

RESEARCH

Open Access



Canonic FFT flow graphs for real-valued even/odd symmetric inputs

Yingjie Lao¹ and Keshab K. Parhi^{2*} 

Abstract

Canonic real-valued fast Fourier transform (RFFT) has been proposed to reduce the arithmetic complexity by eliminating redundancies. In a canonic N -point RFFT, the number of signal values at each stage is canonic with respect to the number of signal values, i.e., N . The major advantage of the canonic RFFTs is that these require the least number of butterfly operations and only real datapaths when mapped to architectures. In this paper, we consider the FFT computation whose inputs are not only real but also even/odd symmetric, which indeed lead to the well-known discrete cosine and sine transforms (DCTs and DSTs). Novel algorithms for generating the flow graphs of canonic RFFTs with even/odd symmetric inputs are proposed. It is shown that the proposed algorithms lead to canonic structures with $\frac{N}{2} + 1$ signal values at each stage for an N -point real even symmetric FFT (REFFT) or $\frac{N}{2} - 1$ signal values at each stage for an N -point RFFT real odd symmetric FFT (ROFFT). In order to remove butterfly operations, several twiddle factor transformations are proposed in this paper. We also discuss the design of canonic REFFT for any composite length. Performances of the canonic REFFT/ROFFT are also discussed. It is shown that the flow graph of canonic REFFT/ROFFT has less number of interconnections, less butterfly operations, and less twiddle factor operations, compared to prior works.

Keywords: Fast Fourier transform (FFT), Real-valued FFT (RFFT), Canonic flow graph, Even symmetric inputs, Odd symmetric inputs, Twiddle factor transformation, Discrete cosine transform (DCT), Discrete sine transform (DSTs)

1 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. There has been a significant interest in improving the performance of FFT for specific applications. One such example is computing FFT of real-valued signals, referred as RFFT. Many physical signals, such as biomedical signals, are real. The real-valued signals exhibit conjugate symmetry in spectral domain giving rise to redundancies. This property can be exploited to reduce both arithmetic and architectural complexities.

Although most FFT algorithms were developed for complex-valued sequences, redundancies and symmetries in all of these algorithms can be exploited to reduce the number of multiplications and storage by roughly a factor of 2 for RFFTs. 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 compute 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 [5–8]. In [5], the computations of $\frac{N}{2} - 1$ conjugate-symmetric samples were eliminated to obtain hardware-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

*Correspondence: parhi@umn.edu

²Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis, MN 55455, USA

Full list of author information is available at the end of the article

datapaths for decimation-in-frequency (DIF) RFFT were proposed in [9]. Real-valued FFT architectures for radix-2³ and radix-2⁴ were presented in [10] based on hybrid datapaths. In contrast to the work in [9], the architectures in [10] do not maintain the canonic property in the number of signal values computed at the output of each FFT stage. The designs of RFFTs for both decimation-in-time (DIT) and DIF approaches that are canonic with respect to the number of signals at the output of each stage (i.e., *data-canonic*) have been proposed in [11]. For a canonic N -point RFFT, the total number of values computed at the output of each stage is guaranteed to be N . Furthermore, each stage only contains maximum $\frac{N}{2}$ real butterflies as opposed to $\frac{N}{2}$ complex butterflies.

This paper explores the design of canonic FFT flow graphs for when inputs are real-valued and also even/odd symmetric, referred as REFFT and ROFFT, respectively. The motivation of this work is that linear-phase FIR filter impulse responses are even/odd symmetric. For example, the type 1 FIR filter has odd number of taps where the values of the taps are even symmetric. As a result, we can improve the computation of $H[k]$ from $h[n]$ by eliminating the redundancies. Therefore, instead of computing $y[n]$ by $x[n] * h[n]$, we can choose to compute the IFFT of $X[k]H[k]$ to obtain the output $y[n]$, as shown in Fig. 1. The complexity of $H[k]$ can be reduced by the proposed REFFT instead of RFFT.

The main contribution of this paper is the design of novel algorithms for canonic REFFT/ROFFT. We also propose twiddle factor transformations, which are required to transform the structures to be canonic and to reduce arithmetic complexity. Note that the dataflows of REFFT and ROFFT indeed lead to DCT type 1 and DST type 1, respectively. A number of prior works have derived data-canonic DCT and DST algebraically [12–14]. However, our starting point is different from these prior works, since we begin to eliminate the redundancy from canonic RFFT structures. In addition, our approach will result in more hardware-efficient architectures and dataflow with more regularity. Furthermore, we present the algorithms for generating REFFT/ROFFT for any composite length, which has not been investigated in the literature. Except type 1, architectures for DCT/DST types 2–4 have also been studied in [15, 16].

The organization of the paper is as follows: Section 2 provides a brief overview of FFT, RFFT, and canonic RFFT

and introduces REFFT/ROFFT. Examples of canonic REFFTs and their generalizations to an algorithm for any $N = 2^n$ size are presented in Section 3. In Section 4, we describe the pre-processing that is required before performing the proposed algorithms. Section 5 presents an approach for generating canonic power-of-two size ROFFT. In Section 6, we present an approach to design canonic REFFT for any composite length. Section 7 discusses the performances of canonic REFFT/ROFFT. Finally, Section 8 concludes the paper.

2 Background

2.1 FFT

The N -point discrete Fourier transform (DFT) for a sequence $x[n]$ is defined as [17]

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

where $W_N = e^{-j\frac{2\pi}{N}}$. FFT is a fast algorithm to compute the DFT [18]. In algorithmic terms, the DFT requires $O(N^2)$ arithmetic operations, whereas the FFT requires $O(N \log N)$ arithmetic operations. The original DFT equation can be rearranged using different radices to design various FFT algorithms [19–22]. These algorithms and architectures provide unique tradeoffs that can be exploited for an intended application. A 16-point radix-2 FFT flow graph is shown in Fig. 2. Note that the minus signs in the lower paths of butterflies are omitted.

2.2 Real-valued FFT

For real-valued inputs $x[n]$, it can be shown that

$$X[k] = X^*[N-k]. \quad (2)$$

In this case, there are $\frac{N}{2} - 1$ conjugate output pairs, i.e., $X[k]$ and $X[N-k]$, for $k = 1, 2, \dots, \frac{N}{2} - 1$. Therefore, only $\frac{N}{2} + 1$ outputs need to be computed in an N -point RFFT, since we can compute either $X[k]$ or $X[N-k]$, along with 2 real output signals $X[0]$ and $X[N/2]$. The total number of purely real and purely imaginary signal values is N . A 16-point RFFT example is shown in Fig. 2. Only 9 samples consisting of 16 values need to be computed at the output. This property of RFFT can be utilized to simplify the computation. The 16-point radix-2 RFFT is shown in Fig. 3. The shaded regions in boxes in Fig. 3 are removed and only 9 outputs of the FFT are needed, where the nodes marked by white circle and black circle respectively represent purely real or purely imaginary signals and complex signals.

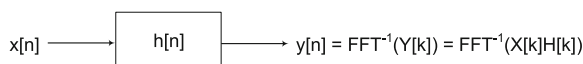


Fig. 1 Obtain $y[n]$ by computing the IFFT of $X[k]H[k]$

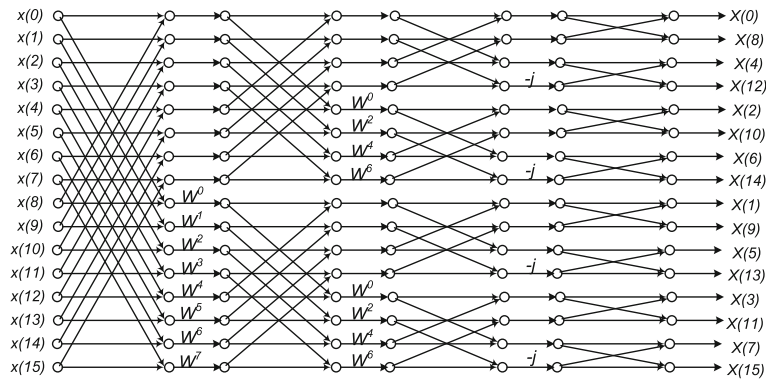


Fig. 2 Flow graph of a 16-point FFT. Input signals are assumed to be complex

2.3 Canonic RFFT

RFFT algorithms can be further optimized, according to the specific application requirements. For example, three types of RFFTs can be defined by considering the numbers of signal, multiplication, and addition, respectively:

Data-canonic (canonic): The RFFT algorithm has the least number of signals at each FFT stage (canonic is always referred to data-canonic in this paper).

Multiplication-canonic: The RFFT algorithm has the least number of multiplications.

Addition-canonic: The RFFT algorithm has the least number of additions.

Note that data-canonic RFFTs are not necessarily multiplication-canonic or addition-canonic. Algorithms for generating canonic RFFT have been presented in [11], where the number of signals is guaranteed to be N at each FFT stage for an N -point RFFT. For the 16-point RFFT as shown in Fig. 3, the outputs are canonic with respect to the number of signals, 16 (i.e., 2 real values and 7 complex values). However, the intermediate stages of the flow graph are not canonic with respect to the number of signals. For instance, there are 10 real values and 6 complex values, i.e., 22 values in total before the butterfly

operations at the second stage. Therefore, Fig. 3 is not canonic with respect to the number of signal values.

In order to reduce the number of signal values to eliminate redundancy, twiddle factor transformations are required. The *push* transformation of the twiddle factors can be described as shown in Fig. 4. We can push a factor of W^k from before the butterfly operation to after the butterfly operation to reduce the number of signal values. For example, we can push a factor of W^2 to the output of the 4th butterfly operation from the top at the 3rd stage in Fig. 3. After the twiddle factor transformation, the top input of the butterfly will be purely real and the bottom input will be purely imaginary. Therefore, the number of signals at this stage can be reduced to 16 from 18, which is canonic with respect to the number of signals. We also need to push the twiddle factors of the 6th, 7th, and 8th butterflies at the 2nd stage to obtain the canonic RFFT, as shown in Fig. 5. After pushing the twiddle factors, the top output of the butterfly can be obtained by appending the 2 inputs, since the top input is purely real and the bottom input is purely imaginary; the bottom output can be eliminated, as it is conjugate symmetric to the top output.

Note that the canonic structures for a certain size RFFT are not unique. This is because the twiddle factors can be

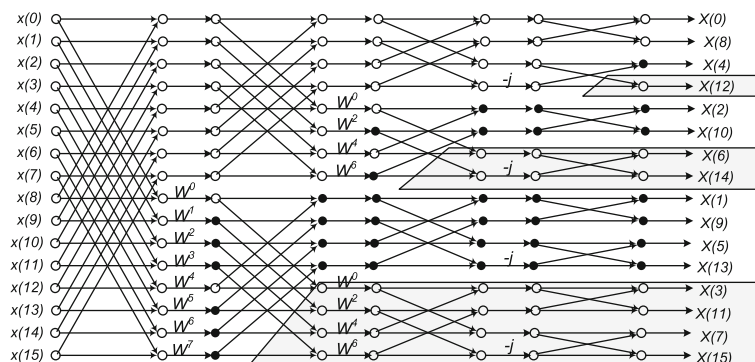


Fig. 3 Flow graph of a 16-point radix-2 real-valued FFT (RFFT). Input signals are purely real, and redundant signals in the shaded regions are removed

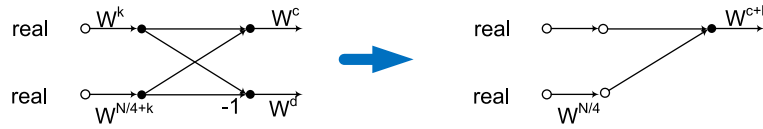


Fig. 4 Twiddle factor transformation: push

moved from one stage to another if the signals before and after are complex without altering the number of signal values. For example, the twiddle factors after the second stage of the bottom part of Fig. 5 can also be pushed to the next stage. This operation does not affect the number of signal values for each stage.

2.4 REFFT/ROFFT

When the inputs of a RFFT are even symmetric or odd symmetric, the outputs will be purely real or purely imaginary [17], which are equivalent to DCT type 1 and DST type 1, respectively. This property can also be exploited to reduce the arithmetic complexity of the RFFT, as the $\frac{N}{2} - 1$ inputs are redundant. In this paper, we present algorithms to generate canonic REFFT/ROFFT from RFFT.

3 Canonic REFFT for power-of-two length

In this section, we present the flow graphs for REFFT which eliminate the redundancies in general RFFT. The number of signals is also guaranteed to be canonic at each stage, i.e., $\frac{N}{2} + 1$ signals.

3.1 4-point REFFT

A 4-point canonic RFFT flow graph is shown in Fig. 6. The nodes marked by white circle and white square 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, due to redundancy, the bottom butterfly at the second stage is removed and the computations of real and imaginary parts of $X[1]$ are separated as shown in Fig. 6.

If the inputs are even symmetric, i.e., $x[1] = x[3]$, the outputs will be purely real. Therefore, $X[1i]$ in Fig. 6 will be 0. As a result, we can eliminate the computation of $X[1i]$ so that there will be only three signals at the output. Furthermore, we can also remove input $x[3]$ to achieve the canonic property from the beginning. However, we need to multiply $x[1]$ by 2, since $x[1] = x[3]$. The operation can be described by Fig. 7, where the butterfly operation of two inputs with the same value is replaced by a multiplication of one input by 2. Finally, the flow graph of a canonic 4-point REFFT can be derived as shown in Fig. 8.

3.2 8-point REFFT

It can be observed that flow graph in the red box of Fig. 9 is the same as the 4-point RFFT. Therefore, we can eliminate the redundancy of this 4-point RFFT by replacing it with the flow graph as shown in Fig. 8. For the bottom half of the first two stages, since $x[1] = x[7]$ and $x[3] = x[5]$, we can remove the bottom butterfly of the first stage. Consequently, the bottom two datapaths at the following stages also need to be removed. It can be calculated that the bottom four signal values after the first stage of Fig. 9 are $x[1] + x[3]$, $x[1] - x[3]$, $x[1] + x[3]$, and $x[3] - x[1]$, respectively. The butterfly simplification shown in Fig. 7 can be applied to the second butterfly operation of the second stage to eliminate redundancy, since the two inputs of the butterfly have the same value. For the twiddle factor operation W^1 after the second stage, if we assume $x[1] - x[3] = a$, then the result of the twiddle factor operation will be

$$(a - a(-j)) * W^1 = \sqrt{2}ae^{j\frac{\pi}{4}}e^{-j\frac{\pi}{4}} = \sqrt{2}a. \quad (3)$$

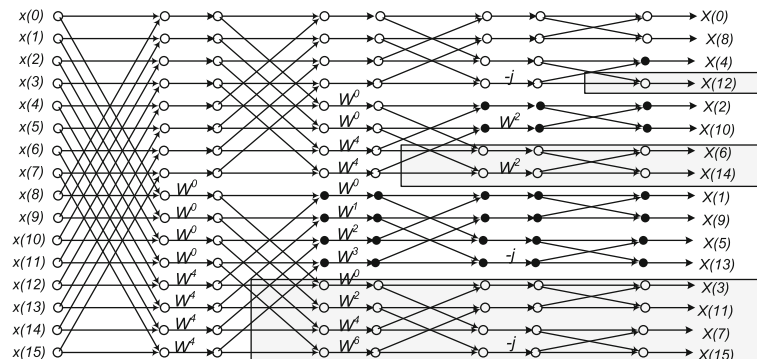
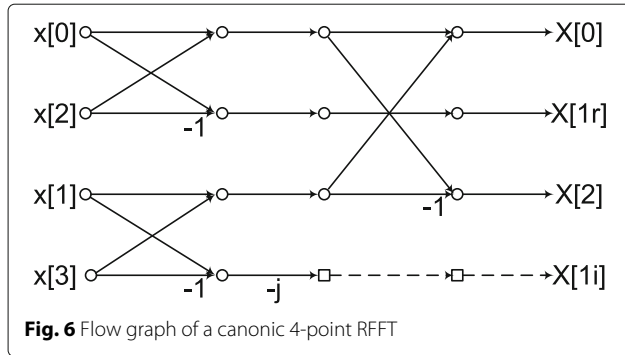


Fig. 5 A canonic 16-point REFFT



Therefore, the W^1 in Fig. 9 should be replaced by $\sqrt{2}$, while the imaginary path is removed. The butterfly simplification can be described in Fig. 10. As a result, the final flow graph is shown in Fig. 11.

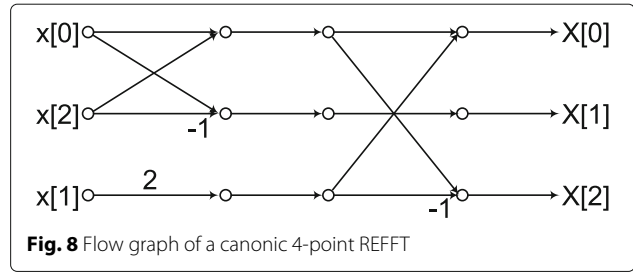
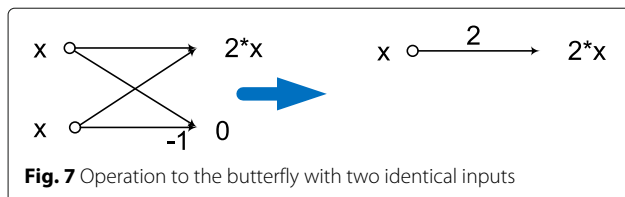
3.3 16-point REFFT

A canonic 16-point RFFT is shown in Fig. 12. In fact, this flow graph is the same as that of Fig. 5 if we separate the real and imaginary signals.

Similarly, the top half of the first three stages can be reduced to the 8-point REFFT as shown in Fig. 11. Furthermore, the last $\frac{N}{4}$ inputs can be removed, as these four signals are redundant, i.e., $x[1] = x[15]$, $x[3] = x[13]$, $x[5] = x[11]$, and $x[7] = x[9]$. In order to study the required operations to eliminate redundancy, we calculate the signal values of the bottom half in Fig. 12, as presented in Table 1.

Since the 9th signal and the 13th signal before the 3rd stage of Fig. 12 have the same value, we can remove the butterfly by using the butterfly simplification as shown in Fig. 7. For the 11th and 15th signals, as the real input and imaginary input of the twiddle factor operation W^2 have the same value, according to Eq. (3), we can also replace the twiddle factor operation W^2 by $\sqrt{2}$, while the imaginary path is removed.

Now, let us consider the remaining signals, i.e., the 10th, 12th, 14th, and 16th signals. We assume $x[1] - x[7] = b$ and $x[3] - x[5] = c$. For simplicity, we consider $W^1 = p - qj$. Then, $W^3 = q - pj$. After calculation, we can get that $\text{Re}[(b + cj) * W^1] = \text{Re}[(c + bj) * W^3] = bp + cq$ and $\text{Im}[(b + cj) * W^1] = -\text{Im}[(c + bj) * W^3] = cp - bq$, respectively. It can be seen that the 2 inputs of the 2nd butterfly operation of the bottom half at the 3rd stage have the same value (i.e., the 10th signal and the 14th signal). Therefore, the butterfly simplification shown in Fig. 7 can be applied



to the butterfly operation to eliminate redundancy. For the butterfly operation whose inputs are the 12th signal and the 16th signal, the operation described in Fig. 13 can be used to reduce the butterfly operation with 2 opposite value inputs to a single datapath. Note that the twiddle factor operation $W^4 = -j$ after the 3rd stage also needs to be moved to the path of the 12th signal. Consequently, the final flow graph is obtained, as shown in Fig. 14.

3.4 Generalization to $N = 2^n$ -point DIF REFFT

In the above sections, we have illustrated that a canonic N -point REFFT can be derived from a canonic $\frac{N}{2}$ -point REFFT. From these examples, according to the regularity of the canonic RFFT flow graph, the proposed method can be summarized in Algorithm 1 from previous sections

Algorithm 1 Generation of a Canonic $N = 2^n$ -Point REFFT

1. For an N -point RFFT, the top half is the same as an $\frac{N}{2}$ -point RFFT. Therefore, this part flow-graph can be simply reduced to the canonic $\frac{N}{2}$ -point REFFT.
2. Since we need to maintain $\frac{N}{2} + 1$ signal values and $\frac{N}{4} + 1$ signals have already been eliminated in the canonic $\frac{N}{2}$ -point REFFT, we need to remove $\frac{N}{4}$ more signals.
3. Since the last quarter of the inputs are redundant, we can remove these $\frac{N}{4}$ signals.
4. Since the last quarter of the inputs are removed, the butterfly operations at the $(n - 1)$ st stage of the bottom half are all removed. If the two inputs of the butterfly have the same value, the operation described in Fig. 7 needs to be performed. If the two inputs of a butterfly operation have the opposite values, the operation described in Fig. 13 needs to be performed. The result is that all the datapaths of the third quarter at the $(n - 1)$ st stage except the $(\frac{5N}{8} + 1)$ st signal become multiplications by 2.
5. The third quarter flow-graph before the $(n - 1)$ st stage is unmodified.
6. When $N \geq 8$, the twiddle factor operation $W_{\frac{N}{8}}^{\frac{5N}{8} + 1}$ after the $(\frac{5N}{8} + 1)$ st signal at the $(n - 1)$ st stage needs to be replaced by $\sqrt{2}$.
7. For the last stage, there are butterfly operations before output pairs $X[k]$ and $X[\frac{N}{2} - k]$, where $0 \leq k \leq \frac{N}{4} - 1$.

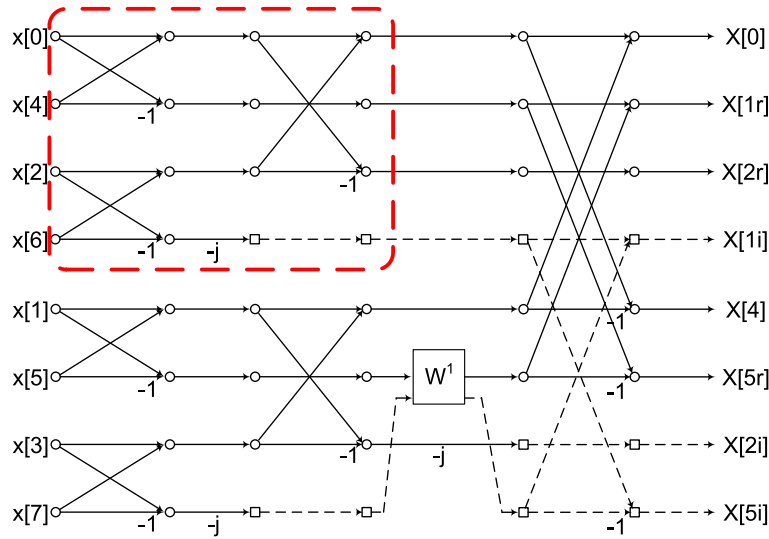


Fig. 9 Flow graph of a canonic 8-point REFFT

(assume we already have the flow graph of a canonic $\frac{N}{2}$ -point REFFT).

Note that the canonic REFFT flow graph can be extended for any $N = 2^n$ -point REFFT recursively. Based on the patterns presented in the above examples and Algorithm 1, given a canonic 32-point REFFT as shown in Fig. 15, a 32-point REFFT is shown in Fig. 16. In this structure, the number of signal values computed at each stage or the output is 17; thus, this structure is canonic.

4 Pre-processing

4.1 Canonic property

In fact, the canonic REFFTs presented above are all obtained from DIF FFTs by twiddle factor transformations as described in [11]. For the canonic REFFTs generated from DIT FFTs, we cannot derive a canonic REFFT directly. For example, we consider the canonic 16-point DIT REFFT as shown in Fig. 17. The first 3 stages of the top half flow graph can also be reduced to the canonic 8-point REFFT as shown in Fig. 11. We calculate the bottom half signal values in Fig. 17 as shown in Table 2. Note that $W^2 = \frac{\sqrt{2}}{2} - \frac{\sqrt{2}}{2}j$. However, in this case, the 10th signal and the 14th signal are neither the same nor opposite. Therefore, we cannot remove this butterfly whose inputs are the 10th signal and the 14th signal by replacing the butterfly operation with a multiplication of 1 input by 2. Furthermore, as the 2 input values are $x[1] - x[7]$ and

$\frac{\sqrt{2}}{2}(x[3] - x[5] - x[7] + x[1])$, respectively, the butterfly operation cannot be reduced to a multiplication with another value. Similarly, the butterfly operation whose inputs are the 12th signal and the 16th signal also cannot be removed. Therefore, the canonic property cannot be achieved, as the number of signals before the 3rd stage will be greater than $\frac{16}{2} + 1 = 9$.

4.2 Pull the twiddle factors

Similar to the twiddle factor transformation as described in Fig. 4, we can perform twiddle factor transformation to turn the 16-point REFFT flow graph in Fig. 17 into the flow graph in Fig. 12. However, the operation will be pulling the twiddle factors to previous stages instead of pushing the twiddle factors to later stages, as shown in Fig. 18. For example, as shown in Fig. 17, we can pull W^1 from after the third stage to before the third stage, which leads to the flow graph as shown in Fig. 12.

According to the work in [11], as the signal values before and after the butterfly operation are both complex, the twiddle factors are free to move. Furthermore, it can be shown that since

$$\begin{aligned}
 X[k] &= \sum_{n=0}^{N-1} x[n] W_N^{nk} \\
 &= \sum_{n=0}^{\frac{N}{2}-1} x[2n] W_N^{2nk} + \sum_{n=0}^{\frac{N}{2}-1} x[2n+1] W_N^{(2n+1)k} \\
 &= \sum_{n=0}^{\frac{N}{2}-1} x[2n] W_N^{2nk} + W_N^k \sum_{n=0}^{\frac{N}{2}-1} x[2n+1] W_N^{2nk} \\
 k &= 0, 1, \dots, N-1,
 \end{aligned} \tag{4}$$



Fig. 10 Butterfly simplification

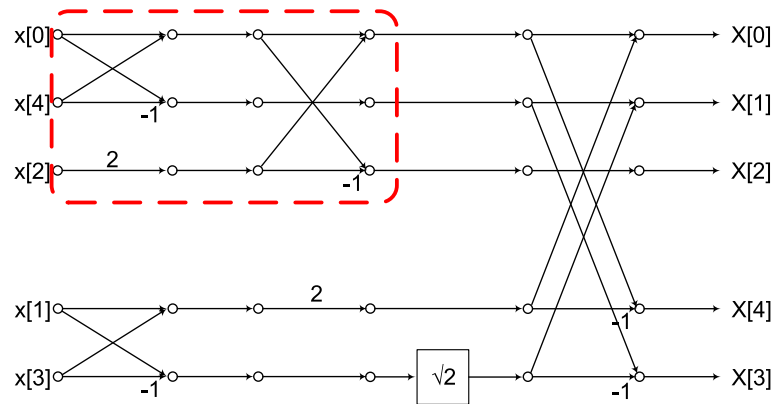


Fig. 11 Flow graph of a canonic 8-point REFFT

the twiddle factors after the $(n - 1)$ st stage at the bottom half will be W_N^k at the path where the output is $X[\frac{N}{2} + k]$. The two output paths of the butterfly operation at the $(n - 1)$ st stage at the bottom half can be expressed as $X[\frac{N}{2} + k]$ and $X[\frac{N}{2} + \frac{N}{4} + k]$, respectively. Thus, the twiddle factors after the $(n - 1)$ st stage always follow the pattern as shown in the left butterfly in Fig. 18 (i.e., W_N^k and $W_N^{k + \frac{N}{4}}$), if the complex butterfly operation has not been removed

in the canonic N -point RFFT. Note that the two twiddle factors will still have the same pattern even if the twiddle factors have already been transformed: as shown in Fig. 19, after transforming W^m , the two twiddle factors after the butterfly can still be $W_N^{k'}$ and $W_N^{k' + \frac{N}{4}}$, if we consider $k' = k - m$.

In conclusion, the goal of the twiddle factor transformation is to make sure the twiddle factor operations before

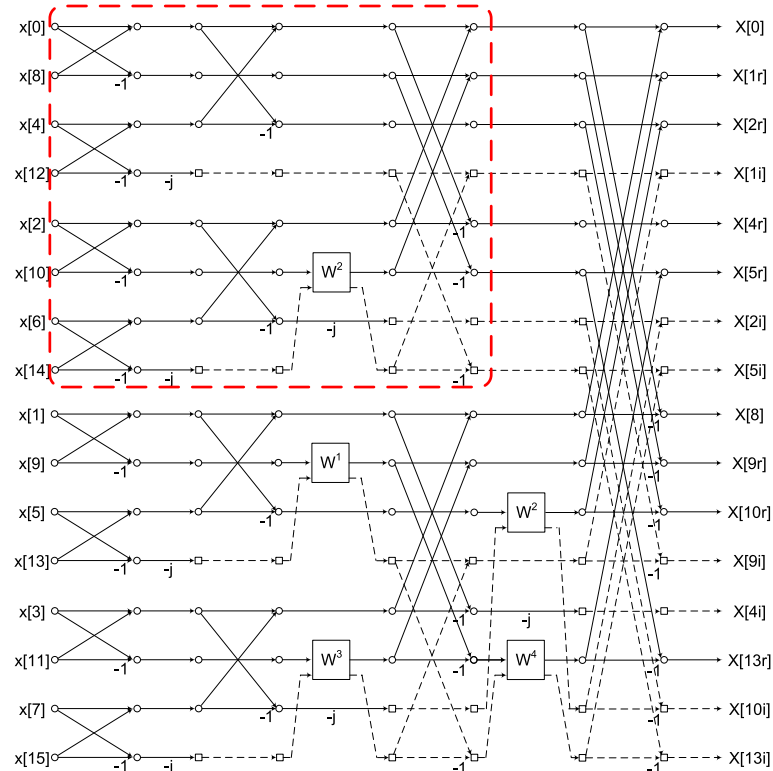


Fig. 12 Flow graph of a canonic 16-point RFFT

Table 1 Signal values of the bottom half in Fig. 12

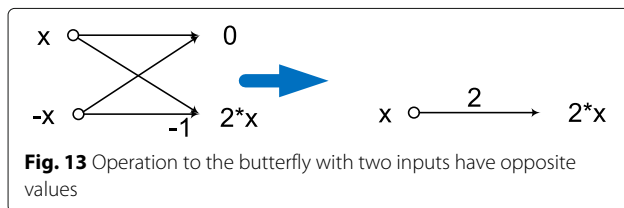
Position	Input	After 1st stage	After 2nd stage	Before 3rd stage
9th	$x[1]$	$x[1] + x[7]$	$x[1] + x[7] + x[5] + x[3]$	$x[1] + x[7] + x[5] + x[3]$
10th	$x[7]$	$x[1] - x[7]$	$x[1] - x[7]$	$\text{Re}[(x[1] - x[7] - (x[5] - x[3]))] * W^1]$
11th	$x[5]$	$x[5] + x[3]$	$x[1] + x[7] - x[5] - x[3]$	$x[1] + x[7] - x[5] - x[3]$
12th	$x[3]$	$x[5] - x[3]$	$(x[5] - x[3])(-j)$	$\text{Im}[(x[1] - x[7] - (x[5] - x[3]))] * W^1]$
13th	$x[3]$	$x[3] + x[5]$	$x[1] + x[7] + x[5] + x[3]$	$x[1] + x[7] + x[5] + x[3]$
14th	$x[5]$	$x[3] - x[5]$	$x[3] - x[5]$	$\text{Re}[(x[3] - x[5] - (x[7] - x[1]))] * W^3]$
15th	$x[7]$	$x[7] + x[1]$	$x[3] + x[5] - x[1] - x[7]$	$(x[3] + x[5] - x[1] - x[7])(-j)$
16th	$x[1]$	$x[7] - x[1]$	$(x[7] - x[1])(-j)$	$\text{Im}[(x[3] - x[5] - (x[7] - x[1]))] * W^3]$

stage n are only $W_N^{\frac{N}{4}}$ or $W_N^{\frac{N}{8}}$ (only the twiddle factor after the $(\frac{7N}{8} + 1)$ st signal at the $(n - 1)$ st stage is $W_N^{\frac{N}{8}}$, which can be replaced by $\sqrt{2}$), when we extend a canonic $\frac{N}{2}$ -point REFFT to a canonic N -point REFFT. If the twiddle factor is $W_N^{\frac{N}{4}} = -j$, the twiddle factor essentially transforms a purely imaginary signal to a purely real signal or transforms a purely real signal to a purely imaginary signal. We know that imaginary signals will equal to 0, as the inputs are even symmetric. Therefore, if the twiddle factor after the butterfly is removed or transformed to $W_N^{\frac{N}{4}}$, then one of the two outputs of the butterfly operation will be 0. In this case, we can eliminate the butterfly operation according to either Fig. 7 or 13.

It can be concluded that twiddle factor transformation is helpful in eliminating butterfly operations, which needs to be applied to the RFFT flow graph before performing the algorithm to generate canonic REFFT.

5 Canonic ROFFT for power-of-two length

In the previous sections, we have presented the approach to generate canonic REFFT. In this section, we present the algorithm to generate canonic ROFFT. As discussed in Section 2, the outputs of the RFFT will be purely imaginary if the inputs are odd symmetric, i.e., $x[k] = -x[N - k]$, where $1 \leq k \leq \frac{N}{2} - 1$. Note that in order to ensure purely imaginary outputs, $x[0]$ and $x[\frac{N}{2}]$ should be equal to 0. Therefore, these two signals can also be removed. As a result, for an N -point ROFFT, a canonic flow graph should only have $\frac{N}{2} - 1$ signal values at each stage. For example, for a canonic 4-point RFFT as shown in Fig. 6, the flow graph for the RFFT when the inputs are odd symmetric only has one signal, as shown in Fig. 20.



When eliminating the redundancies, the difference is that we need to keep imaginary paths, while removing real paths. Therefore, when we extend from $\frac{N}{2}$ -point to N -point, we can choose to remove the third quarter of the inputs instead of the last quarter. The algorithm for generating canonic N -point ROFFT from a canonic $\frac{N}{2}$ -point ROFFT is presented in Algorithm 2. Any $N = 2^n$ -point ROFFT can be derived by using the Algorithm 2.

Algorithm 2 Generation of a Canonic $N = 2^n$ -Point ROFFT

1. For an N -point RFFT, the top half is same as an $\frac{N}{2}$ -point RFFT. Therefore, this part flow-graph can be simply reduced to the canonic $\frac{N}{2}$ -point ROFFT.
2. Since we need to maintain $\frac{N}{2} - 1$ signal values and $\frac{N}{4} - 1$ signals have already been eliminated in the canonic $\frac{N}{2}$ -point ROFFT, we need to remove $\frac{N}{4}$ more signals.
3. Since the third quarter of the inputs are redundant, we can remove these $\frac{N}{4}$ signals.
4. Since the third quarter of the inputs are removed, the butterfly operations at the $(n - 1)$ st stage of the bottom half are all removed. If the two inputs of the butterfly have the same value, the operation described in Fig. 7 needs to be performed. If the two inputs of a butterfly operation have the opposite values, the operation described in Fig. 13 needs to be performed. The result is that all the datapaths of the last quarter at the $(n - 1)$ st stage except the $(\frac{7N}{8} + 1)$ st signal become multiplications of 2.
5. The last quarter flow-graph before the $(n - 1)$ st stage is unmodified.
6. When $N \geq 8$, the twiddle factor operation $W_N^{\frac{N}{8}}$ after the $(\frac{7N}{8} + 1)$ st signal at the $(n - 1)$ st stage needs to be replaced by $\sqrt{2}$.
7. For the last stage, there are butterfly operations before output pairs $X[k]$ and $X[\frac{N}{2} - k]$, where $0 \leq k \leq \frac{N}{4} - 1$.

Given a canonic 16-point RFFT as shown in Fig. 12, according to the Algorithm 2, the flow graph of a canonic 16-point ROFFT is shown in Fig. 21. Note that as discussed above, before performing Algorithm 2, we need to

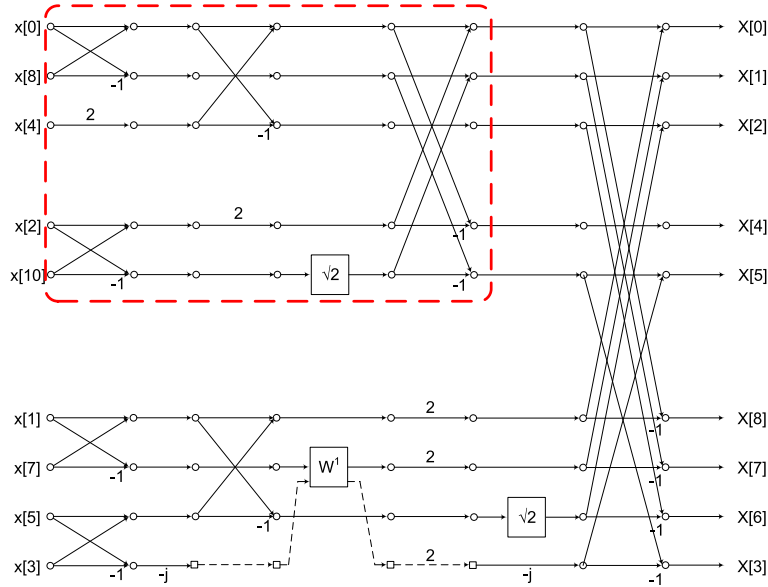


Fig. 14 Flow graph of a canonic 16-point REFFT

pull the twiddle factors from after the $(n - 1)$ st stage to before the $(n - 1)$ st stage if needed.

6 REFFT for any composite length

The algorithm for generating canonic RFFT computation for any composite length has been presented in [23]. In this section, we consider the design of canonic REFFT computation for any composite length. For an N -point REFFT, we need to ensure the number of real samples at each stage is equal to $\lfloor \frac{N}{2} \rfloor + 1$. As shown in Fig. 22a, we should remove $\frac{N-1}{2}$ real signals and keep the other $\frac{N-1}{2}$ real signals and $X[0]$ when N is odd. When N is even as shown in Fig. 22b, we need to remove $\frac{N-2}{2}$ real signals and keep the other $\frac{N-2}{2}$ real signals and $X[0]$ and $X[\frac{N}{2}]$. Consider an N -point REFFT where $N = P \times Q$. To derive the N -point REFFT, we consider the N -point RFFT that constitutes Q P -point RFFTs at the first stage and P Q -point RFFTs at the second stage. We discuss the process for four different cases, i.e., (1) P is odd, Q is odd; (2) P is odd, Q is even; (3) P is even, Q is odd; and (4) P is even, Q is even.

6.1 Subcomponents

If we consider a $P \times Q$ RFFT structure with even symmetric inputs, the inputs of each P -point RFFT at the first stage can be summarized in Table 3.

It can be seen that only the inputs of the first P -point RFFT are even symmetric, as $x_P[k] = x_P[N - k]$. Note that $x_P[k]$ represents the input order in each P -point RFFT. However, for other P -point RFFTs, the inputs are not even symmetric. When Q is even, the inputs of the $(\frac{Q}{2} + 1)$ st

P -point RFFT are $x[kQ + \frac{Q}{2}]$, where $0 \leq k \leq P - 1$, which follow the pattern of $x_P[k] = x_P[P - 1 - k]$.

Moreover, it can also be seen that inputs of the (m) th P -point RFFT and the inputs of $(Q + 2 - m)$ th P -point RFFT are reverse-ordered versions of each other, where $2 \leq m \leq Q$. The relation of the inputs of the two P -point RFFTs can be expressed as

$$x_2[k] = x_1[(-(k + 1))_N]. \quad (5)$$

Note that the actual interval of the inputs of the P -point RFFT is Q . Therefore, according to the DFT time reversal and time shift properties, we can obtain

$$X_2[k] = X_1[(-k)_N] \times W^{-kQ}, \quad (6)$$

which leads to the relation that $X_2[0] = X_1[0]$ and $X_2[k] = X_1[N - k] \times W^{-kQ}$, where $1 \leq k \leq N - 1$. The twiddle factors after the first stage for the (m) th P -point RFFT are $W^{(m-1)k}$, where $1 \leq k \leq P - 1$. As a result, the values after the twiddle factor operations of the (m) th P -point RFFT, $S_1[k]$, and the $(Q + 2 - m)$ th P -point RFFT, $S_2[k]$, can be expressed by

$$S_1[k] = X_1[k] W^{(m-1)k}, \quad (7)$$

$$\begin{aligned} S_2[k] &= X_2[k] W^{(Q-m+1)k} = X_1[(-k)_N] \\ &\quad \times W^{-kQ} W^{(Q-m+1)k} \\ &= X_1[(-k)_N] \times W^{(-m+1)k}, \end{aligned} \quad (8)$$

respectively. We know that the outputs for RFFT are conjugate symmetric:

$$X[k] = X^*[N - k]. \quad (9)$$

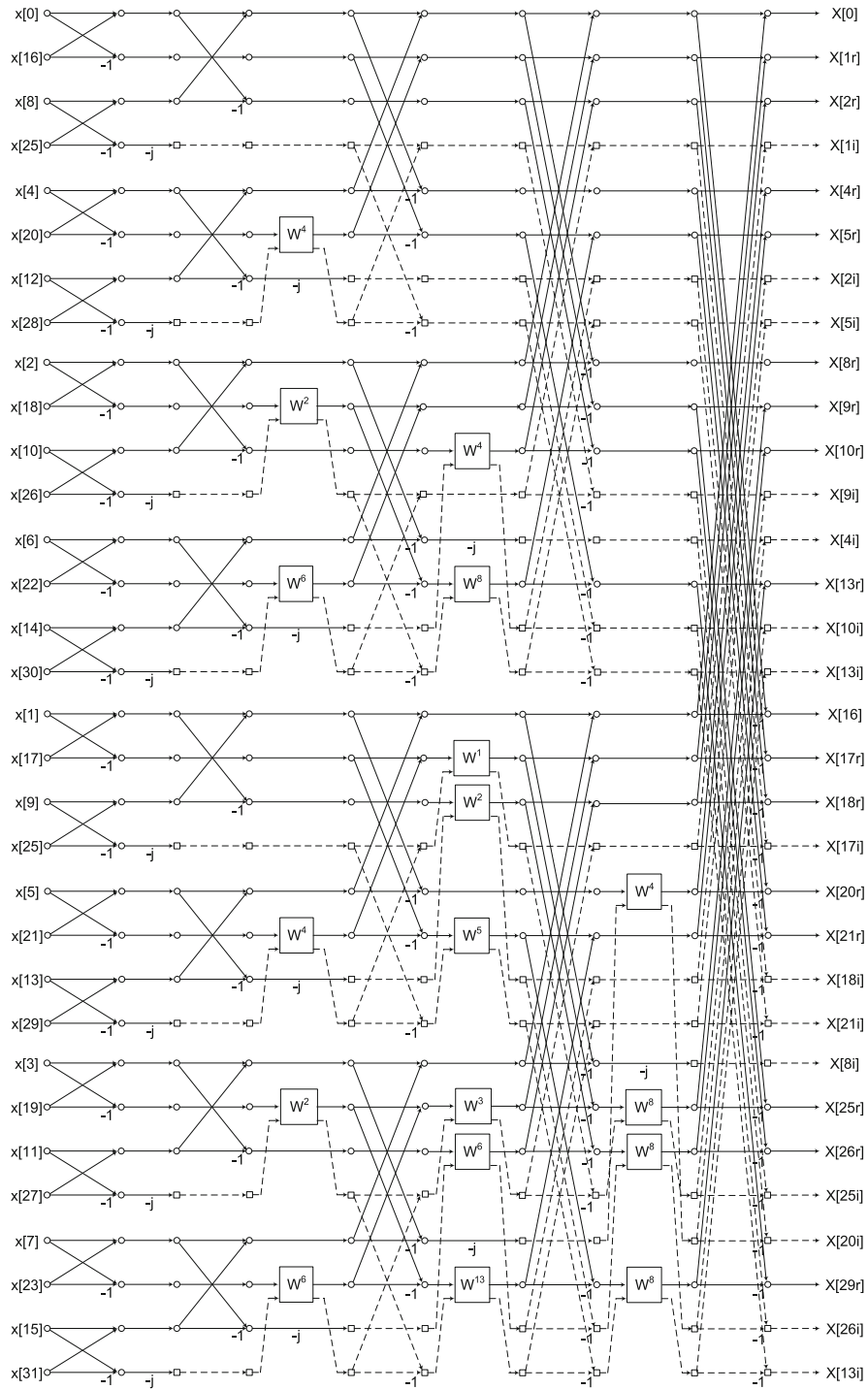


Fig. 15 Flow graph of a canonic 32-point RFFT

Then, for $1 \leq k \leq N - 1$, we have

$$\begin{aligned}
 S_2[k] &= X_1[(-k)_N] \times W^{(-m+1)k} \\
 &= X_1[N - k] \times W^{(-m+1)k} \\
 &= X_1^*[k] \times W^{(-m+1)k}.
 \end{aligned} \tag{10}$$

Therefore, we can conclude that the values of the (m) th P -point RFFT and the $(Q + 2 - m)$ th P -point RFFT after the twiddle factor operations are a conjugate-complex pair:

$$S_1[k] = S_2^*[k]. \tag{11}$$

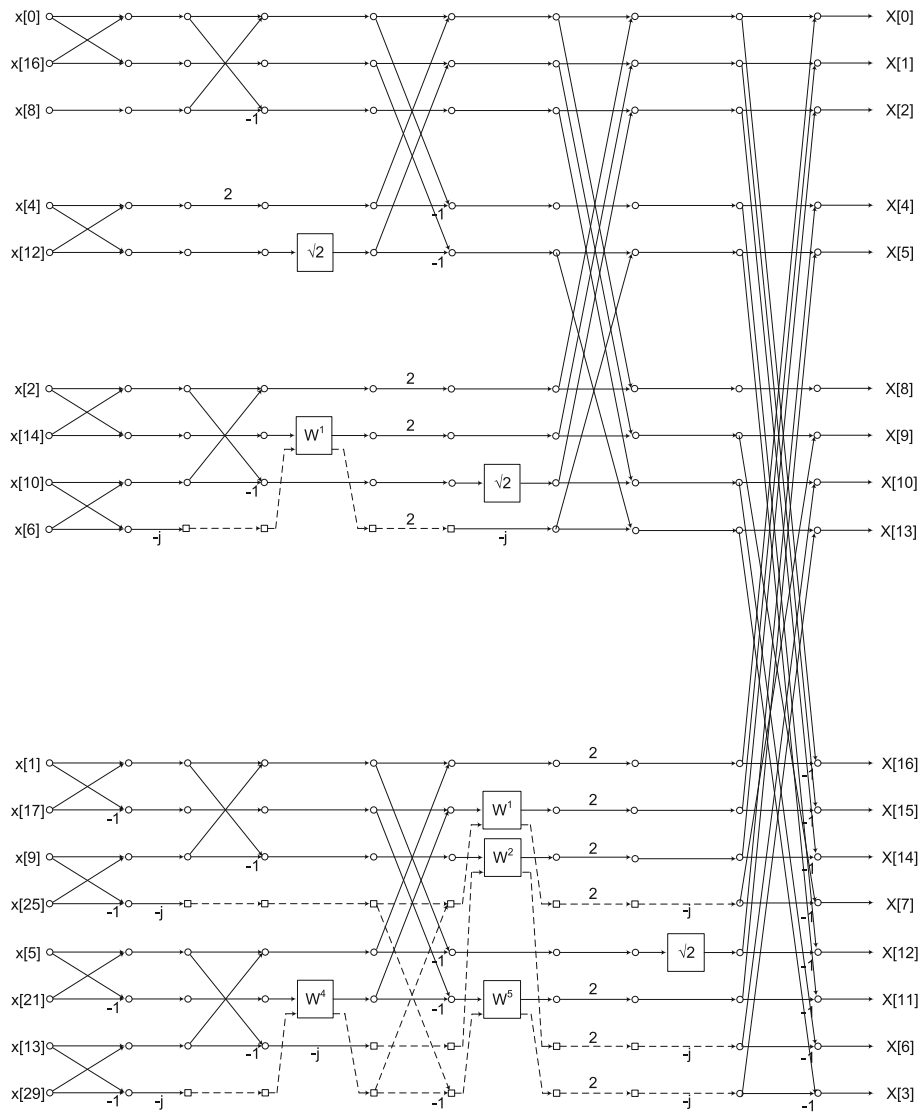


Fig. 16 Flow graph of a canonic 32-point REFFT

Moreover, as $W^{(m-1)k}$ and $W^{(m-1)(N-k)}$ are also a conjugate-complex pair

$$S_1[k] = S_1^*[N - k]. \quad (12)$$

As a result, one of these two P -point REFFT is redundant which can be eliminated, while the outputs of the eliminated P -point REFFT can be obtained by simply conjugating the outputs of the retained P -point REFFT.

Before considering the four cases, we need to consider the designs of the following three FFT dataflows. Note that we only briefly discuss the approaches to remove redundancies of the FFTs with these three input patterns in this paper. Future work will be directed

towards addressing the complete algorithms for generating canonic FFTs with these input patterns.

6.1.1 FFT with Hermitian symmetric inputs (HFFT)

If the inputs of an FFT are Hermitian symmetric, the output will be purely real. We can use the designs of IFFT of Hermitian symmetric signals (RIFFT) such as the work presented in [9] to compute the HFFT. Note that the outputs of the RIFFT need to be reordered to obtain the outputs of the corresponding HFFT. We do not discuss the detailed designs in this paper.

6.1.2 REFFT with odd P and inputs $x_p[k] = x_p[P - 1 - k]$

As we have discussed above, when Q is even, the inputs of the $(\frac{Q}{2} + 1)$ st P -point REFFT are $x[kQ + \frac{Q}{2}]$, where $0 \leq k \leq$

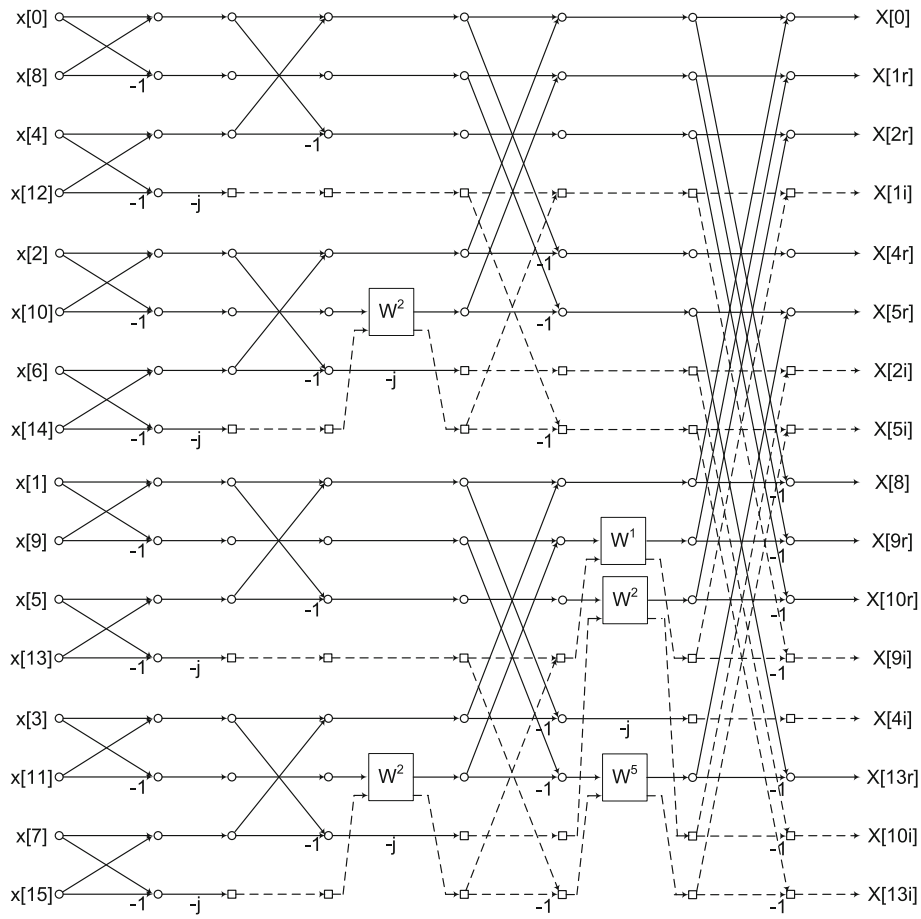


Fig. 17 Flow graph of a canonic 16-point DIT RFFT

$P - 1$, which follow the pattern of $x_p[k] = x_p[P - 1 - k]$ (e.g., $[a, b, c, d, c, b, a]$, when $P = 7$). Furthermore, the outputs of the P -point RFFT connect to twiddle factors $W_N^{\frac{Q}{2}k}$, respectively. We can circularly shift the inputs of an odd size P -point RFFT whose inputs have the pattern of $x_p[k] = x_p[P - 1 - k]$ by $\frac{P-1}{2}Q$ to an odd size REFFT. The circular time shift property can be expressed by

$$x[n] \leftrightarrow X[k], \quad (13)$$

$$x[(n - n_0)_N] \leftrightarrow X[k] W_N^{n_0 k}. \quad (14)$$

Therefore, if we shift the inputs of an odd size P -point RFFT whose inputs have the pattern of $x_p[k] = x_p[P - 1 - k]$ by $\frac{P-1}{2}Q$, the outputs will be $X_P[k] W_N^{\frac{P-1}{2}Qk}$, as the interval of the inputs is Q . If the outputs of

Table 2 Signal values of the bottom half in Fig. 17

Position	Input	After 1st stage	After 2nd stage	Before 3rd stage
9th	$x[1]$	$x[1] + x[7]$	$x[1] + x[7] + x[5] + x[3]$	$x[1] + x[7] + x[5] + x[3]$
10th	$x[7]$	$x[1] - x[7]$	$x[1] - x[7]$	$x[1] - x[7]$
11th	$x[5]$	$x[5] + x[3]$	$x[1] + x[7] - x[5] - x[3]$	$x[1] + x[7] - x[5] - x[3]$
12th	$x[3]$	$x[5] - x[3]$	$(x[5] - x[3])(-j)$	$(x[5] - x[3])(-j)$
13th	$x[3]$	$x[3] + x[5]$	$x[1] + x[7] + x[5] + x[3]$	$x[1] + x[7] + x[5] + x[3]$
14th	$x[5]$	$x[3] - x[5]$	$x[3] - x[5]$	$\frac{\sqrt{2}}{2}(x[3] - x[5] - x[7] + x[1])$
15th	$x[7]$	$x[7] + x[1]$	$x[3] + x[5] - x[1] - x[7]$	$(x[3] + x[5] - x[1] - x[7])(-j)$
16th	$x[1]$	$x[7] - x[1]$	$(x[7] - x[1])(-j)$	$\frac{\sqrt{2}}{2}(-x[3] + x[5] - x[7] + x[1])j$

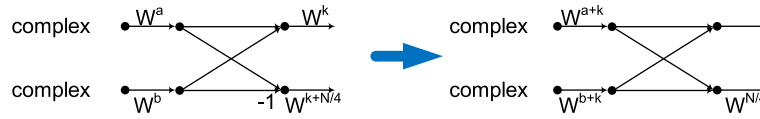


Fig. 18 Twiddle factor transformation: pull

the RFFT connect to twiddle factors $W_N^{\frac{Q}{2}k}$, the values after the twiddle factor operations can be expressed by $X_P[k] W_N^{\frac{P-1}{2}Qk} W_N^{\frac{Q}{2}k} = X_P[k] W_N^{\frac{PQ}{2}k} = X_P[k] (-1)^k$, where $X_P[k]$ here are the outputs of a P -point REFFT. In this case, the values after the twiddle factor operations will be all purely real. The complete operation is shown in Fig. 23. Therefore, we only need to keep $\frac{P+1}{2}$ signals for this P -point RFFT; the deleted $\frac{P-1}{2}$ values after the twiddle factor operation can be obtained by simply alternately negating $X_P[k] (-1)^k$, where $1 \leq k \leq \frac{P-1}{2}$.

6.1.3 RFFT with even P and inputs $x_P[k] = x_P[N-1-k]$

When P and Q are both even, the inputs of the $(\frac{Q}{2} + 1)$ st P -point RFFT also follow the pattern of $x_P[k] = x_P[P-1-k]$ (e.g., $[a, b, c, d, d, c, b, a]$, when $P = 8$), while the outputs also connect to twiddle factors $W_N^{\frac{Q}{2}k}$, respectively. In this case, we can consider a $\frac{P}{2} \times 2$ structure as shown in Fig. 24b. Then, we can pull the twiddle factors before the butterfly operations, as shown in Fig. 24c.

It can be seen from Fig. 24c that the inputs of the two $\frac{P}{2}$ -point RFFTs are reverse-ordered. According to Eq. (6), we can get the relation of the outputs of the two $\frac{P}{2}$ -point RFFTs as below (note the interval of the inputs is $2Q$ in this case):

$$X_2[k] = X_1[(-k)_N] \times W^{-k2Q}, \quad (15)$$

Therefore, the values of the bottom $\frac{P}{2}$ -point RFFT after twiddle factor operation as shown in Fig. 24c are equal to $X_2[k] W^{k\frac{3}{2}Q} = X_1[(-k)_N] \times W^{-k2Q} W^{k\frac{3}{2}Q} = X_1[(-k)_N] W^{-k\frac{Q}{2}}$. Furthermore, according to Eq. (9), we can obtain

$$X_2[k] W^{k\frac{3}{2}Q} = X_1[(-k)_N] W^{-k\frac{Q}{2}} = X_1^*[k] W^{-k\frac{Q}{2}}, \quad (16)$$

which is conjugate of the values of the top $\frac{P}{2}$ -point RFFT after twiddle factor operation as shown in Fig. 24c, i.e., $X_1[k] W^{k\frac{Q}{2}}$. Therefore, the inputs of each butterfly operation as shown Fig. 24c are a conjugate-complex pair. Consequently, we can remove the bottom half of the P -point RFFT to eliminate redundancy as shown in Fig. 24d. The twiddle factor operation $W^{\frac{Q}{2}k}$ needs to be replaced by $2\text{Re}(W^{\frac{Q}{2}k})$, where $1 \leq k \leq \frac{P}{2} - 1$.

6.2 Canonic REFFT generation

In order to generate an N -point canonic REFFT where $N = P \times Q$, we need to make sure there are only $\lfloor \frac{PQ}{2} \rfloor + 1$ signals at each stage.

At the first stage, there are Q P -point RFFTs. The inputs of the first P -point RFFT are even symmetric. Therefore, we only need to keep $\lfloor \frac{P}{2} \rfloor + 1$ outputs. Furthermore, the values of the (m) th P -point RFFT and the $(Q+2-m)$ th P -point RFFT after the twiddle factor operations are a conjugate-complex pair, where $2 \leq m \leq \lfloor \frac{Q+1}{2} \rfloor$. Therefore, we only need to keep half of them. For each P -point RFFT, we use the corresponding canonic REFFT structure, i.e., the number of output signals is equal to P . As a result, if Q is odd, there are $\lfloor \frac{P}{2} \rfloor + 1 + \lfloor \frac{Q-1}{2} \rfloor \times P = \lfloor \frac{PQ}{2} \rfloor + 1$ signals after the first stage, which achieves the canonic property. However, when Q is even, there is one more P -point RFFT, i.e., the $(\frac{Q}{2} + 1)$ st P -point RFFT. The inputs of this RFFT follow the pattern of $x_P[k] = x_P[P-1-k]$ that we can utilize to further eliminate redundancies. Depending on whether P is odd or even, this RFFT either can be transformed to a P -point REFFT as described in Section 6.1.2 or can be reduced to $\frac{P}{2}$ signals as described in Section 6.1.3, respectively. Thus, the total number of signals is $\lfloor \frac{P}{2} \rfloor + 1 + \lfloor \frac{P+1}{2} \rfloor + \frac{Q-2}{2} \times P$ when Q is even, which is equal to $\lfloor \frac{PQ}{2} \rfloor + 1$ as well.

At the second stage, there is one Q -point RFFT whose inputs are the values $X_P[0]$ from the P -point RFFTs at

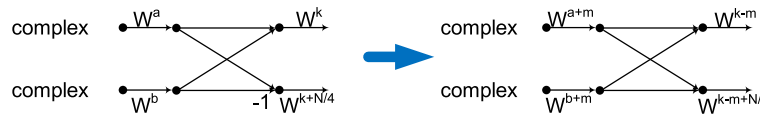
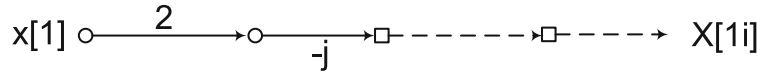


Fig. 19 The two twiddle factors after the $(n-1)$ th stage at the bottom half will still have the same pattern even if the twiddle factors have already been transformed

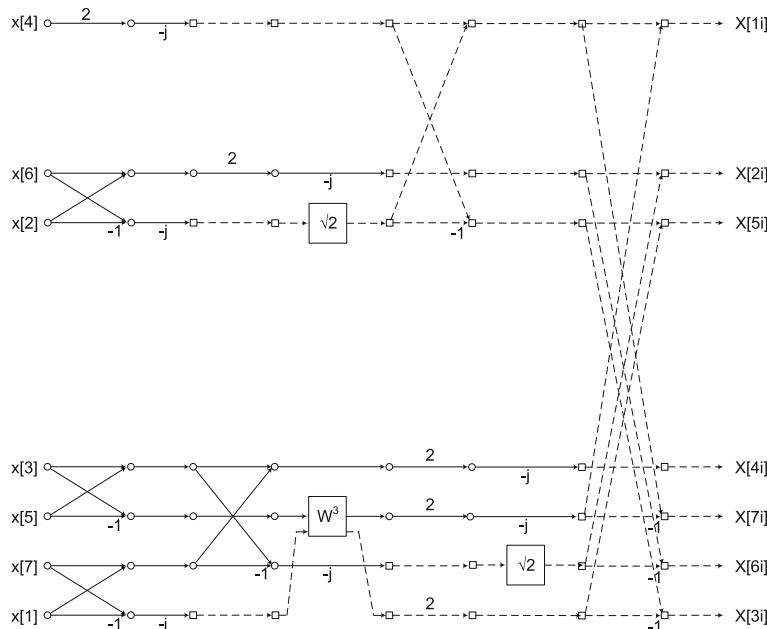
**Fig. 20** Flow graph of a canonic 4-point ROFFT

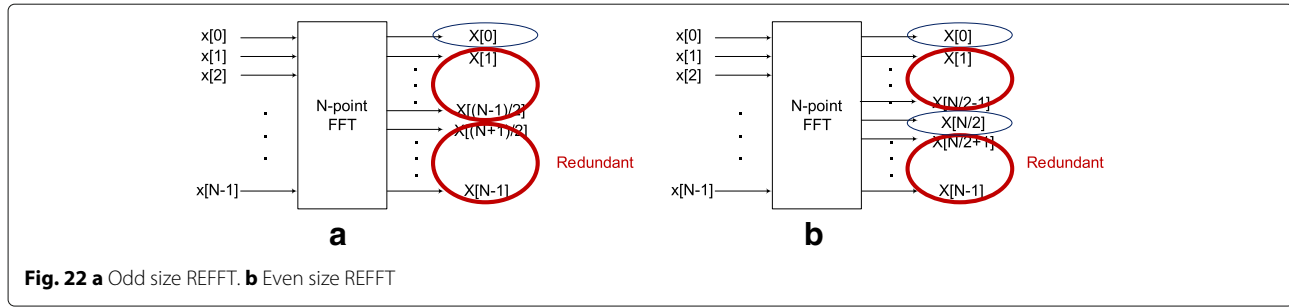
the first stage. Since the outputs $X_P[0]$ from the (m) th P -point RFFT and the inputs of $(Q + 2 - m)$ th P -point RFFT have the same value, the inputs of the first Q -point RFFT at the second stage are also even symmetric. Therefore, it can be reduced to an REFFT with $\lfloor \frac{Q}{2} \rfloor + 1$ signal. Besides, there are $\lfloor \frac{P-1}{2} \rfloor$ Q -point FFTs. Each has inputs $X_P[m]$ after twiddle factor operations from all the P -point RFFTs at the first stage, where $1 \leq m \leq \lfloor \frac{P-1}{2} \rfloor$. Since the values of the (m) th P -point RFFT and the $(Q + 2 - m)$ th P -point RFFT after the twiddle factor operations are a conjugate-complex pair, the inputs of each Q -point FFT are Hermitian symmetric. According to Section 6.1.1, the outputs of these FFTs are purely real. Therefore, we can reduce these Q -point FFTs to HFFTs, which lead to Q signals after each Q -point HFFT. When P is odd, the total number of signals is $\lfloor \frac{Q}{2} \rfloor + 1 + \lfloor \frac{P-1}{2} \rfloor \times Q = \lfloor \frac{PQ}{2} \rfloor + 1$, which is canonic.

However, when P is even, there is one more Q -point RFFT at the second stage, i.e., $(\frac{P}{2} + 1)$ st Q -point RFFT, whose inputs are from $X_P[\frac{P}{2}]$ of each P -point RFFT at the first stage, which are purely real. When P is even and Q is odd, we can circularly shift this Q -point RFFT in frequency to eliminate redundancy according to the

modulation transformation $W_N^{-k_0 n} x[n] \leftrightarrow X[(k - k_0)_N]$, as referred in [23]. Additionally, we have shown that values of the (m) th P -point RFFT and the $(Q + 2 - m)$ th P -point RFFT after the twiddle factor operations are a conjugate-complex pair. Consequently, we can obtain that the $S[\frac{P}{2}]$ values of the (m) th P -point RFFT and the $(Q + 2 - m)$ th P -point RFFT after the twiddle factor operations are the same. Therefore, the inputs of the $(\frac{P}{2} + 1)$ st Q -point RFFT at the second stage are also even symmetric. In conclusion, there are two Q -point REFFTs and $\frac{P-2}{2}$ Q -point HFFTs at the second stage. Thus, the number of signals at the output is equal to $\frac{Q+1}{2} \times 2 + \frac{P-2}{2} \times Q = \frac{PQ}{2} + 1$, which is also canonic with respect to the number of signals.

When P and Q are both even, for the $(\frac{P}{2} + 1)$ st Q -point FFT, we can consider it as a $(2 \times \frac{Q}{2})$ -point FFT, as shown in Fig. 25 [23]. $x[k]$ and $x[k + \frac{Q}{2}]$ go through a butterfly operation first, for $0 \leq k \leq \frac{Q}{2} - 1$. We can perform the operation as shown in Fig. 4 to these butterflies, i.e., push W^k to behind the butterflies. As a result, the top input and the bottom input of the butterfly operation become purely real and purely imaginary, respectively. The bottom output of each butterfly can be eliminated, as it is

**Fig. 21** Flow graph of a canonic 16-point ROFFT



conjugate of the top output. Then, these outputs are processed by one $\frac{Q}{2}$ FFT, as shown in Fig. 26. Note that two real twiddle factor operations at the inputs are transformed to one complex twiddle factor operation at the output for each butterfly of this Q -point FFT. Therefore, we only need to keep $\frac{Q}{2}$ signals for this Q -point FFT in a $P \times Q$ -point RFFT.

Since we have shown in Fig. 26 that the bottom $\frac{Q}{2}$ -point FFT can be deleted, we only need to make sure that the top $\frac{Q}{2}$ -point FFT only involves real datapaths. We consider the flow graph before pushing the twiddle factors, as shown in Fig. 25. The outputs $X[\frac{P}{2}]$ from the first P -point RFFT and the $(\frac{Q}{2} + 1)$ st P -point RFFT at the first stage are purely real and 0, respectively. As a result, the first butterfly in the $(2 \times \frac{Q}{2})$ structure can be reduced to a single datapath, as the bottom input is 0. For the remaining butterflies, the twiddle factors before the two inputs of each butterfly operation can be expressed by $W_{\frac{P}{2}}^k$ and $W_{\frac{PQ}{4} + \frac{P}{2}}^k$, as the outputs $X[\frac{P}{2}]$ from the P -point RFFTs at the first stage are all purely real. Furthermore, we have already proved that the values of the (m) th P -point RFFT and the $(Q + 2 - m)$ th P -point RFFT after the twiddle factor operations are a conjugate-complex pair. Therefore, the (m) th input $x_{\frac{Q}{2}}[m - 1]$ and the $(Q + 2 - m)$ th input $x_{\frac{Q}{2}}[Q - m + 1]$ of the $\frac{Q}{2}$ -point FFT are a conjugate-complex pair. Consequently, the inputs of the $\frac{Q}{2}$ -point are Hermitian symmetric. Thus, each of the remaining butterflies in the $2 \times \frac{Q}{2}$ structure can also be reduced to one single datapath. If $\frac{Q}{2}$ is odd, the $2 \times \frac{Q}{2}$ -point FFT can be reduced to the structure as

shown in Fig. 27, while if $\frac{Q}{2}$ is even, the canonic structure is shown in Fig. 28. In Fig. 28, the twiddle factor $W_{\frac{PQ}{8}}$ is replaced by $\sqrt{2}$ after the input $x[\frac{Q}{4}]$, as twiddle operation can be given by the sum of $W_{\frac{N}{8}}^{\frac{\sqrt{2}}{2}} = \frac{\sqrt{2}}{2} - \frac{\sqrt{2}}{2}j$ and $W_{\frac{N}{8}}^{\frac{5\sqrt{2}}{2}} = j\frac{\sqrt{2}}{2} - \frac{\sqrt{2}}{2}j$. Finally, the canonic REFFT is obtained.

6.3 Examples

6.3.1 $N = P \times Q$, P is odd, Q is odd

For example, we consider the two 15-point canonic RFFTs as shown in Fig. 29 and Fig. 30, respectively. The complex signals are marked bold.

For the 3×5 structure as shown in Fig. 29, we could remove one sample of the first 3-point RFFT at the first stage, since the inputs are even symmetric. Furthermore, we can remove the last two 3-point RFFTs, as they are redundant. As a result, there are $2 + 2 \times 3 = 8$ signals at the first stage. At the second stage, we can remove two samples of the first 5-point RFFT, as its inputs are also even symmetric. The second 5-point FFT can be reduced to HFFT, since the inputs are Hermitian symmetric. Thus, there are $3 + 5 = 8$ signals after the second stage, which is canonic with respect to the number of signals. The final flow graph is shown in Fig. 31.

Similarly, for the 5×3 structure as shown in Fig. 30, we can also design a canonic REFFT as shown in Fig. 32.

6.3.2 $N = P \times Q$, P is odd, Q is even

For example, we consider the $3 \times 2 = 6$ -point canonic RFFT as shown in Fig. 33. The corresponding canonic REFFT is shown in Fig. 34. According to Section 6.1.2, the inputs of the second 3-point RFFT at the first stage can be shifted by 2. As a result, this RFFT can be reduced to the 3-point REFFT, as $x[1] = x[5]$. Note that $(-1)^k$ needs to be added after each output of this 3-point RFFT. It can be seen that there are four signals at each stage in Fig. 34, which is canonic with respect to the number of signals.

6.3.3 $N = P \times Q$, P is even, Q is odd

We can consider another 6-point canonic RFFT as shown in Fig. 35. Note that the second 3-point RFFT at the second stage has been circularly shifted in frequency

Table 3 Inputs of each P -point RFFT at the first stage

	Inputs
1st P -point RFFT	$x[kQ], 0 \leq k \leq P - 1$
2nd P -point RFFT	$x[kQ + 1], 0 \leq k \leq P - 1$
...	...
(m) th P -point RFFT	$x[kQ + m - 1], 0 \leq k \leq P - 1$
...	...
(Q) th P -point RFFT	$x[kQ + Q - 1], 0 \leq k \leq P - 1$

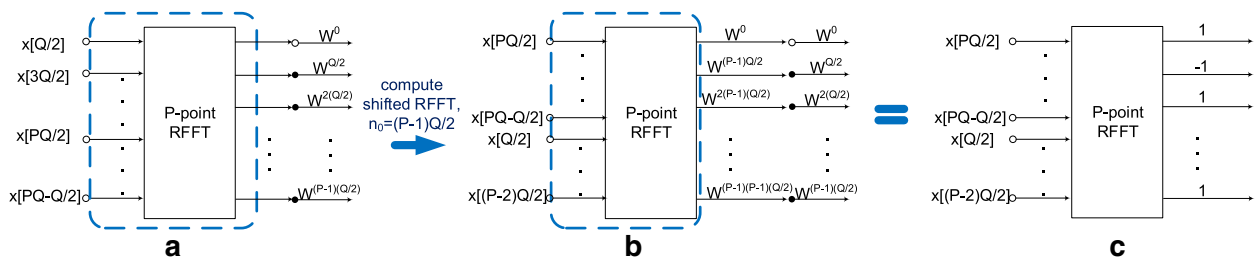


Fig. 23 **a** The original odd size P -point FFT whose inputs have the pattern of $x_p[k] = x_p[P - 1 - k]$. **b** Compute the shifted FFT, $n_0 = \frac{P-1}{2}Q$. **c** The final P -point RFFT whose inputs are even symmetric

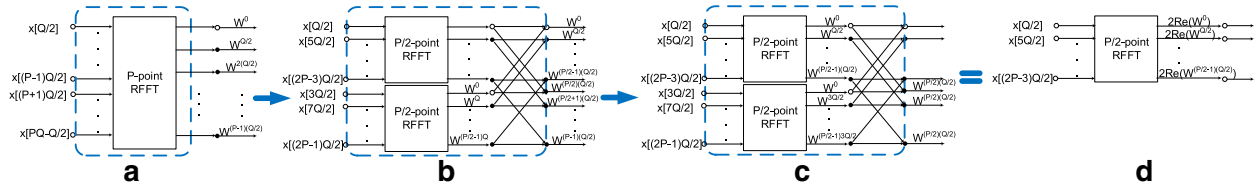


Fig. 24 **a** The original even size P -point FFT whose inputs have the pattern of $x_p[k] = x_p[P - 1 - k]$. **b** The P -point RFFT is considered as a $\frac{P}{2} \times 2$ -point RFFT. **c** Pull the twiddle factors. **d** The final structure which only has $\frac{P}{2}$ signals

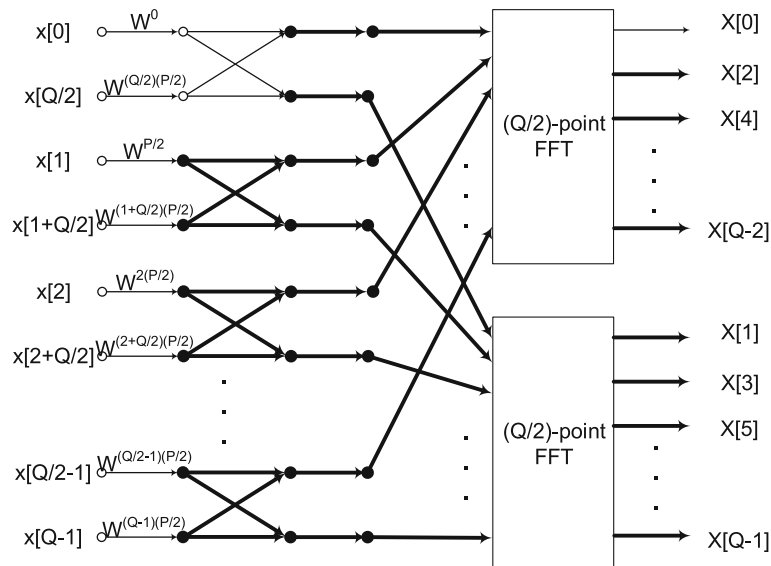


Fig. 25 A Q -point FFT is considered as a $\left(2 \times \frac{Q}{2}\right)$ -point FFT

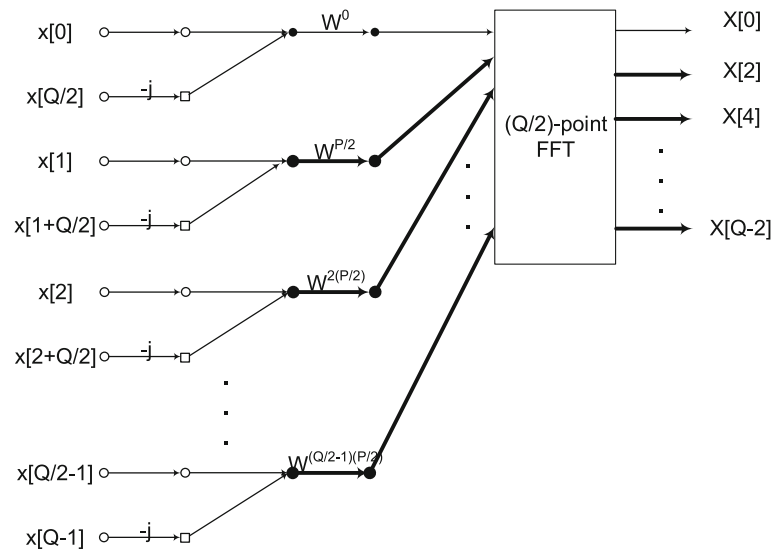


Fig. 26 The $(2 \times \frac{Q}{2})$ -point RFFT after eliminating redundancy

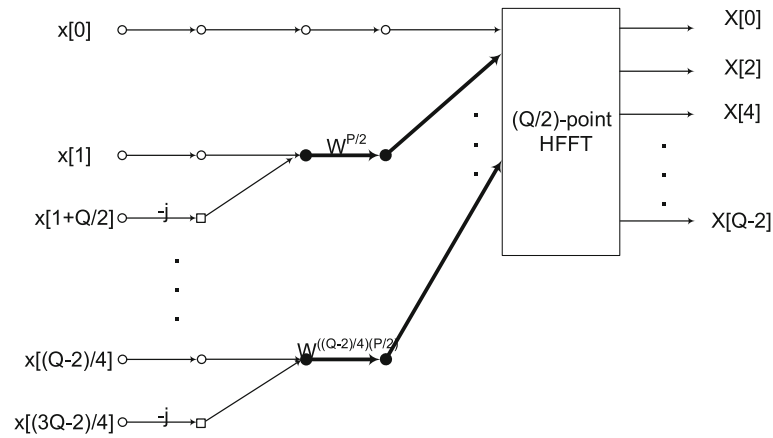


Fig. 27 Remove redundancy of the $(2 \times \frac{Q}{2})$ -point RFFT, when $\frac{Q}{2}$ is odd

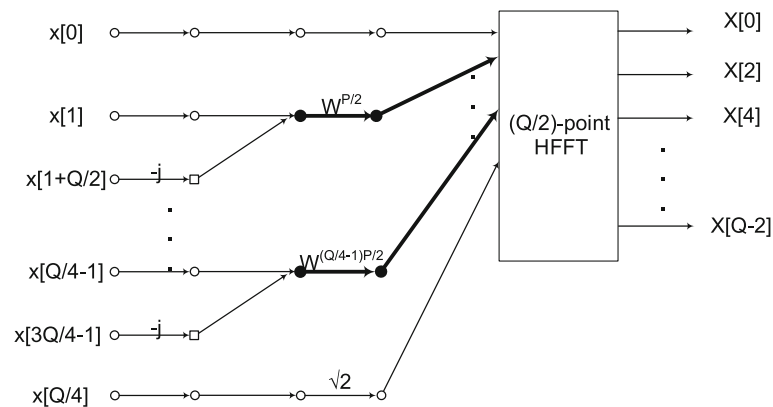


Fig. 28 Remove redundancy of the $(2 \times \frac{Q}{2})$ -point RFFT, when $\frac{Q}{2}$ is even

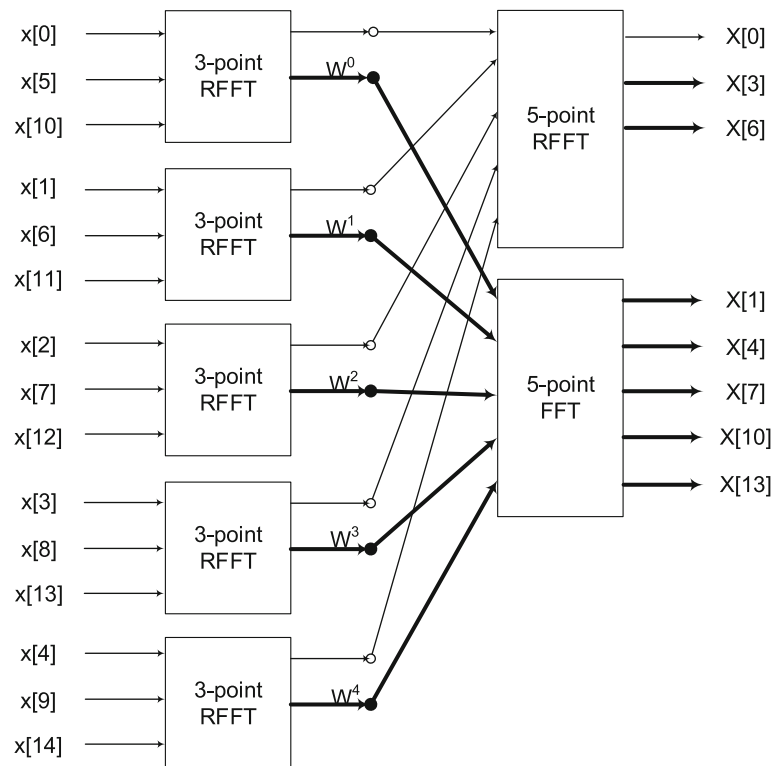


Fig. 29 A 15-point canonic RFFT, where $P=3, Q=5$

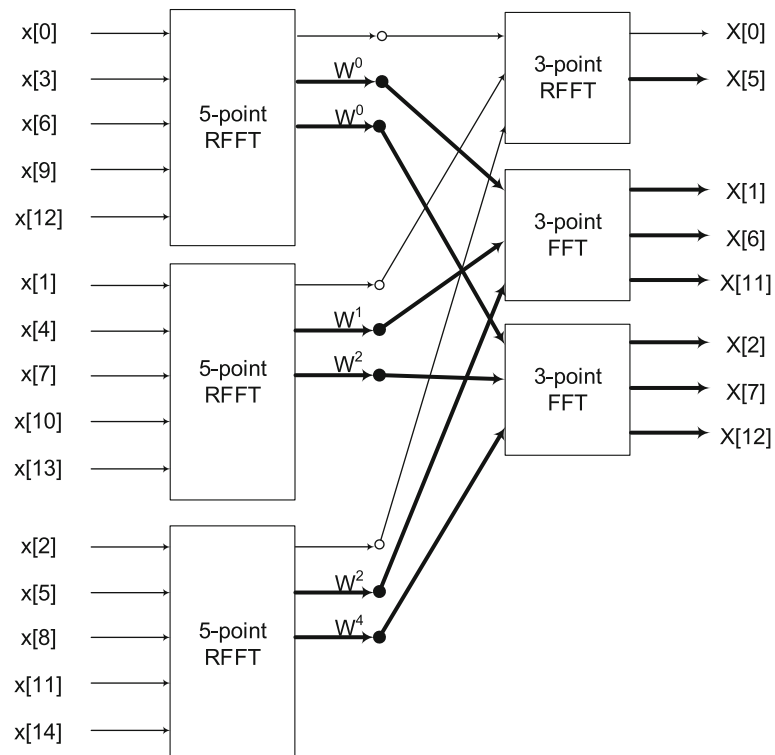


Fig. 30 A 15-point canonic RFFT, where $P=5, Q=3$

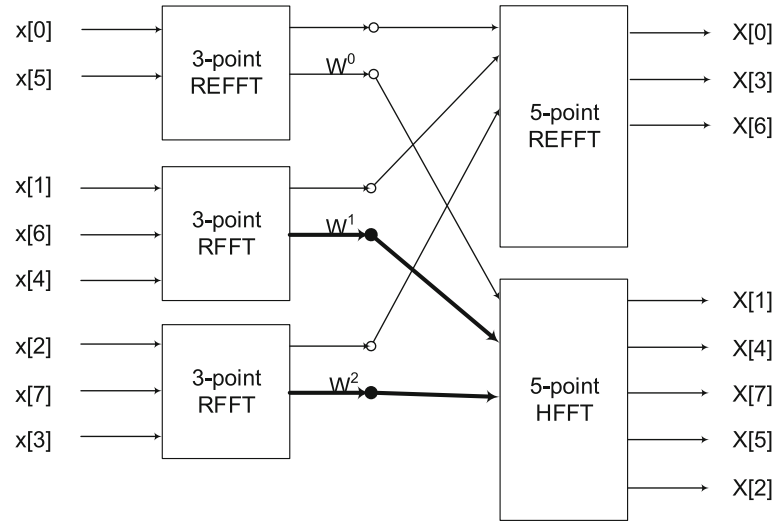


Fig. 31 A 15-point canonic REFFT, where $P = 3, Q = 5$

to eliminate redundancy. The canonic REFFT is shown in Fig. 36. There are also only four signals at each stage.

6.3.4 $N = P \times Q$, P is even, Q is even

All radix- 2^m RFFT structures fall into this category. For example, a radix-4 16-point canonic RFFT is shown in Fig. 37. At the first stage, there is one 4-point REFFT and one 4-point RFFT. For the third 4-point RFFT, the structure can be considered as a 2×2 structure. According to Section 6.1.3, we only need to keep two signals. Therefore, there are nine signals at the first stage in total. At the second stage, the inputs of the first 4-point RFFT are even

symmetric, while the inputs of the second 4-point RFFT are Hermitian symmetric. For the third 4-point FFT, we could also reduce it to the structure only with two signals, based on Fig. 28. The total number of signals at the second stage is also 9. The canonic 16-point REFFT is shown in Fig. 38.

6.4 Summary

Based on the discussion above, we summarize the types of FFTs for the four different cases in Table 4. There are mainly three types of generated subcomponents, i.e., REFFT, RFFT, and HFFT, which is less than the number

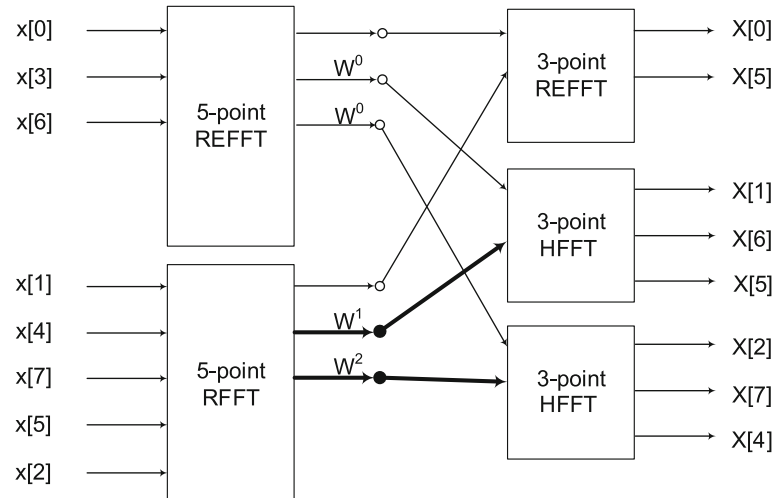


Fig. 32 A 15-point canonic REFFT, where $P = 5, Q = 3$

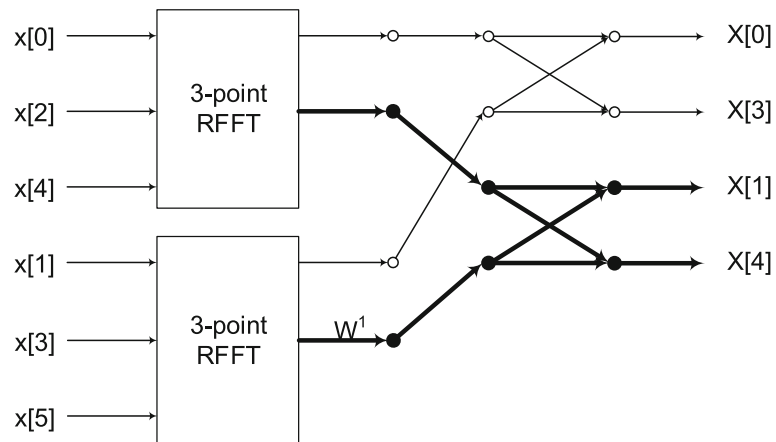


Fig. 33 Flow graph of a canonic 6-point RFFT, where $P = 3, Q = 2$

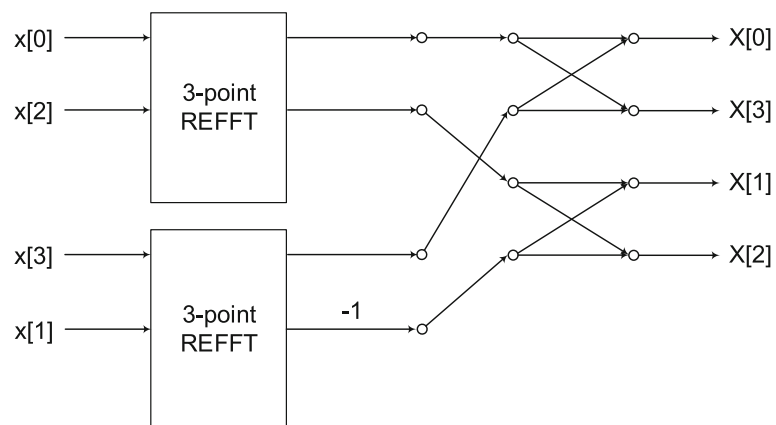


Fig. 34 Flow graph of a canonic 6-point REFFT, where $P = 3, Q = 2$

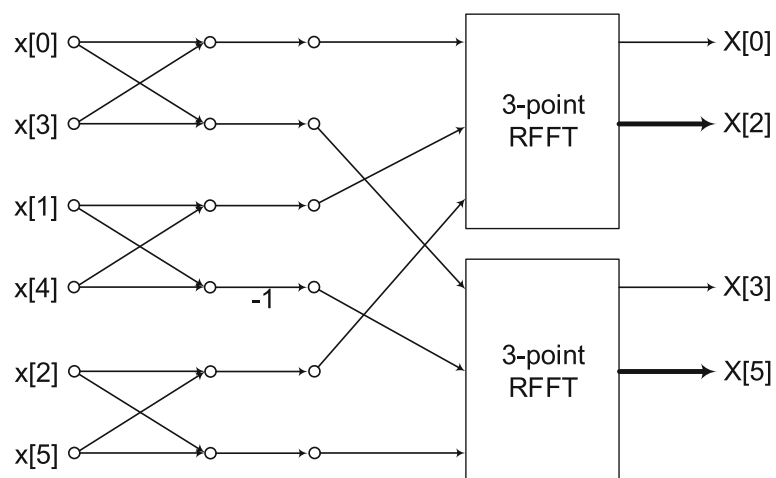


Fig. 35 Flow graph of a canonic 6-point RFFT, where $P = 2, Q = 3$

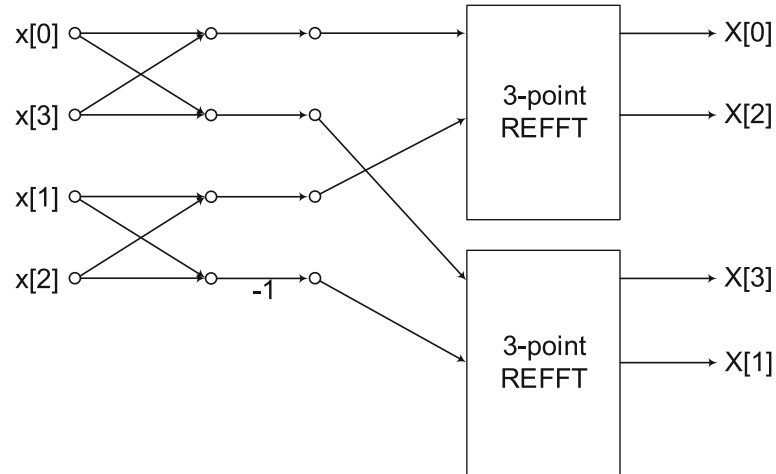


Fig. 36 Flow graph of a canonic 6-point REFFT, where $P = 2, Q = 3$

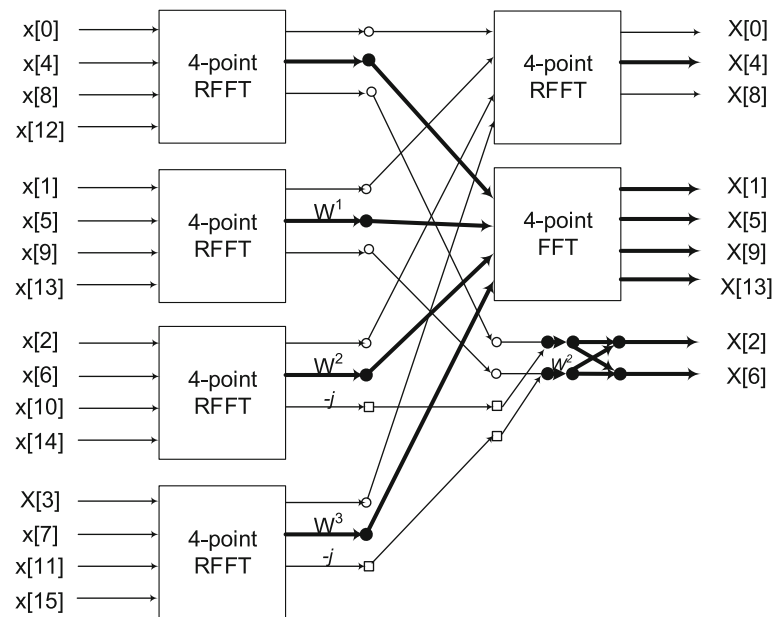


Fig. 37 A 16-point canonic RFFT, where $P = 4, Q = 4$

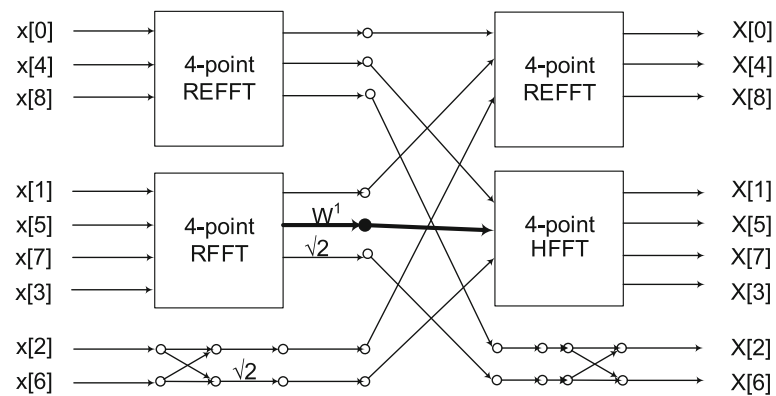


Fig. 38 A 16-point canonic REFFT, where $P = 4, Q = 4$

Table 4 FFT types for canonic REFFT

	<i>P</i> is odd, <i>Q</i> is odd		<i>P</i> is odd, <i>Q</i> is even		<i>P</i> is even, <i>Q</i> is odd		<i>P</i> is even, <i>Q</i> is even	
	<i>P</i> -point	<i>Q</i> -point	<i>P</i> -point	<i>Q</i> -point	<i>P</i> -point	<i>Q</i> -point	<i>P</i> -point	<i>Q</i> -point
#REFFT	1	1	2	1	1	2	1	1
#RFFT	$\frac{Q-1}{2}$	0	$\frac{Q-2}{2}$	0	$\frac{Q-1}{2}$	0	$\frac{Q-2}{2}$	0
#HFFT	0	$\frac{P-1}{2}$	0	$\frac{P-1}{2}$	0	$\frac{P-2}{2}$	0	$\frac{P-2}{2}$
$\# \left(\frac{P}{2} \times 2 \right)$	0	0	0	0	0	0	1	0
$\# \left(2 \times \frac{Q}{2} \right)$	0	0	0	0	0	0	0	1

of types of subcomponents in the dataflow derived algebraically in [14]. Any composite length canonic REFFTs can be obtained by applying the proposed methods for $P \times Q$ decomposition iteratively.

The canonic ROFFT for any composite size can be obtained similarly by following these steps described in this section. We do not discuss these designs in detail in this paper due to lack of space. The only difference is that we need to make sure there are only $\lceil \frac{N}{2} \rceil - 1$ signals for an N -point ROFFT instead of $\lfloor \frac{N}{2} \rfloor + 1$.

7 Performance

In this section, we discuss the performances of the canonic REFFT/ROFFT.

There are less signals in the canonic REFFT/ROFFT, compared to FFT, RFFT, or canonic RFFT, as we remove the redundant inputs from the beginning. Furthermore, the number of butterfly operations in the REFFT/ROFFT flow graph is also reduced, as we remove the butterfly operation if the two inputs of the butterfly operation have the same value or opposite values, as described in Fig. 7 or Fig. 13, respectively. Consequently, the number of twiddle factor operation is also reduced for a power-of-two size RFFT, as one quarter of the datapaths are eliminated when we extend a canonic N -point REFFT/ROFFT from a canonic $\frac{N}{2}$ -point REFFT/ROFFT. Moreover, from the third stage to the last stage, there is one twiddle factor $W_N^{\frac{N}{8}}$ before the stage is replaced by a multiplication by $\sqrt{2}$. Thus, for an $N = 2^n$ -point RFFT, when $n \geq 2$, there will

be $n - 2$ multiplications of $\sqrt{2}$ in the flow graph. Note that we do not consider the multiplications of 2 in the flow graph which are generated by the operations of Fig. 7 and 13 as multipliers, since these only involve 1-bit left shift.

Table 5 compares the performance of the proposed canonic REFFT/ROFFT with FFT, RFFT, and canonic RFFT. Note that we consider a complex butterfly operation as two real butterfly operations.

It can be seen that the proposed canonic REFFTs/ROFFTs have less signals, less butterfly operations, and less twiddle factor operations. Due to the fact that the canonic ROFFT has less signal values at each stage compared to canonic REFFT, the canonic ROFFT also requires less butterfly operations.

8 Conclusions

This paper has proposed novel algorithms to design canonic FFT flow graphs when the inputs are real and even/odd symmetric. A canonic N -point REFFT/ROFFT can be extended from a canonic $\frac{N}{2}$ -point REFFT/ROFFT. Twiddle factor transformations are needed if there are twiddle factors other than $W_N^{\frac{N}{4}}$ and $W_N^{\frac{N}{8}}$ before the last stage. The design of canonic REFFT for any composite length has also been presented. Future work will be directed towards designing efficient architectures for any composite length RFFTs with real-valued even/odd symmetric inputs based on the canonic dataflow developed in this paper.

Table 5 Performance comparison for $N = 2^n$ -point FFT algorithms

FFT algorithm	#Signal values at each stage	#Real butterfly operation	#Twiddle factor operations
Complex FFT	$2N$	$N \log_2 N$	$n \times 2^{n-1} - 2^n + 2$
[6] DIF RFFT	$\geq N$	$\left(\frac{N}{2} - 1 \right) \log_2 N + 1$	$\left(n - \frac{7}{2} \right) \times 2^{n-2} + n - 1$
[11] Canonic DIT RFFT	N	$\frac{N}{2} \log_2 N - \frac{N}{2} + 1$	$(n - 3) \times 2^{n-2} + 1$
[11] Canonic DIF RFFT	N	$\frac{N}{2} \log_2 N - \frac{N}{2} + 1$	$(n - 4) \times 2^{n-2} + n$
Canonic REFFT	$\frac{N}{2} + 1$	$\left(\frac{N}{4} + 1 \right) \log_2 N - \frac{N}{2}$	$2^{n-2} - n + 1^a$
Canonic ROFFT	$\frac{N}{2} - 1$	$\frac{N}{4} \log_2 N - \frac{N}{2}$	$2^{n-2} - n + 1^a$

^aWhen $n \geq 2$, there will also be $n - 2$ multiplications of $\sqrt{2}$ in the flow graph

Acknowledgements

KP thanks Dr. Maureen P. Quirk for suggesting him to work on this problem at the 2015 IEEE ICASSP conference.

Competing interests

The authors declare that they have no competing interests.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Received: 22 November 2016 Accepted: 12 May 2017

Published online: 14 June 2017

References

1. HV Sorensen, DL Jones, M Heideman, CS Burrus, Real-valued fast Fourier transform algorithms. *IEEE Trans. Acoustics Speech Signal Process.* **35**(6), 849–863 (1987)
2. H-F Chi, Z-H Lai, in *IEEE International Symposium on Circuits and Systems (ISCAS)*. A cost-effective memory-based real-valued FFT and Hermitian symmetric IFFT processor for DMT-based wire-line transmission systems (IEEE, Kobe, 2005), pp. 6006–6009
3. M Ayinala, Y Lao, KK Parhi, An in-place FFT architecture for real-valued signals. *IEEE Trans. Circ. Syst. II: Express Briefs.* **60**(10), 652–656 (2013)
4. Y Voronenko, M Puschel, Algebraic signal processing theory: Cooley–Tukey type algorithms for real DFTs. *IEEE Trans. Signal Process.* **57**(1), 205–222 (2009)
5. M Garrido, KK Parhi, J Grajal, A pipelined FFT architecture for real-valued signals. *IEEE Trans. Circ. Syst. I: Regular Papers.* **56**(12), 2634–2643 (2009)
6. M Ayinala, M Brown, KK Parhi, Pipelined parallel FFT architectures via folding transformation. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **20**(6), 1068–1081 (2012)
7. C Cheng, KK Parhi, High-throughput VLSI architecture for FFT computation. *IEEE Trans. Circ. Syst. II: Express Briefs.* **54**(10), 863–867 (2007)
8. M Garrido, J Grajal, M Sánchez, O Gustafsson, Pipelined radix- 2^k feedforward FFT architectures. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **21**(1), 23–32 (2013)
9. SA Salehi, R Amirfattahi, KK Parhi, Pipelined architectures for real-valued FFT and Hermitian-symmetric IFFT with real datapaths. *IEEE Trans. Circ. Syst. II: Express Briefs.* **60**(8), 507–511 (2013)
10. M Ayinala, KK Parhi, FFT architectures for real-valued signals based on radix- 2^3 and radix- 2^4 algorithms. *IEEE Trans. Circ. Syst. I: Regular Papers.* **60**(9), 2422–2430 (2013)
11. M Parhi, Y Lao, KK Parhi, in *48th Asilomar Conference on Signals, Systems and Computers*. Canonic real-valued FFT structures (IEEE, Pacific Grove, 2014), pp. 1261–1265
12. P Duhamel, Implementation of “split-radix” FFT algorithms for complex, real, and real-symmetric data. *IEEE Trans. Acoustics Speech Signal Process.* **34**(2), 285–295 (1986)
13. SA Martucci, Symmetric convolution and the discrete sine and cosine transforms. *IEEE Trans. Signal Process.* **42**(5), 1038–1051 (1994)
14. M Puschel, JM Moura, Algebraic signal processing theory: Cooley–Tukey type algorithms for DCTs and DSTs. *IEEE Trans. Signal Process.* **56**(4), 1502–1521 (2008)
15. J Astola, D Akopian, Architecture-oriented regular algorithms for discrete sine and cosine transforms. *IEEE Trans. Signal Process.* **47**(4), 1109–1124 (1999)
16. X Shao, SG Johnson, Type-II/III DCT/DST algorithms with reduced number of arithmetic operations. *Signal Process.* **88**(6), 1553–1564 (2008)
17. AV Oppenheim, RW Schaffer, *Discrete-time signal processing*. (Pearson Higher Education, Upper Saddle River, 2009)
18. JW Cooley, JW Tukey, An algorithm for the machine calculation of complex Fourier series. *Math. Comput.* **19**(90), 297–301 (1965)
19. RC Singleton, An algorithm for computing the mixed radix fast Fourier transform. *IEEE Trans. Audio Electroacoustics.* **17**(2), 93–103 (1969)
20. GD Bergland, A fast Fourier transform algorithm using base 8 iterations. *Math. Comput.* **22**(102), 275–279 (1968)
21. P Duhamel, H Hollmann, Split radix FFT algorithm. *Electron. Lett.* **20**(1), 14–16 (1984)
22. S He, M Torkelson, in *Proceedings of the Custom Integrated Circuits Conference*. Design and implementation of a 1024-point pipeline FFT processor (IEEE, Santa Clara, 1998), pp. 131–134
23. Y Lao, KK Parhi, in *Proceedings of IEEE Workshop on Signal Processing Systems. Data-canonic real FFT flow-graphs for composite length* (IEEE, Dallas, 2016), pp. 189–194

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► springeropen.com