International Journal of Ethics in Engineering & Management Education
Website: www.ijeee.in (ISSN: 2348-4748, Volume 2, Issue 8, August 2015)

Implementation and Comparison of Radix-2, Radix-4 and Radix-8 FFT Algorithm

Vivek Jaladi
Dept. of ECE (Asst. Prof.)
Lingraj Appa Engineering College, Bidar, Karnataka
Vivec53@gmail.com

K.Pooja Swamy
Dept. of ECE (M.Tech. student)
Lingraj Appa Engineering College, Bidar, Karnataka
Pooja.swamy092@gmail.com

Abstract— In this paper, we have compared the Radix-2 (R2), Radix-4 (R4) and Radix-8 (R8) FFT's based on DIF algorithm. Implementation issues, butterfly architectures, equations involved and bit reversal computation in each of these Radix methods are discussed and elaborated. Also area and time utilization of these architectures along with their performance tradeoffs are discussed.

***Index Terms*— DFT, FFT, DIF, butterfly, Radix-2, Radix-4, Radix-8.**

1. INTRODUCTION

The Fourier Transform is a widely used method in signal processing to estimate spectral content of any signal. The Fourier Transform when applied to an aperiodic discrete signal rather than a continuous signal is called Discrete Time Fourier Transform (DTFT). But DTFT of an aperiodic discrete signal is continuous in frequency domain. Hence to use DTFT in computers, we sample the DTFT. This sampled DTFT is called Discrete Fourier Transform (DFT). Fast Fourier Transform (FFT) is an efficient algorithm for computing DFT. The DFT is defined by equation (1)

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{\left(\frac{-j2\Pi kn}{N}\right)} \quad (1)$$

Where, k = 0 to N-1 and N is the length of the DFT. Calculating the above equation requires N^2 Complex Multiplications and N*(N-1) additions. But by using FFT algorithms the amount of computation involved is reduced by using various architectures as discussed below.

The rest of this paper is organized as follows: Section (2), (3) and (4) describes Radix-2, Radix-4 and Radix-8 FFT algorithms and comparison between R2, R4 & R8 FFT algorithms. In Section (5) simulation results of R2, R4 and R8 are shown and after that we conclude the paper in Section (6).

2. RADIX-2 FFT ALGORITHM

Let us assume, we have $N=2^v$ points whose DFT is to be evaluated and we use the divide and conquer method. First we split the N-point data sequence into $V1$ (n) and $V2$ (n), such that the even-numbered samples of $x(n)$ goes to $V1$ (n) and the odd-numbered samples of $x(n)$ goes to $V2$ (n) respectively, that is

$$V_1(n) = X(2n)$$
$$V_2(n) = X(2n+1), \qquad n=0, 1,\ldots, N/2-1$$

Thus $V1$ (n) and $V2$ (n) are obtained by decimating $x(n)$ by a factor of 2 and hence the resulting FFT algorithm is called as decimation-in-time (DIT) algorithm.
The N-point Discrete Fourier Transform (DFT) of a sequence $x(n)$ is defined by equation (2)

$$X(k) = \sum_{k=0}^{N-1} x(n) W_N^{nk} \quad (2)$$

where, $k = 0, 1, 2, 3,\ldots, (N-1)$

where $x(n)$ is the time domain discrete input signal and $X(k)$ is the frequency domain DFT. The value n represent discrete time index, while k is the frequency domain index.
The twiddle factor W_N is given by

$$W_N = \exp^{-j\left(\frac{2\Pi}{N}\right)}$$
$$= \cos\left(\frac{2\Pi}{N}\right) - j\sin\left(\frac{2\Pi}{N}\right) \quad (3)$$

The Radix-2 DIF FFT algorithm breaks an N-point DFT calculation into a number of 2-point DFTs (2-point Butterflies). In Radix-2 DIF FFT algorithm splits into the following equations.

$$X(k) = X(2k) + X(2k+1) \quad (4)$$

Radix-2 FFT is decomposed to N/2-point DFT sequences as below:

$$X(2k) = \text{DFT}\ \frac{N}{2}\{a_0(n)\} \text{ and}$$

IJEEE

International Journal of Ethics in Engineering & Management Education

Website: www.ijeee.in (ISSN: 2348-4748, Volume 2, Issue 8, August 2015)

$$X(2k+1) = \text{DFT}\,\frac{N}{2}\,\{a_1(n)\}.$$

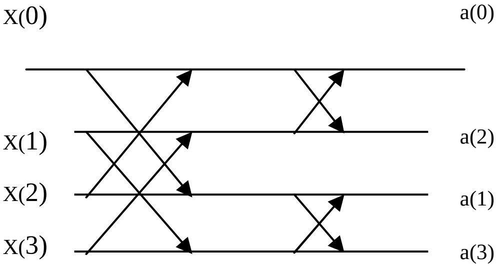

Fig 1: Radix-2 FFT Butterfly

The input sequence $a_0(n)$ and $a_1(n)$ in matrix form can be expressed as:

$$\begin{bmatrix} a_0(n) \\ a_1(n) \end{bmatrix} = T_{2,n} \begin{bmatrix} x(n) \\ x(n+\frac{N}{2}) \end{bmatrix} \quad (5)$$

where,

$$T_{2,n} = \begin{bmatrix} 1 & (1 & 1) \\ W_N^n & (1 & -1) \end{bmatrix} \quad (6)$$

and $n = 0, 1, 2,.., (N/2-1)$ can be modified as circular shift to $X(2k+1)$ as

$$X(k) = X(2k) + X(2k+1) \quad (7)$$

3. RADIX-4 FFT ALGORITHM

The Radix-4 DIF FFT algorithm breaks an N-point DFT calculation into a number of 4-point DFTs (4-point Butterflies) as compared to Radix-2 FFT algorithm. Hence it requires much less butterfly operations. Radix-4 DIF FFT algorithm can be represented as:

$$X(k)=X(4k)+X(4k+1)+X(4k+2)+X(4k+3) \quad (8)$$

where the decomposed N/4-point DFT sequences are defined as:

$$X(4k) = \text{DFT}\,\frac{N}{4}\,\{a_0(n)\},$$

$$X(4k+1)=\text{DFT}\,\frac{N}{4}\,\{a_1(n)\},$$

$$X(4k+2)=\text{DFT}\,\frac{N}{4}\,\{a_2(n)\} \text{ and}$$

$$X(4k+3)=\text{DFT}\,\frac{N}{4}\,\{a_3(n)\}$$

(9)

The input sequence $a_0(n), \ldots, a_3(n)$ in matrix form can be represented as:

$$\begin{bmatrix} a_0(n) \\ a_1(n) \\ a_2(n) \\ a_3(n) \end{bmatrix} = T_{4,n} \begin{bmatrix} x(n) \\ x(n+\frac{N}{4}) \\ x(n+\frac{N}{2}) \\ x(n+\frac{3N}{4}) \end{bmatrix} \quad (10)$$

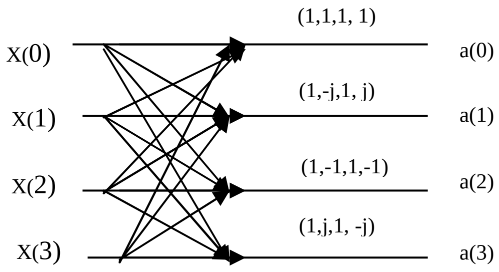

Fig 2: Radix-4 FFT Butterfly

Where, $$T_{4,n} = \begin{bmatrix} 1 & (1 & 1 & 1 & 1) \\ W_N^n & (1 & -j & -1 & j) \\ W_N^{2n} & (1 & -1 & 1 & -1) \\ W_N^{3n} & (1 & j & -1 & -j) \end{bmatrix} \quad (11)$$

And $n=0,1,2,\ldots,(N/4-1)$ can be modified by circular shift to X(4k+3) to get Radix-4 FFT as:

$$X(k) = X(4k) + X(4k+1) + X(4k+2) + X((4k-1))_N \quad (12)$$

Where, $$X((4k-1))_N = \begin{cases} N-1 & k=0 \\ 4k-1 & elsewhere \end{cases} \quad (13)$$

Now equation (11) becomes,

$$T_{4,n} = \begin{bmatrix} 1 & (1 & 1 & 1 & 1) \\ W_N^n & (1 & -j & -1 & j) \\ W_N^{2n} & (1 & -1 & 1 & -1) \\ W_N^{-n} & (1 & j & -1 & -j) \end{bmatrix} \quad (14)$$

International Journal of Ethics in Engineering & Management Education
Website: www.ijeee.in (ISSN: 2348-4748, Volume 2, Issue 8, August 2015)

4. RADIX-8 FFT ALGORITHM

The Radix-8 DIF FFT algorithm breaks the original input sequence into N/8 point sequences as in the following form.

$$X(k)=X(8k)+X(8k+1)+X(8k+2)+X(8k+3)+X(8k+4)+X(8k+5)+X(8k+6)+X(8k+7) \quad (15)$$

where the decomposed N/8-point DFT sequences are defined as:

$$X(8k) = \text{DFT}\frac{N}{8}\{a_0(n)\},$$

$$X(8k+1) = \text{DFT}\frac{N}{8}\{a_1(n)\}$$

$$X(8k+2) = \text{DFT}\frac{N}{8}\{a_2(n)\},$$

$$\vdots$$

$$X(8k+7) = \text{DFT}\frac{N}{8}\{a_7(n)\}$$

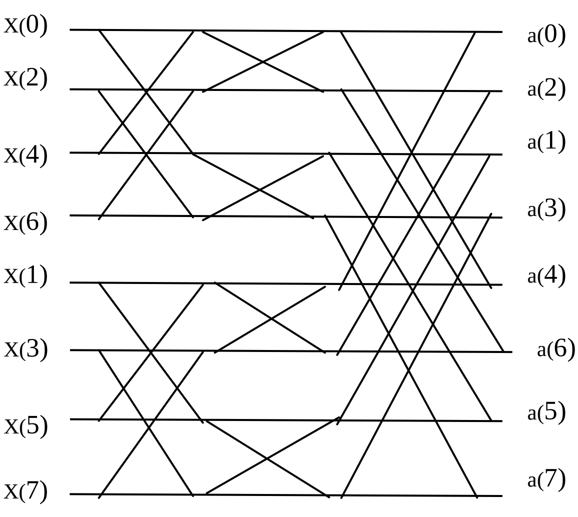

Fig 3: Radix-8 FFT Butterfly

The input sequence $a_0(n)$, …, $a_7(n)$ in matrix form can be expressed as:

$$a_i = S_{N,n}\, T_{8,n}\, x \quad (16)$$

$$\text{where, } a_i = \begin{bmatrix} a_0(n) \\ a_1(n) \\ a_2(n) \\ \vdots \\ a_7(n) \end{bmatrix} \quad (17)$$

$$S_{N,n} = \text{diag}([1\ W_N^{n}\, W_N^{2n}\, W_N^{3n}\, W_N^{4n}\, W_N^{5n}\, W_N^{6n}\, W_N^{7n}\ ]) \quad (18)$$

$$a = \begin{bmatrix} x(n) \\ x(n+\frac{N}{8}) \\ x(n+\frac{N}{4}) \\ \vdots \\ x(n+\frac{7N}{8}) \end{bmatrix} \quad n=0, 1, 2, \ldots, (N/8-1) \quad (19)$$

$$S_{N,n} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & W_N^{n} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & W_N^{2n} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & W_N^{3n} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & W_N^{4n} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & W_N^{5n} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & W_N^{6n} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & W_N^{7n} \end{bmatrix}$$

Equation (15) can be modified by circular shift to *X(8k+5)…X(8k+7)* to get Radix-8 FFT as:

$$X(k)=X(8k)+\ldots\ldots+X(8k+4)+X((8k-3))_N + X((8k-2))_N + X((8k-1))_N$$
(20)

$$S_{N,n} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & W_N^{n} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & W_N^{2n} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & W_N^{3n} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & W_N^{4n} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & W_N^{-3n} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & W_N^{-2n} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & W_N^{-n} \end{bmatrix}$$

where, $X((8k-b))_N$ can be expressed as:

$$X((8k-b))_N = \begin{cases} N-1 & k=0 \\ 8k-b & elsewhere \end{cases} \qquad (21)$$

Now equation (18) becomes

$$S_{N,n} = \text{diag}([1\ W_N^{n}\ W_N^{2n}\ W_N^{3n}\ W_N^{4n}\ W_N^{-3n}\ W_N^{-2n}\ W_N^{-n}]) \qquad (22)$$

After observing equation (22), we need to perform additional circular shift operation on last N/8 sequence at each stage and also number of twiddle factor evaluation will be less.

Table -1: Comparison of Radix-2, Radix-4 and Radix-8 FFT algorithm

If N=64 point	DFT	Radix-2	Radix-4	Radix-8
Real Multiplication	-	264	208	204
Real addition	-	1032	976	972
Complex Multiplication	4096	192	144	112
Complex Addition	4032	384	576	544
Number of stages	-	6	3	2
Number of Butterfly	-	32	16	8
Speed improvement factor	-	21.33	28.44	36.57

5. SIMULATION RESULTS

Upper graph in Fig-5 represents the simulation results of Radix-2 FFT whereas the lower graph represents inbuilt results. We can see that the both graphs are similar for Radix 2 FFT simulated result. Similarly Fig-6 and Fig-7 represent simulated result for Radix-4 and Radix-8 FFT respectively.

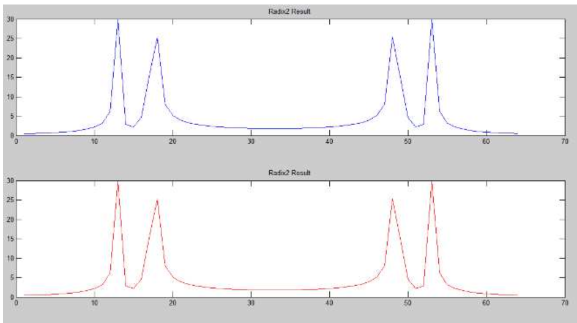

Fig-5: Radix-2 FFT for using 64 point

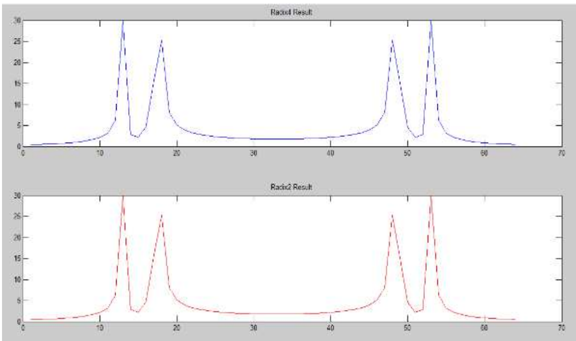

Fig -6: Radix-4 FFT for using 64 point

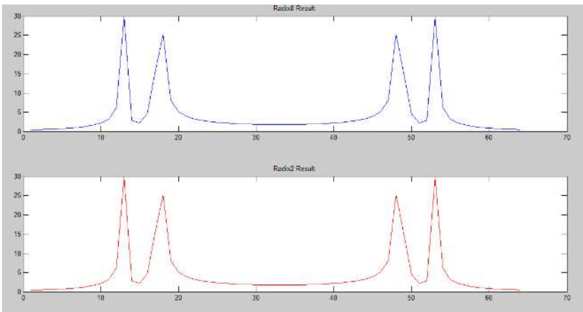

Fig-7: Radix-8 FFT for using 64 point

6. CONCLUSIONS

The Radix2, Radix4 and Radix8 FFT algorithms are studied and simulated in MATLAB Software. Radix4 and Radix8 have an advantage over Radix2 FFT algorithm because single Radix4 and Radix8 butterfly works of four, eight butterflies which will reduce the computation time and also ALU operation. These are achieved with circular shift operation and this modification also reduced the twiddle factors.

International Journal of Ethics in Engineering & Management Education
Website: www.ijeee.in (ISSN: 2348-4748, Volume 2, Issue 8, August 2015)

REFERENCES

[1]. J. G. Proakis and D. G. Manolakis, Digital Signal Processing Principles, Algorithms and Applications, Fourth Edition, Pearson Education, Upper Saddle River, New Jersey 07458.
[2]. Richard G. Lyons, Understanding Digital Signal Processing, Low Price Edition, Addison Wesley, Pearson Education, 2001.
[3]. Implementing the Radix-4 Decimation in Frequency (DIF) Fast Fourier Transform (FFT) Algorithm Using a TMS320C80 DSP, Charles Wu
[4]. Fast Fourier Transform algorithms with application, Todd Mateer, August 2008.
[5]. 64 point Radix-4 FFT Butterfly Realization using FPGA, International journal of engineering and innovative technology, October 2014.

Author Profiles

Vivek Jaladi ,has completed M.Tech. degree from Dayananda Sagar College of Engineering, Bengaluru, Karnataka. He is currently H.O.D in Department of ECE from LAEC ,Bidar , Karnataka.

K.Pooja Swamy, is currently pursuing her M.Tech. In VLSI Design and Embedded Systems in ECE Department of LAEC,Bidar,Karnataka. She has completed B.E. in ECE in GNDEC, Bidar, Karnataka. She is presently working on an efficient architecture using Radix 4 FFT in FPGA. Presently she is a Project Trainee at DLRL,Hyderabad.