Canonic Real-Valued FFT Structures

Megha Parhi, Yingjie Lao, Keshab K. Parhi
Department of Electrical and Computer Engineering
University of Minnesota
Minneapolis, MN 55455
{parhi002, laoxx025, parhi}@umn.edu

***Abstract*—An N-point FFT processes N complex signals to compute N output complex signals using decimation-in-time (DIT) or decimation-in-frequency (DIF) approach. The FFT makes use of $n = log_2 N$ stages of computations where each stage computes N complex signals; N is assumed to be power-of-two. This paper considers implementation of a real signal of length N. Since the degrees of freedom of the input data is N, each stage of the FFT should not need to compute more than N signal values, where a signal value can corresponds to a purely real or purely imaginary value. Any more than N samples computed at any FFT stage is inherently redundant. This paper, for the first time, presents novel DIT and DIF structures for computing real FFT, referred as RFFT, that are canonic with respect to the number signal values computed at each FFT stage. In the proposed structure, in an N-point RFFT, exactly N signal values are computed at the output of each FFT stage and at the output. No prior canonic DIT RFFT structure was presented before. This paper, for the first time, presents a formal approach to compute RFFT using DIT in a canonic manner. While canonic FFT structures based on decimation-in-frequency were presented before, these structures were derived in an *adhoc* way. This paper presents a formal method to derive canonic DIF RFFT structures.**

I. Introduction

FFT is an important topic in Digital Signal Processing (DSP) and is widely used in applications such as telecommunications, biomedical signal processing, and spectral analysis. Nowadays, there has been an increasing interest in the computation of FFT of real-valued signals, referred as RFFT. This is because most of the physical signals, such as biomedical signals, are real. The real-valued signals exhibit conjugate symmetry giving rise to redundancies. This property can be exploited to reduce both arithmetic and architectural complexities.

A number of RFFT computation algorithms and implementations have been proposed for both pipelined and in-place architectures in the literature [1]–[4]. An approach to computing an N-point RFFT using an $\frac{N}{2}$-point complex FFT was presented in [1]. However, this approach requires significant amount of post-processing. Custom pipelined architectures for RFFT have been proposed in [5], [6]. In the work of [6], the computations of $\frac{N}{2} - 1$ conjugate-symmetric samples were eliminated to obtain more efficient RFFT structures, where N represents the size of the FFT. Here, we consider a complex signal as two signals: real part signal and imaginary part signal. Therefore, in these architectures, the number of signals computed at the output is the same as the input, i.e., N. However, although the outputs are canonic in the number of signals, these architectures still exhibit redundancies at the intermediate stages, as they are composed of hybrid datapaths consisting of both complex and real datapaths. Recently, pipelined architectures consisting of only real datapaths for decimation-in-frequency (DIF) RFFT were proposed in [7]. Real-valued FFT architectures for radix 2^3 and radix 2^4 were presented in [8] based on hybrid datapath. The architecture in [8] does not maintain the canonic property in number of signal values computed at the output of each FFT stage.

The goal of this paper is to design general RFFTs that are canonic with respect to the number of signals at the output of each stage, i.e., for an N-point RFFT, the total number of values computed at the output of each stage should be N. Furthermore, each stage should contain maximum $\frac{N}{2}$ real butterflies as opposed to $\frac{N}{2}$ complex butterflies. In this paper, we demonstrate that it is possible to compute only N independent values at each stage for an N-point RFFT. This property has not been explicitly studied before. Although this property is satisfied by only one prior architecture proposed in [7], general approaches for designing canonic RFFT computations have been not presented. This paper makes two contributions: first, we present an approach to design canonic RFFT computation based on decimation-in-time (DIT) approach; second, we present a formal method to design RFFT structures for decimation-in-frequency (DIF) approach as presented in [7].

The organization of the paper is as follows. Section II explains the basics of the RFFT. Section III introduces the canonic DIT RFFTs. An algorithm for canonic DIT RFFT computation and its generalization to arbitrary size RFFT are presented in Section IV. Section V presents an approach for computing canonic DIF RFFT. Section VI presents a comparison of the proposed canonic RFFT structures with prior works. Finally, Section VII concludes the paper.

II. RFFT

The N-point Discrete Fourier Transform (DFT) for a sequence $x[n]$ is defined as [9]:

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j\frac{2\pi}{N}nk} = \sum_{n=0}^{N-1} x[n] W_N^{nk}, \quad k = 0, 1, ..., N-1, \tag{1}$$

where $W_N = e^{-j\frac{2\pi}{N}}$. The flow-graph of a 16-point DIT FFT is shown in Fig. 1. For real-valued inputs $x[n]$, it can be shown that

$$X[k] = X^*[N-k]. \tag{2}$$

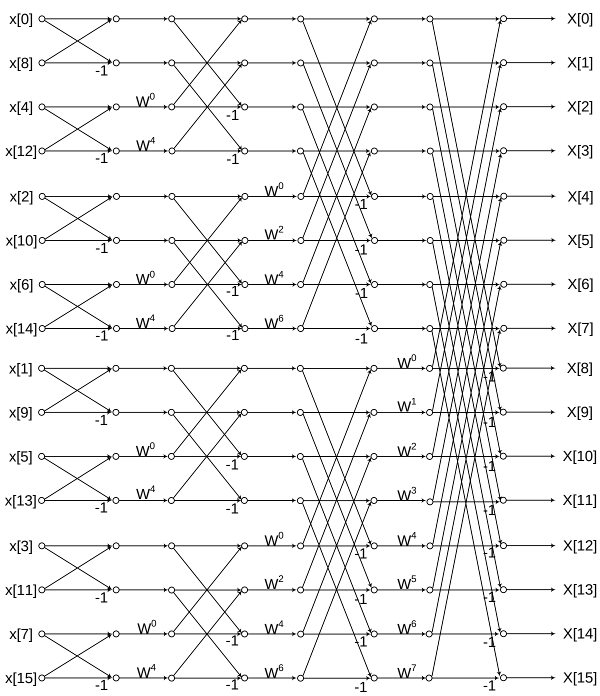

Fig. 1. Flow-graph of a 16-point DIT FFT.

In this case, there are $\frac{N}{2}-1$ conjugate output pairs, i.e., $X[k]$ and $X[N-k]$, for $k=1,2,...,\frac{N}{2}-1$. Therefore, only $\frac{N}{2}+1$ outputs need to be computed in an N-point RFFT flow-graph, since we can compute either $X[k]$ or $X[N-k]$, along with two real output signals $X[0]$ and $X[N/2]$. The total number of purely real and purely imaginary signal values is N. For example, as shown in Fig. 1, for a 16-point FFT, we can choose to only compute $X[0] \sim X[8]$, while $X[9] \sim X[15]$ can be obtained by conjugating $X[1] \sim X[7]$. Thus, only 16 values need to be computed at the output. We present RFFT structures where the size of the signal values computed at each FFT stage is exactly N; such structures satisfy the canonic property.

III. Canonic DIT RFFT

In this section, we present the flow-graphs for canonic DIT RFFT computations which are guaranteed to have N signals at each stage.

A. 4-point RFFT

A 4-point canonic DIT RFFT flow-graph is shown in Fig. 2. The nodes marked by ○ and □ respectively represent purely real and purely imaginary signals. Solid and dashed lines respectively represent purely real and purely imaginary datapaths. In the 4-point RFFT, since $X[1]$ and $X[3]$ are conjugates of each other, we can eliminate $X[3]$ by removing the bottom butterfly at the second stage. The computations of real and imaginary parts of $X[1]$ are separated as shown in Fig. 2. Note that the number of inputs, the number of outputs and the number of signal values at the intermediate stage are all the same and equal 4, the size of FFT.

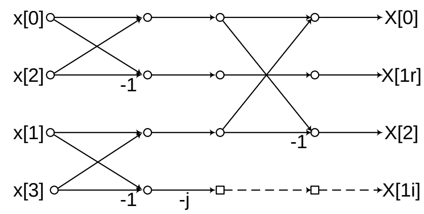

Fig. 2. Flow-graph of a canonic 4-point DIT RFFT.

B. 8-point RFFT

An 8-point canonic DIT RFFT can be obtained by merging two 4-point canonic RFFTs, as shown in Fig. 3. In an 8-point FFT, $\frac{8}{2}-1$ output computations can be eliminated. Since 2 samples have already been eliminated after the second stage by use of the 4-point canonic RFFTs, we need to eliminate $\frac{8}{2}-1-2\times 1=1$ more sample at the output. Thus, we can remove computation of $X[6]$ at the last stage, and the computation of the real part and imaginary part of $X[2]$ are separated. The real parts and imaginary parts of $X[1]$ and $X[5]$ are computed by two independent real butterflies, as these were separated in previous stages. The number of signal values computed at each FFT stage or the output is always 8; thus the structure is canonic.

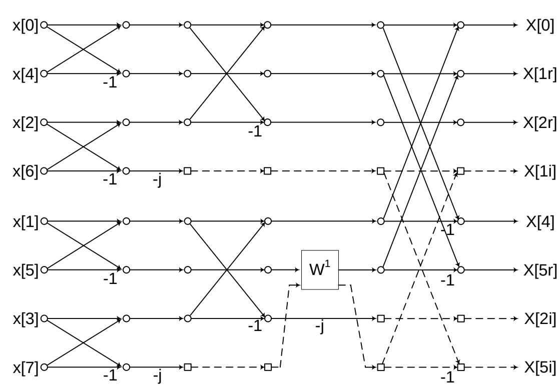

Fig. 3. Flow-graph of a canonic 8-point DIT RFFT.

C. 16-point RFFT

As shown in Fig. 4, 16-point canonic RFFT flow-graph can also be obtained by merging two 8-point canonic RFFTs. At the last stage, only $\frac{16}{2}-1-2\times 3=1$ sample needs to be eliminated.

IV. Algorithm for Canonic DIT RFFT Computation

A. Divide and Conquer

As illustrated in Section III, any N-point RFFT, where N is power-of-two, can be obtained by merging two $\frac{N}{2}$-point

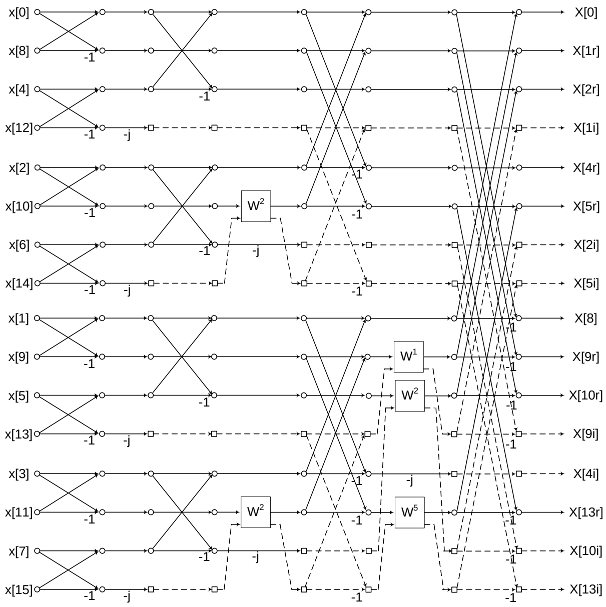

Fig. 4. Flow-graph of a canonic 16-point DIT RFFT.

RFFTs using the DIT property algorithm. This property can be expressed as

$$\begin{aligned} X[k] &= \sum_{n=0}^{N-1} x[n]W_N^{nk} \\ &= \sum_{n=0}^{\frac{N}{2}-1} x[2n]W_N^{2nk} + \sum_{n=0}^{\frac{N}{2}-1} x[2n+1]W_N^{(2n+1)k} \\ &= \sum_{n=0}^{\frac{N}{2}-1} x[2n]W_N^{2nk} + W_N^k \sum_{n=0}^{\frac{N}{2}-1} x[2n+1]W_N^{2nk} \\ & \qquad\qquad k = 0, 1, ..., N-1, \end{aligned} \tag{3}$$

where $\sum_{n=0}^{\frac{N}{2}-1} x[2n]W_N^{2nk}$ and $\sum_{n=0}^{\frac{N}{2}-1} x[2n+1]W_N^{2nk}$ respectively represent two $\frac{N}{2}$-point RFFTs for the even part and the odd part of the inputs. Twiddle factors W_N^k need to be multiplied to the odd part before the butterflies at the last stage, as shown in Fig. 3 and Fig. 4.

Therefore, due to the regularity of the canonic RFFT flow-graph, the canonic RFFT flow-graph can be extended for any $N = 2^n$-point DIT RFFT recursively. Note that for an $\frac{N}{2}$-point canonic RFFT, $\frac{N}{2}/2 - 1 = \frac{N}{4} - 1$ signals have been eliminated. Thus, when merging two $\frac{N}{2}$-point canonic RFFTs, only one more computation and its corresponding butterfly need to be eliminated at the last stage, as $\frac{N}{2} - 1 - 2 \times (\frac{N}{4} - 1) = 1$. We choose to eliminate the computation of $X[\frac{3N}{4}]$. Therefore, the butterfly which computes $X[\frac{N}{4}]$ and $X[\frac{3N}{4}]$ is removed, and the real and imaginary parts of $X[\frac{N}{4}]$ are separated. The twiddle factor before the bottom input of the removed butterfly (i.e., the $(\frac{3N}{4}+1)$th signal) can be calculated as $W_N^{\frac{3N}{4}+1-\frac{N}{2}-1} = W_N^{\frac{N}{4}} = -j$, as W_N^k is the twiddle factors before the bottom inputs of the butterflies at the last stage for $k = 0, 1, ..., \frac{N}{2} - 1$. This operation is further illustrated as in Fig. 5.

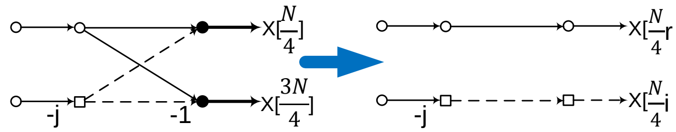

Fig. 5. Remove butterfly and separate real part and imaginary part for DIT RFFT.

B. Generalization to $N = 2^n$-point DIT RFFT

We summarize the pattern for an $N = 2^n$-point canonic DIT RFFT in this section. The key idea of the proposed approach to implement a RFFT involves removing one butterfly for each 2^m-point RFFT at stage m for $2 \leq m \leq n$. Note that there are 2^{n-m} 2^m-point RFFTs at stage m. Therefore, we need to perform the operation as shown in Fig. 5 to the butterflies whose bottom inputs are the $(2^m i + 3 \times 2^{m-2} + 1)$th signals, where $i = 0, 1, ..., 2^{n-m} - 1$. Similarly, it can be proved that the twiddle factors before the $(2^m i + 3 \times 2^{m-2} + 1)$th signal at stage m are

$$W_N^{2^{m-2}} \times W_N^{2^{n-m}} = W_N^{2^{n-2}} = W_N^{\frac{N}{4}} = -j, \tag{4}$$

as the twiddle factors for stage m are in group of 2^m (i.e., 2^{m-1} 0's and $W_N^{0:2^{n-m}:2^{n-1}-2^{n-m}}$).

This pattern is also summarized as in Table I. The eliminations are performed from stage 1 to stage n to obtain the final N-point canonic DIT RFFT flow-graph.

Moreover, it can be observed that the output signals for an N-point canonic RFFT will be $\{S_{\frac{N}{2}}, \frac{N}{2} + S_{\frac{N}{2}}\}$, where $S_{\frac{N}{2}}$ is the output set of an $\frac{N}{2}$-point canonic RFFT; and then replace $X[\frac{N}{4}]$ and $X[\frac{3N}{4}]$ with $X[\frac{N}{4}r]$ and $X[\frac{N}{4}i]$, respectively. This is also consistent with the patterns described above, i.e., remove one butterfly whose bottom input is the $(\frac{3N}{4} + 1)$th signal. This property is also summarized in Table II.

Based on the patterns presented above, a 32-point DIT RFFT structure is designed as shown in Fig. 6. In this structure, the number of signal values computed at the output of each FFT stage or the output is 32; thus, this structure is canonic.

V. Algorithm for Canonic DIF RFFT Computation

In this section, we discuss the canonic computation for DIF RFFT. An alternate flow-graph for a 16-point DIF FFT is shown in Fig. 7. It can be seen that the DIT and DIF only differ in the twiddle factors at different stages. Similar to DIT, we eliminate the redundancies in the FFT computation to achieve the canonic RFFT structure. However, unlike the DIT, the DIF RFFT is *not* symmetric, i.e., the twiddle factors in the top $\frac{N}{2}$-point DIF RFFT are not the same as for the bottom $\frac{N}{2}$-point DIF RFFT. Therefore, we *cannot* directly use the divide and conquer idea to design a canonic DIF RFFT structures.

Moreover, we have to transform the twiddle factors for the DIF RFFT to ensure non-redundancy. For example, for a 16-point RFFT, there will be exactly 22 signal values at the beginning of the second stage, since no butterfly can be removed in the first stage and twiddle factors W^1, W^2, W^3, W^5, W^6, W^7 lead to complex signals. Therefore, we have to

TABLE I. Bottom Input Positions of the Removed Butterflies.

Stage	1	2	3	...	m	...	n
Positions	none	4,8,12,...	7,15,23,...	...	$2^m i + 3 \times 2^{m-2} + 1$, for $i = 0, 1, ..., 2^{n-m} - 1$	...	$3 \times 2^{n-2} + 1$

TABLE II. Canonic DIT RFFT Output Signals.

RFFT Size N	4	8	16	32
Output S_N	0, 1r, 2, 1i	0, 1r, 2r, 1i	0, 1r, 2r, 1i, 4r, 5r, 2i, 5i	0, 1r, 2r, 1i, 4r, 5r, 2i, 5i, 8r, 9r, 10r, 9i, 4i, 13r, 10i, 13i
		4, 5r, 2i, 5i	8, 9r, 10r, 9i, 4i, 13r, 10i, 13i	16, 17r, 18r, 17i, 20r, 21r, 18i, 21i, 8i, 25r, 26r, 25i, 20i, 29r, 26i, 29i

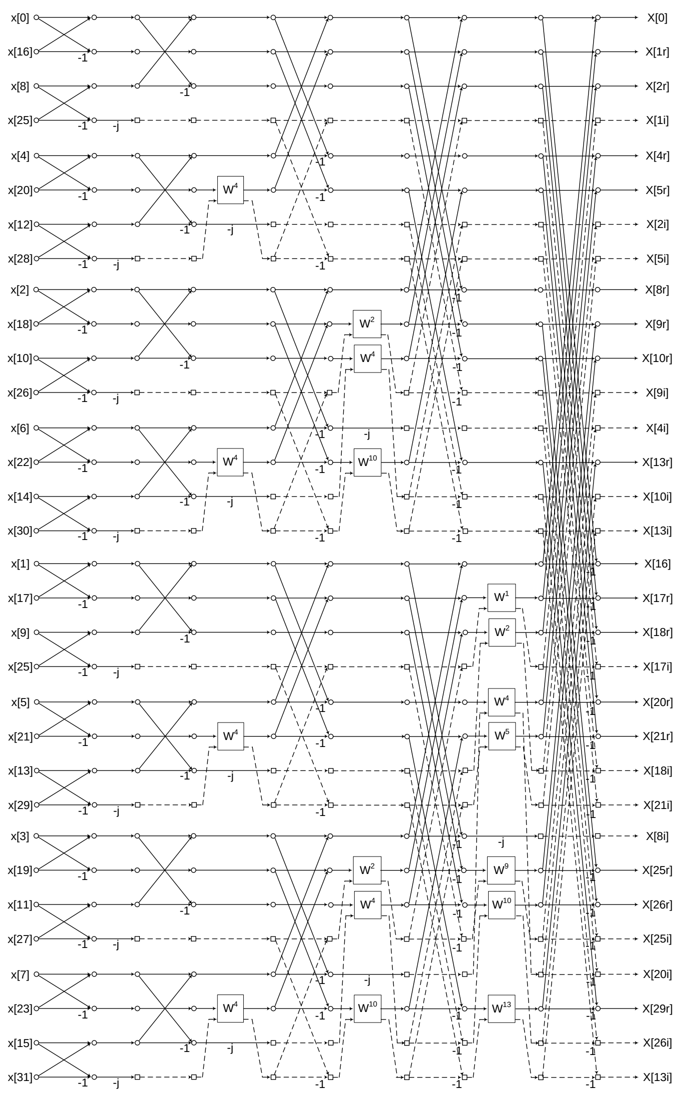

Fig. 6. Flow-graph of a canonic 32-point DIT RFFT.

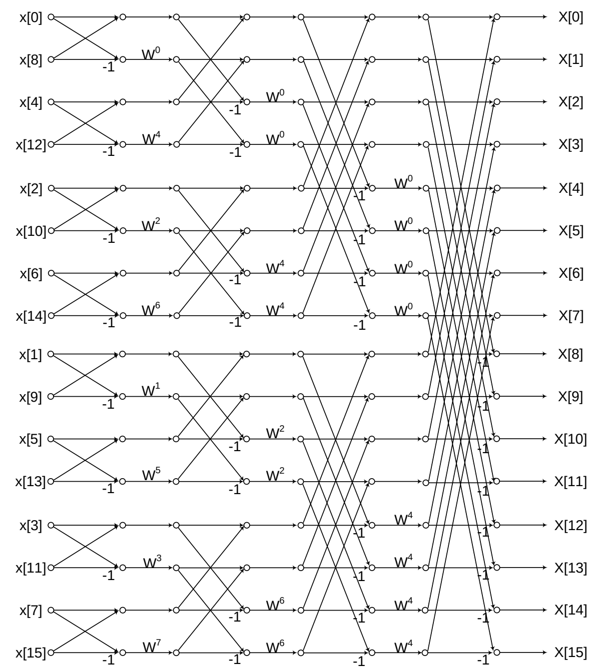

Fig. 7. Alternate flow-graph of a 16-point DIF FFT.

push these twiddle factors to a later stage to guarantee that only 16 signal values are input to the second FFT stage.

We could follow the same pattern as presented in Table I to design canonic DIF RFFT computation. The same number of butterflies will be removed at each stage. It can be observed that the twiddle factors before the top input and the bottom input of the removed butterflies can be expressed as W^k and $W^{k+N/4}$, respectively. Therefore, instead of performing the transformation shown in Fig. 5 for a DIT RFFT, the transformation in Fig. 8 is used to generate a canonic DIF RFFT structure. Fig. 9 shows an example of a 16-point canonic DIF RFFT computation.

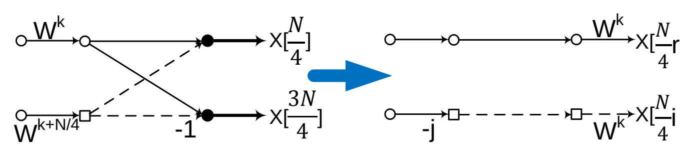

Fig. 8. Remove butterfly and separate real part and imaginary part for DIF RFFT.

In conclusion, the canonic DIT and DIF RFFT computations can be designed by following the same pattern shown in Table I. In order to remove the redundant butterflies and to separate the real and imaginary parts of complex values, we use the transformation in Fig. 5 for a DIT RFFT, or the transformation in Fig. 8 for a DIF RFFT. The patterns for the output signals for both the designs are the same and shown in Table II.

VI. Performance Comparison

The number of butterflies in the proposed canonic RFFT computation is the same as the previous works in [6], [7]. However, these structures require different numbers of twiddle factors. Furthermore, the number of twiddle factor operations even differs in the proposed canonic DIT RFFT and canonic

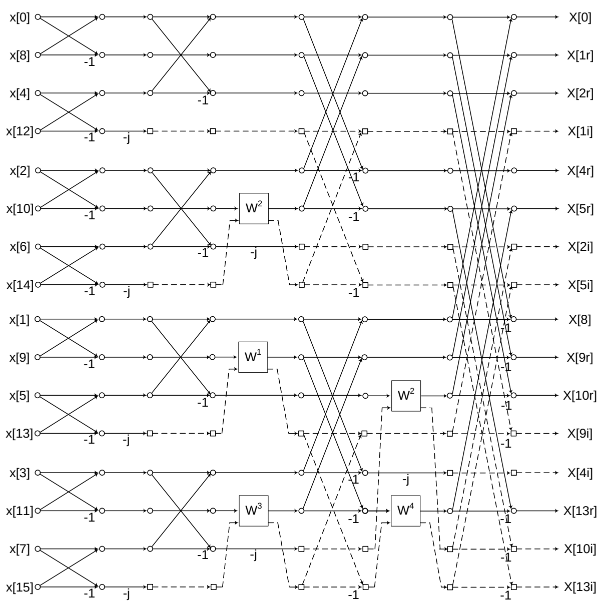

Fig. 9. Flow-graph of a canonic 16-point DIF RFFT.

DIF RFFT, as the transformations for removing the butterflies are different for the DIT RFFT and DIF RFFT, as shown in Fig. 5 and Fig. 8, respectively. Note that we do not consider $W^{\frac{N}{4}}$, $W^{\frac{N}{2}}$, and $W^{\frac{3N}{4}}$ as twiddle factor operations, since they only involve negation or switching.

It can be seen for an $N = 2^n$-point canonic DIT RFFT, the number of twiddle factors before stage k is

$$2^{n-k} \times (2^{k-2} - 1), \text{ for } 3 \leq k \leq n. \tag{5}$$

Therefore, the total number of twiddle factor operations can be calculated as

$$\begin{aligned} \#\text{twiddle factor} &= (n-1) \times 2^{n-2} - 2^{n-1} + 1 \\ &= (n-3) \times 2^{n-2} + 1. \end{aligned} \tag{6}$$

However, for an $N = 2^n$-point canonic DIF RFFT, the number of twiddle factors before stage k is

$$2^{k-3} \times (2^{n-k+1} - 2) - 1, \text{ for } 3 \leq k \leq n. \tag{7}$$

Therefore, the total number of twiddle factor operations can be calculated as

$$\begin{aligned} \#\text{twiddle factor} &= (n-2) \times 2^{n-2} - 2^{n-1} + 2 + (n-2) \\ &= (n-4) \times 2^{n-2} + n. \end{aligned} \tag{8}$$

As a result, it can be concluded that the canonic DIF RFFT computation has less twiddle factor operations than canonic DIT RFFT for $n \geq 4$, while canonic DIT and DIF RFFT computations have the same number of twiddle factors for $n = 2, 3$.

Table III compares the number of twiddle factor operations of the different approaches for the computation of the RFFT. As it is shown, the number of twiddle factor operations in all the approaches has the same order of magnitude. The proposed canonic DIF RFFT computation has less twiddle factor operations than the computation in [6], while it has the same performance as the work in [7].

TABLE III. Comparison of Twiddle Factor Operation and Datapath for $N = 2^n$-point RFFT

RFFT Algorithm	#Twiddle Factor Operations	Datapath
[6] DIF	$(n - \frac{7}{2}) \times 2^{n-2} + n - 1$	hybrid
[7] DIF	$(n-4) \times 2^{n-2} + n$	real
Proposed DIT	$(n-3) \times 2^{n-2} + 1$	real
Proposed DIF	$(n-4) \times 2^{n-2} + n$	real

VII. Conclusion

This paper has introduced the novel notion of canonic RFFT computations where the number of signal values computed at each intermediate FFT stage is same as the size of RFFT. The canonic computations for $N = 2^n$-point DIT and DIF RFFTs are proposed. It is shown that canonic DIF RFFT structures require less twiddle factor operations than the DIT counterparts. The proposed canonic structures are not necessarily canonic with respect to the twiddle factor operations and are non-unique.

References

[1] H. V. Sorensen, D. L. Jones, M. Heideman, and C. S. Burrus, "Real-valued fast fourier transform algorithms," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 35, no. 6, pp. 849–863, 1987.

[2] H.-F. Chi and Z.-H. Lai, "A cost-effective memory-based real-valued FFT and hermitian symmetric IFFT processor for DMT-based wire-line transmission systems," in *IEEE International Symposium on Circuits and Systems (ISCAS)*, 2005, pp. 6006–6009.

[3] M. Ayinala, Y. Lao, and K. K. Parhi, "An in-place FFT architecture for real-valued signals," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 60, no. 10, pp. 652–656, 2013.

[4] Y. Voronenko and M. Puschel, "Algebraic signal processing theory: Cooley–Tukey type algorithms for real DFTs," *IEEE Transactions on Signal Processing*, vol. 57, no. 1, pp. 205–222, 2009.

[5] M. Garrido, K. K. Parhi, and J. Grajal, "A pipelined FFT architecture for real-valued signals," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 56, no. 12, pp. 2634–2643, 2009.

[6] M. Ayinala, M. Brown, and K. K. Parhi, "Pipelined parallel FFT architectures via folding transformation," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 20, no. 6, pp. 1068–1081, 2012.

[7] S. A. Salehi, R. Amirfattahi, and K. K. Parhi, "Pipelined architectures for real-valued FFT and hermitian-symmetric IFFT with real datapaths," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 60, no. 8, pp. 507–511, 2013.

[8] M. Ayinala and K. K. Parhi, "FFT architectures for real-valued signals based on radix-2^3 and radix-2^4 algorithms," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 60, no. 9, pp. 2422–2430, 2013.

[9] A. V. Oppenheim, R. W. Schafer, and J. R. Buck, *Discrete-time signal processing*. Prentice-hall Englewood Cliffs, 1989.