7th National Conference on
"Role of Engineers in Nation Building" organized by VIVA Institute of Technology, Mumbai
(1st and 2nd March 2019)
International Journal of Engineering Research & Science (IJOER)
ISBN: [978-93-5321-411-1] ISSN: [2395-6992] [Vol-5, Issue-3, March- 2019]

A Survey on Different FFT Algorithms

Shoeb Shaikh[1], Chitra Takle[2], Mohini Ghotekar[3], Kushal Suvarna[4]

Department of EXTC, VIVA Institute of Technology, Virar

Abstract *— A time-domain sequence is converted into an equivalent frequency-domain sequence using discrete Fourier transform. A frequency-domain sequence can be converted back to an equivalent time-domain sequence using inverse discrete Fourier transform. Based on the Discrete Fourier transform (DFT), Fast Fourier transform (FFT) is a more productive & practical algorithm with few computations. FFT is a well-organized and logical tool in linear system analysis. FFT is used in everything from broadband to 3G and Digital TV to radio LAN's. Various algorithms have been developed to improve architecture of FFT approach. An overview of the work done by different FFT algorithms approach is present in this paper. The comparison of different architecture is also discussed.*

Keywords— Butterfly Architecture, Radix-2, FFT, DFT.

I. INTRODUCTION

FFT is an effective algorithm that optimizes machine computation and is widely used in wireless communication. The innovation of FFT is accredited to Cooley and Tukey in 1965. There are two main categories of FFT algorithm one is The Cooley-Tukey algorithm and the second is prime factor algorithm. Though some variations are being observed in the mapping process of full FFT into smaller sub transforms. The Cooley-Tukey algorithm are classified into two types of algorithm, mixed-radix algorithm and radix-2 algorithm. FFT take very less number of operations to compute the DFT. FFT and its inverse play a very important role in many DSP applications. FFT algorithm revolutionize DSP by reducing the complexity of DFT from N2 to Nlog2N which cut down the number of complex multiplication in comparison to the regular DFT. The FFT core can perform an N-point FFT in almost 2.4*N clock cycles. This technique finds applications in fields such as: ADSL, DAB, DVB, OFDM systems etc.

Radix-2 is the smallest transform used in 2-point DFT. This transform is the quickest way of calculating FFT. It processes a group of two samples. Only when N is a regular power of 2, Radix-2 algorithm are advantageous. Decimation-in-time (DIT) and decimation-in-frequency are the two radix-2 algorithms. So the FFT algorithm is an efficient algorithm to compute the DFT. Some of the FFT algorithms are as follows:

1) Cooley-Tukey algorithm
2) Prime factor algorithm
3) Winograd FFT algorithm
4) Rader‟s FFT algorithm
5) Bluestein‟s FFT algorithm

A lot of work has been done in the field of FFT. In this paper a survey has been made on comparison of different FFT algorithms, these include Cost, Complexity, Operating frequency, Operation, size of input, adder and multiplier.

II. ALGORITHMS FOR FFT COMPUTATION

In this chapter we discuss some of the renowned methodology to calculate FFT.

2.1 Cooley-Tukey:

It is the most widely accepted FFT algorithm. The basic concept used in this algorithm is to divide the N-point DFT into M, N/M point DFTs [1, 7, 12], for example if M=3 then it is divided into two N/4 DFTs which can be called as radix-4. Likewise we have

International Journal of Engineering Research & Science (IJOER)
ISBN: [978-93-5321-411-1] ISSN: [2395-6992] [Vol-5, Issue-3, March- 2019]

radix- 8, 16…etc. The concept shows repetitions, but it is the oldest implementations reorganize the algorithm to prevent precise repetition. The Cooley–Tukey algorithm separates the DFT into small unit of DFTs, so it can be combined randomly with any other algorithm for the DFT [2, 7, 12].

Two different Procedures are introduced to calculate a Cooley-Tukey FFT:
a) Decimation-in-frequency (DIF)
b) Decimation-in-time (DIT).

The results obtained from FFT are similar to that of DFT but requires comparatively with fewer calculations. These reduced calculations provides significant importance as FFT order increases. If FFT is applied with either the decimation-infrequency (DIF) or the decimation-in-time (DIT) process similar outputs are obtained. From Fig.1, DIT can be stated as a method that breaks the input sequence into shorter sequences i.e. Input sequences are decimated or in inverted order and output is in correct order. DIF (Decimation in frequency domain) is method that breaks the output sequence into shorter sequences i.e. Output sequences are decimated or in bit reversed order and input is in correct sequence as shown in Fig. 2. Therefore a repositioning is required in input side for DIT, also in DIF repositioning block is required at the output side. Decimation-in-time (DIT) and Decimation-in frequency has the Butterfly unit, shown in fig.3. It is the main part of DIT and DIF algorithm. [3]

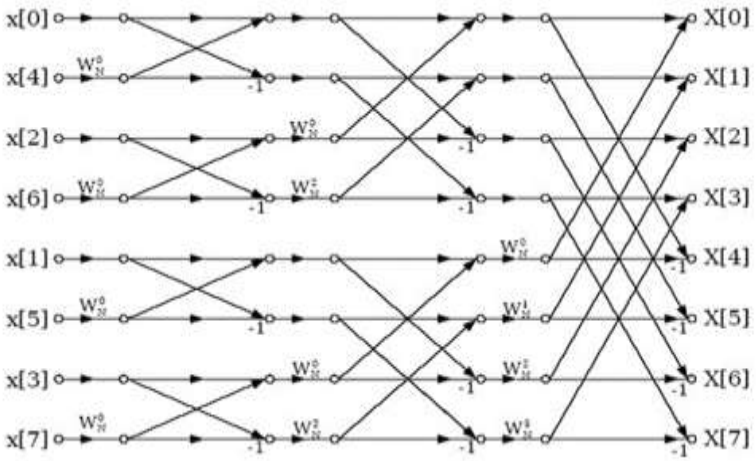

Fig.1. Decimation in Time Domain

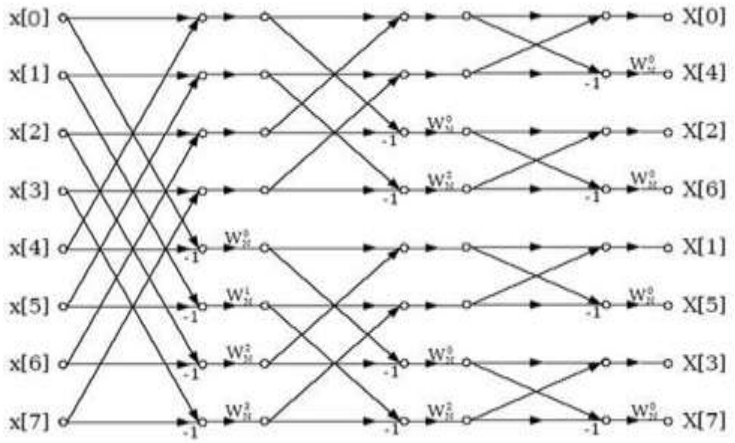

Fig.2. Decimation in Frequency Domain

International Journal of Engineering Research & Science (IJOER)
ISBN: [978-93-5321-411-1] ISSN: [2395-6992] [Vol-5, Issue-3, March- 2019]

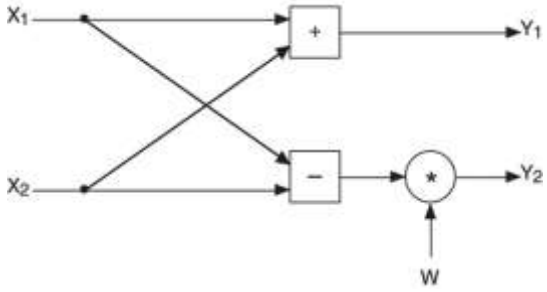

Fig.3. Butterfly Unit

2.2 Winograd Algorithm:

In this algorithm the fundamental concept is that it factorizes Z^N -1 into different polynomials which has coefficients of 1, 0, or −1, and so operation shows very less multiplications , so Winograd can be utilized to get least-multiplication FFTs. It is normally required in search of optimum algorithms for small factors [8]. Winograd stated that calculation method of DFT can done with only irrational multiplications, eventually reducing the number of mutiplications in operations.The hardware used now-a-days has multiplier blocks, dose meant to be inappropriate. It is mostly used with Rader's algorithm [4, 8, 12].

2.3 Rader- Brenner Algorithm:

Rader's algorithm came from Charles M. Rader of MIT Lincoln Laboratory, is a fast Fourier transform (FFT) algorithm that calculates the discrete Fourier transform (DFT) of elite sizes by confirming the DFT as a cyclic convolution [3,5,6]. In this the composite multiplication methods are reinstate by multiplication of complex number by real or imaginary number. It is determined by calculating an N-point DFT with N=2t [12,9]. As this algorithm completely based on the regular intervals of the DFT kernel, it is applicable for any other transform of prime order having same characteristics , similar to discrete Hartly transform or number theoretic transform.

Rader's algorithm is also defined as a peculiar type of Winograd's FFT algorithm, also called the multiplicative Fourier transform algorithm, which applies to an even larger class of sizes. However, for composite sizes such as prime powers, the Cooley–Tukey FFT algorithm is comparatively easier and easy to apply, so Rader's algorithm is applied in base cases of Cooley–Tukey's recursive decomposition of the DFT.

2.4 Brunn's Algorithm:

Bruun's algorithm [10] is a fast Fourier transform (FFT) algorithm has its roots from recursive polynomial-factorization method. It wasgiven for powers of two by G. Bruun in 1978.Its operation includes only real co-efficient till the end computation phase. Due to this reality, it was initially proposed as a way toprecisely compute the discrete Fourier transform (DFT) of real data [10]. Bruun's algorithm has not seen widespread use, however, as approaches based on the ordinary Cooley–Tukey FFT algorithm have been successfully adapted to real data with at least as much efficiency. Moreover, there is proof that Bruun's algorithm may be less accurate than Cooley–Tukey in the face of finite numerical accuracy.

Nonetheless, Bruun's algorithm shows an diffrent algorithmic framework that can show both it's and the Cooley–Tukey algorithm, and thus give a diffrent aspect on FFTs that allows combination of the two algorithms and other generalizations.

2.5 Prime factor Algorithm [Good-Thomas]:

7th National Conference on
"Role of Engineers in Nation Building" organized by VIVA Institute of Technology, Mumbai
(1st and 2nd March 2019)
International Journal of Engineering Research & Science (IJOER)
ISBN: [978-93-5321-411-1] ISSN: [2395-6992] [Vol-5, Issue-3, March- 2019]

The Prime-factor algorithm is a fast Fourier transform algorithm. It is known as the Good-Thomas algorithm that depicts the discrete Fourier transform of a size $N = N_1N_2$ as a two-dimensional $N_1 \times N_2$ DFT, although for the condition when N_1 and N_2 are approximately prime. These smaller transforms of size N_1 and N_2 can then be calculates by employing PFA coercively or by employing different FFT algorithm.

PFA should not be abashed with the mixed-radix generalization of the popular Cooley–Tukey algorithm, which also divides a DFT of size $N = N_1N_2$ into shorter transforms of length N_1 and N_2. The other algorithm can use any factors, but it has the demerits that it also requires added multiplications by roots of unity called twiddle factors, moreover, to the smaller transforms. However, PFA has the drawback that it works only for approximate prime factors (e.g. it is useless for power-of-two sizes) and that it needs a much difficult re-marking of the data depend on the Chinese remainder theorem (CRT). However, that PFA can be combined with mixed-radix Cooley–Tukey, with the former factorizing N into relatively prime components and the following handling repeated factors.

Recall that the DFT is defined by the formula:

$$X_k = \sum_{n=0}^{N-1} x_n \, e^{\frac{-2\pi i}{N} nk} \; k=0,\ldots\ldots\ldots\ldots N-1.$$

The PFA involves a re-indexing of the input and output arrays, which when replaced in the DFT equation changes it into two enclosed DFTs.

- Re-indexing

Suppose that $N = N_1N_2$, where N_1 and N_2 are relatively prime. In this case, we can define a bijective re-indexing of the input n and output k by:

$$n = n_1N_2 + n_2N_1 \; mod \; N,$$

$$k = k_1N_2^{-1}N_2 + k_2N_1^{-1}N_1 \; mod \; N,$$

where N1−1 denotes the modular multiplicative inverse of N_1 modulo N_2 and vice versa for N_2^{-1}; the indices k_a and n_a run from $0,\ldots,N_a-1$ (for a = 1, 2). These inverses only exist for approximately prime N_1 and N_2, and that norms are needed for the first mapping to be objective. While this re-indexing of k is called the CRT mapping, this method is known as the Ruritanian mapping,. The previous refers to the case where k is the solution to the Chinese remainder problem $k = k_2 \; mod \; N_2$ and $k = k_1 \; mod \; N_1$. The Ruritanian mapping can be applied for the output k and the CRT mapping for the input n.

- DFT re-expression

The above re-indexing is then substituted into the formula for the DFT, and in particular into the product nk in the exponent. Because $e^{2\pi i} = 1$, this exponent is evaluated modulo N: any $N_1N_2 = N$ cross term in the nk product can be set to zero. (Similarly, X_k and x_n are implicitly periodic in N, so their subscripts are evaluated modulo N.) The remaining terms give:

$$X_{k_1+N_2^{-1}N_2+k_2+N_1^{-1}N_1} = \sum_{n_1=0}^{N_1-1} \left(\sum_{n_2=0}^{N_2-1} x_{n_1N_2+n_2N_1} \, e^{\frac{-2\pi i}{N_2} n_2k_2} \right) e^{\frac{-2\pi i}{N_1} n_1k_1}$$

International Journal of Engineering Research & Science (IJOER)
ISBN: [978-93-5321-411-1] ISSN: [2395-6992] [Vol-5, Issue-3, March- 2019]

The inner and outer sums are simply DFTs of size N_2 and N_1, respectively. Here, we have used the fact that N1−1N1 is unity when evaluated modulo N2 in the inner sum's exponent, and vice versa for the outer sum's exponent.

III. COMPARATIVE ANALYSIS OF FFT ALGORITHMS

A comparative analysis of the FFT algorithms is depicted below. The table I give a performance comparison of the algorithms discussed. All algorithms of FFT have different disadvantages & merits. In cooley tukey algorithm DIT or DIF method is used to solve FFT which divides N point into M and N/M point DFT's, also cost and complexity is comparatively less but operating frequency is worst. To solve FFT Winograd algorithm uses convolution method. Operating frequency of winogard algorithm is good but cost and complexity is more. For smaller of N points winograd algorithm is more capable, but number of additions become more for larger N, so winograd algorithm is difficult to use. Rader and brenner's algorithm is efficient for prime length N, split radix method is used to solve FFT, also cost is high , complexity is more and operating frequency is worst of Rader and brenner's algorithm. Brunn's algorithm uses polynomial factorization method to solve FFT & has good operating frequency, also it is less complex & cost is also moderate. Prime factor algorithm is very much useful , also operating frequency is good and is cost effective but it is useful only when N1 & N2 are comparatively prime, this is the constrain that Prime factor algorithm has.

Table1: COMPARATIVE ANALYSIS OF THE FFT ALGORITHMS

ALGORITHMS	COST	COMPLEXITY	OPERATING FREQUENCY	OPERATION
Cooley-Tukey Algorithm	Low	Low	Worst	Divide N-point DFT into M.N/M pt DFT's
Winograd Algorithm	High	More	Good	Factorizes Z^N-1 into various polynomials
Rader-Brenner Algorithm	High	More	Worst	Computes N-pt DFT with $N=2^t$
Brunn's Algorithm	Moderate	Less	Good	Computes DFT of real coefficient
Prime factor Algorithm	Low	Moderate	Better	Re-expresses the DFT but only for the case where N_1 and N_2 are relatively prime.

IV. CONCLUSION

We studied different FFT algorithm like Cooley Tukey algorithm, Prime Factor/Good Thomas algorithm, Winograd algorithm, Brunn's algorithm, Rader and brenner's algorithm and Bluestein algorithms. Among the mentioned algorithms operating frequency of Winogard & brunn,s algorithm is great. On the other hand complexity & cost of Cooley tukey algorithm is less comparitively. Winograd algorithm is advantageous for small value of N, as N becomes large its complexity increases. Prime factor/Good Thomas algorithm has better operating frequency compared to cooley tukey algorithm but Prime factor/Good Thomas algorithm can be used only for relatively prime N1 and N2. Comparisons of these different algorithms are given in terms

International Journal of Engineering Research & Science (IJOER)
ISBN: [978-93-5321-411-1] ISSN: [2395-6992] [Vol-5, Issue-3, March- 2019]

of cost, complexity, operating frequency, size of input, size of adder and multiplier. Apart from these, there are other ways to make FFT algorithm more efficient like by replacing complex multiplier by adder and Subtractor unit & by reusing butterfly unit. For power efficient Fast Fourier transform split radix fast Fourier transform (SRFFT) is used in which clock gating approach is used.

REFERENCES

[1] Niladri Mandal, Souragni Ghosh (2012). A Modified Fast FFT Algorithm for OFDM. International Journal of Soft Computing and Engineering (IJSCE), Vol.1, Issue-6, January 2012.

[2] Chandan.M,S.L.Pinjare, Chandra Mohan Umapthy Chandan M, S.L Pinjare (2012). Optimized FFT Design using Constant Coefficient Multiplier. International Journal of Emerging Technology and Advanced Engineering, Vol. 2, Issue 6, June 2012.

[3] Senthil Sivakumar M & Banupriya M & Arockia Jayadhas S (2012). Design of Low Power High Performance 16-Point 2- Parallel Pipelined FFT Architecture. International Journal of Electronics,`Communication& Instrumentation Engineering Research and Development (IJECIERD). Vol. 2, Issue 3 September 2012.

[4] Arman Chahardahcherik, Yousef S. Kavian, Otto Strobel, and Ridha (2011). Implementing FFT Algorithms on FPGA (IJCSNS). International Journal of Computer Science and Network Security, Vol. 11, No.11, November 2011.

[5] Sneha N.Kherde, Meghana Hasamnis(2011). Efficient Design and Implementation of FFT . International Journal of Engineering Science & Technology, Vol. 3 Issue Sup, p10, February 2011.

[6] M. Kannan and S.K. Srivatsa(2007). Low Power Hardware Implementation of High Speed FFT Core. Journal of Computer Science. Vol. 3, Issue 6, 2007.

[7] Cooley, James W., and John W. Tukey (1965) An algorithm for the machine calculation of complex Fourier series, Math. Computer. Vol. 19, 1965.

[8] C. Rader(1965), Discrete Fourier Transform when the Number of Data Samples is Prime, Proceedings of the IEEE 56 , 1968, Vol. 19, No. 90 (April 1965).

[9] Brenner, N.; Rader, C. (1976). A New Principle for Fast Fourier Transformation. IEEE Acoustics, Speech & Signal Processing, Vol. 23, Issue 3, Jun 1976.

[10] Aniket Shukla, Mayuresh Deshmukh(2012), Comparative Study of Various FFT Algorithm Implementation on FPGA, International Journal of Emerging Trends in Signal Processing Vol.1, Issue 1, November 2012.