IOSR Journal of VLSI and Signal Processing (IOSR-JVSP)
Volume 6, Issue 3, Ver. III (May. - Jun. 2016), PP 08-11
e-ISSN: 2319 – 4200, p-ISSN No. : 2319 – 4197
www.iosrjournals.org

Design and Implementation of the Cyclotomic Fast Fourier Transform Architecture over GF(2^3)

Tejaswini Deshmukh[1], Prashant Deshmukh[2], Pravin Dakhole[3]
[1,3]*(Electronics dept., YCCE, Nagpur, India)*
[2]*(Electronics dept., Cipna College of Engg., Amravati, India)*

Abstract: *The hardware design and the FPGA implementation of the Fast Fourier Transform over Galois Field i.e. GF(2^3) is described. By considering the algorithm presented in [2], we have designed the architecture in four stages. The method used for designing is cyclotomic decomposition. The cyclotomic fast Fourier transform is preferred due to low multiplicative complexity. The architecture for GF(2^3) have been proposed in the paper and implemented on FPGA kit Virtex-5.*

Keywords: *Cyclotomic, Fourier Transform, FPGA, Galois Field.*

I. Introduction

The objective of this paper is to design Fast Fourier Transform (FFT) architecture for finite fields. Finite field is also called Galois Field. Galois field is a field which contains finite number of elements. Here we consider GF(2^3) consists of 7 field elements. The FFT designs for complex field and finite field are different. The FFT in complex field finds application throughout the subject of Signal Processing. The finite field FFT is used in cryptography and also have applications in error correcting codes. The Reed Solomon code i.e. RS code is cyclic in nature [6]. And therefore, the Cyclotomic Fast Fourier Transform (CFFT) [1] is useful in RS decoder to reduce the complexity of the decoder.

The method used to design the architecture of FFT in this paper is decomposition of polynomials into a sum of linearized polynomials [2]. The polynomials are evaluated at a set of basis points. 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 [7] which includes operations based on Fourier transform.

The paper proceeds as follows. Section 2 covers basic notions and definitions of the Fourier transform and the method to determine cyclotomic cosets, along with the basic theory of Galois Field. The Cyclotomic Fast Fourier Transform is covered in section 3. Hardware architecture of GF(2^3) have been explained in section 4. Section 5 includes FPGA implementation results of the architectures. Conclusion of the paper is described in section 6. Finally, the paper ends with acknowledgement and references.

II. Definitions

The Fourier transform of a polynomial is the collection of elements.
The Fourier transform can be generated using [2] [3]:

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

is of degree f(x) = n-1 and n | (2^m-1).
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|.
The cyclotomic cosets C_k over modulo n=2^m – 1 for GF(2^m) is calculated as:

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

Where $k_s \equiv k_s 2^{ms} \bmod n$. (3)

III. Cyclotomic Fast Fourier Transform (CFFT)

The cyclotomic cosets are calculated using the equation (3). The cyclotomic cosets for GF(2^3) are:

$C_0 = \{0\}$
$C_1 = C_2 = C_4 = \{1,2,4\}$
$C_3 = C_6 = C_5 = \{3,6,5\}$

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 $n = 2^m-1$. For GF(2^3), the primitive polynomial is X^3+X+1.
The elements of GF (8) can be expressed as the combination of the basis (γ, γ2, γ4) as following:

$$\begin{pmatrix} a^0 \\ a^1 \\ a^2 \\ a^3 \\ a^4 \\ a^5 \\ a^6 \end{pmatrix} = \begin{pmatrix} \gamma+\gamma^2+\gamma^4 \\ \gamma^2+\gamma^4 \\ \gamma+\gamma^4 \\ \gamma \\ \gamma+\gamma^2 \\ \gamma^4 \\ \gamma^2 \end{pmatrix} \tag{4}$$

The decomposition of the polynomial f(x) according to the following equation

$$f(\alpha^j) = \sum_{i=0}^{l} L_i(\alpha^{jk_i}) \tag{5}$$

And the substitution of x by αi gives the frequency components Fj , for i, j = 0, ..., 14. Let us consider the development of some components.

$$f(a^0) = L_0(a^0) + L_1(a^0) + L_2(a^0) = L_0(1) + L_1(\gamma) + L_1(\gamma^2) + L_1(\gamma^4)$$
$$+ L_2(\gamma) + L_2(\gamma^2) + L_2(\gamma^4),$$

$$f(a^1) = L_0(a^0) + L_1(a) + L_2(a^3) = L_0(1) + L_1(\gamma^2) + L_1(\gamma^4) + L_2(\gamma),$$

$$f(a^2) = L_0(a^0) + L_1(a^2) + L_2(a^6) = L_0(1) + L_1(\gamma) + L_1(\gamma^4) + L_2(\gamma^2),$$

$$f(a^3) = L_0(a^0) + L_1(a^3) + L_2(a^2) = L_0(1) + L_1(\gamma) + L_2(\gamma) + L_2(\gamma^4),$$

$$f(a^4) = L_0(a^0) + L_1(a^4) + L_2(a^5) = L_0(1) + L_1(\gamma) + L_1(\gamma^2) + L_2(\gamma^4),$$

$$f(a^5) = L_0(a^0) + L_1(a^5) + L_2(a^1) = L_0(1) + L_1(\gamma^4) + L_2(\gamma^2) + L_2(\gamma^4),$$

$$f(a^6) = L_0(a^0) + L_1(a^6) + L_2(a^4) = L_0(1) + L_1(\gamma^2) + L_2(\gamma) + L_2(\gamma^2)$$

After substituting the values of α from Equation (5) the above system of equations can be written in matrix form as

$$F = \begin{pmatrix} F_0 \\ F_1 \\ F_2 \\ F_3 \\ F_4 \\ F_5 \\ F_6 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 1 & 0 \end{pmatrix} \begin{pmatrix} L_0(1) \\ L_0(\gamma) \\ L_0(\gamma^2) \\ L_0(\gamma^4) \\ L_0(\gamma) \\ L_0(\gamma^2) \\ L_0(\gamma^4) \end{pmatrix}$$

Each Li constitutes a (mi × mi)-matrix. By developing the Lis, above matrix is equivalent to

$$F = A\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \gamma^1 & \gamma^2 & \gamma^4 & 0 & 0 & 0 \\ 0 & \gamma^2 & \gamma^4 & \gamma^1 & 0 & 0 & 0 \\ 0 & \gamma^4 & \gamma^1 & \gamma^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \gamma^1 & \gamma^2 & \gamma^4 \\ 0 & 0 & 0 & 0 & \gamma^2 & \gamma^4 & \gamma^1 \\ 0 & 0 & 0 & 0 & \gamma^4 & \gamma^1 & \gamma^2 \end{bmatrix}\begin{bmatrix} f_0 \\ f_1 \\ f_2 \\ f_4 \\ f_3 \\ f_6 \\ f_5 \end{bmatrix}$$

Which can be written as the following form :

F = ALf (6)

The multiplication of matrix L by matrix f is equivalent to four cyclic convolutions of Li by the corresponding cyclotomic coset of fi. The four-point cyclic convolutions can be represented as,

$$\begin{pmatrix} S_1 \\ S_2 \\ S_3 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \end{pmatrix}\left(\left[\begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}\begin{pmatrix} \gamma \\ \gamma^2 \\ \gamma^4 \end{pmatrix}\right]\left[\begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}\begin{pmatrix} f_1 \\ f_2 \\ f_4 \end{pmatrix}\right]\right)$$

He once by using the convolution results the equation (6) can be re written as

F = AQ (C(Pf)) (7)

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

Above CFFT equation can be transformed into architecture design as follows:

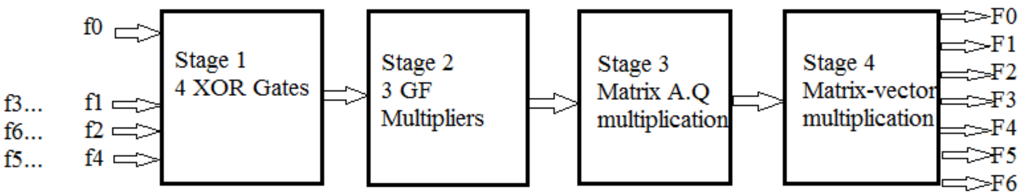

Fig.1. Architecture for GF(2^3)

For GF(2^3), the coset design is:

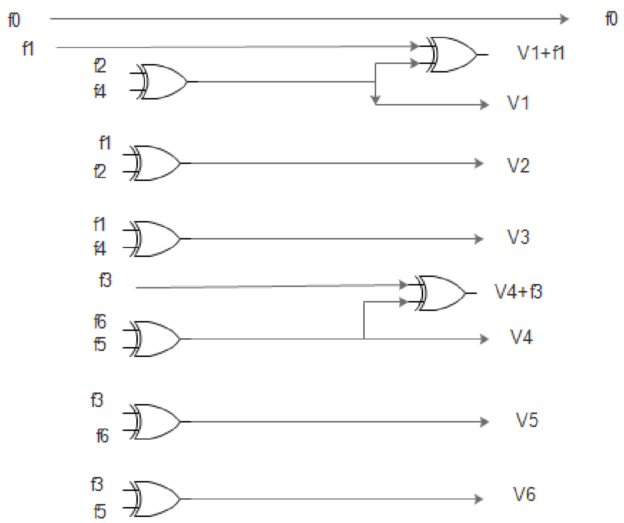

Fig.2. Coset design for GF(2^3)

V. FPGA Implementation

In this subsection we consider the implementation on FPGA (Field Programmable Gate Array). In this paper, we consider implementation on Xilinx's Virtex-5. The performance of these architectures is determined by evaluating the parameters Slice Registers, Slice LUTs, Memory, bonded IOBs, average Fanout of non-clock nets.

We have written Verilog codes for different stages of this architecture. The LUTs required by the designs denotes the area of the architecture. For GF(2^3), 205 Slice LUTs are required amongst the 28,800 available LUTs (1% utilization). The performance of this algorithm is also evaluated in terms of operational complexity. For GF(2^3) CFFT algorithm 8 pre stage adders, 32 post stage adders and 8 multipliers are required. This method uses advantages of cyclotomic decomposition. The architecture processes the input elements set by set instead of sequential processing (i.e symbol by symbol) this leads to reduction in computation time when compared with the sequentially operating algorithm. Table 1 summarizes the implementation results of the architecture of FFT for GF(2^3). Table 2 presents operational complexity of algorithm for GF(2^3).

Table 1 Implementation results for GF(2^3)

Parameters	Used	Available	Utilization
Number of Slice Registers	269	28,800	1%
Number of Slice LUTs	205	28,800	1%
Number used as Memory	61	7,680	1%
Number of bonded IOBs	22	480	4%
Total Memory used (KB)	36	2,160	1%
Average Fanout of Non-Clock Nets	2.83		

Table 2 operational complexity for GF(2^3)

Parametrs	Operational complexity
Pre stage adders	8
Post stage adders	32
Multiplier	6
Total Area	46

VI. Conclusion

We have presented the hardware architecture for GF(2^3) and also the FPGA implementation results for the same. The proposed FFT architecture design is based on the cyclotomic decomposition of polynomials. This architecture in comparison with other architecture presents advantages in terms of complexity. The architecture processes the input elements set by set instead this leads to reduction in computation time and operational complexity.

Acknowledgement

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]. R. Blahut, "Theory and Practice of Error Control Codes," Reading Massachusetts: Addison-Wesley, 1983.

[5]. Shu Lin, Daniel J. Costello Jr., "Error Control Coding," Pearson Education, 2005

[6]. R. Blahut, "Algebraic Codes for Data Transmission," *Cambridge University Press*, 2003.

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