Radix-2/4 Streamlined Real Factor FFT Algorithms

Shaik Qadeer(1), Mohammed Zafar Ali Khan(2) and Syed A. Sattar(3)

(1) *Department of Electrical and Electronics engineering* ,MJ College of Engineering, Hyderabad, India
{email:haqbei@gmail.com}
(2) *Department of Electrical engineering*, Indian Institute of Technology, Hyderabad, India
{email: zafar@iith.ac.in}
(3) *Department of Electronics and communication engineering*, RITS, Hyderabad, India
{email: syedabdulsattar1965@gmail.com }

***Abstract* — In this paper three Real Factor FFT algorithms are presented. Two of them are based on Radix-2 and one on Radix-4. The computational complexity of Radix-2 and Radix-4 is shown as order $\sim 4\frac{1}{2}N\log_2 N$ and $\sim 4\frac{1}{8}N\log_2 N$ respectively unlike their standard counterparts $\sim 5N\log_2 N$ and $\sim 4\frac{1}{4}N\log_2 N$. Moreover, the proposed algorithms also require fewer multiplications than their standard FFTs. We then show that the fixed point implementation of 'real factor' FFT can be modified, with unique scaling procedure, so that its noise to signal ratio (NSR) is lower than the NSR of standard FFT. Finally, implementation issues are presented which verify the suitability of the proposed 'real factor' FFT's.**

Keywords - Discrete Fourier Transform (DFT), Decimation in frequency (DIF), Fast Fourier Transform (FFT), and real-factor FFT

I. INTRODUCTION

The Discrete Fourier transform (DFT) is among the most fundamental operation in digital signal processing [7, 9]. When considering the alternate implementations, the FFT/IFFT algorithm should be chosen keeping in view the execution speed, hardware complexity, flexibility and precision [8, 9]. Most of the above mentioned parameters depend on the exact count of arithmetic operations (real additions and multiplications), herein called flops (floating-point operation), required for a DFT of a given size N which remains an intriguing unsolved mathematical question.

Among the numerous developments that followed Cooley and Tukey's [1] original contribution are the Winograd Fourier transform algorithm (WFTA) [2] and the real factor algorithms [3, 4] for reduction in the order of the multiplicative complexity. However both WFTA and 'real-factor' FFT did not meet expectations once implemented as the number of additions (and data transfers) also matter in the implementation. In addition Rader and Brenner's 'real factor' FFT's are ill-conditioned i.e., "small computational errors lead to large output errors"[3] due to the large values that the twiddle factor can take.

In this paper the solutions for both the problems of Rader and Brenner's 'real factor' FFT algorithm is presented. We first show that the arithmetic complexity (Multiplications plus additions, also known as the 'Flop count') of the Rader and Brenner's 'real factor' FFT can be reduced to about $\sim 4\frac{1}{2}N\log_2 N$ which is less than the arithmetic complexity of Cooley-Tukey radix-2 FFT, the corresponding Radix-4 is also discussed which takes $\sim 4\frac{1}{8}N\log_2 N$ which is again lesser than its counterpart. It is then shown that the modified 'real factor' FFT is free from ill-conditions as the magnitude of all the twiddle factors is less than 1 and hence has lower NSR than radix-2 Cooley-Tukey FFT. As a result the proposed FFT is more suitable as it requires only half the number of real multiplications.

The rest of the paper is organized as follows: In section II, modified DIT version of Rader and Brenner Radix-2 FFT that requires $\sim 5N\log_2 N$ flops is first presented in subsection II-A. In subsection II-B we derive the corresponding radix-2 DIF FFT. In subsection II-C, it is shown that the flop count can be further reduced to $\sim 4\frac{1}{2}N\log_2 N$. Notably this is the best known flop count that can be achieved by a radix-2 FFT. In subsection II-D equivalent Radix-4 FFT algorithm is covered. We then present in section III implementation issues of real factor FFT based on the radix-2 FFTs developed in this paper and conclusions are given in section IV.

II. NEW FFT

In this section, we first modify Rader and Brenner DIT FFT that requires $\sim 5N\log_2 N$ flops. The corresponding DIF version is derived and the algorithm is further modified so that the flop count is reduced down to $\sim 4\frac{1}{2}N\log_2 N$. Finally a similar radix-4 DIF FFT algorithm is also derived.

A. DIT version of Rader and Brenner FFT:

The essence of Rader and Brenner's DIT FFT is as follows: Let $\{A_k\}$ denotes N-point DFT of the sequence a_n of $N=2^M$ i.e. $\{A_k\}=DFT_{N/2}\{a_n\}$. Then radix-2 DIT FFT is given by

$$A_k = B_k + W_N^k D_k, \quad k = 0..N-1 \tag{1}$$

where $W_N^k=exp\ (-j2\pi/N)$ and the $N/2$-point sequences are defined as $\{b_n\} = \{a_{2n}\}_{n=0}^{\frac{N}{2}-1}$, $\{d_n\} = \{a_{2n+1}\}_{n=0}^{\frac{N}{2}-1}$ $\{B_k\}=DFT_{N/2}\{b_n\}$, and $\{D_k\}=DFT_{N/2}\{d_n\}$. Note that for a $N/2$-point sequence, if the index *k, n is* greater than $N/2-1$ we assume $k,\ n\ mod\ N/2$. Since in general D_k and W_N^k are both complex in nature, the basic butterfly operation indicated in (1) requires '4' real multiplications and '4' real additions. Rader and Brenner introduced the following sequence [3]

$$c_n = d_n - d_{n-1} + q_n, \quad n = 0..\frac{N}{2}-1 \tag{2}$$

where $q_n = 2/N \sum_{m=0}^{N/2-1} d_m$. It follows that (1) can be rewritten as

$$A_0 = B_0 + C_0,\ A_{N/2} = B_0 - C_0,\ and$$

$$A_k = B_k - \frac{j}{2} csc\left(\frac{2\pi k}{N}\right) C_k,\ k \neq 0\ \&\ N/2 \tag{3}$$

where $\{C_k\}=DFT_{N/2}\{c_n\}$. and $csc(x)$ is the cosecant of x.

If we change the sign to plus in the definition of c_n then the coefficient of C_k in (3) is $1/2sec(2\pi k/N)$ where $sec(x)$ is secant of x. In either case, (3) requires 2 real multiplications and 2 real additions. But this is at the expense of more additions to find q_n and c_n from b_n. Therefore total real additions ($A(N)$) will be $A(N)=2N+N+2N=5N$. The real multiplications ($M(N)$) are given by $M(N)=N$ and the total flop count ($T(N)$) is given by $T(N)=\sim 6Nlog_2N$.

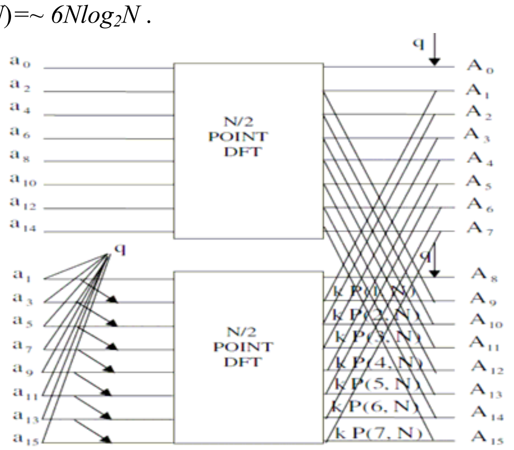

Fig. 1The decomposition of N point DFT into two $\frac{N}{2}$- point DFTs of new DIT Radix-2 FFT algorithm for N=16.

To further reduce the arithmetic complexity here we modify c_n as

$$c_n = d_n - d_{n-1}, \quad n = 0..\frac{N}{2}-1 \tag{4}$$

It follows that (3) is still valid with the exception that C_0 , is defined as

$$C_0 = \sum_{n=0}^{\frac{N}{2}-1} c_n \tag{5}$$

Denote $P(n,N) = {}^{-j}/_{2}\, csc\left(\frac{2\pi n}{N}\right)$, then the DIT computation for the new algorithm is schematically represented for N=16 is shown in Figure 1. The arithmetic complexity of this DIT FFT will become $T(N) = 5N\log_2 N - N$ due to the reduction in the number of addition to compute c_n using (4) instead of (2). Note that the difference between Rader and Brenner's FFT and our proposed FFT is in the computation of C_0. In Rader and Brenner's FFT $1/_N C_0 = q$ is added to all c_n which requires '(2N-2)', real additions, while in our version C_0 is directly computed using (5) which requires '(N-2)' real additions.

B. Derivation of DIF version of New FFT

According to the principle of DIF FFT, the even and odd DFT coefficients are given by

$$A_{2k} = DFT_{\frac{N}{2}}\left\{a_n + a_{n+\frac{N}{2}}\right\},\ k = 0..\frac{N}{2}-1 \tag{6}$$

$$A_{2k+1} = DFT_{\frac{N}{2}}\left\{a_n - a_{n+\frac{N}{2}}\right\} W_N^n,\ k = 0..\frac{N}{2}-1 \tag{7}$$

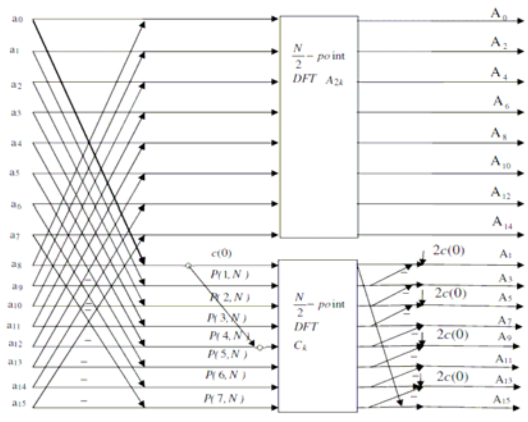

Fig.2 The decomposition of N point DFT into two - point DFTs for new DIF Radix-2 FFT algorithm for N=16.

To eliminate the complex multiplication factor from (7), we define the sequence $\{c_n\}$ as

$$c_n\left[1-W_N^{2n}+\delta(n)\right]=\left\{a_n-a_{n+\frac{N}{2}}\right\}W_N^n,\ n=0..\frac{N}{2}-1 \tag{8}$$

where δ(n) is unit impulse function. Equation (8) can be simplified as

$$c_0=\left\{a_n-a_{n+\frac{N}{2}}\right\}, \tag{9}$$

$$c_n=\frac{-j}{2}csc\left(\frac{2\pi n}{N}\right)\left\{a_n-a_{n+\frac{N}{2}}\right\},\ n=1..(\frac{N}{2}-1) \tag{10}$$

Applying N/2 –point DFT on both sides of (8) we get

$$C_k-C_{k+1}+c_0=A_{2k+1},k=0..\frac{N}{2}-1 \tag{11}$$

where $C_k=DFT_{\frac{N}{2}}\{c_n\}$. The Radix-2 DIF computation for the new algorithm is schematically represented for N=16 is shown in Figure 2.

The number of multiplications can be calculated from (10). For each value of 'n', '2' real multiplications are needed and the total number of real multiplications with given two N/2 -points DFTs is (N-2). Therefore the total number of real multiplications for a N-point DFT, is given by

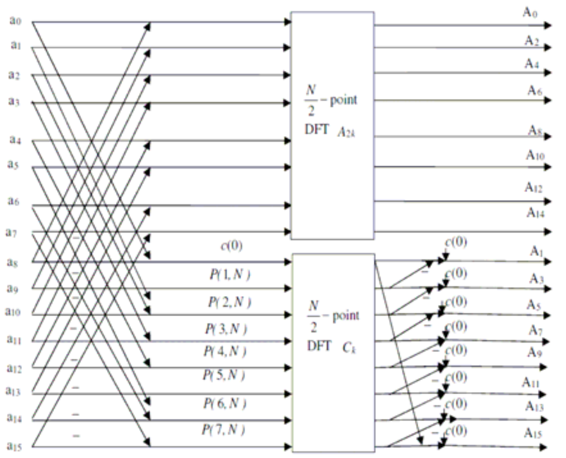

Fig.3 the decomposition of N point DFT into two $\frac{N}{2}$- point DFTs for new modified DIF Radix-2 FFT algorithm for N=16.

$$M(N)=\begin{cases}2M(\frac{N}{2})+N-2, & if\ N=2^n\geq 2\\ 0, & if\ N=1\end{cases} \tag{12}$$

Solving (12) by repeated substitutions we have

$$M(N)=N\log_2 N-N \tag{13}$$

The number of real additions given two N/2-points DFTs is calculated as follows

- 2(*N*/2) real additions for evaluating (6) for n=0..*N*/2-1.
- 2(*N*/2+1)real additions for evaluating (8) for n=0..*N*/2-1.
- 4(*N*/2)real additions for evaluating (11) for n=0..*N*/2-1.

It follows that

$$A(N)=\begin{cases}2A(\frac{N}{2})+4N, & if\ N=2^n\geq 2\\ 0, & if\ N=1\end{cases} \tag{14}$$

Solving (14) by repeated substitutions we have

$$A(N)=4N\log_2 N \tag{15}$$

The total flop count is then given by $T(N)=5N\log_2 N-N$. Importantly, the DIF version of the FFT presented in sub section II-A and II-B requires the same number of computations (real adds + real multiplies) as normal Radix-2 DIT or DIF FFT while requiring lesser multiplies. Also as compared to DIT version of Rader and Brenner's FFT the modified DIT shown in subsection II-A and DIF in II-B requires N fewer real additions.

C. *Modified DIF version of new FFT*

In this sub section we show that the computational complexity can be reduced down further. If we add c_0 to (10) then for *n*= (*N*/4),

$$c_{\frac{N}{4}}=\frac{-j}{2}csc(\pi/2)\left\{a_{N/4}-a_{3N/4}\right\}+c_0 \tag{16}$$

And rest of the sequence $\{c_n\}$, *n = 0…N/4-1, N/4 + 1, .., N/2-1* is given by (9)-(10). Equation (11) for this case modifies to

$$A_{2k+1}=\begin{cases}C_k-C_{k+1}+2c_0, & for\ even\ values\ of\ k\\ C_k-C_{k+1} & otherwise\end{cases} \tag{17}$$

The DIF computation for the new algorithm is schematically represented for N=16 are shown in Figure 2 and Figure 3. With these substitutions it is obvious that the numbers of real multiplications are same. However, as compared to the sub section II-A and II-B the number of real additions are reduced by N/2-2. Accordingly, the number of real additions is given by

$$A(N)=\begin{cases}2A(\frac{N}{2})+\frac{7}{2}N-2, & if\ N=2^n\geq 4\\ 0, & if\ N=1\end{cases} \tag{18}$$

Solving (18) by repeated substitutions we have

$$A(N)=\frac{7}{2}N\log_2 N-N \tag{19}$$

The total flop count is given by $T(N)=4\frac{1}{2}N\log_2 N-2N$. Importantly a flop count of this Radix-2 FFT is less than any other Radix-2 FFTs (including standard DIT and DIF Radix-2 FFTs).

D. New Radix-4 DIF FFT:

According to the principle of Radix-4 DIF FFT, the even and odd DFT coefficients are given by

$$A4k=DFT_{\frac{N}{4}}\left\{a_n+a_{n+\frac{N}{2}}+a_{n+\frac{N}{4}}+a_{n+3\frac{N}{4}}\right\}, \tag{20}$$

$$A4k+2=DFT_{\frac{N}{4}}\left\{a_n+a_{n+\frac{N}{2}}-\left(a_{n+\frac{N}{4}}+a_{n+3\frac{N}{4}}\right)\right\}W_N^{2n}, \tag{21}$$

$$A4k+1=DFT_{\frac{N}{4}}\left\{\left(a_n-a_{n+\frac{N}{2}}\right)-j\left(a_{n+\frac{N}{4}}-a_{n+3\frac{N}{4}}\right)\right\}W_N^{n}, \tag{22}$$

$$A4k+3=DFT_{\frac{N}{4}}\left\{\left(a_n-a_{n+\frac{N}{2}}\right)+j\left(a_{n+\frac{N}{4}}-a_{n+3\frac{N}{4}}\right)\right\}W_N^{3n}, \tag{23}$$

for $k=0..N/4-1$

To eliminate the complex multiplication factor from (21), we similarly define the sequence $\{c_n\}$ as

$$c_n\left[1-W_N^{4n}+\delta(n)\right]=\left\{a_n+a_{n+\frac{N}{2}}-\left(a_{n+\frac{N}{4}}\ a_{n+3\frac{N}{4}}\right)\right\}W_N^{2n},$$

$$n=0..\frac{N}{4}-1 \tag{24}$$

where δ(n) is unit impulse function. Equation (24) can be simplified as

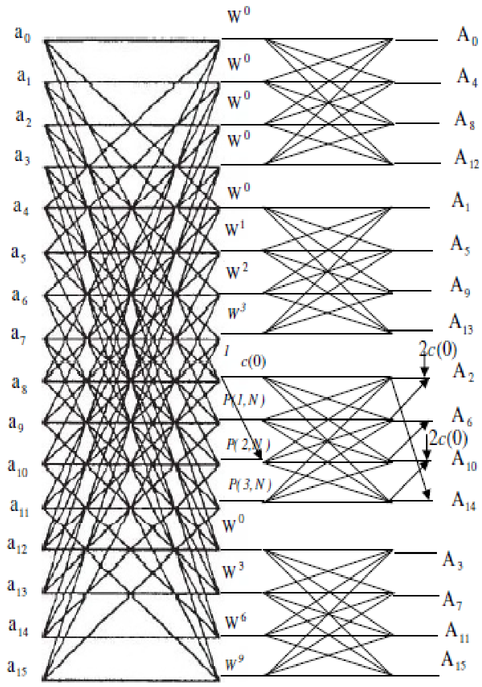

Fig.4. The complete in place computation diagram of new DIF Radix-4 FFT algorithm for N=16.

$$c_0=\left\{a_n+a_{n+\frac{N}{2}}-\left(a_{n+\frac{N}{4}}\ a_{n+3\frac{N}{4}}\right)\right\},$$

$$c_n=\frac{-j}{2}csc\left(\frac{4\pi n}{N}\right)\left\{a_n+a_{n+\frac{N}{2}}-\left(a_{n+\frac{N}{4}}\ a_{n+3\frac{N}{4}}\right)\right\},$$

$$n=1..N/4-1 \tag{25}$$

Applying N/4 –point DFT on both sides of (24) we get

$$C_k-C_{k+1}+c_0=A_{4k+1},k=0..\frac{N}{4}-1 \tag{26}$$

where $C_k=DFT_{\frac{N}{4}}\{c_n\}$ *and* $C_k=DFT_{\frac{N}{4}}\{c_n\}$, As in subsection II-C, if we add c_0 to (25) then for n=*N*/8

$$c_{\frac{N}{8}}=\frac{-j}{2}csc\left(\frac{\pi}{4}\right)\left\{a_{\frac{N}{8}}+a_{\frac{N}{8}+\frac{N}{2}}-\left(a_{\frac{N}{8}+\frac{N}{4}}+a_{\frac{N}{8}+3\frac{N}{4}}\right)\right\}+c_0,$$

And for the rest of the sequence $\{c_n\}$, $n=0\ldots N/8-1,\ N/8+1,\ \ldots,\ N/4-1$ is given by (25). Equation (26) for this case modifies to

$$A_{4k+2} = \begin{cases} C_k - C_{k+1} + 2c_0, \text{ for even values of } k \\ C_k - C_{k+1} \text{ otherwise} \end{cases} \quad (27)$$

Methods	M(N)	T(N)
Standard Radix-2	$2N\log_2 N$	$5N\log_2 N$
RB-FFT	$N\log_2 N$	$6N\log_2 N$
Proposed Radix-2	$N\log_2 N$	$\approx 4\frac{1}{2}N\log_2 N$
Standard Radix-4	$3N\log_2 N$	$\approx 4\frac{1}{4}N\log_2 N$
Proposed Radix-4	$2\frac{1}{2}N\log_2 N$	$\approx 4\frac{1}{8}N\log_2 N$

Table I. Computational Complexity of various FFT's

Denote $P(n,N) = {}^{-j}\!/_{2}\, csc\left(\frac{4\pi n}{N}\right)$, then the DIF computation for the new algorithm is schematically represented for N=16 is shown in Figure 4. The total addition for each butterfly will be $A(N) = 23\frac{N}{4}$ and total multiplication will be $M(N) = 10\frac{N}{4}$. Hence the total flop counts $T(N) = 16\frac{1}{2}\frac{N}{2}$ per butterfly. The cost of the new Radix-4 FFT algorithm can now be represented by the following recurrence ignoring nontrivial multiplications:

$$T(N) = \begin{cases} 4T\left(\frac{N}{4}\right) + 16\frac{1}{2}\frac{N}{2}, & if\ N = 4^n \geq 4 \\ 16, & if\ N = 4 \end{cases} \quad (28)$$

Solving (28) by repeated substitutions we have

$$T(N) = 4\frac{1}{8} N \log_2 N \quad (28.1)$$

E. Comparison of computational Complexity of various FFT algorithms:

The computational complexity comparison for Radix-2/4 FFTs is given in Table. I (Note that RF-FFT means Real Factor while RB-FFT means Rader and Brenner's FFT). Observe that the new DIF version of modified real factor FFT presented in section II requires not only lesser multiplications but also the total flop counts (T (N)).

III. Implementation issues of proposed FFT

In this section we consider the implementation issues of 'real factor' FFTs presented in section II (Note that we cover here only Radix-2 DIF and other can be similarly obtained [11, 12, and 15]). The noise to signal ratio (NSR) of Rader and Brenner's algorithm is known to be very poor because of the large values of $csc\left({}^{2\pi n}\!/_{N}\right)$ [3]. Accordingly we first modify the real factor FFT for fixed point implementation in section III-A to eliminate the problem of ill computation; then we discussed the remaining implementation issues in III-B and expected execution times is given in section III-C.

A. Modification to real factor FFT for fixed point implementation

To eliminate the problem of ill-conditioned we proposed a scaling method to (9)-(10) as

$$\hat{c}_0 = \frac{c_0}{2^l} = \left\{ a_0 - a_{\frac{N}{2}} \right\} \quad (29)$$

$$\hat{c}_n = \frac{c_n}{2^l} = \frac{-j\, csc\left(\frac{2\pi n}{N}\right)}{2^{l+1}} \left\{ a_n - a_{n+\frac{N}{2}} \right\}, \quad n = 1..\left(\frac{N}{2} - 1\right) \quad (30)$$

where l is chosen such that $csc(2\pi/N) \leq 2^{l+1}$. It follows that $csc(2\pi/N)/2^{l+1} \leq 1,\ for\ all\ n$. This modification ensures that the real multiplication factors are less than 1. Applying DFT on both sides of (8), with this modification we get

$$\left(\hat{C}_k - \hat{C}_{k+1} + \hat{c}_0\right) 2^l = A_{2k+1},\ k = 0..\frac{N}{2} - 1 \quad (31)$$

where $\hat{C}_k = DFT_{\frac{N}{2}}\left\{ \hat{c}_n \right\}$ and $DFT_{\frac{N}{2}}\left\{ \hat{c}_n \delta(n) \right\} = \hat{c}_0$.

The purely imaginary twiddle factor is now given by $P(n,N) = \frac{-j}{2^{l+1}} csc\left(\frac{2\pi n}{N}\right)$. A similar algorithm with purely real twiddle factor can be similarly developed. Note that in both the cases $0 \leq P(n,N) \leq 1$. The advantage of the presented fixed point algorithm can be observed from (31) wherein the effect of large values of $csc(\theta)$ has been captured in l and the effective precision at the output is $l+b$ bits [6, 10].

A. In-place computation of proposed FFT algorithm:

Figure 5 shows the in-Place computation diagram of the proposed real factor FFT, which implements (6) through (17) for N=8. It is observed that Part-I of algorithm is similar to Cooley-Tukey's radix-2 DIF FFT algorithms, [1] except the twiddle factor 'P(n ,N)', which is defined earlier (but not shown in algorithm). Once a butterfly operation is performed on a pair of complex number (a, b) to produce (A, B), there is no need to save the input pair (a, b), except the first value of vector 'c' as shown in Figure 5. Hence we can store the results (A, B) in the same locations as (a, b).

DOI 10.5013/IJSSST.a.13.06.01

ISSN: 1473-804x online, 1473-8031 print

Consequently we require a fixed number $2N+\left(\frac{N}{2}-1\right)$ of storage registers in order to store the results (note that N is complex number) of the computations at each stage. Since $2N+\left(\frac{N}{2}-1\right)$ storage locations are used throughout the computation of the N-Point DFT, we can say that the computations are done in-place. A second observation is need of data shuffling (DS) in Part-II of proposed algorithm.

TABLE II. Expected Execution time of FFT variants for Radix-2, N=2^{υ}

S.No	Problem Length	C-T in m sec	Proposed R2DIF in m sec	Proposed R4DIF in m sec
	$N=2^{\upsilon}$	$5N\log_2 N$	$4\frac{1}{2}N\log_2 N-2N$	$\approx 4\frac{1}{8}N\log_2 N$
1	2	0.0005	0.0005*	0.0005*
2	4	0.0020	0.0014	0.0013
3	8	0.0060	0.0046	0.0042
4	16	0.0160	0.0128	0.0116
5	32	0.0400	0.0328	0.0298
6	64	0.0960	0.0800	0.0728
7	128	0.2240	0.1888	0.1720
8	256	0.5120	0.4352	0.3968
9	512	1.1520	0.9856	0.8992
10	1024	2.5600	2.2016	2.0096
11	2048	5.6320	4.8640	4.4416
12	4096	12.2880	10.6496	9.7280
13	8192	26.6240	23.1424	21.1456
14	16384	57.3440	49.9712	45.6704

*Note that our algorithms works for N>2.

There is a need of data decimation after each additional addition as shown in Figure 6, as at the end of Part-I the order of data of our algorithm is bit reversed. We define this decimation of data as DS. The DS needed for N=8 is shown in Figure 6. Observe that '2' DS operations are required for N=8. In general if $N=2^{\upsilon}$, then $(\nu-1)$ DS operations are required to compute N-point FFT. Note that the number of times data decimations/shuffling are needed in the computation of DFT using C-T FFT is $(\nu-1)$. Hence the computation requirement for both DS and Bit reversed operations are equal.

B. *Expected Execution Time of Various FFT algorithms:*

To help things in perspective, it is instructive to see the differences in operation counts in terms of expected execution times, [13]. The above expected times are calculated by considering the case of TMS320C500 DSP processor whose speed of operation is 50ns (20MIPS) for proposed Radix-2 FFT. Assuming that both multiplication and addition takes same time we get the results as shown in above Table II.

IV. CONCLUSION

In this paper we have shown that the arithmetic complexity of real factor FFTs is significantly less than that of standard FFT. We have shown that these FFTs have a significant advantage compared to corresponding standard FFTs. When it comes to fixed point implementation, as it has lower number of noise sources (real multiplications) and hence lower NSR. This algorithm is of practical importance as Radix-2/4 FFTs are generally preferred in ASIC implementation and for software defined radio (cognitive radio) applications due to ease of re-configurability. Over all these factors are in favor of real factor FFTs. However, the disadvantage is additional memory storage of C_0 required for each stage. But as the cost of memory is decreasing storage may not be a significant factor.

A part of this paper was presented at EUSIPCO 2010, [14].

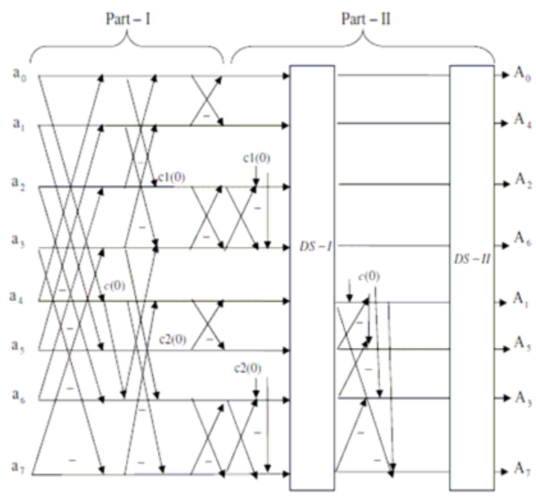

Fig. 5 In-Place computation diagram for N=8 of the Proposed Real Factor FFT algorithm.

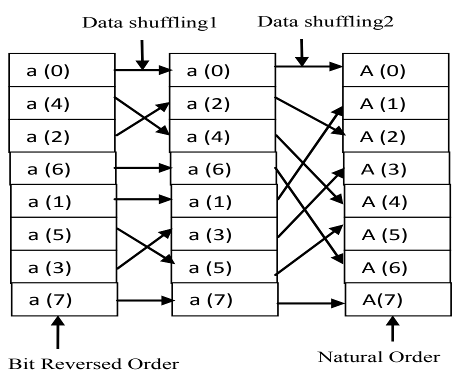

Fig.6 Shuffling of Data and bit reversal

DOI 10.5013/IJSSST.a.13.06.01 ISSN: 1473-804x online, 1473-8031 print

REFERENCES

[1] J. W. Cooley and J. W. Tukey, “An algorithm for the machineComputation of the complex Fourier series,” *Math. Computation*, vol. 19, pp. 297301, Apr. 1965.

[2] S. Winograd, “On computing the discrete Fourier transform,” *Proc. Nat. Acad. Sci. USA*, Vol. 73, April 1976, pp. 1005–1006.

[3] C. M. Rader and N. M. Brenner, “A new principle for fast Fourier transformation,” *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 24, pp. 264-265, 1976.

[4] K.M. Cho and G.C. Temes, “Real-factor FFT algorithms,” *Proc.ICASSP 78, Tulsa, UK*, April 1978, pp. 634–637.

[5] A. V. Oppenheim and C. J. Weinstein, “Effects of finite registerlength in digital filtering and the fast Fourier transform,” *Proc. of IEEE*. 1972, pp. 957–976

[6] T. Tran, B. Liu, “Fixed-point fast Fourier transform error analysis”, *IEEE Trans. on ASSP*, 1976, vol.24(6), pp. 563–573.

[7] W. M. Gentleman and G. Sande, “Fast Fourier transform for fun and profit,” in 1966 *Fall Joint Cornput. Conf., AFIPS Proc.*,v ol. 29. New York: Spartan, pp. 563–578.

[8] Wade Lowdermilk and Fred Harris, “Finite Arithmetic Considerations for the FFT Implemented in FPGA-Based EmbeddedProcessors in Synthetic Instruments”, *In IEEE Instrumentation and Measurement Magazine*, August 2007.

[9] W.-H. Chang and T. Nguyen, On the Fixed-Point Accuracy Analysis of FFT Algorithms, *IEEE Transactions On Signa Processing*, Vol. 56, No. 10, pp. 4673-4682, October 2008.

[10] P. D. Welch, “A fixed point fast Fourier transform error analysis”, *IEEE Trans. Audio Electroacoust*., vol. AU-17, pp. 151–157, June 1969.

[11] A. V. Oppenheim and R. Schafer,“ *Digital Signal Processing*”.Pearson Education, 2004.

[12] Sanjit Mitra, “*Digital Signal Processing: A Computer Approach*”3rd edition, Chapter 2, McGraw- Hill, 2006.

[13] Eleanor Chu and Alan George, “*Inside the FFT Black Box: Serial and Parallel FFT algorithms*”Page No.282, Appendix A, CRC Press.

[14] M.Z.A. Khan and Shaik Qadeer, “Streamlined Real Factor *EUSIPCO2010, Aalborg,Denmark*,August23-27,2010.

[15] J.G. Proakis and D.G. Manolakis,"*Digital Signal ProcessingPrinciples,Algortihms,and applications*",Pearson 3rd Edition.

DOI 10.5013/IJSSST.a.13.06.01 ISSN: 1473-804x online, 1473-8031 print