An Extended Split-Radix FFT Algorithm

Daisuke Takahashi

***Abstract*—An extended split-radix fast Fourier transform (FFT) algorithm is proposed. The extended split-radix FFT algorithm has the same asymptotic arithmetic complexity as the conventional split-radix FFT algorithm. Moreover, this algorithm has the advantage of fewer loads and stores than either the conventional split-radix FFT algorithm or the radix-4 FFT algorithm.**

I. Introduction

THE split-radix fast Fourier transform FFT algorithm [1] is known as the FFT algorithm that uses the lowest total number of operations. The split-radix idea can be extended to other radix pairs. Vetterli and Duhamel [2] have shown that the radix-p/p^2 split-radix approach is generalized to length-p^m DFTs. However, an implementation of a radix-2/8 algorithm for length-2^m DFTs has not yet been presented.

This paper is concerned with the implementation and evaluation of a radix-2/8 split-radix FFT algorithm. We will show here that the radix-2/8 split-radix FFT algorithm has the same asymptotic arithmetic complexity as the conventional split-radix FFT algorithm, and this algorithm has the advantage of fewer loads and stores than the conventional split-radix FFT algorithm.

II. An Extended Split-Radix FFT Algorithm

The basic idea of an extended split-radix FFT algorithm is the application of a radix-2 index map to the even-indexed terms and a radix-8 [3] index map to the odd-indexed terms. That is, the extended split-radix FFT algorithm is based on the synthesis of one half-length and four eighth-length DFTs.

The DFT of N points is given by

$$X_k = \sum_{n=0}^{N-1} x_n W_N^{nk}, \quad k = 0, \ldots, N-1 \tag{1}$$

where $W_N = \exp(-j2\pi/N)$, X_k and x_n are sequences of complex numbers.

If $N = 2^m$, then

$$X_{2k} = \sum_{n=0}^{N/2-1} [x_n + x_{n+N/2}] W_N^{2nk} \tag{2}$$

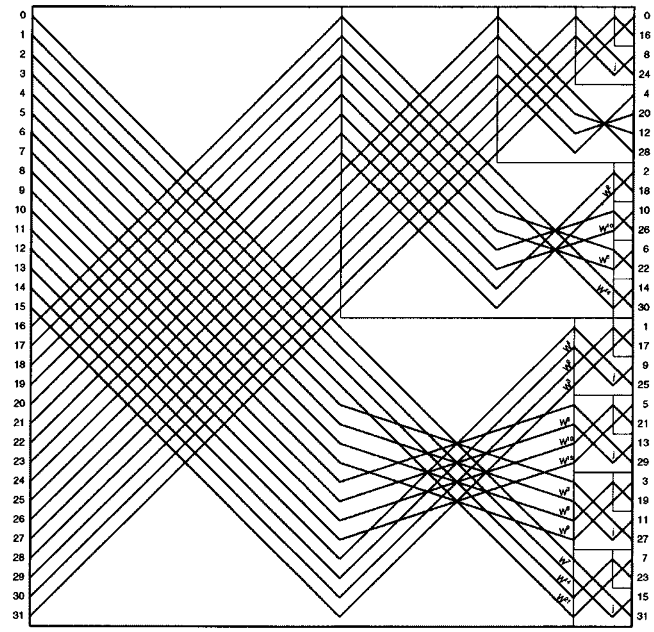

Fig. 1. Graph of a DIF extended split-radix transform structure ($N = 32$).

for the even index terms, and

$$\begin{aligned} X_{8k+1} &= \sum_{n=0}^{N/8-1} \Big[\{(x_n - x_{n+N/2}) - j(x_{n+N/4} - x_{n+3N/4})\} \\ &\quad + \frac{1}{\sqrt{2}}\{(1-j)(x_{n+N/8} - x_{n+5N/8}) \\ &\quad - (1+j)(x_{n+3N/8} - x_{n+7N/8})\} \Big] W_N^n W_N^{8nk} \end{aligned} \tag{3}$$

$$\begin{aligned} X_{8k+3} &= \sum_{n=0}^{N/8-1} \Big[\{(x_n - x_{n+N/2}) + j(x_{n+N/4} - x_{n+3N/4})\} \\ &\quad - \frac{1}{\sqrt{2}}\{(1+j)(x_{n+N/8} - x_{n+5N/8}) \\ &\quad - (1-j)(x_{n+3N/8} - x_{n+7N/8})\} \Big] W_N^{3n} W_N^{8nk} \end{aligned} \tag{4}$$

$$\begin{aligned} X_{8k+5} &= \sum_{n=0}^{N/8-1} \Big[\{(x_n - x_{n+N/2}) - j(x_{n+N/4} - x_{n+3N/4})\} \\ &\quad - \frac{1}{\sqrt{2}}\{(1-j)(x_{n+N/8} - x_{n+5N/8}) \\ &\quad - (1+j)(x_{n+3N/8} - x_{n+7N/8})\} \Big] W_N^{5n} W_N^{8nk} \end{aligned} \tag{5}$$

Manuscript received November 22, 2000. This work was supported by the Grant-in-Aid for Encouragement of Young Scientists Contract 12780190, Ministry of Education, Science, Sports and Culture of Japan. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Y. Hua.

The author is with the Department of Information and Computer Sciences, Saitama University, Saitama 338-8570, Japan (e-mail: daisuke@ics.saitama-u.ac.jp).

Publisher Item Identifier S 1070-9908(01)02789-4.

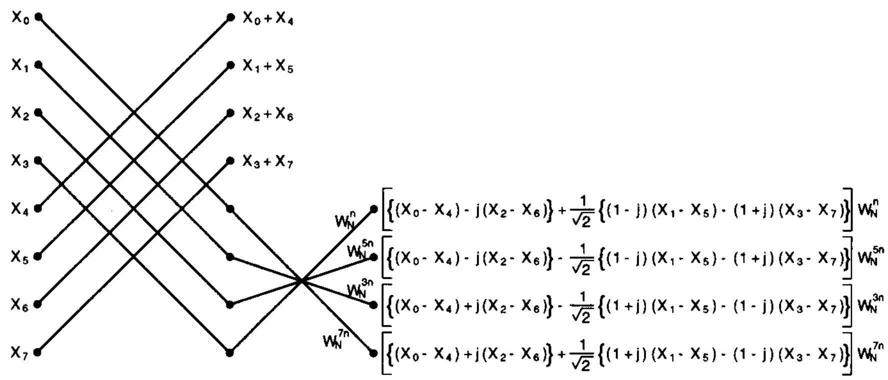

Fig. 2. Butterfly used in the graph of a DIF extended split-radix FFT.

TABLE I
NUMBER OF NONTRIVIAL REAL MULTIPLICATIONS AND ADDITIONS FOR FFT ALGORITHMS USING FOUR REAL MULTIPLICATIONS AND TWO REAL ADDITIONS PER COMPLEX MULTIPLICATION FOR N COMPLEX DATA POINTS

N	radix-2		radix-4		radix-8		Conventional split-radix [4]		Extended split-radix	
	Mults	Adds	Mults	Adds	Mults	Adds	Mults	Adds	Mults	Adds
8	4	52			4	52	4	52	4	52
16	28	148	24	144			24	144	28	148
32	108	388					84	372	92	380
64	332	964	264	920	248	928	248	912	252	932
128	908	2308					660	2164	668	2220
256	2316	5380	1800	5080			1656	5008	1660	5140
512	5644	12292			3992	11632	3988	11380	3932	11676
1024	13324	27652	10248	25944			9336	25488	9148	26180
2048	30732	61444					21396	56436	20892	57996
4096	69644	135172	53256	126296	48280	126832	48248	123792	46844	127220
8192	155660	294916					107412	269428	103900	276988
16384	344076	638980	262152	595288			236664	582544	228412	599076

$$\begin{aligned} X_{8k+7} &= \sum_{n=0}^{N/8-1} \Bigg[\{(x_n - x_{n+N/2}) + j(x_{n+N/4} - x_{n+3N/4})\} \\ &\quad + \frac{1}{\sqrt{2}}\{(1+j)(x_{n+N/8} - x_{n+5N/8}) \\ &\quad - (1-j)(x_{n+3N/8} - x_{n+7N/8})\} \Bigg] W_N^{7n} W_N^{8nk} \qquad (6) \end{aligned}$$

for the odd index terms.

The first stage of an extended split-radix decimation-in-frequency decomposition then replaces a DFT of length N by one DFT of length $N/2$ and four DFTs of length $N/8$. The length-N DFT is then obtained by successive use of such decompositions up to the last two stages, where some conventional split-radix butterflies (without twiddle factors) are needed, and to the last stage, where some usual radix-2 butterflies (without twiddle factors) are needed.

A graph of a decimation-in-frequency (DIF) extended split-radix transform structure is shown in Fig. 1, and the general elementary butterfly used in the diagram is illustrated in detail in Fig. 2.

III. EVALUATION

Let $M(N)$ [respectively, $A(N)$] be the number of real multiplications (respectively, additions) needed to perform a DFT of length N with the extended split-radix FFT algorithm.

It requires four real multiplications, 36 real additions, and four complex multiplications to evaluate (2)–(6).

We obtain

$$\begin{aligned} M(N) &= M(N/2) + 4M(N/8) + (5/2)N - 16, \\ A(N) &= A(N/2) + 4A(N/8) + (11/2)N - 8 \end{aligned}$$

and, with the initial conditions $M(2) = 0$, $M(4) = 0$, $M(8) = 4$, and $A(2) = 4$, $A(4) = 16$, $A(8) = 52$. It is assumed that the complex multiplications are done using four real multiplications and two real additions.

The following formulas can be used to generate counts for longer lengths that use the four-multiply two-add scheme:

$$\begin{aligned} M(N) &= (5/4)N\log_2 N - (57/16)N + 4 \\ &\quad - (7/(32N))(\alpha^{\log_2 N} + \beta^{\log_2 N}) \\ &\quad - (j13\sqrt{7}/(224N))(\alpha^{\log_2 N} - \beta^{\log_2 N}), \\ A(N) &= (11/4)N\log_2 N - (31/16)N + 2 \\ &\quad - (1/(32N))(\alpha^{\log_2 N} + \beta^{\log_2 N}) \\ &\quad - (j11\sqrt{7}/(224N))(\alpha^{\log_2 N} - \beta^{\log_2 N}) \end{aligned}$$

where $\alpha = -1 + j\sqrt{7}$ and $\beta = -1 - j\sqrt{7}$.

On the other hand, the following give the same counts for the three-multiply three-add scheme:

$$\begin{aligned} M(N) &= N\log_2 N - (11/4)N + 3 \\ &\quad - (1/(8N))(\alpha^{\log_2 N} + \beta^{\log_2 N}) \\ &\quad - (j3\sqrt{7}/(56N))(\alpha^{\log_2 N} - \beta^{\log_2 N}), \\ A(N) &= 3N\log_2 N - (11/4)N + 3 \\ &\quad - (1/(8N))(\alpha^{\log_2 N} + \beta^{\log_2 N}) \\ &\quad - (j3\sqrt{7}/(56N))(\alpha^{\log_2 N} - \beta^{\log_2 N}) \end{aligned}$$

where $\alpha = -1 + j\sqrt{7}$ and $\beta = -1 - j\sqrt{7}$.

The number of nontrivial real multiplications and additions used for radix-2, radix-4, radix-8, conventional split-radix, and extended split-radix FFT algorithms using four-multiply two-add scheme for N complex data points are shown in Table I.

The number of loads, stores, multiplications, and additions used for radix-2, radix-4, radix-8, conventional split-radix, and extended split-radix butterflies using four-multiply two-add scheme is given in Table II. In calculating the number of loads and the number of stores, we assume that enough registers are available to perform an entire butterfly in the registers without using any intermediate stores or loads.

In Table III, the asymptotic number of loads, stores, multiplications, and additions used by each algorithm is given. The extended split-radix FFT algorithm requires fewer loads and stores than the radix-4 FFT algorithm or the conventional split-radix FFT algorithm. In particular, in comparison with the conventional split-radix FFT algorithm, the extended split-radix FFT algorithm asymptotically saves 25% of the loads and stores.

TABLE II
NUMBER OF LOADS, STORES, MULTIPLICATIONS, AND ADDITIONS FOR GENERAL BUTTERFLIES USED IN FFT ALGORITHMS. THE NUMBER OF LOADS INCLUDES LOADING ALL OF THE CONSTANTS

Algorithm	Loads	Stores	Mults	Adds
Radix-2	6	4	4	6
Radix-4	14	8	12	22
Radix-8	30	16	32	66
Conventional split-radix	12	8	8	16
Extended split-radix	24	16	20	44

TABLE III
NUMBER OF LOADS, STORES, MULTIPLICATIONS, AND ADDITIONS DIVIDED BY $N\log_2 N$ USED BY FFT ALGORITHMS TO COMPUTE AN N POINT DFT. LOWER ORDER TERMS HAVE BEEN OMITTED

Algorithm	Loads	Stores	Mults	Adds
Radix-2	3	2	2	3
Radix-4	7/4	1	3/2	11/4
Radix-8	5/4	2/3	4/3	11/4
Conventional split-radix	2	4/3	4/3	8/3
Extended split-radix	3/2	1	5/4	11/4

IV. CONCLUSION

An extended split-radix FFT algorithm has been presented. The algorithm has the same asymptotic arithmetic complexity as the conventional split-radix FFT algorithm. If the FFTs are being computed on a machine that has enough registers to perform an entire extended split-radix FFT algorithm, those FFTs will use fewer loads and stores than the conventional split-radix FFT algorithm or the radix-4 FFT algorithm. It may therefore be preferable to use the extended split-radix FFT algorithm rather than the conventional split-radix FFT algorithm.

REFERENCES

[1] P. Duhamel and H. Hollmann, “Split-radix FFT algorithm,” *Electron. Lett.*, vol. 20, pp. 14–16, Jan. 1984.
[2] 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.
[3] G. D. Bergland, “A fast Fourier transform algorithm using base 8 iterations,” *Math. Comput.*, vol. 22, pp. 275–279, Apr. 1968.
[4] H. V. Sorensen, M. T. Heideman, and C. S. Burrus, “On computing the split-radix FFT,” *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-34, pp. 152–156, Feb. 1986.