Virtex 5			
170	340	100	8.363ns

The paper mainly focuses on reducing the additive complexity of the FFT architecture. Therefore, the number of XOR gates required before modifying the architecture are 121 and 170 for Spartan 6 and Virtex 5 respectively.

The results after modifying the architecture with CSE are:

Table 8. Synthesis Report – Architecture with CSE

XOR gates	LUTs required	IOBs	Max Combinational path delay
Spartan 6			
54	176	132	6.93ns
Virtex 5			
73	173	132	9.849ns

After applying CSE, the number of XOR gates are reduced to 54 and 73 for Spartan 6 and Virtex 5 respectively.

VIII. CONCLUSION

It is clearly evident from the above-mentioned results that the additive complexity of the architecture after applying CSE reduces by a considerable amount. The area of the architecture is also reduced. The number of XOR gates required before applying CSE is 121 and 170 for Spartan 6 and Virtex 5 respectively. Whereas, after applying CSE XOR gates reduces to 54 and 73. Thus, the additive complexity of the FFT architecture is reduced.

Graphically, the comparison between two architectures can be plotted as:

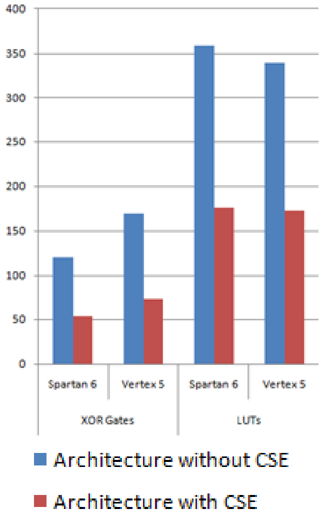

Fig.5. Complexity comparison of architectures.

IX. Future Scope

This paper primarily focuses on FFT of 15 elements represented as GF(2^4). But, based on the technique mentioned in the paper, the FFT for higher powers of 2 can also be generated. To generate DFT over larger fields like GF(2^{11}) or GF(2^{12}), the algorithm mentioned in [11] can be used. Different multipliers like Karatsuba multiplier, Montgomery multiplier etc. can be used and compared with above-mentioned results.

Acknowledgment

The authors would like to thank Prof. P. Trifonov and Prof. S. Fedorenko for the details of the algorithm used in this paper. The authors would also like to thank Prof. Ali AL Ghouwayel, Prof. Yves Louët, Prof. Amor Nafkha and Prof. Jacques Palicot for the details of CFFT hardware architecture.

References

[1] Trifonov, Peter. "On the additive complexity of the cyclotomic FFT algorithm." In *Information Theory Workshop (ITW), 2012 IEEE*, pp. 537-541. IEEE, 2012.

[2] Trifonov, P. V., and S. V. Fedorenko. "A method for fast computation of the Fourier transform over a finite field." *Problems of Information Transmission* 39, no. 3 (2003): 231-238.

[3] Ghouwayel Ali Al, Yves Louet, Amor Nafkha, and Jacques Palicot. "On the FPGA implementation of the Fourier transform over finite fields GF (2m)." In *Communications and Information Technologies, 2007. ISCIT'07. International Symposium on*, pp. 146-151. IEEE, 2007.

[4] Wu, Xuebin, Zhiyuan Yan, and Jun Lin. "Reduced-Complexity Decoders of Long Reed-Solomon Codes Based on Composite Cyclotomic Fourier Transforms." *Signal Processing, IEEE Transactions on* 60.7(2012):3920-3925.

[5] Wu, Ning, et al. "Improving Common Subexpression Elimination Algorithm with A New Gate-Level Delay Computing Method." *Proceedings of the World Congress on Engineering and Computer Science*. Vol. 2.2013

[6] R. Blahut, “Theory and Practice of Error Control Codes,” Reading Massachusetts: Addison-Wesley, 1983.

[7] Shu Lin, Daniel J. Costello Jr., “Error Control Coding,” Pearson Education, 2005

[8] Chen, Ning, and Zhiyuan Yan. "Cyclotomic FFTs with reduced additive complexities based on a novel common subexpression elimination algorithm." *Signal Processing, IEEE Transactions on* 57.3 (2009): 1010-1020.

[9] Chang, Ching-Hsien, Chin-Liang Wang, and Yu-Tai Chang. "Efficient VLSI architectures for fast computation of the discrete Fourier transform and its inverse." *Signal Processing, IEEE Transactions on* 48.11 (2000): 3206-3216.

[10] Fedorenko, Sergei, and Peter Trifonov. "On computing the fast Fourier transform over finite fields." *Proc. 8th Int. Workshop on Algebraic and Combinatorial Coding Theory, Tsarskoe Selo, Russia*. 2002.

[11] Wu, Xuebin, et al. "Composite cyclotomic Fourier transforms with reduced complexities." *Signal Processing, IEEE Transactions on* 59.5 (2011): 2136-2145.

[12] Zhang, Xiaoqiang, et al. "A low-delay common subexpression elimination algorithm for constant matrix multiplications over GF (2 m)." *Industrial Electronics and Applications (ICIEA), 2015 IEEE 10th Conference on*. IEEE, 2015.

[13] Ahmed, Elias, and Jonathan Rose. "The effect of LUT and cluster size on deep-submicron FPGA performance and density." *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on* 12.3 (2004): 288-298.

[14] Gao, Shuhong. "A new algorithm for decoding Reed-Solomon codes."*Communications, Information and Network Security*. Springer US, 2003. 55-68.

[15] R. Blahut,“Algebraic Codes for Data Transmission,” *Cambridge University Press*, 2003.

Authors’ Profiles

Tejaswini P. Deshmukh, is an Assistant Professor at Yeshwantrao Chavan College of Engineering, Nagpur University. She is currently working towards the Ph.D. degree in Electronics Department of YCCE Research Centre, Nagpur University. Her research interest includes Digital Signal Processing, Digital Communication and Digital Image Processing. Her teaching interest includes Signals and Systems, Digital Signal Processing and Embedded Systems.

Pooja R. Pawar, is a post graduate student at Yeshwantrao Chavan College of Engineering, Nagpur University. She is currently pursing post graduation under the guidance of Tejaswini Deshmukh.

Pravin Dakhole, is a Professor at Electronics department at Yeshwantrao Chavan College of Engineering Research Centre. His research interest includes VLSI. His teaching interest includes Advanced Digital System Design, and Verification and Testing.

Prashant Deshmukh, is a Professor at Sipna College of Engineering. He has teaching experience of 7 - 8 years. His research interest includes Signal Processing and VLSI.

How to cite this paper: Tejaswini P. Deshmukh, Pooja R. Pawar, P. R. Deshmukh, P. K. Dakhole,"Design of Fast Fourier Transform Architecture for GF(2^4) with Reduced Operational Complexity", International Journal of Image, Graphics and Signal Processing(IJIGSP), Vol.8, No.10, pp.35-41, 2016.DOI: 10.5815/ijigsp.2016.10.05