School of Computing
National University of Singapore
CS5240 Theoretical Foundations in Multimedia

Discrete Fourier Transform

1 Discrete Fourier Transform (DFT)

You have seen how the DTFT analyzes a signal $x[n]$ in terms of complex exponentials (i.e. cosines and sines) of different frequencies ω. The forward transform:

$$X(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x[n]e^{-j\omega n} \tag{1}$$

computes the complex coefficients $X(e^{j\omega})$ as the inner product of the signal $x[n]$ and the orthonormal basis "vectors" $e^{j\omega n}$. Strictly speaking, we need a $\frac{1}{\sqrt{2\pi}}$ factor to normalize the basis vectors, but this is compensated by the $\frac{1}{2\pi}$ factor in the inverse DTFT:

$$x[n] = \frac{1}{2\pi}\int_{-\pi}^{\pi} X(e^{j\omega})e^{j\omega n}d\omega \tag{2}$$

Recall from linear algebra theory that the result of computing these inner products is to express $x[n]$ in the new coordinate basis $e^{j\omega n}$. Thus, each $X(e^{j\omega})$ indicates how much its corresponding basis vector $e^{j\omega n}$ is present in $x[n]$.

This makes the DTFT a great analytical tool. Unfortunately, it is not very practical for numerical computation, because $X(e^{j\omega})$ is a continuous function of a continuous variable ω. We would prefer a discrete and finite version for practical computation.

The Discrete Fourier Transform (DFT) is such a tool. It analyzes a finite-duration sequence in terms of a discrete and finite set of frequencies. Furthermore, there are fast algorithms to compute the DFT.

The DFT is defined as:

$$X[k] = \sum_{n=0}^{N-1} x[n]\, W_N^{nk}, \quad 0 \le k \le N-1 \tag{3}$$

where $W_N \triangleq e^{-j2\pi/N}$ and N is the length of the signal $x[n]$, i.e. $x[n]$ is 0 outside the interval $0 \le n \le N-1$.

The inverse DFT is:

$$x[n] = \frac{1}{N}\sum_{k=0}^{N-1} X[k]\, W_N^{-nk}, \quad 0 \le n \le N-1 \tag{4}$$

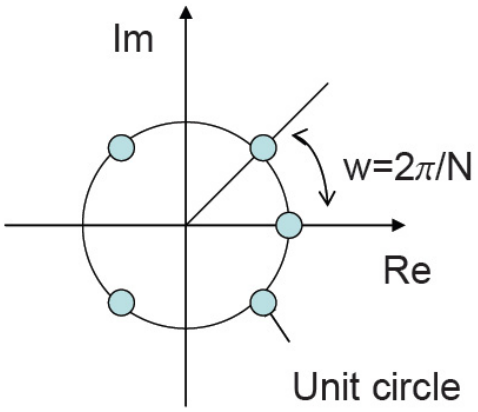


Figure 1: Basis vectors $e^{j(2\pi/N)k}, k = 0, 1, \ldots, N-1$, are equally-space points on the unit circle.

Notes:

1. The DFT transforms a length-N sequence $x[n]$ into another length-N sequence $X[k]$.
2. The DTFT analyzes a signal using an infinite number of frequencies, $\omega \in [0, 2\pi]$ or $\omega \in [-\pi, \pi]$.
3. In the DTFT, all the basis vectors $e^{-j\omega n}$ lie on the unit circle in the complex plane. In the DFT, the basis vectors are N equally-spaced points on the unit circle (Figure 1).
4. This means that the DFT is a sampled-version of the DTFT. The sampling is done in the frequency domain (see Figure 2).
5. We can write the DFT in matrix form:

$$\underbrace{\begin{bmatrix} X[0] \\ X[1] \\ \vdots \\ X[N-1] \end{bmatrix}}_{\mathbf{X}} = \underbrace{\begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & W_N & W_N^2 & \cdots & W_N^{N-1} \\ 1 & W_n^2 & W_N^4 & \cdots & W_N^{2N-2} \\ \vdots & & & & \vdots \\ 1 & W_N^{N-1} & W_N^{2N-2} & \cdots & W_N^{(N-1)^2} \end{bmatrix}}_{\mathbf{V}\ (\text{DFT matrix})} \underbrace{\begin{bmatrix} x[0] \\ x[1] \\ \vdots \\ x[N-1] \end{bmatrix}}_{\mathbf{x}} \tag{5}$$

6. This inverse DFT is thus:

$$\mathbf{x} = \frac{1}{N}\mathbf{V}^H\mathbf{X}$$

   Observe that $\mathbf{V}$ is symmetric, but not Hermitian.

Note that there is a duality lurking in the background here. Figure 3 shows how sampling in one domain results in periodization in the other domain. Starting from a continuous time signal $x(t)$, its Continuous Fourier Transform (CFT) may be computed as $X(j\Omega)$ (Search the web to find the transform formula; CFT is outside the scope of this course). When you

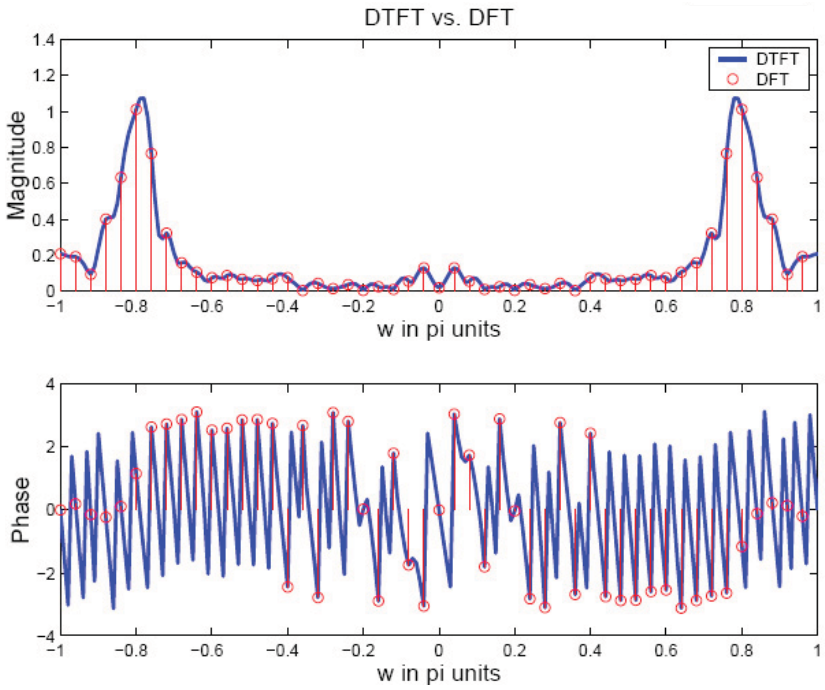


Figure 2: DFT is a sampled version of DTFT. The sampling is done in the frequency domain.

sample $x(t)$ in time to get $x[n]$, you are making $X(j\Omega)$ periodic, i.e. $X(e^{j\omega})$ is a periodized version of $X(j\Omega)$, with period 2π.

Now, when you compute the N-point DFT, you are sampling $X(e^{j\omega})$ in the frequency domain. This causes $x[n]$ to be periodized, with period N. This periodized version is $\tilde{x}[n]$.

2 Properties of DFT:

In Table 1, the expressions given specify $x[n]$ for $0 \leq n \leq N-1$ and $X[k]$ for $0 \leq k \leq N-1$. Both $x[n]$ and $X[k]$ are equal to zero outside those ranges.

Many properties of the DFT are similar to those of the DTFT, e.g. linearity, duality. Some are different. One such difference is in convolution.

When using the DFT for convolution, we are computing the circular convolution, not the linear convolution. This is because the DFT, being a frequency-sampled version of the DTFT, actually causes the signal $x[n]$ to be periodically extended in the time domain. This means that the length-N signal $x[n]$ is repeated infinitely with period N. The new periodic signal $\tilde{x}[n]$ is related to $x[n]$ by: $\tilde{x}[n] = x[((n))_N]$, where $((n))_N$ means "n modulo N", e.g. $((3))_5 = ((8))_5 = ((-2))_5 = ((5k+3))_5 = 3$

Let $\textcircled{N}$ denote the N-point circular convolution operation. Then, convolving the finite-length sequences $x_1[n]$ and $x_2[n]$ is equivalent to the following:

Table 1: Summary of Properties of DFT

Finite-Length Sequence (Length N)	N-point DFT (Length N)
1. $x[n]$	$X[k]$
2. $x_1[n]$,$x_2[n]$	$X_1[k]$,$X_2[k]$
3. $ax_1[n] + bx_2[n]$	$aX_1[k] + bX_2[k]$
4. $X[n]$	$Nx[((-k))_N]$
5. $x[((n-m))_N]$	$W_N^{km}X[k]$
6. $W_N^{-ln}x[n]$	$X[((k-l))_N]$
7. $\sum_{m=0}^{N-1} x_1(m)x_2[((n-m))_N]$	$X_1[k]X_2[k]$
8. $x_1[n]x_2[n]$	$\frac{1}{N}\sum_{l=0}^{N-1} X_1(l)X_2[((k-l))_N]$
9. $x^*[n]$	$X^*[((-k))_N]$
10. $x^*[((-n))_N]$	$X^*[k]$
11. $Re\{x[n]\}$	$X_e[k] = \frac{1}{2}\{X[((k))_N] + X^*[((-k))_N]\}$
12. $jIm\{x[n]\}$	$X_o[k] = \frac{1}{2}\{X[((k))_N] - X^*[((-k))_N]\}$
13. $x_e[n] = \frac{1}{2}\{x[n] + x^*[((-n))_N]\}$	$Re\{X[k]\}$
14. $x_o[n] = \frac{1}{2}\{x[n] - x^*[((-n))_N]\}$	$jIm\{X[k]\}$
Properties 15-17 apply only when $x[n]$ is real.	
15. Symmetric properties	$\begin{cases} X[k] & = X^*[((-k))]_N \\ Re\{X[k]\} & = Re\{X[((-k))_N]\} \\ Im\{X[k]\} & = -Im\{X[((-k))_N]\} \\ \lvert X[k]\rvert & = \lvert X[((-k))_N]\rvert \\ \sphericalangle\{X[k]\} & = -\sphericalangle\{X[((-k))_N]\} \end{cases}$
16. $x_e[n] = \frac{1}{2}\{x[n] + x[((-n))_N]\}$	$Re\{X[k]\}$
17. $x_o[n] = \frac{1}{2}\{x[n] - x[((-n))_N]\}$	$jIm\{X[k]\}$

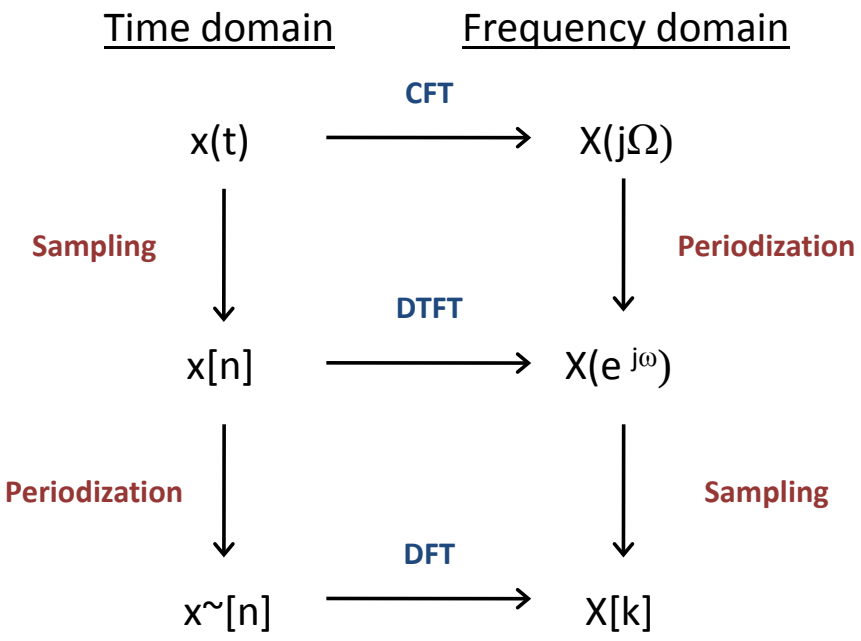


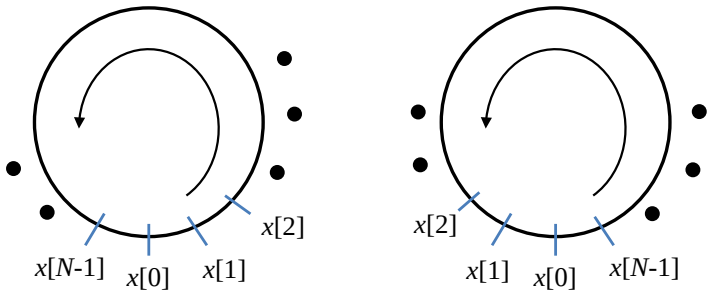
Figure 3: Duality in sampling.

1. $\tilde{x}_3[n] = \tilde{x}_1[n] * \tilde{x}_2[n]$, i.e. linearly convolve the periodic extensions of $x_1[n], x_2[n]$.
2. $x_3[n] = \begin{cases} \tilde{x}_3[n], & \text{for } 0 \le n \le N-1 \\ 0, & \text{otherwise} \end{cases}$

This computes $x_3[n] = x_1[n] \textcircled{N} x_2[n]$.

2.1 Circular convolution:

The previous description of circular convolution is OK conceptually. Practically, we do it differently. First, consider circular folding: $x[((-n))_N]$



$x[((n))_N]$

Wrap signal anticlockwise
Read anticlockwise

$x[((-n))_N]$

Wrap signal clockwise
Read anticlockwise

Circular convolution is defined by:

$$x_1[n] \textcircled{N} x_2[n] = \sum_{m=0}^{N-1} x_1[m]x_2[((n-m))_N], \quad 0 \leq n \leq N-1 \tag{6}$$

e.g. Let $x_1[n] = 1, 2, 2, x_2[n] = 1, 2, 3, 4$. Compute $x_1[n] \textcircled{4} x_2[n] = y[n]$.

Now $x_1[m] = \{1, 2, 2, 0\}, x_2[((-m))_4] = \{1, 4, 3, 2\}$

$$\begin{aligned} y[0] &=< x_1[m],\, x_2[((0-m))_4] > \\ &=< \{1, 2, 2, 0\},\, \{1, 4, 3, 2\} > \\ &= 1 + 8 + 6 + 0 = 15 \end{aligned}$$

$$\begin{aligned} y[1] &= < x_1[m],\, x_2[((1-m))_4] > \\ &= < \{1, 2, 2, 0\},\, \{2, 1, 4, 3\} > \\ &= 2 + 2 + 8 + 0 = 12 \end{aligned}$$

$$\begin{aligned} y[2] &=< x_1[m],\, x_2[((2-m))_4] > \\ &=< \{1, 2, 2, 0\},\, \{3, 2, 1, 4\} > \\ &= 3 + 4 + 2 + 0 = 9 \end{aligned}$$

$$\begin{aligned} y[3] &= < x_1[m],\, x_2[((3-m))_4] > \\ &= < \{1, 2, 2, 0\},\, \{4, 3, 2, 1\} > \\ &= 4 + 6 + 4 + 0 = 14 \end{aligned}$$

So, $y[n] = \{15, 12, 9, 14\}$

DFT approach:

1. Compute DFT of $x_1, x_2 \rightarrow X_1[k], X_2[k]$
2. Multiply to get $Y[k] = X_1[k]X_2[k]$
3. Inverse DFT of $Y[k]$

Note: 4-point DFT: $W_4 = e^{-j2\pi/4} = e^{-j\pi/2} = -j$

$$\begin{aligned} X_1[k] = \sum_{n=0}^{3} x_1[n]W_4^{nk} &= 1 + 2(-j)^k + 2(-j)^{2k},\ 0 \le k \le 3 \\ &= \{5, -1-2j, 1, -1+2j\} \end{aligned}$$

$$\begin{aligned} X_2[k] = \sum_{n=0}^{3} x_2[n]W_4^{nk} &= 1 + 2(-j)^k + 3(-j)^{2k} + 4(-j)^{3k},\ 0 \le k \le 3 \\ &= \{10, -2+2j, -2, -2-2j\} \end{aligned}$$

$$\begin{aligned} Y[k] &= X_1[k]X_2[k] \\ &= \{50, 6+2j, -2, 6-2j\} \end{aligned}$$

$$\begin{aligned} y[n] &= \frac{1}{4}\sum_{k=0}^{3} Y[k]W_4^{-nk} \\ &= \frac{1}{4}\left[50 + (6+2j)(-j)^{-n} + (-2)(-j)^{-2n} + (6-2j)(-j)^{-3n}\right], \quad 0 \le n \le 3 \end{aligned}$$

$y[n] = \{15, 12, 9, 14\}$ as before.

Q: What happens if we perform an 8-point circular convolution $x_1[n]$ (8) $x_2[n]$? Will we get the same result?
Q: How can we use circular convolution to do linear convolution?

Circular convolution in matrix form:
e.g. $x = \{x_0, x_1, x_2, x_3\}$, $h = \{h_0, h_1, h_2, h_3\}$

$$\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} h_0 & h_3 & h_2 & h_1 \\ h_1 & h_0 & h_3 & h_2 \\ h_2 & h_1 & h_0 & h_3 \\ h_3 & h_2 & h_1 & h_0 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

Notice the structure of the matrix: circulant matrix, $\mathbf{C}$.

Theorem: Every circulant matrix can be diagonalized by the DFT matrix.

Thus, $\mathbf{C} = \mathbf{V}^H\mathbf{D}\mathbf{V}$

So $\mathbf{y} = \mathbf{C}\mathbf{x} = \mathbf{V}^H\mathbf{D}\mathbf{V}\mathbf{x}$

where $\mathbf{V}^H$ is the inverse DFT; $\mathbf{D}$ is multiplication by eigenvalues; and $\mathbf{V}\mathbf{x}$ computes the DFT of $\mathbf{x}$.

The above shows how circular convolution can be done using the DFT. But is this more efficient than direct convolution? Count the number of multiplications needed:

Direct convolution: $O(N^2)$
Fast Fourier Transform: $O(N \log_2 N)$
Therefore convolution via FFT: FFTs of h, x, then multiplication, then inverse FFT = $O(3N \log N + N)$.

It is clear the using FFT is much faster than direct convolution.

3 Fast Fourier Transform

Direct computation of the DFT is slow: requires $O(N^2)$ multiplications of complex numbers. So although the Fourier Transform was already know for 150 years, it remained a theoretical analysis tool only, with not much practical use. In 1965, James Cooley and John Tukey at the IBM T.J. Watson Research Center published a landmark paper on how to speed up the computation of the DFT, requiring only $O(N \log N)$ multiplications.

This created a boom in DSP, because the FFT could be directly implemented in hardware. Many other fast versions of the DFT were also proposed: Goertzel, Winograd, Chirp transforms. We illustrate the basic idea here:

Assume N is even, The DFT is given by:

$$\begin{aligned}\mathbf{X}[k] &= \sum_{n=0}^{N-1} x[n]W_N^{nk},\ k = 0, 1, \ldots, N-1 \\ &= \sum_{n \text{ even}} x[n]W_N^{nk} + \sum_{n \text{ odd}} x[n]W_N^{nk}\end{aligned}$$

Let $n = 2r$ for even n, $n = 2r + 1$ for odd n. Then

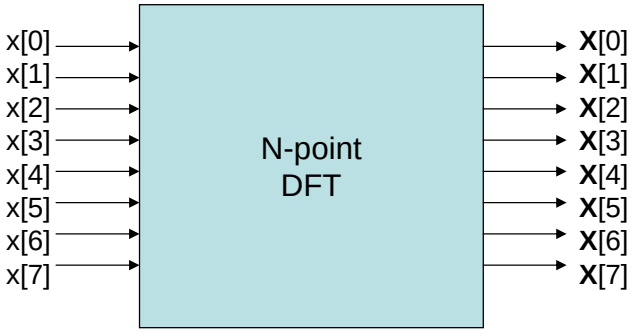
$$\begin{aligned}\mathbf{X}[k] &= \sum_{r=0}^{N/2-1} x[2r]W_N^{2rk} + \sum_{r=0}^{N/2-1} x[2r+1]W_N^{(2r+1)k} \\ &= \sum_{r=0}^{N/2-1} x[2r](W_N^2)^{rk} + W_N^k \sum_{r=0}^{N/2-1} x[2r+1](W_N^2)^{rk}\end{aligned}$$

But $W_n^2 = (e^{-j2\pi/N})^2 = e^{-j2\pi/(N/2)} = W_{N/2}$
So $\mathbf{X}[k] = \sum_{r=0}^{N/2-1} x[2r](W_{N/2})^{rk} + W_N^k \sum_{r=0}^{N/2-1} x[2r+1](W_{N/2})^{rk}$

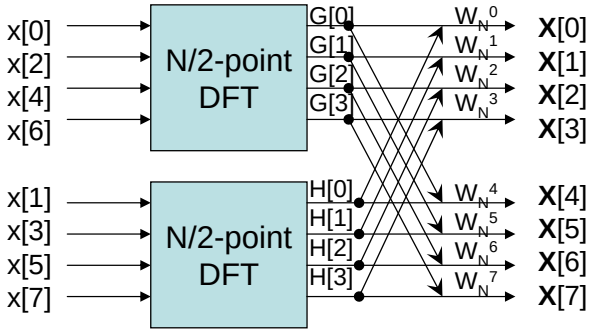
$$\mathbf{X}[k] = \underbrace{G[k]}_{N/2\text{-point DFT of even samples}} + W_N^k \underbrace{H[k]}_{N/2\text{-point DFT of odd samples}} \quad (7)$$

Equation (7) allows us to re-draw (A) as (B) (for $N = 8$, see below)

(A)

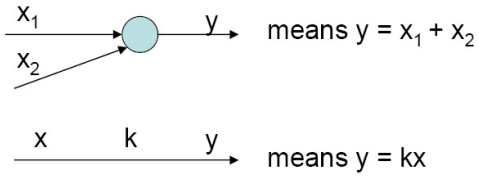


(B)

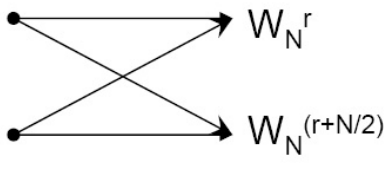


Note: $H[4] = H[0], H[5] = H[1], G[4] = G[0]$, etc.

These diagram are called flow graphs.



If $N = 2^v$(power of 2), then we can use the same trick to express an $N/2$-point DFT as two $N/4$-point DFTs. (Figure 9.5) We can keep doing this recursion until we reach a 2-point DFT, which is just a sum and difference operation (Figure 9.6). Figure 9.7 shows the complete decomposition of an 8-point DFT into these "butterfly" operations.



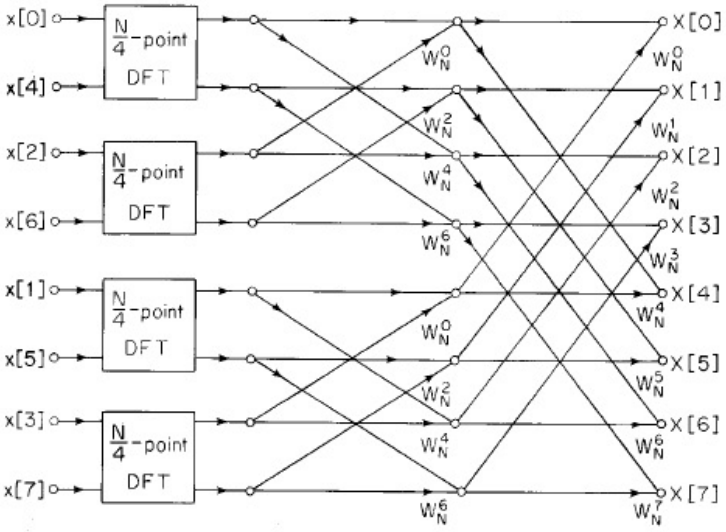


Figure 9.5 Result of substituting Fig. 9.4 into Fig. 9.3.

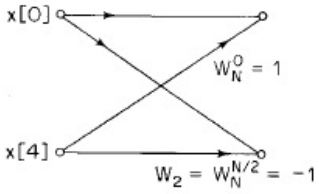


Figure 9.6 Flow graph of a 2-point DFT.

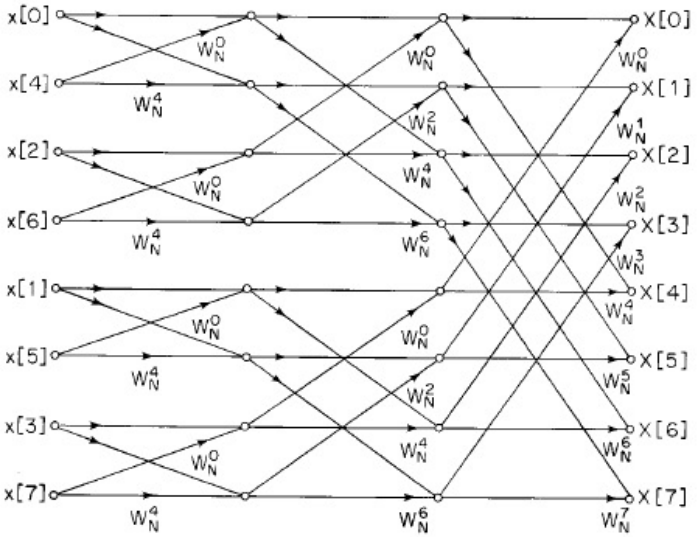


Figure 9.7 Flow graph of complete decimation-in-time decomposition of an 8-point DFT computation.