IJETST- Vol.||03||Issue||02||Pages 3534-3536||February||ISSN 2348-9480 2016

Open access Journal International Journal of Emerging Trends in Science and Technology
Impact Factor: 2.838 DOI: http://dx.doi.org/10.18535/ijetst/v3i02.06

Radix-2 DIT-FFT Algorithm for Real Valued Sequence

Authors
Supriya. P. Sarvade[1], Dr. Shridhar. K[2], Varun. P. Sarvade[3]
[1]PG Student, Department of Electronics & Communication Engineering,
Basaveshwar Engineering College, Bagalkot, Karnataka, India
Email: *supriyasarvade@gmail.com*
[2]Professor, Department of Electronics and Communication Engineering
Basaveshwar Engineering College, Bagalkot, Karnataka, India
Email: *shridhar.ece@gmail.com*
[3]Assistant Professor, Department of Computer Science & Engineering, Biluru Gurubasava Mahaswamiji Institute of Technology
Mudhol, District Bagalkot, Karnataka, India
Email: *varunpsarvade@gmail.com*

Abstract
This paper intends to present a Radix-2 Decimation-in-Time (DIT) algorithm for the computation of Fast Fourier Transform (FFT) of real-valued sequences which are integral part of all real time signal processing. The fact that the real valued sequence possesses conjugate symmetric property, is used here to reduce the computational complexity and memory requirements for the computation of Discrete Fourier Transform (DFT). A comparison of computational complexity between the proposed algorithm and existing FFT algorithms reveals the increased efficiency of the proposed algorithm.
Keywords: *DIT FFT, Symmetric FFT, Conjugate Symmetry, Real-valued series.*

Introduction

Around 1805, Carl Friedrich Gauss invented a technique for computing the coefficients of what is now called [4] a Discrete Fourier Series. Then in 1965, J. W. Cooley and J. W. Tukey published a paper [1] based on some other works of the early twentieth century which gave rise to a technique which is now known as the “Fast Fourier Transform.” This technique could be implemented on computer which could compute the coefficients of a discrete Fourier series faster than ever it was possible. Since, advent of Cooley-Tukey Fast Fourier transform (FFT), number of complex multiplications required to compute the N-point DFT of a complex sequence is reduced from N^2 to $(N/2)\log_2 N$.

In this paper for real valued sequences, an effort has been made to reduce the computational complexity further which resulted in a new version of the algorithm that is superior to the existing Cooley-Tukey FFT algorithm. The real-valued sequences [5] which are invariably involved in real-time signal processing, exhibit conjugate symmetry in Fourier domain giving rise to redundancies with respect to computations. This could be further exploited to reduce the computational complexity and the memory requirements.

FFT algorithm for real-valued sequence

Divide and conquer approach is used in decimation-in-time algorithm wherein the input sequence x(n) is divided into smaller and smaller subsequences. An N-point transform is divided into two N/2-point transforms, then each N/2-point transform is further divided into two N/4-point transforms, and this is continued till 2-point DFTs are obtained, as it was done by Cooley-Tukey in his algorithm [1]. In this paper an effort has been made to exploit conjugate symmetry property of Fourier Transform of a real sequence i.e. $X(k) = X^*(N-k)$.

Let x(n) represent a real-valued sequence of length N, where N is a power of 2.

Input sequence: *x(0), x(1), ... x(N-1)*

DFT of an N-point sequence is given by [1]-[2]

$$X(k) = \sum_{n=0}^{N-1} x(n)\, W_N^{nk}, \qquad 0 \le k \le N-1 \qquad (1)$$

Where,

$$W_N = e^{-j(2\pi/N)} \qquad (2)$$

W_N is complex and periodic with period N.

Decimating the sequence x(n) into two sequences of length N/2, one composing of even indexed samples of x(n) & the other of odd indexed samples, we get

$$X(k) = \sum_{n=0}^{N/2-1} x(2n)\, W_N^{2nk} + \sum_{n=0}^{N/2-1} x(2n+1)\, W_N^{(2n+1)k} \qquad (3)$$

$$\text{Let, } G(k) = \sum_{n=0}^{N/2-1} x(2n)\, W_{N/2}^{nk}, \qquad (4)$$

$$H(k) = \sum_{n=0}^{N/2-1} x(2n+1)\, W_{N/2}^{nk} \qquad (5)$$

Now, substituting (4) and (5) into (3), we get

$$X(k) = G(k) + W_N^k\, H(k), \qquad 0 \le k \le N-1 \qquad (6)$$

IJETST- Vol.||03||Issue||02||Pages 3534-3536||February||ISSN 2348-9480 2016

Where G(k) and H(k) are N/2-point DFT's of even and odd indexed samples respectively.

For $k = (k + N/2)$ equation (6) becomes,

$$X\left(k + \frac{N}{2}\right) = G\left(k + \frac{N}{2}\right) + W_N^{\left(k + \frac{N}{2}\right)} H\left(k + \frac{N}{2}\right) \qquad (7)$$

Since G(k) and H(k) are periodic with period N/2,

$$G\left(k + \frac{N}{2}\right) = G(k) \text{ and } H\left(k + \frac{N}{2}\right) = H(k) \qquad (8)$$

Hence,

$$X\left(k + \frac{N}{2}\right) = G(k) + W_N^{\left(k + \frac{N}{2}\right)} H(k) \qquad (9)$$

Using property $W_N^{\left(k + \frac{N}{2}\right)} = -W_N^k$, we get

$$X\left(k + \frac{N}{2}\right) = G(k) - W_N^k H(k) \qquad (10)$$

Applying conjugate symmetry property $X(k) = X^*(N - k)$ on equation (10), we get

$$X^*\left(N - \left(k + \frac{N}{2}\right)\right) = G^*(N - k) - W_N^k H^*(N - k) \qquad (11)$$

$$X^*\left(\frac{N}{2} - k\right) = G^*(N - k) - W_N^k H^*(N - k) \qquad (12)$$

Taking conjugate on both sides, we get

$$X\left(\frac{N}{2} - k\right) = G(N - k) - W_N^{-k} H(N - k) \qquad (13)$$

Replacing 'k' with $\left(\frac{N}{2} - k\right)$ into (13), we get

$$X(k) = G\left(N - \frac{N}{2} + k\right) - W_N^{\left(k - \frac{N}{2}\right)} H\left(N - \frac{N}{2} + k\right) \qquad (14a)$$

$$= G\left(k + \frac{N}{2}\right) - W_N^{\left(k - \frac{N}{2}\right)} H\left(k + \frac{N}{2}\right) \qquad (14b)$$

Using periodicity property of G(k) and H(k) equation (14b) can be rewritten as,

$$X(k) = G(k) - W_N^{\left(k - \frac{N}{2}\right)} H(k), \quad 0 \le k \le \left(\frac{N}{4} - 1\right) \qquad (15)$$

Similarly representing G(k) and H(k) as sum of two N/4-point DFTs.

$$G(k) = A(k) - W_N^{\left(k - \frac{N}{4}\right)} B(k), \quad 0 \le k \le \left(\frac{N}{8} - 1\right) \qquad (16)$$

$$H(k) = C(k) - W_N^{\left(k - \frac{N}{4}\right)} D(k), \quad 0 \le k \le \left(\frac{N}{8} - 1\right) \qquad (17)$$

Using equations (15), (16) and (17) we can draw signal flow graph as shown in the fig.1 Comparing this algorithm with FFT algorithm for complex sequence, it can be noted that the length of sequence X(k) has been reduced from N/2 to N/4, and sequences G(k) and H(k) from N/4 to N/8, indicating reduction in the complexity.

Signal flow resulting from the algorithm

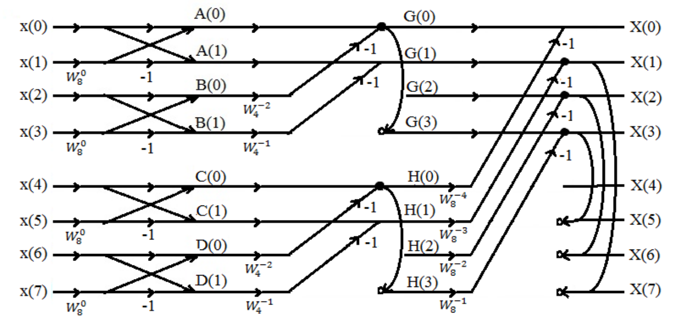

Fig.1. Signal flow graph of the proposed algorithm for N=8.Where ' ⭘ ' indicates complex conjugate of ' ● '

Number of complex multiplications required in first stage is

$$\eta_1 = \left(\frac{N}{2}\right)^2 + \left(\frac{N}{2}\right)^2 + N \qquad (18a)$$

$$= \left(\frac{8}{2}\right)^2 + \left(\frac{8}{2}\right)^2 + 8 = 40 \qquad (18b)$$

Similarly number of complex multiplications required after second stage is

$$\eta_2 = 4\left(\frac{N}{8}\right)^2 + 2\left(\frac{N}{4}\right) + \left(\frac{N}{2}\right) \qquad (19a)$$

$$= 4\left(\frac{8}{8}\right)^2 + 2\left(\frac{8}{4}\right) + \left(\frac{8}{2}\right) = 12 \qquad (19b)$$

And number of complex multiplications required after third stage which gives the total number of multiplications required for proposed algorithm is

$$\eta_3 = \left(\frac{N}{4}\right) + \left(\frac{N}{4}\right) + \left(\frac{N}{4}\right) \qquad (20a)$$

$$= \left(\frac{8}{4}\right) + \left(\frac{8}{4}\right) + \left(\frac{8}{4}\right) = 6 \qquad (20b)$$

Hence total number of complex multiplications required to evaluate 8-point DFT is equal to 6. Similarly for different values of N, a comparison of computational complexity is given in below table 1.

Table 1: Comparison of computational Complexity between existing FFT Algorithm and Proposed Algorithm

Number of points N	*FFT Algorithm*		*Proposed Algorithm*	
	Complex Multiplications	*Complex Additions*	*Complex Multiplications*	*Complex Additions*
4	4	8	2	4
8	12	24	6	12
16	32	64	16	32
64	192	384	96	192
128	448	896	224	448
256	1024	2048	512	1024
1024	5120	10240	2560	5120

If we compare the proposed algorithm with the Cooley-Tukey's FFT algorithm, it is clear that the computational complexity has been reduced to half for real valued sequences.

Arithmatic complexity

The total number of complex multiplications required to evaluate the N-point DFT with first stage is

$$\eta_1 = \left(\frac{N}{2}\right)^2 + \left(\frac{N}{2}\right)^2 + N \qquad (21a)$$

$$= \left(\frac{N^2}{2}\right) + N \qquad (21b)$$

The first and second terms of equation (21a) are number of complex multiplications required to compute N/2-point DFT of even and odd indexed samples respectively and the third term is the number of complex multiplications required for combining.

The total number of complex multiplications after second stage is

$$\eta_2 = 4\left(\frac{N}{8}\right)^2 + 2\left(\frac{N}{4}\right) + \left(\frac{N}{2}\right) \qquad (22a)$$

$$= \left(\frac{N^2}{16}\right) + N \qquad (22b)$$

Continuing this process of decimation till we end-up with 2-point DFTs, the total number of computations then becomes as given in the table 2.

Table 2: Computational Complexity of different Algorithms

	Direct Computation	FFT Algorithm	Proposed Algorithm
Complex Multiplications	N^2	$\frac{N}{2}log_2N$	$\frac{N}{4}log_2N$
Complex Additions	$N^2 - N$	$Nlog_2N$	$\frac{N}{2}log_2N$

Conclusion

All real-time signal processing algorithms encounter real signals as their input which exhibit conjugate symmetry in Fourier domain. The redundancy in this conjugate symmetry is exploited in the proposed algorithm which leads to further reduction in the computational complexity. The proposed algorithm is computationally more efficient than the existing Cooley-Tukey FFT algorithms with respect to real-time series processing.

References

1. W. Cooley and J. W. Tukey, "An algorithm for the machine calculation of complex Fourier series," Math. Coinput, vol. 19, pp.297—301, 1965.
2. V. Oppenheim and R. W. Schafer, "Digital Signal Processing," Englewood Cliffs, NJ: Prentice-Hail, 1975.
3. G. D. Bergland, "A fast Fourier transform for real-valued series," Comm.ACM,v.11,pp.703-710,1968.
4. Bergland, Glenn D, "A Radix-Eight Fast Fourier Transform Subroutine for Real-Valued Series," IEEE Transactions on audio and electroacoustics,17(2):138-44,1969.
5. B. Raja Sekhar and K. M. M. Prabhu, "Radix-2 Decimation-in-Frequency Algorithm for the Computation of the Real-Valued FFT," IEEE Transctions on signal processing, vol. 47, pp.1181-1184 April 1999.