Arithmetic Complexity of the Split-Radix FFT Algorithms

Saad Bouguezel, M. Omair Ahmad, Fellow, IEEE, and M.N.S. Swamy, Fellow, IEEE

Department of Electrical and Computer Engineering
Concordia University 1455 de Maisonneuve Blvd. West
Montreal, P.Q., Canada H3G 1M8
E-mail: {b_saad, omair, swamy}@ece.concordia.ca

ABSTRACT

In this paper, a radix-2/16 decimation-in-frequency (DIF) fast Fourier transform (FFT) algorithm and its higher radix version, namely radix-4/16 DIF FFT algorithm, are proposed by suitably mixing the radix-2, radix-4 and radix-16 index maps, and combing some of the twiddle factors. It is shown that the proposed algorithms and the existing radix-2/4 and radix-2/8 FFT algorithms require exactly the same number of arithmetic operations (multiplications+additions). Moreover, by using techniques similar to those introduced in this paper, it can be shown that all the possible split-radix FFT algorithms of the type radix-$2^r/2^{rs}$ for computing a 2^m-point DFT require exactly the same number of arithmetic operations.

I. INTRODUCTION

The discrete Fourier transform (DFT) plays an important role in digital signal processing applications. One of the most interesting approaches for designing fast Fourier transform (FFT) algorithms is that of the split-radix introduced by Duhamel and Hollmann in [1], since it leads to algorithms having a good compromise between the arithmetic and structural complexities. In [1], the authors have claimed that the computation of the odd terms of the split-radix DFT through a radix-8 does not improve the algorithm. In [2], the author has stated that "it can easily be checked out that a 2/8-split-radix algorithm is worse than a 2/4-split-radix algorithm from an arithmetic complexity point of view". In [3], the authors have claimed that the radix-2/4 FFT algorithm for computing a 2^m-point DFT is the best among a general class of possible split-radix algorithms from the point of view of arithmetic complexity. In 2001, Takahashi [4] directly used the radix-8 in the computation of the odd terms of the split-radix FFT. This, of course, led to an algorithm with an increased arithmetic complexity. However, efficient radix-2/8 FFT algorithms have been recently developed [5], [6] that require exactly the same number of arithmetic operations as in the case of the radix-2/4 FFT algorithm.

In this paper, we design two split-radix FFT algorithms, radix-2/16 and radix-4/16 decimation-in-frequency (DIF) FFT algorithms, using radix-2, radix-4 and radix-16 index maps. By suitably mixing these index maps and combing some of the twiddle factors, the proposed algorithms are shown to require exactly the same number of arithmetic operations as in the existing radix-2/4 and radix-2/8 FFT algorithms.

II. PROPOSED RADIX-2/16 FFT ALGORITHM

The DFT of length N is defined by

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

where $W_N = \exp(-j2\pi/N)$ and $j = \sqrt{-1}$. The first stage of the decomposition in the radix-2/4 DIF FFT algorithm proposed in [1] consists of decomposing the N-point DFT given by (1) into one $N/2$-point DFT given by

$$X(2n) = \sum_{k=0}^{N/2-1} y_0(k) W_{N/2}^{nk}, \ 0 \le n \le \frac{N}{2}-1 \tag{2}$$

and two $N/4$-point DFTs given by

$$X(4n+q) = \sum_{k=0}^{N/4-1} y_q(k) W_N^{qk} W_{N/4}^{nk}, \ 0 \le n \le \frac{N}{4}-1, \ q=1,3 \tag{3}$$

where

$$y_0(k) = x(k) + x\left(k+\frac{N}{2}\right), \tag{4}$$

$$y_q(k) = \left(x(k) - x\left(k+\frac{N}{2}\right)\right) + (-j)^q \left(x\left(k+\frac{N}{4}\right) - x\left(k+3\frac{N}{4}\right)\right) \tag{5}$$

In order to develop a radix-2/16 DIF FFT algorithm, we further decompose the odd-indexed terms given by (3) using the radix-4 index maps. This enables us to express the odd-indexed terms by

$$X(16n+\alpha)=\sum_{k=0}^{N/16-1} g_\alpha(k) W_{N/16}^{nk},\ 0\le n\le \frac{N}{16}-1,$$
$$\alpha=1,5,9,13,3,7,11,15 \quad (6)$$

The input sequences of the eight $N/16$-point DFTs given by (6) can be expressed as

$$\begin{bmatrix} g_q(k) \\ g_{q+4}(k) \\ g_{q+8}(k) \\ g_{q+12}(k) \end{bmatrix} = \mathbf{T}_{k,q} \begin{bmatrix} y_q(k) \\ y_q\left(k+\frac{N}{16}\right) \\ y_q\left(k+\frac{N}{8}\right) \\ y_q\left(k+3\frac{N}{16}\right) \end{bmatrix},$$
$$0\le k\le \frac{N}{16}-1,\ q=1,3 \quad (7)$$

where

$$\mathbf{T}_{k,q}=\begin{bmatrix} W_N^{qk} & 0 & 0 & 0 \\ 0 & W_N^{(q+4)k} & 0 & 0 \\ 0 & 0 & W_N^{(q+8)k} & 0 \\ 0 & 0 & 0 & W_N^{(q+12)k} \end{bmatrix} \mathbf{W}_4 \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & W_N^{qN/16} & 0 & 0 \\ 0 & 0 & W_N^{qN/8} & 0 \\ 0 & 0 & 0 & W_N^{q3N/16} \end{bmatrix} \quad (8)$$

and the matrix $\mathbf{W}_4$ is the operator of the 4-point DFT.

Finally, the proposed radix-2/16 DIF FFT algorithm consists of decomposing the N-point DFT given by (1) into one $N/2$-point DFT given by (2) and eight $N/16$-point DFTs given by (6). This process is repeated successively for each of the new resulting DFTs until some 8-, 4-, or 2-point DFTs need to be computed. The flowgraph of the general butterfly of the proposed radix-2/16 DIF FFT algorithm can easily be obtained using (4), (5), and (7). Due to lack of space, the flowgraph of the butterfly is not given in this paper. For given values of k, $k\neq 0$, and q, the twiddle factor matrix given by (8) introduces six complex multiplications and a multiplication by the twiddle factor $W_N^{qN/8}$ in the computation of (7). However, for $k=0$, the number of complex multiplications reduces to two. Thus, the butterfly corresponding to $k=0$ can be considered as a special butterfly. In order to further reduce the number of operations, we introduce a new special butterfly for $k=\frac{N}{32}$ by using the fact that $W_N^{\alpha k}=W_N^{(\alpha-q)k}W_N^{qk}$ and combining some of the twiddle factors in (8). Then, for $k=\frac{N}{32}$, (8) can be expressed as

$$\mathbf{T}_{k,q}=\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & W_N^{4k} & 0 & 0 \\ 0 & 0 & W_N^{8k} & 0 \\ 0 & 0 & 0 & W_N^{12k} \end{bmatrix} \mathbf{W}_4 \cdot \begin{bmatrix} W_N^{qk} & 0 & 0 & 0 \\ 0 & W_N^{qk}W_N^{qN/16} & 0 & 0 \\ 0 & 0 & W_N^{qk}W_N^{qN/8} & 0 \\ 0 & 0 & 0 & W_N^{qk}W_N^{q3N/16} \end{bmatrix},$$
$$k=\frac{N}{32},\ q=1,3 \quad (9)$$

Now, it is clear that for a given value of q, (9) introduces only four complex multiplications and multiplications by $W_N^{N/8}$ and $W_N^{3N/8}$ in the computation of (7).

III. PROPOSED RADIX-4/16 FFT ALGORITHM

The length N is assumed to be an integral power of four. Let us first start by carrying out a DIF decomposition of (1) using the radix-4 index maps. This provides in the first stage four $N/4$-point DFTs given by

$$X(4n+q)=\sum_{k=0}^{N/4-1} \widetilde{y}_q(k) W_N^{qk} W_{N/4}^{nk},\ 0\le n\le \frac{N}{4}-1,$$
$$q=0,1,2,3 \quad (10)$$

where the sequences $y_q(k)$, for $q=0$, 1, 2 and 3, are given by

$$\begin{bmatrix} \widetilde{y}_0(k) \\ \widetilde{y}_1(k) \\ \widetilde{y}_2(k) \\ \widetilde{y}_3(k) \end{bmatrix} = \mathbf{W}_4 \begin{bmatrix} x(k) \\ x(k+N/4) \\ x(k+N/2) \\ x(k+3N/4) \end{bmatrix} \quad (11)$$

The decomposition of (1) into the DFTs given by (10) is recognized as the first stage of the well-known radix-4 DIF FFT algorithm. In order to develop a radix-4/16 DIF FFT algorithm, we further decompose the DFTs corresponding to $q=1$, 2 and 3 in (10) using the radix-4 index maps. This enables us to write

$$X(16n+\alpha)=\sum_{k=0}^{N/16-1} \widetilde{g}_\alpha(k) W_{N/16}^{nk},\ 0\le n\le \frac{N}{16}-1,$$
$$\alpha=1,5,9,13,2,6,10,14,3,7,11,15 \quad (12)$$

The input sequences of the twelve $N/16$-point DFTs given by (12) can be expressed as

$$\begin{bmatrix} \widetilde{g}_q(k) \\ \widetilde{g}_{q+4}(k) \\ \widetilde{g}_{q+8}(k) \\ \widetilde{g}_{q+12}(k) \end{bmatrix} = \mathbf{T}_{k,q} \begin{bmatrix} \widetilde{y}_q(k) \\ \widetilde{y}_q\left(k+\frac{N}{16}\right) \\ \widetilde{y}_q\left(k+\frac{N}{8}\right) \\ \widetilde{y}_q\left(k+3\frac{N}{16}\right) \end{bmatrix},$$
$$0\le k\le \frac{N}{16}-1,\ q=1,2,3 \quad (13)$$

Finally, the proposed radix-4/16 DIF FFT algorithm consists of decomposing the N-point DFT given by (1) into one $N/4$-point DFT corresponding to (10) for $q=0$ and twelve $N/16$-point DFTs given by (12). This decomposition process is repeated successively for each of the new resulting DFTs until only 4-point DFTs need to be computed. The flowgraph of the general butterfly of the proposed radix-4/16 DIF FFT algorithm can easily be obtained using (11) and (13). Again, due to lack of space, the flowgraph of the butterfly is not given. For a given value of k, $k \neq 0$, (13) requires six complex multiplications and a multiplication by $W_N^{N/8}$ or $W_N^{3N/8}$ for $q=1$ or 3, whereas only four complex multiplications along with multiplications by $W_N^{N/8}$ and $W_N^{3N/8}$ are required for $q=2$. However, for $k=0$, the number of complex multiplications reduces to two for $q=1$ or 3, whereas no complex multiplications is required for $q=2$. Thus, the butterfly corresponding to $k=0$ can be considered as a special butterfly. In order to further reduce the number of operations, we introduce a new special butterfly for $k=\frac{N}{32}$ by using a technique similar to that presented in Section II. Then, for $q=1$ and 3, (13) can be computed efficiently using the matrix $\mathbf{T}_{\frac{N}{32},q}$ given by (9). However, for $q=2$, we rearrange the matrix given by (8) as

$$\mathbf{T}_{k,2} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & W_N^{8k} & 0 \\ 0 & 0 & 0 & W_N^{8k} \end{bmatrix} \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 \end{bmatrix} \cdot \\ \cdot \begin{bmatrix} W_N^{2k} & 0 & 0 & 0 \\ 0 & W_N^{2k}W_N^{N/8} & 0 & 0 \\ 0 & 0 & W_N^{6k} & 0 \\ 0 & 0 & 0 & -jW_N^{6k}W_N^{N/8} \end{bmatrix} \cdot \\ \cdot \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & W_N^{N/4} & 0 \\ 0 & 0 & 0 & W_N^{N/4} \end{bmatrix}, \\ k = \frac{N}{32} \quad (14)$$

Now, it is clear that for $k=\frac{N}{32}$ and $q=1$ or 3, (13) requires only four complex multiplications along with multiplications by $W_N^{N/8}$ and $W_N^{3N/8}$. For $q=2$, (13) requires only four complex multiplications.

IV. ARITHMETIC COMPLEXITIES OF THE PROPOSED ALGORITHMS

In the proposed radix-2/16 FFT algorithm, the first stage of the decomposition is carried out, as discussed in Section II, by repeating $\left(\frac{N}{16}-2\right)$ times the general butterfly and the two special butterflies corresponding to $k=0$ and $k=N/32$. It can be shown that the expressions for the numbers of real multiplications and real additions required by the proposed radix-2/16 algorithm for the computation of a length-N DFT are

$$M_N^{42} = 52\frac{N}{16} - 44 + M_{N/2}^{42} + 8M_{N/16}^{42}, \\ M_{16}^{42} = 24,\ M_8^{42} = 4,\ M_4^{42} = M_2^{42} = 0 \quad (15)$$

$$A_N^{42} = 60\frac{N}{16} - 20 + 3N + A_{N/2}^{42} + 8A_{N/16}^{42}, \\ A_{16}^{42} = 144,\ A_8^{42} = 52,\ A_4^{42} = 16,\ A_2^{42} = 4 \quad (16)$$

if a complex multiplication is performed using four real multiplications and two real additions (4mult-2add scheme). The corresponding expressions are

$$M_N^{33} = 40\frac{N}{16} - 32 + M_{N/2}^{33} + 8M_{N/16}^{33}, \\ M_{16}^{33} = 20,\ M_8^{33} = 4,\ M_4^{33} = M_2^{33} = 0 \quad (17)$$

$$A_N^{33} = 72\frac{N}{16} - 32 + 3N + A_{N/2}^{33} + 8A_{N/16}^{33}, \\ A_{16}^{33} = 148,\ A_8^{33} = 52,\ A_4^{33} = 16,\ A_2^{33} = 4 \quad (18)$$

if a complex multiplication is performed using three real multiplications and three real additions (3mult-3add scheme).

In the first stage of the proposed radix-4/16 FFT algorithm, the decomposition of the N-point DFT is achieved, as discussed in Section III, by repeating $\left(\frac{N}{16}-2\right)$ times the general butterfly and the two special butterflies corresponding to $k=0$ and $k=N/32$. Again, it can be shown that the expressions for the numbers of real multiplications and real additions required by the proposed radix-4/16 algorithm are

$$M_N^{42} = 72\frac{N}{16} - 64 + M_{N/4}^{42} + 12M_{N/16}^{42}, \\ M_{16}^{42} = 24,\ M_4^{42} = 0 \quad (19)$$

$$A_N^{42} = 88\frac{N}{16} - 32 + 4N + A_{N/4}^{42} + 12A_{N/16}^{42}, \\ A_{16}^{42} = 144,\ A_4^{42} = 16 \quad (20)$$

if the 4mult-2add scheme is considered. The corresponding expressions are

$$M_N^{33} = 56\frac{N}{16} - 48 + M_{N/4}^{33} + 12M_{N/16}^{33}, \\ M_{16}^{33} = 20,\ M_4^{33} = 0 \quad (21)$$

$$A_N^{33} = 104\frac{N}{16} - 48 + 4N + A_{N/4}^{33} + 12A_{N/16}^{33}, \\ A_{16}^{33} = 148,\ A_4^{33} = 16 \quad (22)$$

if the 3mult-3add scheme is considered,.

The arithmetic complexities of the proposed radix-2/16 and radix-4/16 FFT algorithms are compared to those of the existing radix-2/4 [1], [7] and radix-2/8 [5], [6] FFT algorithms in Tables I, II and III. It is clear from these tables that the four algorithms require exactly the same number of arithmetic operations (multiplications+additions) irrespectively of whether the 4mult-2add or 3mult-3add scheme is used. If the 3mult-3add scheme is used, the four algorithms have, in addition, the same number of multiplications. However, in the case of the 4mult-2add scheme, the radix-2/8 algorithm requires the lowest number of multiplications. It should be pointed out that the radix-2/4 algorithm requires one general butterfly and two special butterflies corresponding to $k = 0$ and $k = N/8$. The radix-2/8 algorithm also requires one general butterfly and two special butterflies corresponding to $k = 0$ and $k = N/16$. By defining a special butterfly for $k = N/64$ and using a technique similar to that introduced in this paper for combining the twiddle factors, it can be shown that the number of arithmetic operations required by the radix-2/32 FFT algorithm is identical to that required by any other split-radix FFT. Similar results can be established for all the possible split-radix FFT algorithms of the type radix-$2^r/2^{rs}$.

V. CONCLUSION

In this paper, we have shown that by using a mixture of radix-2, radix-4 and radix-16 index maps in the decomposition of the DFT, and suitably combining some of the twiddle factors, the resulting radix-2/16 and radix-4/16 FFT algorithms require exactly the same number of arithmetic operations as in the existing radix-2/4 or radix-2/8 FFT algorithms. Finally, using techniques similar to those introduced in this paper, it can be shown that all the possible split-radix FFT algorithms of the type radix-$2^r/2^{rs}$ for computing a 2^m-point DFT require exactly the same number of arithmetic operations.

REFERENCES

[1] P. Duhamel and H. Hollmann, "Split radix FFT algorithm," *Electron. Lett.*, vol. 20, pp. 14-16, Jan. 1984.

[2] P. Duhamel, "Implementation of 'split-radix' FFT algorithms for complex , real, and real-symmetric data," *IEEE Trans. Acoust., Speech, Signal Processing,* vol. 34, pp. 285-295, April. 1986.

[3] M. Vetterli and P. Duhamel, "Split-radix algorithms for length-p^m DFT's," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 37, pp. 57-64, Jan. 1989.

[4] D. Takahashi, "An extended split-radix FFT algorithm," *IEEE Signal Processing Letters,* vol. 8, pp. 145-147, May 2001.

[5] S. Bouguezel, M.O. Ahmad and M.N.S. Swamy, "An efficient split-radix FFT algorithm," in *Proc. IEEE Int. Symp. Circuits and Systems*, vol. 4, May 2003, pp. 65-68.

[6] S. Bouguezel, M.O. Ahmad and M.N.S. Swamy, "A new radix-2/8 FFT algorithm for length-$q \times 2^m$ DFTs," *IEEE Trans. Circuits and Syst. I,* vol. 51, pp. 1723-1732, Sept. 2004.

[7] G. Bi and Y. Q. Chen, "Fast DFT algorithms for length $N = q * 2^m$," *IEEE Trans. Circuits and Syst. II*: *Analog and Digital Signal Processing*, vol. 45, pp. 685-690, June 1998.

TABLE I
NUMBER OF MULTIPLICATIONS USING
THE 4MULT-2ADD SCHEME

N	Radix-2/4	Radix-2/8	Radix-2/16	Radix-4/16
16	24	24	24	24
32	84	84	84	
64	248	240	248	248
128	660	636	652	
256	1656	1592	1632	1624
512	3988	3812	3924	
1024	9336	8896	9192	9144
2048	21396	20364	21020	
4096	48248	45832	47344	47000

TABLE II
NUMBER OF ADDITIONS USING THE 4MULT-2ADD SCHEME

N	Radix-2/4	Radix-2/8	Radix-2/16	Radix-4/16
16	144	144	144	144
32	372	372	372	
64	912	920	912	912
128	2164	2188	2172	
256	5008	5072	5032	5040
512	11380	11556	11444	
1024	25488	25928	25632	25680
2048	56436	57468	56812	
4096	123792	126208	124696	125040

TABLE III
ARITHMETIC COMPLEXITIES USING
THE 3MULT-3ADD SCHEME

	Radix-2/4 FFT, Radix-2/8 FFT and Radix-2/16 FFT		Radix-4/16 FFT	
N	Mults.	Adds.	Mults.	Adds.
16	20	148	20	148
32	68	388		
64	196	964	196	964
128	516	2308		
256	1284	5380	1284	5380
512	3076	12292		
1024	7172	27652	7172	27652
2048	16388	61444		
4096	36868	135172	36868	135172