Unit -3---DISCRETE FOURIER TRANSFORM

Dr.K.B.Khanchandani
Profesor ,Deptt of Electronics and Tele.Engg
SSGMCE SHEGAON

DECIMATION IN FREQUENCY (DIF) RADIX-2 FFT

- In decimation in frequency algorithm, the frequency domain sequence *X(k) is decimated.*
- *In* this algorithm, the *N-point time domain sequence is converted to two numbers of N/2-point* sequences.
- Then each *N/2-point sequence is converted to two numbers of N/4-point* sequences. This process is continued until we get *N/2 numbers of 2-point sequences.*
- *Finally,* the 2-point DFT of each 2-point sequence is computed. The 2-point DFTs of *N/2 numbers of 2*-point sequences will give *N-samples, which is the N-point DFT of the time domain sequence.*
- Here the equations for *N/2-point sequences, N/4-point sequences, etc., are obtained by* decimation of frequency domain sequences. Hence this method is called DIF.

To derive the decimation-in-frequency form of the FFT algorithm for N, a power of 2, we can first divide the given input sequence $x(n) = \{x(0), x(1), x(2), x(3), x(4), x(5), x(6), x(7)$ into the first half and last half of the points so that its DFT $X(k)$ is

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

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

It is important to observe that while the above equation for $X(k)$ contains two summations over $N/2$-points, each of these summations is not an $N/2$-point DFT, since W_N^{nk} rather than $W^{nk\,N/2}$

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

$$= \sum_{n=0}^{N/2-1} \left[x(n) W_N^{\ nk} + (-1)^{nk} x(n+\frac{N}{2}) W_N^{\ nk} \right]$$

$$= \sum_{n=0}^{N/2-1} \left[x(n) + (-1)^{nk} x(n+\frac{N}{2}) \right] W_N^{\ nk}$$

Let us split $X(k)$ into even and odd numbered samples. For even values of k, the $X(k)$ can be written as

$$X(2K) = \sum_{n=0}^{N/2-1} \left[x(n) + (-1)^{2k} x(n+\frac{N}{2}) W_N^{\ 2nk} \right]$$

$$= \sum_{n=0}^{N/2-1} \left[x(n) + x(n+\frac{N}{2}) \right] W_N^{\ nk}$$

For odd values of k, the X(k) can be written as

$$X(2K+1) = \sum_{n=0}^{N/2-1} \left[x(n) + (-1)^{2k+1} x(n+\frac{N}{2}) W_N^{\ (2k+1)n} \right]$$

$$= \sum_{n=0}^{N/2-1} \left[x(n) - x(n+\frac{N}{2}) \right] W_N^{\ nk} W_{N/2}^{\ nk}$$

The above equations for $X(2k)$ and $X(2k + 1)$ can be recognized as $N/2$-point DFTs. $X(2k)$ is the DFT of the sum of first half and last half of the input sequence, i.e. of $\{x(n) + x(n + N/2)\}$ and $X(2k + 1)$ is the DFT of the product W_N^n with the difference of first half and last half of the input, i.e. of $\{x(n) \square x(n + N/2)\} W_N{}^n$.

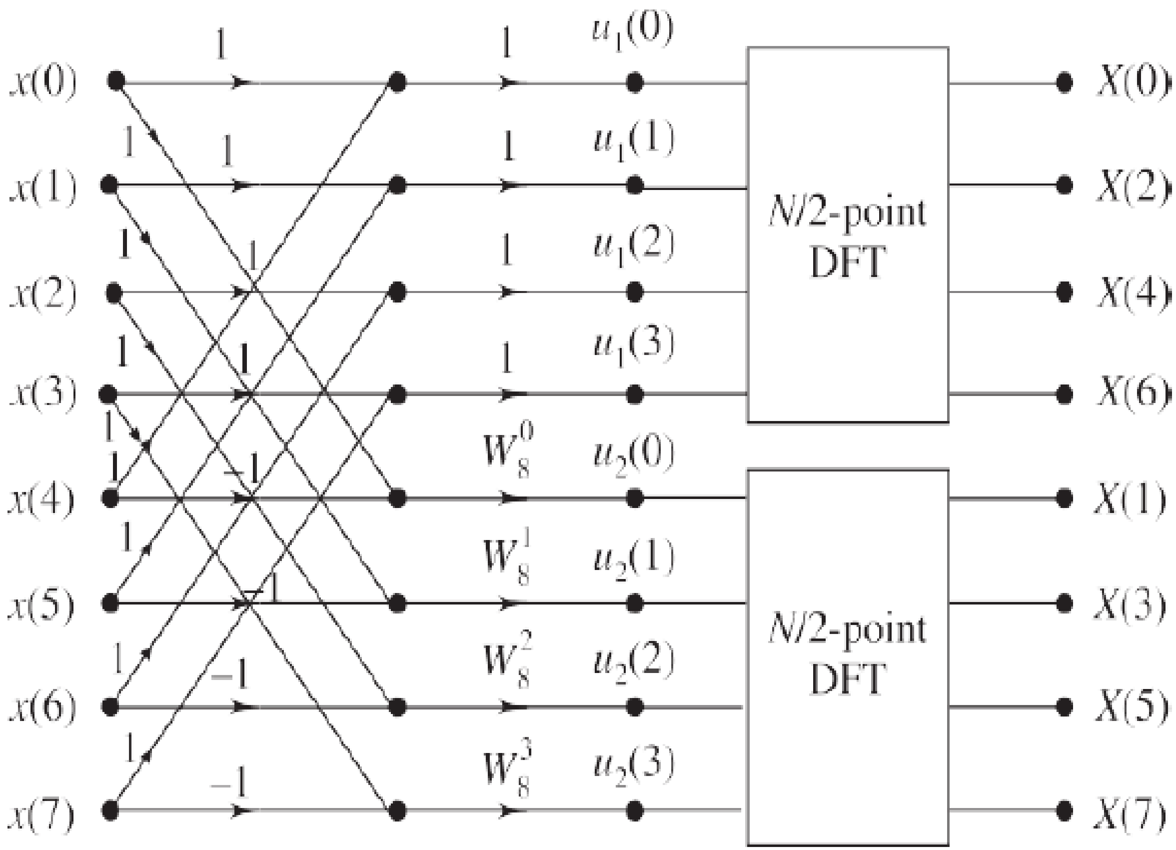
If we define new time domain sequences, $u_1(n)$ and $u_2(n)$ consisting of $N/2$-samples, such that

$$u_1(n) = x(n) + x\left(n + \frac{N}{2}\right); \quad \text{for } n = 0, 1, 2, \ldots, \frac{N}{2} - 1$$

and

$$u_2(n) = \left[x(n) - x\left(n + \frac{N}{2}\right)\right] W_N^n; \quad \text{for } n = 0, 1, 2, \ldots, \frac{N}{2} - 1$$

then the DFTs $U_1(k) = X(2k)$ and $U_2(k) = X(2k + 1)$ can be computed by first forming the sequences $u_1(n)$ and $u_2(n)$, then computing the $N/2$-point DFTs of these two sequences to obtain the even numbered output points and odd numbered output points respectively. The procedure suggested above is illustrated in Figure 7.9 for the case of an 8-point sequence.



FFow graph of the DIF decomposition of an N–point DF† computation into two $N/2$–point DF† computations $N = 8$.

Now each of the $N/2$-point frequency domain sequences, $U_1(k)$ and $U_2(k)$ can be decimated into two numbers of $N/4$-point sequences and four numbers of new $N/4$-point sequences can be obtained from them.

Let the new sequences be $v_{11}(n)$, $v_{12}(n)$, $v_{21}(n)$, $v_{22}(n)$. On similar lines as discussed above, we can get

$$v_{11}(n) = u_1(n) + u_1(n+2); \quad \text{for } n = 0, 1, 2, \ldots, \frac{N}{4} - 1$$

$$v_{12}(n) = [u_1(n) - u_1(n+2)]W_{N/2}^{n}; \quad \text{for } n = 0, 1, 2, \ldots, \frac{N}{4} - 1$$

$$v_{21}(n) = u_2(n) + u_2(n+2); \quad \text{for } n = 0, 1, 2, \ldots, \frac{N}{4} - 1$$

$$v_{22}(n) = [u_2(n) - u_2(n+2)]W_{N/2}^{n}; \quad \text{for } n = 0, 1, 2, \ldots, \frac{N}{4} - 1$$

This process is continued till we get only 2-point sequences. The DFT of those 2-point sequences is the DFT of $x(n)$, i.e. $X(k)$ in bit reversed order.

The third stage of computation for $N = 8$ is shown in Figure 7.11.

The entire process of decimation involves m stages of decimation where $m = \log_2 N$. The computation of the N-point DFT via the DIF FFT algorithm requires $(N/2) \log_2 N$ complex multiplications and $(N - 1) \log_2 N$ complex additions (i.e. total number of computations remains same in both DIF and DIT).

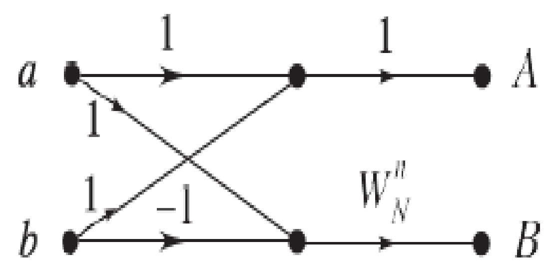


Figure 7.12 Basic butterfFy diagram for DIF FI†.

The signal flow graph or butterfly diagram of all the three stages together is shown in Figure 7.13.

Observing the basic calculations, each stage involves $N/2$ butterflies of the type shown in Figure 7.12.

The butterfly computation involves the following operations:

(i) In each computation two complex numbers a and b are considered.

(ii) The sum of the two complex numbers is computed which forms a new complex number A.

(iii) Subtract the complex number b from a to get the term $(a - b)$. The difference term $(a - b)$ is multiplied with the phase factor or twiddle factor W_N^n to form a new complex number B.

TKE 8-POINT DFT USIKG RADIX-2 DIF FFT

The DIF computations for an 8-sample sequence are given below in detail

Let $x(n) = \{x(0), x(1), x(2), x(3), x(4), x(5), x(6), x(7)\}$ be the given 8-sample sequence.

First stage of COMPUTATION

In the first stage of computation, two numbers of 4-point sequences $u_1(n)$ and $u_2(n)$ are obtained from the given 8-point sequence $x(n)$ as shown below.

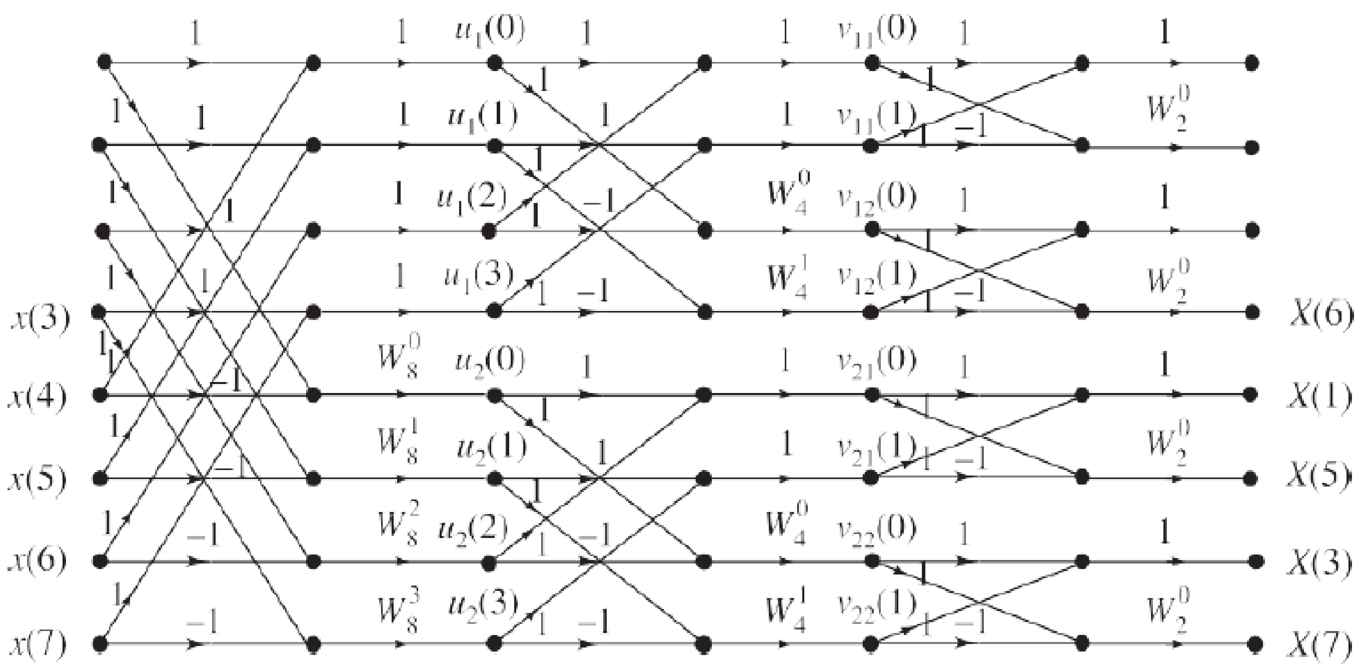


Figure 7.13 SignaF fFow graph or butterfFy diagram for the 8–point radix–2 DIF FF† aFgorithm.

Seconb stage of CONPUTATION

In the second stage of computation, four numbers of 2-point sequences $v_{11}(n)$, $v_{12}(n)$ and $v_{21}(n)$, $v_{22}(n)$ are obtained form the two 4-point sequences $u_1(n)$ and $u_2(n)$ obtained in stage one

Thirb stage of CONPUTATION

In the third stage of computation, the 2-point DFTs of the 2-point sequences obtained in the second stage . The computation of 2-point DFTs is done by the butterfly operation shown in Figure 7.14(c).

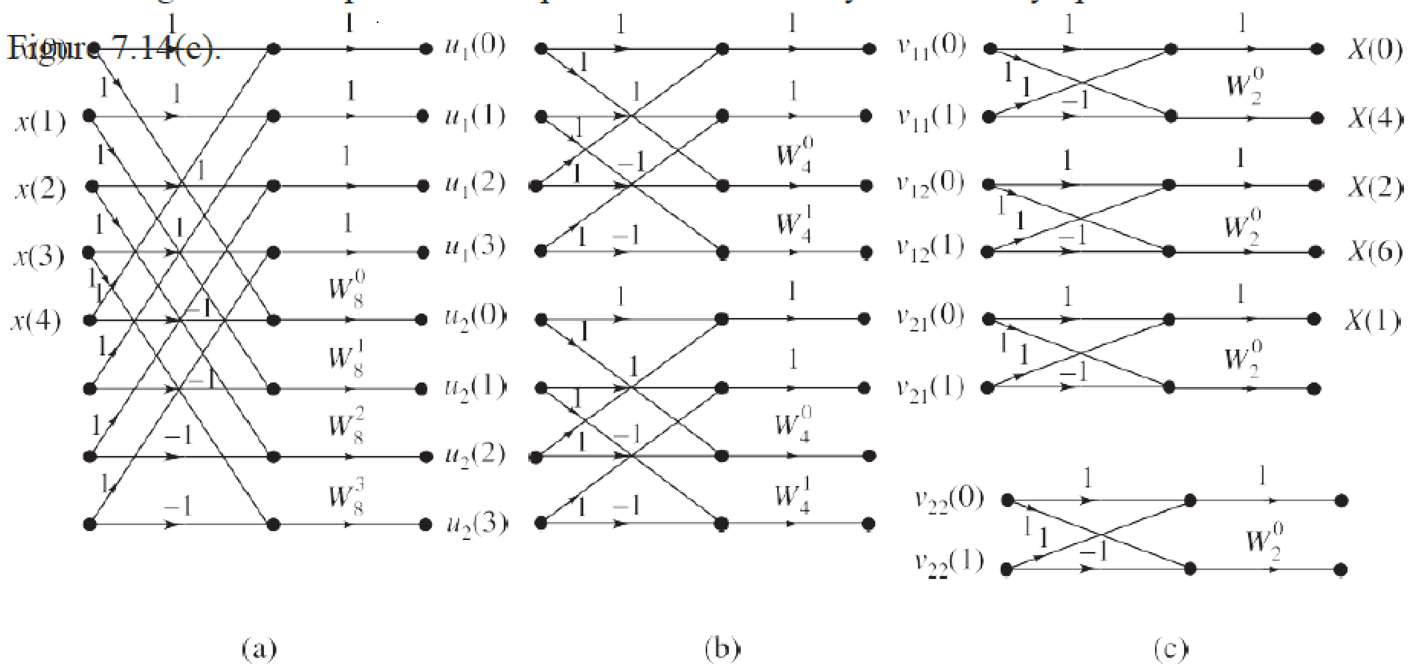


Figure 7.14 (a)–(c) †he first, second and third stages of computation of 8–point DF† by Radix–2 DIF FF†.

EXAMPLE : Implement the decimation-in-frequency FFT algorithm of *N-point DFT where N = 8. Also explain the steps involved in this algorithm.*

- ***Solution: The 8-point radix-2 DIF FFT algorithm***
- 1. It involves 3 stages of computation. The input to the first stage is the input time sequence *x(n) in normal order. The output of first stage is the input to the second stage and the output of second stage is the input to the third stage. The output of third stage is the 8-point DFT in bit reversed order.*
- 2. In DIF algorithm, the frequency domain sequence *X(k) is decimated.*
- 3. In this algorithm, the *N-point time domain sequence is converted to two numbers of N/2-point sequences. Then each N/2-point sequence is converted to two numbers of N/4-point sequences. Thus, we get 4 numbers of N/4, i.e. 2-point sequences.*
- 4. Finally, the 2-point DFT of each 2-point sequence is computed. The 2-point DFTs of *N/2 number of 2-point sequences will give N-samples which is the N-point DFT of the time domain sequence. The implementation of the 8-point radix-2 DIF FFT algorithm is shown in Figure*

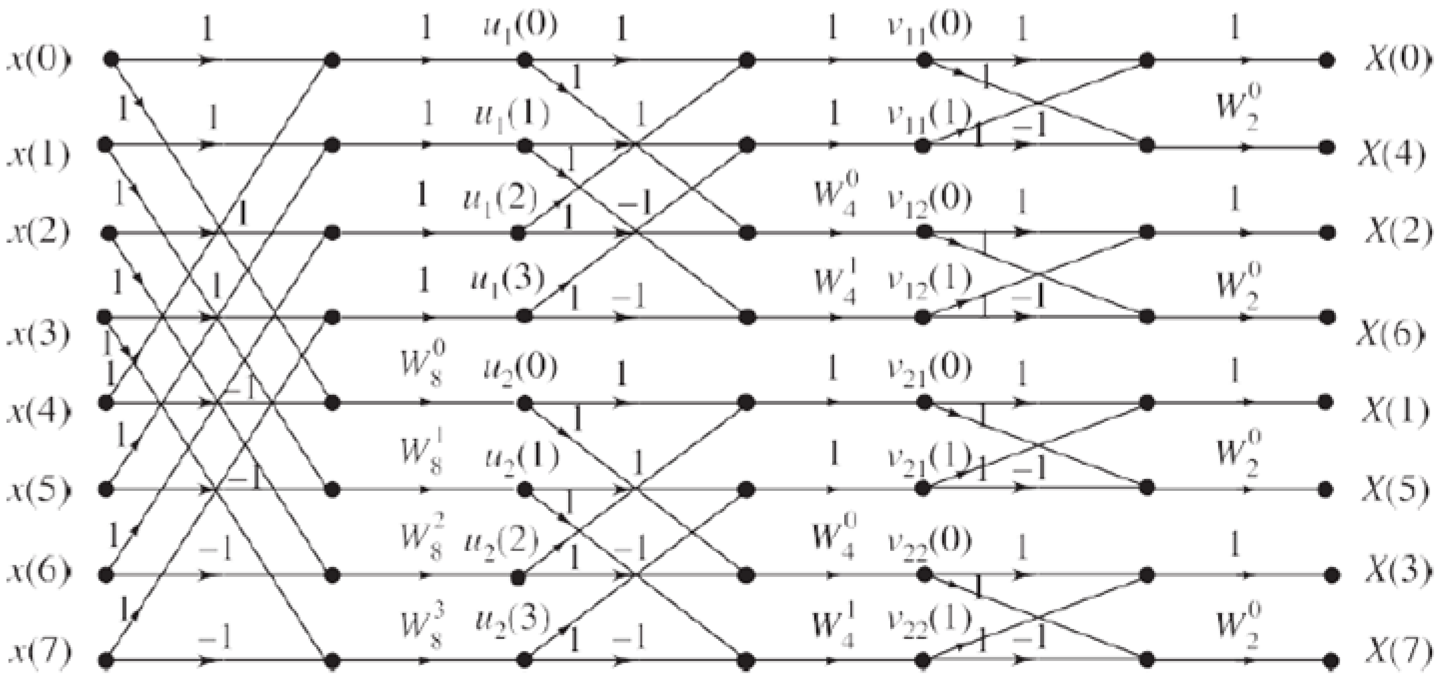


Figure Butterfly Fine diagram for 8–point radix–2 DIF FFT algorithm.

EXAMPLE An 8-point sequence is given by $x(n) = \{2, 2, 2, 2, 1, 1, 1, 1\}$. Compute the 8-point DFT of $x(n)$ by

(a) Radix-2 DIT FFT algorithm
(b) Radix-2 DIF FFT algorithm

Also sketch the magnitude and phase spectrum.

Solution: (a) 8-point DFT by Radix-2 DIT FFT algorithm

The given sequence is $x(n) = \{x(0), x(1), x(2), x(3), x(4), x(5), x(6), x(7)\}$

$$= \{2, 2, 2, 2, 1, 1, 1, 1\}$$

The given sequence in bit reversed order is

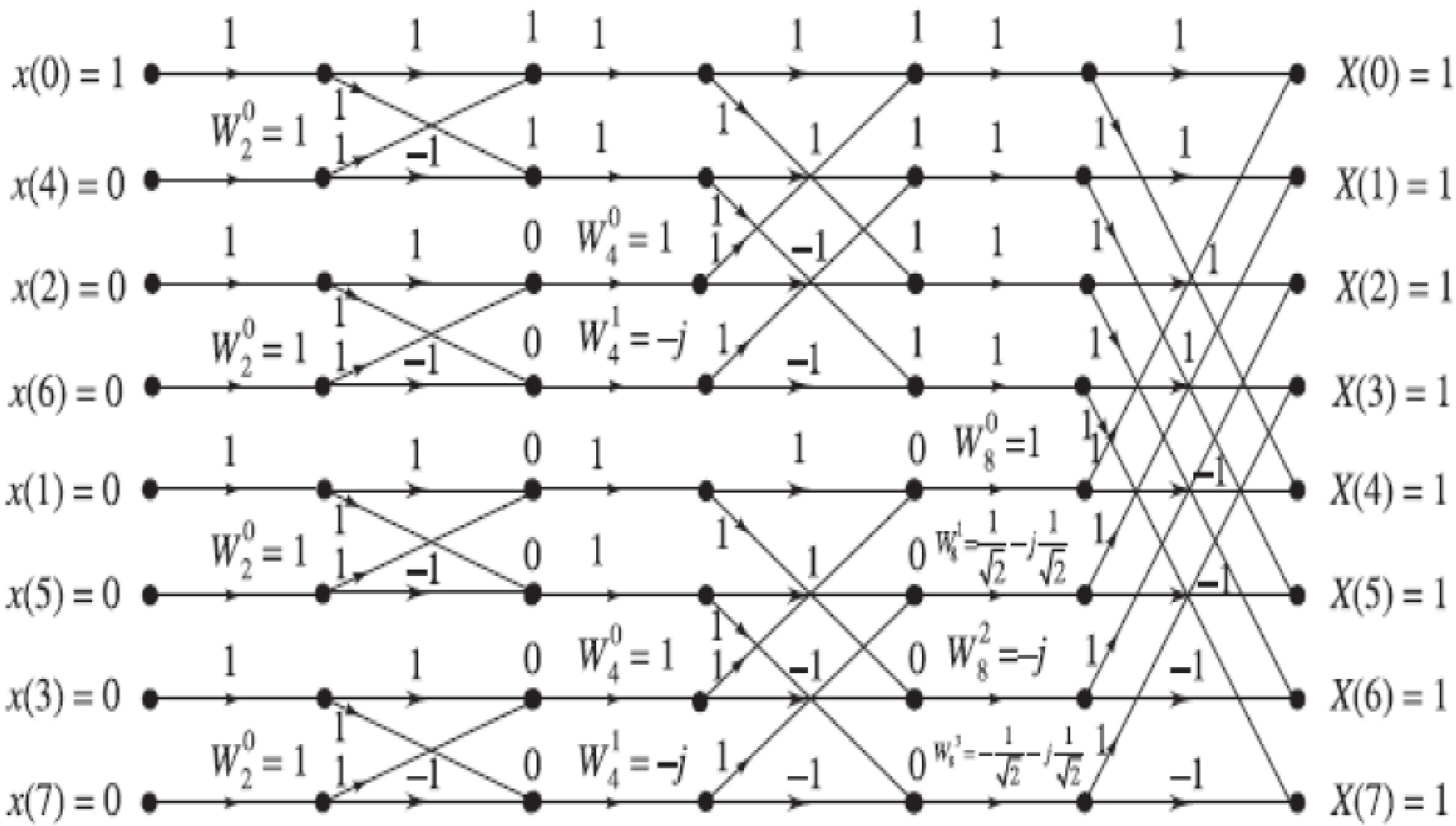
$$x_r(n) = \{x(0), x(4), x(2), x(6), x(1), x(5), x(3), x(7)\}$$

$$= \{2, 1, 2, 1, 2, 1, 2, 1\}$$

For DIT FFT, the input is in bit reversed order and the output is in normal order. The computation of 8-point DFT of $x(n)$, i.e. $X(k)$ by Radix-2 DIT FFT algorithm is shown in Figure .

From Figure , we get the 8-point DFT of $x(n)$ as

$$X(k) = \{12, 1 \square j2.414, 0, 1 \square j0.414, 0, 1 + j0.414, 0, 1 + j2.414\}$$



x(0) = 1
x(4) = 0
x(2) = 0
x(6) = 0
x(1) = 0
x(5) = 0
x(3) = 0
x(7) = 0
W_2^0 = 1
W_4^0 = 1
W_4^1 = −j
W_8^0 =1
W_8^1 = 1/√2 − j 1/√2
W_8^2 = −j
W_8^3 = −1/√2 − j 1/√2
X(0) = 1
X(1) = 1
X(2) = 1
X(3) = 1
X(4) = 1
X(5) = 1
X(6) = 1
X(7) = 1

(b) 8-point DFT by radix-2 DIF FFT algorithm

For DIF FFT, the input is in normal order and the output is in bit reversed order. The computation of DFT by radix-2 DIF FFT algorithm is shown in Figure

From Figure , we observe that the 8-point DFT in bit reversed order is

$$X_r(k) = \{X(0), X(4), X(2), X(6), X(1), X(5), X(3), X(7)\}$$

$$= \{12, 0, 0, 0, 1 \square j2.414, 1 + j0.414, 1 \square j0.414, 1 + j2.414\}$$

The 8-point DFT in normal order is

$$X(k) = \{X(0), X(1), X(2), X(3), X(4), X(5), X(6), X(7)\}$$

$$= \{12, 1 \square j2.414, 0, 1 \square j0.414, 0, 1 + j0.414, 0, 1 + j2.414\}$$

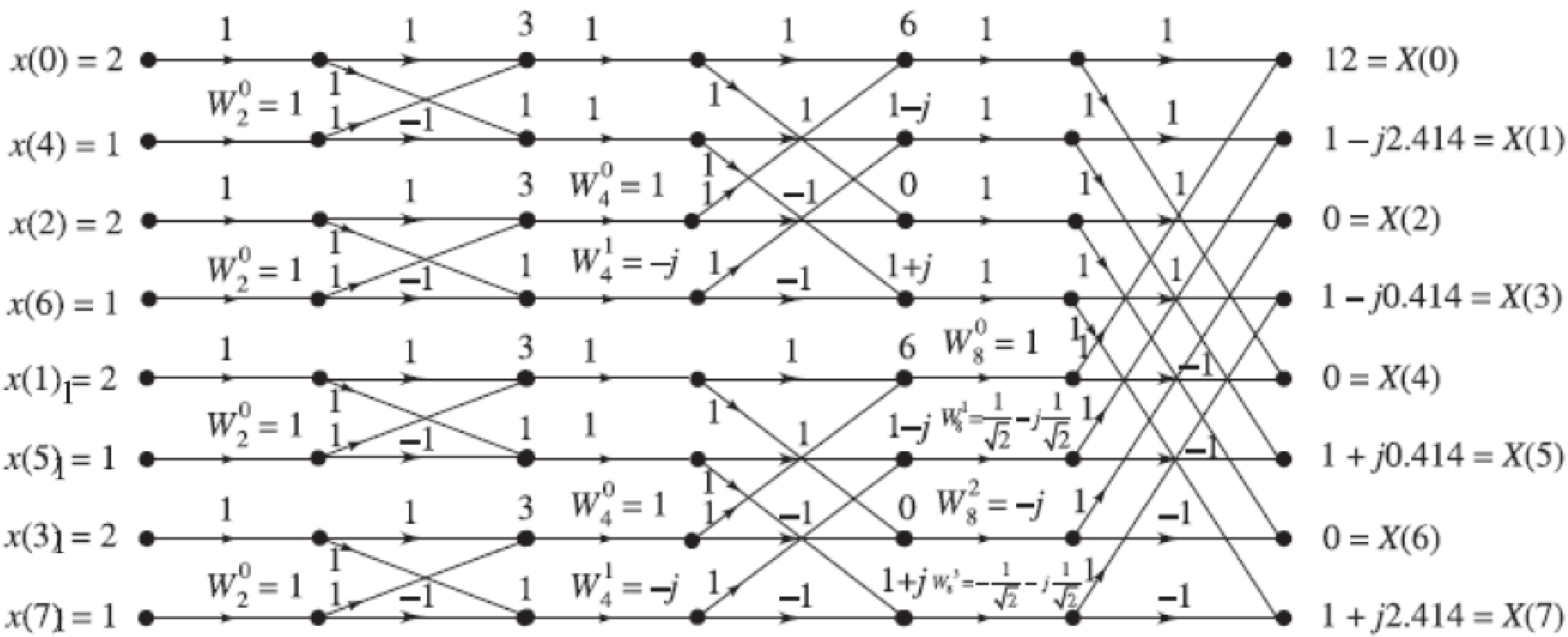


Figure Computation of 8–point DF† of $x(n)$ by radix–2 DIF FF† aFgorithm.

- CONPARISON of DIT (DECIMATION-IN-TINE) and DIF (DECINATION-IN-FREQUENCY) ALGORITHMS
-

1. In DIT, the input is bit reversed while the output is in normal order. For DIF, the reverse is true, i.e. the input is in normal order, while the output is bit reversed. However, both DIT and DIF can go from normal to shuffled data or vice versa.

2. Considering the butterfly diagram, in DIT, the complex multiplication takes place before the add subtract operation, while in DIF, the complex multiplication takes place after the add subtract operation.

- Both algorithms require the same number of operations to compute DFT.
- Both algorithms require bit reversal at some place during computation.