Chapter 4: Digital Filter Realizations

Outline

- Fast Fourier Transforms Digital Filter Design
- Decimation and Interpolation
- Random Signals, and Applications.

4. Fast Fourier transforms digital filter design

- We look at the computational requirements of the DFT and derive some fast algorithms for computing the DFT.
- These algorithms are known, generically, as **Fast Fourier transforms (FFTs)**
- The **radix-2** decimation in time FFT, an algorithm published in **1965** by Cooley and Tukey.

Motivation for *Fast* Fourier Transform (FFT)

- Signal processing
- Image processing
- Solving Poisson's Equation nearly optimally
- Fast multiplication of large integers

Digital Filters

- The term digital filter, or simply filter, is often used to refer to a **discrete-time system**.
- A digital filter is defined by J. E Kaiser as a ". . . **computational process** or **algorithm** by which a sampled signal or sequence of numbers (acting as the **input**) is transformed into a second sequence of numbers termed the **output signal**.
- The computational process may be that of low pass filtering (smoothing), band pass filtering, interpolation, the generation of derivatives, etc."
- **Filters** may be characterized in terms of their system properties, such as **linearity, shift-invariance, causality, stability**, etc., and they may be classified in terms of the form of their **frequency response**.

4.1. Radix-2 FFT Algorithms

The N -point DFT of an N -point sequence x(n) is

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

- **Because x(n) may be either real or complex, evaluating X(k) requires on the order of N complex multiplications and N complex additions for each value of k.**
- **There are N values of X(k), computing an N-point DFT requires N^2 complex multiplications and additions.**

4.1.1 Decimation in Time FFT

- The decimation-in-time FFT algorithm is based on splitting (decimating) x(n) into smaller sequences and finding X(k) from the DFTs of these decimated sequences.
- This section describes how this decimation leads to an efficient algorithm when the sequence length is a power of 2.
- Let x(n) be a sequence of length N = 2^v, and suppose that x(n) is split (decimated) into two subsequences, each of length N/2. As illustrated in Fig. 4-1, the first sequence, g(n), is formed from the even-index terms,

$$g(n) = x(2n) \qquad n = 0, 1, \ldots, \frac{N}{2} - 1$$

and the second, $h(n)$, is formed from the odd-index terms,

$$h(n) = x(2n+1) \qquad n = 0, 1, \ldots, \frac{N}{2} - 1$$

In terms of these sequences, the N-point DFT of $x(n)$ is

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

$$= \sum_{l=0}^{\frac{N}{2}-1} g(l) W_N^{2lk} + \sum_{l=0}^{\frac{N}{2}-1} h(l) W_N^{(2l+1)k}$$

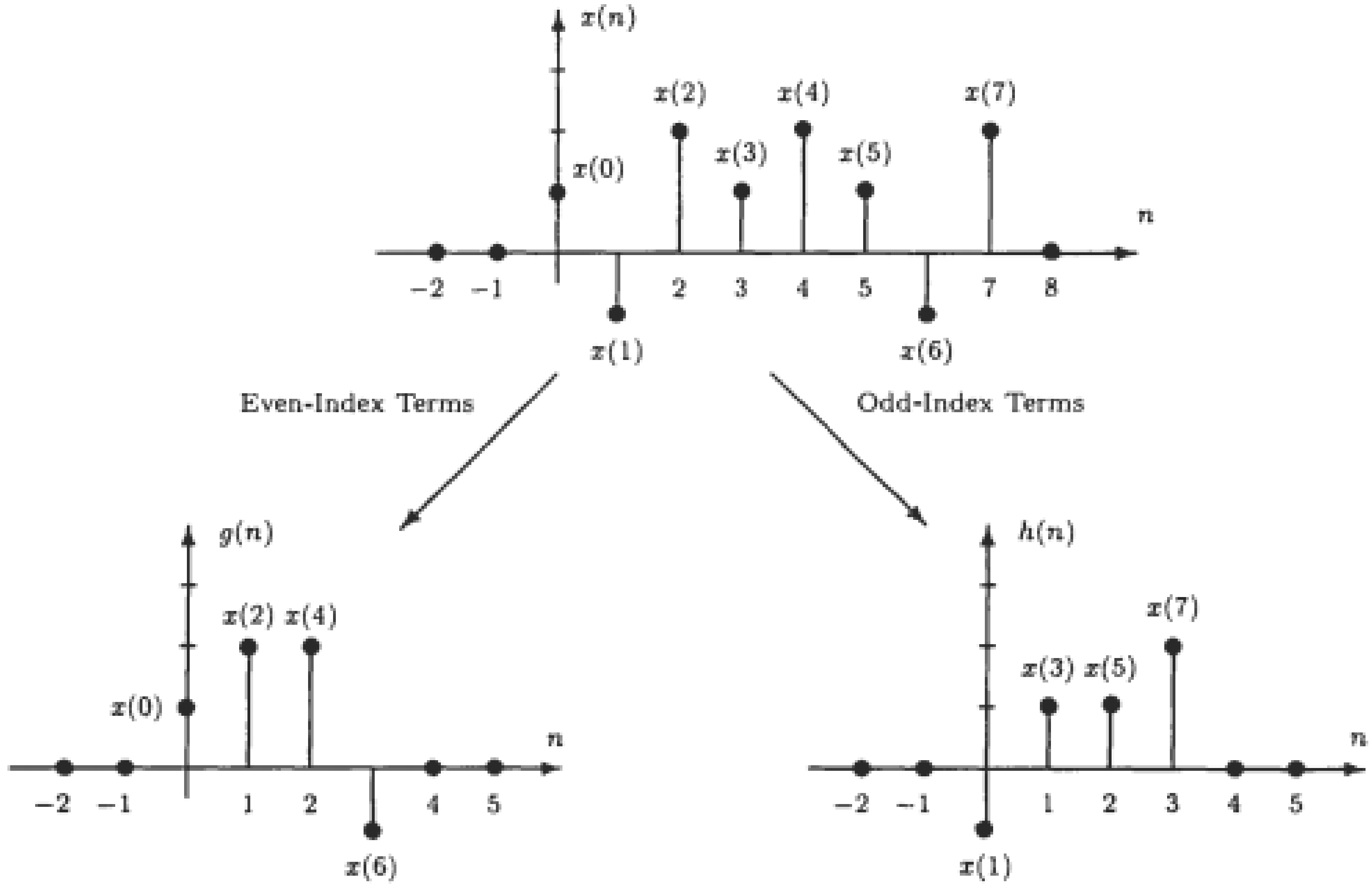


Fig. 4 -1 Decimating a sequence of length $N = 8$ by a factor of 2.

$$X(k) = \sum_{l=0}^{\frac{N}{2}-1} g(l) W_{N/2}^{lk} + W_N^k \sum_{l=0}^{\frac{N}{2}-1} h(l) W_{N/2}^{lk}$$

Note that the first term is the $N/2$-point DFT of $g(n)$, and the second is the $N/2$-point DFT of $h(n)$:

$$X(k) = G(k) + W_N^k H(k) \qquad k = 0, 1, \ldots, N-1$$

Although the $N/2$-point DFTs of $g(n)$ and $h(n)$ are sequences of length $N/2$, the periodicity of the complex exponentials allows us to write

$$G(k) = G\left(k + \frac{N}{2}\right) \qquad H(k) = H\left(k + \frac{N}{2}\right)$$

Therefore, $X(k)$ may be computed from the $N/2$-point DFTs $G(k)$ and $H(k)$. Note that because

$$W_N^{k+N/2} = W_N^k W_N^{N/2} = -W_N^k$$

then
$$W_N^{k+\frac{N}{2}} H\left(k + \tfrac{N}{2}\right) = -W_N^k H(k)$$

$$G(k) = \sum_{n=0}^{\frac{N}{2}-1} g(n) W_{N/2}^{nk} = \sum_{n \text{ even}}^{\frac{N}{2}-1} g(n) W_{N/2}^{nk} + \sum_{n \text{ odd}}^{\frac{N}{2}-1} g(n) W_{N/2}^{nk}$$

Bit Reversal

- To perform the computations in place, the input sequence x(n) must be stored (or accessed) in non sequential order.
- The shuffling of the input sequence that takes place is due to the successive decimations of x(n).
- The ordering that results corresponds to a **bit-reversed** indexing of the original sequence.
- In other words, if the index n is written in binary form, the order in which in the input sequence must be accessed is found by reading the binary representation for n in reverse order as illustrated ill the table below for N = 8:

- Alternate forms of FFT algorithms may be derived from the **decimation in time FFT** by manipulating the flow graph and rearranging the order in which the results of each stage of the computation are stored.

n	Binary	Bit-Reversed Binary	n'
0	000	000	0
1	001	100	4
2	010	010	2
3	011	110	6
4	100	001	1
5	101	101	5
6	110	011	3
7	111	111	7

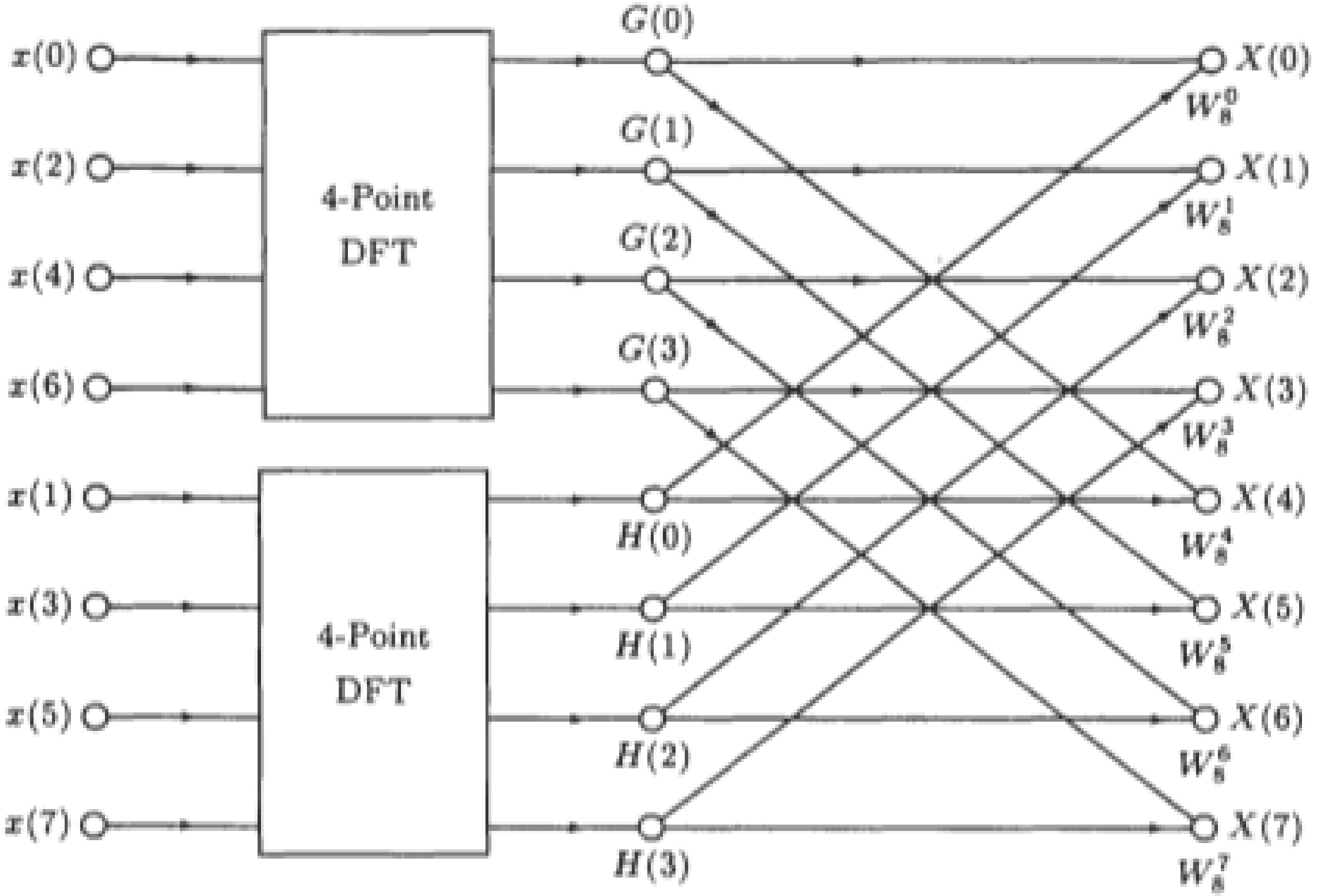
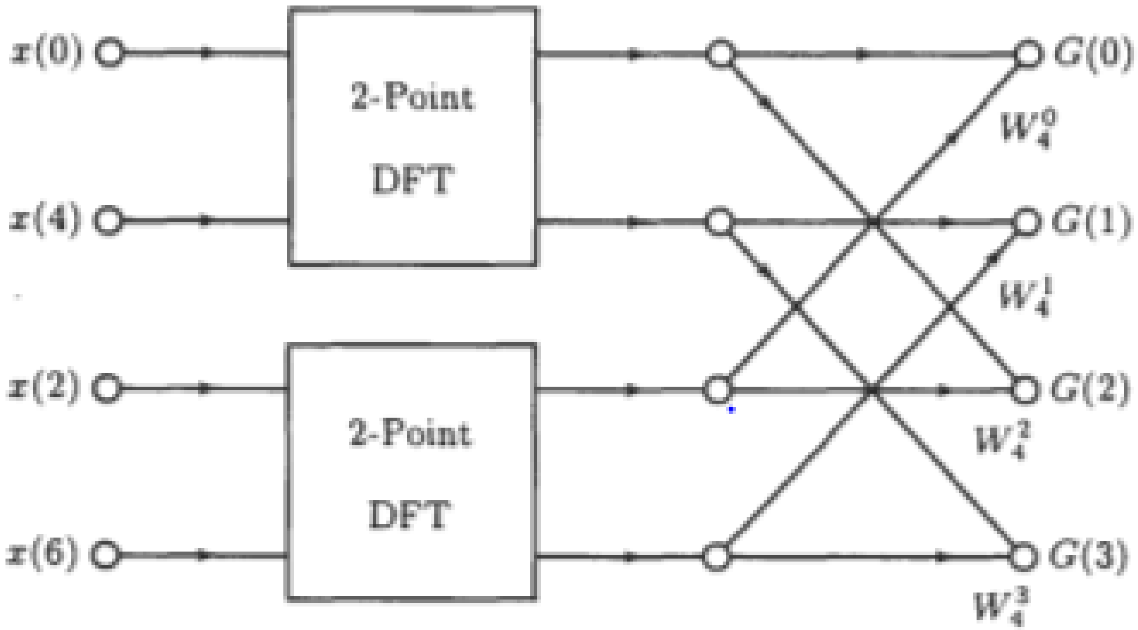


Fig. 4-2. An Eight-point decimation-in-time FFT algorithm after the first decimation.

$$G(k)=\sum_{n=0}^{\frac{N}{4}-1} g(2n)W_{N/4}^{nk}+W_{N/2}^{k}\sum_{n=0}^{\frac{N}{4}-1} g(2n+1)W_{N/4}^{nk}$$

where the first term is the N/4-point DFT of the even samples of g(n), and the second is the N/4-point DFT of the odd samples.



4-3. Decimation of the four-point DFT into two two-point DFTs in the decimation-in-time FFT.

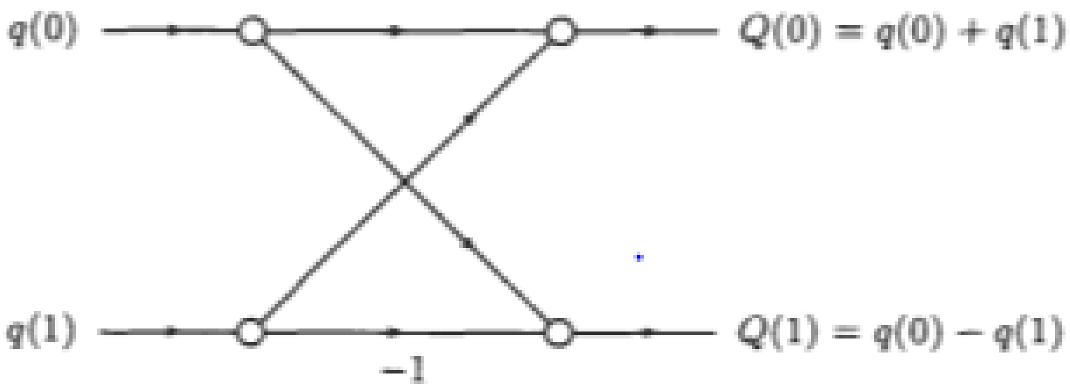
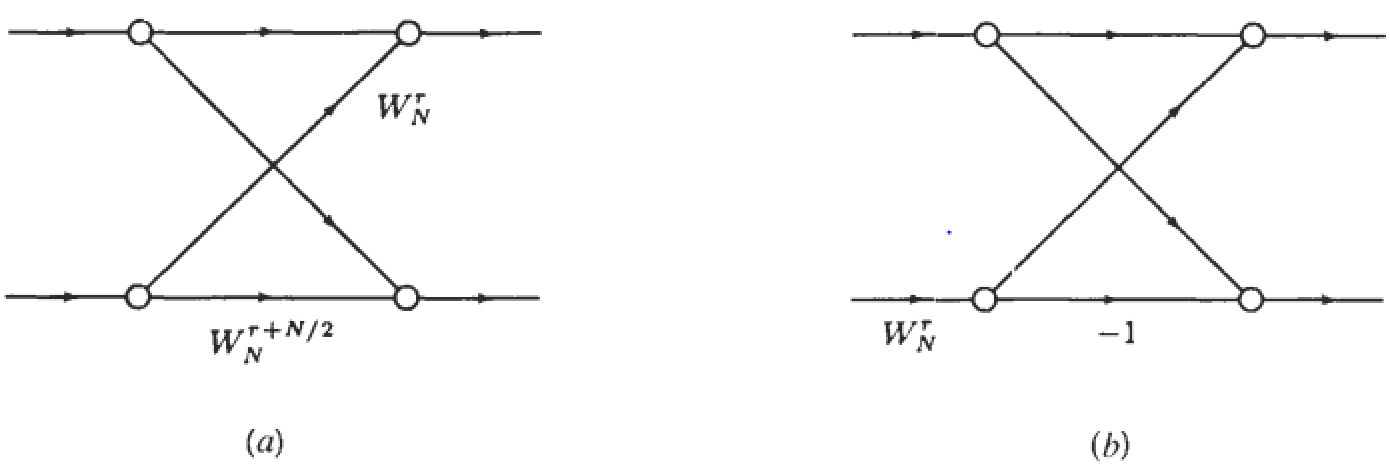


Fig. 4-4. A two-point DFT.



(a) The butterfly, which is the basic computational clement of the FFT algorithm

(b) A simplified butterfly with only one complex multiplication.

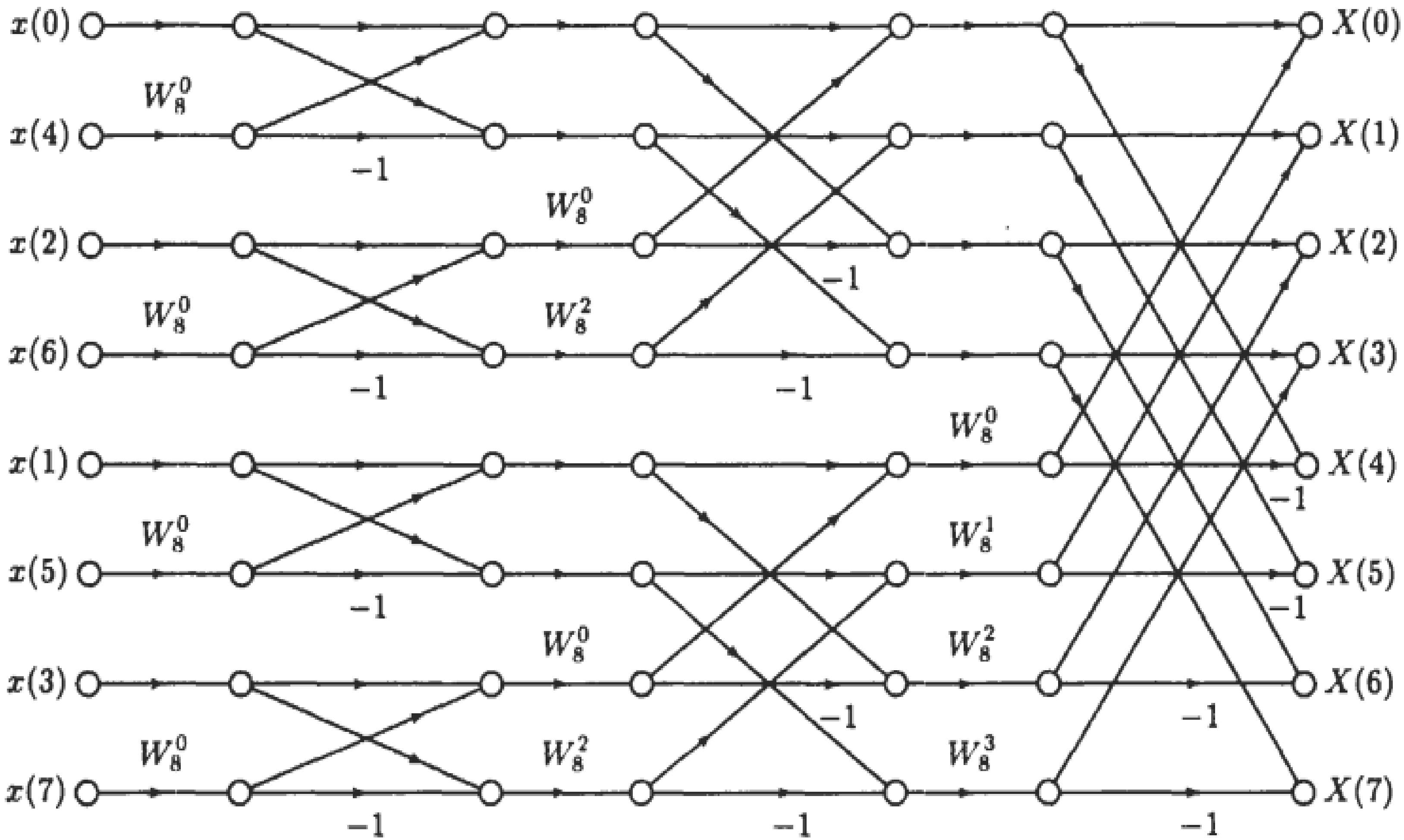


Fig. 4-5. A complete Eight-point radix-2 decimation-in-lime FFT

Computing an N-point DFT using a radix-2 decimation-in-time FFT is much more efficient than calculating the DFT directly. For example, if $N = 2^\nu$, there are $\log_2 N = \nu$ *stages* of computation. Because each stage requires $N/2$ complex multiplies by the twiddle factors W_N^r and N complex additions, there are a total of $\frac{1}{2}N \log_2 N$ complex multiplications and $N \log_2 N$ complex additions.

Example

Assume that a complex multiply takes 1 μs and that the amount of time to compute a DFT is determined by the amount of time it takes to perform all of the multiplications.

(*a*) How much time does it take to compute a 1024-point DFT directly?

(*b*) How much time is required if an FFT is used?

(*c*) Repeat parts (*a*) and (*b*) for a 4096-point DFT.

(*a*) Including possible multiplications by ± 1, computing an N-point DFT directly requires N^2 complex multiplications. If it takes 1 μs per complex multiply, the direct evaluation of a 1024-point DFT requires

$$t_{DFT} = (1024)^2 \cdot 10^{-6} \text{ s} \approx 1.05 \text{ s}$$

(*b*) With a radix-2 FFT, the number of complex multiplications is approximately $(N/2)\log_2 N$ which, for $N = 1024$, is equal to 5120. Therefore, the amount of time to compute a 1024-point DFT using an FFT is

$$t_{FFT} = 5120 \cdot 10^{-6} \text{ms} = 5.12 \text{ ms}$$

(*c*) If the length of the DFT is increased by a factor of 4 to $N = 4096$, the number of multiplications necessary to compute the DFT directly increases by a factor of 16. Therefore, the time required to evaluate the DFT directly is

$$t_{DFT} = 16.78 \text{ s}$$

If, on the other hand, an FFT is used, the number of multiplications is

$$2{,}048 \cdot \log_2 4{,}096 = 24{,}576$$

and the amount of time to evaluate the DFT is

$$t_{FFT} = 24.576 \text{ ms}$$

4.2 Decimation in Frequency FFT

- Another class of FFT algorithms may be derived by decimating the **output sequence X(k)** into smaller and smaller subsequences.
- These algorithms are called **decimation-in-frequency FFTs** and may be derived as follows.
- Let N be a power of 2, N = 2^v. and consider separately evaluating the even-index and odd-index samples of X(k). The even samples are

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

Separating this sum into the first $N/2$ points and the last $N/2$ points, and using the fact that $W_N^{2nk} = W_{N/2}^{nk}$, this becomes

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

With a change in the indexing on the second sum we have

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

Finally, because $W_{N/2}^{(n+\frac{N}{2})k} = W_{N/2}^{nk}$,

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

which is the $N/2$-point DFT of the sequence that is formed by adding the first $N/2$ points of $x(n)$ to the last $N/2$.
Proceeding in the same way for the odd samples of $X(k)$ leads to

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

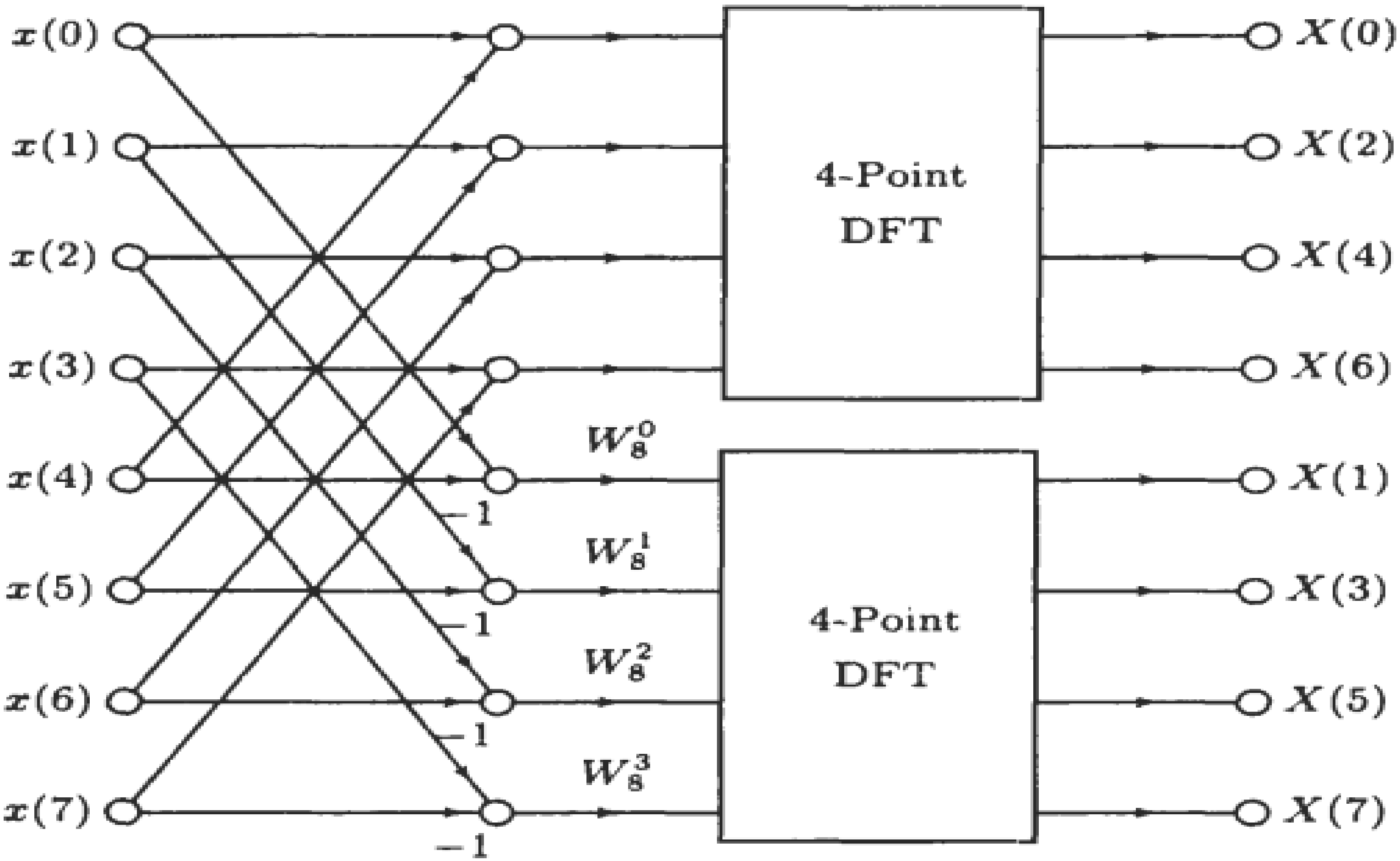


Fig. 4-6. An Eight-point decimation-in-frequency FFT algorithm after the first stage of' decimation.

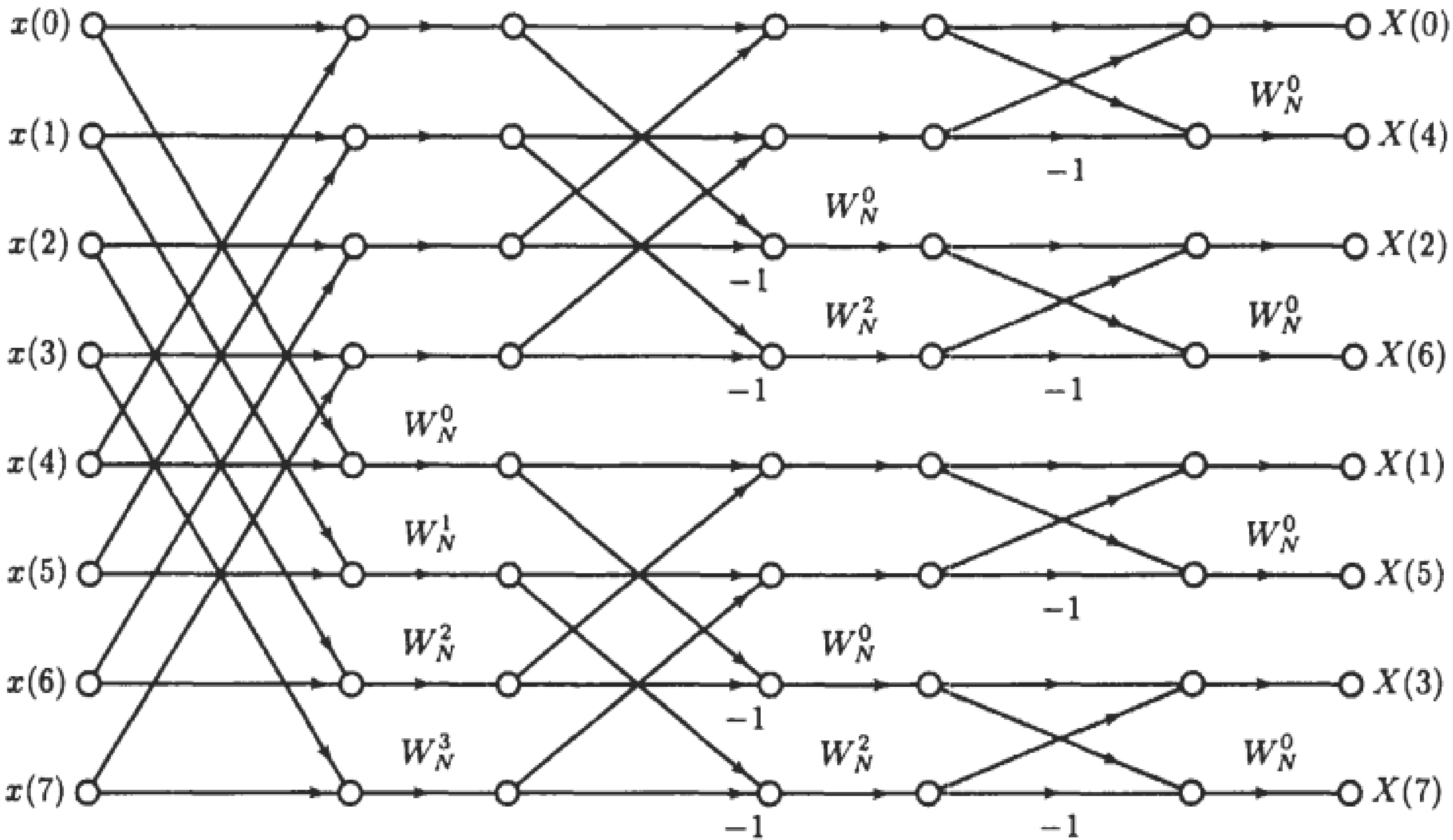
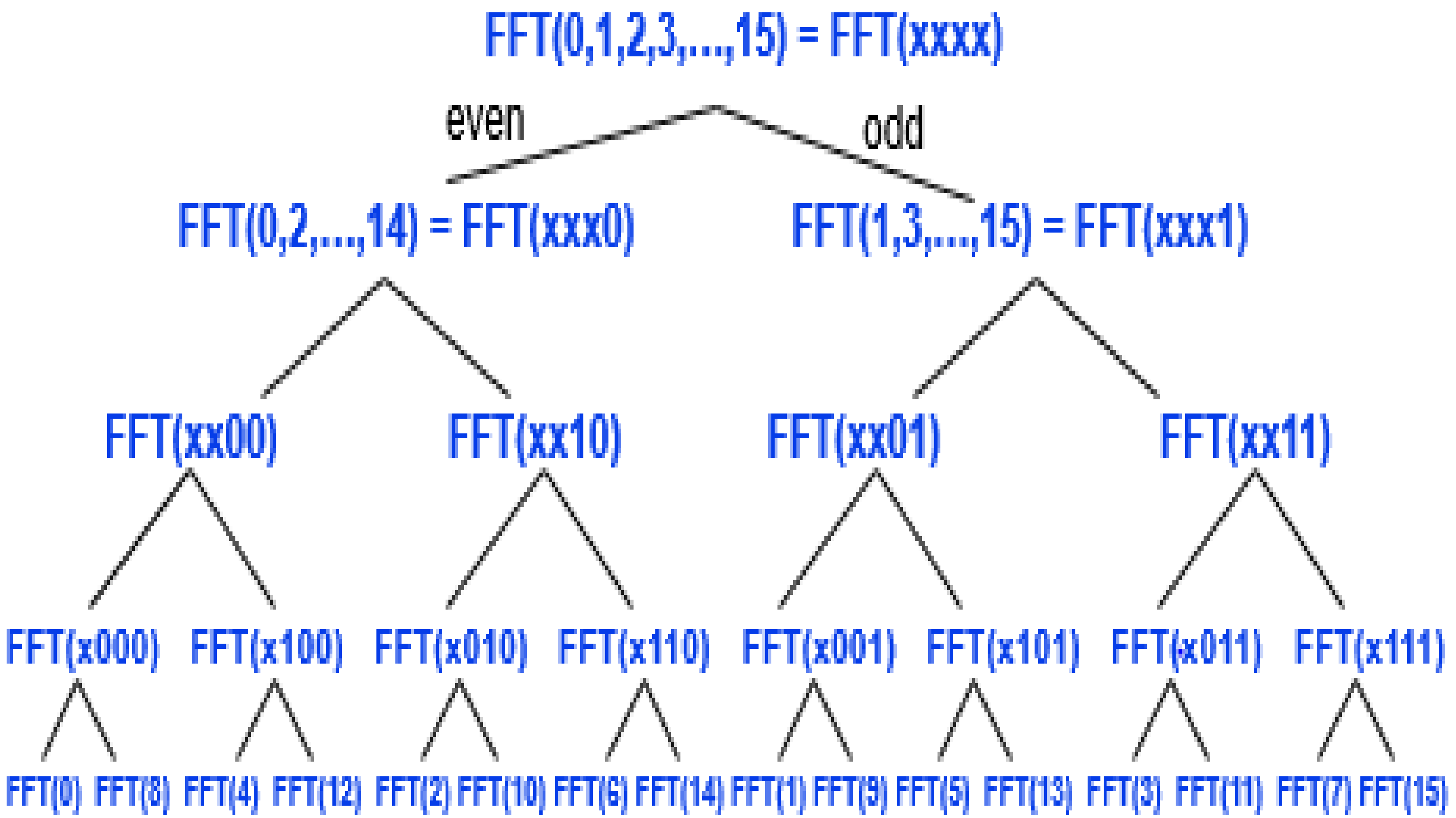
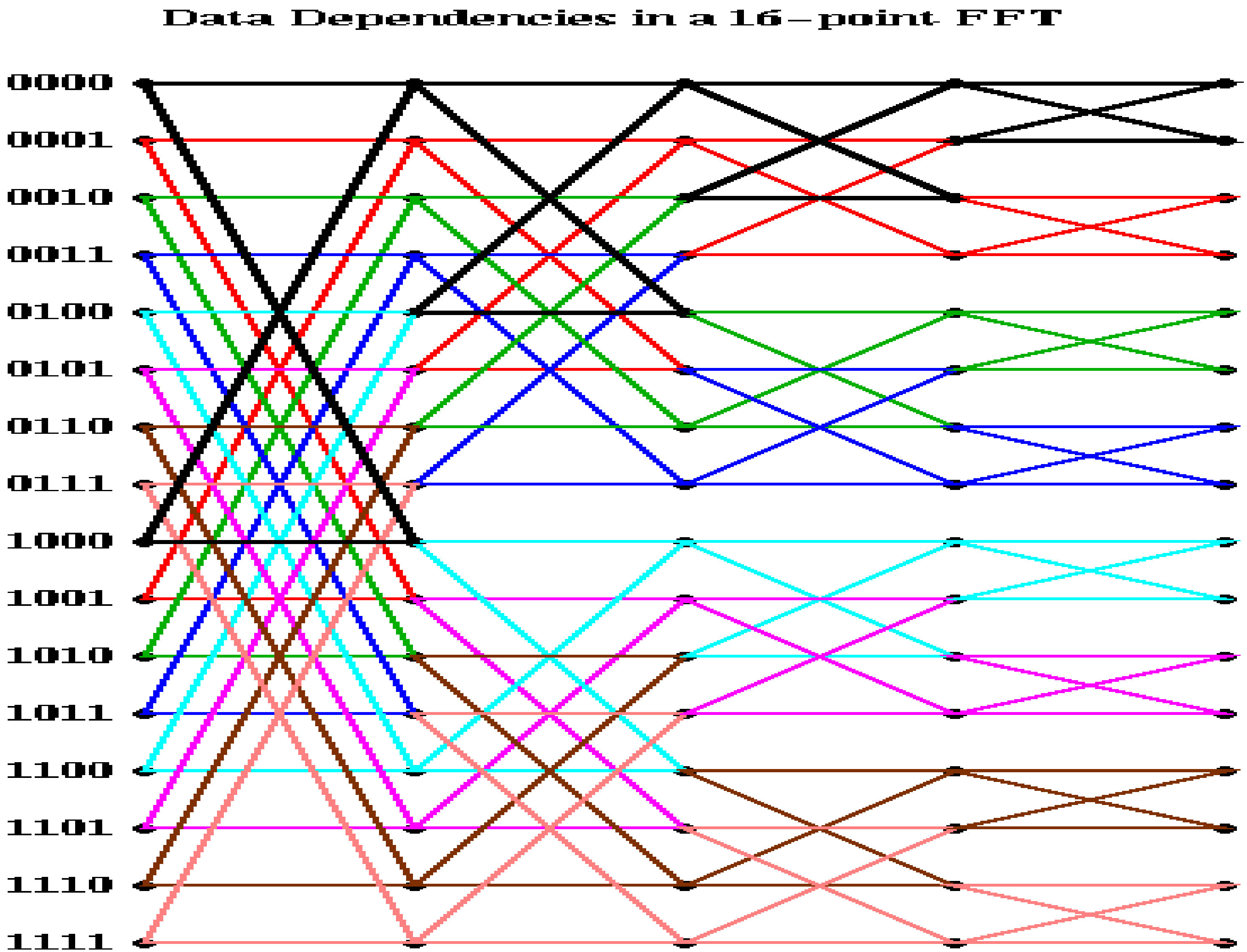


Fig. 4-7. Eight-point radix-2 decimation-in-frequency FFT.


FFT(0,1,2,3,…,15) = FFT(xxxx)
even
odd
FFT(0,2,…,14) = FFT(xxx0)
FFT(1,3,…,15) = FFT(xxx1)
FFT(xx00)
FFT(xx10)
FFT(xx01)
FFT(xx11)
FFT(x000)
FFT(x100)
FFT(x010)
FFT(x110)
FFT(x001)
FFT(x101)
FFT(x011)
FFT(x111)
FFT(0) FFT(8) FFT(4) FFT(12) FFT(2) FFT(10) FFT(6) FFT(14) FFT(1) FFT(9) FFT(5) FFT(13) FFT(3) FFT(11) FFT(7) FFT(15)


Data Dependencies in a 16−point FFT
0000
0001
0010
0011
0100
0101
0110
0111
1000
1001
1010
1011
1100
1101
1110
1111

- **Filters** may be characterized in terms of their system properties, such as linearity, shift-invariance, causality, stability, etc., and they may be classified in terms of the form of their frequency response.
- Some of these classifications are described below.

4.2.1 Linear Phase

A linear shift-invariant system is said to have linear phase if its frequency response is of the form

$$H(e^{j\omega}) = A(e^{j\omega})e^{-j\alpha\omega}$$

where α is a real number and $A(e^{j\omega})$ is a real-valued function of ω. Note that the phase of $H(e^{j\omega})$ is

$$\phi_h(\omega) = \begin{cases} -\alpha\omega & \text{when } A(e^{j\omega}) \geq 0 \\ -\alpha\omega + \pi & \text{when } A(e^{j\omega}) < 0 \end{cases}$$

Similarly, a filter is said to have *generalized linear phase* if the frequency response has the form

$$H(e^{j\omega}) = A(e^{j\omega})e^{-j(\alpha\omega-\beta)}$$

- **Thus, filters with linear phase or generalized linear phase have a constant group delay.**

4.2.2 All pass

- A system is said to be all pass filter if the frequency response magnitude is **constant**:

$$|H(e^{j\omega})| = c$$

An example of an allpass filter is the system that has a frequency response

$$H(e^{j\omega}) = \frac{e^{-j\omega} - \alpha}{1 - \alpha e^{-j\omega}}$$

where α is any real number with $|\alpha| < 1$. The unit sample response of this allpass filter is

$$h(n) = -\alpha\delta(n) + (1 - \alpha^2)\alpha^{n-1}u(n-1)$$

4.2.3 Frequency Selective Filters

- Many of the filters that are important in applications have piecewise constant frequency response magnitudes.
- These include the **low-pass, high-pass, band pass**, and **band stop** filters that are illustrated in Fig. 4-1.
- The intervals over which the frequency response magnitude is equal to **1** are called the **pass bands,** and the intervals over which it is equal to **0** are called the **stop bands**.
- The frequencies that mark the edges of the pass bands and stop bands are the cutoff frequencies.

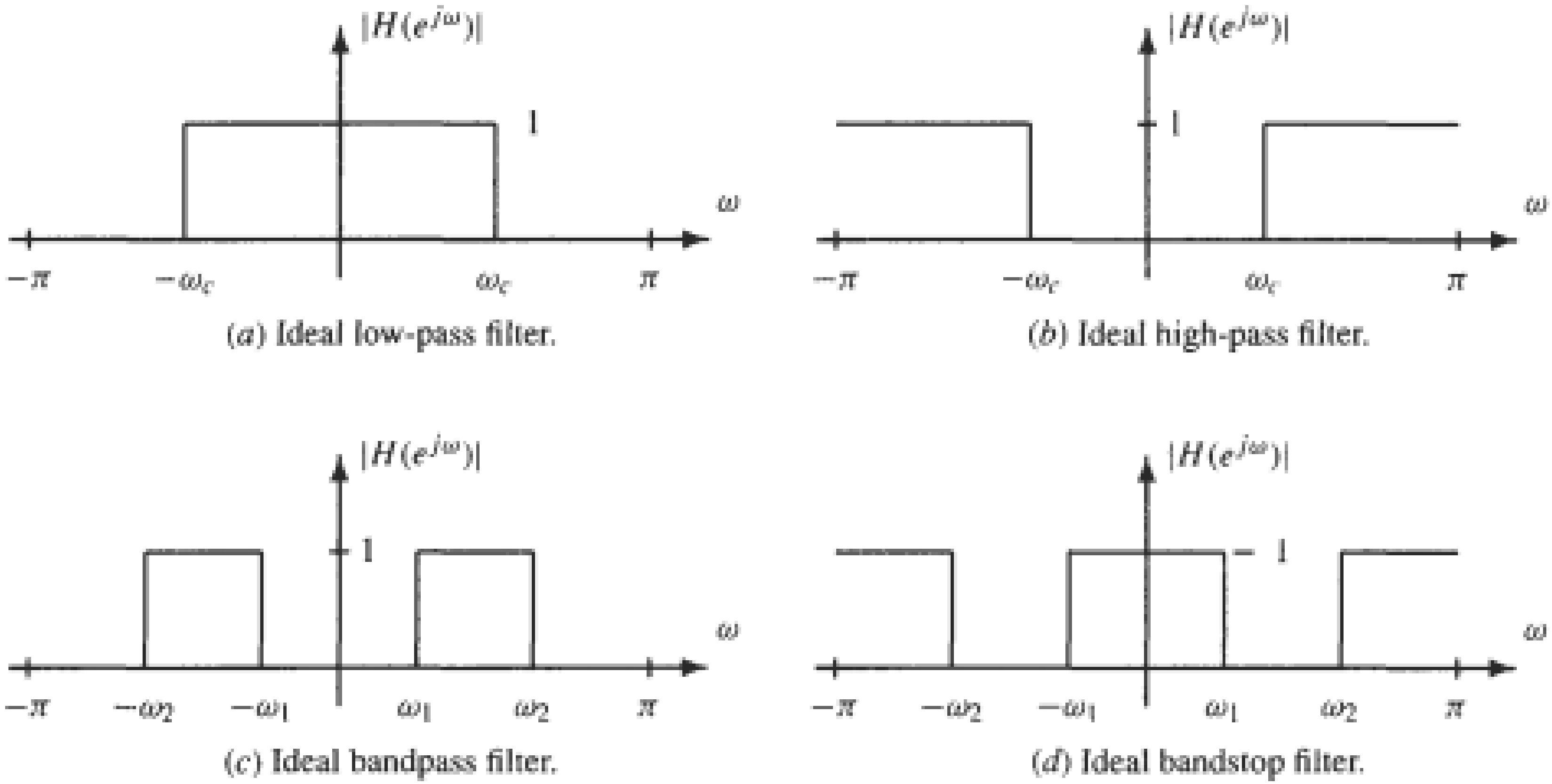
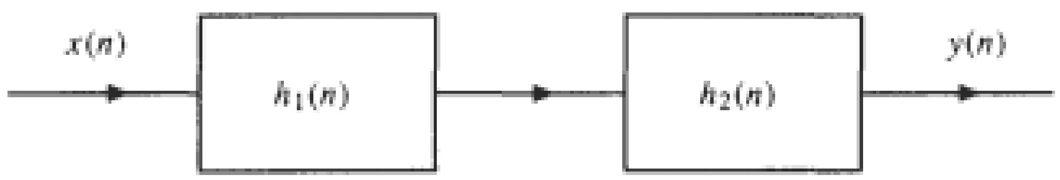


Fig. 4-1. Ideal Filters.

4.3 Interconnection of Systems

- **Filters are often interconnected to create systems that have desirable properties.**
- **Two common types of connections are series (cascade) and parallel. A cascade of two linear shift-invariant systems is shown in the figure below.**



A cascade is equivalent to a single linear shift-invariant system with a unit sample response

$$h(n) = h_1(n) * h_2(n)$$

and a frequency response

$$H(e^{j\omega}) = H_1(e^{j\omega})H_2(e^{j\omega})$$

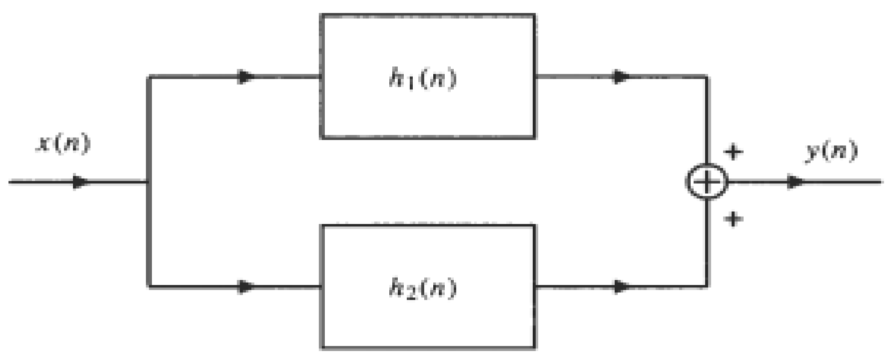
Note that the log magnitude of the cascade is the *sum* of the log magnitudes of the individual systems,

$$20\log|H(e^{j\omega})| = 20\log|H_1(e^{j\omega})| + 20\log|H_2(e^{j\omega})|$$

and the phase and group delay are additive,

$$\phi(\omega) = \phi_1(\omega) + \phi_2(\omega)$$
$$\tau(\omega) = \tau_1(\omega) + \tau_2(\omega)$$

A parallel connection of two linear shift-invariant systems is shown in the figure below.



A parallel network is equivalent to a single linear shift-invariant system with a unit sample response

$$h(n) = h_1(n) + h_2(n)$$

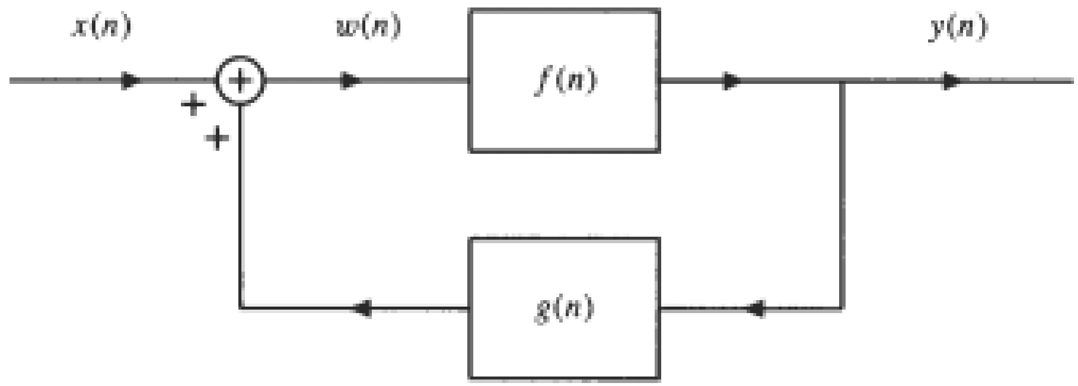
Therefore, the frequency response of the parallel network is

$$H(e^{j\omega}) = H_1(e^{j\omega}) + H_2(e^{j\omega})$$

Example 4.3 The cascade of a low-pass filter with a high-pass filter may be used to implement a band pass filter. For example, the ideal band pass filter shown in Fig. 4-1(c) may be realized by cascading a low-pass filter with cutoff frequency ω_2 with a high-pass filter that has a cutoff frequency ω_1.

Similarly, the band stop filter shown in Fig. 4-1(d) may be realized with a parallel connection of a low-pass filter with cutoff frequency ω_1 and a high-pass filter with a cutoff frequency ω_2 with $\omega_2 > \omega_1$.

Another interconnection of systems that is commonly found in control applications is the feedback network shown in the figure below.



This network may be analyzed as follows. With

$$w(n) = x(n) + g(n) * y(n)$$

and

$$y(n) = f(n) * w(n)$$

We may use the Fourier analysis techniques described in the following section to show that the frequency response of this system, if it exists, is

$$H(e^{j\omega}) = \frac{F(e^{j\omega})}{1 - F(e^{j\omega})G(e^{j\omega})}$$

4.4 Discrete Time Fourier Transform

The frequency response of a linear shift-invariant system is found by multiplying $h(n)$ by a complex exponential, $e^{-jn\omega}$, and summing over n. The discrete-time Fourier transform (DTFT) of a sequence, $x(n)$, is defined in the same way,

$$X(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x(n)e^{-jn\omega}$$

Given $X(e^{j\omega})$, the sequence $x(n)$ may be recovered using the inverse DTFT,

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

The inverse DTFT may be viewed as a decomposition of $x(n)$ into a linear combination of all complex exponentials that have frequencies in the range $-\pi < \omega \leq \pi$. Table 2-1 contains a list of some useful DTFT pairs.

Table 4-1 Some Common DTFT Pairs

Sequence	Discrete-Time Fourier Transform
$\delta(n)$	1
$\delta(n - n_0)$	$e^{-jn_0\omega}$
1	$2\pi\delta(\omega)$
$e^{jn\omega_0}$	$2\pi\delta(\omega - \omega_0)$
$a^n u(n), \quad \lvert a \rvert < 1$	$\dfrac{1}{1 - ae^{-j\omega}}$
$-a^n u(-n-1), \quad \lvert a \rvert > 1$	$\dfrac{1}{1 - ae^{-j\omega}}$
$(n+1)a^n u(n), \quad \lvert a \rvert < 1$	$\dfrac{1}{(1 - ae^{-j\omega})^2}$
$\cos n\omega_0$	$\pi\delta(\omega + \omega_0) + \pi\delta(\omega - \omega_0)$

Example 4.4 Suppose $X(e^{j\omega})$ consists of an impulse at frequency $\omega = \omega_0$:

$$X(e^{j\omega}) = \delta(\omega - \omega_0)$$

Using the inverse DTFT, we have

$$x(n) = \frac{1}{2\pi}\int_{-\pi}^{\pi} X(e^{j\omega})e^{jn\omega}d\omega = \frac{1}{2\pi}e^{jn\omega_0}$$

Note that although $x(n)$ is not absolutely summable, by allowing the DTFT to contain impulses, we may consider the DTFT of sequences that contain complex exponentials. As another example, if

$$X(e^{j\omega}) = \pi\delta(\omega - \omega_0) + \pi\delta(\omega + \omega_0)$$

computing the inverse DTFT, we find

$$x(n) = \tfrac{1}{2}e^{jn\omega_0} + \tfrac{1}{2}e^{-jn\omega_0} = \cos(n\omega_0)$$

Table 4-2 Properties of the DTFT

Property	Sequence	Discrete-Time Fourier Transform
Linearity	$ax(n) + by(n)$	$aX(e^{j\omega}) + bY(e^{j\omega})$
Shift	$x(n - n_0)$	$e^{-jn_0\omega}X(e^{j\omega})$
Time-reversal	$x(-n)$	$X(e^{-j\omega})$
Modulation	$e^{jn\omega_0}x(n)$	$X(e^{j(\omega-\omega_0)})$
Convolution	$x(n) * y(n)$	$X(e^{j\omega})Y(e^{j\omega})$
Conjugation	$x^*(n)$	$X^*(e^{-j\omega})$
Derivative	$nx(n)$	$j\dfrac{dX(e^{j\omega})}{d\omega}$
Multiplication	$x(n)y(n)$	$\dfrac{1}{2\pi}\int_{-\pi}^{\pi} X(e^{j\theta})Y(e^{j(\omega-\theta)})d\theta$

Note: Given the DTFTs $X(e^{j\omega})$ and $Y(e^{j\omega})$ of $x(n)$ and $y(n)$, this table lists the DTFTs of sequences that are formed from $x(n)$ and $y(n)$.

Applications

- Some applications of the DTFT in discrete-time signal analysis.
- Finding the frequency response of an LSI system that is described by a difference equation,
- Performing convolutions,
- Solving difference equations that have zero initial conditions, and
- Designing inverse systems.

Chapter 5: FIR and IIR Filters

Outline

- Filter Design Methods
- Interpolation and Decimation

4.2 Filter Design

- **Filter design** process begins with the **filter specifications**, which may include constraints on the magnitude and/or phase of the frequency response, constraints on the unit sample response or step response of the filter, specification of the type of filter (e.g., finite-length impulse response (FIR) or IIR), and the filter order.
- Once the specifications have been defined,
- The next step is to find a set of filter coefficients that produce an acceptable filter.
- After the filter has been designed, the last step is to implement the system in hardware or software, quantizing the filter coefficients if necessary, and choosing an appropriate filter structure

2.2.1 Filter specifications

- Before a filter can be designed, a set of filter specifications must be defined.
- For example, suppose that we would like to design a low-pass filter with a **cutoff frequency** ω_c.
- The frequency response of an ideal low-pass filter with linear phase and a cutoff frequency ω_c is

$$H_d(e^{j\omega}) = \begin{cases} e^{-j\alpha\omega} & |\omega| \le \omega_c \\ 0 & \omega_c < |\omega| \le \pi \end{cases}$$

which has a unit sample response

$$h_d(n) = \frac{\sin(n-\alpha)\omega_c}{\pi(n-\alpha)}$$

- Because this filter is unrealizable (non causal and unstable), it is necessary to relax the ideal constraints on the frequency response and allow some deviation from the ideal response.
- The specifications for a low-pass filter will typically have the form

$$1-\delta_p < |H(e^{j\omega})| \leq 1+\delta_p \qquad 0 \leq |\omega| < \omega_p$$

$$|H(e^{j\omega})| \leq \delta_s \qquad \omega_s \leq |\omega| < \pi$$

- As illustrated in Fig. 4-1. Thus, the specifications include the **pass band cutoff frequency, ω_p the stop band cutoff frequency, ω_s the pass band deviation, δ_p. and the stop band deviation, δ_s.**
- The pass band and stop band deviations are often given in decibels (dB) as follows: $\alpha_p = -20\log(1-\delta_p)$

$$\alpha_s = -20\log(\delta_s)$$

- The interval [ω_p, ω_s] is called the transition hand.
- Once the filter specifications have been defined, the next step is to design a filter that meets these specifications.

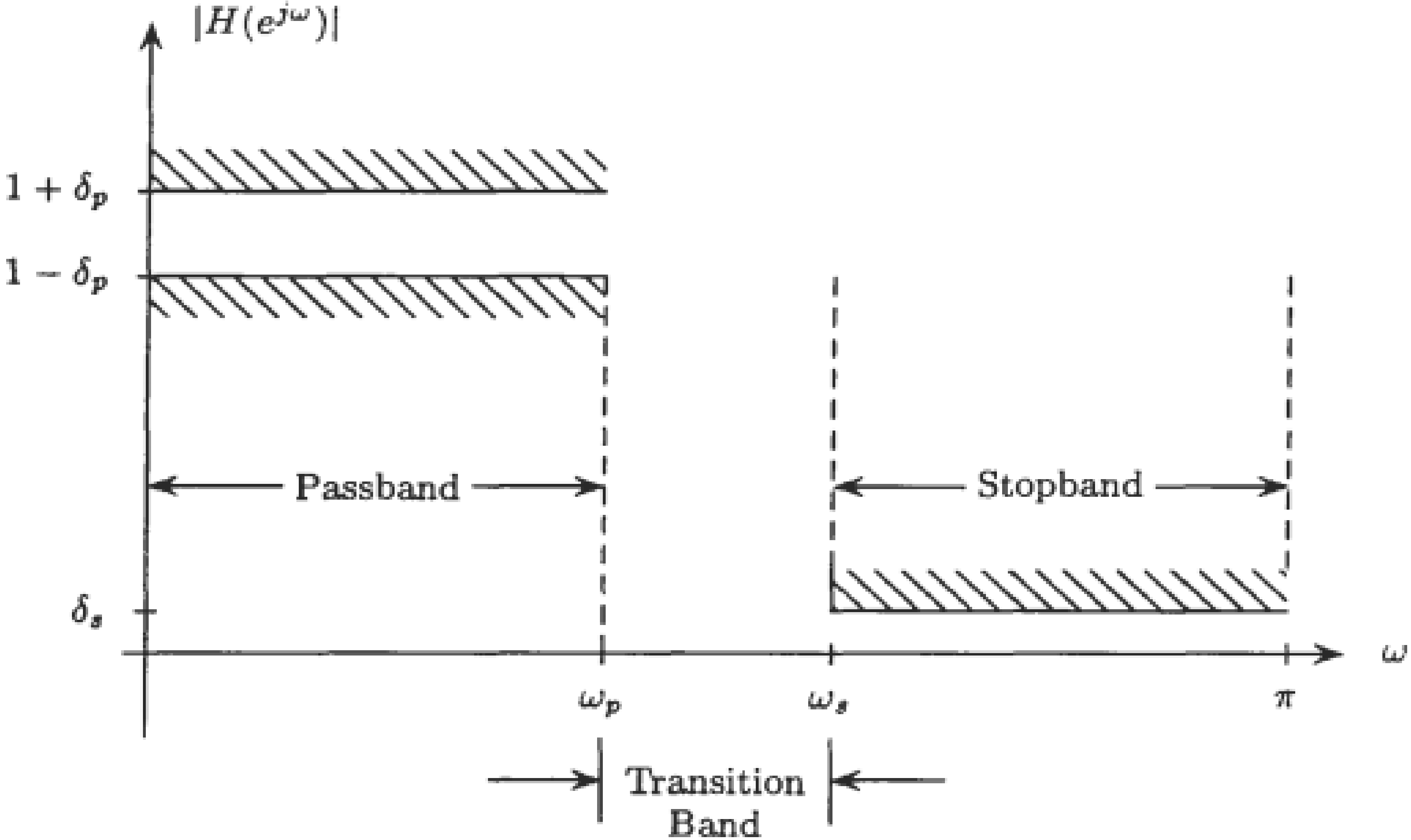


Fig. 4-1. Filter specifications for a low-pass filter,

4.3 FIR Filter Design

The frequency response of an Nth-order causal FIR filter is

$$H(e^{j\omega}) = \sum_{n=0}^{N} h(n)e^{-jn\omega}$$

- The design of an FIR filter involves finding the coefficients h(n) that result in a frequency response that satisfies a given set of filter specifications.
- **FIR filters** have two important advantages over **IIR filters**. First, they are guaranteed to be stable, even after the filter coefficients have been quantized.
- Second, they may be easily constrained to have (generalized) linear phase. Because FIR filters are generally designed to have linear phase, in the following we consider the design of linear phase FIR filters.

4.3.1 Linear Phase FIR Design Using Windows

Let $h_d(n)$ be the unit sample response of an ideal frequency selective filter with linear phase,

$$H_d(e^{j\omega}) = A(e^{j\omega})e^{-j(\alpha\omega-\beta)}$$

Because $h_d(n)$ will generally be infinite in length, it is necessary to find an FIR approximation to $H_d(e^{j\omega})$. With the window design method, the filter is designed by windowing the unit sample response,

$$h(n) = h_d(n)w(n)$$

where $w(n)$ is a finite-length window that is equal to zero outside the interval $0 \le n \le N$ and is symmetric about its midpoint:

$$w(n) = w(N-n)$$

The effect of the window on the frequency response may be seen from the complex convolution theorem,

$$H(e^{j\omega}) = \frac{1}{2\pi}H_d(e^{j\omega}) * W(e^{j\omega}) = \frac{1}{2\pi}\int_{-\pi}^{\pi} H_d(e^{j\theta})W(e^{j(\omega-\theta)})\,d\theta$$

- The ideal frequency response is smoothed by the discrete-time Fourier transform of the window, $W(e^{j\omega})$.
- There are many different types of windows that may be used in the window design method, a few of which are listed in Table 4-1.

- How well the frequency response of a filter designed with the window design method approximates a desired response, $H_d(e^{j\omega})$, is determined by two factors (see Fig. 4-2):

1. The width of the main lobe of $W(e^{j\omega})$.
2. The peak side-lobe amplitude of $W(e^{j\omega})$.

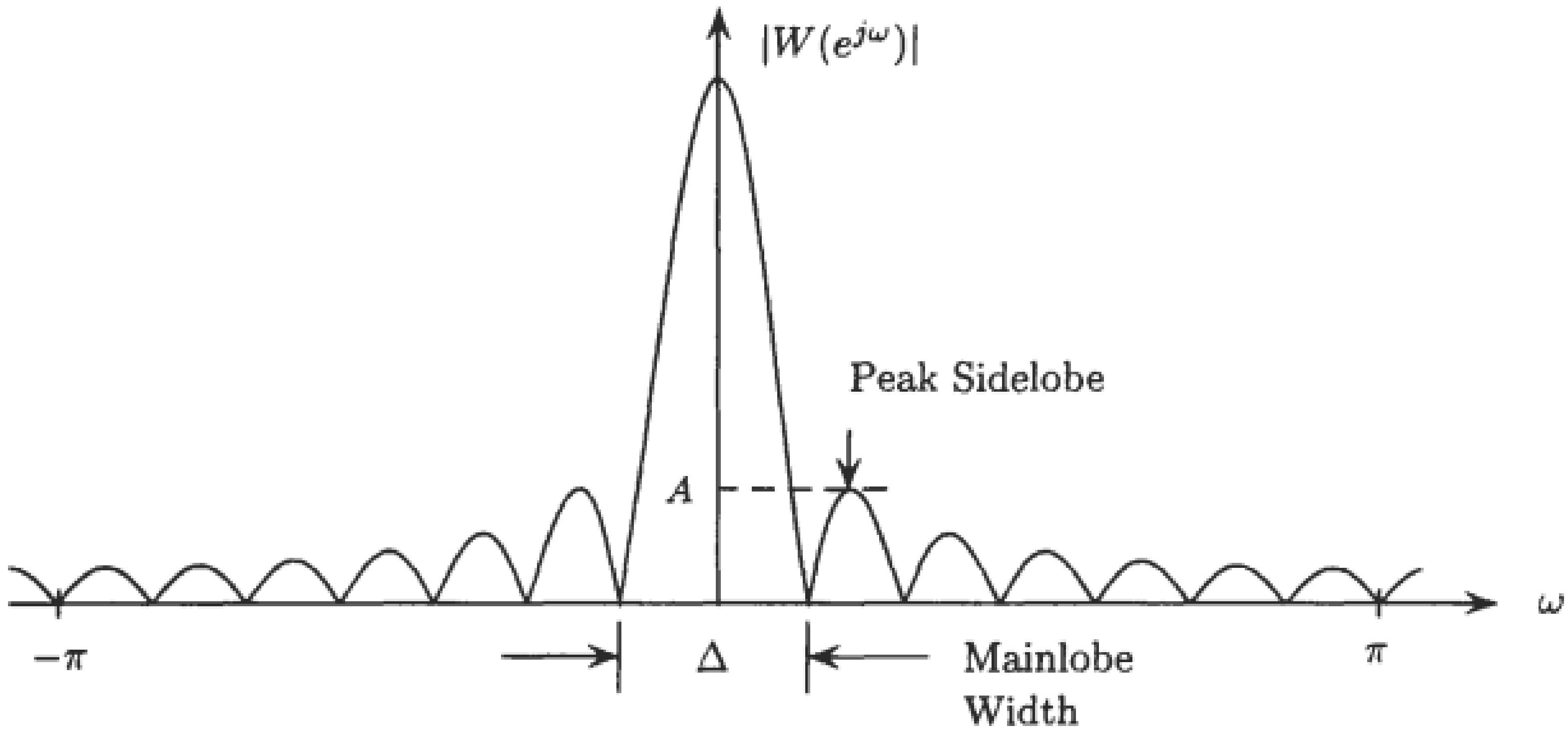


Fig. 4-2. The DTFT of a typical window, which is characterized by the width of its main lobe. A. and the peak amplitude of its side lobes, A, relative to the amplitude of $W(e^{j\omega})$ at $\omega = 0$.

- Ideally, the main-lobe width should be narrow, and the side-lobe amplitude should be small.
- However, for a fixed-length window, these cannot be minimized independently. Some general properties of windows are as follows:

1. As the length N of the window increases, the width of the main lobe decreases, which results in a decrease in the transition width between pass bands and stop bands. This relationship is given approximately by

$$N \Delta f = c$$

- ✓ where Δf is the transition width, and c is a parameter that depends on the window.

2. The peak side-lobe amplitude of the window is determined by the shape of the window, and it is essentially independent of the window length.
3. If the window shape is changed to decrease the side-lobe amplitude, the width of the main lobe will generally increase.

❖ Listed in Table 4.2 are the side-lobe amplitudes of several windows along with the approximate transition width and stop band attenuation that results when the given window is used to design an Nth-order low-pass filter.

Table 4-1 Some Common Windows

Rectangular	$w(n) = \begin{cases} 1 & 0 \le n \le N \\ 0 & \text{else} \end{cases}$
Hanning[1]	$w(n) = \begin{cases} 0.5 - 0.5\cos\left(\frac{2\pi n}{N}\right) & 0 \le n \le N \\ 0 & \text{else} \end{cases}$
Hamming	$w(n) = \begin{cases} 0.54 - 0.46\cos\left(\frac{2\pi n}{N}\right) & 0 \le n \le N \\ 0 & \text{else} \end{cases}$
Blackman	$w(n) = \begin{cases} 0.42 - 0.5\cos\left(\frac{2\pi n}{N}\right) + 0.08\cos\left(\frac{4\pi n}{N}\right) & 0 \le n \le N \\ 0 & \text{else} \end{cases}$

In the literature, this window is also called a Hann window or a von Hann window

❖ **Table 4-2 The Peak Side-Lobe Amplitude of Some Common Windows and the Approximate. Transition Width and Stop band Attenuation of an Nth-Order Low-Pass Filter Designed Using the Given Window.**

Window	Side-Lobe Amplitude (dB)	Transition Width (Δf)	Stopband Attenuation (dB)
Rectangular	−13	$0.9/N$	−21
Hanning	−31	$3.1/N$	−44
Hamming	−41	$3.3/N$	−53
Blackman	−57	$5.5/N$	−74

Example 4.3.1 Suppose that we would like to design an FIR linear phase low-pass filter according to the following specifications:

$$0.99 \le |H(e^{j\omega})| \le 1.01 \qquad 0 \le |\omega| \le 0.19\pi$$

$$|H(e^{j\omega})| \le 0.01 \qquad 0.21\pi \le |\omega| \le \pi$$

For a stopband attenuation of $20\log(0.01) = -40$ dB, we may use a Hanning window. Although we could also use a Hamming or a Blackman window, these windows would overdesign the filter and produce a larger stopband attenuation at the expense of an increase in the transition width. Because the specification calls for a transition width of $\Delta\omega = \omega_s - \omega_p = 0.02\pi$, or $\Delta f = 0.01$, with

$$N\,\Delta f = 3.1$$

for a Hanning window (see Table 9.2), an estimate of the required filter order is

$$N = \frac{3.1}{\Delta f} = 310$$

The last step is to find the unit sample response of the ideal low-pass filter that is to be windowed. With a cutoff frequency of $\omega_c = (\omega_s + \omega_p)/2 = 0.2\pi$, and a delay of $\alpha = N/2 = 155$, the unit sample response is

$$h_d(n) = \frac{\sin[0.2\pi(n - 155)]}{(n - 155)\pi}$$

In addition to the windows listed in Table 9-1, Kaiser developed a *family* of windows that are defined by

$$w(n) = \frac{I_0[\beta(1 - [(n-\alpha)/\alpha]^2)^{1/2}]}{I_0(\beta)} \qquad 0 \le n \le N$$

where $\alpha = N/2$, and $I_0(\cdot)$ is a zeroth-order modified Bessel function of the first kind, which may be easily generated using the power series expansion

$$I_0(x) = 1 + \sum_{k=1}^{\infty}\left[\frac{(x/2)^k}{k!}\right]^2$$

- The parameter β determines the shape of the window and thus controls the trade-off between main-lobe width and side-lobe amplitude.
- A Kaiser window is nearly optimum in the sense of having the most energy in its main lobe for a given side-lobe amplitude. Table 4-3 illustrates the effect of changing the parameter β.

- There are two empirically derived relationships for the Kaiser window that facilitate the use of these windows to design FIR filters.
- The first relates the stop band ripple of a low-pass filter,

$\alpha_s = -20\log(\delta_s)$, to the parameter β,

$$\beta = \begin{cases} 0.1102(\alpha_s - 8.7) & \alpha_s > 50 \\ 0.5842(\alpha_s - 21)^{0.4} + 0.07886(\alpha_s - 21) & 21 \le \alpha_s \le 50 \\ 0.0 & \alpha_s < 21 \end{cases}$$

Table 4-3 Characteristics of the Kaiser Window as a Function of β

Parameter β	Side Lobe (dB)	Transition Width ($N\Delta f$)	Stopband Attenuation (dB)
2.0	−19	1.5	−29
3.0	−24	2.0	−37
4.0	−30	2.6	−45
5.0	−37	3.2	−54
6.0	−44	3.8	−63
7.0	−51	4.5	−72
8.0	−59	5.1	−81
9.0	−67	5.7	−90
10.0	−74	6.4	−99

The second relates N to the transition width Δf and the stopband attenuation α_s,

$$N = \frac{\alpha_s - 7.95}{14.36\Delta f} \qquad \alpha_s \geq 21$$

Note that if $\alpha_s < 21$ dB, a rectangular window may be used ($\beta = 0$), and $N = 0.9/\Delta f$.

Example 4.3.2

Suppose that we would like to design a low-pass filter with a cutoff frequency $\omega_c = \pi/4$, a transition width $\Delta\omega = 0.02\pi$, and a stopband ripple $\delta_s = 0.01$. Because $\alpha_s = -20\log(0.01) = -40$, the Kaiser window parameter is

$$\beta = 0.5842(40-21)^{0.4} + 0.07886(40-21) = 3.4$$

With $\Delta f = \Delta\omega/2\pi = 0.01$, we have

$$N = \frac{40-7.95}{14.36\cdot(0.01)} = 224$$

Therefore,

$$h(n) = h_d(n)w(n)$$

where

$$h_d(n) = \frac{\sin[(n-112)\pi/4]}{(n-112)\pi}$$

is the unit sample response of the ideal low-pass filter.

- Although it is simple to design a filter using the window design method, there are some limitations with this method.
- **First**, it is necessary to find a closed-form expression for $h_d(n)$ (or it must be approximated using a very long DFT).
- **Second**, for a frequency selective filter, the transition widths between frequency bands, and the ripples with in these bands, will be approximately the same.
- As a result, the window design method requires that the filter be designed to the tightest tolerances in all of the bands by selecting the smallest transition width and the smallest ripple.
- **Finally**, window design filters are not, in general, optimum in the sense that they do not have the smallest possible ripple for a given filter order and a given set of cutoff frequencies.

4.3.2 Frequency Sampling Filter Design

Another method for FIR filter design is the frequency sampling approach. In this approach, the desired frequency response, $H_d(e^{j\omega})$, is first uniformly sampled at N equally spaced points between 0 and 2π:

$$H(k) = H_d\left(e^{j2\pi k/N}\right) \qquad k = 0, 1, \ldots, N-1$$

These frequency samples constitute an N-point DFT, whose inverse is an FIR filter of order $N-1$:

$$h(n) = \frac{1}{N}\sum_{k=0}^{N-1} H(k)e^{j2\pi nk/N} \qquad 0 \leq n \leq N-1$$

The relationship between $h(n)$ and $h_d(n)$ (see Chap. 3) is

$$h(n) = \sum_{k=-\infty}^{\infty} h_d(n+kN) \qquad 0 \leq n \leq N-1$$

- Although the frequency samples match the ideal frequency response exactly, there is no control on how the samples are interpolated between the samples.
- Because filters designed with the frequency sampling method are not generally very good, this method is often modified by introducing one or more transition samples as illustrated in Fig. 4-3.
- These transition samples are optimized in an iterative manner to maximize the stop band attenuation or minimize the pass band ripple.

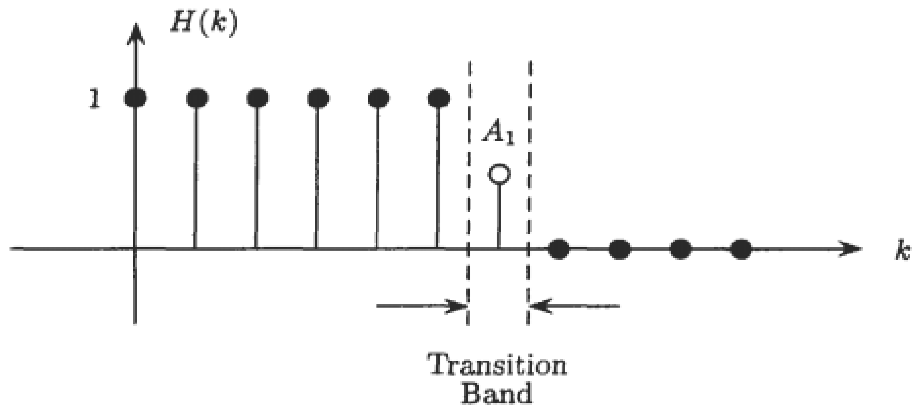


Fig. 4-3. Introducing a transition sample with an amplitude of A_1 in the frequency sampling method.

4.3.3 Equiripple Linear Phase Filters

The design of an FIR low-pass filter using the window design technique is simple and generally results in a filter with relatively good performance. However, in two respects, these filters are not optimal:

1. First, the passband and stopband deviations, δ_p and δ_s, are approximately equal. Although it is common to require δ_s to be much smaller than δ_p, these parameters cannot be independently controlled in the window design method. Therefore, with the window design method, it is necessary to *overdesign* the filter in the passband in order to satisfy the stricter requirements in the stopband.
2. Second, for most windows, the ripple is not uniform in either the passband or the stopband and generally decreases when moving away from the transition band. Allowing the ripple to be uniformly distributed over the entire band would produce a smaller *peak ripple*.

An equiripple linear phase filter, on the other hand, is optimal in the sense that the magnitude of the ripple is minimized in all bands of interest for a given filter order, N. In the following discussion, we consider the design of a type I linear phase filter. The results may be easily modified to design other types of linear phase filters.

The frequency response of an FIR linear phase filter may be written as

$$H(e^{j\omega}) = A(e^{j\omega})e^{-j\alpha\omega}$$

where the amplitude, $A(e^{j\omega})$, is a real-valued function of ω. For a type I linear phase filter,

$$h(n) = h(N - n)$$

where N is an even integer. The symmetry of $h(n)$ allows the frequency response to be expressed as

$$A(e^{j\omega}) = \sum_{k=0}^{L} a(k)\cos(k\omega)$$

where $L = N/2$ and

$$a(0) = h\left(\frac{N}{2}\right)$$

$$a(k) = h\left(k + \frac{N}{2}\right) \qquad k = 1, 2, \ldots, \frac{N}{2}$$

The terms $\cos(k\omega)$ may be expressed as a sum of powers of $\cos\omega$ in the form

$$\cos(k\omega) = T_k(\cos\omega)$$

where $T_k(x)$ is a kth-order Chebyshev polynomial [see Eq. (9.9)]. Therefore, Eq. (9.4) may be written as

$$A(e^{j\omega}) = \sum_{k=0}^{L} \alpha(k)(\cos\omega)^k$$

Thus, $A(e^{j\omega})$ is an Lth-order polynomial in $\cos\omega$.

With $A_d(e^{j\omega})$ a desired amplitude, and $W(e^{j\omega})$ a positive weighting function, let

$$E(e^{j\omega}) = W(e^{j\omega})[A_d(e^{j\omega}) - A(e^{j\omega})]$$

be a weighted approximation error. The equiripple filter design problem thus involves finding the coefficients $a(k)$ that minimize the maximum absolute value of $E(e^{j\omega})$ over a set of frequencies, $\mathcal{F}$,

$$\min_{a(k)} \left\{ \max_{\omega \in \mathcal{F}} |E(e^{j\omega})| \right\}$$

For example, to design a low-pass filter, the set $\mathcal{F}$ will be the frequencies in the passband, $[0, \omega_p]$, and the stopband, $[\omega_s, \pi]$, as illustrated in Fig. 9-4. The transition band, (ω_p, ω_s), is a *don't care* region, and it is not

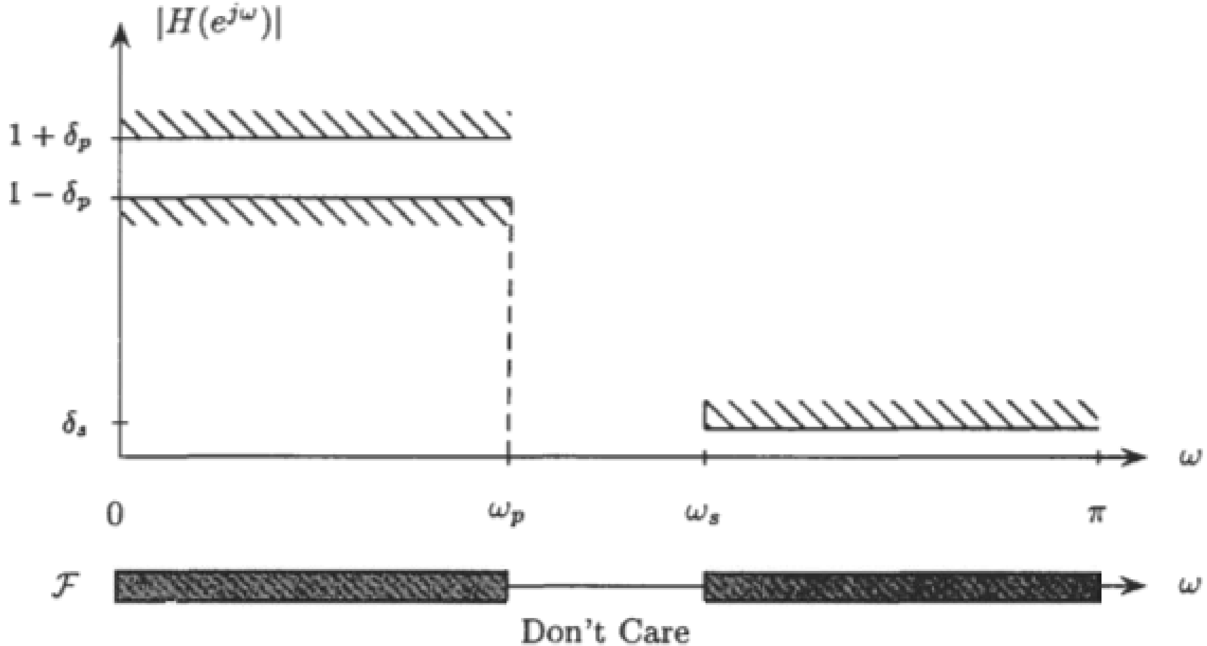


Fig. 4-4. The set R in the equiripple filter design problem, consisting of the pass band

$[0, \omega_p]$ and the stopband $[\omega_p, \omega_s]$. The transition band (ω_p, ω_s) is a don't care region.

considered in the minimization of the weighted error. The solution to this optimization problem is given in the *alternation theorem*, which is as follows:

Alternation Theorem: Let $\mathcal{F}$ be a union of closed subsets over the interval $[0, \pi]$. For a positive weighting function $W(e^{j\omega})$, a necessary and sufficient condition for

$$A(e^{j\omega}) = \sum_{k=0}^{L} a(k)\cos(k\omega)$$

to be the unique function that minimizes the maximum value of the weighted error $|E(e^{j\omega})|$ over the set $\mathcal{F}$ is that the $E(e^{j\omega})$ have at least $L+2$ *alternations*. That is to say, there must be at least $L+2$ *extremal frequencies*,

$$\omega_0 < \omega_1 < \cdots < \omega_{L+1}$$

over the set $\mathcal{F}$ such that

$$E(e^{j\omega_k}) = -E(e^{j\omega_{k+1}}) \qquad k = 0, 1, \ldots, L$$

and

$$|E(e^{j\omega_k})| = \max_{\omega\in\mathcal{F}} |E(e^{j\omega})| \qquad k = 0, 1, \ldots, L+1$$

Thus, the alternation theorem states that the optimum filter is equiripple. Although the alternation theorem specifies the minimum number of extremal frequencies (or ripples) that the optimum filter must have, it may have more. For example, a low-pass filter may have either $L+2$ or $L+3$ extremal frequencies. A low-pass filter with $L+3$ extrema is called an *extraripple filter*.

From the alternation theorem, it follows that

$$W(e^{j\omega_k})[A_d(e^{j\omega_k}) - A(e^{j\omega_k})] = (-1)^k \epsilon \qquad k = 0, 1, \ldots, L+1$$

where
$$\epsilon = \pm \max_{\omega \in \mathcal{F}} |E(e^{j\omega})|$$

is the maximum absolute weighted error. These equations may be written in matrix form in terms of the unknowns $a(0), \ldots, a(L)$ and ϵ as follows:

$$\begin{bmatrix} 1 & \cos(\omega_0) & \cdots & \cos(L\omega_0) & 1/W(e^{j\omega_0}) \\ 1 & \cos(\omega_1) & \cdots & \cos(L\omega_1) & -1/W(e^{j\omega_1}) \\ \vdots & \vdots & \vdots & \vdots & \\ 1 & \cos(\omega_L) & \cdots & \cos(L\omega_L) & (-1)^L/W(e^{j\omega_L}) \\ 1 & \cos(\omega_{L+1}) & \cdots & \cos(L\omega_{L+1}) & (-1)^{L+1}/W(e^{j\omega_{L+1}}) \end{bmatrix} \begin{bmatrix} a(0) \\ a(1) \\ \vdots \\ a(L) \\ \epsilon \end{bmatrix} = \begin{bmatrix} A_d(e^{j\omega_0}) \\ A_d(e^{j\omega_1}) \\ \vdots \\ A_d(e^{j\omega_L}) \\ A_d(e^{j\omega_{L+1}}) \end{bmatrix}$$

Given the extremal frequencies, these equations may be solved for $a(0), \ldots, a(L)$ and ϵ. To find the extremal frequencies, there is an efficient iterative procedure known as the Parks-McClellan algorithm, which involves the following steps:

1. Guess an initial set of extremal frequencies.
2. Find ϵ by solving Eq. (9.5). The value of ϵ has been shown to be

$$\epsilon = \frac{\sum_{k=0}^{L+1} b(k) D(e^{j\omega_k})}{\sum_{k=0}^{L+1} (-1)^k b(k)/W(e^{j\omega_k})}$$

where $$b(k) = \prod_{i=1, i\neq k}^{L+1} \frac{1}{\cos(\omega_k) - \cos(\omega_i)}$$

3. Evaluate the weighted error function over the set $\mathcal{F}$ by interpolating between the extremal frequencies using the Lagrange interpolation formula.
4. Select a new set of extremal frequencies by choosing the $L + 2$ frequencies for which the interpolated error function is maximum.
5. If the extremal frequencies have changed, repeat the iteration from step 2.

A design formula that may be used to estimate the equiripple filter order for a low-pass filter with a transition width Δf, passband ripple δ_p, and stopband ripple δ_s is

$$N = \frac{-10\log(\delta_p\delta_s) - 13}{14.6\Delta f}$$

Example 4.3.3

Suppose that we would like to design an equiripple low-pass filter with a passband cutoff frequency

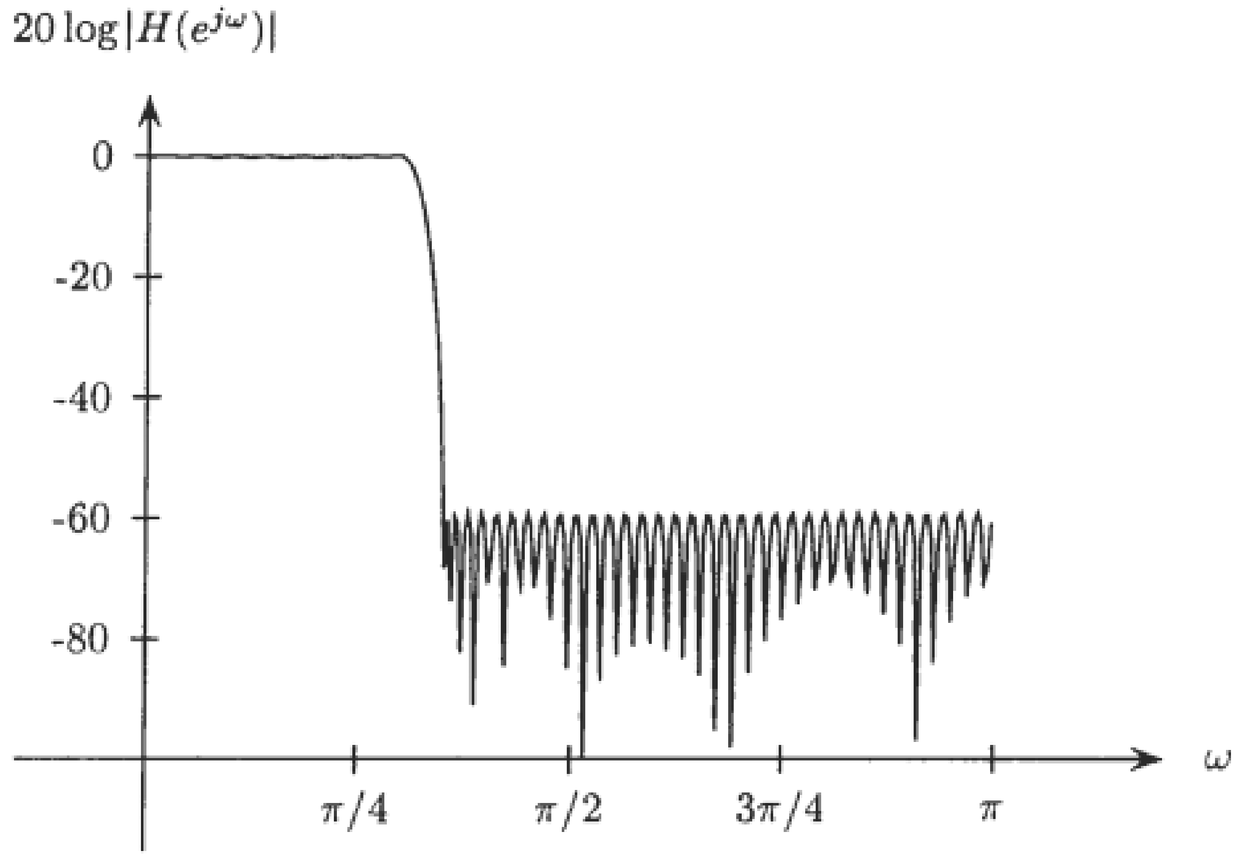
$\omega_p = 0.3\pi$, a stopband cutoff frequency $\omega_s = 0.35\pi$, a passband ripple of $\delta_p = 0.01$, and a stopband ripple of $\delta_s = 0.001$. Estimating the filter using Eq. (9.6), we find

$$N = \frac{-10\log(\delta_p\delta_s) - 13}{14.6\Delta f} = 102$$

Because we want the ripple in the stopband to be 10 times smaller than the ripple in the passband, the error must be weighted using the weighting function

$$W(e^{j\omega}) = \begin{cases} 1 & 0 \le |\omega| \le 0.3\pi \\ 10 & 0.35\pi \le |\omega| \le \pi \end{cases}$$

Using the Parks-McClellan algorithm to design the filter, we obtain a filter with the frequency response magnitude shown below.


$20\log|H(e^{j\omega})|$
0
-20
-40
-60
-80
$\pi/4$
$\pi/2$
$3\pi/4$
π
ω

4.4 IIR Filter Design

- There are two general approaches used to design IIR digital filters. The most common is to design an analog IIR filter and then map it into an equivalent digital filter because the art of analog filter design is highly advanced.
- Therefore, it is prudent to consider optimal ways for mapping these filters into the discrete-time domain. Furthermore, because there are powerful design procedures that facilitate the design of analog filters, this approach to IIR filter design is relatively simple.
- The second approach to design IIR digital filters is to use an algorithmic design procedure, which generally requires the use of a computer to solve a set of linear or nonlinear equations.

- These methods may be used to design digital filters with arbitrary frequency response characteristics for which no analog filter prototype exists or to design filters when other types of constraints are imposed on the design.
- In this section, we consider the approach of mapping analog filters into digital filters. Initially, the focus will be on the design of digital low-pass filters from analog low-pass filters.
- Techniques for transforming these designs into more general frequency selective filters will then be discussed.

4.4.1 Analog Low-Pass Filter Prototypes

- To design an IIR digital low-pass filter from an analog low-pass filter, we must first know how to design an analog low-pass filter.
- Most analog filter approximation methods were developed for the design of passive systems having a gain less than or equal to 1.

Therefore, a typical set of specifications for these filters is as shown in Fig. 4-5(a), with the pass band specifications having the form

$$1 - \delta_p \le |H_a(j\Omega)| \le 1$$

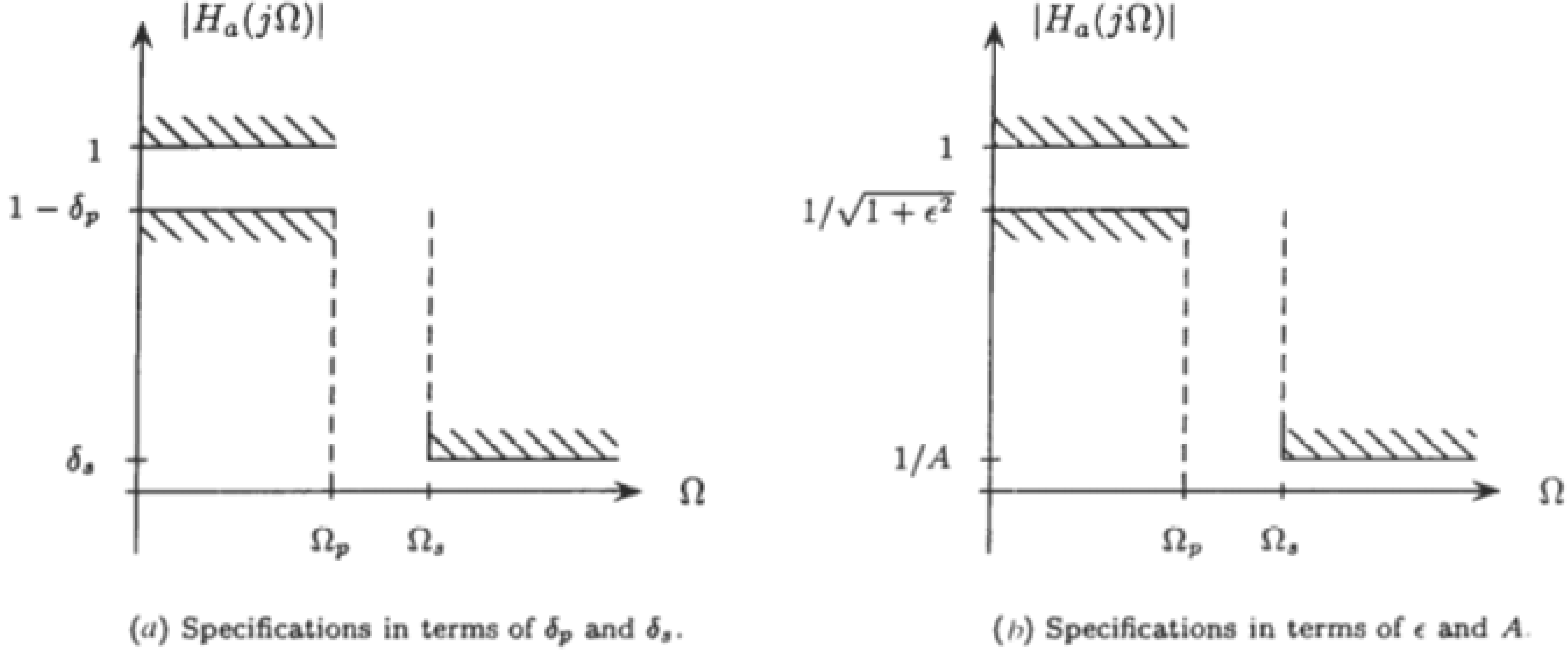


Fig. 4-5. Two different conventions for specifying the pass band and stop band deviations for an analog low-pass filter.