VOL. 10, NO. 10, JUNE 2015 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

AREA MINIMIZED FFT ARCHITECTURE USING SPLIT RADIX ALGORITHM FOR LENGTH L=RADIX-3 AND RADIX-2BX3C

Karthick S., Archana T. and Rekha P.
Department of Electronics and Communication Engineering, Saveetha Engineering College, Thandalam, Chennai, India
E-Mail: karthick@saveetha.ac.in

ABSTRACT

Fast Fourier Transform (FFT) is a most proficient algorithm for computing the Discrete Fourier Transform (DFT). FFT is specially used in analysis of autocorrelation, disambiguation, recognition of patterns, statistics and data analysis. The new proposed algorithm for computing a length $L=2^Bx3^C$ FFT. It is used to minimize the twist of hardware depict and arithmetic operations. Additionally, the propound fragment design can accomplish by the mixer of radix-3 and radix-2^Bx3^C FFT algorithm. It is entirely changeable of Split Radix Fast Fourier Transform (SRFFT) algorithm. As a consequence, the new proffer design of length L is used to belittle the area of system on chips and mathematical calculation. More compromise can achieve by combining the two architectures implemented in the same FFT. It naturally furnishes a broad selection of accessible length of FFT's.

Keywords: digital signal processing (DSP), discrete fourier transform (DFT), fast fourier transform (FFT), radix 6, split radix fast fourier trasform (SRFFT).

INTRODUCTION

Discrete Fourier Transform (DFT) is vast spread utilized in numerous fields of Engineering and Science [1]-[6]. In particular, we present important computational algorithms, called Fast Fourier Transform (FFT). FFT is a method for computing the DFT with reduced number of complex multiplications. FFT algorithms are foundation on the essential principle of length L in to consecutively smaller discrete fourier transforms. There are basically two classes of FFT algorithms. They are Decimation-in-time (DIT) and Decimation-in-Frequency (DIF). In Decimation-in-time, the input is bit reversed while the output is in natural order. The sequences for which we need the DFT is successively divided in to smaller sequences and the DFTs of these sub sequences are combined in certain pattern to obtain the requires DFT of the entire sequence. In the Decimation-in-time the input is normal order while the output is bit reversed.

The fast fourier transform algorithm exploit the two basic properties: Symmetry and Periodicity. In order to meet the aloft execution and real time needs of up-to-date applications, hardware designers have always tried to fulfil efficient architecture for the computation of FFT. The most broadly used algorithms are called length N=2m, that can be expressed as radix-2, readix-4, radix-8 [7]-[10]. Later, the designers have found the new design algorithms for length N=3m, that can be expanded by radix-3, radix-6, radix-9 [11]-[13]. Later, The Scientist introduces the new Split Radix Fast Fourier Transform (SRFFT) algorithm [14]-[17]. The engineers have carried out and resulted in the quick implement on this group of algorithms for computing the length L=Mr FFT have arised in the presentation of the concept for length L=3, L=6 and L=9 [18]. Due to scanty efficiency, the algorithms for length Mr. The corresponding length 3r, 6r, 9r algorithms are applied in many existing applications. The main advantage of new proposed design for length L=2Sx3T is used to reduce the area of chip compared to previous existing algorithms. It is much more efficiency than previous algorithms.

The proposed design of length L=2Sx3T is broadly applied in present applications such as digital signal processing, Image processing, Orthogonal Frequency Division Multiplexing (OFDM) and ultra wideband [19]-[20]. The new hardware design includes two cases. In first case, design and calculate the FFT architecture of radix-3. In second case, design and calculate the FFT architecture of radix-6. In this case, combine the 2 point as well as 3 point FFT architecture. Finally, desingn and calculate the FFT architecture using mixed radix algorithm. (3-points and 6-points FFT can be performed with SRFFT). Apply some different conditions on the length L=2Sx3T. In first condition, the algorithm decomposition of FFT size L=12 in to one of the length is L/3 another four length is L/6 sub FFTs. In second condition, the algorithm decomposition of a FFT size L=24 in to one of the length is L/3 another eight length is L/6. Similarly, apply the value of length L. But, the length L is must divisible by 3 and 6. These sub FFTs input sequences are used to reduce the hardware complexity.

The remaining of this paper arranged as the following sections: In Section II basic explanation of radix FFT algorithm. In Section III introduction of new proposed design algorithm. In Section IV comparision and analysis of present and existing algorithm. In Section V experimental results of proposed design. In Section VI is conclusion and the future work.

BAISIC EXPLANANTION OF RADIX FFT ALGORITHM

Fast Fourier Transform algorithm (FFT) is a numerically efficient algorithm used to compute the DFT.

An efficient implementation of the DFT is possible using the mixed radix FFT calculation. Some different type's radix applied in proposed and future enhancement work. That can be explained by the sub sections A, B and C.

2-Point and 3-point FFT algorithm

The DFTs of the decimated sequences are given below:

$$Z(p) = \sum_{l=0}^{L-1} x(l)\, wLKl\,,\ 0 \le k \le L-1 \quad (1)$$

$$WL = e^{-j2\pi/L}$$

where x(l) is the input sequences, WL is the phase factor or twiddle factor.

In Figure-1 illustrate the flow graph of 2-point FFT. Normally, Decimation-in-time or Decimation-in-frequency algorithm used in this graph. And also phase factor is multiplied in below arrow or in both (above and below arrows). But in this flowgraph twiddle factor is not multiply to each other. This process is used in the main proposed design. So the complex multiplications terms can be eliminated. Normal addition and subtraction process is involved in this algorithm. In Figure-2 represents the butterfly diagram of 3-point FFT. 3-points FFT needs totally 4 multiplications and 12 additions. But certain algorithms assume that a 3-points FFT is calculated with 2 multiplications and 12 additions are required. Because one evaluated with bit shift. The basix radix-3 equation can be writtenas,

$$Z(p) = \sum_{l=0}^{L/3-1} x(3l)\, wL3Kl + \sum_{l=0}^{L/3-1} x(3l+1)\, wL(3l+1)K + \sum_{l=0}^{L/3-2} x(3l+2)\, wL(3l+2)K(2)$$

these equation (2) can be reduced to

$$Z(p) = \sum_{l=0}^{L/3-1} x(3l)\, wL3Kl$$

using the new 3-point FFT flowgraph. In this butterfly process is applied in the new proposed hardware design. The complex multiplications as well as complex additions terms can be reduced.

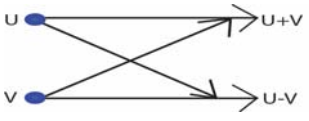


Figure-1. Flow graph of 2-point FFT.

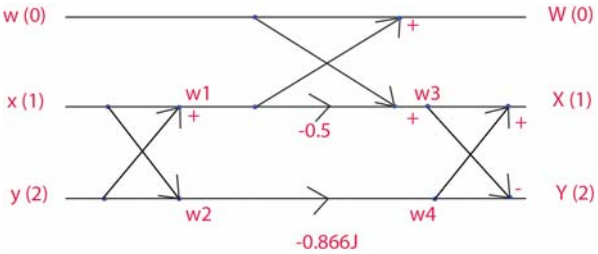


Figure-2. Flow graph of 3-point FFT.

4-Point SRFFT and 5-point FFT algorithm

In Figure-3 illustrate the flow graph of 4-point SRFFT algorithm. In this algorithm exploits this idea by using both different types of radix decomposition in the same FFT algorithm. The basic equation used in the SRFFT algorithm is,

$$X(4K+1) = \sum_{l=0}^{L/4-1} \{[x(l) - x(l+L/2)] - j[x(l+L/4) - x(l+3L/4)]\}\, wLlwL/4lK \quad (3)$$

$$X(4K+3) = \sum_{l=0}^{L/4-1} \{[x(l) - x(l+L/2)] + j[x(l+L/4) - x(l+3L/4)]\}\, wL3lwL/4lK \quad (4)$$

In this concept is mainly used in the new proposed hardware design. If we used in this algorithm means more computational intension can be reduced compared to the previous algorithms. In Figure-4 represents the butterfly diagram of 5-point FFT algorithm. The basic radix-5 equation can be written as follows:

$$Z(p) = \sum_{l=0}^{L/5-1} x(5l)\, wL5Kl + wL5Kl \sum_{l=0}^{L/5-1} x(5l+1)\, wL5Kl + wL2K \sum_{l=0}^{L/5-1} x(5l+2)\, wL5Kl + wL3K \sum_{l=0}^{L/5-1} x(5l+3)\, wL5Kl + wL4K \sum_{l=0}^{L/5-1} x(5l+4)\, wL5Kl \quad (5)$$

In equation (5) is recognized as an L/5 point DFT. Although the index k ranges over the L values. Where k=0, 1.....L-1 each of the sums can be computed only for k=0, 1.....L/5-1. Since, they are periodic with period L/5. The transform of Z(P) can be broken in to five parts as given below:

$$y(K) = A(K) + wLK\, B(K) + wL2K\, C(K) + wL3K\, D(K) + wL4K\, E(K) \quad (6)$$

y(K+L/5)=A(K)+ wL1/5wLK B(K)+ wL2/5wL2K C(K)+ wL3/5wL3KD (K)+ wL4/5wL4KE(K) (7)

this equation can be used in future enhancement work.

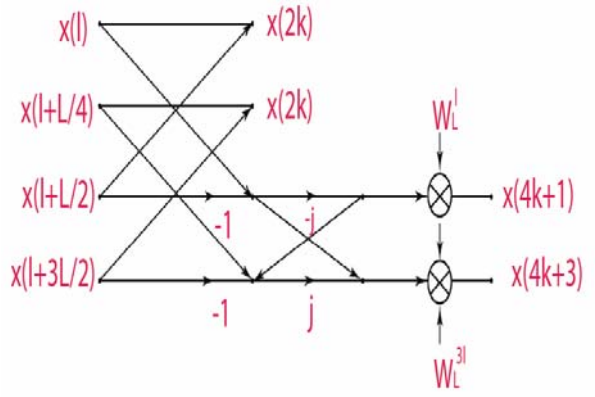


Figure-3. Flow graph of 4-point SRFFT.

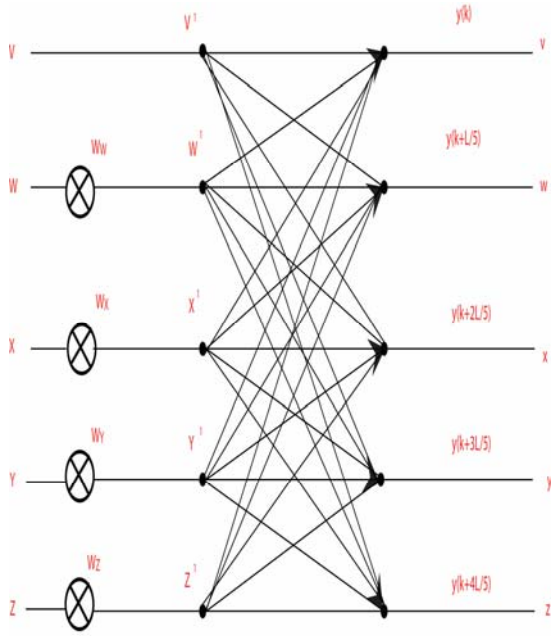


Figure-4. Flow graph of 5-point FFT.

6-Point FFT algorithm

In Figure-5 illustrate the flow graph of 6-point FFT algorithm. It is the combination of 2-point FFT and 3-point FFT flowgraphs. More complexcity can be reduced by this architecture. In this architecture first calculate the algorithm of 2-point FFT. Then calculate the algorithm of 3-point FFT. Finally, connected in these two types of flow graph in the same FFT architecture. This is called a radix-6 FFT algorithms.

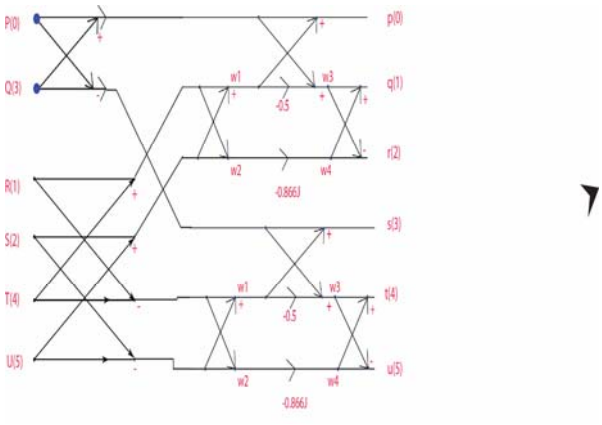


Figure-5. Flow graph of 6-point FFT.

PROPOSED ALGORITHM

In Figure-6 and Figure-7 represents the block diagram of 12-point and 24-point FFT architecture. This architecture is a mixer of 2-point, 3-point and 4-point SRFFT algorithm. These all points are combined and implemented in the single FFT architecture. That is corresponding to architectures shown in Figure-8 and Figure-9. Consider the 12-point FFT architecture, in equation (1) the length L of the input sequences are divisible by 3 and 6. The length $L=2^S x 3^T$is a best new proposed design algorithm. The length L values 12, 24, 36....etc. This type of architecture is used to reduce the memory size.

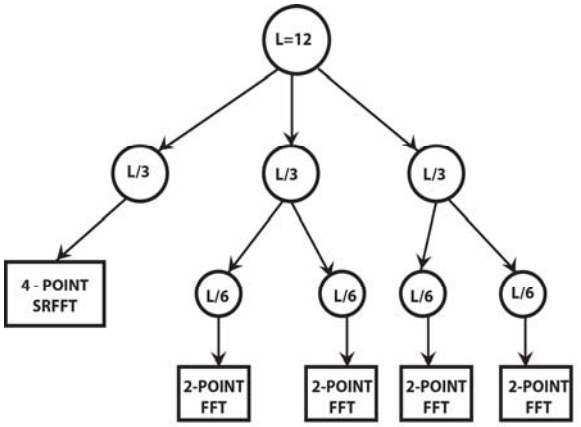


Figure-6. Block diagram of 12-point FFT Architecture.

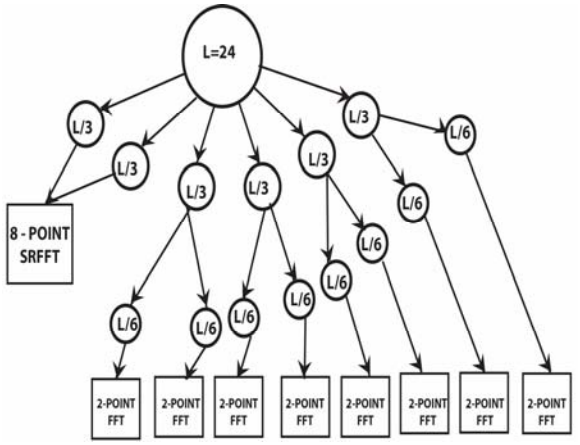


Figure-7. Block diagram of 24-point FFT Architecture.

In Figure-6 the block diagram of length L=12 point FFT. In this condition, the length L can be divided in to three length L/3 sub DFTs. One of the lengths L/3 directly calculated as 4-point Split radix algorithm and another two L/3 lengths are again sub divided in to four L/6 lengths. These four L/6 lengths are calculated as 2-point fast fourier transform algorithm. That is corresponding to 12-point FFT architecture shown in Figure-8. In Figure-7, the block diagram of length L=24 point FFT. In this condition, the length L can be divided in to six length L/3 sub DFTs. One of the two lengths L/3 is calculated as 8-point Split radix algorithm and another four L/3 lengths are again sub divided in to eight L/6 lengths. These eight L/6 lengths are calculated as 2-point fast fourier transform algorithm. That corresponding new proposed 24-point FFT architecture is shown in Figure-9. These types of architectures majorly applied in Orthogonal Frequency Division Multiple Access (OFDMA) and Single Carrier- Frequency Division Multiple Access (SC-FDMA). The length L can be denoted as $L=2^S x 3^T$, where T-1<=S. The decomposition of the DFT length L is expressed by the equation (1) can be modified as follows:

$$Z(p) = \sum_{l=0}^{L/3-1} x(3l)\, w_{L/3}^{Kl} + w^{k}{}_{2}{}^{S}\, w^{k}{}_{3}{}^{T} \sum_{l=0}^{L/6-1} x(6l) +$$

$$_{2}{}^{S} +_{3}{}^{T} w_{L/6}{}^{Kl} + w^{k}{}_{3}{}^{T} \sum_{l=0}^{L/6-1} x(6l) + {}_{2}{}^{S} w_{L/6}{}^{Kl} + w^{-k}{}_{3}{}^{T} + \sum_{l=0}^{L/6-1} x(6l) - {}_{2}{}^{S} w_{L/6}{}^{Kl} + w^{-k}{}_{2}{}^{S}\, w^{-k}{}_{3}{}^{T} \sum_{l=0}^{L/6-1} x(6l) - {}_{2}{}^{S} - {}_{3}{}^{T} w_{L/6}{}^{Kl} \quad (8)$$

In equation (8) can be used in 12-point and 24-point FFT architecture.

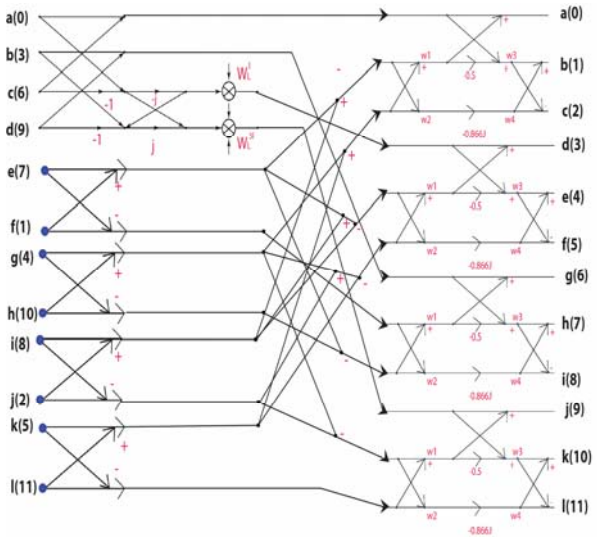


Figure-8. 12-point FFT Architecture.

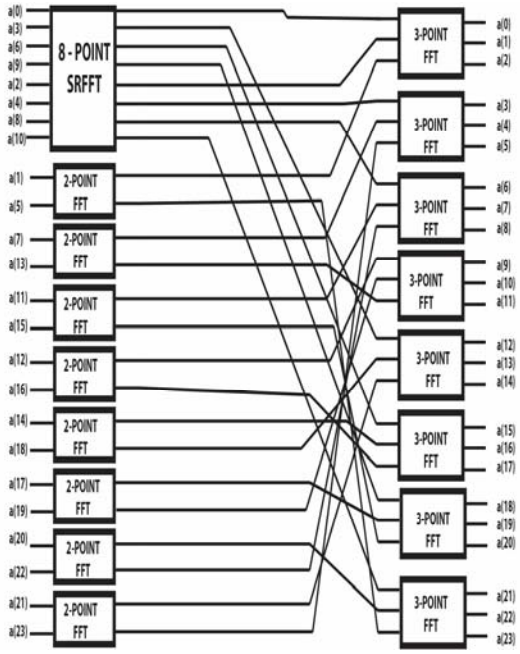


Figure-9. 24-point FFT Architecture.

COMPARISION AND ANALYSIS

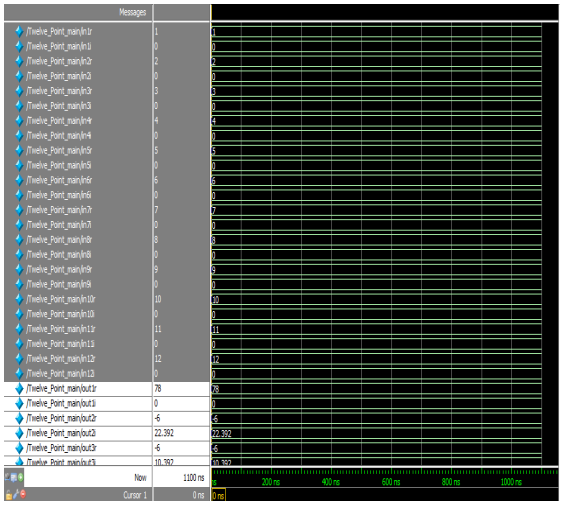
In this section we compare the proposed and existing algorithms. That is shown in below Table-1.

Table-1. Comparision of arithmetic operations(additions and multiplications).

N points	Radix FFT algorithm		
	Radix 6	**Split radix**	**Proposed design algorithm**
36	Multiplications=156 Additions=482	Multiplications=136 Additions=492	Multiplications=128 Additions=464
72	Multiplications=396 Additions= 1178	Multiplications=300 Additions=1168	Multiplications=276 Additions=1092
108	Multiplications=780 Additions=2018	Multiplications=668 Additions=2082	Multiplications=656 Additions=1952
216	Multiplications=1644 Additions=4538	Multiplications=1388 Additions=4672	Multiplications=1340 Additions=4364
432	Multiplications=3804 Additions=10370	Multiplications=2936 Additions=10430	Multiplications=2912 Additions=9768

RESULTS

The simulation result of twelve point FFT and twenty four point FFT architecture is shown in Figure-10 and Figure-11. This design is developed by using Modelsim 6.4 and Xilinx ISE simulator. The number of slice registers used 9 out of 19200, no of bonded IOBs used 61 out of 400, number of block of random access memory used 1 out of 32 and the area utlization is only 3%.



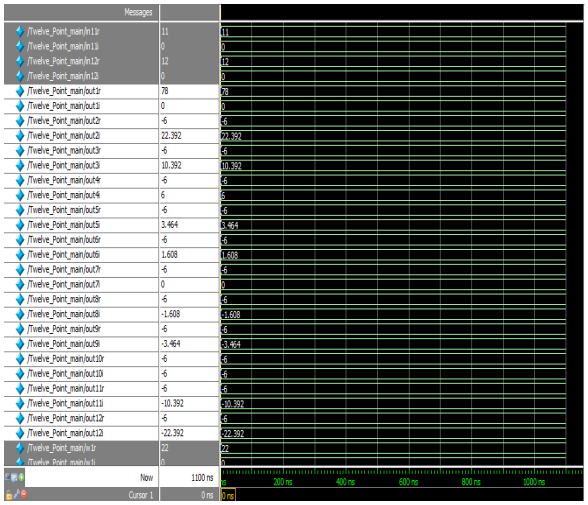


Figure-10. 12-point FFT Architecture.

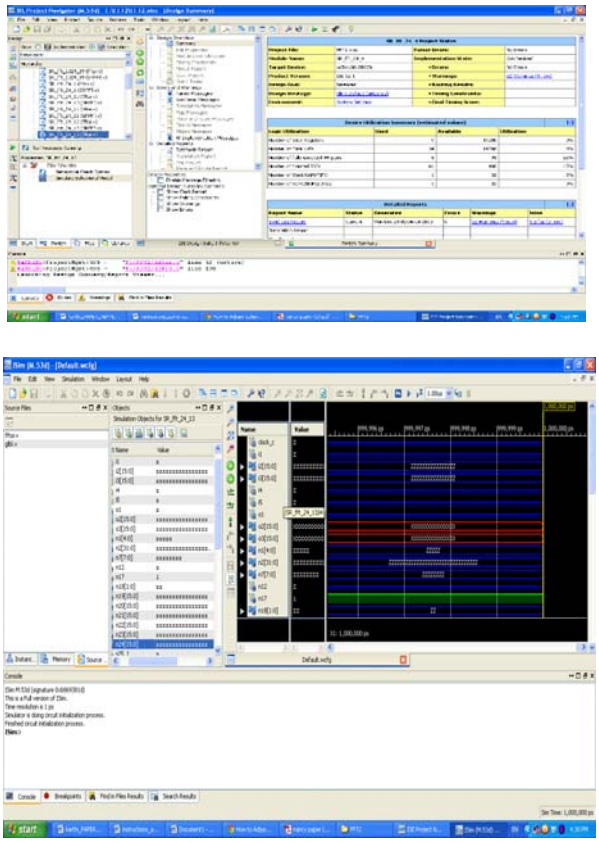


Figure-11. 24-point FFT Architecture.

CONCLUION AND FUTURE WORK

The FFT architecture of area can be reduced by using the split radix algorithm for length L=radix-3 and radix-$2^B x3^C$. The computation complexity can be reduced by using the combination of 2-point FFT, 3-point FFT and 6-point FFT algorithm. More flexibility can achieve by split radix fast fourier transform (SRFFT) algorithm. In future, the proposed design algorithm for Length L= radix-5 and radix-$2^E x5^F$ FFT, we will do some works in this way.

REFERENCES

[1] S. Karthick and S. Dhandapani. 2014. Design and Implementation of FFT using Split Radix Algorithm for Length L=$2^s x3^t$. International conference on Electrical, Communication and Computing, ICECC, pp. 1148-1153.

[2] Wei-Hsin Chang and Truong Ngyen. 2006. An OFDM-Specified lossless FFT architecture," IEEE Transaction on Circuits and Systems, vol. 53, no. 6, pp. 1235-1243.

[3] M. Frigo and S. Johnson. 2005. The design and implementation of fftw3. Proc. IEEE, vol. 93, no.2, pp. 216-231.

[4] D. Sepiashvili. 2000. Performance Models and Search Methods for Optimal FFT Implementations. M.Sc. thesis, Carnegie Mellon Univ. Pittsburgh, PA.

[5] J. Garcia, J.A. Michell and A.M. Buron. 1999. VLSI Configurable delay commutator for a pipeline Split radix FFT architecture. IEEE Transaction on Signal Processing, vol. 47, no. 11, pp. 3098-3107.

[6] J. Keiner, S. Kunis and D. Potts. 2005. Using nfft 3-A software library for various nonequispaced fast Fourier transforms. ACM Trans. Math. Softw. (TOMS), vol. 36, no. 4, pp. 19-19.

[7] I.R. Mactaggart, Jack and A. Mervyn.1984. A Single Chip radix-2 FFT butterfly architecture using parallel data distributed arithmetic. IEEE Journal of Solid State Circuits, vol. 19, no. 3, pp. 368-373.

[8] T. Lenart and V. Owall. 2006. Architectures for Dynamic data scaling 2/4/8k Pipeline FFT cores. IEEE Transaction on Very Large Scale Integration (VLSI) systems, vol. 14, no. 11, pp. 1286-1290.

[9] I.R. Mactaggart, Jack and A. Mervyn. 1984. A Single Chip radix-2 FFT butterfly architecture using parallel data distributed arithmetic. IEEE Journal of Solid State Circuits. vol. 19, no. 3, pp. 368-373.

[10] E.E. Swartzlander, W.K.W. Young and S.J. Joseph. 1984. A radix 4 delay commutator for fast Fourier transform processor implementation. IEEE Journal of Solid-State Circuits, vol. 19, no. 5, pp. 702-709.

[11] E. Dubois and A. Venetsanopoulos. 1978. A new algorithm for the radix 3 FFT. IEEE Transaction on Speech and Signal Processing, vol. 26, no. 3, pp. 222-225.

[12] S. Prakash and V.V. Rao. 1981. A new radix-6 FFT algorithm. IEEE Transaction on Acoustics, Speech and Signal Processing, vol. 29, no. 4, pp. 939-941.

[13] K.M.M. Prabhu and A. Nagesh. 1993. New radix-3 and -6 decimation-in-frequency fast Hartley transform algorithms. Canadian Journal of Electrical and Computer Engineering, vol. 18, no. 2, pp. 65-69.

[14] D. Takahashi. 2001. An extended split radix FFT algorithm. IEEE Signal Processing Letters, vol. 8, no. 5, pp. 145-147.

[15] Weihua Zhengand KenLi Li. 2013. Split Radix Algorithm for Length 6 DFT. IEEE Signal Processing Letters, vol. 20, no. 7, pp. 713-716.

[16] M. Omair Ahmad, M.N.S. Swamy, and Saad Bouguezel. 2007. A General Class of Split-Radix FFT Algorithms for the Computation of the DFT of Length-2m. IEEE Transaction on Signal Processing, Vol. 55, No. 8, pp. 4127-4138.

[17] Karsten Østergaard Noe, Michael Schacht Hansen, Thomas Sangild Sørensen and Tobias Schaeffter. 2008. Accelerating the Nonequispaced Fast Fourier Transform on Commodity Graphics Hardware. IEEE Transaction on Medical Imaging, Vol. 27, No. 4, pp. 538-547.

[18] Lun, Daniel Pak-Kong and Wan-Chi Siu.1993. Fast radix-3/9 discrete Hartley transform. IEEE Transaction on Signal Processing, vol. 41, no. 6, pp. 2494-2499.

[19] M.Z.A. Khan, S.A. Qadeer, M.Y. Khan. 2006. Fast Computation of Partial DFT for Comb Spectrum Evaluation. IEEE Signal Processing Letters, vol. 13, no. 12, pp. 721-724.

[20] Wwn-Chang Yeh and Chein-Wei Jen. 2003. High-speed and low-power split radix FFT. IEEE Transaction on Signal Processing, vol. 51, no. 3, pp. 864-874.