DIGITAL SIGNAL PROCESSING (23APC0415)

UNIT-I Introduction to discrete time signals and systems: Introduction to digital signal processing, Review of discrete-time signals and systems, Analysis of discrete-time linear time invariant systems, frequency domain representation of discrete time signals and systems

Z–Transform: Definition, ROC, Properties, Poles and Zeros in Z-plane, the inverse Z- Transform, System analysis, Transfer function, BIBO stability, System Response to standard signals, Solution of difference equations with initial conditions, Illustrative Problems, analysis of linear time-invariant systems in the z-domain, pole-zero stability.

UNIT - II Discrete Fourier Transform: Introduction, Discrete Fourier Series, properties of DFS, Discrete Fourier Transform, Inverse DFT, properties of DFT, Linear and Circular convolution, convolution using DFT, sampling, Quantization effects.

Fast Fourier Transform: Introduction, Fast Fourier Transform, Radix-2 Decimation in time and Decimation in frequency FFT, Inverse FFT (Radix-2).

UNIT - III IIR Filters: Introduction to digital filters, Analog filter approximations – Butterworth and Chebyshev, Design of IIR Digital filters from analog filters by Impulse invariant and bilinear transformation methods, Frequency transformations,

Basic structures of IIR Filters - Direct form- I, Direct form-II, Cascade form and Parallel form realizations.

UNIT - IV FIR Filters: Introduction, Characteristics of FIR filters with linear phase, Frequency response of linear phase FIR filters, Design of FIR filters using Fourier series and windowing methods (Rectangular, Triangular, Raised Cosine, Hanging, Hamming, Blackman), Comparison of IIR & FIR filters, Basic structures of FIR Filters – Direct form, Cascade form, Linear phase realizations.

UNIT - V Architectures for Programmable DSP Devices: Architecture of TMS320C5X: Introduction, Bus Structure, Central Arithmetic Logic Unit, Auxiliary Register ALU, Index Register, Block Move Address Register, Parallel Logic Unit, Memory mapped registers, program controller, some flags in the status registers, On- chip memory, On-chip peripherals.

UNIT -1
INTRODUCTION
REVIEW OF DISCRETE TIME SIGNALS AND SYSTEMS

Anything that carries some information can be called as signals. Some examples are ECG, EEG, ac power, seismic, speech, interest rates of a bank, unemployment rate of a country, temperature, pressure etc.
A **signal** is also defined as any physical quantity that varies with one or more independent variables.
A **discrete time signal** is the one which is not defined at intervals between two successive samples of a signal. It is represented as graphical, functional, tabular representation and sequence.
Some of the elementary discrete time signals are unit step, unit impulse, unit ramp, exponential and sinusoidal signals (as you read in signals and systems).

Classification of discrete time signals
Energy and Power signals

$$E \equiv \sum_{n=-\infty}^{\infty} |x(n)|^2$$

If the value of E is finite, then the signal x(n) is called energy signal.

$$P = \lim_{N \to \infty} \frac{1}{2N+1} \sum_{n=-N}^{N} |x(n)|^2$$

If the value of the P is finite, then the signal x(n) is called Power signal.

Periodic and Non periodic signals
A discrete time signal is said to be periodic if and only if it satisfies the condition X (N+n) =x (n), otherwise non periodic

Symmetric (even) and Anti-symmetric (odd) signals
The signal is said to be even if x(-n)=x(n)
The signal is said to be odd if x(-n)= - x(n)

Causal and non causal signal
The signal is said to be causal if its value is zero for negative values of 'n'.

Some of the **operations** on discrete time signals are shifting, time reversal, time scaling, signal multiplier, scalar multiplication and signal addition or multiplication.

Discrete time systems
A discrete time signal is a device or algorithm that operates on discrete time signals and produces another discrete time output.

Classification of discrete time systems
Static and dynamic systems
A system is said to be static if its output at present time depend on the input at present time only.

Causal and non causal systems
A system is said to be causal if the response of the system depends on present and past values of the input but not on the future inputs.

Linear and non linear systems
A system is said to be linear if the response of the system to the weighted sum of inputs should be equal to the corresponding weighted sum of outputs of the systems. This principle is called superposition principle.

Time invariant and time variant systems
A system is said to be time invariant if the characteristics of the systems do not change with time.

Stable and unstable systems
A system is said to be stable if bounded input produces bounded output only.

TIME DOMAIN ANALYSIS OF DISCRETE TIME SIGNALS AND SYSTEMS

Representation of an arbitrary sequence
Any signal x(n) can be represented as weighted sum of impulses as given below

$$x(n) = \sum_{k=-\infty}^{\infty} x(k)\delta(n-k)$$

The response of the system for unit sample input is called impulse response of the system h(n)

$$y(n) = T[x(n)] = T\left[\sum_{k=-\infty}^{\infty} x(k)\delta(n-k)\right]$$

$$= \sum_{k=-\infty}^{\infty} x(k)T[\delta(n-k)]$$

$$= \sum_{k=-\infty}^{\infty} x(k)h(n,k)$$

By time invariant property, we have

$$y(n) = \sum_{k=-\infty}^{\infty} x(k)h(n-k)$$

The above equation is called **convolution sum**.
Some of the properties of convolution are commutative law, associative law and distributive law.

Correlation of two sequences
It is basically used to compare two signals. It is the measure of similarity between two signals. Some of the applications are communication systems, radar, sonar etc.
The cross correlation of two sequences x(n) and y(n) is given by

$$r_{xy}(l) = \sum_{n=-\infty}^{\infty} x(n)y(n-l) \qquad l = 0, \pm 1, \pm 2, \ldots$$

One of the important properties of cross correlation is given by

$$r_{xy}(l) = r_{yx}(-l)$$

The auto correlation of the signal x(n) is given by

$$r_{xx}(l) = \sum_{n=-\infty}^{\infty} x(n)x(n-l)$$

Linear time invariant systems characterized by constant coefficient difference equation
The response of the first order difference equation is given by

$$y(n) = a^{n+1}y(-1) + \sum_{k=0}^{n} a^k x(n-k) \qquad n \geq 0$$

The first part contain initial condition y(-1) of the system, the second part contains input x(n) of the system.
The response of the system when it is in relaxed state at n=0 or
y(-1)=0 is called **zero state response** of the system or **forced response**.

$$y_{zs}(n) = \sum_{k=0}^{n} a^k x(n-k) \qquad n \geq 0$$

The output of the system at zero input condition x(n)=0 is called **zero input response** of the system or **natural response**.

The impulse response of the system is given by zero state response of the system

$$y_{zs}(n) = \sum_{k=0}^{n} a^k \delta(n-k)$$
$$= a^n \qquad n \geq 0$$

The total response of the system is equal to sum of natural response and forced responses.

FREQUENCY DOMAIN ANALYSIS OF DISCRETE TIME SIGNALS AND SYSTEMS

A s we have observed from the discussion o f Section 4.1, the Fourier series representation o f a continuous-time periodic signal can consist of an infinite number of frequency components, where the frequency spacing between two successive harmonically related frequencies is 1 / *T p,* and where *Tp* is the fundamental period.

Since the frequency range for continuous-time signals extends infinity on both sides it is possible to have signals that contain an infinite number of frequency components.

In contrast, the frequency range for discrete-time signals is unique over the interval. A discrete-time signal of fundamental period *N* can consist of frequency components separated by 2*n* / *N* radians.

Consequently, the Fourier series representation o f the discrete-time periodic signal will contain at most *N* frequency components. This is the basic difference between the Fourier series representations for continuous-time and discrete-time periodic signals.

4.2.1 The Fourier Series for Discrete-Time Periodic Signals

Suppose that we are given a periodic sequence $x(n)$ with period N, that is, $x(n) = x(n+N)$ for all n. The Fourier series representation for $x(n)$ consists of N harmonically related exponential functions

$$e^{j2\pi kn/N} \qquad k = 0, 1, \ldots, N-1$$

and is expressed as

$$x(n) = \sum_{k=0}^{N-1} c_k e^{j2\pi kn/N} \tag{4.2.1}$$

where the $\{c_k\}$ are the coefficients in the series representation.

To derive the expression for the Fourier coefficients, we use the following formula:

$$\sum_{n=0}^{N-1} e^{j2\pi kn/N} = \begin{cases} N, & k = 0, \pm N, \pm 2N, \ldots \\ 0, & \text{otherwise} \end{cases} \tag{4.2.2}$$

Note the similarity of (4.2.2) with the continuous-time counterpart in (4.1.3). The proof of (4.2.2) follows immediately from the application of the geometric summation formula

$$\sum_{n=0}^{N-1} a^n = \begin{cases} N, & a = 1 \\ \dfrac{1-a^N}{1-a}, & a \neq 1 \end{cases} \tag{4.2.3}$$

The expression for the Fourier coefficients c_k can be obtained by multiplying both sides of (4.2.1) by the exponential $e^{-j2\pi ln/N}$ and summing the product from $n = 0$ to $n = N-1$. Thus

$$\sum_{n=0}^{N-1} x(n) e^{-j2\pi ln/N} = \sum_{n=0}^{N-1} \sum_{k=0}^{N-1} c_k e^{j2\pi (k-l)n/N} \tag{4.2.4}$$

If we perform the summation over n first, in the right-hand side of (4.2.4), we obtain

$$\sum_{n=0}^{N-1} e^{j2\pi (k-l)n/N} = \begin{cases} N, & k-l = 0, \pm N, \pm 2N, \ldots \\ 0, & \text{otherwise} \end{cases} \tag{4.2.5}$$

where we have made use of (4.2.2). Therefore, the right-hand side of (4.2.4) reduces to Nc_l and hence

$$c_l = \frac{1}{N} \sum_{n=0}^{N-1} x(n) e^{-j2\pi ln/N} \qquad l = 0, 1, \ldots, N-1 \tag{4.2.6}$$

Thus we have the desired expression for the Fourier coefficients in terms of the signal $x(n)$.

4.2.3 The Fourier Transform of Discrete-Time Aperiodic Signals

Just as in the case of continuous-time aperiodic energy signals, the frequency analysis of discrete-time aperiodic finite-energy signals involves a Fourier transform of the time-domain signal. Consequently, the development in this section parallels to a large extent, that given in Section 4.1.3.

The Fourier transform of a finite-energy discrete-time signal $x(n)$ is defined as

$$X(\omega) = \sum_{n=-\infty}^{\infty} x(n)e^{-j\omega n} \tag{4.2.23}$$

Physically, $X(\omega)$ represents the frequency content of the signal $x(n)$. In other words, $X(\omega)$ is a decomposition of $x(n)$ into its frequency components.

We observe two basic differences between the Fourier transform of a discrete-time finite-energy signal and the Fourier transform of a finite-energy analog signal. First, for continuous-time signals, the Fourier transform, and hence the spectrum of the signal, have a frequency range of $(-\infty, \infty)$. In contrast, the frequency range for a discrete-time signal is unique over the frequency interval of $(-\pi, \pi)$ or, equivalently, $(0, 2\pi)$. This property is reflected in the Fourier transform of the signal. Indeed, $X(\omega)$ is periodic with period 2π, that is,

$$\begin{aligned} X(\omega + 2\pi k) &= \sum_{n=-\infty}^{\infty} x(n)e^{-j(\omega+2\pi k)n} \\ &= \sum_{n=-\infty}^{\infty} x(n)e^{-j\omega n}e^{-j2\pi kn} \\ &= \sum_{n=-\infty}^{\infty} x(n)e^{-j\omega n} = X(\omega) \end{aligned} \tag{4.2.24}$$

Hence $X(\omega)$ is periodic with period 2π. But this property is just a consequence of the fact that the frequency range for any discrete-time signal is limited to $(-\pi, \pi)$ or $(0, 2\pi)$, and any frequency outside this interval is equivalent to a frequency within the interval.

The second basic difference is also a consequence of the discrete-time nature of the signal. Since the signal is discrete in time, the Fourier transform of the signal involves a summation of terms instead of an integral, as in the case of continuous-time signals.

Since $X(\omega)$ is a periodic function of the frequency variable ω, it has a Fourier series expansion, provided that the conditions for the existence of the Fourier series, described previously, are satisfied. In fact, from the definition of the Fourier transform $X(\omega)$ of the sequence $x(n)$, given by (4.2.23), we observe that $X(\omega)$ has the form of a Fourier series. The Fourier coefficients in this series expansion are the values of the sequence $x(n)$.

To demonstrate this point, let us evaluate the sequence $x(n)$ from $X(\omega)$. First, we multiply both sides (4.2.23) by $e^{j\omega m}$ and integrate over the interval $(-\pi, \pi)$. Thus we have

$$\int_{-\pi}^{\pi} X(\omega)e^{j\omega m}d\omega = \int_{-\pi}^{\pi}\left[\sum_{n=-\infty}^{\infty} x(n)e^{-j\omega n}\right]e^{j\omega m}d\omega \tag{4.2.25}$$

The integral on the right-hand side of (4.2.25) can be evaluated if we can interchange the order of summation and integration. This interchange can be made if the series

$$X_N(\omega) = \sum_{n=-N}^{N} x(n)e^{-j\omega n}$$

converges uniformly to $X(\omega)$ as $N \to \infty$. Uniform convergence means that, for every ω, $X_N(\omega) \to X(\omega)$, as $N \to \infty$. The convergence of the Fourier transform is discussed in more detail in the following section. For the moment, let us assume that the series converges uniformly, so that we can interchange the order of summation and integration in (4.2.25). Then

$$\int_{-\pi}^{\pi} e^{j\omega(m-n)}d\omega = \begin{cases} 2\pi, & m = n \\ 0, & m \neq n \end{cases}$$

Consequently,

$$\sum_{n=-\infty}^{\infty} x(n)\int_{-\pi}^{\pi} e^{j\omega(m-n)}d\omega = \begin{cases} 2\pi x(m), & m = n \\ 0, & m \neq n \end{cases} \tag{4.2.26}$$

By combining (4.2.25) and (4.2.26), we obtain the desired result that

$$x(n) = \frac{1}{2\pi}\int_{-\pi}^{\pi} X(\omega)e^{j\omega n}d\omega \tag{4.2.27}$$

If we compare the integral in (4.2.27) with (4.1.9), we note that this is just the expression for the Fourier series coefficient for a function that is periodic with period 2π. The only difference between (4.1.9) and (4.2.27) is the sign on the exponent in the integrand, which is a consequence of our definition of the Fourier transform as given by (4.2.23). Therefore, the Fourier transform of the sequence $x(n)$, defined by (4.2.23), has the form of a Fourier series expansion.

The z-transform

See Oppenheim and Schafer, Second Edition pages 94–139, or First Edition pages 149–201.

1 Introduction

The z-transform of a sequence $x[n]$ is

$$X(z) = \sum_{n=-\infty}^{\infty} x[n]z^{-n}.$$

The z-transform can also be thought of as an operator $\mathcal{Z}\{\cdot\}$ that transforms a sequence to a function:

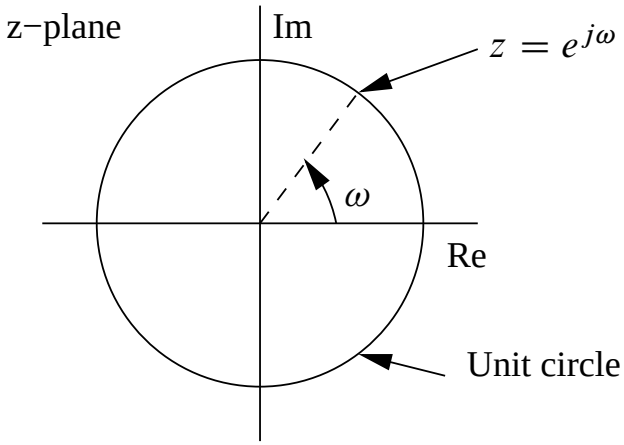
$$\mathcal{Z}\{x[n]\} = \sum_{n=-\infty}^{\infty} x[n]z^{-n} = X(z).$$

In both cases z is a continuous complex variable.

We may obtain the Fourier transform from the z-transform by making the substitution $z = e^{j\omega}$. This corresponds to restricting $|z| = 1$. Also, with $z = re^{j\omega}$,

$$X(re^{j\omega}) = \sum_{n=-\infty}^{\infty} x[n](re^{j\omega})^{-n} = \sum_{n=-\infty}^{\infty} (x[n]r^{-n})\, e^{-j\omega n}.$$

That is, the z-transform is the Fourier transform of the sequence $x[n]r^{-n}$. For $r = 1$ this becomes the Fourier transform of $x[n]$. The Fourier transform therefore corresponds to the z-transform evaluated on the unit circle:



The inherent periodicity in frequency of the Fourier transform is captured naturally under this interpretation.

The Fourier transform does not converge for all sequences — the infinite sum may not always be finite. Similarly, the z-transform does not converge for all sequences or for all values of z. The set of values of z for which the z-transform converges is called the **region of convergence (ROC)**.

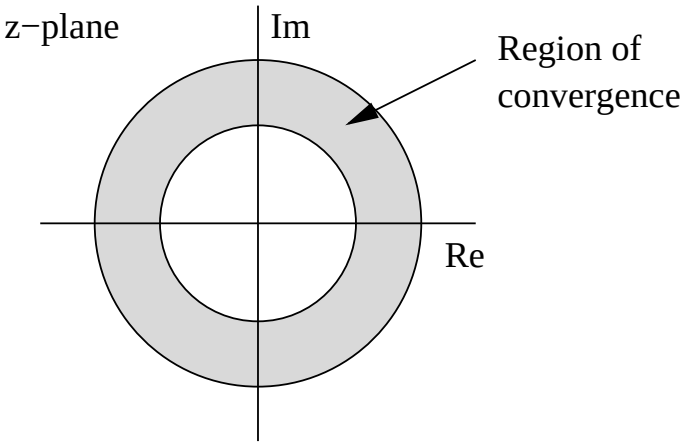
The Fourier transform of $x[n]$ exists if the sum $\sum_{n=-\infty}^{\infty} |x[n]|$ converges. However, the z-transform of $x[n]$ is just the Fourier transform of the sequence $x[n]r^{-n}$. The z-transform therefore exists (or converges) if

$$X(z) = \sum_{n=-\infty}^{\infty} |x[n]r^{-n}| < \infty.$$

This leads to the condition

$$\sum_{n=-\infty}^{\infty} |x[n]||z|^{-n} < \infty$$

for the existence of the z-transform. The ROC therefore consists of a ring in the z-plane:



In specific cases the inner radius of this ring may include the origin, and the outer radius may extend to infinity. If the ROC includes the unit circle $|z| = 1$, then the Fourier transform will converge.

Most useful z-transforms can be expressed in the form

$$X(z) = \frac{P(z)}{Q(z)},$$

where $P(z)$ and $Q(z)$ are polynomials in z. The values of z for which $P(z) = 0$ are called the **zeros** of $X(z)$, and the values with $Q(z) = 0$ are called the **poles**. The zeros and poles completely specify $X(z)$ to within a multiplicative constant.

Example: right-sided exponential sequence
Consider the signal $x[n] = a^n u[n]$. This has the z-transform

$$X(z) = \sum_{n=-\infty}^{\infty} a^n u[n] z^{-n} = \sum_{n=0}^{\infty} (az^{-1})^n.$$

Convergence requires that

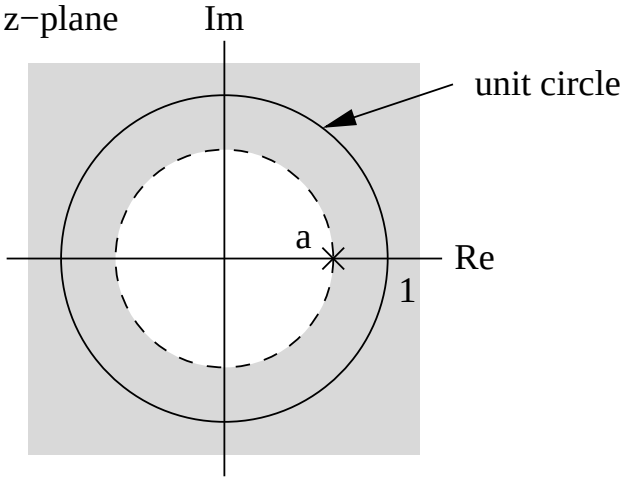
$$\sum_{n=0}^{\infty} |az^{-1}|^n < \infty,$$

which is only the case if $|az^{-1}| < 1$, or equivalently $|z| > |a|$. In the ROC, the

series converges to

$$X(z) = \sum_{n=0}^{\infty} (az^{-1})^n = \frac{1}{1 - az^{-1}} = \frac{z}{z - a}, \qquad |z| > |a|,$$

since it is just a geometric series. The z-transform has a region of convergence for any finite value of a.



The Fourier transform of $x[n]$ only exists if the ROC includes the unit circle, which requires that $|a| < 1$. On the other hand, if $|a| > 1$ then the ROC does not include the unit circle, and the Fourier transform does not exist. This is consistent with the fact that for these values of a the sequence $a^n u[n]$ is exponentially growing, and the sum therefore does not converge.

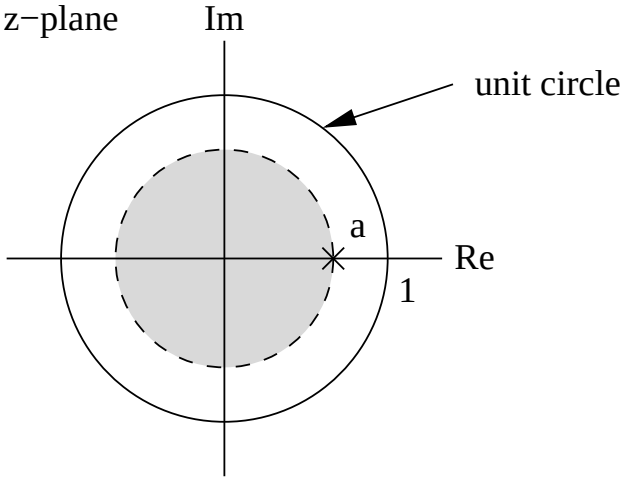
Example: left-sided exponential sequence

Now consider the sequence $x[n] = -a^n u[-n-1]$. This sequence is left-sided because it is nonzero only for $n \leq -1$. The z-transform is

$$\begin{aligned} X(z) &= \sum_{n=-\infty}^{\infty} -a^n u[-n-1] z^{-n} = -\sum_{n=-\infty}^{-1} a^n z^{-n} \\ &= -\sum_{n=1}^{\infty} a^{-n} z^n = 1 - \sum_{n=0}^{\infty} (a^{-1} z)^n. \end{aligned}$$

For $|a^{-1}z| < 1$, or $|z| < |a|$, the series converges to

$$X(z) = 1 - \frac{1}{1 - a^{-1}z} = \frac{1}{1 - az^{-1}} = \frac{z}{z - a}, \qquad |z| < |a|.$$



Note that the expression for the z-transform (and the pole zero plot) is exactly the same as for the right-handed exponential sequence — *only the region of convergence is different*. Specifying the ROC is therefore critical when dealing with the z-transform.

Example: sum of two exponentials

The signal $x[n] = \left(\frac{1}{2}\right)^n u[n] + \left(-\frac{1}{3}\right)^n u[n]$ is the sum of two real exponentials. The z-transform is

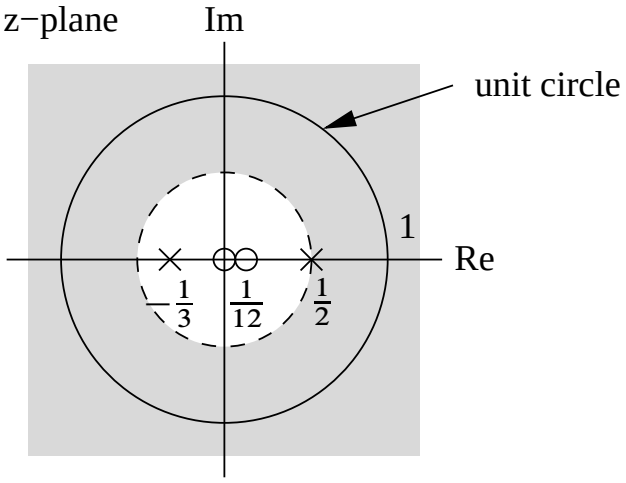
$$\begin{aligned} X(z) &= \sum_{n=-\infty}^{\infty} \left\{ \left(\frac{1}{2}\right)^n u[n] + \left(-\frac{1}{3}\right)^n u[n] \right\} z^{-n} \\ &= \sum_{n=-\infty}^{\infty} \left(\frac{1}{2}\right)^n u[n] z^{-n} + \sum_{n=-\infty}^{\infty} \left(-\frac{1}{3}\right)^n u[n] z^{-n} \\ &= \sum_{n=0}^{\infty} \left(\frac{1}{2} z^{-1}\right)^n + \sum_{n=0}^{\infty} \left(-\frac{1}{3} z^{-1}\right)^n . \end{aligned}$$

From the example for the right-handed exponential sequence, the first term in this sum converges for $|z| > 1/2$, and the second for $|z| > 1/3$. The combined transform $X(z)$ therefore converges in the intersection of these regions, namely

when $|z| > 1/2$. In this case

$$X(z) = \frac{1}{1 - \frac{1}{2}z^{-1}} + \frac{1}{1 + \frac{1}{3}z^{-1}} = \frac{2z(z - \frac{1}{12})}{(z - \frac{1}{2})(z + \frac{1}{3})}.$$

The pole-zero plot and region of convergence of the signal is



Example: finite length sequence

The signal

$$x[n] = \begin{cases} a^n & 0 \le n \le N-1 \\ 0 & \text{otherwise} \end{cases}$$

has z-transform

$$\begin{aligned} X(z) &= \sum_{n=0}^{N-1} a^n z^{-n} = \sum_{n=0}^{N-1} (az^{-1})^n \\ &= \frac{1 - (az^{-1})^N}{1 - az^{-1}} = \frac{1}{z^{N-1}} \frac{z^N - a^N}{z - a}. \end{aligned}$$

Since there are only a finite number of nonzero terms the sum always converges when az^{-1} is finite. There are no restrictions on a ($|a| < \infty$), and the ROC is the entire z-plane with the exception of the origin $z = 0$ (where the terms in the sum are infinite). The N roots of the numerator polynomial are at

$$z_k = ae^{j(2\pi k/N)}, \qquad k = 0, 1, \ldots, N-1,$$

since these values satisfy the equation $z^N = a^N$. The zero at $k = 0$ cancels the pole at $z = a$, so there are no poles except at the origin, and the zeros are at

$$z_k = ae^{j(2\pi k/N)}, \qquad k = 1, \ldots, N-1.$$

2 Properties of the region of convergence

The properties of the ROC depend on the nature of the signal. Assuming that the signal has a finite amplitude and that the z-transform is a rational function:

- The ROC is a ring or disk in the z-plane, centered on the origin ($0 \leq r_R < |z| < r_L \leq \infty$).
- The Fourier transform of $x[n]$ converges absolutely if and only if the ROC of the z-transform includes the unit circle.
- The ROC cannot contain any poles.
- If $x[n]$ is finite duration (ie. zero except on finite interval $-\infty < N_1 \leq n \leq N_2 < \infty$), then the ROC is the entire z-plane except perhaps at $z = 0$ or $z = \infty$.
- If $x[n]$ is a right-sided sequence then the ROC extends outward from the outermost finite pole to infinity.
- If $x[n]$ is left-sided then the ROC extends inward from the innermost nonzero pole to $z = 0$.
- A two-sided sequence (neither left nor right-sided) has a ROC consisting of a ring in the z-plane, bounded on the interior and exterior by a pole (and not containing any poles).
- The ROC is a connected region.

3 The inverse z-transform

Formally, the inverse z-transform can be performed by evaluating a Cauchy integral. However, for discrete LTI systems simpler methods are often sufficient.

3.1 Inspection method

If one is familiar with (or has a table of) common z-transform pairs, the inverse can be found by inspection. For example, one can invert the z-transform

$$X(z) = \left(\frac{1}{1 - \frac{1}{2} z^{-1}} \right), \qquad |z| > \frac{1}{2},$$

using the z-transform pair

$$a^n u[n] \overset{\mathcal{Z}}{\longleftrightarrow} \frac{1}{1 - a z^{-1}}, \qquad \text{for } |z| > |a|.$$

By inspection we recognise that

$$x[n] = \left(\frac{1}{2} \right)^n u[n].$$

Also, if $X(z)$ is a sum of terms then one may be able to do a term-by-term inversion by inspection, yielding $x[n]$ as a sum of terms.

3.2 Partial fraction expansion

For any rational function we can obtain a partial fraction expansion, and identify the z-transform of each term. Assume that $X(z)$ is expressed as a ratio of polynomials in z^{-1}:

$$X(z) = \frac{\sum_{k=0}^{M} b_k z^{-k}}{\sum_{k=0}^{N} a_k z^{-k}}.$$

It is always possible to factor $X(z)$ as

$$X(z) = \frac{b_0}{a_0} \frac{\prod_{k=1}^{M}(1 - c_k z^{-1})}{\prod_{k=1}^{N}(1 - d_k z^{-1})},$$

where the c_k's and d_k's are the nonzero zeros and poles of $X(z)$.

- If $M < N$ and the poles are all first order, then $X(z)$ can be expressed as

$$X(z) = \sum_{k=1}^{N} \frac{A_k}{1 - d_k z^{-1}}.$$

  In this case the coefficients A_k are given by

$$A_k = (1 - d_k z^{-1}) X(z)\big|_{z=d_k}.$$

- If $M \geq N$ and the poles are all first order, then an expansion of the form

$$X(z) = \sum_{r=0}^{M-N} B_r z^{-r} + \sum_{k=1}^{N} \frac{A_k}{1 - d_k z^{-1}}$$

  can be used, and the B_r's be obtained by long division of the numerator by the denominator. The A_k's can be obtained using the same equation as for $M < N$.

- The most general form for the partial fraction expansion, which can also deal with multiple-order poles, is

$$X(z) = \sum_{r=0}^{M-N} B_r z^{-r} + \sum_{k=1, k\neq i}^{N} \frac{A_k}{1 - d_k z^{-1}} + \sum_{m=1}^{s} \frac{C_m}{(1 - d_i z^{-1})^m}.$$

  Ways of finding the C_m's can be found in most standard DSP texts.

The terms $B_r z^{-r}$ correspond to shifted and scaled impulse sequences, and invert to terms of the form $B_r \delta[n - r]$. The fractional terms

$$\frac{A_k}{1 - d_k z^{-1}}$$

correspond to exponential sequences. For these terms the ROC properties must be used to decide whether the sequences are left-sided or right-sided.

Example: inverse by partial fractions

Consider the sequence $x[n]$ with z-transform

$$X(z) = \frac{1 + 2z^{-1} + z^{-2}}{1 - \frac{3}{2}z^{-1} + \frac{1}{2}z^{-2}} = \frac{(1 + z^{-1})^2}{(1 - \frac{1}{2}z^{-1})(1 - z^{-1})}, \qquad |z| > 1.$$

Since $M = N = 2$ this can be expressed as

$$X(z) = B_0 + \frac{A_1}{1 - \frac{1}{2}z^{-1}} + \frac{A_2}{1 - z^{-1}}.$$

The value B_0 can be found by long division:

$$\begin{array}{r} 2 \\ \frac{1}{2}z^{-2} - \frac{3}{2}z^{-1} + 1 \overline{)\, z^{-2} + 2z^{-1} + 1} \\ \underline{z^{-2} - 3z^{-1} + 2} \\ 5z^{-1} - 1 \end{array}$$

so

$$X(z) = 2 + \frac{-1 + 5z^{-1}}{(1 - \frac{1}{2}z^{-1})(1 - z^{-1})}.$$

The coefficients A_1 and A_2 can be found using

$$A_k = (1 - d_k z^{-1}) X(z)\big|_{z = d_k},$$

so

$$A_1 = \left.\frac{1 + 2z^{-1} + z^{-2}}{1 - z^{-1}}\right|_{z^{-1}=2} = \frac{1 + 4 + 4}{1 - 2} = -9$$

and

$$A_2 = \left.\frac{1 + 2z^{-1} + z^{-2}}{1 - \frac{1}{2}z^{-1}}\right|_{z^{-1}=1} = \frac{1 + 2 + 1}{1/2} = 8.$$

Therefore

$$X(z) = 2 - \frac{9}{1 - \frac{1}{2}z^{-1}} + \frac{8}{1 - z^{-1}}.$$

Using the fact that the ROC is $|z| > 1$, the terms can be inverted one at a time by inspection to give

$$x[n] = 2\delta[n] - 9(1/2)^n u[n] + 8u[n].$$

3.3 Power series expansion

If the z-transform is given as a power series in the form

$$\begin{aligned} X(z) &= \sum_{n=-\infty}^{\infty} x[n]z^{-n} \\ &= \ldots + x[-2]z^2 + x[-1]z^1 + x[0] + x[1]z^{-1} + x[2]z^{-2} + \ldots, \end{aligned}$$

then any value in the sequence can be found by identifying the coefficient of the appropriate power of z^{-1}.

Example: finite-length sequence
The z-transform

$$X(z) = z^2(1 - \frac{1}{2}z^{-1})(1 + z^{-1})(1 - z^{-1})$$

can be multiplied out to give

$$X(z) = z^2 - \frac{1}{2}z - 1 + \frac{1}{2}z^{-1}.$$

By inspection, the corresponding sequence is therefore

$$x[n] = \begin{cases} 1 & n = -2 \\ -\frac{1}{2} & n = -1 \\ -1 & n = 0 \\ \frac{1}{2} & n = 1 \\ 0 & \text{otherwise} \end{cases}$$

or equivalently

$$x[n] = 1\delta[n+2] - \frac{1}{2}\delta[n+1] - 1\delta[n] + \frac{1}{2}\delta[n-1].$$

Example: power series expansion
Consider the z-transform

$$X(z) = \log(1 + az^{-1}), \qquad |z| > |a|.$$

Using the power series expansion for $\log(1+x)$, with $|x| < 1$, gives

$$X(z) = \sum_{n=1}^{\infty} \frac{(-1)^{n+1} a^n z^{-n}}{n}.$$

The corresponding sequence is therefore

$$x[n] = \begin{cases} (-1)^{n+1} \frac{a^n}{n} & n \geq 1 \\ 0 & n \leq 0. \end{cases}$$

Example: power series expansion by long division
Consider the transform

$$X(z) = \frac{1}{1 - az^{-1}}, \qquad |z| > |a|.$$

Since the ROC is the exterior of a circle, the sequence is right-sided. We therefore divide to get a power series in powers of z^{-1}:

$$\begin{array}{r} 1+az^{-1}+a^2z^{-2}+\cdots \\ 1-az^{-1}\big)\overline{\;1\phantom{+az^{-1}+a^2z^{-2}+\cdots}} \\ \underline{1-az^{-1}\phantom{+a^2z^{-2}+\cdots}} \\ az^{-1}\phantom{+a^2z^{-2}+\cdots} \\ \underline{az^{-1}-a^2z^{-2}} \\ a^2z^{-2}+\cdots \end{array}$$

or

$$\frac{1}{1 - az^{-1}} = 1 + az^{-1} + a^2z^{-2} + \cdots.$$

Therefore $x[n] = a^n u[n]$.

Example: power series expansion for left-sided sequence
Consider instead the z-transform

$$X(z) = \frac{1}{1 - az^{-1}}, \qquad |z| < |a|.$$

Because of the ROC, the sequence is now a left-sided one. Thus we divide to obtain a series in powers of z:

$$\begin{array}{r} -a^{-1}z - a^{-2}z^2 - \cdots \\ -a + z \overline{)\; z} \\ \underline{z - a^{-1}z^2} \\ az^{-1} \end{array}$$

Thus $x[n] = -a^n u[-n-1]$.

4 Properties of the z-transform

In this section, if $X(z)$ denotes the z-transform of a sequence $x[n]$ and the ROC of $X(z)$ is indicated by R_x, then this relationship is indicated as

$$x[n] \overset{\mathcal{Z}}{\longleftrightarrow} X(z), \qquad \text{ROC} = R_x.$$

Furthermore, with regard to nomenclature, we have two sequences such that

$$x_1[n] \overset{\mathcal{Z}}{\longleftrightarrow} X_1(z), \qquad \text{ROC} = R_{x_1}$$

$$x_2[n] \overset{\mathcal{Z}}{\longleftrightarrow} X_2(z), \qquad \text{ROC} = R_{x_2}.$$

4.1 Linearity

The linearity property is as follows:

$$ax_1[n] + bx_2[n] \overset{\mathcal{Z}}{\longleftrightarrow} aX_1(z) + bX_2(z), \qquad \text{ROC contains} R_{x_1} \cap R_{x_1}.$$

4.2 Time shifting

The time-shifting property is as follows:

$$x[n-n_0] \overset{\mathcal{Z}}{\longleftrightarrow} z^{-n_0} X(z), \qquad \text{ROC} = R_x.$$

(The ROC may change by the possible addition or deletion of $z = 0$ or $z = \infty$.) This is easily shown:

$$\begin{aligned} Y(z) &= \sum_{n=-\infty}^{\infty} x[n-n_0] z^{-n} = \sum_{m=-\infty}^{\infty} x[m] z^{-(m+n_0)} \\ &= z^{-n_0} \sum_{m=-\infty}^{\infty} x[m] z^{-m} = z^{-n_0} X(z). \end{aligned}$$

Example: shifted exponential sequence
Consider the z-transform

$$X(z) = \frac{1}{z - \frac{1}{4}}, \qquad |z| > \frac{1}{4}.$$

From the ROC, this is a right-sided sequence. Rewriting,

$$X(z) = \frac{z^{-1}}{1 - \frac{1}{4} z^{-1}} = z^{-1} \left(\frac{1}{1 - \frac{1}{4} z^{-1}} \right), \qquad |z| > \frac{1}{4}.$$

The term in brackets corresponds to an exponential sequence $(1/4)^n u[n]$. The factor z^{-1} shifts this sequence one sample to the right. The inverse z-transform is therefore

$$x[n] = (1/4)^{n-1} u[n-1].$$

Note that this result could also have been easily obtained using a partial fraction expansion.

4.3 Multiplication by an exponential sequence

The exponential multiplication property is

$$z_0^n x[n] \overset{\mathcal{Z}}{\longleftrightarrow} X(z/z_0), \qquad \text{ROC} = |z_0| R_x,$$

where the notation $|z_0|R_x$ indicates that the ROC is scaled by $|z_0|$ (that is, inner and outer radii of the ROC scale by $|z_0|$). All pole-zero locations are similarly scaled by a factor z_0: if $X(z)$ had a pole at $z = z_1$, then $X(z/z_0)$ will have a pole at $z = z_0 z_1$.

- If z_0 is positive and real, this operation can be interpreted as a shrinking or expanding of the z-plane — poles and zeros change along radial lines in the z-plane.
- If z_0 is complex with unit magnitude ($z_0 = e^{j\omega_0}$) then the scaling operation corresponds to a rotation in the z-plane by and angle ω_0. That is, the poles and zeros rotate along circles centered on the origin. This can be interpreted as a shift in the frequency domain, associated with modulation in the time domain by $e^{j\omega_0 n}$. If the Fourier transform exists, this becomes

$$e^{j\omega_0 n} x[n] \overset{\mathcal{F}}{\longleftrightarrow} X(e^{j(\omega - \omega_0)}).$$

Example: exponential multiplication
The z-transform pair

$$u[n] \overset{\mathcal{Z}}{\longleftrightarrow} \frac{1}{1 - z^{-1}}, \qquad |z| > 1$$

can be used to determine the z-transform of $x[n] = r^n \cos(\omega_0 n) u[n]$. Since $\cos(\omega_0 n) = 1/2 e^{j\omega_0 n} + 1/2 e^{-j\omega_0 n}$, the signal can be rewritten as

$$x[n] = \frac{1}{2}(r e^{j\omega_0})^n u[n] + \frac{1}{2}(r e^{-j\omega_0})^n u[n].$$

From the exponential multiplication property,

$$\frac{1}{2}(re^{j\omega_0})^n u[n] \overset{\mathcal{Z}}{\longleftrightarrow} \frac{1/2}{1 - re^{j\omega_0}z^{-1}}, \qquad |z| > r$$
$$\frac{1}{2}(re^{-j\omega_0})^n u[n] \overset{\mathcal{Z}}{\longleftrightarrow} \frac{1/2}{1 - re^{-j\omega_0}z^{-1}}, \qquad |z| > r,$$

so

$$\begin{aligned} X(z) &= \frac{1/2}{1 - re^{j\omega_0}z^{-1}} + \frac{1/2}{1 - re^{-j\omega_0}z^{-1}}, \qquad |z| > r \\ &= \frac{1 - r\cos\omega_0 z^{-1}}{1 - 2r\cos\omega_0 z^{-1} + r^2 z^{-2}}, \qquad |z| > r. \end{aligned}$$

4.4 Differentiation

The differentiation property states that

$$nx[n] \overset{\mathcal{Z}}{\longleftrightarrow} -z\frac{dX(z)}{dz}, \qquad \text{ROC} = R_x.$$

This can be seen as follows: since

$$X(z) = \sum_{n=-\infty}^{\infty} x[n]z^{-n},$$

we have

$$-z\frac{dX(z)}{dz} = -z\sum_{n=-\infty}^{\infty}(-n)x[n]z^{-n-1} = \sum_{n=-\infty}^{\infty} nx[n]z^{-n} = \mathcal{Z}\{nx[n]\}.$$

Example: second order pole
The z-transform of the sequence

$$x[n] = na^n u[n]$$

can be found using

$$a^n u[n] \overset{\mathcal{Z}}{\longleftrightarrow} \frac{1}{1 - az^{-1}}, \qquad |z| > a,$$

to be

$$X(z) = -z\frac{d}{dz}\left(\frac{1}{1-az^{-1}}\right) = \frac{az^{-1}}{(1-az^{-1})^2}, \qquad |z| > a.$$

4.5 Conjugation

This property is

$$x^*[n] \overset{\mathcal{Z}}{\longleftrightarrow} X^*(z^*), \qquad \text{ROC} = R_x.$$

4.6 Time reversal

Here

$$x^*[-n] \overset{\mathcal{Z}}{\longleftrightarrow} X^*(1/z^*), \qquad \text{ROC} = \frac{1}{R_x}.$$

The notation $1/R_x$ means that the ROC is inverted, so if R_x is the set of values such that $r_R < |z| < r_L$, then the ROC is the set of values of z such that $1/r_l < |z| < 1/r_R$.

Example: time-reversed exponential sequence
The signal $x[n] = a^{-n}u[-n]$ is a time-reversed version of $a^n u[n]$. The z-transform is therefore

$$X(z) = \frac{1}{1-az} = \frac{-a^{-1}z^{-1}}{1-a^{-1}z^{-1}}, \qquad |z| < |a^{-1}|.$$

4.7 Convolution

This property states that

$$x_1[n] * x_2[n] \overset{\mathcal{Z}}{\longleftrightarrow} X_1(z)X_2(z), \qquad \text{ROC contains} R_{x_1} \cap R_{x_2}.$$

Example: evaluating a convolution using the z-transform

The z-transforms of the signals $x_1[n] = a^n u[n]$ and $x_2[n] = u[n]$ are

$$X_1(z) = \sum_{n=0}^{\infty} a^n z^{-n} = \frac{1}{1 - az^{-1}}, \qquad |z| > |a|$$

and

$$X_2(z) = \sum_{n=0}^{\infty} z^{-n} = \frac{1}{1 - z^{-1}}, \qquad |z| > 1.$$

For $|a| < 1$, the z-transform of the convolution $y[n] = x_1[n] * x_2[n]$ is

$$Y(z) = \frac{1}{(1 - az^{-1})(1 - z^{-1})} = \frac{z^2}{(z - a)(z - 1)}, \qquad |z| > 1.$$

Using a partial fraction expansion,

$$Y(z) = \frac{1}{1 - a}\left(\frac{1}{1 - z^{-1}} - \frac{a}{1 - az^{-1}}\right), \qquad |z| > 1,$$

so

$$y[n] = \frac{1}{1 - a}(u[n] - a^{n+1}u[n]).$$

4.8 Initial value theorem

If $x[n]$ is zero for $n < 0$, then

$$x[0] = \lim_{z \to \infty} X(z).$$

Some common z-transform pairs are:

Sequence	Transform	ROC
$\delta[n]$	1	All z
$u[n]$	$\frac{1}{1-z^{-1}}$	$\|z\| > 1$
$-u[-n-1]$	$\frac{1}{1-z^{-1}}$	$\|z\| < 1$
$\delta[n-m]$	z^{-m}	All z except 0 or ∞
$a^n u[n]$	$\frac{1}{1-az^{-1}}$	$\|z\| > \|a\|$
$-a^n u[-n-1]$	$\frac{1}{1-az^{-1}}$	$\|z\| < \|a\|$
$n a^n u[n]$	$\frac{az^{-1}}{(1-az^{-1})^2}$	$\|z\| > \|a\|$
$-n a^n u[-n-1]$	$\frac{az^{-1}}{(1-az^{-1})^2}$	$\|z\| < \|a\|$
$\begin{cases} a^n & 0 \le n \le N-1, \\ 0 & \text{otherwise} \end{cases}$	$\frac{1-a^N z^{-N}}{1-az^{-1}}$	$\|z\| > 0$
$\cos(\omega_0 n)u[n]$	$\frac{1-\cos(\omega_0)z^{-1}}{1-2\cos(\omega_0)z^{-1}+z^{-2}}$	$\|z\| > 1$
$r^n \cos(\omega_0 n)u[n]$	$\frac{1-r\cos(\omega_0)z^{-1}}{1-2r\cos(\omega_0)z^{-1}+r^2 z^{-2}}$	$\|z\| > r$

4.9 Relationship with the Laplace transform

Continuous-time systems and signals are usually described by the Laplace transform. Letting $z = e^{sT}$, where s is the complex Laplace variable

$$s = d + j\omega,$$

we have

$$z = e^{(d+j\omega)T} = e^{dT} e^{j\omega T}.$$

Therefore

$$|z| = e^{dT} \quad \text{and} \quad \sphericalangle z = \omega T = 2\pi f / f_s = 2\pi\omega/\omega_s,$$

where ω_s is the sampling frequency. As ω varies from ∞ to ∞, the s-plane is mapped to the z-plane:

- The $j\omega$ axis in the s-plane is mapped to the unit circle in the z-plane.
- The left-hand s-plane is mapped to the inside of the unit circle.
- The right-hand s-plane maps to the outside of the unit circle.

FREQUENCY DOMAIN SAMPLING: THE DISCRETE FOURIER TRANSFORM

Before we introduce the DFT, we consider the sampling of the Fourier transform of an aperiodic discrete-time sequence. Thus, we establish the relationship between the sampled Fourier transform and the DFT.

5.1.1 Frequency-Domain Sampling and Reconstruction of Discrete-Time Signals

We recall that aperiodic finite-energy signals have continuous spectra. Let us consider such an aperiodic discrete-time signal $x(n)$ with Fourier transform

$$X(\omega) = \sum_{n=-\infty}^{\infty} x(n)e^{-j\omega n} \tag{5.1.1}$$

Suppose that we sample $X(\omega)$ periodically in frequency at a spacing of $\delta\omega$ radians between successive samples. Since $X(\omega)$ is periodic with period 2π, only samples in the fundamental frequency range are necessary. For convenience, we take N equidistant samples in the interval $0 \le \omega < 2\pi$ with spacing $\delta\omega = 2\pi/N$, as shown in Fig. 5.1. First, we consider the selection of N, the number of samples in the frequency domain.

If we evaluate (5.1.1) at $\omega = 2\pi k/N$, we obtain

$$X\left(\frac{2\pi}{N}k\right) = \sum_{n=-\infty}^{\infty} x(n)e^{-j2\pi kn/N} \qquad k = 0, 1, \ldots, N-1 \tag{5.1.2}$$

The summation in (5.1.2) can be subdivided into an infinite number of summations, where each sum contains N terms. Thus

$$\begin{aligned} X\left(\frac{2\pi}{N}k\right) = & \cdots + \sum_{n=-N}^{-1} x(n)e^{-j2\pi kn/N} + \sum_{n=0}^{N-1} x(n)e^{-j2\pi kn/N} \\ & + \sum_{n=N}^{2N-1} x(n)e^{-j2\pi kn/N} + \cdots \\ = & \sum_{l=-\infty}^{\infty} \sum_{n=lN}^{lN+N-1} x(n)e^{-j2\pi kn/N} \end{aligned}$$

If we change the index in the inner summation from n to $n - lN$ and interchange the order of the summation, we obtain the result

$$X\left(\frac{2\pi}{N}k\right) = \sum_{n=0}^{N-1}\left[\sum_{l=-\infty}^{\infty} x(n-lN)\right]e^{-j2\pi kn/N} \tag{5.1.3}$$

for $k = 0, 1, 2, \ldots, N-1$.

The signal

$$x_p(n) = \sum_{l=-\infty}^{\infty} x(n-lN) \tag{5.1.4}$$

obtained by the periodic repetition of $x(n)$ every N samples, is clearly periodic with fundamental period N. Consequently, it can be expanded in a Fourier

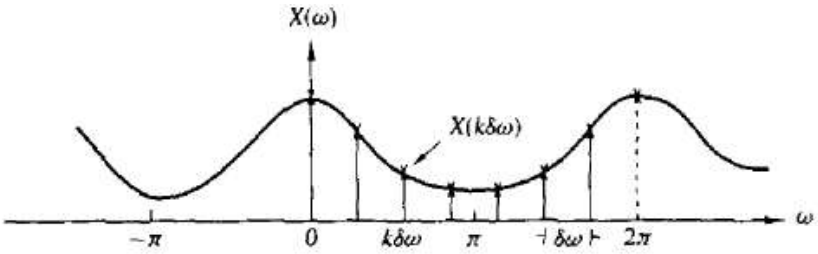


Figure 5.1 Frequency-domain sampling of the Fourier transform.

series as

$$x_p(n) = \sum_{k=0}^{N-1} c_k e^{j2\pi kn/N} \qquad n = 0, 1, \ldots, N-1 \tag{5.1.5}$$

with Fourier coefficients

$$c_k = \frac{1}{N}\sum_{n=0}^{N-1} x_p(n)e^{-j2\pi kn/N} \qquad k = 0, 1, \ldots, N-1 \tag{5.1.6}$$

Upon comparing (5.1.3) with (5.1.6), we conclude that

$$c_k = \frac{1}{N}X\left(\frac{2\pi}{N}k\right) \qquad k = 0, 1, \ldots, N-1 \tag{5.1.7}$$

Therefore,

$$x_p(n) = \frac{1}{N}\sum_{k=0}^{N-1} X\left(\frac{2\pi}{N}k\right)e^{j2\pi kn/N} \qquad n = 0, 1, \ldots, N-1 \tag{5.1.8}$$

5.1.2 The Discrete Fourier Transform (DFT)

The development in the preceding section is concerned with the frequency-domain sampling of an aperiodic finite-energy sequence $x(n)$. In general, the equally spaced frequency samples $X(2\pi k/N)$, $k = 0, 1, \ldots, N-1$, do not uniquely represent the original sequence $x(n)$ when $x(n)$ has infinite duration. Instead, the frequency samples $X(2\pi k/N)$, $k = 0, 1, \ldots, N-1$, correspond to a periodic sequence $x_p(n)$ of period N, where $x_p(n)$ is an aliased version of $x(n)$, as indicated by the relation in (5.1.4), that is,

$$x_p(n) = \sum_{l=-\infty}^{\infty} x(n - lN) \tag{5.1.15}$$

When the sequence $x(n)$ has a finite duration of length $L \leq N$, then $x_p(n)$ is simply a periodic repetition of $x(n)$, where $x_p(n)$ over a single period is given as

$$x_p(n) = \begin{cases} x(n), & 0 \leq n \leq L-1 \\ 0, & L \leq n \leq N-1 \end{cases} \tag{5.1.16}$$

Consequently, the frequency samples $X(2\pi k/N)$, $k = 0, 1, \ldots, N-1$, uniquely represent the finite-duration sequence $x(n)$. Since $x(n) \equiv x_p(n)$ over a single period (padded by $N - L$ zeros), the original finite-duration sequence $x(n)$ can be obtained from the frequency samples $\{X(2\pi k/N)\}$ by means of the formula (5.1.8).

It is important to note that *zero padding* does not provide any additional information about the spectrum $X(\omega)$ of the sequence $\{x(n)\}$. The L equidistant samples of $X(\omega)$ are sufficient to reconstruct $X(\omega)$ using the reconstruction formula (5.1.13). However, padding the sequence $\{x(n)\}$ with $N - L$ zeros and computing an N-point DFT results in a "better display" of the Fourier transform $X(\omega)$.

In summary, a finite-duration sequence $x(n)$ of length L [i.e., $x(n) = 0$ for $n < 0$ and $n \geq L$] has a Fourier transform

$$X(\omega) = \sum_{n=0}^{L-1} x(n)e^{-j\omega n} \qquad 0 \leq \omega \leq 2\pi \tag{5.1.17}$$

where the upper and lower indices in the summation reflect the fact that $x(n) = 0$ outside the range $0 \leq n \leq L-1$. When we sample $X(\omega)$ at equally spaced frequencies $\omega_k = 2\pi k/N$, $k = 0, 1, 2, \ldots, N-1$, where $N \geq L$, the resultant samples are

$$\begin{aligned} X(k) \equiv X\left(\frac{2\pi k}{N}\right) &= \sum_{n=0}^{L-1} x(n)e^{-j2\pi kn/N} \\ X(k) &= \sum_{n=0}^{N-1} x(n)e^{-j2\pi kn/N} \qquad k = 0, 1, 2, \ldots, N-1 \end{aligned} \tag{5.1.18}$$

where for convenience, the upper index in the sum has been increased from $L-1$ to $N-1$ since $x(n) = 0$ for $n \geq L$.

The relation in (5.1.18) is a formula for transforming a sequence $\{x(n)\}$ of length $L \leq N$ into a sequence of frequency samples $\{X(k)\}$ of length N. Since the frequency samples are obtained by evaluating the Fourier transform $X(\omega)$ at a set of N (equally spaced) discrete frequencies, the relation in (5.1.18) is called the *discrete Fourier transform* (DFT) of $x(n)$. In turn, the relation given by (5.1.10), which allows us to recover the sequence $x(n)$ from the frequency samples

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) e^{j2\pi kn/N} \qquad n = 0, 1, \ldots, N-1 \tag{5.1.19}$$

is called the *inverse DFT* (IDFT). Clearly, when $x(n)$ has length $L < N$, the N-point IDFT yields $x(n) = 0$ for $L \leq n \leq N-1$. To summarize, the formulas for the DFT and IDFT are

DFT

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-j2\pi kn/N} \qquad k = 0, 1, 2, \ldots, N-1 \tag{5.1.18}$$

IDFT

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) e^{j2\pi kn/N} \qquad n = 0, 1, 2, \ldots, N-1 \tag{5.1.19}$$

5.1.3 The DFT as a Linear Transformation

The formulas for the DFT and IDFT given by (5.1.18) and (5.1.19) may be expressed as

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{kn} \qquad k = 0, 1, \ldots, N-1 \tag{5.1.20}$$

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) W_N^{-kn} \qquad n = 0, 1, \ldots, N-1 \tag{5.1.21}$$

where, by definition,

$$W_N = e^{-j2\pi/N} \tag{5.1.22}$$

which is an Nth root of unity.

With these definitions, the N-point DFT may be expressed in matrix form as

$$\mathbf{X}_N = \mathbf{W}_N \mathbf{x}_N \tag{5.1.24}$$

where $\mathbf{W}_N$ is the matrix of the linear transformation. We observe that $\mathbf{W}_N$ is a symmetric matrix. If we assume that the inverse of $\mathbf{W}_N$ exists, then (5.1.24) can be inverted by premultiplying both sides by $\mathbf{W}_N^{-1}$. Thus we obtain

$$\mathbf{x}_N = \mathbf{W}_N^{-1} \mathbf{X}_N \tag{5.1.25}$$

Relationship to the Fourier series coefficients of a periodic sequence. A periodic sequence $\{x_p(n)\}$ with fundamental period N can be represented in a Fourier series of the form

$$x_p(n) = \sum_{k=0}^{N-1} c_k e^{j2\pi nk/N} \qquad -\infty < n < \infty \tag{5.1.29}$$

where the Fourier series coefficients are given by the expression

$$c_k = \frac{1}{N} \sum_{n=0}^{N-1} x_p(n) e^{-j2\pi nk/N} \qquad k = 0, 1, \ldots, N-1 \tag{5.1.30}$$

If we compare (5.1.29) and (5.1.30) with (5.1.18) and (5.1.19), we observe that the formula for the Fourier series coefficients has the form of a DFT. In fact, if we define a sequence $x(n) = x_p(n)$, $0 \le n \le N-1$, the DFT of this sequence is simply

$$X(k) = Nc_k \tag{5.1.31}$$

Furthermore, (5.1.29) has the form of an IDFT. Thus the N-point DFT provides the exact line spectrum of a periodic sequence with fundamental period N.

Relationship to the Fourier transform of an aperiodic sequence. We have already shown that if $x(n)$ is an aperiodic finite energy sequence with Fourier transform $X(\omega)$, which is sampled at N equally spaced frequencies $\omega_k = 2\pi k/N$, $k = 0, 1, \ldots, N-1$, the spectral components

$$X(k) = X(\omega)|_{\omega=2\pi k/N} = \sum_{n=-\infty}^{\infty} x(n) e^{-j2\pi nk/N} \qquad k = 0, 1, \ldots, N-1 \tag{5.1.32}$$

are the DFT coefficients of the periodic sequence of period N, given by

$$x_p(n) = \sum_{l=-\infty}^{\infty} x(n - lN) \tag{5.1.33}$$

Thus $x_p(n)$ is determined by aliasing $\{x(n)\}$ over the interval $0 \le n \le N-1$. The finite-duration sequence

$$\hat{x}(n) = \begin{cases} x_p(n), & 0 \le n \le N-1 \\ 0, & \text{otherwise} \end{cases} \tag{5.1.34}$$

bears no resemblance to the original sequence $\{x(n)\}$, unless $x(n)$ is of finite duration and length $L \leq N$, in which case

$$x(n) = \hat{x}(n) \qquad 0 \leq n \leq N-1 \tag{5.1.35}$$

Only in this case will the IDFT of $\{X(k)\}$ yield the original sequence $\{x(n)\}$.

Relationship to the *z*-transform. Let us consider a sequence $x(n)$ having the z-transform

$$X(z) = \sum_{n=-\infty}^{\infty} x(n)z^{-n} \tag{5.1.36}$$

with a ROC that includes the unit circle. If $X(z)$ is sampled at the N equally spaced points on the unit circle $z_k = e^{j2\pi k/N}$, $0, 1, 2, \ldots, N-1$, we obtain

$$\begin{aligned} X(k) &\equiv X(z)|_{z=e^{j2\pi k/N}} \qquad k = 0, 1, \ldots, N-1 \\ &= \sum_{n=-\infty}^{\infty} x(n)e^{-j2\pi nk/N} \end{aligned} \tag{5.1.37}$$

The expression in (5.1.37) is identical to the Fourier transform $X(\omega)$ evaluated at the N equally spaced frequencies $\omega_k = 2\pi k/N$, $k = 0, 1, \ldots, N-1$, which is the topic treated in Section 5.1.1.

If the sequence $x(n)$ has a finite duration of length N or less, the sequence can be recovered from its N-point DFT. Hence its z-transform is uniquely determined by its N-point DFT. Consequently, $X(z)$ can be expressed as a function of the DFT $\{X(k)\}$ as follows

$$\begin{aligned} X(z) &= \sum_{n=0}^{N-1} x(n)z^{-n} \\ X(z) &= \sum_{n=0}^{N-1} \left[\frac{1}{N} \sum_{k=0}^{N-1} X(k)e^{j2\pi kn/N} \right] z^{-n} \\ X(z) &= \frac{1}{N} \sum_{k=0}^{N-1} X(k) \sum_{n=0}^{N-1} \left(e^{j2\pi k/N} z^{-1}\right)^n \\ X(z) &= \frac{1-z^{-N}}{N} \sum_{k=0}^{N-1} \frac{X(k)}{1-e^{j2\pi k/N}z^{-1}} \end{aligned} \tag{5.1.38}$$

When evaluated on the unit circle, (5.1.38) yields the Fourier transform of the finite-duration sequence in terms of its DFT, in the form

$$X(\omega) = \frac{1-e^{-j\omega N}}{N} \sum_{k=0}^{N-1} \frac{X(k)}{1-e^{-j(\omega-2\pi k/N)}} \tag{5.1.39}$$

Relationship to the Fourier series coefficients of a continuous-time signal. Suppose that $x_a(t)$ is a continuous-time periodic signal with fundamental period $T_p = 1/F_0$. The signal can be expressed in a Fourier series

$$x_a(t) = \sum_{k=-\infty}^{\infty} c_k e^{j2\pi k F_0} \tag{5.1.40}$$

where $\{c_k\}$ are the Fourier coefficients. If we sample $x_a(t)$ at a uniform rate $F_s = N/T_p = 1/T$, we obtain the discrete-time sequence

$$\begin{aligned} x(n) \equiv x_a(nT) &= \sum_{k=-\infty}^{\infty} c_k e^{j2\pi k F_0 nT} = \sum_{k=-\infty}^{\infty} c_k e^{j2\pi kn/N} \\ &= \sum_{k=0}^{N-1} \left[\sum_{l=-\infty}^{\infty} c_{k-lN} \right] e^{j2\pi kn/N} \end{aligned} \tag{5.1.41}$$

It is clear that (5.1.41) is in the form of an IDFT formula, where

$$X(k) = N \sum_{l=-\infty}^{\infty} c_{k-lN} \equiv N\tilde{c}_k \tag{5.1.42}$$

and

$$\tilde{c}_k = \sum_{l=-\infty}^{\infty} c_{k-lN} \tag{5.1.43}$$

Thus the $\{\tilde{c}_k\}$ sequence is an aliased version of the sequence $\{c_k\}$.

PROPERTIES OF DFT:

Property	Time Domain	Frequency Domain
Notation	$x(n), y(n)$	$X(k), Y(k)$
Periodicity	$x(n) = x(n+N)$	$X(k) = X(k+N)$
Linearity	$a_1x_1(n) + a_2x_2(n)$	$a_1X_1(k) + a_2X_2(k)$
Time reversal	$x(N-n)$	$X(N-k)$
Circular time shift	$x((n-l))_N$	$X(k)e^{-j2\pi kl/N}$
Circular frequency shift	$x(n)e^{j2\pi ln/N}$	$X((k-l))_N$
Complex conjugate	$x^*(n)$	$X^*(N-k)$
Circular convolution	$x_1(n) \textcircled{N} x_2(n)$	$X_1(k)X_2(k)$
Circular correlation	$x(n) \textcircled{N} y^*(-n)$	$X(k)Y^*(k)$
Multiplication of two sequences	$x_1(n)x_2(n)$	$\frac{1}{N}X_1(k) \textcircled{N} X_2(k)$
Parseval's theorem	$\sum_{n=0}^{N-1} x(n)y^*(n)$	$\frac{1}{N}\sum_{k=0}^{N-1} X(k)Y^*(k)$

LINEAR FILTERING METHODS BASED ON THE DFT

Since the D F T provides a discrete frequency representation o f a finite-duration Sequence in the frequency domain, it is interesting to exp lore its use as a computational tool for linear system analysis and, especially, for linear filtering. We have already established that a system with frequency response $H\{w)y$ w hen excited with an input signal that has a spectrum possesses an output spectrum.

The output sequence *y(n)* is determined from its spectrum via the inverse Fourier transform. Computationally, the problem with this frequency domain approach is that are functions o f the

continuous variable. As a consequence, the computations cannot be done on a digital computer, since the computer can only store and perform computations on quantities at discrete frequencies.
On the other hand, the DFT does lend itself to computation on a digital computer. In the discussion that follows, we describe how the DFT can be used to perform linear filtering in the frequency domain. In particular, we present a computational procedure that serves as an alternative to time-domain convolution.

In fact, the frequency-domain approach based on the DFT, is computationally m ore efficient than time-domain convolution due to the existence of efficient algorithms for computing the DFT . These algorithms, which are described in Chapter 6, are collectively called fast Fourier transform (FFT) algorithms.

5.3.1 Use of the DFT in Linear Filtering

In the preceding section it was demonstrated that the product of two DFTs is equivalent to the circular convolution of the corresponding time-domain sequences. Unfortunately, circular convolution is of no use to us if our objective is to determine the output of a linear filter to a given input sequence. In this case we seek a frequency-domain methodology equivalent to linear convolution.

Suppose that we have a finite-duration sequence $x(n)$ of length L which excites an FIR filter of length M. Without loss of generality, let

$$x(n) = 0, \qquad n < 0 \text{ and } n \geq L$$

$$h(n) = 0, \qquad n < 0 \text{ and } n \geq M$$

where $h(n)$ is the impulse response of the FIR filter.

The output sequence $y(n)$ of the FIR filter can be expressed in the time domain as the convolution of $x(n)$ and $h(n)$, that is

$$y(n) = \sum_{k=0}^{M-1} h(k)x(n-k) \tag{5.3.1}$$

Since $h(n)$ and $x(n)$ are finite-duration sequences, their convolution is also finite in duration. In fact, the duration of $y(n)$ is $L + M - 1$.

The frequency-domain equivalent to (5.3.1) is

$$Y(\omega) = X(\omega)H(\omega) \tag{5.3.2}$$

If the sequence $y(n)$ is to be represented uniquely in the frequency domain by samples of its spectrum $Y(\omega)$ at a set of discrete frequencies, the number of distinct samples must equal or exceed $L+M-1$. Therefore, a DFT of size $N \geq L+M-1$, is required to represent $\{y(n)\}$ in the frequency domain.

Now if

$$\begin{aligned} Y(k) &\equiv Y(\omega)|_{\omega=2\pi k/N} \qquad k = 0, 1, \ldots, N-1 \\ &= X(\omega)H(\omega)|_{\omega=2\pi k/N} \qquad k = 0, 1, \ldots, N-1 \end{aligned}$$

then

$$Y(k) = X(k)H(k) \qquad k = 0, 1, \ldots, N-1 \tag{5.3.3}$$

where $\{X(k)\}$ and $\{H(k)\}$ are the N-point DFTs of the corresponding sequences $x(n)$ and $h(n)$, respectively. Since the sequences $x(n)$ and $h(n)$ have a duration less than N, we simply pad these sequences with zeros to increase their length to N. This increase in the size of the sequences does not alter their spectra $X(\omega)$ and $H(\omega)$, which are continuous spectra, since the sequences are aperiodic. However, by sampling their spectra at N equally spaced points in frequency (computing the N-point DFTs), we have increased the number of samples that represent these sequences in the frequency domain beyond the minimum number (L or M, respectively).

Since the $N = L+M-1$-point DFT of the output sequence $y(n)$ is sufficient to represent $y(n)$ in the frequency domain, it follows that the multiplication of the N-point DFTs $X(k)$ and $H(k)$, according to (5.3.3), followed by the computation of the N-point IDFT, must yield the sequence $\{y(n)\}$. In turn, this implies that the N-point circular convolution of $x(n)$ with $h(n)$ must be equivalent to the linear convolution of $x(n)$ with $h(n)$. In other words, by increasing the length of the sequences $x(n)$ and $h(n)$ to N points (by appending zeros), and then circularly convolving the resulting sequences, we obtain the same result as would have been obtained with linear convolution. Thus with zero padding, the DFT can be used to perform linear filtering.

Unit -2

FAST FOURIER TRANSFORM

EFFICIENT COMPUTATION OF DFT:

In this section we represent several methods for computing dft efficiently. In view of the importance of the DFT in various digital signal processing applications such as linear filtering, correlation analysis and spectrum analysis, its efficient computation is a topic that has received considerably attention by many mathematicians, engineers and scientists. Basically the computation is done using the formula method.

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

where

$$W_N = e^{-j2\pi/N}$$

In general, the data sequence $x(n)$ is also assumed to be complex valued.

Similarly, the IDFT becomes

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

We observe that for each value of k, direct computation of $X(k)$ involves N complex multiplications ($4N$ real multiplications) and $N-1$ complex additions ($4N-2$ real additions). Consequently, to compute all N values of the DFT requires N^2 complex multiplications and $N^2 - N$ complex additions.

6.1.1 Direct Computation of the DFT

For a complex-valued sequence $x(n)$ of N points, the DFT may be expressed as

$$X_R(k) = \sum_{n=0}^{N-1} \left[x_R(n) \cos \frac{2\pi kn}{N} + x_I(n) \sin \frac{2\pi kn}{N} \right] \qquad (6.1.6)$$

$$X_I(k) = -\sum_{n=0}^{N-1} \left[x_R(n) \sin \frac{2\pi kn}{N} - x_I(n) \cos \frac{2\pi kn}{N} \right] \qquad (6.1.7)$$

The direct computation of (6.1.6) and (6.1.7) requires:

1. $2N^2$ evaluations of trigonometric functions.
2. $4N^2$ real multiplications.
3. $4N(N-1)$ real additions.
4. A number of indexing and addressing operations.

These operations are typical of DFT computational algorithms. The operations in items 2 and 3 result in the DFT values $X_R(k)$ and $X_I(k)$. The indexing and addressing operations are necessary to fetch the data $x(n)$, $0 \le n \le N-1$, and the phase factors and to store the results. The variety of DFT algorithms optimize each of these computational processes in a different way.

Divide-and-Conquer Approach to Computation of the DFT

The development of computationally efficient algorithms for the DFT is made possible if we adopt a divide-and-conquer approach. This approach is based on the decomposition of an N-point DFT into successively smaller DFT. This basic approach leads to a family o f computationally efficient algorithm s know n collectively as FFT algorithms.

T o illustrate the basic notions, let us consider the computation of an N point DFT , where N can be factored as a product of two integers, that is, $N = L\,M$

Algorithm 1

1. Store the signal column-wise.
2. Compute the M-point DFT of each row.
3. Multiply the resulting array by the phase factors W_N^{lq}.
4. Compute the L-point DFT of each column
5. Read the resulting array row-wise.

Algorithm 2

1. Store the signal row-wise.
2. Compute the L-point DFT at each column.
3. Multiply the resulting array by the factors W_N^{pm}.
4. Compute the M-point DFT of each row.
5. Read the resulting array column-wise.

6.1.3 Radix-2 FFT Algorithms

Let us consider the computation of the $N = 2^v$ point DFT by the divide-and-conquer approach specified by (6.1.16) through (6.1.18). We select $M = N/2$ and $L = 2$. This selection results in a split of the N-point data sequence into two $N/2$-point data sequences $f_1(n)$ and $f_2(n)$, corresponding to the even-numbered and odd-numbered samples of $x(n)$, respectively, that is,

$$\begin{aligned} f_1(n) &= x(2n) \\ f_2(n) &= x(2n+1), \qquad n = 0, 1, \ldots, \frac{N}{2} - 1 \end{aligned} \tag{6.1.23}$$

Thus $f_1(n)$ and $f_2(n)$ are obtained by decimating $x(n)$ by a factor of 2, and hence the resulting FFT algorithm is called a decimation-in-time algorithm.

Now the N-point DFT can be expressed in terms of the DFTs of the decimated sequences as follows:

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{kn} \qquad k = 0, 1, \ldots, N-1$$

$$= \sum_{n \text{ even}} x(n) W_N^{kn} + \sum_{n \text{ odd}} x(n) W_N^{kn} \tag{6.1.24}$$

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

But $W_N^2 = W_{N/2}$. With this substitution, (6.1.24) can be expressed as

$$\begin{aligned} X(k) &= \sum_{m=0}^{(N/2)-1} f_1(m) W_{N/2}^{km} + W_N^k \sum_{m=0}^{(N/2)-1} f_2(m) W_{N/2}^{km} \\ &= F_1(k) + W_N^k F_2(k) \qquad k = 0, 1, \ldots, N-1 \end{aligned} \tag{6.1.25}$$

where $F_1(k)$ and $F_2(k)$ are the $N/2$-point DFTs of the sequences $f_1(m)$ and $f_2(m)$, respectively.

Since $F_1(k)$ and $F_2(k)$ are periodic, with period $N/2$, we have $F_1(k + N/2) = F_1(k)$ and $F_2(k + N/2) = F_2(k)$. In addition, the factor $W_N^{k+N/2} = -W_N^k$. Hence (6.1.25) can be expressed as

$$X(k) = F_1(k) + W_N^k F_2(k) \qquad k = 0, 1, \ldots, \frac{N}{2} - 1 \tag{6.1.26}$$

$$X\left(k + \frac{N}{2}\right) = F_1(k) - W_N^k F_2(k) \qquad k = 0, 1, \ldots, \frac{N}{2} - 1 \tag{6.1.27}$$

To be consistent with our previous notation, we may define

$$G_1(k) = F_1(k) \qquad k = 0, 1, \ldots, \frac{N}{2} - 1$$

$$G_2(k) = W_N^k F_2(k) \qquad k = 0, 1, \ldots, \frac{N}{2} - 1$$

Then the DFT $X(k)$ may be expressed as

$$\begin{aligned} X(k) &= G_1(k) + G_2(k) \qquad k = 0, 1, \ldots, \frac{N}{2} - 1 \\ X(k + \frac{N}{2}) &= G_1(k) - G_2(k) \qquad k = 0, 1, \ldots, \frac{N}{2} - 1 \end{aligned} \tag{6.1.28}$$

$N/4$-point sequences

$$\begin{aligned} v_{11}(n) &= f_1(2n) \qquad n = 0, 1, \ldots, \frac{N}{4} - 1 \\ v_{12}(n) &= f_1(2n+1) \qquad n = 0, 1, \ldots, \frac{N}{4} - 1 \end{aligned} \tag{6.1.29}$$

and $f_2(n)$ would yield

$$\begin{aligned} v_{21}(n) &= f_2(2n) \qquad n = 0, 1, \ldots, \frac{N}{4} - 1 \\ v_{22}(n) &= f_2(2n+1) \qquad n = 0, 1, \ldots, \frac{N}{4} - 1 \end{aligned} \tag{6.1.30}$$

By computing $N/4$-point DFTs, we would obtain the $N/2$-point DFTs $F_1(k)$ and $F_2(k)$ from the relations

$$\begin{aligned} F_1(k) &= V_{11}(k) + W_{N/2}^k V_{12}(k) \qquad k = 0, 1, \ldots, \frac{N}{4} - 1 \\ F_1\left(k + \frac{N}{4}\right) &= V_{11}(k) - W_{N/2}^k V_{12}(k) \qquad k = 0, 1, \ldots, \frac{N}{4} - 1 \end{aligned} \tag{6.1.31}$$

$$\begin{aligned} F_2(k) &= V_{21}(k) + W_{N/2}^k V_{22}(k) \qquad k = 0, 1, \ldots, \frac{N}{4} - 1 \\ F_2\left(k + \frac{N}{4}\right) &= V_{21}(k) - W_{N/2}^k V_{22}(k) \qquad k = 0, \ldots, \frac{N}{4} - 1 \end{aligned} \tag{6.1.32}$$

where the $\{V_{ij}(k)\}$ are the $N/4$-point DFTs of the sequences $\{v_{ij}(n)\}$.

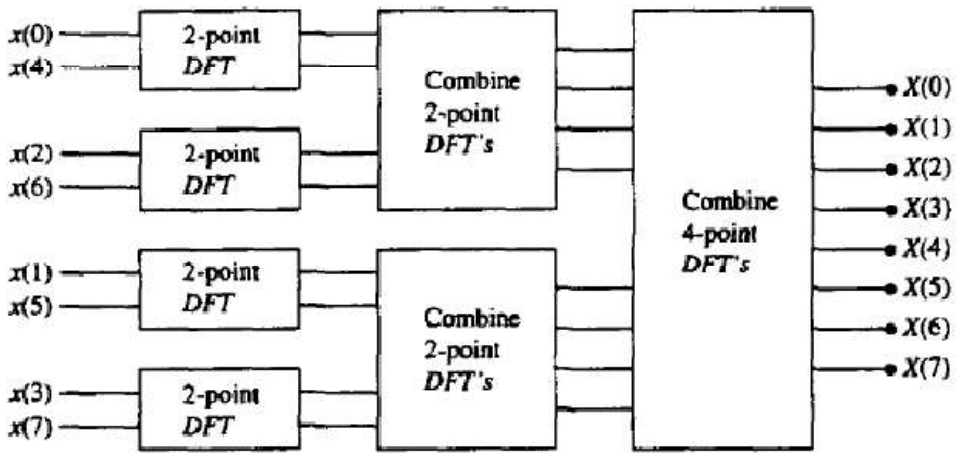


Figure 6.5 Three stages in the computation of an $N = 8$-point DFT.

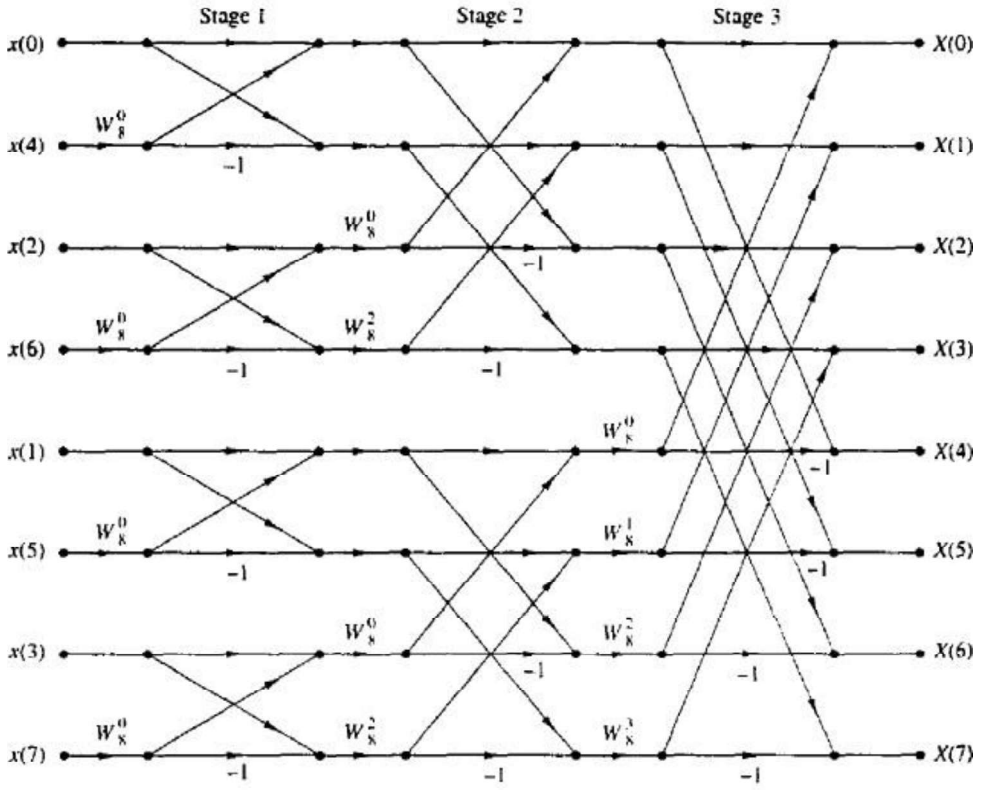


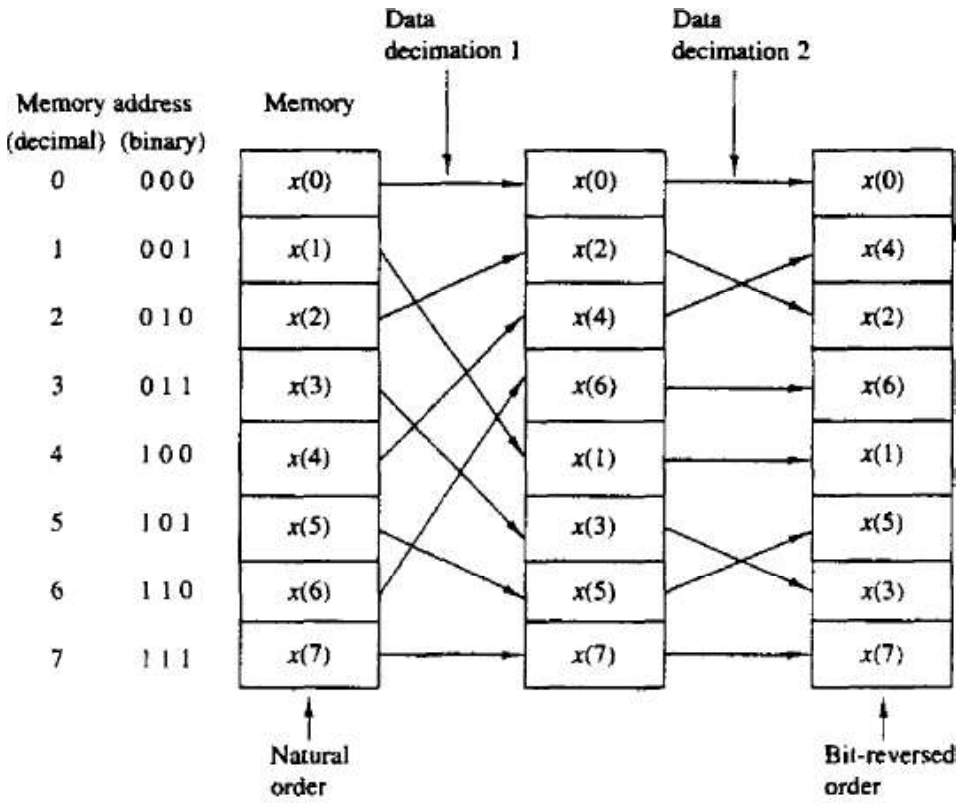
Figure 6.6 Eight-point decimation-in-time FFT algorithm.

Another important radix-2 FFT algorithm, called the decimation-in-frequency algorithm, is obtained by using the divide-and-conquer approach described in Section 6.1.2 with the choice of $M = 2$ and $L = N/2$. This choice of parameters implies a column-wise storage of the input data sequence. To derive the algorithm, we begin by splitting the DFT formula into two summations, one of which involves the sum over the first $N/2$ data points and the second sum involves the last $N/2$ data points. Thus we obtain

$$\begin{aligned} X(k) &= \sum_{n=0}^{(N/2)-1} x(n)W_N^{kn} + \sum_{n=N/2}^{N-1} x(n)W_N^{kn} \\ &= \sum_{n=0}^{(N/2)-1} x(n)W_N^{kn} + W_N^{Nk/2} \sum_{n=0}^{(N/2)-1} x\left(n+\frac{N}{2}\right) W_N^{kn} \end{aligned} \tag{6.1.33}$$

Since $W_N^{kN/2} = (-1)^k$, the expression (6.1.33) can be rewritten as

$$X(k) = \sum_{n=0}^{(N/2)-1} \left[x(n) + (-1)^k x\left(n+\frac{N}{2}\right)\right] W_N^{kn} \tag{6.1.34}$$



(a)

Now, let us split (decimate) $X(k)$ into the even- and odd-numbered samples. Thus we obtain

$$X(2k) = \sum_{n=0}^{(N/2)-1} \left[x(n) + x\left(n + \frac{N}{2}\right) \right] W_{N/2}^{kn} \qquad k = 0, 1, \ldots, \frac{N}{2} - 1 \tag{6.1.35}$$

and

$$X(2k+1) = \sum_{n=0}^{(N/2)-1} \left\{ \left[x(n) - x\left(n + \frac{N}{2}\right) \right] W_N^n \right\} W_{N/2}^{kn} \qquad k = 0, 1, \ldots, \frac{N}{2} - 1 \tag{6.1.36}$$

where we have used the fact that $W_N^2 = W_{N/2}$.

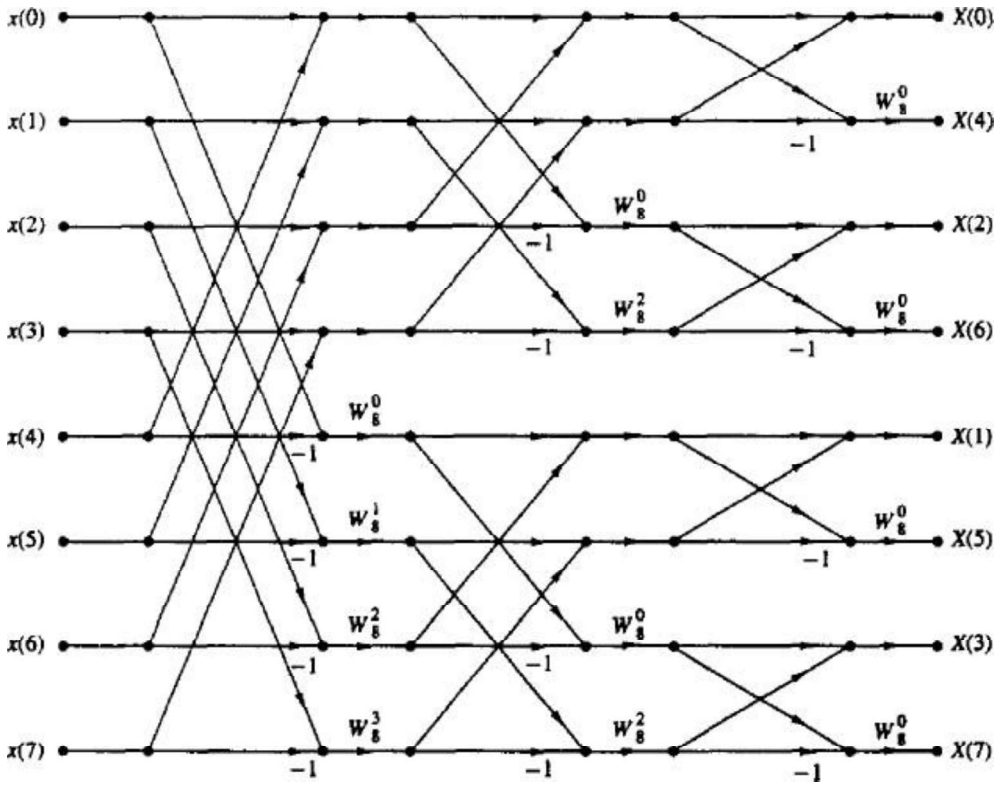
If we define the $N/2$-point sequences $g_1(n)$ and $g_2(n)$ as

$$\begin{aligned} g_1(n) &= x(n) + x\left(n + \frac{N}{2}\right) \\ g_2(n) &= \left[x(n) - x\left(n + \frac{N}{2}\right) \right] W_N^n \qquad n = 0, 1, 2, \ldots, \frac{N}{2} - 1 \end{aligned} \tag{6.1.37}$$

then

$$\begin{aligned} X(2k) &= \sum_{n=0}^{(N/2)-1} g_1(n) W_{N/2}^{kn} \\ X(2k+1) &= \sum_{n=0}^{(N/2)-1} g_2(n) W_{N/2}^{kn} \end{aligned} \tag{6.1.38}$$

We observe from Fig. 6.11, that the input data $x(n)$ occurs in natural order, but the output DFT occurs in bit-reversed order. We also note that the computations are performed in place. However, it is possible to reconfigure the decimation-in-frequency algorithm so that the input sequence occurs in bit-reversed order while the output DFT occurs in normal order. Furthermore, if we abandon the requirement that the computations be done in place, it is also possible to have both the input data and the output DFT in normal order.



Filter type	$Q(\omega)$	$P(\omega)$
$h(n) = h(M-1-n)$ M odd (case 1)	1	$\sum_{k=0}^{(M-1)/2} a(k)\cos\omega k$
$h(n) = h(M-1-n)$ M even (case 2)	$\cos\frac{\omega}{2}$	$\sum_{k=0}^{(M/2)-1} \tilde{b}(k)\cos\omega k$
$h(n) = -h(M-1-n)$ M odd (case 3)	$\sin\omega$	$\sum_{k=0}^{(M-3)/2} \tilde{c}(k)\cos\omega k$
$h(n) = -h(M-1-n)$ M even (case 4)	$\sin\frac{\omega}{2}$	$\sum_{k=0}^{(M/2)-1} \tilde{d}(k)\cos\omega k$

IIR FILTER DESIGN

DESIGN OF IIR FILTERS FROM ANALOG FILTERS

Just as in the design of FIR filters, there are several methods that can be used to design digital filters having an infinite-duration unit sample response. The techniques described in this section are all based on converting an analog filter into a digital filter. Analog filter design is a mature and well developed field, so it is not surprising that we begin the design of a digital filter in the analog domain and then convert the design into the digital domain.

An analog filter can be described by its system function.

$$H_a(s) = \frac{B(s)}{A(s)} = \frac{\sum_{k=0}^{M} \beta_k s^k}{\sum_{k=0}^{N} \alpha_k s^k} \tag{8.3.1}$$

where $\{\alpha_k\}$ and $\{\beta_k\}$ are the filter coefficients, or by its impulse response, which is related to $H_a(s)$ by the Laplace transform

$$H_a(s) = \int_{-\infty}^{\infty} h(t)e^{-st}\,dt \tag{8.3.2}$$

Alternatively, the analog filter having the rational system function $H(s)$ given in (8.3.1), can be described by the linear constant-coefficient differential equation

$$\sum_{k=0}^{N} \alpha_k \frac{d^k y(t)}{dt^k} = \sum_{k=0}^{M} \beta_k \frac{d^k x(t)}{dt^k} \tag{8.3.3}$$

where $x(t)$ denotes the input signal and $y(t)$ denotes the output of the filter.

Each of these three equivalent characterizations of an analog filter leads to alternative methods for converting the filter into the digital domain, as will be described in Sections 8.3.1 through 8.3.4. We recall that an analog linear time-invariant system with system function $H(s)$ is stable if all its poles lie in the left half of the s-plane. Consequently, if the conversion technique is to be effective, it should possess the following desirable properties:

1. The $j\Omega$ axis in the s-plane should map into the unit circle in the z-plane. Thus there will be a direct relationship between the two frequency variables in the two domains.

2. The left-half plane (LHP) of the s-plane should map into the inside of the unit circle in the z-plane. Thus a stable analog filter will be converted to a stable digital filter.

We mentioned in the preceding section that physically realizable and stable IIR filters cannot have linear phase. Recall that a linear-phase filter must have a system function that satisfies the condition

$$H(z) = \pm z^{-N} H(z^{-1}) \tag{8.3.4}$$

where z^{-N} represents a delay of N units of time. But if this were the case, the filter would have a mirror-image pole outside the unit circle for every pole inside the unit circle. Hence the filter would be unstable. Consequently, a causal and stable IIR filter cannot have linear phase.

If the restriction on physical realizability is removed, it is possible to obtain a linear-phase IIR filter, at least in principle. This approach involves performing a time reversal of the input signal $x(n)$, passing $x(-n)$ through a digital filter $H(z)$, time-reversing the output of $H(z)$, and finally, passing the result through $H(z)$ again. This signal processing is computationally cumbersome and appears to offer no advantages over linear-phase FIR filters. Consequently, when an application requires a linear-phase filter, it should be an FIR filter.

In the design of IIR filters, we shall specify the desired filter characteristics for the magnitude response only. This does not mean that we consider the phase response unimportant. Since the magnitude and phase characteristics are related, as indicated in Section 8.1, we specify the desired magnitude characteristics and accept the phase response that is obtained from the design methodology.

8.3.1 IIR Filter Design by Approximation of Derivatives

One of the simplest methods for converting an analog filter into a digital filter is to approximate the differential equation in (8.3.3) by an equivalent difference equation. This approach is often used to solve a linear constant-coefficient differential equation numerically on a digital computer.

For the derivative $dy(t)/dt$ at time $t = nT$, we substitute the *backward difference* $[y(nT) - y(nT - 1)]/T$. Thus

$$\left.\frac{dy(t)}{dt}\right|_{t=nT} = \frac{y(nT) - y(nT - T)}{T}$$

$$= \frac{y(n) - y(n - 1)}{T} \tag{8.3.5}$$

where T represents the sampling interval and $y(n) \equiv y(nT)$. The analog differentiator with output $dy(t)/dt$ has the system function $H(s) = s$, while the digital system that produces the output $[y(n) - y(n - 1)]/T$ has the system function $H(z) = (1 - z^{-1})/T$. Consequently, as shown in Fig. 8.29, the frequency-domain

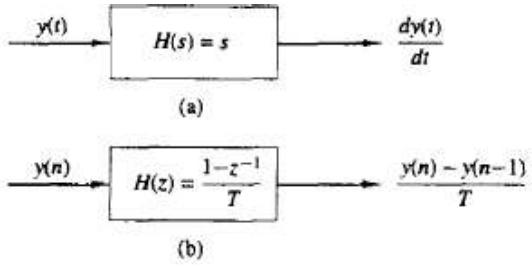


Figure 8.29 Substitution of the backward difference for the derivative implies the mapping $s = (1 - z^{-1})/T$.

equivalent for the relationship in (8.3.5) is

$$s = \frac{1 - z^{-1}}{T} \tag{8.3.6}$$

The second derivative $d^2y(t)/dt^2$ is replaced by the second difference, which is derived as follows:

$$\begin{aligned} \left.\frac{d^2y(t)}{dt^2}\right|_{t=nT} &= \frac{d}{dt}\left[\frac{dy(t)}{dt}\right]_{t=nT} \\ &= \frac{[y(nT) - y(nT - T)]/T - [y(nT - T) - y(nT - 2T)]/T}{T} \\ &= \frac{y(n) - 2y(n-1) + y(n-2)}{T^2} \end{aligned} \tag{8.3.7}$$

In the frequency domain, (8.3.7) is equivalent to

$$s^2 = \frac{1 - 2z^{-1} + z^{-2}}{T^2} = \left(\frac{1 - z^{-1}}{T}\right)^2 \tag{8.3.8}$$

It easily follows from the discussion that the substitution for the kth derivative of $y(t)$ results in the equivalent frequency-domain relationship

$$s^k = \left(\frac{1 - z^{-1}}{T}\right)^k \tag{8.3.9}$$

Consequently, the system function for the digital IIR filter obtained as a result of the approximation of the derivatives by finite differences is

$$H(z) = H_a(s)|_{s=(1-z^{-1})/T} \tag{8.3.10}$$

where $H_a(s)$ is the system function of the analog filter characterized by the differential equation given in (8.3.3).

Let us investigate the implications of the mapping from the s-plane to the z-plane as given by (8.3.6) or, equivalently,

$$z = \frac{1}{1 - sT} \tag{8.3.11}$$

If we substitute $s = j\Omega$ in (8.2.11), we find that

$$z = \frac{1}{1 - j\Omega T}$$

$$= \frac{1}{1+\Omega^2T^2} + j\frac{\Omega T}{1+\Omega^2T^2} \tag{8.3.12}$$

As Ω varies from $-\infty$ to ∞, the corresponding locus of points in the z-plane is a circle of radius $\frac{1}{2}$ and with center at $z = \frac{1}{2}$, as illustrated in Fig. 8.30.

It is easily demonstrated that the mapping in (8.3.11) takes points in the LHP of the s-plane into corresponding points inside this circle in the z-plane and points in the RHP of the s-plane are mapped into points outside this circle. Consequently, this mapping has the desirable property that a stable analog filter is transformed into a stable digital filter. However, the possible location of the poles of the digital filter are confined to relatively small frequencies and, as a consequence, the mapping is restricted to the design of lowpass filters and bandpass filters having relatively small resonant frequencies. It is not possible, for example, to transform a highpass analog filter into a corresponding highpass digital filter.

In an attempt to overcome the limitations in the mapping given above, more complex substitutions for the derivatives have been proposed. In particular, an Lth-order difference of the form

$$\left.\frac{dy(t)}{dt}\right|_{t=nT} = \frac{1}{T}\sum_{k=1}^{L}\alpha_k\frac{y(nT+kT) - y(nT-kT)}{T} \tag{8.3.13}$$

has been proposed, where $\{\alpha_k\}$ are a set of parameters that can be selected to optimize the approximation. The resulting mapping between the s-plane and the z-plane is now

$$s = \frac{1}{T}\sum_{k=1}^{L}\alpha_k(z^k - z^{-k}) \tag{8.3.14}$$

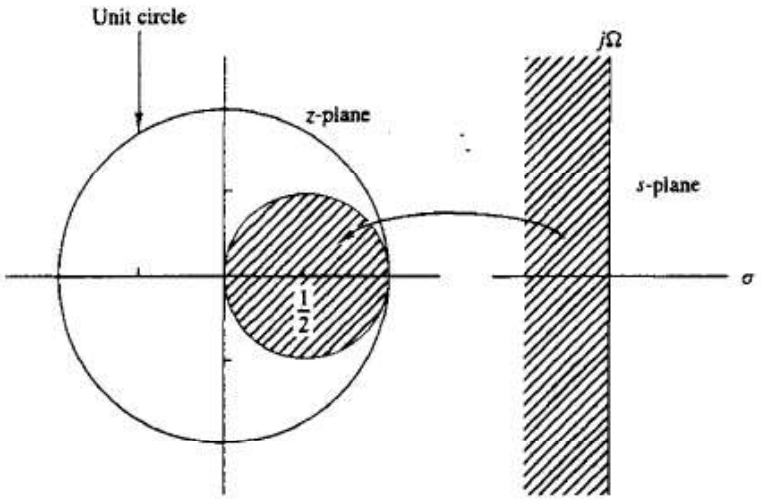


Figure 8.30 The mapping $s = (1 - z^{-1})/T$ takes LHP in the s-plane into points inside the circle of radius $\frac{1}{2}$ and center $z = \frac{1}{2}$ in the z-plane.

When $z = e^{j\omega}$, we have

$$s = j\frac{2}{T}\sum_{k=1}^{L}\alpha_k \sin \omega k \tag{8.3.15}$$

which is purely imaginary. Thus

$$\Omega = \frac{2}{T}\sum_{k=1}^{L}\alpha_k \sin \omega k \tag{8.3.16}$$

is the resulting mapping between the two frequency variables. By proper choice of the coefficients $\{\alpha_k\}$ it is possible to map the $j\Omega$-axis into the unit circle. Furthermore, points in the LHP in s can be mapped into points inside the unit circle in z.

Despite achieving the two desirable characteristics with the mapping of (8.3.16), the problem of selecting the set of coefficients $\{\alpha_k\}$ remains. In general, this is a difficult problem. Since simpler techniques exist for converting analog filters into IIR digital filters, we shall not emphasize the use of the Lth-order difference as a substitute for the derivative.

8.3.2 IIR Filter Design by Impulse Invariance

In the impulse invariance method, our objective is to design an IIR filter having a unit sample response $h(n)$ that is the sampled version of the impulse response of the analog filter. That is,

$$h(n) \equiv h(nT) \qquad n = 0, 1, 2, \ldots \tag{8.3.17}$$

where T is the sampling interval.

To examine the implications of (8.3.17), we refer back to Section 4.2.9. Recall that when a continuous time signal $x_a(t)$ with spectrum $X_a(F)$ is sampled at a rate $F_s = 1/T$ samples per second, the spectrum of the sampled signal is the periodic repetition of the scaled spectrum $F_s X_a(F)$ with period F_s. Specifically, the relationship is

$$X(f) = F_s \sum_{k=-\infty}^{\infty} X_a[(f - k)F_s] \tag{8.3.18}$$

where $f = F/F_s$ is the normalized frequency. Aliasing occurs if the sampling rate F_s is less than twice the highest frequency contained in $X_a(F)$.

Expressed in the context of sampling the impulse response of an analog filter with frequency response $H_a(F)$, the digital filter with unit sample response $h(n) \equiv h_a(nT)$ has the frequency response

$$H(f) = F_s \sum_{k=-\infty}^{\infty} H_a[(f - k)F_s] \tag{8.3.19}$$

or, equivalently,

$$H(\omega) = F_s \sum_{k=-\infty}^{\infty} H_a[(\omega - 2\pi k)F_s] \tag{8.3.20}$$

or

$$H(\Omega T) = \frac{1}{T} \sum_{k=-\infty}^{\infty} H_a\left(\Omega - \frac{2\pi k}{T}\right) \tag{8.3.21}$$

Figure 8.31 depicts the frequency response of a lowpass analog filter and the frequency response of the corresponding digital filter.

It is clear that the digital filter with frequency response $H(\omega)$ has the frequency response characteristics of the corresponding analog filter if the sampling interval T is selected sufficiently small to completely avoid or at least minimize the effects of aliasing. It is also clear that the impulse invariance method is inappropriate for designing highpass filters due the to spectrum aliasing that results from the sampling process.

To investigate the mapping of points between the z-plane and the s-plane implied by the sampling process, we rely on a generalization of (8.3.21) which relates the z-transform of $h(n)$ to the Laplace transform of $h_a(t)$. This relationship is

$$H(z)|_{z=e^{sT}} = \frac{1}{T} \sum_{k=-\infty}^{\infty} H_a\left(s - j\frac{2\pi k}{T}\right) \tag{8.3.22}$$

where

$$H(z) = \sum_{n=0}^{\infty} h(n)z^{-n}$$

$$H(z)|_{z=e^{sT}} = \sum_{n=0}^{\infty} h(n)e^{-sTn} \tag{8.3.23}$$

Note that when $s = j\Omega$, (8.3.22) reduces to (8.3.21), where the factor of j in $H_a(\Omega)$ is suppressed in our notation.

Let us consider the mapping of points from the s-plane to the z-plane implied by the relation

$$z = e^{sT} \tag{8.3.24}$$

If we substitute $s = \sigma + j\Omega$ and express the complex variable z in polar form as $z = re^{j\omega}$, (8.3.24) becomes

$$re^{j\omega} = e^{\sigma T}e^{j\Omega T}$$

Clearly, we must have

$$\begin{aligned} r &= e^{\sigma T} \\ \omega &= \Omega T \end{aligned} \tag{8.3.25}$$

Consequently, $\sigma < 0$ implies that $0 < r < 1$ and $\sigma > 0$ implies that $r > 1$. When $\sigma = 0$, we have $r = 1$. Therefore, the LHP in s is mapped inside the unit circle in z and the RHP in s is mapped outside the unit circle in z.

Also, the $j\Omega$-axis is mapped into the unit circle in z as indicated above. However, the mapping of the $j\Omega$-axis into the unit circle is not one-to-one. Since ω is unique over the range $(-\pi, \pi)$, the mapping $\omega = \Omega T$ implies that the interval $-\pi/T \leq \Omega \leq \pi/T$ maps into the corresponding values of $-\pi \leq \omega \leq \pi$. Furthermore, the frequency interval $\pi/T \leq \Omega \leq 3\pi/T$ also maps into the interval $-\pi \leq \omega \leq \pi$ and, in general, so does the interval $(2k-1)\pi/T \leq \Omega \leq (2k+1)\pi/T$, when k is an integer. Thus the mapping from the analog frequency Ω to the frequency variable ω in the digital domain is many-to-one, which simply reflects the effects of aliasing due to sampling. Figure 8.32 illustrates the mapping from the s-plane to the z-plane for the relation in (8.3.24).

To explore further the effect of the impulse invariance design method on the characteristics of the resulting filter, let us express the system function of the analog filter in partial-fraction form. On the assumption that the poles of the analog filter are distinct, we can write

$$H_a(s) = \sum_{k=1}^{N} \frac{c_k}{s - p_k} \tag{8.3.26}$$

where $\{p_k\}$ are the poles of the analog filter and $\{c_k\}$ are the coefficients in the partial-fraction expansion. Consequently,

$$h_a(t) = \sum_{k=1}^{N} c_k e^{p_k t} \qquad t \geq 0 \tag{8.3.27}$$

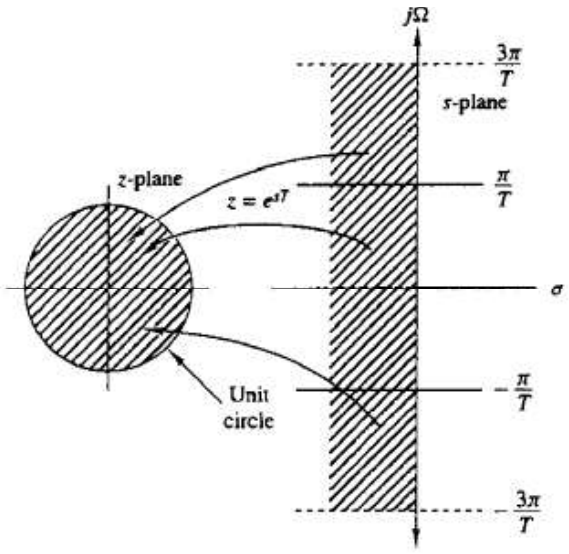


Figure 8.32 The mapping of $z = e^{sT}$ maps strips of width $2\pi/T$ (for $\sigma < 0$) in the s-plane into points in the unit circle in the z-plane.

If we sample $h_a(t)$ periodically at $t = nT$, we have

$$\begin{aligned} h(n) &= h_a(nT) \\ &= \sum_{k=1}^{N} c_k e^{p_k T n} \end{aligned} \tag{8.3.28}$$

Now, with the substitution of (8.3.28), the system function of the resulting digital IIR filter becomes

$$\begin{aligned} H(z) &= \sum_{n=0}^{\infty} h(n) z^{-n} \\ &= \sum_{n=0}^{\infty} \left(\sum_{k=1}^{N} c_k e^{p_k T n} \right) z^{-n} \\ &= \sum_{k=1}^{N} c_k \sum_{n=0}^{\infty} (e^{p_k T} z^{-1})^n \end{aligned} \tag{8.3.29}$$

The inner sum in (8.3.29) converges because $p_k < 0$ and yields

$$\sum_{n=0}^{\infty} (e^{p_k T} z^{-1})^n = \frac{1}{1 - e^{p_k T} z^{-1}} \tag{8.3.30}$$

Therefore, the system function of the digital filter is

$$H(z) = \sum_{k=1}^{N} \frac{c_k}{1 - e^{p_k T} z^{-1}} \tag{8.3.31}$$

We observe that the digital filter has poles at

$$z_k = e^{p_k T} \qquad k = 1, 2, \ldots, N \tag{8.3.32}$$

8.3.3 IIR Filter Design by the Bilinear Transformation

The IIR filter design techniques described in the preceding two sections have a severe limitation in that they are appropriate only for lowpass filters and a limited class of bandpass filters.

In this section we describe a mapping from the s-plane to the z-plane, called the bilinear transformation, that overcomes the limitation of the other two design

methods described previously. The bilinear transformation is a conformal mapping that transforms the $j\Omega$-axis into the unit circle in the z-plane only once, thus avoiding aliasing of frequency components. Furthermore, all points in the LHP of s are mapped inside the unit circle in the z-plane and all points in the RHP of s are mapped into corresponding points outside the unit circle in the z-plane.

The bilinear transformation can be linked to the trapezoidal formula for numerical integration. For example, let us consider an analog linear filter with system function

$$H(s) = \frac{b}{s + a} \tag{8.3.33}$$

This system is also characterized by the differential equation

$$\frac{dy(t)}{dt} + ay(t) = bx(t) \tag{8.3.34}$$

Instead of substituting a finite difference for the derivative, suppose that we integrate the derivative and approximate the integral by the trapezoidal formula. Thus

$$y(t) = \int_{t_0}^{t} y'(\tau)d\tau + y(t_0) \tag{8.3.35}$$

where $y'(t)$ denotes the derivative of $y(t)$. The approximation of the integral in (8.3.35) by the trapezoidal formula at $t = nT$ and $t_0 = nT - T$ yields

$$y(nT) = \frac{T}{2}[y'(nT) + y'(nT - T)] + y(nT - T) \tag{8.3.36}$$

Now the differential equation in (8.3.34) evaluated at $t = nT$ yields

$$y'(nT) = -ay(nT) + bx(nT) \tag{8.3.37}$$

We use (8.3.37) to substitute for the derivative in (8.3.36) and thus obtain a difference equation for the equivalent discrete-time system. With $y(n) \equiv y(nT)$ and $x(n) \equiv x(nT)$, we obtain the result

$$\left(1 + \frac{aT}{2}\right) y(n) - \left(1 - \frac{aT}{2}\right) y(n - 1) = \frac{bT}{2}[x(n) + x(n - 1)] \tag{8.3.38}$$

The z-transform of this difference equation is

$$\left(1 + \frac{aT}{2}\right) Y(z) - \left(1 - \frac{aT}{2}\right) z^{-1}Y(z) = \frac{bT}{2}(1 + z^{-1})X(z)$$

Consequently, the system function of the equivalent digital filter is

$$H(z) = \frac{Y(z)}{X(z)} = \frac{(bT/2)(1 + z^{-1})}{1 + aT/2 - (1 - aT/2)z^{-1}}$$

or, equivalently,

$$H(z) = \frac{b}{\frac{2}{T}\left(\frac{1 - z^{-1}}{1 + z^{-1}}\right) + a} \tag{8.3.39}$$

Clearly, the mapping from the s-plane to the z-plane is

$$s = \frac{2}{T}\left(\frac{1 - z^{-1}}{1 + z^{-1}}\right) \tag{8.3.40}$$

This is called the *bilinear transformation*.

Although our derivation of the bilinear transformation was performed for a first-order differential equation, it holds, in general, for an Nth-order differential equation.

To investigate the characteristics of the bilinear transformation, let

$$z = re^{j\omega}$$

$$s = \sigma + j\Omega$$

Then (8.3.40) can be expressed as

$$\begin{aligned} s &= \frac{2}{T}\frac{z-1}{z+1} \\ &= \frac{2}{T}\frac{re^{j\omega} - 1}{re^{j\omega} + 1} \\ &= \frac{2}{T}\left(\frac{r^2 - 1}{1 + r^2 + 2r\cos\omega} + j\frac{2r\sin\omega}{1 + r^2 + 2r\cos\omega}\right) \end{aligned}$$

Consequently,

$$\sigma = \frac{2}{T}\frac{r^2 - 1}{1 + r^2 + 2r\cos\omega} \tag{8.3.41}$$

$$\Omega = \frac{2}{T}\frac{2r\sin\omega}{1 + r^2 + 2r\cos\omega} \tag{8.3.42}$$

First, we note that if $r < 1$, then $\sigma < 0$, and if $r > 1$, then $\sigma > 0$. Consequently, the LHP in s maps into the inside of the unit circle in the z-plane and the RHP in s maps into the outside of the unit circle. When $r = 1$, then $\sigma = 0$ and

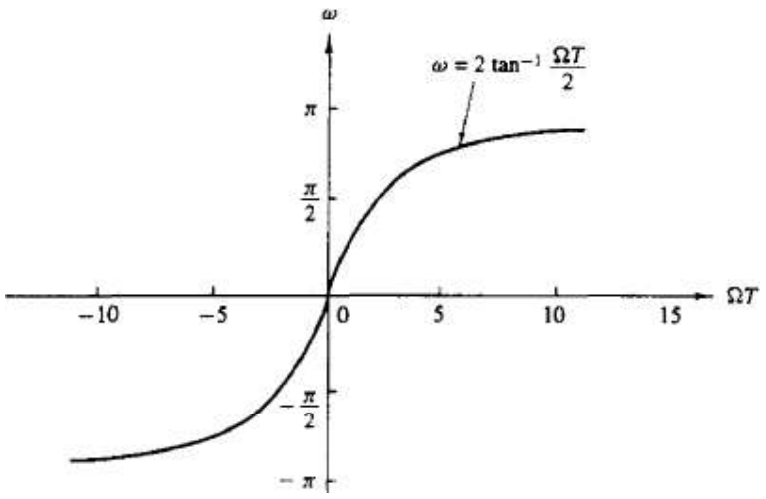
$$\begin{aligned} \Omega &= \frac{2}{T}\frac{\sin\omega}{1 + \cos\omega} \\ &= \frac{2}{T}\tan\frac{\omega}{2} \end{aligned} \tag{8.3.43}$$

or, equivalently,

$$\omega = 2\tan^{-1}\frac{\Omega T}{2} \tag{8.3.44}$$

The relationship in (8.3.44) between the frequency variables in the two domains is illustrated in Fig. 8.36. We observe that the entire range in Ω is mapped only once into the range $-\pi \le \omega \le \pi$. However, the mapping is highly nonlinear. We observe a frequency compression or *frequency warping*, as it is usually called, due to the nonlinearity of the arctangent function.

It is also interesting to note that the bilinear transformation maps the point $s = \infty$ into the point $z = -1$. Consequently, the single-pole lowpass filter in



Chebyshev filters. There are two types of Chebyshev filters. Type I Chebyshev filters are all-pole filters that exhibit equiripple behavior in the passband and a monotonic characteristic in the stopband. On the other hand, the family of type II Chebyshev filters contains both poles and zeros and exhibits a monotonic behavior in the passband and an equiripple behavior in the stopband. The zeros of this class of filters lie on the imaginary axis in the s-plane.

The magnitude squared of the frequency response characteristic of a type I Chebyshev filter is given as

$$|H(\Omega)|^2 = \frac{1}{1 + \epsilon^2 T_N^2(\Omega/\Omega_p)} \tag{8.3.51}$$

where ϵ is a parameter of the filter related to the ripple in the passband and $T_N(x)$ is the Nth-order Chebyshev polynomial defined as

$$T_N(x) = \begin{cases} \cos(N \cos^{-1} x), & |x| \le 1 \\ \cosh(N \cosh^{-1} x), & |x| > 1 \end{cases} \tag{8.3.52}$$

The Chebyshev polynomials can be generated by the recursive equation

$$T_{N+1}(x) = 2x\,T_N(x) - T_{N-1}(x) \qquad N = 1, 2, \ldots \tag{8.3.53}$$

where $T_0(x) = 1$ and $T_1(x) = x$. From (8.3.53) we obtain $T_2(x) = 2x^2 - 1$, $T_3(x) = 4x^3 - 3x$, and so on.

Some of the properties of these polynomials are as follows:

1. $|T_N(x)| \le 1$ for all $|x| \le 1$.
2. $T_N(1) = 1$ for all N.
3. All the roots of the polynomial $T_N(x)$ occur in the interval $-1 \le x \le 1$.

The filter parameter ϵ is related to the ripple in the passband, as illustrated in Fig. 8.39, for N odd and N even. For N odd, $T_N(0) = 0$ and hence $|H(0)|^2 = 1$. On the other hand, for N even, $T_N(0) = 1$ and hence $|H(0)|^2 = 1/(1 + \epsilon^2)$. At the band edge frequency $\Omega = \Omega_p$, we have $T_N(1) = 1$, so that

$$\frac{1}{\sqrt{1+\epsilon^2}} = 1 - \delta_1$$

or, equivalently,

$$\epsilon^2 = \frac{1}{(1-\delta_1)^2} - 1 \tag{8.3.54}$$

where δ_1 is the value of the passband ripple.

The poles of a type I Chebyshev filter lie on an ellipse in the s-plane with major axis

$$r_1 = \Omega_p \frac{\beta^2 + 1}{2\beta} \tag{8.3.55}$$

and minor axis

$$r_2 = \Omega_p \frac{\beta^2 - 1}{2\beta} \tag{8.3.56}$$

where β is related to ϵ according to the equation

$$\beta = \left[\frac{\sqrt{1+\epsilon^2}+1}{\epsilon}\right]^{1/N} \tag{8.3.57}$$

The pole locations are most easily determined for a filter of order N by first locating the poles for an equivalent Nth-order Butterworth filter that lie on circles of radius r_1 or radius r_2, as illustrated in Fig. 8.40. If we denote the angular positions of the poles of the Butterworth filter as

$$\phi_k = \frac{\pi}{2} + \frac{(2k+1)\pi}{2N} \qquad k = 0, 1, 2, \ldots, N-1 \tag{8.3.58}$$

then the positions of the poles for the Chebyshev filter lie on the ellipse at the coordinates (x_k, y_k), $k = 0, 1, \ldots, N-1$, where

$$\begin{aligned} x_k &= r_2 \cos\phi_k, \qquad k = 0, 1, \ldots, N-1 \\ y_k &= r_1 \sin\phi_k, \qquad k = 0, 1, \ldots, N-1 \end{aligned} \tag{8.3.59}$$

A type II Chebyshev filter contains zeros as well as poles. The magnitude squared of its frequency response is given as

$$|H(\Omega)|^2 = \frac{1}{1 + \epsilon^2[T_N^2(\Omega_s/\Omega_p)/T_N^2(\Omega_s/\Omega)]} \tag{8.3.60}$$

where $T_N(x)$ is, again, the Nth-order Chebyshev polynomial and Ω_s is the stopband frequency as illustrated in Fig. 8.41. The zeros are located on the imaginary axis at the points

$$s_k = j\frac{\Omega_s}{\sin\phi_k} \qquad k = 0, 1, \ldots, N-1 \tag{8.3.61}$$

The poles are located at the points (v_k, w_k), where

$$v_k = \frac{\Omega_s x_k}{\sqrt{x_k^2 + y_k^2}} \qquad k = 0, 1, \ldots, N-1 \tag{8.3.62}$$

$$w_k = \frac{\Omega_s y_k}{\sqrt{x_k^2 + y_k^2}} \qquad k = 0, 1, \ldots, N-1 \tag{8.3.63}$$

where $\{x_k\}$ and $\{y_k\}$ are defined in (8.3.59) with β now related to the ripple in the stopband through the equation

$$\beta = \left[\frac{1 + \sqrt{1 - \delta_2^2}}{\delta_2}\right]^{1/N} \tag{8.3.64}$$

$$N = \frac{\log\left[\left(\sqrt{1 - \delta_2^2} + \sqrt{1 - \delta_2^2(1 + \epsilon^2)}\right)/\epsilon\delta_2\right]}{\log\left[(\Omega_s/\Omega_p) + \sqrt{(\Omega_s/\Omega_p)^2 - 1}\right]}$$

$$= \frac{\cosh^{-1}(\delta/\epsilon)}{\cosh^{-1}(\Omega_s/\Omega_p)}$$

where, by definition, $\delta_2 = 1/\sqrt{1 + \delta^2}$.

Frequency Transformations in the Analog Domain

Type of transformation	Transformation	Band edge frequencies of new filter
Lowpass	$s \longrightarrow \frac{\Omega_p}{\Omega_p'} s$	Ω_p'
Highpass	$s \longrightarrow \frac{\Omega_p \Omega_p'}{s}$	Ω_p'
Bandpass	$s \longrightarrow \Omega_p \frac{s^2 + \Omega_l \Omega_u}{s(\Omega_u - \Omega_l)}$	Ω_l, Ω_u
Bandstop	$s \longrightarrow \Omega_p \frac{s(\Omega_u - \Omega_l)}{s^2 + \Omega_u \Omega_l}$	Ω_l, Ω_u

Frequency Transformations in the Digital Domain

Type of transformation	Transformation	Parameters
Lowpass	$z^{-1} \longrightarrow \dfrac{z^{-1} - a}{1 - az^{-1}}$	ω_p' = band edge frequency of new filter $a = \dfrac{\sin[(\omega_p - \omega_p')/2]}{\sin[(\omega_p + \omega_p')/2]}$
Highpass	$z^{-1} \longrightarrow -\dfrac{z^{-1} + a}{1 + az^{-1}}$	ω_p' = band edge frequency new filter $a = -\dfrac{\cos[(\omega_p + \omega_p')/2]}{\cos[(\omega_p - \omega_p')/2]}$
Bandpass	$z^{-1} \longrightarrow -\dfrac{z^{-2} - a_1 z^{-1} + a_2}{a_2 z^{-2} - a_1 z^{-1} + 1}$	ω_l = lower band edge frequency ω_u = upper band edge frequency $a_1 = -2\alpha K/(K + 1)$ $a_2 = (K - 1)/(K + 1)$ $\alpha = \dfrac{\cos[(\omega_u + \omega_l)/2]}{\cos[(\omega_u - \omega_l)/2]}$ $K = \cot\dfrac{\omega_u - \omega_l}{2}\tan\dfrac{\omega_p}{2}$
Bandstop	$z^{-1} \longrightarrow -\dfrac{z^{-2} - a_1 z^{-1} + a_2}{a_2 z^{-2} - a_1 z^{-1} + 1}$	ω_l = lower band edge frequency ω_u = upper band edge frequency $a_1 = -2\alpha/(K + 1)$ $a_2 = (1 - K)/(1 + K)$ $\alpha = \dfrac{\cos[(\omega_u + \omega_l)/2]}{\cos[(\omega_u - \omega_l)/2]}$ $K = \tan\dfrac{\omega_u - \omega_l}{2}\tan\dfrac{\omega_p}{2}$

****

Conversion of direct-form FIR filter coefficients to lattice coefficients. Suppose that we are given the FIR coefficients for the direct-form realization or, equivalently, the polynomial $A_m(z)$, and we wish to determine the corresponding lattice filter parameters $\{K_i\}$. For the m-stage lattice we immediately obtain the parameter $K_m = \alpha_m(m)$. To obtain K_{m-1} we need the polynomials $A_{m-1}(z)$ since, in general, K_m is obtained from the polynomial $A_m(z)$ for $m = M-1, M-2, \ldots, 1$. Consequently, we need to compute the polynomials $A_m(z)$ starting from $m = M-1$ and "stepping down" successively to $m = 1$.

$$K_m = \alpha_m(m) \quad \alpha_{m-1}(0) = 1$$

$$\alpha_{m-1}(k) = \frac{\alpha_m(k) - K_m\beta_m(k)}{1 - K_m^2}$$

$$= \frac{\alpha_m(k) - \alpha_m(m)\alpha_m(m-k)}{1 - \alpha_m^2(m)} \qquad 1 \le k \le m-1$$

STRUCTURES FOR IIR SYSTEMS

In this section we consider different IIR system s structures described by the difference equation given by the system function. Just as in the case o f FIR system s, there are several types o f structures or realizations, including direct-form structures, cascade-form structures, lattice structures, and lattice-ladder structures. In addition, IIR systems lend them selves to a parallel form realization. We begin by describing two direct-form realizations.

DIRECT FORM STRUCTURES:

The rational system function as given by (7.1.2) that characterizes an IIR system can be viewed as two systems in cascade, that is,

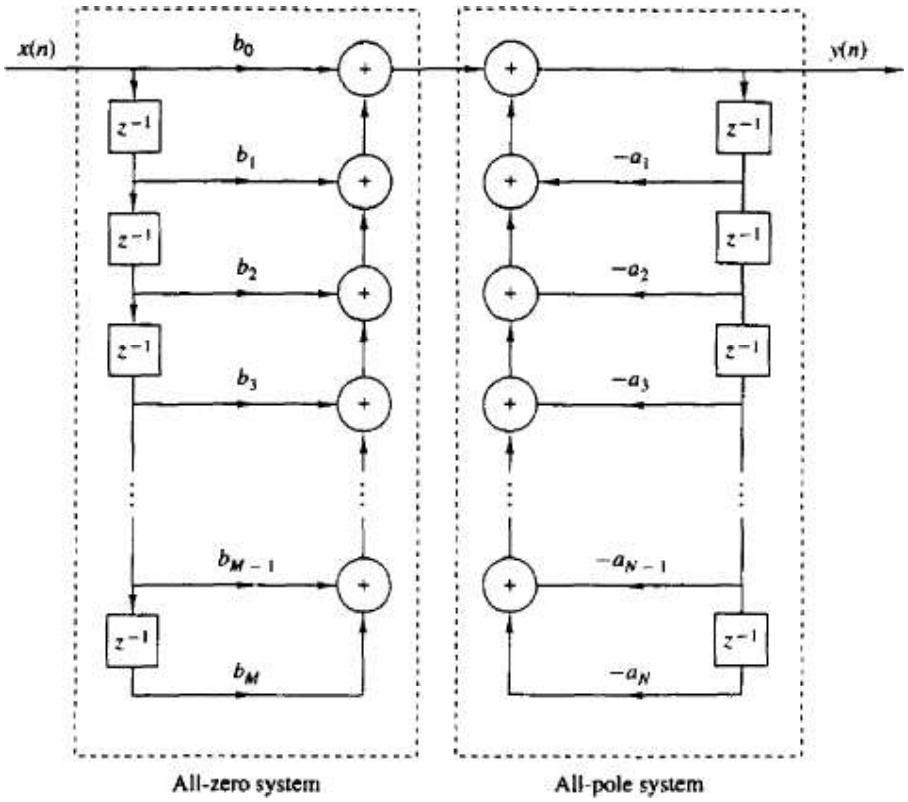
$$H(z) = H_1(z)H_2(z) \tag{7.3.1}$$

where $H_1(z)$ consists of the zeros of $H(z)$, and $H_2(z)$ consists of the poles of $H(z)$,

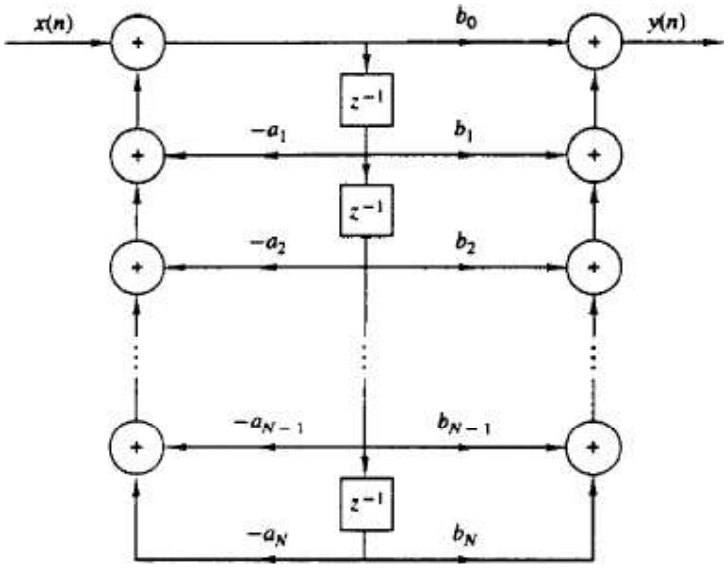
$$H_1(z) = \sum_{k=0}^{M} b_k z^{-k} \tag{7.3.2}$$

and

$$H_2(z) = \frac{1}{1 + \sum_{k=1}^{N} a_k z^{-k}} \tag{7.3.3}$$

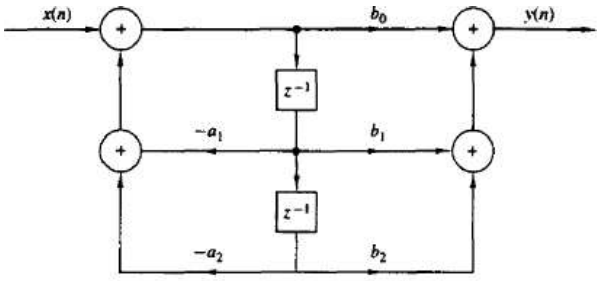


DIRECT FORM II

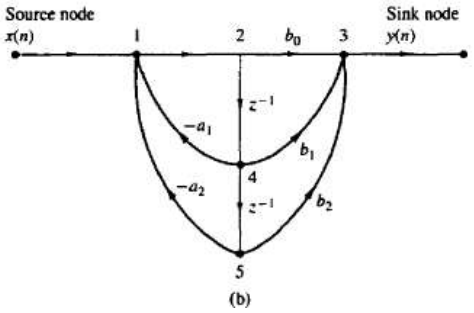


Signal Flow Graphs and Transposed Structures

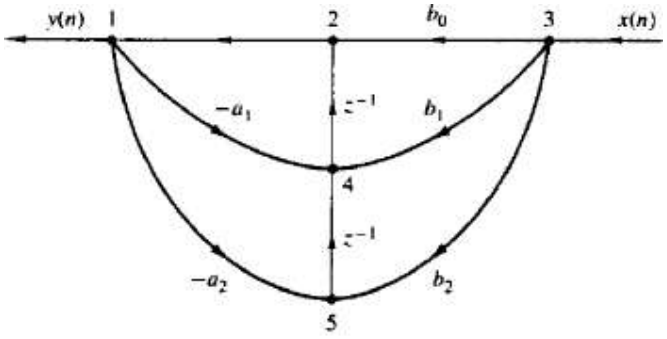
A signal flow graph provides an alternative, N but equivalent, graphical representation to a block diagram structure that we have been using to illustrate various system realizations. T he basic elements o f a flow graph are branches and nodes. A signal flow graph is basically a set o f directed branches that connect at nodes. By definition, the signal out of a branch is equal to the branch gain (system function) times the signal into the branch. Furthermore, the signal at anode o f a flow graph is equal to the sum o f the signals from all branches connecting to the node.



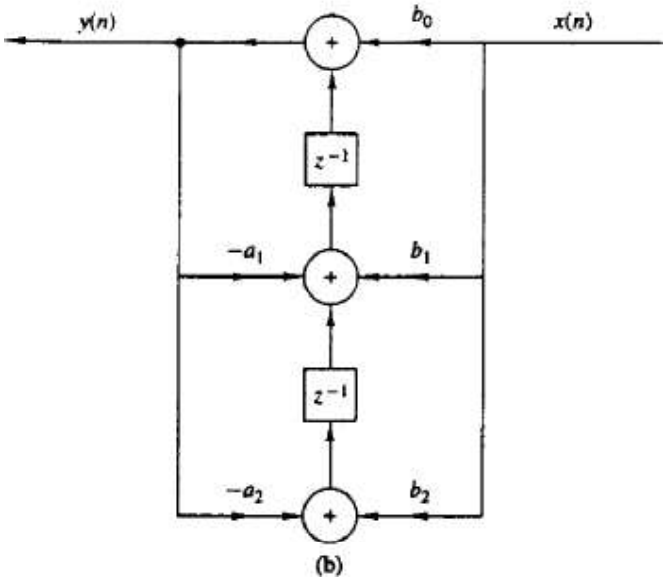
(a)



(b)



(a)



(b)

Cascade-Form Structures

Let us consider a high-order IIR system with system function given by equation. Without loss o f generality we assume that $N > M$. **T h e system can be factored into a cascade o f second-order subsystem s, such that** H (z) **can b e expressed as**

$$H(z) = \prod_{k=1}^{K} H_k(z)$$

where K is the integer part of $(N+1)/2$. $H_k(z)$ has the general form

$$H_k(z) = \frac{b_{k0} + b_{k1}z^{-1} + b_{k2}z^{-2}}{1 + a_{k1}z^{-1} + a_{k2}z^{-2}}$$

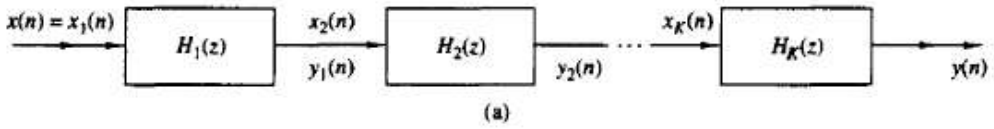
The general form of the cascade structure is illustrated in Fig. 7.19. If we use the direct form II structure for each of the subsystems, the computational algorithm for realizing the IIR system with system function $H(z)$ is described by the following set of equations.

$$y_0(n) = x(n) \tag{7.3.16}$$

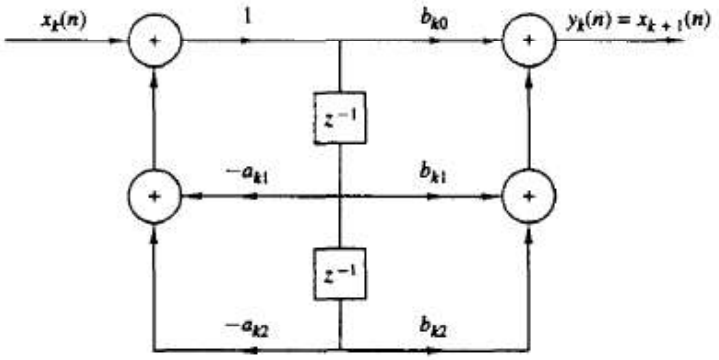
$$w_k(n) = -a_{k1}w_k(n-1) - a_{k2}w_k(n-2) + y_{k-1}(n) \qquad k = 1, 2, \ldots, K \tag{7.3.17}$$

$$y_k(n) = b_{k0}w_k(n) + b_{k1}w_k(n-1) + b_{k2}w_k(n-2) \qquad k = 1, 2, \ldots, K \tag{7.3.18}$$

$$y(n) = y_K(n) \tag{7.3.19}$$



(a)

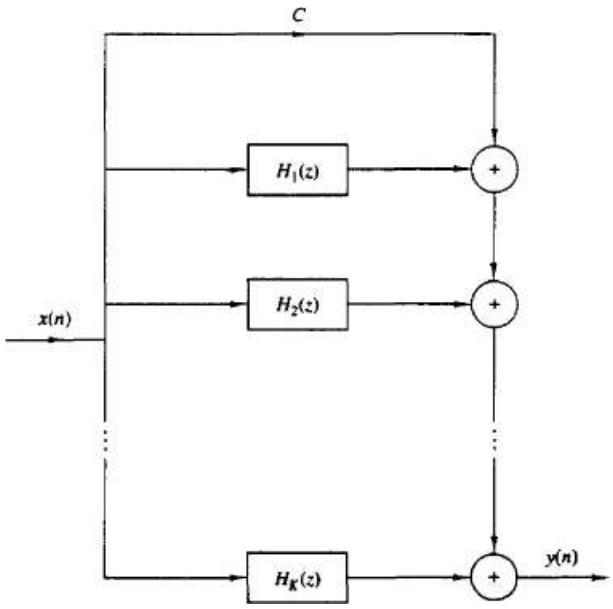


(b)

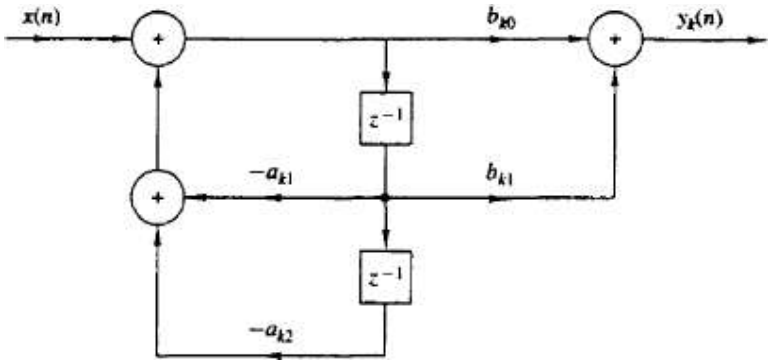
Parallel-Form Structures

A parallel-form realization o f an IIR system can be obtained by performing a partial-fraction expansion o f $H(z)$. Without loss o f generality, w e again assume that $N > M$ and that the poles are distinct. Then, by performing a partial-fraction expansion o f $H(z)$, we obtain the result

$$H(z) = C + \sum_{k=1}^{N} \frac{A_k}{1 - p_k z^{-1}}$$



The realization of second order form is given by



The general form of parallel form of structure is f\given by

$$w_k(n) = -a_{k1}w_k(n-1) - a_{k2}w_k(n-2) + x(n) \qquad k = 1, 2, \ldots, K$$

$$y_k(n) = b_{k0}w_k(n) + b_{k1}w_k(n-1) \qquad k = 1, 2, \ldots, K$$

$$y(n) = Cx(n) + \sum_{k=1}^{K} y_k(n)$$

Lattice and Lattice-Ladder Structures for IIR Systems

In Section 7.2.4 we developed a lattice filter structure that is equivalent to an FIR system. In this section we extend the development to IIR systems.

Let us begin with an all-pole system with system function

$$H(z) = \frac{1}{1 + \sum_{k=1}^{N} a_N(k)z^{-k}} = \frac{1}{A_N(z)} \tag{7.3.26}$$

The direct form realization of this system is illustrated in Fig. 7.23. The difference equation for this IIR system is

$$y(n) = -\sum_{k=1}^{N} a_N(k)y(n-k) + x(n) \tag{7.3.27}$$

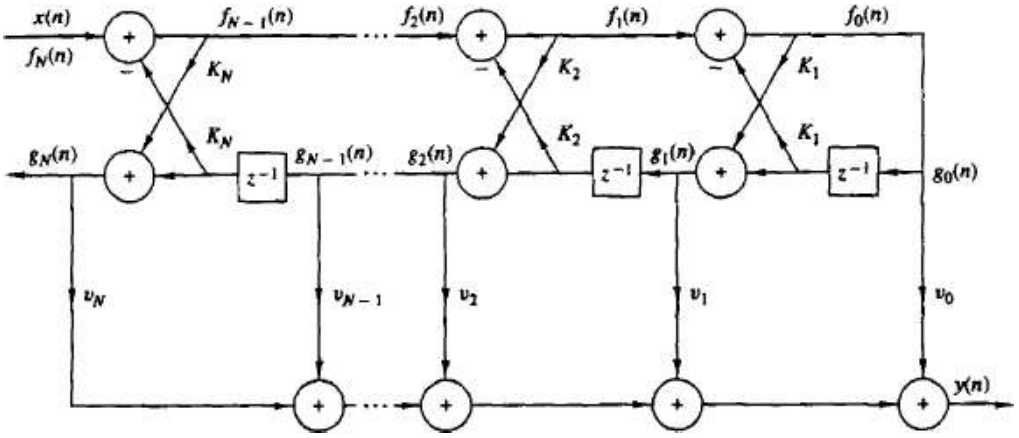
It is interesting to note that if we interchange the roles of input and output [i.e., interchange $x(n)$ with $y(n)$ in (7.3.27)], we obtain

$$x(n) = -\sum_{k=1}^{N} a_N(k)x(n-k) + y(n)$$

or, equivalently,

$$y(n) = x(n) + \sum_{k=1}^{N} a_N(k)x(n-k) \tag{7.3.28}$$

We note that the equation in (7.3.28) describes an FIR system having the system function $H(z) = A_N(z)$, while the system described by the difference equation in (7.3.27) represents an IIR system with system function $H(z) = 1/A_N(z)$.



UNIT – 4
IIR AND FIR FILTERS

The transfer function is obtained by taking Z transform of finite sample impulse response. The filters designed by using finite samples of impulse response are called FIR filters.

Some of the advantages of FIR filter are linear phase, both recursive and non recursive, stable and round off noise can be made smaller.

Some of the disadvantages of FIR filters are large amount of processing is required and non integral delay may lead to problems.

DESIGN OF FIR FILTERS

An FIR filter of length M with input $x(n)$ and output $y(n)$ is described by the difference equation

$$\begin{aligned} y(n) &= b_0x(n) + b_1x(n-1) + \cdots + b_{M-1}x(n-M+1) \\ &= \sum_{k=0}^{M-1} b_k x(n-k) \end{aligned} \tag{8.2.1}$$

where $\{b_k\}$ is the set of filter coefficients. Alternatively, we can express the output sequence as the convolution of the unit sample response $h(n)$ of the system with the input signal. Thus we have

$$y(n) = \sum_{k=0}^{M-1} h(k)x(n-k) \tag{8.2.2}$$

where the lower and upper limits on the convolution sum reflect the causality and finite-duration characteristics of the filter. Clearly, (8.2.1) and (8.2.2) are identical in form and hence it follows that $b_k = h(k)$, $k = 0, 1, \ldots, M-1$.

The filter can also be characterized by its system function

$$H(z) = \sum_{k=0}^{M-1} h(k)z^{-k} \tag{8.2.3}$$

which we view as a polynomial of degree $M-1$ in the variable z^{-1}. The roots of this polynomial constitute the zeros of the filter.

An FIR filter has linear phase if its unit sample response satisfies the condition

$$h(n) = \pm h(M-1-n) \qquad n = 0, 1, \ldots, M-1 \tag{8.2.4}$$

When the symmetry and antisymmetry conditions in (8.2.4) are incorporated into (8.2.3), we have

$$\begin{aligned} H(z) &= h(0) + h(1)z^{-1} + h(2)z^{-2} + \cdots + h(M-2)z^{-(M-2)} + h(M-1)z^{-(M-1)} \\ &= z^{-(M-1)/2}\left\{h\left(\frac{M-1}{2}\right) + \sum_{n=0}^{(M-3)/2} h(n)\left[z^{(M-1-2k)/2} \pm z^{-(M-1-2k)/2}\right]\right\} \quad M \text{ odd} \\ &= z^{-(M-1)/2}\sum_{n=0}^{(M/2)-1} h(n)[z^{(M-1-2k)/2} \pm z^{-(M-1-2k)/2}] \qquad M \text{ even} \end{aligned} \tag{8.2.5}$$

Now, if we substitute z^{-1} for z in (8.2.3) and multiply both sides of the resulting equation by $z^{-(M-1)}$, we obtain

$$z^{-(M-1)}H(z^{-1}) = \pm H(z) \tag{8.2.6}$$

When $h(n) = h(M-1-n)$, $H(\omega)$ can be expressed as

$$H(\omega) = H_r(\omega)e^{-j\omega(M-1)/2} \tag{8.2.7}$$

where $H_r(\omega)$ is a real function of ω and can be expressed as

$$H_r(\omega) = h\left(\frac{M-1}{2}\right) + 2\sum_{n=0}^{(M-3)/2} h(n)\cos\omega\left(\frac{M-1}{2} - n\right) \qquad M \text{ odd} \tag{8.2.8}$$

$$H_r(\omega) = 2\sum_{n=0}^{(M/2)-1} h(n)\cos\omega\left(\frac{M-1}{2} - n\right) \qquad M \text{ even} \tag{8.2.9}$$

The phase characteristic of the filter for both M odd and M even is

$$\Theta(\omega) = \begin{cases} -\omega\left(\dfrac{M-1}{2}\right), & \text{if } H_r(\omega) > 0 \\ -\omega\left(\dfrac{M-1}{2}\right) + \pi, & \text{if } H_r(\omega) < 0 \end{cases} \tag{8.2.10}$$

When

$$h(n) = -h(M-1-n)$$

the unit sample response is *antisymmetric*. For M odd, the center point of the antisymmetric $h(n)$ is $n = (M-1)/2$. Consequently,

$$h\left(\frac{M-1}{2}\right) = 0$$

It is straightforward to show that the frequency response of an FIR filter with an antisymmetric unit sample response can be expressed as

$$H(\omega) = H_r(\omega)e^{j[-\omega(M-1)/2 + \pi/2]} \tag{8.2.11}$$

where

$$H_r(\omega) = 2\sum_{n=0}^{(M-3)/2} h(n)\sin\omega\left(\frac{M-1}{2} - n\right) \qquad M \text{ odd} \tag{8.2.12}$$

$$H_r(\omega) = 2\sum_{n=0}^{(M/2)-1} h(n)\sin\omega\left(\frac{M-1}{2} - n\right) \qquad M \text{ even} \tag{8.2.13}$$

The phase characteristic of the filter for both M odd and M even is

$$\Theta(\omega) = \begin{cases} \dfrac{\pi}{2} - \omega\left(\dfrac{M-1}{2}\right), & \text{if } H_r(\omega) > 0 \\ \dfrac{3\pi}{2} - \omega\left(\dfrac{M-1}{2}\right), & \text{if } H_r(\omega) < 0 \end{cases} \tag{8.2.14}$$

The choice of a symmetric or antisymmetric unit sample response depends on the application. As we shall see later, a symmetric unit sample response is suitable for some applications, while an antisymmetric unit sample response is more suitable for other applications. For example, if $h(n) = -h(M-1-n)$ and M is odd, (8.2.12) implies that $H_r(0) = 0$ and $H_r(\pi) = 0$. Consequently, (8.2.12) is not suitable as either a lowpass filter or a highpass filter. Similarly, the antisymmetric unit sample response with M even also results in $H_r(0) = 0$, as can be easily verified from (8.2.13). Consequently, we would not use the antisymmetric condition in the design of a lowpass linear-phase FIR filter. On the other hand, the symmetry condition $h(n) = h(M-1-n)$ yields a linear-phase FIR filter with a nonzero response at $\omega = 0$, if desired, that is,

$$H_r(0) = h\left(\frac{M-1}{2}\right) + 2\sum_{n=0}^{(M-3)/2} h(n), \qquad M \text{ odd} \tag{8.2.15}$$

$$H_r(0) = 2\sum_{n=0}^{(M/2)-1} h(n), \qquad M \text{ even} \tag{8.2.16}$$

8.2.2 Design of Linear-Phase FIR Filters Using Windows

In this method we begin with the desired frequency response specification $H_d(\omega)$ and determine the corresponding unit sample response $h_d(n)$. Indeed, $h_d(n)$ is related to $H_d(\omega)$ by the Fourier transform relation

$$H_d(\omega) = \sum_{n=0}^{\infty} h_d(n) e^{-j\omega n} \tag{8.2.17}$$

where

$$h_d(n) = \frac{1}{2\pi}\int_{-\pi}^{\pi} H_d(\omega) e^{j\omega n} d\omega \tag{8.2.18}$$

Thus, given $H_d(\omega)$, we can determine the unit sample response $h_d(n)$ by evaluating the integral in (8.2.18).

In general, the unit sample response $h_d(n)$ obtained from (8.2.17) is infinite in duration and must be truncated at some point, say at $n = M - 1$, to yield an FIR filter of length M. Truncation of $h_d(n)$ to a length $M - 1$ is equivalent to multiplying $h_d(n)$ by a "rectangular window," defined as

$$w(n) = \begin{cases} 1, & n = 0, 1, \ldots, M-1 \\ 0, & \text{otherwise} \end{cases} \tag{8.2.19}$$

Thus the unit sample response of the FIR filter becomes

$$\begin{aligned} h(n) &= h_d(n)w(n) \\ &= \begin{cases} h_d(n), & n = 0, 1, \ldots, M-1 \\ 0, & \text{otherwise} \end{cases} \end{aligned} \tag{8.2.20}$$

It is instructive to consider the effect of the window function on the desired frequency response $H_d(\omega)$. Recall that multiplication of the window function $w(n)$ with $h_d(n)$ is equivalent to convolution of $H_d(\omega)$ with $W(\omega)$, where $W(\omega)$ is the frequency-domain representation (Fourier transform) of the window function, that is,

$$W(\omega) = \sum_{n=0}^{M-1} w(n)e^{-j\omega n} \tag{8.2.21}$$

Thus the convolution of $H_d(\omega)$ with $W(\omega)$ yields the frequency response of the (truncated) FIR filter. That is,

$$H(\omega) = \frac{1}{2\pi} \int_{-\pi}^{\pi} H_d(\nu)W(\omega - \nu)d\nu \tag{8.2.22}$$

The Fourier transform of the rectangular window is

$$\begin{aligned} W(\omega) &= \sum_{n=0}^{M-1} e^{-j\omega n} \\ &= \frac{1 - e^{-j\omega M}}{1 - e^{-j\omega}} = e^{-j\omega(M-1)/2} \frac{\sin(\omega M/2)}{\sin(\omega/2)} \end{aligned} \tag{8.2.23}$$

This window function has a magnitude response

$$|W(\omega)| = \frac{|\sin(\omega M/2)|}{|\sin(\omega/2)|} \qquad -\pi \le \omega \le \pi \tag{8.2.24}$$

and a piecewise linear phase

$$\Theta(\omega) = \begin{cases} -\omega\left(\dfrac{M-1}{2}\right), & \text{when } \sin(\omega M/2) \ge 0 \\ -\omega\left(\dfrac{M-1}{2}\right) + \pi, & \text{when } \sin(\omega M/2) < 0 \end{cases} \tag{8.2.25}$$

The magnitude response of the window function is illustrated in Fig. 8.4 for $M = 31$ and 61. The width of the main lobe [width is measured to the first zero of $W(\omega)$]

Name of window	Time-domain sequence, $h(n), 0 \le n \le M-1$
Bartlett (triangular)	$1 - \dfrac{2\left\lvert n - \dfrac{M-1}{2}\right\rvert}{M-1}$
Blackman	$0.42 - 0.5\cos\dfrac{2\pi n}{M-1} + 0.08\cos\dfrac{4\pi n}{M-1}$
Hamming	$0.54 - 0.46\cos\dfrac{2\pi n}{M-1}$
Hanning	$\dfrac{1}{2}\left(1 - \cos\dfrac{2\pi n}{M-1}\right)$
Kaiser	$\dfrac{I_0\left[\alpha\sqrt{\left(\dfrac{M-1}{2}\right)^2 - \left(n - \dfrac{M-1}{2}\right)^2}\right]}{I_0\left[\alpha\left(\dfrac{M-1}{2}\right)\right]}$
Lanczos	$\left\{\dfrac{\sin\left[2\pi\left(n - \dfrac{M-1}{2}\right)\Big/(M-1)\right]}{2\pi\left(n - \dfrac{M-1}{2}\right)\Big/\left(\dfrac{M-1}{2}\right)}\right\}^L \quad L > 0$
	$1, \left\lvert n - \dfrac{M-1}{2}\right\rvert \le \alpha\dfrac{M-1}{2} \quad 0 < \alpha < 1$
Tukey	$\dfrac{1}{2}\left[1 + \cos\left(\dfrac{n - (1+\alpha)(M-1)/2}{(1-\alpha)(M-1)/2}\pi\right)\right]$
	$\alpha(M-1)/2 \le \left\lvert n - \dfrac{M-1}{2}\right\rvert \le \dfrac{M-1}{2}$

8.2.3 Design of Linear-Phase FIR Filters by the Frequency-Sampling Method

In the frequency sampling method for FIR filter design, we specify the desired frequency response $H_d(\omega)$ at a set of equally spaced frequencies, namely

$$\omega_k = \tfrac{2\pi}{M}(k+\alpha) \qquad \begin{aligned} &k = 0, 1, \ldots, \frac{M-1}{2} \quad M \text{ odd} \\ &k = 0, 1, \ldots, \frac{M}{2} - 1 \quad M \text{ even} \\ &\alpha = 0 \quad \text{or} \quad \tfrac{1}{2} \end{aligned} \tag{8.2.30}$$

and solve for the unit sample response $h(n)$ of the FIR filter from these equally

spaced frequency specifications. To reduce sidelobes, it is desirable to optimize the frequency specification in the transition band of the filter. This optimization can be accomplished numerically on a digital computer by means of linear programming techniques as shown by Rabiner et al. (1970).

In this section we exploit a basic symmetry property of the sampled frequency response function to simplify the computations. Let us begin with the desired frequency response of the FIR filter, which is [for simplicity, we drop the subscript in $H_d(\omega)$],

$$H(\omega) = \sum_{n=0}^{M-1} h(n)e^{-j\omega n} \tag{8.2.31}$$

Suppose that we specify the frequency response of the filter at the frequencies given by (8.2.30). Then from (8.2.31) we obtain

$$H(k+\alpha) \equiv H\left(\frac{2\pi}{M}(k+\alpha)\right)$$

$$H(k+\alpha) \equiv \sum_{n=0}^{M-1} h(n)e^{-j2\pi(k+\alpha)n/M} \qquad k = 0, 1, \ldots, M-1 \tag{8.2.32}$$

It is a simple matter to invert (8.2.32) and express $h(n)$ in terms of $H(k+\alpha)$. If we multiply both sides of (8.2.32) by the exponential, $\exp(j2\pi km/M)$, $m = 0, 1, \ldots, M-1$, and sum over $k = 0, 1, \ldots, M-1$, the right-hand side of (8.2.32) reduces to $Mh(m)\exp(-j2\pi\alpha m/M)$. Thus we obtain

$$h(n) = \frac{1}{M}\sum_{k=0}^{M-1} H(k+\alpha)e^{j2\pi(k+\alpha)n/M} \qquad n = 0, 1, \ldots, M-1 \tag{8.2.33}$$

The relationship in (8.2.33) allows us to compute the values of the unit sample response $h(n)$ from the specification of the frequency samples $H(k+\alpha)$, $k = 0, 1, \ldots, M-1$. Note that when $\alpha = 0$, (8.2.32) reduces to the discrete Fourier transform (DFT) of the sequence $\{h(n)\}$ and (8.2.33) reduces to the inverse DFT (IDFT).

Since $\{h(n)\}$ is real, we can easily show that the frequency samples $\{H(k+\alpha)\}$ satisfy the symmetry condition

$$H(k+\alpha) = H^*(M-k-\alpha) \tag{8.2.34}$$

This symmetry condition, along with the symmetry conditions for $\{h(n)\}$, can be used to reduce the frequency specifications from M points to $(M+1)/2$ points for M odd and $M/2$ points for M even. Thus the linear equations for determining $\{h(n)\}$ from $\{H(k+\alpha)\}$ are considerably simplified.

In particular, if (8.2.11) is sampled at the frequencies $\omega_k = 2\pi(k+\alpha)/M$, $k = 0, 1, \ldots, M-1$, we obtain

$$H(k+\alpha) = H_r\left(\frac{2\pi}{M}(k+\alpha)\right)e^{j[\beta\pi/2 - 2\pi(k+\alpha)(M-1)/2M]} \tag{8.2.35}$$

where $\beta = 0$ when $\{h(n)\}$ is symmetric and $\beta = 1$ when $\{h(n)\}$ is antisymmetric. A simplication occurs by defining a set of real frequency samples $\{G(k+m)\}$

$$G(k+\alpha) = (-1)^k H_r\left(\frac{2\pi}{M}(k+\alpha)\right) \qquad k = 0, 1, \ldots, M-1 \tag{8.2.36}$$

We use (8.2.36) in (8.2.35) to eliminate $H_r(\omega_k)$. Thus we obtain

$$H(k+\alpha) = G(k+\alpha)e^{j\pi k}e^{j[\beta\pi/2 - 2\pi(k+\alpha)(M-1)/2M]} \tag{8.2.37}$$

Now the symmetry condition for $H(k+\alpha)$ given in (8.2.34) translates into a corresponding symmetry condition for $G(k+\alpha)$, which can be exploited by substituting into (8.2.33), to simplify the expressions for the FIR filter impulse response $\{h(n)\}$ for the four cases $\alpha = 0$, $\alpha = \frac{1}{2}$, $\beta = 0$, and $\beta = 1$. The results are summarized in Table 8.3. The detailed derivations are left as exercises for the reader.

8.2.4 Design of Optimum Equiripple Linear-Phase FIR Filters

The window method and the frequency-sampling method are relatively simple techniques for designing linear-phase FIR filters. However, they also possess some minor disadvantages, described in Section 8.2.6, which may render them undesirable for some applications. A major problem is the lack of precise control of the critical frequencies such as ω_p and ω_s.

The filter design method described in this section is formulated as a Chebyshev approximation problem. It is viewed as an optimum design criterion in the sense that the weighted approximation error between the desired frequency response and the actual frequency response is spread evenly across the passband and evenly across the stopband of the filter minimizing the maximum error. The resulting filter designs have ripples in both the passband and the stopband.

To describe the design procedure, let us consider the design of a lowpass filter with passband edge frequency ω_p and stopband edge frequency ω_s. From the general specifications given in Fig. 8.2, in the passband, the filter frequency response satisfies the condition

$$1 - \delta_1 \le H_r(\omega) \le 1 + \delta_1 \qquad |\omega| \le \omega_p \tag{8.2.43}$$

Similarly, in the stopband, the filter frequency response is specified to fall between the limits $\pm\delta_2$, that is,

$$-\delta_2 \le H_r(\omega) \le \delta_2 \qquad |\omega| > \omega_s \tag{8.2.44}$$

Thus δ_1 represents the ripple in the passband and δ_2 represents the attenuation or ripple in the stopband. The remaining filter parameter is M, the filter length or the number of filter coefficients.

Case 1: Symmetric unit sample response $h(n) = h(M-1-n)$ and M Odd. In this case, the real-valued frequency response characteristic $H_r(\omega)$ is

$$H_r(\omega) = h\left(\frac{M-1}{2}\right) + 2\sum_{n=0}^{(M-3)/2} h(n)\cos\omega\left(\frac{M-1}{2} - n\right) \tag{8.2.45}$$

If we let $k = (M-1)/2 - n$ and define a new set of filter parameters $\{a(k)\}$ as

$$a(k) = \begin{cases} h\left(\dfrac{M-1}{2}\right), & k = 0 \\ 2h\left(\dfrac{M-1}{2} - k\right), & k = 1, 2, \ldots, \dfrac{M-1}{2} \end{cases} \tag{8.2.46}$$

then (8.2.45) reduces to the compact form

$$H_r(\omega) = \sum_{k=0}^{(M-1)/2} a(k)\cos\omega k \tag{8.2.47}$$

Case 2: Symmetric unit sample response $h(n) = h(M-1-n)$ and M Even. In this case, $H_r(\omega)$ is expressed as

$$H_r(\omega) = 2\sum_{n=0}^{(M/2)-1} h(n)\cos\omega\left(\frac{M-1}{2} - n\right) \tag{8.2.48}$$

Again, we change the summation index from n to $k = M/2 - n$ and define a new set of filter parameters $\{b(k)\}$ as

$$b(k) = 2h\left(\frac{M}{2} - k\right), k = 1, 2, \ldots, M/2 \tag{8.2.49}$$

With these substitutions (8.2.48) becomes

$$H_r(\omega) = \sum_{k=1}^{M/2} b(k)\cos\omega\left(k - \frac{1}{2}\right) \tag{8.2.50}$$

In carrying out the optimization, it is convenient to rearrange (8.2.50) further into the form

$$H_r(\omega) = \cos\frac{\omega}{2}\sum_{k=0}^{(M/2)-1} \tilde{b}(k)\cos\omega k \tag{8.2.51}$$

where the coefficients $\{\tilde{b}(k)\}$ are linearly related to the coefficients $\{b(k)\}$. In fact, it can be shown that the relationship is

$$\begin{aligned} \tilde{b}(0) &= \tfrac{1}{2}b(1) \\ \tilde{b}(k) &= 2b(k) - \tilde{b}(k-1) \qquad k = 1, 2, 3, \ldots, \frac{M}{2} - 2 \\ \tilde{b}\left(\frac{M}{2} - 1\right) &= 2b\left(\frac{M}{2}\right) \end{aligned} \tag{8.2.52}$$

Case 3: Antisymmetric unit sample response $h(n) = -h(M-1-n)$ and M Odd. The real-valued frequency response characteristic $H_r(\omega)$ for this case is

$$H_r(\omega) = 2 \sum_{n=0}^{(M-3)/2} h(n) \sin \omega \left(\frac{M-1}{2} - n \right) \tag{8.2.53}$$

If we change the summation in (8.2.53) from n to $k = (M-1)/2 - n$ and define a new set of filter parameters $\{c(k)\}$ as

$$c(k) = 2h\left(\frac{M-1}{2} - k \right) \qquad k = 1, 2, \ldots, (M-1)/2 \tag{8.2.54}$$

then (8.2.53) becomes

$$H_r(\omega) = \sum_{k=1}^{(M-1)/2} c(k) \sin \omega k \tag{8.2.55}$$

As in the previous case, it is convenient to rearrange (8.2.55) into the form

$$H_r(\omega) = \sin \omega \sum_{k=0}^{(M-3)/2} \tilde{c}(k) \cos \omega k \tag{8.2.56}$$

Case 4: Antisymmetric unit sample response $h(n) = -h(M-1-n)$ and M Even. In this case, the real-valued frequency response characteristic $H_r(\omega)$ is

$$H_r(\omega) = 2 \sum_{n=0}^{(M/2)-1} h(n) \sin \omega \left(\frac{M-1}{2} - n \right) \tag{8.2.58}$$

A change in the summation index from n to $k = M/2 - n$ combined with a definition of a new set of filter coefficients $\{d(k)\}$, related to $\{h(n)\}$ according to

$$d(k) = 2h\left(\frac{M}{2} - k \right) \qquad k = 1, 2, \ldots, \frac{M}{2} \tag{8.2.59}$$

results in the expression

$$H_r(\omega) = \sum_{k=1}^{M/2} d(k) \sin \omega \left(k - \frac{1}{2} \right) \tag{8.2.60}$$

As in the previous two cases, we find it convenient to rearrange (8.2.60) into the form

$$H_r(\omega) = \sin \frac{\omega}{2} \sum_{k=0}^{(M/2)-1} \tilde{d}(k) \cos \omega k \tag{8.2.61}$$

Filter type	$Q(\omega)$	$P(\omega)$
$h(n) = h(M-1-n)$ M odd (case 1)	1	$\sum_{k=0}^{(M-1)/2} a(k)\cos\omega k$
$h(n) = h(M-1-n)$ M even (case 2)	$\cos\frac{\omega}{2}$	$\sum_{k=0}^{(M/2)-1} \tilde{b}(k)\cos\omega k$
$h(n) = -h(M-1-n)$ M odd (case 3)	$\sin\omega$	$\sum_{k=0}^{(M-3)/2} \tilde{c}(k)\cos\omega k$
$h(n) = -h(M-1-n)$ M even (case 4)	$\sin\frac{\omega}{2}$	$\sum_{k=0}^{(M/2)-1} \tilde{d}(k)\cos\omega k$

IIR FILTER DESIGN

DESIGN OF IIR FILTERS FROM ANALOG FILTERS

Just as in the design of FIR filters, there are several methods that can be used to design digital filters having an infinite-duration unit sample response. The techniques described in this section are all based on converting an analog filter into a digital filter. Analog filter design is a mature and well developed field, so it is not surprising that we begin the design of a digital filter in the analog domain and then convert the design into the digital domain.

An analog filter can be described by its system function.

$$H_a(s) = \frac{B(s)}{A(s)} = \frac{\sum_{k=0}^{M} \beta_k s^k}{\sum_{k=0}^{N} \alpha_k s^k} \tag{8.3.1}$$

where $\{\alpha_k\}$ and $\{\beta_k\}$ are the filter coefficients, or by its impulse response, which is related to $H_a(s)$ by the Laplace transform

$$H_a(s) = \int_{-\infty}^{\infty} h(t)e^{-st}\,dt \tag{8.3.2}$$

Alternatively, the analog filter having the rational system function $H(s)$ given in (8.3.1), can be described by the linear constant-coefficient differential equation

$$\sum_{k=0}^{N} \alpha_k \frac{d^k y(t)}{dt^k} = \sum_{k=0}^{M} \beta_k \frac{d^k x(t)}{dt^k} \tag{8.3.3}$$

where $x(t)$ denotes the input signal and $y(t)$ denotes the output of the filter.

Each of these three equivalent characterizations of an analog filter leads to alternative methods for converting the filter into the digital domain, as will be described in Sections 8.3.1 through 8.3.4. We recall that an analog linear time-invariant system with system function $H(s)$ is stable if all its poles lie in the left half of the s-plane. Consequently, if the conversion technique is to be effective, it should possess the following desirable properties:

1. The $j\Omega$ axis in the s-plane should map into the unit circle in the z-plane. Thus there will be a direct relationship between the two frequency variables in the two domains.

UNIT -3
STRUCTURES OF FIR AND IIR SYSTEMS

STRUCTURES FOR THE REALIZATION OF DISCRETE-TIME SYSTEMS

The major factors that influence our choice o f a specific realization are computational complexity, memory requirements, and finite-word-length effects in the computations.

STRUCTURES FOR FIR SYSTEMS

In general, an FIR system is described by the difference equation

$$y(n) = \sum_{k=0}^{M-1} b_k x(n-k) \qquad (7.2.1)$$

or, equivalently, by the system function

$$H(z) = \sum_{k=0}^{M-1} b_k z^{-k} \qquad (7.2.2)$$

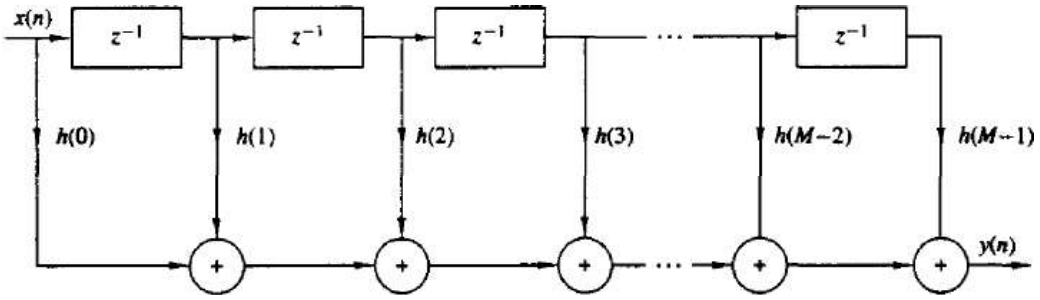
Furthermore, the unit sample response of the FIR system is identical to the coefficients $\{b_k\}$, that is,

$$h(n) = \begin{cases} b_n, & 0 \le n \le M-1 \\ 0, & \text{otherwise} \end{cases}$$

Direct-Form Structure

The direct form realization follows immediately from the non recursive difference equation given below

$$y(n) = \sum_{k=0}^{M-1} h(k)x(n-k)$$



Cascade-Form Structures

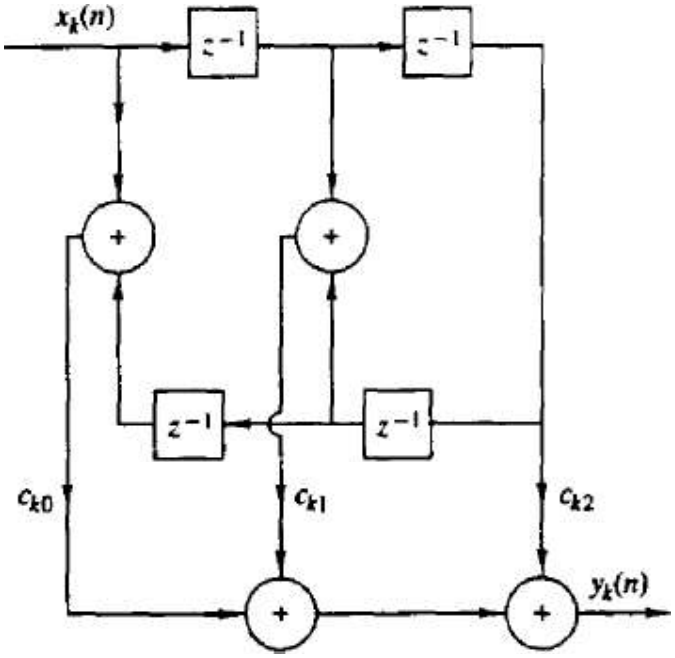
The cascaded realization follows naturally system function given by equation. It is simple matter to factor H(z) into second order FIR system so that

$$H(z) = \prod_{k=1}^{K} H_k(z)$$

where

$$H_k(z) = b_{k0} + b_{k1}z^{-1} + b_{k2}z^{-2} \qquad k = 1, 2, \ldots, K$$

$$H_k(z) = c_{k0}(1 - z_k z^{-1})(1 - z_k^* z^{-1})(1 - z^{-1}/z_k)(1 - z^{-1}/z_k^*)$$
$$= c_{k0} + c_{k1}z^{-1} + c_{k2}z^{-2} + c_{k1}z^{-3} + z^{-4}$$



Frequency-Sampling Structures

The frequency-sampling realization is an alternative structure for an FIR filter in which the parameters that characterize the filter are the values o f the desired frequency response instead of the impulse response *h(n)*. To derive the frequency sampling structure, we specify the desired frequency response at a set o f equally spaced frequencies, namely

$$\omega_k = \frac{2\pi}{M}(k + \alpha) \qquad k = 0, 1, \ldots, \frac{M-1}{2} \quad M \text{ odd}$$
$$k = 0, 1, \ldots, \frac{M}{2} - 1 \quad M \text{ even}$$
$$\alpha = 0 \text{ or } \tfrac{1}{2}$$

The frequency response of the system is given by

$$H(\omega) = \sum_{n=0}^{M-1} h(n)e^{-j\omega n}$$

$$H(k+\alpha) = H\left(\frac{2\pi}{M}(k+\alpha)\right)$$

$$= \sum_{n=0}^{M-1} h(n)e^{-j2\pi(k+\alpha)n/M} \qquad k = 0, 1, \ldots, M-1$$

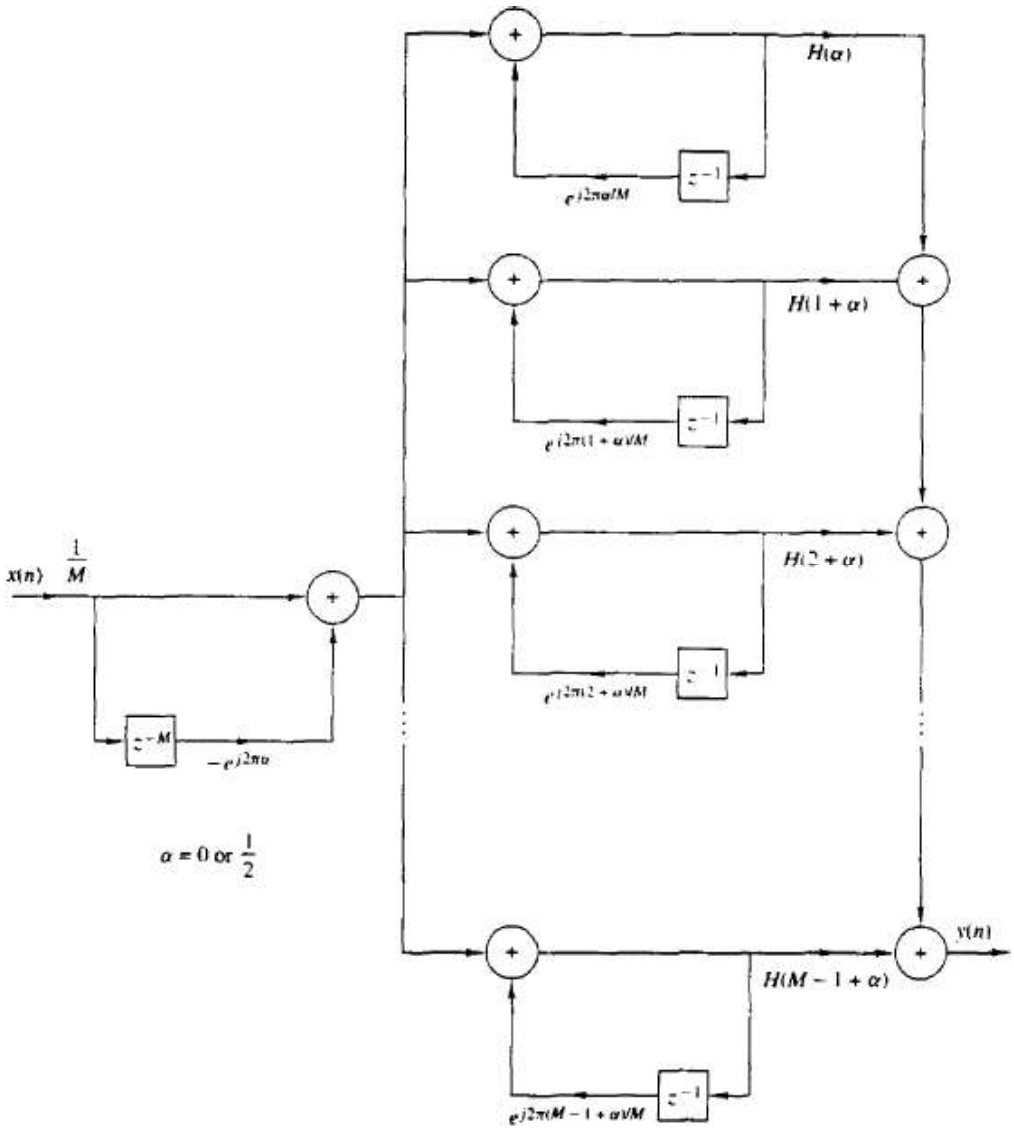
$$h(n) = \frac{1}{M}\sum_{k=0}^{M-1} H(k+\alpha)e^{j2\pi(k+\alpha)n/M} \qquad n = 0, 1, \ldots, M-1$$

$$H(z) = \sum_{n=0}^{M-1} h(n)z^{-n}$$

$$= \sum_{n=0}^{M-1}\left[\frac{1}{M}\sum_{k=0}^{M-1} H(k+\alpha)e^{j2\pi(k+\alpha)n/M}\right]z^{-n}$$

$$H(z) = \sum_{k=0}^{M-1} H(k+\alpha)\left[\frac{1}{M}\sum_{n=0}^{M-1}(e^{j2\pi(k+\alpha)/M}z^{-1})^n\right]$$

$$= \frac{1-z^{-M}e^{j2\pi\alpha}}{M}\sum_{k=0}^{M-1}\frac{H(k+\alpha)}{1-e^{j2\pi(k+\alpha)/M}z^{-1}}$$



Lattice Structure

In this section w e introduce another F IR filter structure, called the lattice filter or Lattice realization. Lattice filters are used extensively in digital speech processing And in the implementation of adaptive filters. Let us begin the development by considering a sequence of FIR filters with system functions

$$H_m(z) = A_m(z) \qquad m = 0, 1, 2, \ldots, M-1 \tag{7.2.17}$$

where, by definition, $A_m(z)$ is the polynomial

$$A_m(z) = 1 + \sum_{k=1}^{m} \alpha_m(k) z^{-k} \qquad m \geq 1 \tag{7.2.18}$$

and $A_0(z) = 1$. The unit sample response of the mth filter is $h_m(0) = 1$ and $h_m(k) = \alpha_m(k)$, $k = 1, 2, \ldots, m$. The subscript m on the polynomial $A_m(z)$ denotes the degree of the polynomial. For mathematical convenience, we define $\alpha_m(0) = 1$.

If $\{x(n)\}$ is the input sequence to the filter $A_m(z)$ and $\{y(n)\}$ is the output sequence, we have

$$y(n) = x(n) + \sum_{k=1}^{m} \alpha_m(k) x(n-k) \tag{7.2.19}$$

Next, let us consider an FIR filter for which $m = 2$. In this case the output from a direct-form structure is

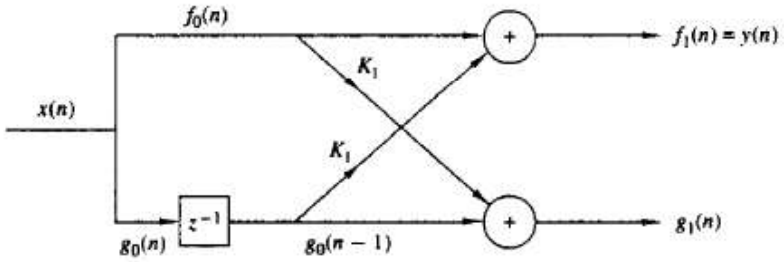
$$y(n) = x(n) + \alpha_2(1)x(n-1) + \alpha_2(2)x(n-2) \tag{7.2.22}$$

By cascading two lattice stages as shown in Fig. 7.10, it is possible to obtain the same output as (7.2.22). Indeed, the output from the first stage is

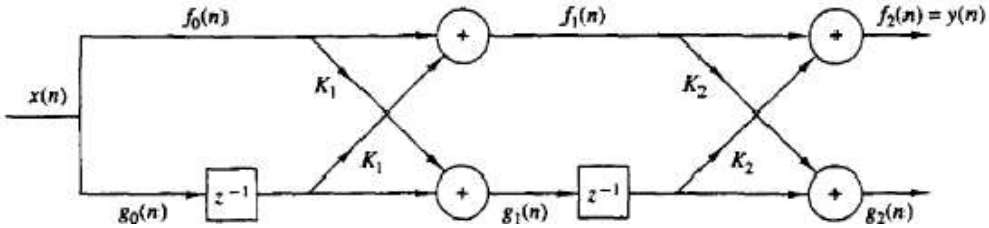
$$\begin{aligned} f_1(n) &= x(n) + K_1 x(n-1) \\ g_1(n) &= K_1 x(n) + x(n-1) \end{aligned} \tag{7.2.23}$$

The output from the second stage is

$$\begin{aligned} f_2(n) &= f_1(n) + K_2 g_1(n-1) \\ g_2(n) &= K_2 f_1(n) + g_1(n-1) \end{aligned} \tag{7.2.24}$$



$$f_0(n) = g_0(n) = x(n)$$
$$f_1(n) = f_0(n) + K_1 g_0(n-1) = x(n) + K_1 x(n-1)$$
$$g_1(n) = K_1 f_0(n) + g_0(n-1) = K_1 x(n) + x(n-1)$$



$$\begin{aligned} f_2(n) &= x(n) + K_1 x(n-1) + K_2[K_1 x(n-1) + x(n-2)] \\ &= x(n) + K_1(1 + K_2)x(n-1) + K_2 x(n-2) \end{aligned}$$

The general form of lattice structure for m stage is given by'

$$\begin{aligned} f_0(n) &= g_0(n) = x(n) \\ f_m(n) &= f_{m-1}(n) + K_m g_{m-1}(n-1) \qquad m = 1, 2, \ldots, M-1 \\ g_m(n) &= K_m f_{m-1}(n) + g_{m-1}(n-1) \qquad m = 1, 2, \ldots, M-1 \end{aligned}$$

Conversion of lattice coefficients to direct-form filter coefficients. The direct-form FIR filter coefficients $\{\alpha_m(k)\}$ can be obtained from the lattice coefficients $\{K_i\}$ by using the following relations:

$$A_0(z) = B_0(z) = 1 \tag{7.2.47}$$

$$A_m(z) = A_{m-1}(z) + K_m z^{-1} B_{m-1}(z) \qquad m = 1, 2, \ldots, M-1 \tag{7.2.48}$$

$$B_m(z) = z^{-m} A_m(z^{-1}) \qquad m = 1, 2, \ldots, M-1 \tag{7.2.49}$$

Conversion of direct-form FIR filter coefficients to lattice coefficients. Suppose that we are given the FIR coefficients for the direct-form realization or, equivalently, the polynomial $A_m(z)$, and we wish to determine the corresponding lattice filter parameters $\{K_i\}$. For the m-stage lattice we immediately obtain the parameter $K_m = \alpha_m(m)$. To obtain K_{m-1} we need the polynomials $A_{m-1}(z)$ since, in general, K_m is obtained from the polynomial $A_m(z)$ for $m = M-1, M-2, \ldots, 1$. Consequently, we need to compute the polynomials $A_m(z)$ starting from $m = M-1$ and "stepping down" successively to $m = 1$.

$$K_m = \alpha_m(m) \quad \alpha_{m-1}(0) = 1$$

$$\alpha_{m-1}(k) = \frac{\alpha_m(k) - K_m\beta_m(k)}{1 - K_m^2}$$

$$= \frac{\alpha_m(k) - \alpha_m(m)\alpha_m(m-k)}{1 - \alpha_m^2(m)} \qquad 1 \le k \le m-1$$

STRUCTURES FOR IIR SYSTEMS

In this section we consider different IIR system s structures described by the difference equation given by the system function. Just as in the case o f FIR system s, there are several types o f structures or realizations, including direct-form structures, cascade-form structures, lattice structures, and lattice-ladder structures. In addition, IIR systems lend them selves to a parallel form realization. We begin by describing two direct-form realizations.

DIRECT FORM STRUCTURES:

The rational system function as given by (7.1.2) that characterizes an IIR system can be viewed as two systems in cascade, that is,

$$H(z) = H_1(z)H_2(z) \tag{7.3.1}$$

where $H_1(z)$ consists of the zeros of $H(z)$, and $H_2(z)$ consists of the poles of $H(z)$,

$$H_1(z) = \sum_{k=0}^{M} b_k z^{-k} \tag{7.3.2}$$

and

$$H_2(z) = \frac{1}{1 + \sum_{k=1}^{N} a_k z^{-k}} \tag{7.3.3}$$

UNIT V
ARCHITECHURE OF TMS320C5X

TMS320 Family Overview

The TMS320 family consists of two types of single-chip DSPs:

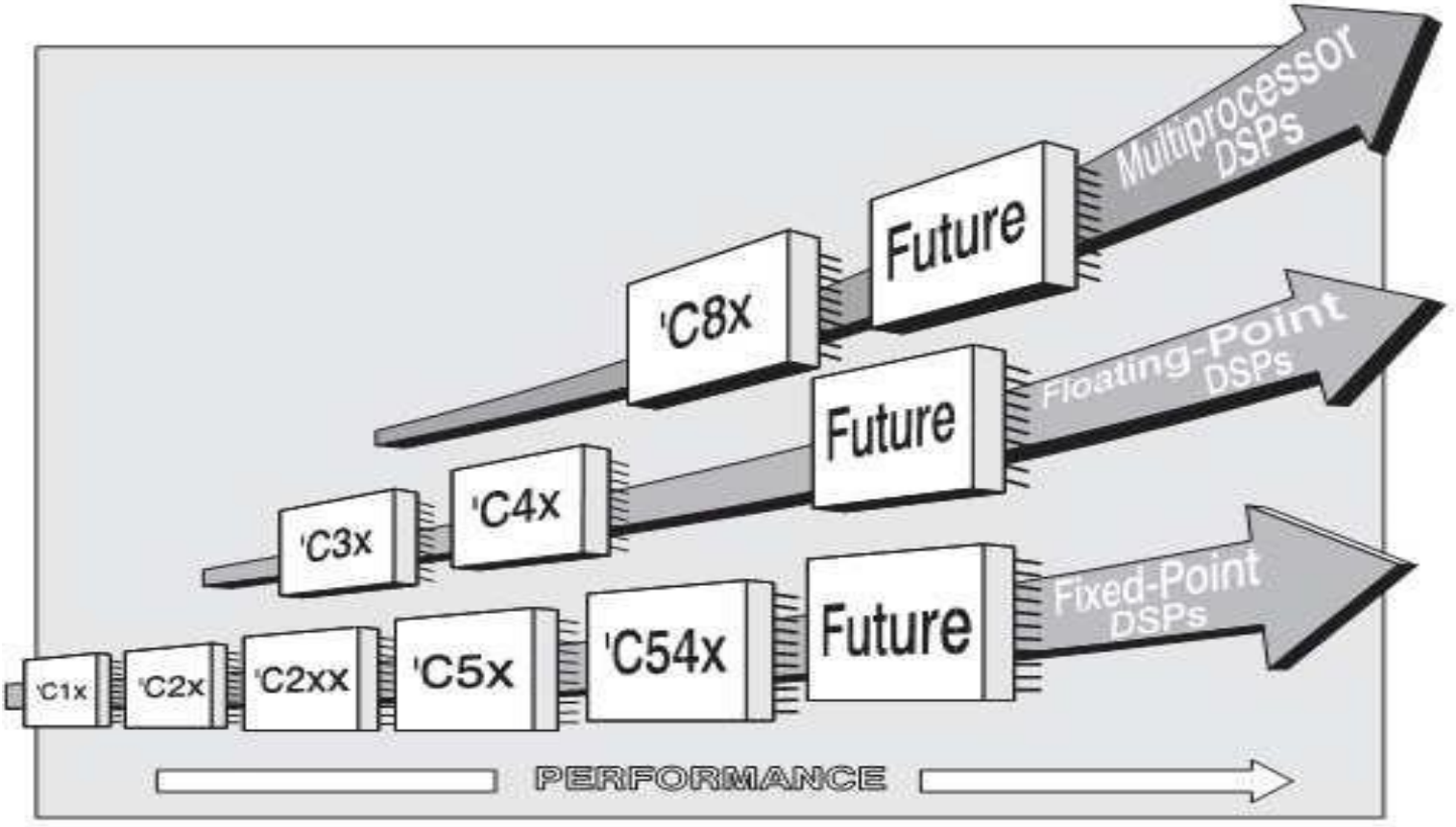
- *16-bit fixed-point* and *32-bit floating-point*. These DSPs possess the **operational flexibility of high-speed controllers** and the **numerical capability of array processors.**

The following characteristics make this family the ideal choice for a wide range of processing applications:

- Very flexible instruction set
- Inherent operational flexibility
- High-speed performance
- Innovative, parallel architectural design
- Cost-effectiveness

Evaluation of the TMS320 family

Figure 1–1. Evolution of the TMS320 Family



TMS320 Overview

- The ’C5x generation consists of the ’C50, ’C51, ’C52, ’C53, ’C53S, ’C56, ’C57, and ’C57S DSPs, which are fabricated by CMOS integrated-circuit technology.
- Their architectural design is based on the ***C25***.
- The ***operational flexibility and speed of the ’C5x*** are the result of combining an advanced ***Harvard architecture*** (which has separate buses for program memory and data memory),
- A CPU with ***application-specific hardware logic, on-chip peripherals, on-chip memory***, and a ***highly specialized instruction set.***
- The ’C5x is designed to execute up to ***50 million instructions per second (MIPS).***

Advantages of TMS320

The ’C5x devices offer these advantages:

- Enhanced TMS320 architectural design for ***increased performance and versatility.***
- Modular architectural design for fast development of ***spin-off devices.***
- Advanced integrated-circuit processing technology for increased performance and ***low power consumption.***
- ***Source code compatibility*** with ’C1x, ’C2x, and ’C2xx DSPs for fast and easy performance upgrades.
- Enhanced ***instruction set for faster algorithms*** and for optimized high-level language operation.
- ***Reduced power consumption*** and increased radiation hardness because of *new static design techniques.*

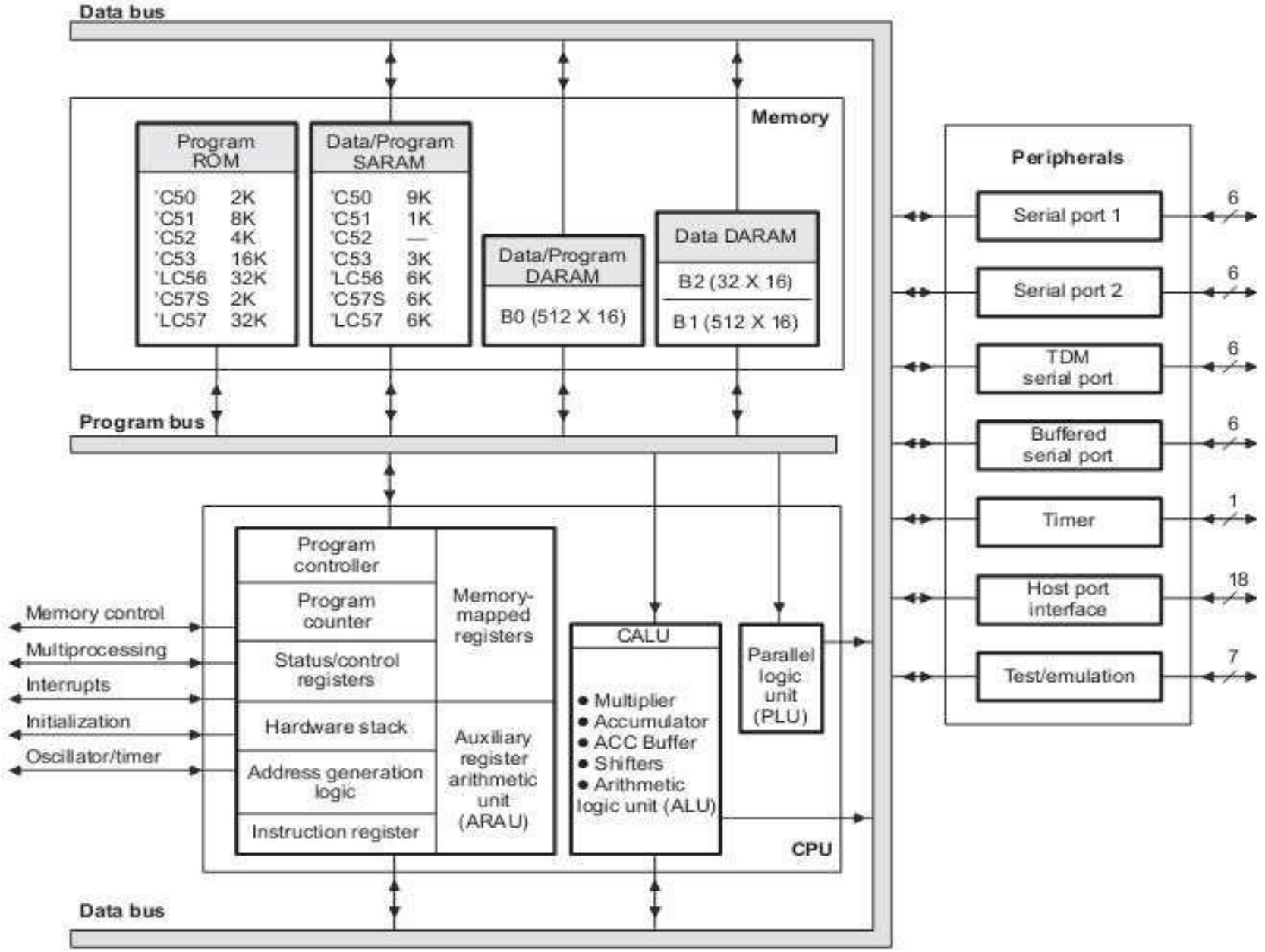
TMS320C5x Key Features

- Compatibility: Source-code compatible with ’C1x, ’C2x, and ’C2xx devices
- Speed: 20-/25-/35-/50-ns single-cycle fixed-point instruction execution time (50/40/28.6/20 MIPS)
- Power
  - 3.3-V and 5-V static CMOS technology with two power-down modes
  - Power consumption control with IDLE1 and IDLE2 instructions for power-down modes

Architectural Overview

- This chapter provides an overview of the architectural structure of the ’C5x,
- which consists of the buses, on-chip memory, central processing unit (CPU), and on-chip peripherals.
- The ’C5x uses an advanced, modified Harvard-type architecture based on the ’C25 architecture and maximizes processing power with separate buses for program memory and data memory.
- The instruction set supports data transfers between the two memory spaces.

Figure 2–1. 'C5x Functional Block Diagram



Bus Structure

- The ’C5x architecture is built around four major buses:
  - Program bus (PB)
  - Program address bus (PAB)
  - Data read bus (DB)
  - Data read address bus (DAB)
    - The ***PAB provides addresses to program memory space*** for both reads and writes.
    - The ***PB*** also ***carries the instruction code and immediate operands*** from program memory space to the CPU.
    - The ***DB interconnects various elements of the CPU*** to data memory space.
    - The ***program and data buses*** can work together to transfer data from ***on-chip data memory*** and ***internal or external program memory*** to the ***multiplier for single-cycle multiply/accumulate operations.***

Central Processing Unit (CPU)

- The ’C5x CPU consists of these elements:
  - Central arithmetic logic unit (CALU)
  - Parallel logic unit (PLU)
  - Auxiliary register arithmetic unit (ARAU)
  - Memory-mapped registers
  - Program controller

Central Arithmetic Logic Unit (CALU)

The CPU uses the CALU to perform 2s-complement arithmetic. The CALU consists of these elements:

- 16-bit x 16 -bit multiplier
- 32-bit arithmetic logic unit (ALU)
- 32-bit accumulator (ACC)
- 32-bit accumulator buffer (ACCB)
- Additional shifters at the outputs of both the accumulator and the product register (PREG)

Parallel Logic Unit (PLU)

- The CPU includes an independent PLU, which operates separately from, but in parallel with, the ALU.
- The PLU performs Boolean operations or the bit manipulations required of high-speed controllers.
- The PLU can *set, clear, test, or toggle* bits in a status register, control register, or any data memory location.
- The PLU provides a *direct logic operation* path to data memory values without affecting the contents of the ACC or PREG.

Auxiliary Register Arithmetic Unit (ARAU)

- The CPU includes an ***unsigned 16-bit arithmetic logic unit*** that calculates ***indirect addresses by using inputs from the auxiliary registers (ARs), index register (INDX), and auxiliary register compare register (ARCR).***
- The ***ARAU can auto index the current AR*** while the data memory location is being addressed and can index either by +/- 1 or by the contents of the INDX.
- As a result, ***accessing data does not require the CALU for address manipulation; therefore, the CALU is free for other operations in parallel***

Memory-Mapped Registers

- The memory-mapped registers are used for ***indirect data address pointers, temporary storage, CPU status and control, or integer arithmetic processing through the ARAU.***

- Since the memory-mapped registers are a component of the data memory space, they can be written to and read from in the same way as any other data memory location.

Program Controller

- ***The program controller consists of these elements:***
  - ❑ Program counter
  - ❑ Status and control registers
  - ❑ Hardware stack
  - ❑ Address generation logic
  - ❑ Instruction register

❖ The program controller contains ***logic circuitry that decodes the operational instructions, manages the CPU pipeline, stores the status of CPU operations, and decodes the conditional operations.***

On-Chip Memory

- The ’C5x architecture contains a considerable amount of on-chip memory to aid in system performance and integration:
  - ❑ ***Program read-only memory (ROM)***
  - ❑ ***Data/program dual-access RAM (DARAM)***
  - ❑ ***Data/program single-access RAM (SARAM)***

➢ The ’C5x has a total address range of ***224K words X 16 bits.*** The memory space is divided into four individually selectable memory segments:

Memory space is divided into four individually selectable memory segments:

- ✓ **64K-**word program memory space,
- ✓ **64K**-word local data memory space,
- ✓ **64K-**word input/ output ports,
- ✓ 32K-word global data memory space.

1. **Program ROM :** This memory is used for ***booting program code*** from **slower external ROM or EPROM** to **fast on-chip or external RAM.**
2. **Data/Program Dual-Access RAM :** All ’C5x DSPs carry a 1056-word X 16-bit on-chip dual-access RAM (DARAM).

   The DARAM is divided into three individually selectable memory blocks:

   - 512-word data or program DARAM block B0,
   - 512-word data DARAM block B1,
   - 32-word data DARAM block B2.

The DARAM is ***primarily intended to store data values*** but, when needed, can be ***used to store programs as well.***

DARAM improves the ***operational speed*** of the ’C5x CPU as The CPU operates with a ***4-deep pipeline.***

3. Data/Program Single-Access RAM :

- All ’C5x DSPs except the ’C52 carry a 16-bit on-chip single-access RAM (SARAM) of various sizes
- Code can be ***booted from an off-chip ROM*** and ***then executed at full speed, once it is loaded into the on-chip SARAM.***

The SARAM can be configured by software in one of three ways:

- All SARAM configured as ***data memory***
- All SARAM configured as ***program memory***
- SARAM configured as both ***data memory and program memory***

- **On-Chip Memory Protection :**
  - The ’C5x DSPs have a ***maskable option*** that protects the contents of on-chip memories. When the related bit is set, no externally originating instruction can access the on-chip memory spaces.
- **On-Chip Peripherals :** All ’C5x DSPs have the same CPU structure; however, they have different on-chip peripherals connected to their CPUs. The ’C5x DSP on-chip peripherals available are:
  - **Clock generator**
  - **Hardware timer**
  - **Software-programmable wait-state generators**
  - **Parallel I/O ports**
  - **Host port interface (HPI)**
  - **Serial port**
  - **Buffered serial port (BSP)**
  - **Time-division multiplexed (TDM) serial port**

- **User-maskable interrupts**

Peripherals

1. **Serial Port :** Three different kinds of serial ports are available:
   - a general-purpose serial port,
   - a time-division multiplexed (TDM) serial port,
   - a buffered serial port (BSP).
     - Each ’C5x contains at ***least one general-purpose, high-speed synchronous, full-duplexed serial port interface that provides direct communication with serial devices such as codecs, serial analog-to-digital (A/D) converters, and other serial systems.***
     - The serial port is **capable of operating** at up to **one-fourth the machine cycle rate** (CLKOUT1).
     - The ***serial port transmitter and receiver are double-buffered*** and ***individually controlled by maskable external interrupt signals.*** Data is framed either as **bytes** or as **words.**

2. Buffered Serial Port (BSP):

- The BSP available on the ’C56 and ’C57 devices is a **full-duplexed, double-buffered serial port** and **an auto buffering unit (ABU).**
- The ABU supports ***high-speed data transfer*** and ***reduces interrupt latencies.***

3. TDM Serial Port:

- The TDM serial port available on the ’C50, ’C51, and ’C53 devices is a ***full-duplexed serial port***
- that can be configured by ***software*** either for ***synchronous operations*** or for ***time-division multiplexed*** operations.
- The TDM serial port is commonly used in ***multiprocessor applications.***

4. User-Maskable Interrupts:

- **Four *external interrupt lines (INT1 –INT4)***
- **Five *internal interrupts***,
- A ***timer interrupt*** and
- **Four *serial port interrupts***, are user maskable.

5. Test/Emulation:

- On the ’C50, ’LC50, ’C51, ’LC51, ’C53, ’LC53, ’C57S and ’LC57S, ***an IEEE standard 1149.1*** (JTAG) interface with ***boundary scan capability*** is ***used for emulation and test***

6. Clock Generator: The clock generator consists of an ***internal oscillator and a phase-locked loop (PLL)*** circuit. The clock generator can be ***driven internally by a crystal resonator circuit*** or ***driven externally by a clock source***

7. Hardware Timer: A 16-bit hardware timer with a 4-bit pre-scaler is available. The timer can be ***stopped, restarted, reset, or disabled*** by **specific status bits.**

8. Software-Programmable Wait-State Generators: Software-programmable wait-state logic is incorporated in ’C5x DSPs allowing wait-state generation without any external hardware for interfacing with slower off-chip memory and I/O devices.

9. Parallel I/O Ports: A total of ***64K I/O ports are available, sixteen*** of these ports are memory-mapped in data memory space. Each of the I/O ports can be addressed by the ***IN or the OUT instruction.***

10. Host Port Interface (HPI): The HPI available on the ***’C57S and ’LC57 is an 8-bit parallel I/O port*** that pro-vides an interface to a ***host processor.***

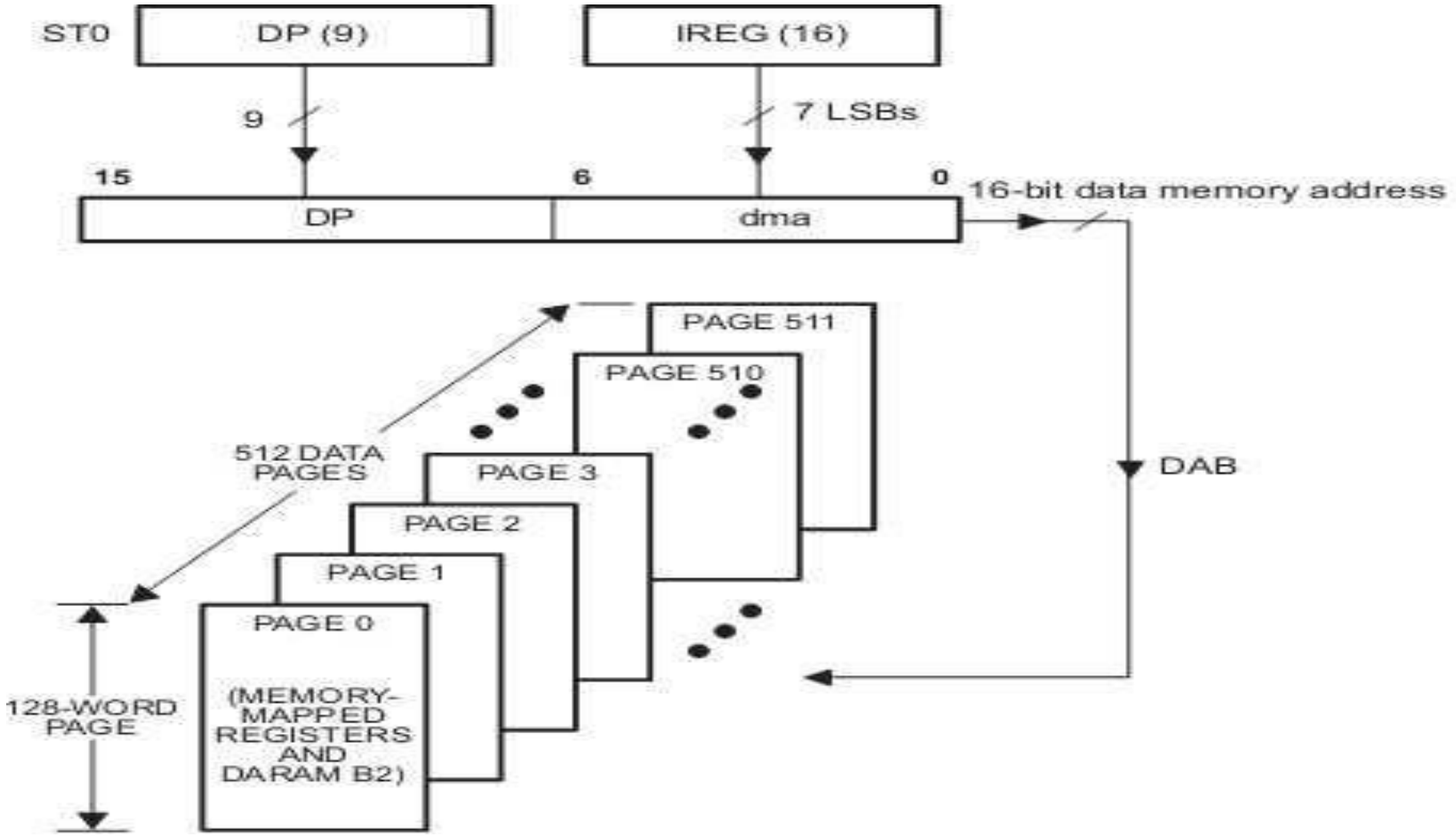
Addressing Modes

- Direct addressing
- Indirect addressing
- Immediate addressing
- Dedicated-register addressing
- Memory-mapped register addressing
- Circular addressing

1. Direct Addressing

- In the direct memory addressing mode, the instruction contains the ***lower 7 bits of the data memory address (dma).***
- The 7-bit dma is concatenated with ***the 9 bits of the data memory page pointer (DP) in status register 0*** to form the full 16-bit data memory address.
- This ***16-bit data memory address*** is placed on an internal direct data memory address bus ***(DAB).***
- The DP points to one of 512 possible data memory pages and the 7-bit address in the instruction points to one of 128 words within that data memory page.
- You can load the DP bits by using the LDP or the LST #0 instruction.

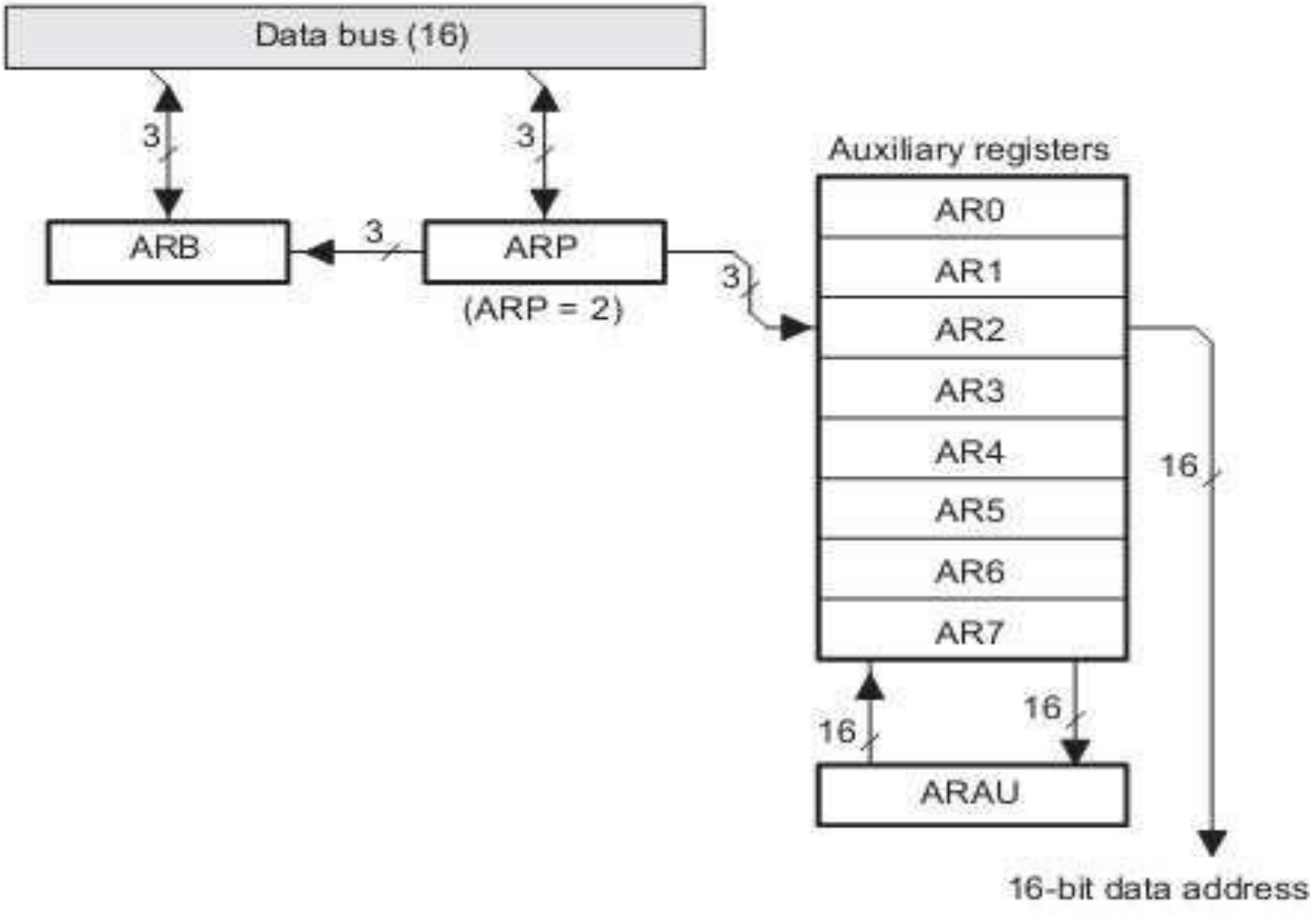
Direct Addressing…



2.Indirect Addressing

- ***Eight 16-bit auxiliary registers (AR0–AR7)*** provide flexible and powerful indirect addressing.
- In indirect addressing, any location in the ***64K-word data memory*** space can be accessed using a ***16-bit address*** contained in an AR.
- Figure shows the hardware for indirect addressing.

Indirect Addressing….



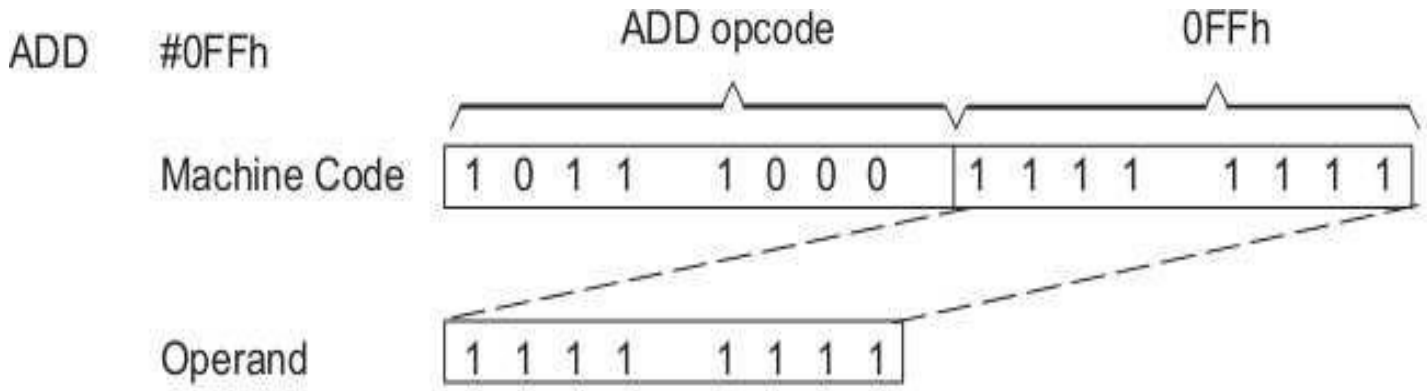
- To select a specific AR, load the auxiliary register pointer (ARP) with ***a value from 0 through 7,*** designating AR0 through AR7, respectively.
- The register pointed to by the ARP is referred to as the current auxiliary register (current AR).
- You can load the address into the AR using the LAR instruction

3.Immediate Addressing

- In immediate addressing, the instruction word(s) contains the value of the ***immediate operand.*** The ’C5x has both 1-word (8-bit, 9-bit, and 13-bit constant) short immediate instructions
- 2-word (16-bit constant) long immediate instructions.
- This mode is indicated by symbol *# .for e.g*
- *ADD # 56h : adds 56h to ACC.*
- *ADD # 4567h : adds 4567h to ACC*

3.1. Short Immediate addressing

- In short immediate instructions, the operand is contained within the instruction machine code. Figure shows an example of the short immediate mode.
- Note that in this example, the lower 8 bits are the operand and will be added to the ACC by the CALU.



3.2. Long Immediate Addressing

- In long immediate instructions, the operand is contained in ***the second word of a two-word instruction.*** There are two long immediate addressing modes:

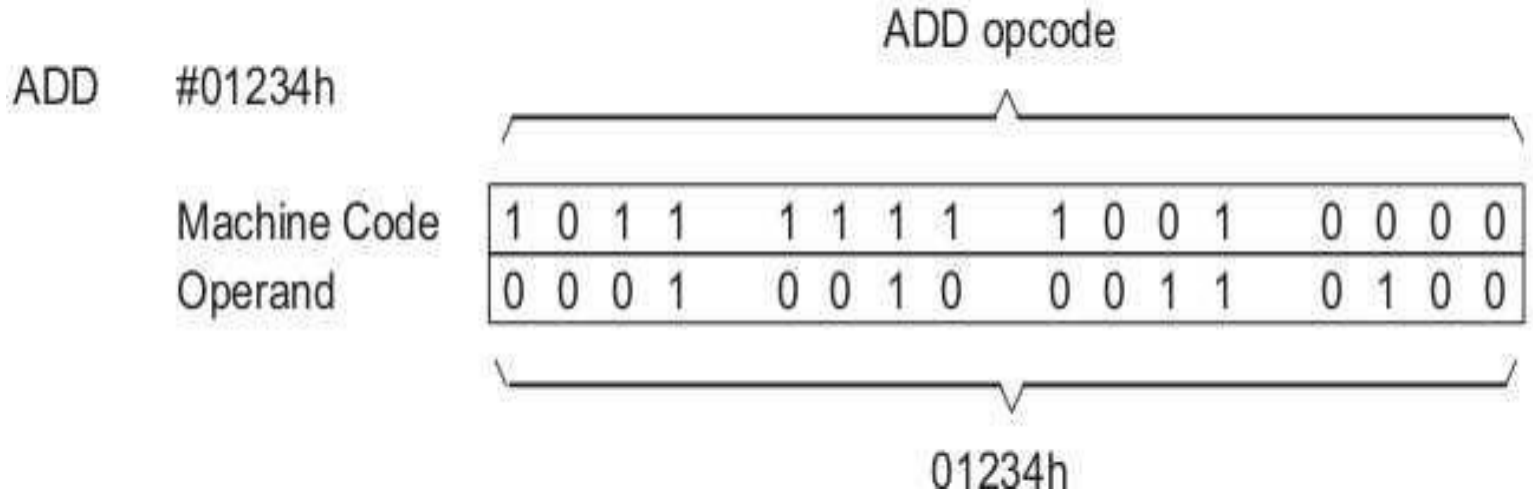
➢ One-operand instructions

➢ Two-operand instructions

Long Immediate Addressing with Single/No Data memory Access :

Figure shows an example of long immediate addressing with no data memory access. In Figure the second word of the 2-word instruction is added to the ACC by the CALU**.**

Long Immediate Addressing Mode — No Data Memory Access



4. Dedicated-Register Addressing

- The dedicated-registered addressing mode operates like the long immediate addressing mode,
- Where address comes from one of two special-purpose memory-mapped registers in the CPU:

➢ The block move address register (BMAR)

➢ The dynamic bit manipulation register (DBMR).

The advantage of this addressing mode is that the address of the block of memory to be acted upon can be changed during execution of the program.

- The syntax for dedicated-register addressing can be stated in one of two ways:

- Specify BMAR by its predefined symbol:

```
BLDD BMAR,DAT100 ;DP = 0.  BMAR contains the value 200h.
```

The content of data memory location 200h is copied to data memory location 100 on the current data page.

- Exclude the immediate value from a parallel logic unit (PLU) instruction:

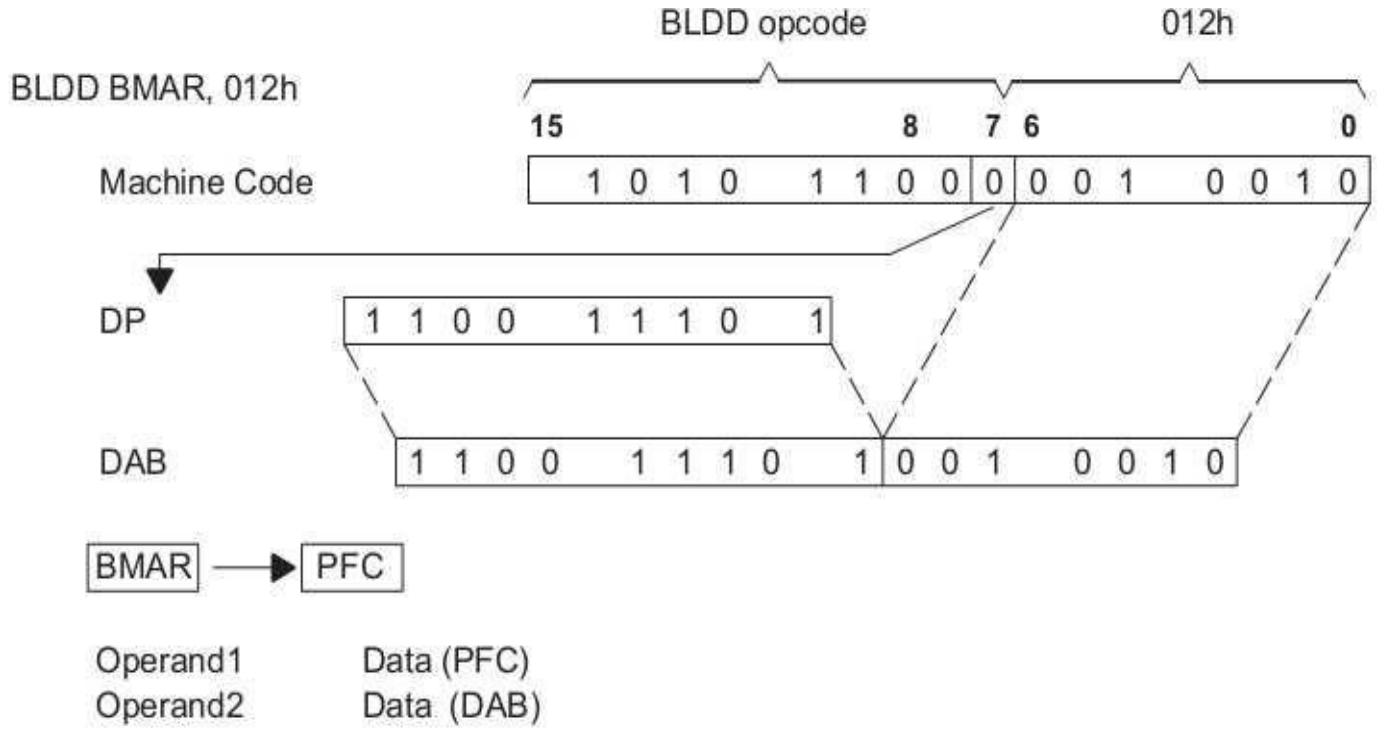
```
OPL DAT10 ;DP = 6. DBMR contains the value FFF0h. ;Address 030Ah contains the value 01h
```

The content of data memory location 030Ah is ORed with the content of the DBMR. The resulting value FFF1h is stored back in memory location
030Ah.

4.1. Using the Contents of the BMAR

- The BLDD, BLDP, and BLPD instructions use the BMAR to point at the source or destination space of a block move.
- The MADD and MADS instructions also use the BMAR to address an operand in program memory for a multiply-accumulate operation.

Figure shows how the BMAR is used in the dedicated-register addressing mode. **Bits 15 through 8** of the machine code contain the opcode. **Bit 7, with a value of 0,** defines the addressing mode as direct, and **bits 6 through 0** contain the dma.

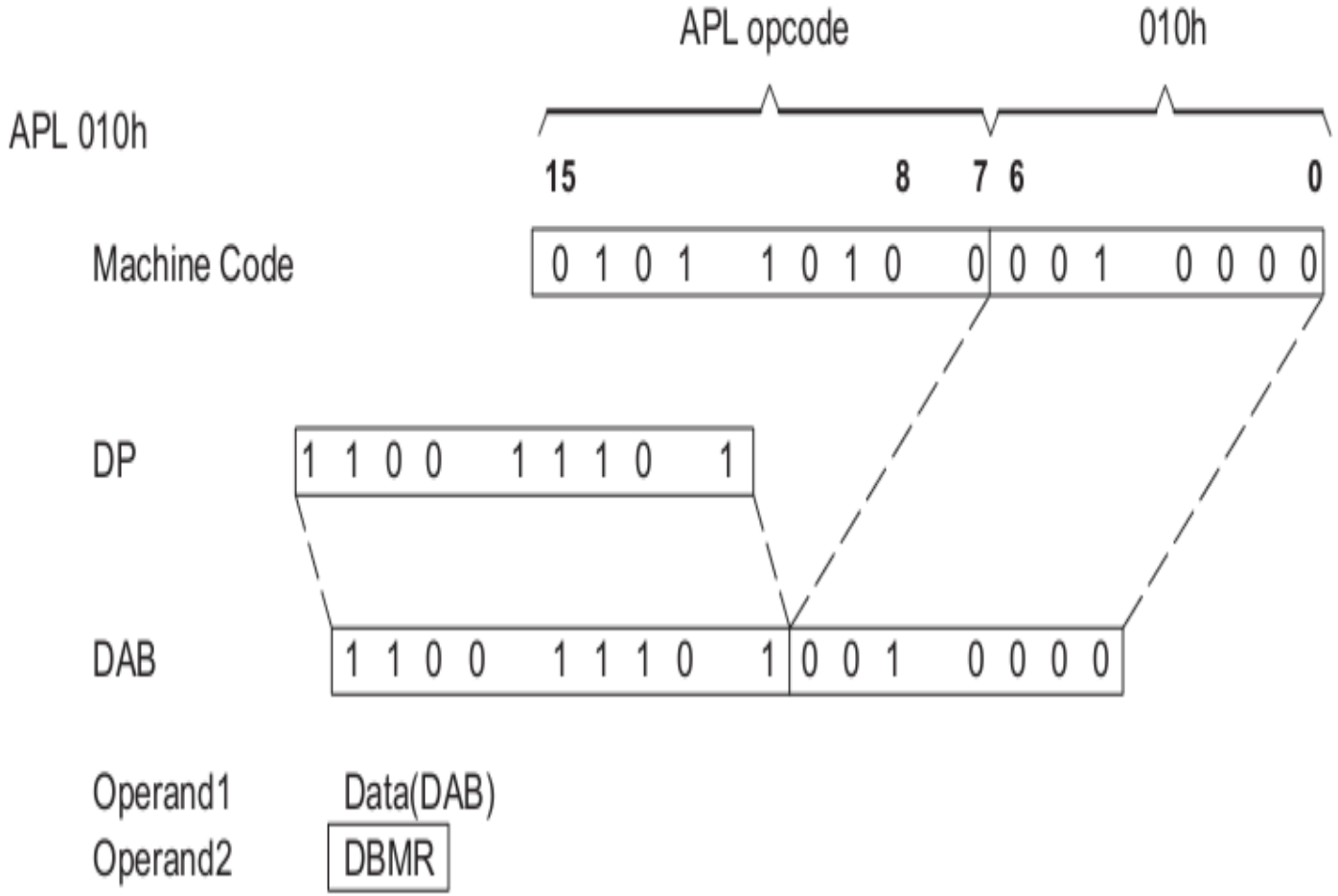


Note: DAB is the 16-bit internal data memory address bus.

4.2. Using the Contents of the DBMR

- The APL, CPL, OPL, and XPL instructions use the PLU and the contents of the DBMR when an immediate value is not specified as one of the operands.
- Figure illustrates how the DBMR is used as an AND mask in the APL instruction.
- **Bits 15 through 8** of the machine code contain the opcode.
- **Bit 7,with a value of 0,** defines the addressing mode as direct, and **bits 6 through 0** contain the dma.

Figure. Dedicated-Register Addressing Using the DBMR



Note: DAB is the 16-bit internal data memory address bus.

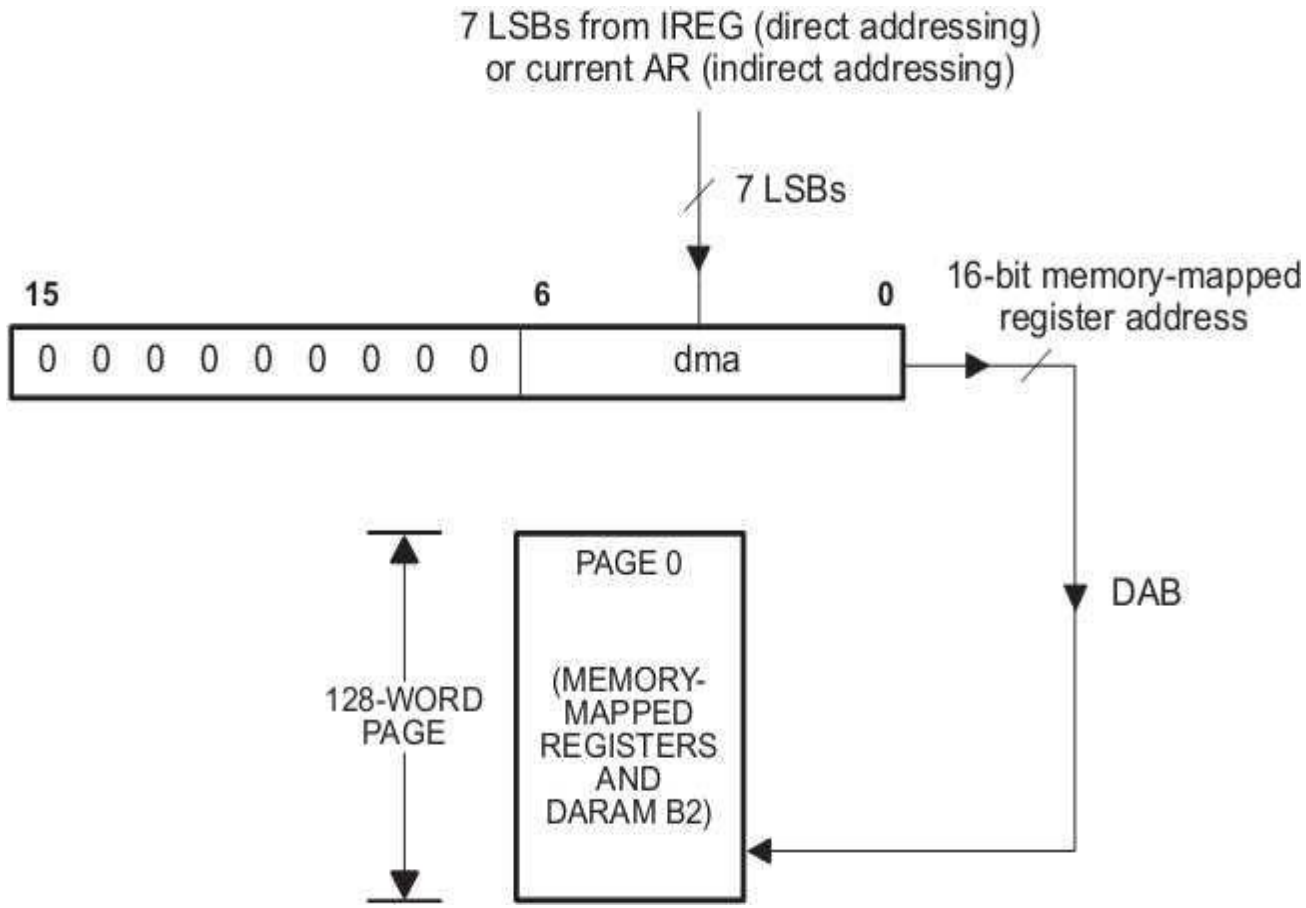
5. Memory-Mapped Register Addressing

- With memory-mapped register addressing, ***you can modify the memory-mapped registers without affecting the current data page pointer value.***
- In addition, ***you can modify any scratch pad RAM (DARAM B2) location or data page 0.***
- The memory-mapped register addressing mode operates like the ***direct addressing mode,*** except that the 9 MSBs of the address are forced to 0 instead of being loaded with the contents of the DP.
- This allows you to ***address the memory-mapped registers of data page 0 directly without the overhead of changing the DP or auxiliary register.***

- The following instructions operate in the memory-mapped register addressing mode. Using these instructions does not affect the contents of the DP:
  - ❑ LAMM — Load accumulator with memory-mapped register
  - ❑ LMMR — Load memory-mapped register
  - ❑ SAMM — Store accumulator in memory-mapped register
  - ❑ SMMR — Store memory-mapped register

Figure illustrates how this is done by forcing the 9 MSBs of the data memory address to 0, regardless of the current value of the DP when direct addressing is used or of the current AR value when indirect addressing is used.

Fig: Memory-Mapped Register Addressing



6. Circular Addressing

- Many algorithms such as ***convolution, correlation, and finite impulse response (FIR) filters*** can use circular buffers in memory to implement a sliding window, which contains the most recent data to be processed.
- The ’C5x supports two concurrent circular buffers operating via the ARs.
- The following five memory-mapped registers control the circular buffer operation:
  - **CBSR1 — Circular buffer 1 start register**
  - **CBSR2 — Circular buffer 2 start register**
  - **CBER1 — Circular buffer 1 end register**
  - **CBER2 — Circular buffer 2 end register**
  - **CBCR — Circular buffer control register**

1. To define circular buffers, you first load the start and end addresses into the corresponding buffer registers;
2. Load a value between the start and end
3. Registers for the circular buffer into an AR.
4. Load the proper AR value, and set the corresponding circular buffer enable bit in the CBCR.