

A Fast Algorithm Based on SRFFT for Length $N = q \times 2^m$ DFTs

Weihua Zheng, Kenli Li, and Keqin Li, *Senior Member, IEEE*

Abstract—In this brief, we present a fast algorithm for computing length- $q \times 2^m$ discrete Fourier transforms (DFT). The algorithm divides a DFT of size- $N = q \times 2^m$ decimation in frequency into one length- $N/2$ DFT and two length- $N/4$ DFTs. The length- $N/2$ sub-DFT is recursively decomposed decimation in frequency, and the two size- $N/4$ sub-DFTs are transformed into two dimension and the terms with the same rotating factor are arranged in a column. Thus, the scaled DFTs (SDFTs) are obtained, simplifying the real multiplications of the proposed algorithm. A further improvement can be achieved by the application of radix-2/8, modified split-radix FFT (MSRFFT), and Wang's algorithm for computing its length- 2^m and length- q sub-DFTs. Compared with the related algorithms, a substantial reduction of arithmetic complexity and more accurate precision are obtained.

Index Terms—Fast Fourier transform (FFT), modified split-radix FFT (MSRFFT), radix 2/8 FFT algorithm, scaled discrete Fourier transform (SDFT), split-radix FFT (SRFFT).

I. INTRODUCTION

THE MOTIVATION of the brief is to develop an algorithm to improve two significant performances of the FFT algorithms in [2]–[4], computational efficiency and computational accuracy, for computing length- $N = q \times 2^m$ DFTs, where q is an odd integer. The complexity issue has been well studied [5], especially for the algorithms proposed for computing length- 2^m DFTs [6]. By decomposing only, it is difficult to further improve arithmetic complexity. The development of FFTs requires another effective technique, such as the scaled DFT (SDFT) technique. In addition to reducing computational complexity, the SDFT technique can improve computational precision.

The main contribution of the brief is to obtain a reduction of arithmetic complexity and more accurate precision, which is achieved by the following techniques. For a length- $N = q \times 2^m$ DFT, this brief divides its decimation in frequency into one length- $N/2$ sub-DFT and two length- $N/4$ sub-DFTs. Two length- $N/4$ sub-DFTs are then expressed in a two-dimension form. Terms with rotating factors $w_{N/q}^{n_0}$,

$w_{N/q}^{n_0+N/4q}, \dots, w_{N/q}^{n_0+(q-1)N/4q}$ are arranged into a column, where $n_0 \in (0, N/4q)$. From each of the $N/(4q)$ columns, the common factor $w_{N/q}^{n_0}$ is extracted as a scaling factor of a length- q sub-DFT. In this way, $N/(4q)$ length- q scaled sub-DFTs are obtained. Wang's algorithm [1] is used to evaluate these length- q scaled/unscaled sub-DFTs. For length- $N/(4q)$ sub-DFTs, the proposed algorithm can directly utilize other effective algorithms, such as the radix 2/8 FFT algorithm, the modified SRFFT (MSRFFT) algorithm, and the algorithms in [7], achieving a further reduction of operations count or a reduction of the accesses to a lookup table and data transfers.

The rest of the brief is organized as follows. Section II presents a derivation of the proposed algorithm. Section III discusses the computational complexity and compares the proposed algorithm with other algorithms. Section IV gives a comparison of the L_2 relative error and a simple discussion on the roundoff of SDFT and unscaled DFT. Section V concludes the brief.

II. PROPOSED ALGORITHM

Given a length- N sequence $x(n)$, its DFT $X(k)$ is also a length- N sequence defined by

$$X(k) = \sum_{n=0}^{N-1} x(n)w_N^{nk}, \quad 0 \leq k < N \quad (1)$$

where $w_N = e^{-j2\pi/N}$ and $j = \sqrt{-1}$. In the proposed algorithm, the sequence length N is assumed to be $q \times 2^m$, where q is an odd integer greater than 1 and $m \geq 0$.

A. DIF Decomposition

For (1), when $m = 0$, the DFT is a length- q DFT and can be directly computed with a custom-built algorithm. When $m = 1$ or $N = 2q$, the decomposition of (1) into even-indexed and odd-indexed outputs provides

$$X(2k) = \sum_{n=0}^{N/2-1} u(n)w_{N/2}^{nk}, \quad 0 \leq k < N/2, \quad (2)$$

$$X((2k+q) \bmod N) = \sum_{n=0}^{N/2-1} (-1)^n v(n)w_{N/2}^{nk}, \quad 0 \leq k < N/2. \quad (3)$$

The decomposition consists of dividing the DFT into two q -points DFTs given by (2) and (3). The sequences $u(n)$ in (2)

Manuscript received September 3, 2013; accepted November 9, 2013. Date of publication January 9, 2014; date of current version February 21, 2014. This work was supported by the National Science Foundation for Distinguished Young Scholars of Hunan under project 12JJ1011 and was partially funded by National Natural Science Foundation of China under Grants 61133005, 61070057, 61370098, and 61370095. This brief was recommended by Associate Editor B. D. Liu. (*Corresponding author: Kenli Li.*)

W. Zheng and K. Li are with the College of Information Science and Engineering, Hunan University, Changsha 410082, China (e-mail: zhengdavid@hnu.edu.cn; lkl@hnu.edu.cn).

K. Li is with the College of Information Science and Engineering, Hunan University, Changsha 410082, China, and also with the Department of Computer Science, State University of New York, New Paltz, NY 12561-2443 USA (e-mail: lik@newpaltz.edu).

Digital Object Identifier 10.1109/TCSIL.2013.2291098

and $v(n)$ in (3) can be expressed in a matrix form as

$$\begin{bmatrix} v(n) \\ v(n) \end{bmatrix} = H_2 \begin{bmatrix} x(n) \\ x(N/2 + n) \end{bmatrix} \quad (4)$$

where

$$H_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}.$$

In a similar way, when $m > 1$ or $N > 2q$, the decomposition of the DFT in (1) provides

$$X(2k) = \sum_{n=0}^{N/2-1} u(n)w_{N/2}^{nk}, \quad 0 \leq k < N/2 \quad (5)$$

for the even-indexed terms and

$$\begin{aligned} &X((4k+q) \bmod N) \\ &= \sum_{n=0}^{N/4-1} a_e(n)w_{N/q}^n w_{N/4}^{nk}, \quad 0 \leq k < N/4 \end{aligned} \quad (6)$$

$$\begin{aligned} &X((4k+3q) \bmod N) \\ &= \sum_{n=0}^{N/4-1} a_o(n)w_{N/q}^{3n} w_{N/4}^{nk}, \quad 0 \leq k < N/4 \end{aligned} \quad (7)$$

for the odd-indexed terms, where

$$\begin{aligned} \begin{bmatrix} a_e(n) \\ a_o(n) \end{bmatrix} &= H_2 J \begin{bmatrix} v(n) \\ v(N/4 + n) \end{bmatrix}, \\ J &= \begin{bmatrix} 1 & 0 \\ 0 & (-1)^{(q+1)/2} j \end{bmatrix}. \end{aligned} \quad (8)$$

Equations (6) and (7) contain two sub-DFTs of size- $N/4$ whose input terms are rotated by $w_{N/q}^n$ and $w_{N/4}^{3n}$. The length- $N/2$ DFT will be recursively divided with the DIF algorithm described in this subsection and the two length- $N/4$ sub-DFTs will be evaluated with a two-dimension fast algorithm described in the next subsection.

B. Two-Dimension Transform

The straightforward calculations of (6) and (7) cannot obtain any scaled sub-SDFT. It is desirable that the DFTs in (6) and (7) are expressed in a two-dimension form and the input terms of an SDFT are arranged in a row or a column. We now consider this desirable transformation. Let the indices n and k be expressed as

$$\begin{aligned} n &= (n_1 L/q + n_0 q) \bmod L, \\ k &= k_1 q + k_0 \end{aligned} \quad (9)$$

where $L = N/4$, $0 \leq n_1 < q$, $0 \leq n_0 < L/q$, $0 \leq k_1 < L/q$, and $0 \leq k_0 < q$. Then, we have

$$\begin{aligned} b_e(n_0, n_1) &= (-j)^p a_e((n_1 L/q + n_0 q) \bmod L), \\ b_o(n_0, n_1) &= j^p a_o((n_1 L/q + n_0 q) \bmod L) \end{aligned} \quad (10)$$

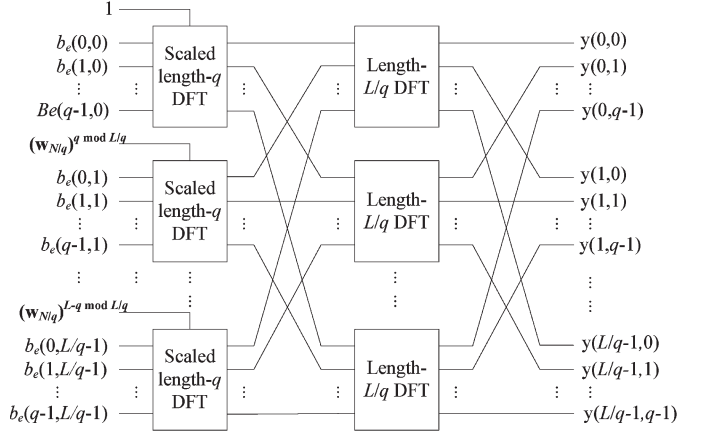


Fig. 1. General flowgraph of (11) and (13).

where $p = (n_1 + n_0 q^2/L) \bmod q$. Equations (6) and (7) can be rewritten as

$$\begin{aligned} y(k_0, k_1) &= \sum_{n_0=0}^{L/q-1} \sum_{n_1=0}^{q-1} w_{N/q}^{n_0 q \bmod L/q} b_e(n_0, n_1) w_L^{(n_1 L/q + n_0 q)(k_1 q + k_0)} \\ &= \sum_{n_0=0}^{L/q-1} Q_e(n_0, k_0) w_{L/q}^{n_0(k_1 q + k_0)} \end{aligned} \quad (11)$$

$$\begin{aligned} y'(k_0, k_1) &= \sum_{n_0=0}^{L/q-1} \sum_{n_1=0}^{q-1} w_{N/q}^{3n_0 q \bmod L/q} b_o(n_0, n_1) w_L^{(n_1 L/q + n_0 q)(k_1 q + k_0)} \\ &= \sum_{n_0=0}^{L/q-1} Q_o(n_0, k_0) w_{L/q}^{n_0(k_1 q + k_0)} \end{aligned} \quad (12)$$

where $y(k_0, k_1) = X(4k_1 q + 4k_0 + q)$, $y'(k_0, k_1) = X(4k_1 q + 4k_0 + 3q)$, and

$$Q_e(n_0, k_0) = w_{N/q}^{n_0 q \bmod L/q} \sum_{n_1=0}^{q-1} b_e(n_0, n_1) w_q^{n_1 k_0} \quad (13)$$

$$Q_o(n_0, k_0) = w_{N/q}^{3n_0 q \bmod L/q} \sum_{n_1=0}^{q-1} b_o(n_0, n_1) w_q^{n_1 k_0}. \quad (14)$$

$Q_e(n_0, k_0)$ and $Q_o(n_0, k_0)$ are two q -points scaled DFTs. Since $(k_1 q + k_0) \bmod L/q$ can be mapped to data set $\{0, 1, \dots, L/q - 1\}$, length- L/q sub-DFTs $y(k_0, k_1)$ in (11) and $y'(k_0, k_1)$ in (12) can be directly evaluated by any algorithm suitable for length- 2^m DFTs.

There are two special cases for the DFTs in (11) and (12). The first is the case when $L = q$ (it includes not only the case when $N = 4q$, but also the case when the lengths of the sub-DFTs yielded by the recursive decompositions are equal to q). The DFTs are two length- q DFTs and can be computed directly with a custom-built algorithm. The other is the case when $L = 2q$ (similarly, it includes not only the case when $N = 8q$, but also the case when the lengths of the sub-DFTs yielded by the recursive decompositions are equal to $2q$). Their compositions provide

$$y(k_0, k_1) = Q_e(0, k_0) + (-1)^{k_1 q + k_0} Q_e(1, k_0) \quad (15)$$

$$y'(k_0, k_1) = Q_o(0, k_0) + (-1)^{k_1 q + k_0} Q_o(1, k_0). \quad (16)$$

TABLE I
COMPARISON OF SMALL N DFT ALGORITHMS

	PFA presented in [1]			Algorithms used in [2][3]			WFTA presented in [8]		
	M_p/M_p^s	A_p	Total	M_p/M_p^s	A_p	Total	M_p/M_p^s	A_p	Total
length-1 DFT	0/2	0	0/2	0/2	0	0/2	0/2	0	0/2
length-3 DFT	2/6	12	14/18	2/6	12	14/18	4/6	12	16/18
length-5 DFT	10/14	32	42/46	8/14	36	44/50	8/12	34	42/46
length-7 DFT	20/22	68	88/90	/	/	/	16/18	72	88/90
length-9 DFT	20/26	80	100/106	16/22	84	100/106	16/22	90	106/112
length-15 DFT	34/40	156	190/196	30/42	168	198/210	/	/	/

Generally, for $L > 2q$, the sequence $\{Q_e(0, k_0), Q_e(1, k_0), \dots, Q_e(L/q - 1, k_0)\}$ is the input sequence of the k_0 th length- L/q sub-DFT in (11), and the sequence $\{Q_o(0, k_0), Q_o(1, k_0), \dots, Q_o(L/q - 1, k_0)\}$ is the input sequence of the k_0 th length- L/q sub-DFT in (12). All these length- L/q sub-DFTs will be evaluated with an algorithm such as SRFFT, radix 2/8, MSRFFT, radix-4, or radix-8 algorithms.

We now summarize the scheme of the proposed algorithm for computing length- $N = q \times 2^m$ DFTs. The sequence $x(n)$ of length- N is decomposed into a length- $N/2$ sub-sequence $u(n)$ and two length- $N/4$ sub-sequences a_e and a_o . By transforming the length- $N/4$ sub-sequences into two-dimension form, scaling factors are extracted and the scaled sub-DFTs are obtained, as given in (11) and (12). The length- q unscaled/scaled sub-DFTs, defined in (13) and (14), will be calculated with the custom-built algorithm. Their outputs will be assembled into length- $N/(4q)$ sub-DFTs, which will be evaluated with any algorithm that can compute the DFTs of powers-of-two, such as SRFFT, radix 2/8, radix-8, MSRFFT, etc. This process is repeated recursively for the new resulting subsequences of length- $N/2$ unless its length is equal to q or $2q$. A general signal flowgraph for the DFT of length- $N/4$ illustrated in (11) and (13) is given in Fig. 1. It is similar to that of (12) and (14).

C. q -Points Unscaled/Scaled DFTs

The SDFT technique is often used as a trick in some algorithms [2], [3]. Its influence on complexity does not gain enough attention. However, the SDFT technique plays an important pole in some algorithms, such as the MSRFFT algorithm in [9].

A length- q SDFT is defined as

$$X(k) = s \sum_{n=0}^{q-1} x(n) w_N^{nk}, \quad 0 \leq k < q \quad (17)$$

where s is the scaling factor, a real number. Some small N FFT algorithms have been introduced in [1] and [8]. Let M_q^s , M_q , and A_q be, respectively, the number of real multiplication required by a SDFT of length- q , the number of real multiplications, and the number of real additions of a DFT of length- q . A comparison of the number of operations of WFTA, PFA, and the algorithms in [2] and [3] is given in Table I. The Wang's algorithms in [1] has the least number of operations among these algorithms.

One particularly important aspect is that the algorithms in [1] have the structure in which "preweave" and "postweave" stages consist of only additions, negations, and multiplications

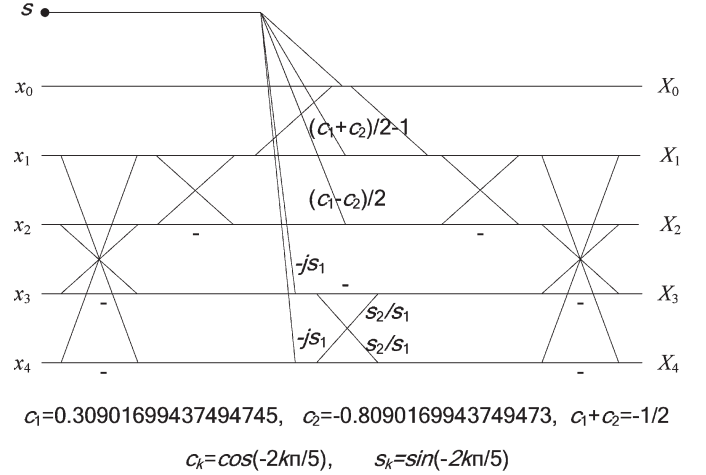


Fig. 2. Flowgraph of scaled DFT of five points.

TABLE II
ARITHMETIC COMPLEXITY OF $4q$ - AND $2q$ -POINTS DFTS

N	Multiplications	Additions
$2q$	$2M_q$	$2A_q + 2qA_2$
$4q$	$4M_q$	$4A_q + qA_4$

by j . The central stage contains only multiplication by real coefficient [8]. This structure facilitates the implementation of the SDFTs. The scaling factors will be attached to the coefficients in the central part. A signal flowgraph of the length-5 SDFT is given as an instance in Fig. 2. For different small odd q , the signal flowgraph is similar.

III. COMPUTATIONAL COMPLEXITY

In this section, we consider the performance of the proposed algorithm by analyzing its computational complexity and comparing it with the algorithms in [2]–[4], [9], and [10]. Computational complexity is expressed in terms of the number of real operations. We assume that a complex multiplication requires four real multiplications and two real additions.

A. Arithmetic Complexity

A length- $2q$ DFT can be computed using (2) and (3). A DFT of length- $4q$ can be evaluated by (5), (6), and (7). Their computational complexities are given in Table II.

When $m \geq 3$ or $N \geq 8q$, the length- $N = q \times 2^m$ DFT is decomposed into one length- $N/2$ DFT and two length- $N/4$ DFTs. Each length- $N/4$ DFT contains $2^{m-2} - 1$ length- q SDFTs, one unscaled length- q DFT, and q length- $N/(4q)$

DFTs. The length- $N/2$ DFT will be recursively decomposed until its size reaches q or $2q$. To summarize the above discussion, the computation of a length- $N = q \times 2^m$ contains the computations of $2^m - 2m$ length- q SDFTs, $2m$ nonscaled length- q DFTs, $2q$ length- 2^l DFT's (l is a variable in the range from 1 to $m - 2$), and $8N - 4qm - 8q$ extra real additions and $2N - 8mq + 8q$ extra real multiplications. Let M_N and A_N be the number of real multiplications and the number of real additions of the proposed algorithm for a length- $N = q \times 2^m$ DFT, respectively. One can easily obtain

$$M_{N=q \times 2^m} = 2q \sum_{l=1}^{m-2} M_{N=2^l} + (2^m - 2m)M_q^s + 2mM_q + 2N - 8mq + 8q \quad (18)$$

$$A_{N=q \times 2^m} = 2q \sum_{l=1}^{m-2} A_{N=2^l} + 2^m A_q + 8N - 4qm - 8q. \quad (19)$$

If a length- $N = 2^l$ sub-DFT is implemented with the SRFFT algorithm, we can replace $M_{N=2^l}$ by $(4/3)2^{ll} - (38/9)2^l + 6 + (-1)^l(2/9)$, and replace $A_{N=2^l}$ by $(8/3)2^{ll} - (16/9)2^l + 2 - (-1)^l(2/9)$. In this case, its computational complexity can be expressed as

$$M_N = \frac{4}{3}Nm - \frac{56}{9}N + 4qm + \frac{56}{9}q - (m \bmod 2)\frac{4}{9}q + (2^m - 2m)M_q^s + 2mM_q \quad (20)$$

$$A_N = \frac{8}{3}Nm - \frac{16}{9}N + \frac{16}{9}q + (m \bmod 2)\frac{4}{9}q + 2^m A_q. \quad (21)$$

Although the number of real multiplications and the number of real additions are different from (20) and (21), if a length- $N = 2^l$ sub-DFT is calculated with radix 2/8 or a higher split-radix algorithm, the total operation count are the same.

If a length- $N = 2^l$ sub-DFT is evaluated with the MSRFFT algorithm, the number of real additions is identical to that required by the SRFFT algorithm. The number of real multiplications required by the MSRFFT algorithm, $10/9 \times 2^l l - 76/27 \times 2^l - 2l + 6 - 2/9 \times (-1)^l \times l + (-1)^l \times 22/27$, will be used to replace M_{2^l} . Thus, its number of real multiplications can be obtained by

$$M_N = \frac{10}{9}Nm - \frac{112}{27}N - 2qm^2 + 10qm - \frac{116}{27}q - (m \bmod 2)\frac{44}{27}q - (-1)^m(m - 2 + (m \bmod 2))\frac{2}{9}q + (2^m - 2m)M_q^s + 2mM_q. \quad (22)$$

For different q , the symbols A_q , M_q , and M_q^s in (20)–(22) can be replaced by the values of A_q , M_q , and M_q^s in Table I.

B. Comparison

In this subsection, we compare the computational complexity of the proposed algorithm with the existing algorithms reported in [2]–[4], [9], and [10] in terms of the ratio of arithmetic complexity to $N \log_2^N$.

For $q = 1$, the arithmetic complexity of the proposed algorithm lies between that of SRFFT algorithm and that of the algorithm used for computing the sub-DFTs of length- $N/(4q)$. For $q > 1$, the proposed algorithm achieves a reduction

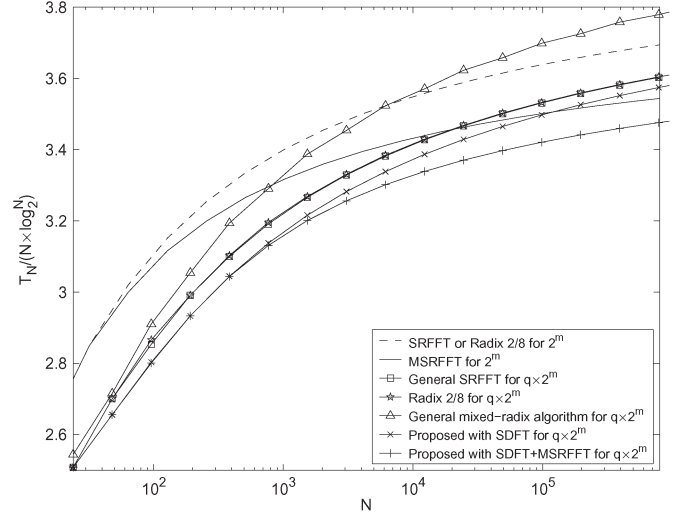


Fig. 3. Computational complexity for $N = 3 \times 2^m$.

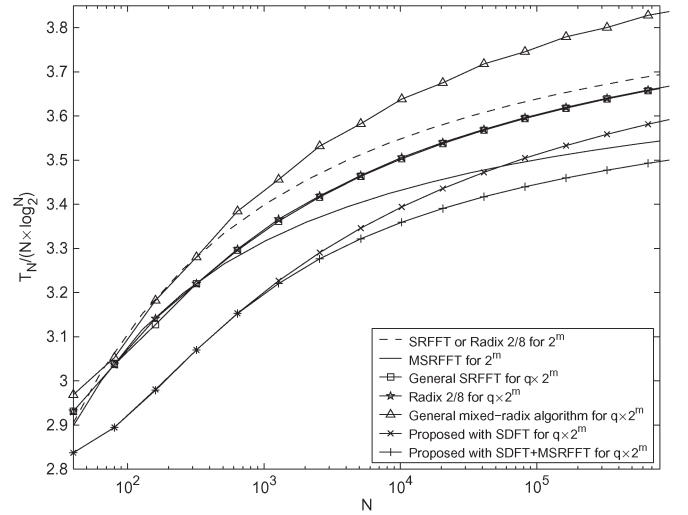


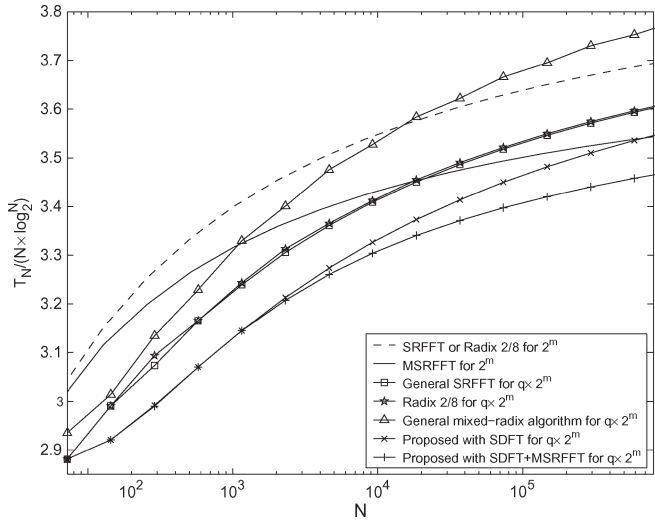
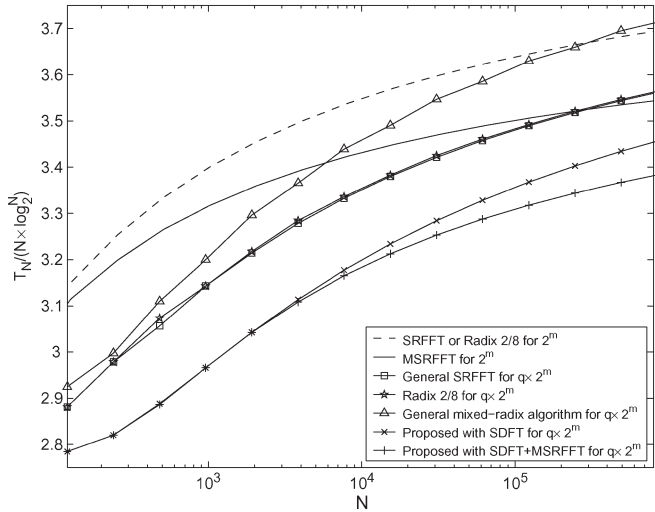
Fig. 4. Computational complexity for $N = 5 \times 2^m$.

of operations over algorithms in [2]–[4]. The reason for the reduction lies in three aspects. First, the reduction results in the scaled q -points DFTs. Second, the reduction is derived from the higher efficiency of the algorithm used for computing q -points DFTs, as given in Table I. The final part depends on the algorithm used for computing the length- $N/(4q)$ sub-DFTs.

Figs. 3–6 show the comparisons of arithmetic complexity. It can be seen that the proposed algorithm, when N is no greater than 10^6 , achieves reduction of operations over the algorithms in [2]–[4]. When N is greater than 10^6 , the proposed algorithm with the SDFT+MSRFFT technique will continue to remain the lowest complexity, and the proposed algorithm with only the SDFT technique will require more operations than the MSRFFT algorithm but fewer operations than the others.

IV. QUANTITATIVE LOSS

In order to measure the roundoff loss [11], [12], we compute the L_2 relative error of the related algorithms compared to the exact results, for pseudo random inputs $x(n) \in [-0.5, 0.5]$, in 32-bits single precision, on Pentium E6700 with Windows 7

Fig. 5. Computational complexity for $N = 9 \times 2^m$.Fig. 6. Computational complexity for $N = 15 \times 2^m$.TABLE III
 L_2 RELATIVE ERROR FOR $N = 3 \times 2^m$ ($\times 10^{-8}$)

N	[2]	[3]	[4]	Proposed Algorithm Using SDFT		
				SRFFT	Radix 2/8	MSRFFT
96	8.30	8.10	8.26	7.76	7.65	7.76
192	8.68	9.30	8.61	8.85	8.99	8.85
384	9.71	9.78	9.77	9.75	9.43	9.85
768	10.8	10.3	10.8	10.6	10.5	10.6
1536	10.6	10.8	10.7	10.7	11.0	11.3
3072	11.5	11.5	11.6	11.3	11.2	11.6
6144	11.8	12.1	12.0	12.0	11.9	12.4
12288	12.4	12.4	12.5	12.3	12.2	12.7
24576	12.8	12.9	13.0	12.7	12.8	13.2

and Microsoft Visual Studio 2005. The coefficients are precomputed, by rounding/truncating the 64-bits double precision coefficients into 32-bits single precision coefficients, and restored in a table. The exact results are from the implementation of the common approach in the case of 64-bit double precision. The results in Table III show that the proposed algorithm, while using SDFT for computing the length- q DFTs, has a smaller quantization loss than the algorithms in [2]–[4].

One can see from Fig. 2 that a coefficient of SDFT is a product of two real numbers. The coefficients can be precomputed with higher accuracy. By using this technique, the evaluations of coefficients of SDFT are free-loss. Compared with the unscaled DFT, SDFT has higher precision.

V. CONCLUSION

In this brief, we have proposed a new algorithm for computing a DFT of length- $N = q \times 2^m$. When $q > 1$, a reduction of arithmetic complexity and an improvement in computational precision can be obtained over the algorithms in [2]–[4]. The proposed algorithm can calculate a DFT in several ways. If the MSRFFT algorithm is used for powers-of-two sub-DFTs, the real multiplications will be further reduced. If the radix 2/8 FFT algorithm or a higher Split-Radix algorithm in [7] is used for computing powers-of-two sub-DFTs, the reductions of data transfers and access to the look-up tables can be obtained [13]. The proposed algorithm is a mixed-radix algorithm more suitable for a pipelined architecture than a memory-based architecture [14], [15]. In a memory-based architecture, the irregularity of computations will influence the final time spent. However, the idea of the proposed algorithm can be applied to the fixed-radix algorithms, and a compromise between regularity and arithmetic complexity can be reached.

REFERENCES

- [1] Z. Wang, "A prime factor fast W transform algorithm," *IEEE Trans. Signal Process.*, vol. 40, no. 9, pp. 2361–2368, Sep. 1992.
- [2] G. Bi and Y. Chen, "Fast DFT algorithms for length $N = q \times 2^m$," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 45, no. 6, pp. 685–690, Jun. 1998.
- [3] S. Bouguezel, M. Ahmad, and M. Swamy, "A new radix-2/8 FFT algorithm for length- $N = q \times 2^m$ DFTs," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 51, no. 9, pp. 1723–1732, Sep. 2004.
- [4] C. Hsiao, Y. Chen, and C. Lee, "A generalized mixed-radix algorithm for memory-based FFT processors," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 57, no. 1, pp. 26–30, Apr. 2010.
- [5] W. Zheng and K. Li, "Split radix algorithm for length 6^m DFT," *IEEE Signal Process. Lett.*, vol. 20, no. 7, pp. 713–716, Jul. 2013.
- [6] P. Duhamel and M. Vetterli, "Fast Fourier transforms: A tutorial review and a state of the art," *Signal Process.*, vol. 19, no. 4, pp. 259–299, Apr. 1990.
- [7] S. Bouguezel, M. Ahmad, and M. Swamy, "A general class of split-radix FFT algorithms for the computation of the DFT of length- 2^m ," *IEEE Trans. Signal Process.*, vol. 55, no. 8, pp. 4127–4138, Aug. 2007.
- [8] M. Macleod, "Multiplierless Winograd and prime factor FFT implementation," *IEEE Signal Process. Lett.*, vol. 11, no. 9, pp. 740–743, Sep. 2004.
- [9] S. G. Johnson and M. Frigo, "A modified split-radix FFT with fewer arithmetic operations," *IEEE Trans. Signal Process.*, vol. 55, no. 1, pp. 111–119, Jan. 2007.
- [10] P. Duhamel and H. Hollmann, "Split-radix FFT algorithm," *Electron. Lett.*, vol. 20, no. 1, pp. 14–16, Jan. 1984.
- [11] B. Liu and Tran-Thong, "Fixed-point fast Fourier transform error analysis," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 24, no. 6, pp. 563–573, Dec. 1976.
- [12] W. Chang and T. Nguyen, "On the fixed-point accuracy analysis of FFT algorithms," *IEEE Trans. Signal Process.*, vol. 56, no. 10, pp. 4673–4682, Oct. 2008.
- [13] G. Bi, G. Li, and X. Li, "A unified expression for split-radix DFT algorithms," in *Proc. ICCAS*, 2010, pp. 323–326.
- [14] E. G. Larsson and O. Gustafsson, "The impact of dynamic voltage and frequency scaling on multicore DSP algorithm design [exploratory DSP]," *IEEE Signal Process. Mag.*, vol. 28, no. 3, pp. 127–144, May 2011.
- [15] S.-J. Huang and S.-G. Chen, "A high-throughput radix-16 FFT processor with parallel and normal input/output ordering for IEEE 802.15. 3c systems," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 59, no. 8, pp. 1752–1765, Aug. 2012.