4.6 Decimation-in-Frequency Algorithm

DIT algorithm is based on the decomposition of the DFT computation by forming smaller and smaller subsequences of the sequence $x(n)$. In DIF algorithm the output sequence $X(k)$ is divided into smaller and smaller subsequences. In this algorithm the input sequence $x(n)$ is partitioned into two sequences each of length $\frac{N}{2}$ samples. The first sequence $x_1(n)$ consists of first $\frac{N}{2}$ samples of $x(n)$ and the second sequence $x_2(n)$ consists of the last $\frac{N}{2}$ samples of $x(n)$ i.e.,

$$x_1(n) = x(n), \quad n = 0, 1, 2, \ldots N/2 - 1 \tag{4.23}$$

$$x_2(n) = x\,(n + N/2) \quad n = 0, 1, 2, \ldots N/2 - 1 \tag{4.24}$$

If $N = 8$ the first sequence $x_1(n)$ has values for $0 \leq n \leq 3$ and $x_2(n)$ has values for $4 \leq n \leq 7$.

The N-point DFT of $x(n)$ can be written as

$$X(k) = \sum_{n=0}^{\frac{N}{2}-1} x(n)W_N^{nk} + \sum_{n=\frac{N}{2}}^{N-1} x(n)W_N^{nk}$$

$$= \sum_{n=0}^{\frac{N}{2}-1} x_1(n)W_N^{nk} + \sum_{n=0}^{\frac{N}{2}-1} x_2(n)W_N^{(n+N/2)k}$$

$$= \sum_{n=0}^{\frac{N}{2}-1} x_1(n)W_N^{nk} + W_N^{Nk/2} \sum_{n=0}^{\frac{N}{2}-1} x_2(n)W_N^{nk}$$

$$= \sum_{n=0}^{\frac{N}{2}-1} x_1(n) W_N^{nk} + e^{-j\pi k} \sum_{n=0}^{\frac{N}{2}-1} x_2(n) W_N^{nk}$$

when k is even $e^{-j\pi k} = 1$

$$X(2k) = \sum_{n=0}^{\frac{N}{2}-1} [x_1(n) + x_2(n)] W_N^{2nk}$$

$$= \sum_{n=0}^{\frac{N}{2}-1} [x_1(n) + x_2(n)] W_{N/2}^{nk}$$

$$= \sum_{n=0}^{\frac{N}{2}-1} f(n) W_{N/2}^{nk} \qquad (4.25a)$$

where

$$f(n) = x_1(n) + x_2(n) \qquad (4.25b)$$

Eq.(4.25a) is the $\frac{N}{2}$-point DFT of the $\frac{N}{2}$-point sequence $f(n)$ obtained by adding the first-half and the last-half of the input sequence. When k is odd

$$e^{-j\pi k} = -1$$

$$X(2k+1) = \sum_{n=0}^{\frac{N}{2}-1} [x_1(n) - x_2(n)] W_N^{(2k+1)n}$$

$$= \sum_{n=0}^{\frac{N}{2}-1} [x_1(n) - x_2(n)] W_N^n W_{N/2}^{nk}$$

$$= \sum_{n=0}^{\frac{N}{2}-1} g(n) W_{N/2}^{nk} \qquad (4.26)$$

where

$$g(n) = [x_1(n) - x_2(n)] W_N^n \qquad (4.27)$$

Eq.(4.26) is the $\frac{N}{2}$-point DFT of the sequence $g(n)$ obtained by subtracting the second half of the input sequence from the first half and then multiplying the resulting sequence with W_N^n.

From Eq.(4.25a) and Eq.(4.26) we find that the even and odd samples of the DFT can be obtained from the $\frac{N}{2}$-point DFTs of $f(n)$ and $g(n)$ respectively.

The Eq.(4.25b) and Eq.(4.27) can be represented by a butterfly as shown in Fig. 4.10. This is the basic operation of DIF algorithm.

From Eq.(4.25), for $N = 8$, we have

$$X(0) = \sum_{n=0}^{3} [x_1(n) + x_2(n)] = \sum_{n=0}^{3} f(n) = f(0) + f(1) + f(2) + f(3) \qquad (4.28)$$

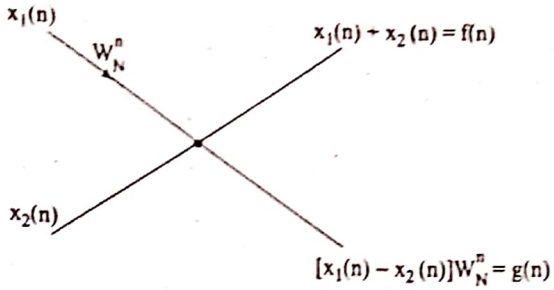

Fig. 4.10 Flow graph of basic butterfly diagram for DIF algorithm

$$X(2) = \sum_{n=0}^{3}[x_1(n) + x_2(n)]W_8^{2n} = \sum_{n=0}^{3} f(n)W_8^{2n}$$

$$= f(0) + f(1)W_8^2 - f(2) - f(3)W_8^2 \qquad \boxed{W_8^4 = (e^{j2\pi/8})^4 = e^{j\pi} = -1;\quad W_8^8 = (e^{j2\pi/8})^8 = e^{j2\pi} = 1} \qquad (4.29)$$

$$X(4) = \sum_{n=0}^{3}[x_1(n) + x_2(n)]W_8^{4n} = \sum_{n=0}^{3} f(n)W_8^{4n} = \sum_{n=0}^{3} f(n)(-1)^n$$

$$= f(0) - f(1) + f(2) - f(3) \qquad (4.30)$$

$$X(6) = \sum_{n=0}^{3}[x_1(n) + x_2(n)]W_8^{6n} = \sum_{n=0}^{3} f(n)(-W_8^2)^n$$

$$= f(0) - f(1)W_8^2 - f(2) + f(3)W_8^2 \qquad (4.31)$$

From Eq. (4.26) we have

$$X(1) = \sum_{n=0}^{3}[x_1(n) - x_2(n)]W_8^n = \sum_{n=0}^{3} g(n) = g(0) + g(1) + g(2) + g(3) \qquad (4.32)$$

$$X(3) = \sum_{n=0}^{3}[x_1(n) - x_2(n)]W_8^{3n} = \sum_{n=0}^{3} g(n)W_8^{2n}$$

$$= g(0) + g(1)W_8^2 - g(2) - g(3)W_8^2 \qquad (4.33)$$

$$X(5) = \sum_{n=0}^{3}[x_1(n) - x_2(n)]W_8^{5n} = \sum_{n=0}^{3} g(n)W_8^{4n} = \sum_{n=0}^{3} g(n)(-1)^n$$

$$= g(0) - g(1) + g(2) - g(3) \qquad (4.34)$$

$$X(7) = \sum_{n=0}^{3}[x_1(n) - x_2(n)]W_8^{7n} = \sum_{n=0}^{3} g(n)(-W_8^2)^n$$

$$= g(0) - g(1)W_8^2 - g(2) + g(3)W_8^2 \qquad (4.35)$$

We have seen that the even-indexed samples of $X(k)$ can be obtained from the 4-point DFT of the sequence $f(n)$ where

$$f(n) = x_1(n) + x_2(n) \quad n = 0, 1, \ldots \frac{N}{2} - 1$$

i.e.,
$$\begin{aligned} f(0) &= x_1(0) + x_2(0) \\ f(1) &= x_1(1) + x_2(1) \\ f(2) &= x_1(2) + x_2(2) \\ f(3) &= x_1(3) + x_2(3) \end{aligned} \tag{4.36}$$

The odd-indexed samples of $X(k)$ can be obtained from the 4-point DFT of the sequence $g(n)$ where $g(n) = [x_1(n) - x_2(n)]W_8^n$

i.e.,
$$\begin{aligned} g(0) &= [x_1(0) - x_2(0)]W_8^0 \\ g(1) &= [x_1(1) - x_2(1)]W_8^1 \\ g(2) &= [x_1(2) - x_2(2)]W_8^2 \\ g(3) &= [x_1(3) - x_2(3)]W_8^3 \end{aligned} \tag{4.37}$$

Using the above information and the butterfly structure shown in Fig. 4.10 we can draw the flow graph of 8-point DFT shown in Fig. 4.11.

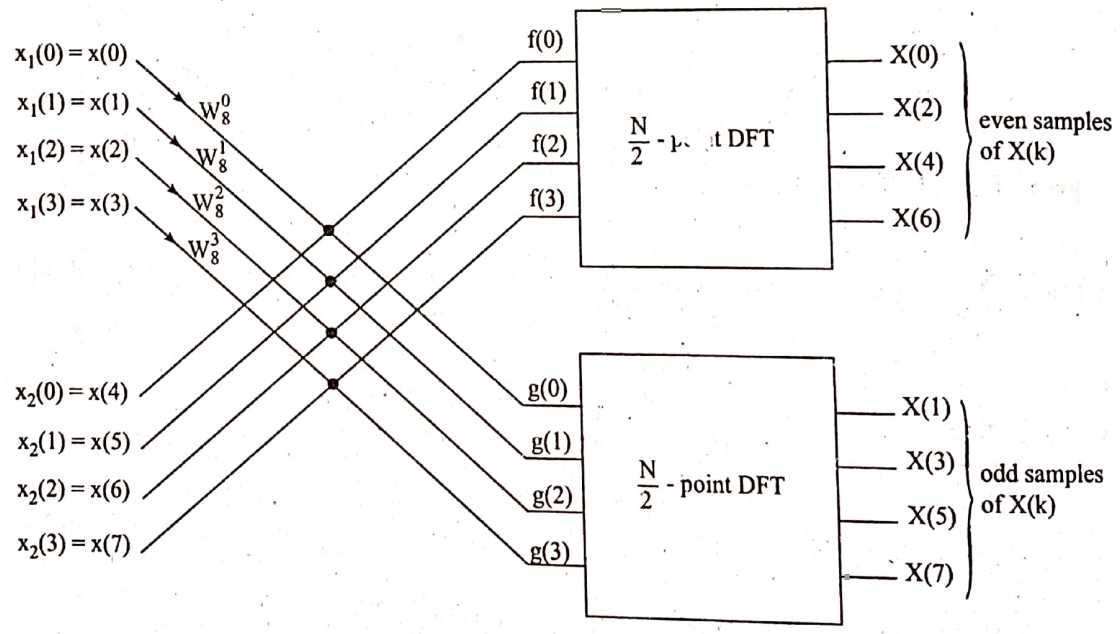

Fig. 4.11 Reduction of an 8-point DFT to two 4-point DFTs by decimation in frequency

Now each $\frac{N}{2}$-point DFT can be computed by combining the first half and the last half of the input points for each of the $\frac{N}{2}$-point DFTs and then computing $\frac{N}{4}$-point DFTs. For the 8-point DFT example the resultant flow graph is shown in Fig. 4.12.

The 2-point DFT can be found by adding and subtracting the input points. The Fig. 4.12 can be further reduced as in Fig. 4.13.

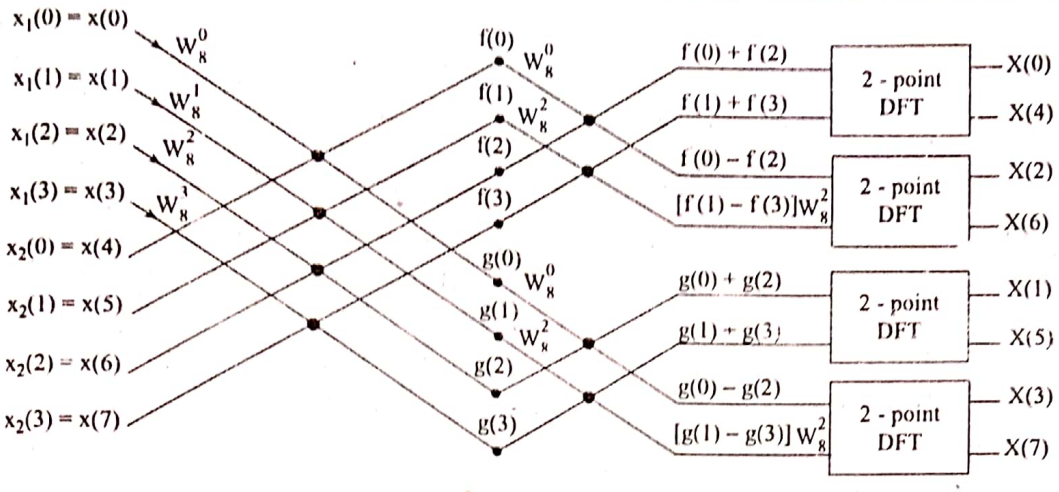

Fig. 4.12 Flow graph of decimation in frequency decomposition of an 8-point DFT into four 2-point DFT computations

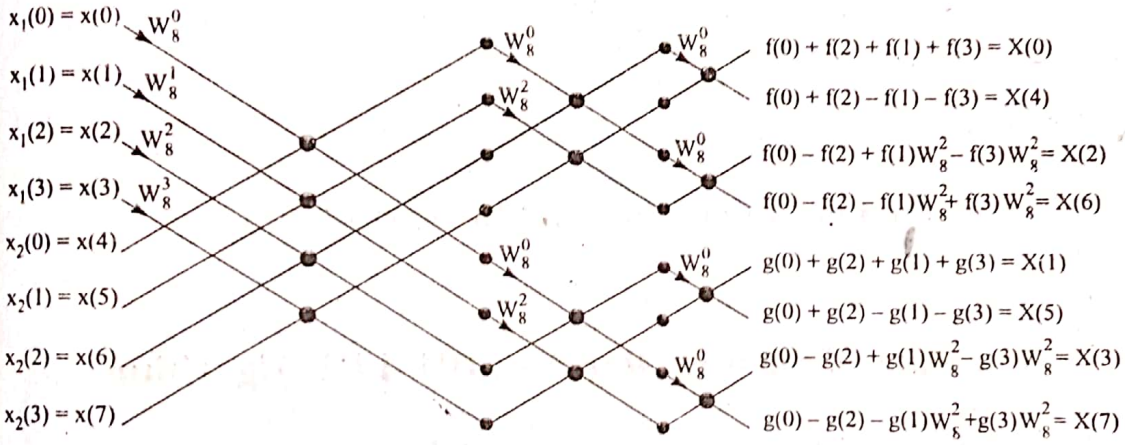

Fig. 4.13 Flow graph of 8-point DIF-FFT algorithm

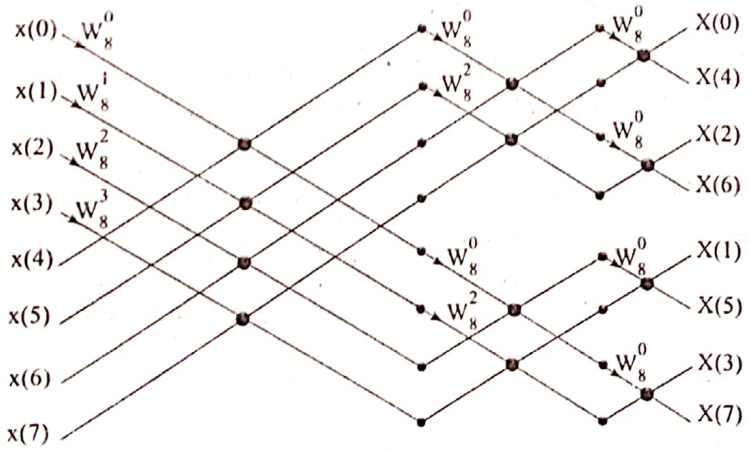

Fig. 4.14 Flow graph of complete decimation in frequency decomposition of an 8-point DFT computation

The complete flow graph of 8-point DFT using DIF algorithm is shown in Fig. 4.14. From Fig. 4.14 we observe that for DIF algorithm the input sequence is in natural order, while the output sequence is in bit reversal order, whereas the reverse is true for the DIT algorithm. From Fig. 4.13 we see that the number of complex multiplications required to evaluate $X(k)$ is $\frac{N}{2}\log_2 N$ and the number of complex additions required is $N\log_2 N$. Thus the total number of computations are same for DIT and DIF algorithms. The basic computational block in the diagram is the "butterfly" shown in Fig. 4.15. Like DIT algorithm, DIF algorithm also is an in-place algorithm where the same locations are used to store both the input and output sequences.

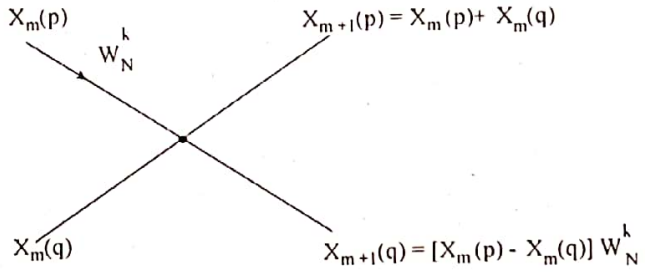

Fig. 4.15 Basic Computational diagram for DIF-FFT

4.7 Summary of Steps for Radix - 2 DIF-FFT Algorithm

1. The number of input samples $N = 2^M$, where, M is number of stages.
2. The input sequence is in natural order.
3. The number of stages in the flow graph is given by $M = \log_2 N$.
4. Each stage consists of $\frac{N}{2}$ butterflies.
5. Inputs/outputs for each butterfly are separated by 2^{M-m} samples, where m represents the stage index i.e., for first stage $m = 1$ and for second stage $m = 2$ so on.
6. The number of complex multiplications is given by $\frac{N}{2}\log_2 N$.
7. The number of complex additions is given by $N\log_2 N$.
8. The twiddle factor exponents are a function of the stage index m and is given by

$$k = \frac{Nt}{2^{M-m+1}}, \quad t = 0, 1, 2, \ldots, 2^{M-m} - 1 \tag{4.38}$$

9. The number of sets or sections of butterflies in each stage is given by the formula 2^{m-1}.
10. The exponent repeat factor (ERF), which is the number of times the exponent sequence associated with m repeated is given by 2^{m-1}.

4.8 Differences and Similarities between DIT and DIF Algorithms

Differences

1. For decimation-in-time (DIT), the input is bit-reversed while the output is in natural order. Whereas, for decimation-in-frequency the input is in natural order while the output is bit reversed order.
2. The DIF butterfly is slightly different from the DIT wherein DIF the complex multiplication takes place after the add-subtract operation.

Similarities

Both algorithms require $N \log_2 N$ operations to compute the DFT. Both algorithms can be done in-place and both need to perform bit reversal at some place during the computation.

Example 4.7 The butterfly diagram of $N = 64$ FFT algorithm has a twiddle factor W_{64}^0 for one of the butterflies in the last stage. Is the FFT a decimation-in-time or decimation-in-frequency algorithm.

Solution The FFT is a decimation-in-frequency algorithm, since the decimation-in-frequency algorithm has W_{64}^0 terms in the last stage.

Pratice Problem 4.4 The butterfly shown in Fig. 4.16 was taken from a decimation-in-frequency FFT with $N = 16$. Determine which of the four stages have bufferfly of this form?

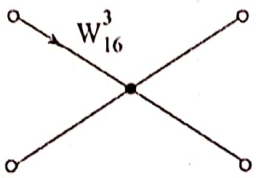

Fig. 4.16

Ans: First stage