MOTOROLA

Implementing IIR/FIR Filters

with

Motorola's DSP56000/SPS/DSP56001

Digital Signal Processors

Motorola Digital Signal Processors

Implementing IIR/FIR Filters with Motorola's DSP56000/DSP56001

by
John Lane and Garth Hillman
Digital Signal Processing Division

Table of Contents

Table of Contents

Illustrations

Illustrations

Illustrations

Illustrations

Illustrations

Illustrations

SECTION 1

Introduction

"Two classes of frequency-selective digital filters are considered: infinite impulse response (IIR) and finite impulse response (FIR) filters."

This application note considers the design of frequency-selective filters, which modify the frequency content and phase of input signals according to some specification. Two classes of frequency-selective digital filters are considered: infinite impulse response (IIR) and finite impulse response (FIR) filters. The design process consists of determining the coefficients of the IIR or FIR filters, which results in the desired magnitude and phase response being closely approximated.

Therefore, this application note has a two-fold purpose:

1. to provide some intuitive insight into digital filters, particularly how the coefficients are calculated in the digital domain so that a desired frequency response is obtained, and

2. to show how to implement both classes of digital filters (IIR and FIR) on the DSP56001.

It is assumed that most readers are analog designers learning digital signal processing (DSP). The approach used reflects this assumption in that digital filters are initially presented from an analog point of view. Hopefully, this approach will simplify the transition from the analog s-domain transfer functions to the
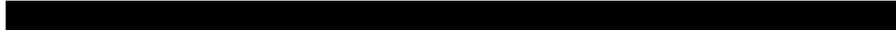
equivalent functions in the digital z-domain. In keeping with this analog perspective, IIR filters will be discussed first since the equivalent of FIR filters are infrequently encountered in the analog world.

SECTION 1 is a brief review of lowpass, highpass, bandpass, and bandstop analog filters. The s-domain formulas governing the key characteristics, magnitude-frequency response, $G(\Omega)$, and phase-frequency response, $\phi(\Omega)$, are derived from first principals. Damping factor, d, cutoff frequency, Ω_c, for lowpass and highpass filters, center frequency, Ω_0, for bandpass and bandstop filters, and quality factor, Q, are defined for the various filter types.

SECTION 2 introduces the bilinear transformation so that analog s-domain designs can be transformed into the digital z-domain and the correct coefficients thereby determined. The form of the formulas for the z-domain filter coefficients thus determined are generalized in terms of the key filter characteristics in the z-domain so that the engineer can design digital filters directly without the necessity of designing the analog equivalent and transforming the design back into the digital domain.
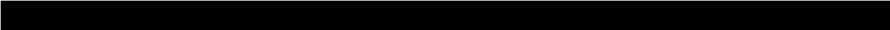
In the analog domain, the performance of the filter depends on the tolerance of the components. Similarly, in the digital domain, the filter performance is limited by the precision of the arithmetic used to implement the filters. In particular, the performance of digital filters is extremely sensitive to overflow, which

occurs when the accumulator width is insufficient to represent all the bits resulting from many consecutive additions. This condition is similar to the condition in the analog world in which the signal output is larger than the amplifier power supply so that saturation occurs. A short analysis of the gain at critical nodes in the filters is given in **SECTION 3** and **SECTION 4** to provide some insight into the scaling requirements for different forms of IIR filters. For this reason, the signal flow graphs developed are centralized about the accumulator nodes.

The analysis of IIR filters focuses on second-order sections. Clearly, higher order filters are often required. Therefore, a brief discussion of how second order sections can be cascaded to yield higher order filters is given in **SECTION 5**. Because the analysis becomes complex quickly, the discussion naturally leads to using commercially available filter design software such as Filter Design and Analysis System (FDAS) from Momentum Data Systems, Inc. **SECTION 6** concludes by showing how the filter coefficients just discussed can be used in DSP56000 code to implement practical digital filters. Examples of complete filter designs are given, including the code, coefficients, frequency response, and maximum sample frequency.

FIR filters are discussed in **SECTION 7**. Initially, FIR filters are contrasted with IIR filters to show that in many ways they are complementary, each satisfying weaknesses of the other. An intuitive approach
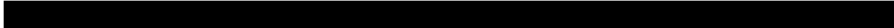
is taken to calculating the filter coefficients by starting from a desired arbitrary frequency response. The importance of and constraint imposed by linear phase is emphasized. Having developed an intuitive appreciation of what FIR filter coefficients are, the use of FDAS to accelerate the design process is described. **SECTION 7** concludes by showing how the filter coefficients just determined can be used in DSP56000 code to implement practical digital filters. An example of a passband digital filter using a Kaiser-window design approach is presented.

1.1 Analog RCL Filter Types

In the following paragraphs, the analog RCL filter network will be analyzed for the four basic filter types: lowpass, highpass, bandpass, and band-stop. Analyzing analog filter types shows that designing digital IIR filters is, in many cases, much simpler than designing analog filters.

In this analysis, as in all of the following cases, the input is assumed to be a steady-state signal containing a linear combination of sinusoidal components whose rms (or peak) amplitudes are constant in time. This assumption allows simple analytic techniques to be used in determining the network response. Even though these results will then be applied to real-world signals that may not satisfy the original steady-state assumption, the de-

viation of the actual response from the predicted response is small enough to neglect in most cases. General analysis techniques consist of a linear combination of steady-state and transient response solutions to the differential equations describing the network.
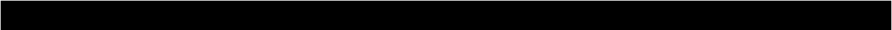
1.2   Analog Lowpass Filter

The passive RCL circuit forming a lowpass filter network is shown in Figure 1-1 where the transfer function, H(s), is derived from a voltage divider analysis of the RCL network. This approach is valid since the effect of C and L can be described as a complex impedance (or reactance, X_c and X_L) under steady state conditions; s is a complex variable of the complex transfer function, H(s). The filter frequency response is found by evaluating H(s) with $s = j\Omega$, where $\Omega = 2\pi f$ and f is the frequency of a sinusoidal component of the input signal. The output signal is calculated from the product of the input signal and $H(j\Omega)$. To facilitate analysis, the input and output signal components are described by the complex value, $e^{j\Omega t} = \cos \Omega t + j \sin \Omega t$. The actual physical input and output signal components are found by taking the real part of this value. The input is $R\{e^{j\Omega t}\} = \cos \Omega t$; the output is $R\{H(j\Omega)e^{j\Omega t}\} = G(\Omega) \cos [\Omega t + \phi(\Omega)]$. The previous technique is based upon the solution of the differential equations describing the network when the input is steady state. Describing the circuit response by H(s) instead of

solving the differential equation is a common simplification used in this type of analysis.

$$\frac{V_0}{V_i} = \frac{X_C \| R}{X_L \| R}$$

$$X_c = 1/j\Omega C$$

$$= \frac{(R/j\Omega C)/(R + 1/j\Omega C)}{j\Omega L + (R/j\Omega C)/(R + 1/j\Omega C)}$$

$$X_L = j\Omega L$$

$$= \frac{1/LC}{-\Omega^2 + j\Omega/RC + 1/LC}$$

$$\Omega_c = \frac{1}{\sqrt{LC}}$$

$$= \frac{\Omega_c^2}{-\Omega^2 + jd\Omega_c\Omega + \Omega_c^2}$$
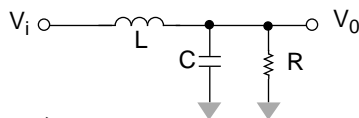
$$d = \sqrt{\frac{L}{R^2 C}}$$

Let $s = j\Omega$ and define $H(s) = V_o/V_i$; then,

$$H(s) = \frac{1}{(s/\Omega_c)^2 + d(s/\Omega_c) + 1}$$

which is the s-domain transfer function. The gain, $G(\Omega)$, of the filter is:

$$G(\Omega) \equiv \sqrt{H(s)H*(s)}\Big|_{s = j\omega}$$

$$= \frac{1}{\sqrt{(1 - \Omega^2/\Omega_c^2)^2 + (d\Omega/\Omega_c)^2}}$$

where $*$ denotes complex conjugate.
The phase angle, $\phi\Omega$, is the angle between the imaginary and real components of $H(s)$.

$$\phi(\Omega) \equiv \tan^{-1}[I\{H(s)\}/R\{H(s)\}]$$

$$= -\tan^{-1}\left[\frac{d(\Omega/\Omega_c)}{1 - (\Omega/\Omega_c)^2}\right] \qquad \text{for } \Omega \leq \Omega_c$$

$$= -\pi - \tan^{-1}\left[\frac{d(\Omega/\Omega_c)}{1 - (\Omega/\Omega_c)^2}\right] \qquad \text{for } \Omega > \Omega_c$$

Figure 1-1 *s-Domain Analysis of Second-Order Lowpass Analog Filter*
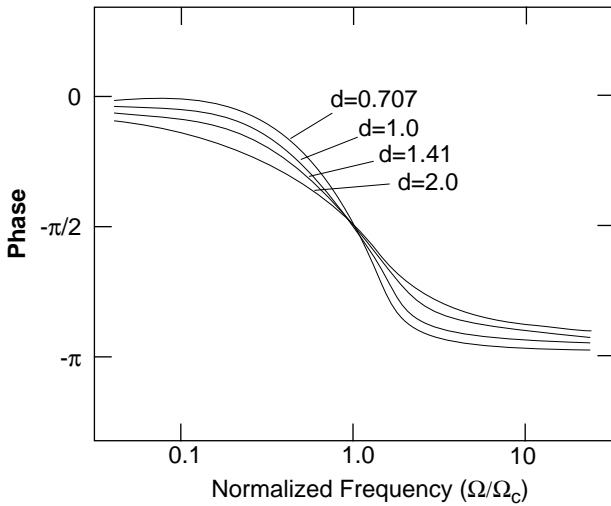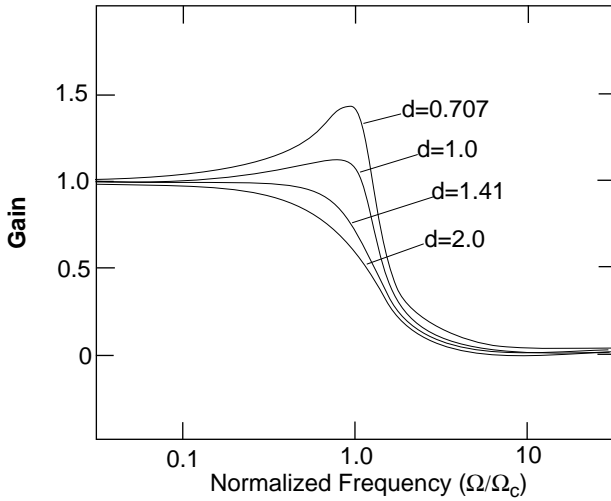
Figure 1-2 *Gain and Phase Response of Second-Order Lowpass Analog Filter at Various Values of Damping Factor, d*

The magnitude of H(s) is defined as the gain, $G(\Omega)$,

of the system; whereas, the ratio of the imaginary part to real part of H(s), $I\{H(j\Omega)\}/R\{H(j\Omega)\}$, is the tangent of the phase, $\phi(\Omega)$, introduced by the filter. If the input signal is $A_k \sin(\Omega_k t + \phi_K)$, then the output signal is $A_K G(\Omega_K) \sin[\Omega_K t + \phi_K + \phi(\Omega_K)]$ Figure 1-2 shows the gain, $G(\Omega)$, and phase, $\phi(\Omega)$, plots for the second-order lowpass network of Figure 1-1 for various values of damping factor, d; d also controls the amplitude and position of the peak of the normalized response curve.

The frequency corresponding to the peak amplitude can be easily found by taking the derivative of $G(\Omega)$ (from the equation for $G(\Omega)$ in Figure 1-1) with respect to Ω and setting it equal to zero. Solving the resultant equation for Ω then defines Ω_M as the frequency where the peak amplitude occurs. The peak amplitude is then $G_M = G(\Omega_M)$:

$$\Omega_M = \Omega_c \sqrt{(1 - d^2/2)} \qquad \text{Eqn. 1-1}$$

$$G_M = \frac{1}{d\sqrt{(1 - d^2/4)}} \qquad \text{Eqn. 1-2}$$

for $d < \sqrt{2}$. For $d > \sqrt{2}$, $\Omega_M = 0$ is the position of the peak amplitude where $G_M = 1$. When $d = \sqrt{2}$, $G_M = 1$, which gives the maximally flat response curve used in the Butterworth filter design (usually applies only to a set of cascaded sections). Note that Ω_C for a lowpass filter is that frequency where the gain is $G(\Omega c) = 1/d$ and the phase is $\phi(\Omega c) = -\pi/2$.

1.3 Analog Highpass Filter

The passive RCL circuit forming a highpass filter network is shown in Figure 1-3 where the transfer function, H(s), is again derived from a voltage divider analysis of the RCL network. The gain and phase response are plotted in Figure 1-4 for different values of damping coefficient. As evidenced, the highpass filter response is the mirror image of the lowpass filter response.

1.4 Analog Bandstop Filter

The analog RCL network for a bandstop filter network is simply the sum of the lowpass and highpass transfer functions shown in Figure 1-5 where the transfer function, H(s), is again derived from a voltage divider analysis of the RCL network. The gain and phase response are plotted in Figure 1-6 for different values of quality factor, Q, (where Q = 1/d). Neglecting the departure of real RCL components' values from the ideal case, the attenuation at the center frequency, f_0, is infinite. Also, note that the phase undergoes a 1 80-degree shift when passing through the center frequency (zero in the s-plane).

Q for bandpass and bandstop filters is a measure of the width, $\Delta\Omega$, of the stopband with respect to the center frequency, Ω_O, i.e., $\Delta\Omega = Q^{-1}\Omega_o$. $\Delta\Omega$ is measured at the points where $G(\Omega) = 1/\sqrt{2}$.

$$\frac{V_0}{V_i} = \frac{X_L \| R}{X_C + X_L \| R}$$

$X_c = 1/j\Omega C$

$V_i \circ\!\!-\!\!\parallel\!\!-\!\!\bullet\!\!-\!\!\bullet\!\!-\!\!\circ V_0$
C L R

$$= \frac{j\Omega L R/(j\Omega L + R)}{1/j\Omega C + j\Omega L R/j\Omega L + R}$$

$X_L = j\Omega L$

$$= \frac{-\Omega^2}{-\Omega^2 + j\Omega/RC + 1/LC}$$

$$\Omega_c = \frac{1}{\sqrt{LC}}$$

$$= \frac{-\Omega^2}{-\Omega^2 + jd\Omega_c\Omega + \Omega_c^2}$$
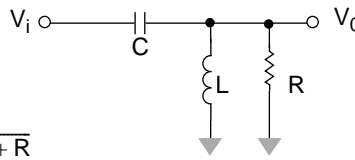
$$d = \sqrt{\frac{L}{R^2 C}}$$

Let $s = j\Omega$ and define $H(s) = V_o/V_i$; then,

$$H(s) = \frac{(s/\Omega_c)^2}{(s/\Omega_c)^2 + d(s/\Omega_c) + 1}$$

which is the s-domain transfer function. The gain, $G(\Omega)$, of the filter is:

$$G(\Omega) \equiv \sqrt{H(s)H*(s)}\Big|_{s = j\omega}$$
$$= \frac{(\Omega/\Omega_c)^2}{\sqrt{\left(1 - \Omega^2/\Omega_c^2\right)^2 + (d\Omega/\Omega_c)^2}}$$

where $*$ denotes complex conjugate.

The phase angle, $\phi\Omega$, is the angle between the imaginary and real components of $H(s)$.

$$\phi(\Omega) \equiv \tan^{-1}[I\{H(s)\}/R\{H(s)\}]$$

$$= \pi - \tan^{-1}\left[\frac{d(\Omega/\Omega_c)}{1 - (\Omega/\Omega_c)^2}\right] \qquad \text{for } \Omega \leq \Omega_c$$

$$= -\tan^{-1}\left[\frac{d(\Omega/\Omega_c)}{1 - (\Omega/\Omega_c)^2}\right] \qquad \text{for } \Omega > \Omega_c$$

Figure 1-3 *s-Domain Analysis of Second-Order Highpass Analog Filter*
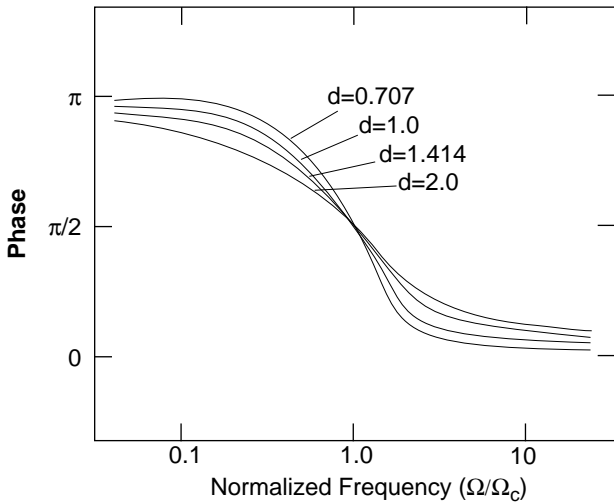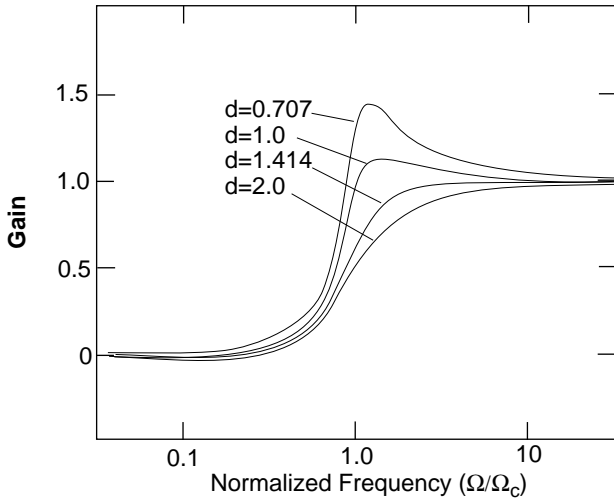
Figure 1-4 *Gain and Phase Response of Second-Order Highpass Analog Filter at Various Values of Damping Factor, d*

1.5 Analog Bandpass Filter

The passive RCL circuit forming a bandpass filter network is shown in Figure 1-7 where the transfer function, H(s), is again derived from a voltage divider analysis of the RCL network. The gain and phase response are plotted in Figure 1-8 for different values of Q. The lowpass gain approaches an asymptotic function of $G = (f_c/f)^2$ for $f/f_O \gg 1$. The highpass asymptotic gain is $G = (f_C/f)^2$ for $f/f_C \ll 1$; whereas, the bandstop case approaches unity at zero and infinity with a true zero at the center frequency. The bandpass gain, on the other hand, approaches $G = Q^{-1}f/f_C$ for $f/f_c \ll 1$ and $G = Q^{-1}f_C/f$ for $f/f_c \gg 1$. The primary differences to note in the bandpass response are:

1. the stopband attenuation is 6 dB/octave or 20 dB/decade (since it goes as $1/f$); whereas, the lowpass and highpass go as $1/f^2$ (12 dB/octave, 40 dB/decade)

2. the stopband attenuation asymptote is dependent on the quality factor; whereas, for the lowpass and highpass cases, the stopband attenuation asymptote is independent of damping factor, d

3. the maximum value of gain is unity regardless of the filter Q.

The specific features characterizing the bandpass, lowpass, highpass, and bandstop analog networks are found to be nearly identical in the digital IIR filter equivalents when the sampling frequency is very high as compared to the frequencies of interest. For this reason, it is important to understand the basic properties of the four filter types before proceeding to the digital domain.

$$\frac{V_0}{V_i} = \frac{R}{x_C \| X_L + R}$$

$$V_i \circlearrowleft \!\!\!\!\!\!\!\!\!\!\!\!\!\!\! \underset{C}{\overset{L}{\frown\!\frown\!\frown}} \!\!\!\!\!\!\!\!\!\!\!\!\!\!\! \circlearrowright V_0 \qquad X_c = 1/j\Omega C$$

$$R$$
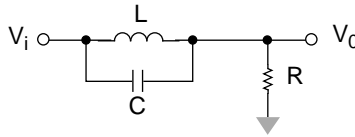
$$X_L = j\Omega L$$

$$= \frac{R}{R + (j\Omega L/j\Omega C)/(j\Omega L + 1/j\Omega C)}$$

$$= \frac{-\Omega^2 + 1/LC}{-\Omega^2 + j\Omega/RC + 1/LC} \qquad \Omega_c = \frac{1}{\sqrt{LC}}$$

$$= \frac{-\Omega^2 + \Omega_0^2}{-\Omega^2 + j\Omega_0\Omega/Q + \Omega_0^2} \qquad Q = 1/d = \sqrt{R^2 C/L}$$

Let $s = j\Omega$ and define $H(s) = V_0/V_i$; then,

$$H(s) = \frac{(s/\Omega_0)^2 + 1}{(s/\Omega_0)^2 + s/\Omega_0 Q + 1}$$

which is the s-domain transfer function. The gain, $G(\Omega)$, of the filter is:

$$G(\Omega) \equiv \left.\sqrt{H(s)H*(s)}\right|_{s = j\omega}$$

$$= \frac{\left|1 - (\Omega/\Omega_0)^2\right|}{\sqrt{(1 - \Omega^2/\Omega_0^2)^2 + \Omega/\Omega_0 Q)^2}}$$

where $*$ denotes complex conjugate.

The phase angle, $\phi\Omega$, is the angle between the imaginary and real components of $H(s)$.

$$\phi(\Omega) \equiv \tan^{-1}[I\{H(s)\}/R\{H(s)\}]$$

$$= -\tan^{-1}\left[\frac{(\Omega/\Omega_0 Q)}{1 - (\Omega/\Omega_0)^2}\right]$$

Figure 1-5 *s-Domain Analysis of Second-Order Bandstop Analog Filter*

Figure 1-6 *Gain and Phase Response of Second-Order Bandstop Analog Filter at Various Values of Damping Factor, d*

$$\frac{V_0}{V_i} = \frac{R}{X_L + X_C + R}$$

$V_i \circ\!\!-\!\!\curvearrowright\!\!-\!\!|\!|\!-\!\!\bullet\!\!-\!\!\circ V_0$
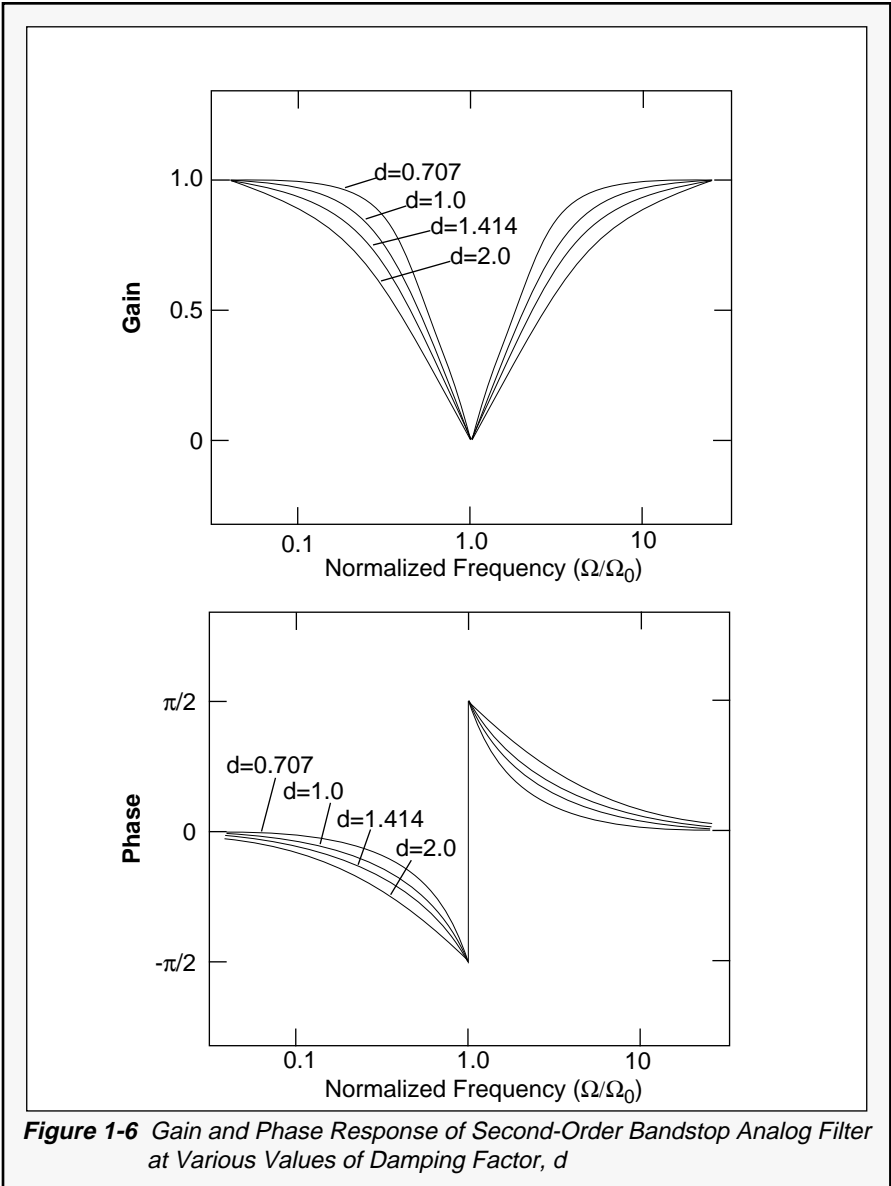
L   C

R
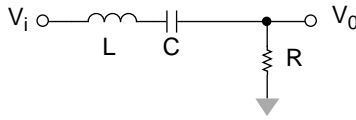
$X_c = 1/j\Omega C$

$X_L = j\Omega L$

$$= \frac{R}{j\Omega L + 1/j\Omega + R}$$

$$= \frac{j\Omega R/L}{-\Omega^2 + j\Omega R/L + 1/LC}$$

$$\Omega_c = \frac{1}{\sqrt{LC}}$$

$$= \frac{j\Omega_0\Omega/Q}{-\Omega^2 + j\Omega_0\Omega/Q + \Omega_0^2}$$

$$Q = 1/d = \sqrt{R^2 C/L}$$

Let $s = j\Omega$ and define $H(s) = V_0/V_i$; then,

$$H(s) = \frac{s/\Omega_0 Q}{(s/\Omega_0)^2 + s/\Omega_0 Q + 1}$$

which is the s-domain transfer function. The gain, $G(\Omega)$, of the filter is:

$$G(\Omega) \equiv \sqrt{H(s)H*(s)}\Big|_{s = j\omega}$$

$$= \frac{\Omega/\Omega_0 Q}{\sqrt{(1 - \Omega^2/\Omega_0^2)^2 + (\Omega/\Omega_0 Q)^2}}$$

where $*$ denotes complex conjugate.

The phase angle, $\phi\Omega$, is the angle between the imaginary and real components of $H(s)$.

$$\phi(\Omega) \equiv \tan^{-1}[I\{H(s)\}/R\{H(s)\}]$$

$$= \tan^{-1}\left[\frac{\Omega/\Omega_0 Q}{1 - (\Omega/\Omega_0)^2}\right]$$

Figure 1-7 *s-Domain Analysis of Second-Order Bandpass Analog Filter*
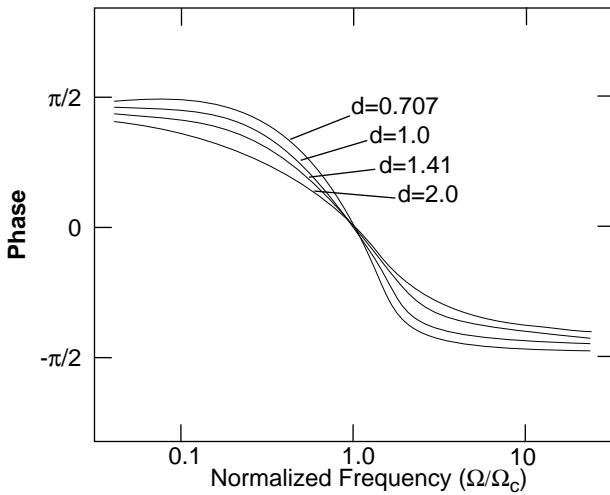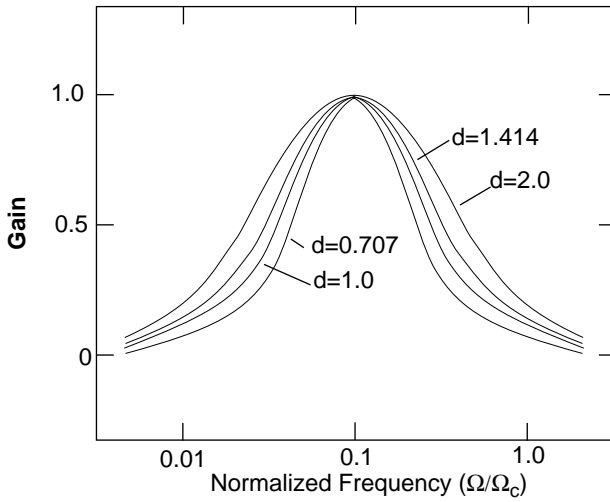
Figure 1-8 *Gain and Phase Response of Second-Order Bandpass Analog Filter at Various Values of Damping Factor, d*

Second-Order Direct-Form IIR Digital Filter Sections

"In the direct-form implementation, the a_i and b_i are used directly in the difference equation, which can be easily programmed on a high-speed DSP such as the DSP56001."

The traditional approach to deriving the digital filter coefficients has been to start with the digital z-domain description, transform to the analog s-domain to understand how to design filters, then transform back to the digital domain to implement the filter. This approach is not used in this report. Instead, formulas are developed relating the s-domain filter to the z-domain filter so transformations to and from one domain to the other are no longer necessary.

The Laplace or s-transform in the analog domain was developed to facilitate the analysis of continuous time signals and systems. For example, using Laplace transforms the concepts of poles and zeros, making system analysis much faster and more systematic. The Laplace transform of a continuous time signal is:

$$X(s) = L\{x(t)\}$$

$$\int_0^\infty x(t)e^{-st}dt \qquad \text{Eqn. 2-3}$$

where: *L = the Laplace transform operator and implies the operation described in Eqn. 2-3*

In the digital domain, the continuous signal, x(t), is first sampled, then quantized by an analog-to-digital (A/D) converter before being processed. That is, the signal is only known at discrete points in time, which are multiples of the sampling interval, $T = 1/f_s$, where f_s is the sampling frequency. Because of the sampled characteristic of a digital signal, its z-transform is given by a summation (as opposed to an integral):

$$X(z) = Z\{x(n)\}$$

$$= \sum_{n = -\infty}^{\infty} x(n)z^{-n} \qquad \text{Eqn. 2-4}$$

where: *Z*   *is the z-transform operator as described by the operation of Eqn. 2-4*

       x(n)   *is the quantized values from the A/D converter of the continuous time signal, x(t), at discrete times, t = nT*

One property of the z-transform which will be used later in this report is the time shifting property. The time shifting property states:

$$x(n - k) = Z^{-1}\left\{z^{-k}X(z)\right\} \qquad \text{Eqn. 2-5}$$

The proof of this property follows directly from the definition of the z-transform. An obvious question arises: "If the s-domain of a signal, x(t), is known and that same signal is digitized, what is the relationship between the s-domain transform and the z-transform?" The relationship or mapping is not

unique and depends on the viewpoint used. It is obvious that the trivial mapping, s = z, is inappropriate; the signal has been sampled. When a bandlimited signal is sampled (i.e., multiplied by a periodic impulse function), the spectrum of the resulting signal is repetitive as shown in Figure 2-9 (see Reference 10).

Figure 2-9 *Spectrum of Bandlimited Signal Repeated at Multiples of the Sampling Frequency, f_s.*
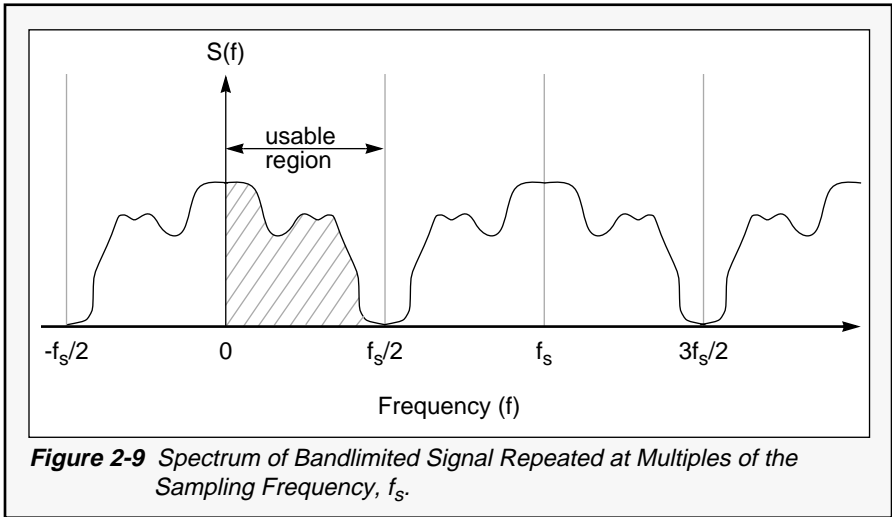
Clearly, the spectrum consists of the spectrum of the bandlimited signal repeated at multiples of the sampling frequency, $\omega_s = 2\pi f_s$. That is, the resulting spectrum is unique only between 0 and $\omega_s/2$ or multiples thereof; whereas, before the signal was sampled, the energy at frequencies greater than those in the signal was independent of the signal. Therefore, acceptable mappings would either reflect the cyclic nature of the spectrum of the sampled test or at least be linear over the frequencies of interest.

One mapping or transformation from the s-domain to the z-domain discussed later in this report is the bilinear transformation. To understand the origin of this transformation, consider the simple first-order linear analog filter with the system function:

$$H(s) = \frac{Y(s)}{X(s)} = \frac{b}{s+a} \qquad \text{Eqn. 2-6}$$

Recall the differentiation property of the s-transform when $x(t) = L^{-1}\{X(s)\}$ (where L^{-1} is the inverse Laplace transform operation); then the time derivative of $x(t)$ is:

$$\frac{d}{dt}x(t) = L^{-1}\{sX(s)\} \qquad \text{Eqn. 2-7}$$

where: $\frac{d}{dt}x(t)$ is the time derivative of x(t)

Using the differentiation property of Eqn. 2-7, the linear system described by Eqn. 2-6 can be expressed as follows:

$$\frac{d}{dt}y(t) + ay(t) = bx(t) \qquad \text{Eqn. 2-8}$$

If this differential equation is solved by expressing $y(t)$ with the trapezoidal integration formula,

$$y(t) = \int_{t_0}^{t} \frac{d}{d\tau}y(\tau)d\tau + y(t_0) \qquad \text{Eqn. 2-9}$$

where the approximate solution is given by:

$$y(t) = \frac{1}{2}\left[\frac{d}{dt}y(t) + \frac{d}{dt}y(t_0)\right](t - t_0) + y(t_0) \quad \text{Eqn. 2-10}$$

then using $\frac{d}{dt}y(t)$ from Eqn. 2-8 with $t = nT$ and $t_0 = (n-1)T$, Eqn. 2-10 can be expressed as follows:

$$(2 + aT)y(n) - (2 - aT)y(n - 1) = bT[x(n) + x(n - 1)]$$

Eqn. 2-11

Taking the z-transform of this difference of this difference equation and using the time shifting property of the z-transform, Eqn. 2-5 results in the z-domain system function:

$$H(z) = \frac{Y(z)}{X(z)} = \frac{b}{\frac{2}{T}\left(\frac{1 - z^{-1}}{1 + z^{-1}}\right) + a} \qquad \text{Eqn. 2-12}$$

Clearly, the mapping between the s-plane and the z-plane is:

$$s = \frac{2}{T}\left(\frac{1 - z^{-1}}{1 + z^{-1}}\right) \qquad \text{Eqn. 2-13}$$

This mapping is called bilinear transformation.

Although this transformation was developed using a first-order system, it holds, in general, for an Nth-order system (see Reference 14). By letting $s = \sigma + j\Omega$ and $z = re^{j\theta}$, it can be shown that the left-half plane in the s-domain is mapped inside the unit $r = 1$ circle in the z-domain under the bilinear transformation. More importantly, when $r = 1$ and $\sigma = 0$, the frequencies in the s-domain and the z-domain are related by:

$$\Omega = \frac{2}{T}\tan\frac{\theta}{2} \qquad\qquad \text{Eqn. 2-14}$$

or equivalently:

$$\theta = 2\tan^{-1}\frac{\Omega T}{2} \qquad\qquad \text{Eqn. 2-15}$$

where: θ is the digital domain normalized frequency equal to $2\pi f/f_s$

Ω is the analog domain frequency used in the analysis of the previous section

On the $j\Omega$ axis or equivalently along the frequency axis, the scale has been changed nonlinearly. The gain and phase values depicted on the vertical axis of Figure 1-2, Figure 1-4, Figure 1-6, and Figure 1-8 remain exactly the same in the digital domain (or z-plane). The horizontal (frequency) axis is modified so that an infinite frequency in the analog domain maps to one-half of the sample frequency, $f_s/2$, in the digital domain; whereas, for frequencies much less than $f_s/2$, the mapping is approximately 1:1

with $\theta = \Omega$. In summary, the bilinear transformation is a one-to-one nonlinear mapping from the s-domain into the z-domain in which high frequencies ($\Omega > 2\pi f_s/4$) in the s-domain are compressed into a small interval in the z-domain. Therefore, the gain and phase expressions of the previous section can be directly transformed into the digital domain by simply substituting Eqn. 2-14 into the corresponding expressions. This will be done for each filter type in the following paragraphs.

First, it is appropriate to introduce the direct-form implementation of a digital filter by noting that, in general, if the bilinear transformation of Eqn. 2-13 is substituted into the transfer function, H(s), of the previous section, the resulting H(z) will have the following generalized form:

$$H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}} \qquad \text{Eqn. 2-16}$$

where the digital domain coefficients, a_i and b_i, are exactly related to the s-domain characteristics of the system such as the center frequency, bandwidth, etc. In the direct-form implementation, the a_i and b_i are used directly in the difference equation, which can be easily programmed on a high-speed DSP such as the DSP56001. The time-domain difference equation is derived from the z-domain transfer functions by applying the inverse z-transform in general and the inverse time shifting

property in particular as follows:

$$Z^{-1}\{H(z)\} = Z^{-1}\{Y(z)/X(z)\}$$

$$= Z^{-1}\left\{\left[b_0 + b_1 z^{-1} + b_2 z^{-2}\right]/\left[1 + a_1 z^{-1} + a_2 z^{-2}\right]\right\}$$

<div align="right">Eqn. 2-17</div>

so that:

$$(z)\left[1 + a_1 z^{-1} + a_2 z^{-2}\right]\right\} = Z^{-1}\left\{X(z)\left[b_0 + b_1 z^{-1} + b_2\right.\right.$$

therefore, using the inverse time shifting property of Eqn. 2-5:

$$Z^{-1}\left\{X(z)z^{-k}\right\} = \{x(n-k)\}$$

$$\text{and}$$

$$Z^{-1}\left\{Y(z)z^{-k}\right\} = \{y(n-k)\}$$

Eqn. 2-17 becomes:

$$y(n) = b_0 x(n) + b_1 x(n-1) + b_2 x(n-2) + a_1 y(n-1) - a_2 y(n-2)$$

<div align="right">Eqn. 2-18</div>

Eqn. 2-18 can be directly implemented in software, where x(n) is the sample input and y(n) is the corresponding filtered digital output. When the filter output is calculated using Eqn. 2-18, y(n) is calculated using the direct-form implementation of the digital filter.

There are other implementations which can be used for the same system (filter) transfer function, H(z). The canonic-form implementation and the transpose-form implementation are discussed in subsequent sections. First, the directform implementation will be applied to the transfer function, H(s), developed in **SECTION 1 Introduction**.

2.6 Digital Lowpass Filter

Using the analog transfer function, H(s), from Figure 1-1, Eqn. 2-13 and Eqn. 2-14, the digital transfer function, H(z), becomes that shown in Figure 2-10, where the coefficients α, β, and γ are expressed in terms of the digital cutoff frequency, θ_c, and the damping factor, d. The value of the transfer function at $\theta = \theta_c$ in the digital domain is identical to the value of the s-domain transfer function at $\Omega = \Omega_c$:

$$H_z(e^{j\theta c}) = H_s(j\Omega_c) \qquad \text{Eqn. 2-19}$$

As shown in Figure 2-11, the digital gain and phase response calculated from the equations of

---
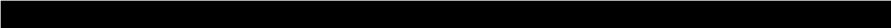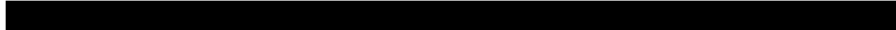
Figure 2-10 are similar to the analog plots shown in Figure 1-2, except for the asymmetry introduced by the zero at $f_s/2$. That is, the frequency axis is modified so that a gain of zero at $f = \infty$ in the s-domain corresponds to a gain of zero at $f = f_s/2$ in the z-domain. The fact that the magnitude of the transfer functions, H(s) and H(z), is identical once the proper frequency transformation is made is very useful for understanding the digital filter and its relationship to the analog equivalent. This fact is also useful for purposes of scaling the gain since the maximum magnitude of $G_s(\Omega_M) = G_Z(\theta_M)$, where Ω_M and θ_M are related by Eqn. 2-14. In other words, scaling analysis of the digital transfer function, H(z), can be done in the s-domain (the algebra is often easier to manage). Scaling of the gain is an essential part of digital filter implementation since the region of numeric calculations on fixed-point DSPs such as the DSP56001 are usually restricted to a range of -1 to 1.

Using the formulas given in Figure 2-10 with Eqn. 1-2 guarantees the behavior of the digital filter. Since the gain is scaled to unity at $f = 0$ (DC), the input data, x(n), in Figure 2-10 must be scaled down by a factor of $1/G_M$ from Eqn. 1-2 if the entire dynamic range of the digital network is to be utilized. The alternative procedure is automatic gain control to insure that x(n) is smaller than $1/G_M$ before it arrives at the filter input. For $d \geq \sqrt{2}$, the input does not require scaling since the gain of the filter will never exceed unity.
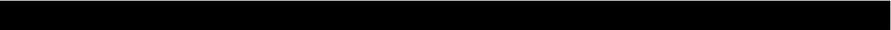
The DSP56001 code to implement the second-order lowpass filter section is shown in Figure 2-12. The address register modifiers are initially set to M0 = 4, M4 = 1, and M5 = 1 to allow use of the circular buffer or modulo addressing in this particular implementation (see Reference 8). Typically, this code would be an interrupt routine driven by the input data (A/D converter, for example) sample rate clock. The basic filter code and the interrupt overhead and data I/O moves for this second-order filter could be executed at a sample rate of nearly 1 MHz on the DSP56001.

2.7 Digital Highpass Filter

The highpass filter is nearly identical to the lowpass filter as shown by the formulas in Figure 2-13. As with the analog case, the digital highpass filter is just the mirror image of the lowpass filter (see Figure 2-14). The frequency transformation from high to low in the analog case is $\Omega \rightarrow l/\Omega$; whereas, in the digital case, it is $\theta \rightarrow \pi/\theta$.

The DSP56001 code is shown in Figure 2-15; as seen by comparison to the code shown in Figure 2-12, the same instruction sequence is used. The only difference is the coefficient data, which is calculated by the formulas given in Figure 2-13. The scaling mode is turned on so that a move from the A or B accumulator to the X or Y register or to memory results in an automatic multiply by two. The scaling mode is used not only in the code for the lowpass

case but also in the code for the highpass, band-stop, and bandpass cases.

Figure 2-10 *Direct-Form Implementation of Second-Order Lowpass IIR Filter and Analytical Formulas Relating Desired Response to Filter Coefficients*

The figure contains the following sections:

Network Diagram

Transfer Function

$$H(z) = \frac{\alpha(1 + 2z^{-1} + z^{-2})}{1/2 - \gamma z^{-1} + \beta z^{-2}}$$

Gain

$$G(\theta) = \frac{(1 + \cos\theta)(1 - \cos\theta_c)}{[(d\sin\theta\sin\theta_c)^2 + 4(\cos\theta - \cos\theta_c)^2]^{1/2}}$$

Phase

$$\phi(\theta) = \begin{cases} \tan^{-1}\left[\dfrac{2(\cos\theta - \cos\theta_c)}{d\sin\theta\sin\theta_c}\right] & \text{for } \theta \leq \theta_c \\[3mm] -\pi + \tan^{-1}\left[\dfrac{2(\cos\theta - \cos\theta_c)}{d\sin\theta\sin\theta_c}\right] & \text{for } \theta > \theta_c \end{cases}$$

Coefficients

$$\beta = \left(\frac{1}{2}\right)\frac{1 - d/2\sin\theta_c}{1 + d/2\sin\theta_c}$$

$$\gamma = (1/2 + \beta)\cos\theta_c$$

$$\alpha = (1/2 + \beta - \gamma)/4$$
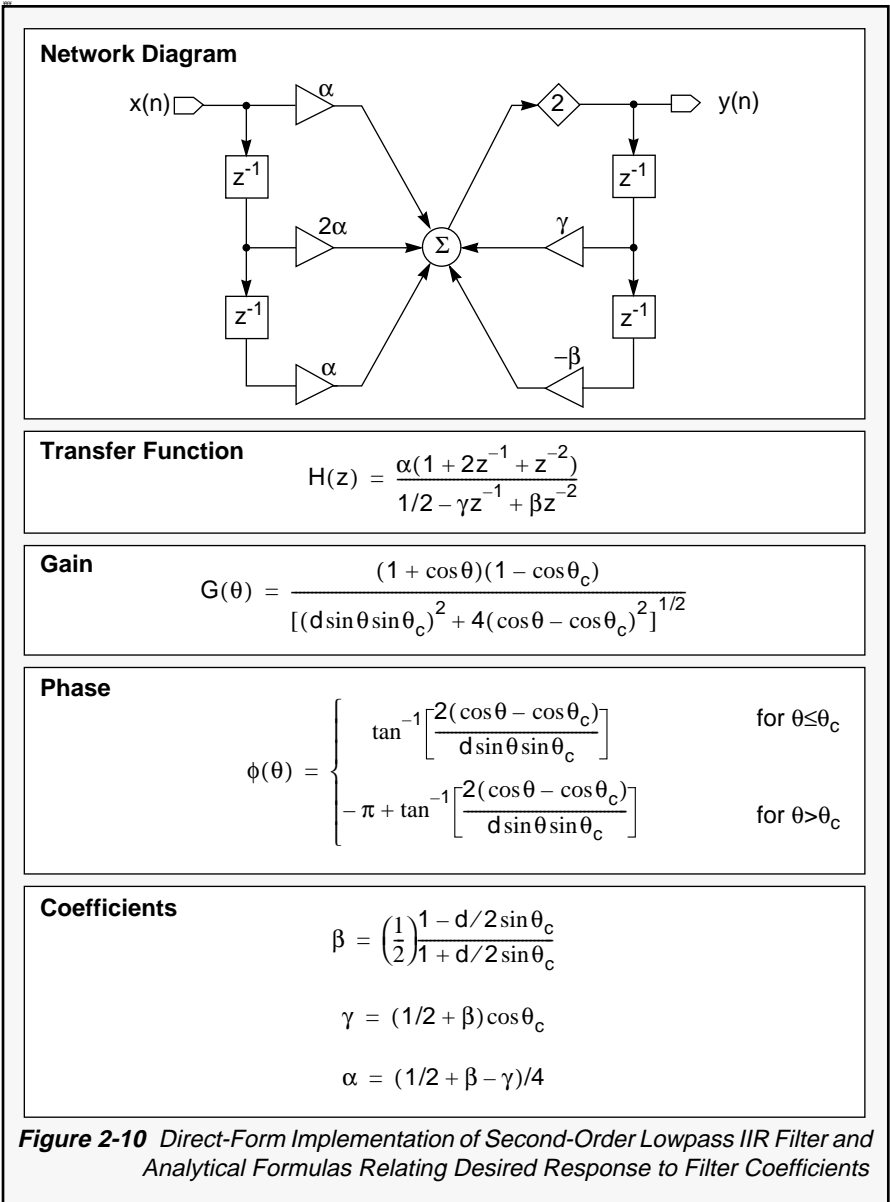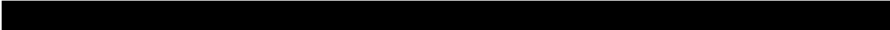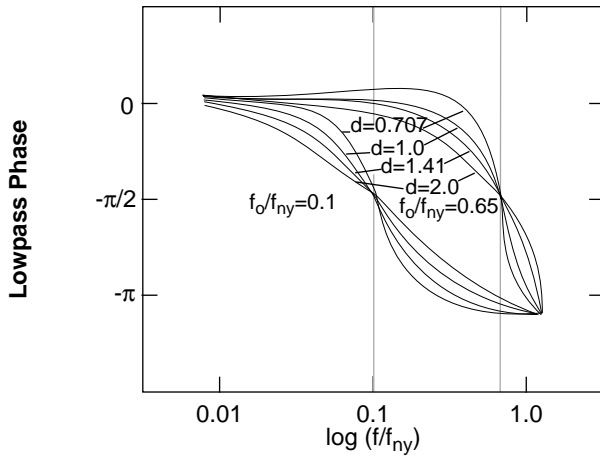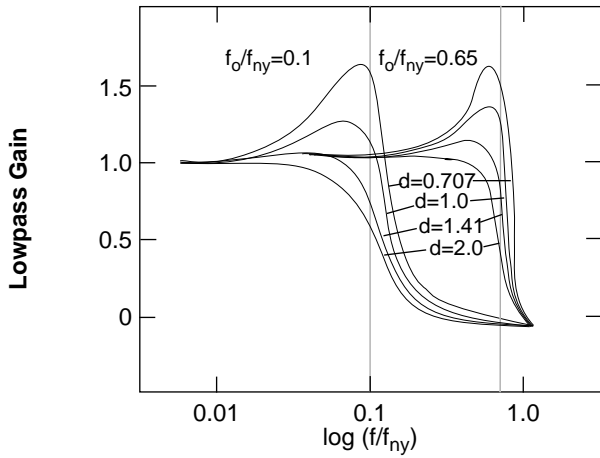
Figure 2-11 *Gain and Phase Response of Second-Order Lowpass IIR Filter*

NOTE: Nyquist frequency, f_{ny}, is equal to one-half the sample frequency f_s.

Difference Equation

$$y(n) = 2\{\alpha[x(n) + 2x(n-1) + x(n-2)] + \gamma y(n-1) - \beta y(n-2)\}$$

Data Structures

X:(R0)

2α
α
γ
$-\beta$
α

Y:(R4)

x(n-1)
x(n-2)

Y:(R5)

y(n-1)
y(n-2)

DSP56001 Code

```
                                    ;Y1=x(n) (Input)
                                    ;X0=α
MPY   X0,Y1,A   X:(R0)+,X0 Y:(R4)+,Y0 ;A=αx(n)
MAC   X0,Y0,A   X:(R0)+,X0 Y:(R4),Y0  ;A=A+2αx(n-1)
MAC   X0,Y0,A   X:(R0)+,X0 Y:(R5)+,Y0 ;A=A+αx(n-2)
MAC   X0,Y0,A   X:(R0)+,X0 Y:(R5),Y0  ;A=A+γy(n-1)
MAC   X0,Y0,A   X:(R0)+,X0 Y1,Y:(R4)  ;A=A-βy(n-2)
MOVE            A,X1       A,Y:(R5)    ;y(n-2)=2A (assumes scaling
                                    ;mode is set).
                                    ;X1 is Output.
```

Total Instruction Cycles

6 Icyc @ 20 MHz = 600ns

Figure 2-12 *DSP56001 Code and Data Structures for Second-Order Direct-Form Implementation of a Lowpass IIR Filter*

Figure 2-13 *Direct-Form Implementation of Second Order Highpass IIR Filter and Analytical Formulas Relating Desired Response to Filter Coefficients*

The figure contains the following sections:

Network Diagram

Transfer Function

$$H(z) = \frac{\alpha(1 - 2z^{-1} + z^{-2})}{1/2 - \gamma z^{-1} + \beta z^{-2}}$$

Gain

$$G(\theta) = \frac{(1 - \cos\theta)(1 + \cos\theta_c)}{[(d\sin\theta\sin\theta_c)^2 + 4(\cos\theta - \cos\theta_c)^2]^{1/2}}$$

Phase

$$\phi(\theta) = \begin{cases} \pi + \tan^{-1}\left[\dfrac{2(\cos\theta - \cos\theta_c)}{d\sin\theta\sin\theta_c}\right] & \text{for } \theta \leq \theta_c \\[3mm] \tan^{-1}\left[\dfrac{2(\cos\theta - \cos\theta_c)}{d\sin\theta\sin\theta_c}\right] & \text{for } \theta > \theta_c \end{cases}$$

Coefficients

$$\beta = \left(\frac{1}{2}\right)\frac{1 - d/2\sin\theta_c}{1 + d/2\sin\theta_c}$$

$$\gamma = (1/2 + \beta)\cos\theta_c$$

$$\alpha = (1/2 + \beta + \gamma)/4$$
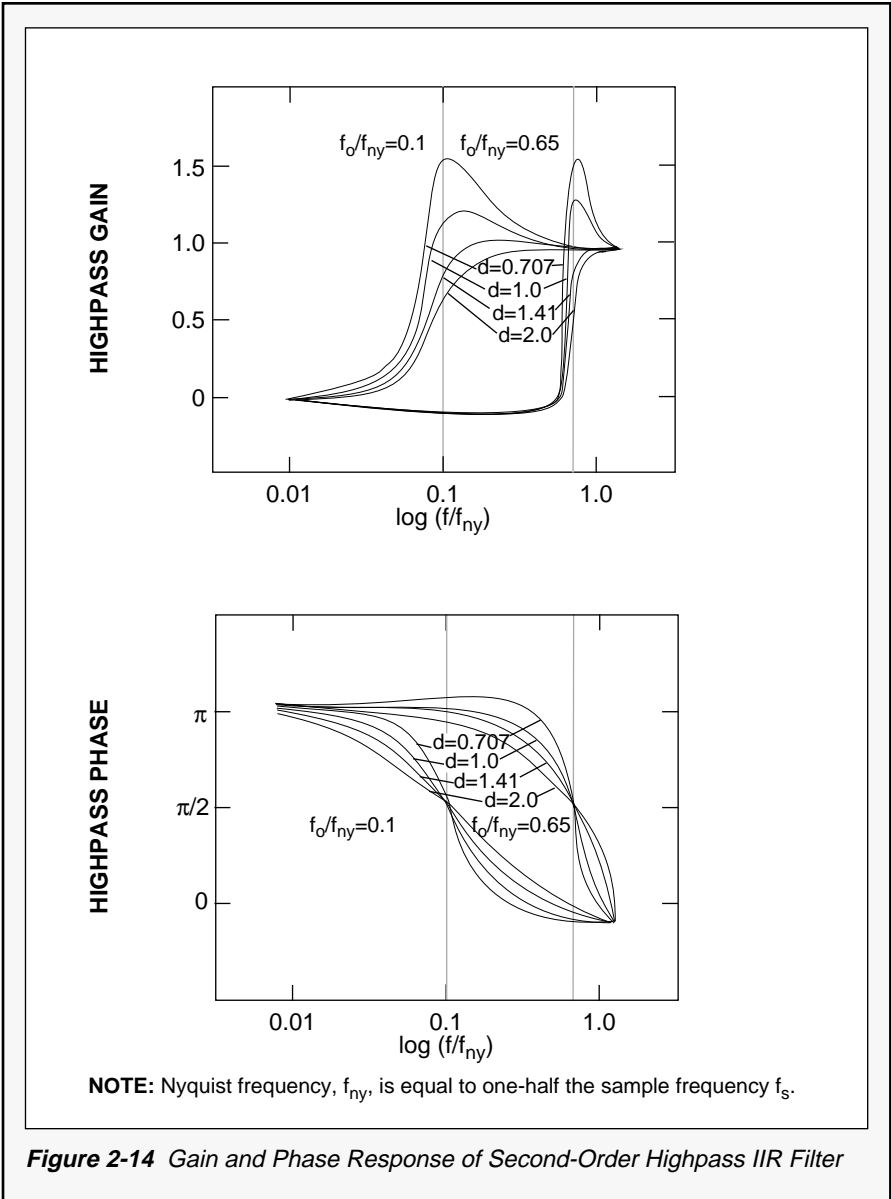
Figure 2-14 Gain and Phase Response of Second-Order Highpass IIR Filter

Difference Equation

$$y(n) = 2\{\alpha[x(n)-2x(n-1)+x(n-2)]+\gamma y(n-1)-\beta y(n-2)\}$$

Data Structures

X:(R0)

-2α
α
γ
$-\beta$
α

Y:(R4)

x(n-1)
x(n-2)

Y:(R5)

y(n-1)
y(n-2)

DSP56001 Code

```
                                  ;Y1=x(n) (Input)
                                  ;X0=α
MPY   X0,Y1,A  X:(R0)+,X0  Y:(R4)+,Y0  ;A=αx(n)
MAC   X0,Y0,A  X:(R0)+,X0  Y:(R4),Y0   ;A=A-2αx(n-1)
MAC   X0,Y0,A  X:(R0)+,X0  Y:(R5)+,Y0  ;A=A+αx(n-2)
MAC   X0,Y0,A  X:(R0)+,X0  Y:(R5),Y0   ;A=A+γy(n-1)
MAC   X0,Y0,A  X:(R0)+,X0  Y1,Y:(R4)   ;A=A-βy(n-2)
MOVE           A,X1        A,Y:(R5)    ;y(n)=2A (assumes scaling
                                  ;mode is set).
                                  ;X1 is Output.
```

Total Instruction Cycles

6 Icyc @ 20 MHz = 600ns

Figure 2-15 *DSP56001 Code and Data Structures for Second-Order Direct-Form Implementation of a Highpass IIR Filter*

2.8 Digital Bandstop Filter

The formulas and network diagram for the digital bandstop filter are presented in Figure 2-16. The DSP56001 code from Figure 2-17 is identical to that for the lowpass and highpass cases except for the coefficient data calculated from the equations of Figure 2-16. Scaling of this filter is not a problem for the singlesection case since the gain from the equation in Figure 2-16 never exceeds unity (as is true in the analog case as seen by the gain equation from Figure 1-5). Figure 2-18 is the calculated gain and phase of the digital filter, which should compare to the response curves of the equivalent analog filter plotted in Figure 1-6.

2.9 Digital Bandpass Filter

Because there is one less coefficient in the bandpass network (see Figure 2-19), one instruction can be saved in the DSP56001 code implementation shown in Figure 2-20. Otherwise, the instructions are identical to those in the other three filter routines. Like the second-order bandstop network, the maximum response at the center frequency, θ_0, is unity for any value of Q so that scaling need not be considered in the implementation of a single-section bandpass filter. This is true when the formulas for α, β, and γ (from Figure 2-19) are used in the direct-form implementation in Figure 2-20. Figure 2-21 is the calculated gain and phase of the digital filter, which should compare to the response curves of the equivalent analog filter plotted in Figure 1-8.

Network Diagram

x(n) → ... α ... 2 ... → y(n)

z^{-1}

$-2\alpha\cos\theta_0$ Σ γ

z^{-1}

α $-\beta$
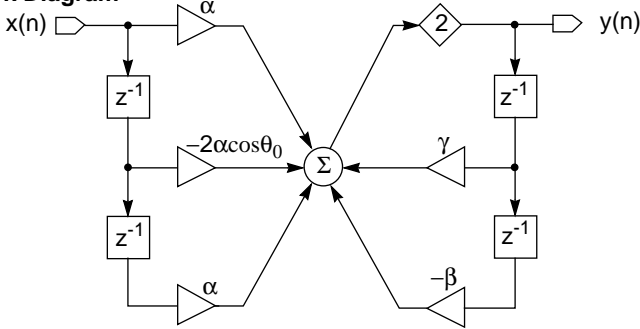
Transfer Function

$$H(z) = \frac{\alpha(1 - 2\cos\theta_0 z^{-1} + z^{-2})}{1/2 - \gamma z^{-1} + \beta z^{-2}}$$

Gain

$$G(\theta) = \frac{|\cos\theta - \cos\theta_c|}{[(d\sin\theta\sin\theta_0)^2 + 4(\cos\theta - \cos\theta_0)^2]^{1/2}}$$

Phase

$$\phi(\theta) = \tan^{-1}\left[\frac{2(\cos\theta - \cos\theta_0)}{d\sin\theta\sin\theta_0}\right]$$

Coefficients

$$d = \frac{2\tan(\theta_0/2Q)}{\sin\theta_0}$$

$$\beta = \left(\frac{1}{2}\right)\frac{1 - \tan(\theta_0/2Q)}{1 + \tan(\theta_0/2Q)}$$

$$\gamma = (1/2 + \beta)\cos\theta_0$$

$$\alpha = (1/2 + \beta)/2$$

Figure 2-16 *Direct-Form Implementation of Second-Order Bandstop IIR Filter and Analytical Formulas Relating Desired Response to Filter Coefficients*

Difference Equation

$$y(n) = 2\{\alpha[x(n)-2\cos\theta_0 x(n-1) + x(n-2)] + \gamma y(n-1) - \beta y(n-2)\}$$

Data Structures

X:(R0)

$-2\alpha\cos\theta_0$
α
γ
$-\beta$
α

Y:(R4)

x(n-1)
x(n-2)

Y:(R5)

y(n-1)
y(n-2)

DSP56001 Code

```
                                    ;Y1=x(n) (Input)
                                    ;X0=α
MPY   X0,Y1,A   X:(R0)+,X0  Y:(R4)+,Y0  ;A=αx(n)
MAC   X0,Y0,A   X:(R0)+,X0  Y:(R4),Y0   ;A=A-2αcosq₀x(n-1)
MAC   X0,Y0,A   X:(R0)+,X0  Y:(R5)+,Y0  ;A=A+αx(n-2)
MAC   X0,Y0,A   X:(R0)+,X0  Y:(R5),Y0   ;A=A+γγy(n-1)
MAC   X0,Y0,A   X:(R0)+,X0  Y1,Y:(R4)   ;A=A-βy(n-2)
MOVE            A,X1        A,Y:(R5)     ;y(n)=2A (assumes scaling
                                    ; mode is set).
                                    ;X1 is Output.
```

Total Instruction Cycles

6 Icyc @ 20 MHz = 600ns

Figure 2-17 *DSP56001 Code and Data Structures for Second-Order Direct-Form Implementation of a Bandstop IIR Filter*
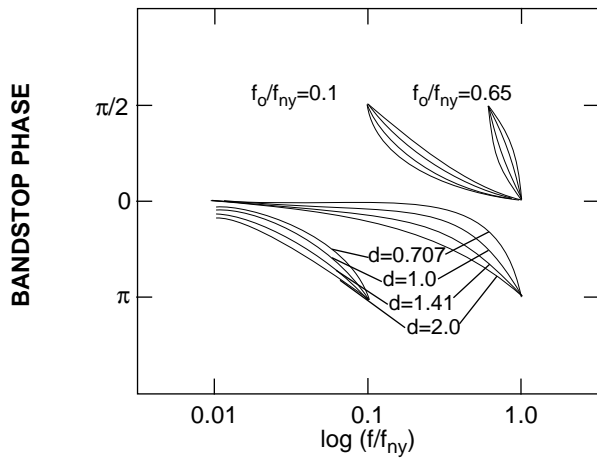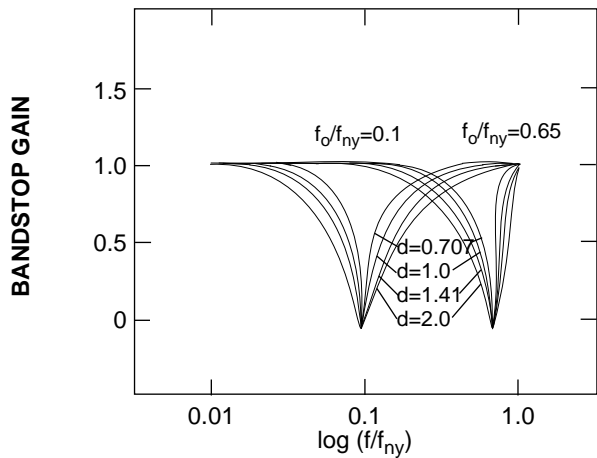
Figure 2-18 Gain and Phase Response of Second-Order Bandstop IIR Filter

NOTE: Nyquist frequency, f_{ny}, is equal to one-half the sample frequency f_s.

2.10 Summary of Digital Coefficients

Figure 2-22 gives a summary of the coefficient values for the four basic filter types. Note that the coefficient, β has the same form for all four filter types and that it can only assume values between 0 and 1/2 for practical filters. β is bounded by 1/2 because Q (or d) and θ_0 are not independent. For Q >> 1, $\beta \rightarrow 1/2$; whereas, for $\theta_0 = f_s/4$ and Q = 1/2, $\beta \rightarrow 0$. These properties are independent of the form of implementation; they are only dependent on the form of the transfer function. Alternate implementations (difference equations) will be described in the following sections.

Note that the Q described in Figure 2-22 meets the traditional requirements (i.e., Q is the ratio of the bandwidth at the -3 dB points divided by the center frequency). The formula for β can be modified in the case of the bandpass or bandstop filter by replacing the damping coefficient, d, with the formula for Q. When the coefficients are described in this manner, a constant Q filter results. When the bandwidth is any function of center frequency, this relationship between d and Q makes it impossible to implement a bandpass or bandstop filter by replacing Q with the desired function of bandwidth and center frequency.

Figure 2-23 shows the relationship between the pole of the second-order section and the center frequency. Note that the pole is on the real axis for d>2, where d is also constrained by d < 2/sin θ_0.

Network Diagram
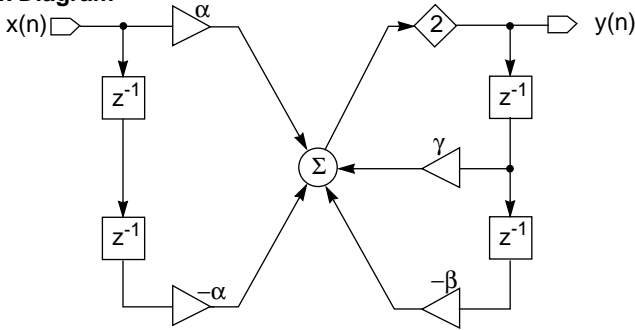
Transfer Function

$$H(z) = \frac{\alpha(1 - z^{-2})}{1/2 - \gamma z^{-1} + \beta z^{-2}}$$

Gain

$$G(\theta) = \frac{d\sin\theta_o \sin\theta}{[(d\sin\theta\sin\theta_0)^2 + 4(\cos\theta - \cos\theta_0)^2]^{1/2}}$$

Phase

$$\phi(\theta) = -\tan^{-1}\left[\frac{2(\cos\theta - \cos\theta_0)}{d\sin\theta\sin\theta_0}\right]$$

Coefficients

$$d = \frac{2\tan(\theta_0/2Q)}{\sin\theta_0}$$

$$\beta = \left(\frac{1}{2}\right)\frac{1 - \tan(\theta_0/2Q)}{1 + \tan(\theta_0/2Q)}$$

$$\gamma = (1/2 + \beta)\cos\theta_0$$

$$\alpha = (1/2 - \beta)/2$$

Figure 2-19 *Direct-Form Implementation of Second-Order Bandpass IIR Filter and Analytical Formulas Relating Desired Response to Filter Coefficients*

Difference Equation

$$y(n) = 2\{\alpha[x(n) - x(n-2)] + \gamma y(n-1) - \beta y(n-2)\}$$

Data Structures

X:(R0)

$-\alpha$
γ
$-\beta$
α

Y:(R4)

x(n-1)
x(n-2)

Y:(R5)

y(n-1)
y(n-2)

DSP56001 Code

```
                                        ;Y1=x(n) (Input)
                                        ;X0=α
MPY     X0,Y1,A  X:(R0)+,X0 Y:(R4),Y0   ;A=αx(n)
MAC     X0,Y0,A  X:(R0)+,X0 Y:(R5)+,Y0  ;A=A-αx(n-2)
MAC     X0,Y0,A  X:(R0)+,X0 Y:(R5),Y0   ;A=A+γy(n-1)
MAC     X0,Y0,A  X:(R0)+,X0 Y1,Y:(R4)+  ;A=A-βy(n-2)
MOVE             A,X1       A,Y:(R5)     ;y(n)=2A (assumes scaling
                                        ; mode is set).
                                        ;X1 is Output.
```
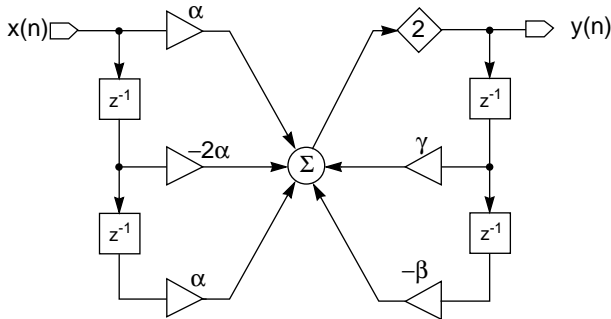
Total Instruction Cycles

5 Icyc @ 20 MHz = 500ns

Figure 2-20 *DSP56001 Code and Data Structures for Second-Order Direct-Form Implementation of a Bandpass IIR Filter*
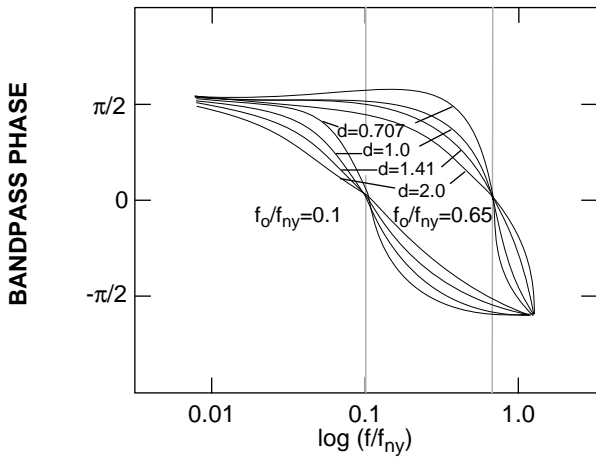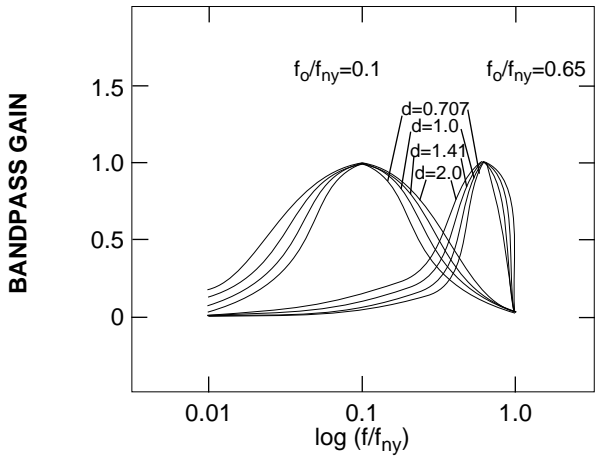
Figure 2-21 *Gain and Phase Response of Second-Order Bandpass IIR Filter*

Z-Domain Transfer Function

$$H(z) = \frac{\alpha(1 + \mu z^{-1} + \sigma z^{-2})}{1/2 - \gamma z^{-1} + \beta z^{-2}}$$

Difference Equation (Direct Form)

$$y(n) = 2\{\alpha[x(n) + \mu x(n-1) + \sigma x(n-2)] + \gamma y(n-1) - \beta y(n-2)\}$$

Coefficients

$$\beta = \frac{1}{2}\left[\frac{1 - 1/2d\sin\theta_0}{1 + 1/2d\sin\theta_0}\right] \qquad d = \frac{2\tan(\theta_0/2Q)}{\sin\theta_0} \qquad \gamma = (1/2 + \beta)\cos\theta_0$$

where $0 < \beta < 1/2$ and $\quad Q = \dfrac{\theta_0}{\Delta_0} = \dfrac{2\pi f_0/f_s}{2\pi(f_2 - f_1)/f_s} = \dfrac{f_0}{f_2 - f_1}$

where f_0 is the center frequency of the bandpass or bandstop filter, f_1 and f_2 are the half-power points (where gain is equal to $1/\sqrt{2}$), and f_s is the sample frequency. Note the f_0 is replaced with f_c in the lowpass and highpass cases.

Numerator Coefficients				
Type	α	μ	σ	**Unity Gain at**
Lowpass	$(1/2 + \beta - \gamma)/4$	2	1	$f = 0$
Highpass	$(1/2 + \beta + \gamma)/4$	-2	1	$f = f_s/2$
Bandpass	$(1/2 - \beta)/2$	0	-1	$f = f_0$
Bandstop	$(1/2 + \beta)/2$	$-2\cos\theta_0$	1	$f = 0$ and $f = f_s/2$

NOTE: $\theta_0 = 2\pi f_0/f_s$

Figure 2-22 Summary of Digital Coefficients for the Four Basic Filter Types

Pole Equation of H(z)

$$Z_p = r\cos\theta_p + jr\sin\theta_p$$

$$= \gamma \pm j\sqrt{2\beta - \gamma^2}$$

For d<2

$$= \frac{\cos\theta_0 \pm j\sin\theta_0\sqrt{1 - (1/2d)^2}}{1 + \frac{1}{2}d\sin\theta_0}$$

where: $\beta = \frac{1}{2}(2 - d\sin\theta_0)/(2 + d\sin\theta_0)$ and $\gamma = (1/2 + \beta)\cos\theta_0$

Distance from Origin to Pole is $|Z_p| = \sqrt{2\beta}$

For d>2 $\quad Z_p = \gamma - \sqrt{\gamma^2 - 2\beta}$

$$= \frac{\cos\theta_0 - \sin\theta_0\sqrt{(1/2d)^2 - 1}}{1 + \frac{1}{2}d\sin\theta_0}$$

where: $\theta_p = 0$ To satisfy requirement 0<β<1/2 results in $\frac{1}{2}d\sin\theta_0 < 1$
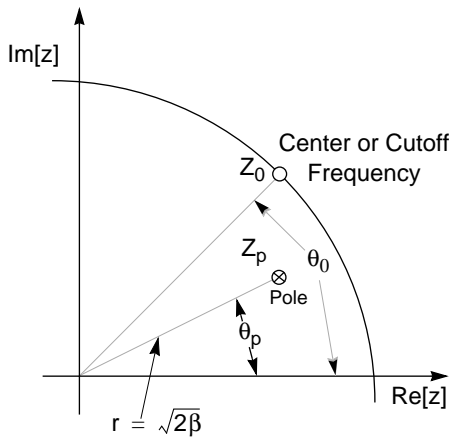
Figure 2-23 *Pole Location and Analysis of Second-Order Section*

Single-Section Canonic Form
(Direct Form II)

"The canonic (direct form II) network has trade-offs that must be carefully understood and analyzed for the particular application."

The single-section canonic form network is discussed in the following paragraphs.

3.11 The Canonic-Form Difference Equation

The direct-form difference equation, rewritten from Eqn. 2-18, is:

$$y(n) = \sum_{i=0}^{2} b_i x(n-1) - \sum_{j=1}^{2} a_j y(n-1) \qquad \text{Eqn. 3-20}$$

Eqn. 3-20 can be represented by the diagram of Figure 3-24 (a). This diagram is the same as those shown in Figure 2-10, Figure 2-13, Figure 2-16, and Figure 2-19, except that the summations have been separated to highlight the correspondence with Eqn. 3-20. From this diagram, it is clear that the direct-form implementation requires four delay elements or, equivalently, four internal memory locations.
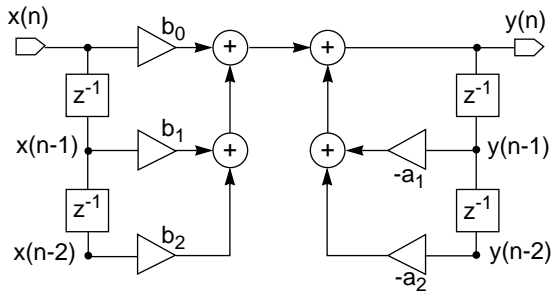
(a) Direct-Form Network Diagram of Equation 3-1
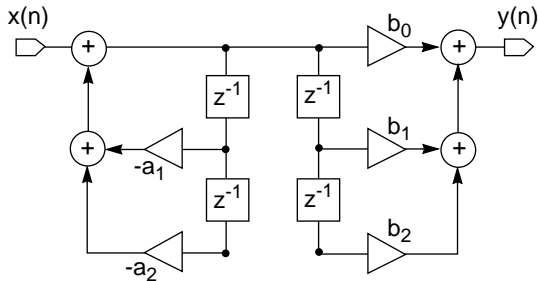
(b) Interchanged Left and Right Halves of Direct-form Network
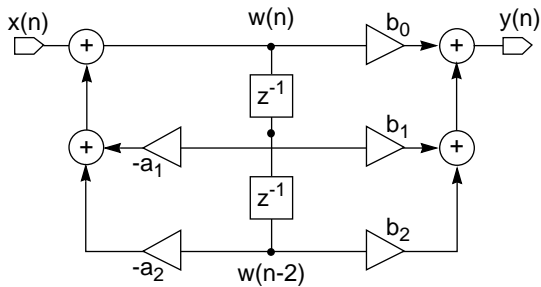
(c) Collapsing Delay Terms (part 'a' above) Resulting in Canonic-Form Diagram

Figure 3-24 *The Second-Order Canonic (Direct Form II) IIR Filter Network*

The diagram of Figure 3-24(b) represents the same transfer function implemented by Eqn. 3-20, but now the delay variable is w(n). Comparison of this network with that of the direct-form network of Figure 3-24 (a) shows that interchanging the order of the left and right halves does not change the overall system response (see Reference 11).

The delay elements can then be collapsed to produce the final canonic-form network shown in Figure 3-24 (c). As a result, the memory requirements for the system are reduced to the minimum (two locations); therefore, this realization of the IIR filter is often referred to as the canonic form. The system difference equations for the canonic realization are:

$$y(n) = \sum_{i=0}^{2} b_i(w(n-i)) \qquad \text{Eqn. 3-21a}$$

where:

$$w(n) = x(n) - \sum_{j=1}^{2} aj(w(n-j)) \qquad \text{Eqn. 3-2b}$$

To prove that Eqn. 3-21a and Eqn. 3-2b are equivalent to Eqn. 3-20, the following procedure can be used. First, combine Eqn. 3-21a and Eqn. 3-2b:

$$y(n) = \sum_{i=0}^{2} b_i \left[x(n-i) - \sum_{j=1}^{2} ajw(n-j-i) \right]$$

$$= \sum_{i=0}^{2} b_i x(n-i) - \sum_{j=1}^{2} aj \sum_{i=0}^{2} b_i w(n-i-j)$$

$$= \sum_{i=0}^{2} b_i x(n-i) - \sum_{j=1}^{2} ajy(n-j) \qquad \text{Eqn. 3-3}$$

The last step uses the definition for y(n) from Eqn. 3-21a. The result is exactly equivalent to Eqn. 3-20, the direct-form difference equation. To utilize the scaling mode on the DSP56001, it is advantageous to write Eqn. 3-21a and Eqn. 3-2b as follows:

$$y(n) = 2\left\{\frac{1}{2}w(n) + \frac{\mu}{2}w(n-1) + \frac{\alpha}{2}w(n-2)\right\}$$

Eqn. 3-4a

and

$$w(n) = 2\{\alpha x(n) + \gamma w(n-1) - \beta w(n-2)\}$$ Eqn. 3-4b

where: the coefficients have been substituted for the a_j and b_j

From these equations, it can be seen that both y(n) and w(n) depend on a sum of products and therefore are the output of an accumulator. The accumulators on the DSP56001 have eight extension bits; thus, the sum can exceed unity by 255 without overflowing. However, when the contents of a 56-bit accumulator are transferred to a 24-bit register or memory location (i.e., delay element), an overflow error may occur. The digital filter designer must insure that only values less than unity are stored. Of course, even if memory had the precision of the accumulators, overflow can still occur if the accumulator itself experiences overflow. However, intermediate sums are allowed to exceed the capacity of the accumulator if the final result can be represented in the accumulator (a result of the cir-

cular nature of twos-complement arithmetic).
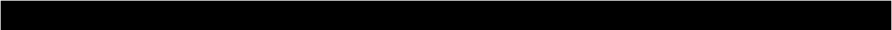
To insure that overflow does not occur, it is necessary to calculate the gain at the internal nodes (accumulator output) of a filter network. Although canonic realization is susceptible to overflow, it is advantageous because of the minimum storage requirements and implementation in fewer instruction cycles.

3.12 Analysis of Internal Node Gain

Calculating the gain at any internal node is no different than calculating the total network gain. In Figure 3-25, the transfer function of node w(n) is $H_w(z)$. $H_w(z)$ represents the transfer function from the input node, x(n), to the internal node, w(n). The gain, $G_w(\theta)$, at w(n) is found in the standard manner by evaluating the magnitude of $H_w(z)$ as shown in Figure 3-26. Note that the flow diagram of Figure 3-25 uses the automatic scaling mode (multiply by a factor of two when transferring data from the accumulator to memory) so that the coefficients are by definition less than one. The peak gain, g_o, of $G_w(\theta)$ is found by taking the derivative of $G_w(\theta)$ with respect to θ and setting the result equal to zero. A simple expression for g_o is derived as follows:

$$g_o = \frac{\alpha}{\left(\frac{1}{2}-\beta\right)\sin\theta p} \text{ for } \left[\frac{\left(\frac{1}{2}+\beta\right)^2}{2\beta}\right]\cos\theta_o \le 1 \qquad \text{Eqn. 3-5}$$

---

where: θ_p *is the angle (see Figure 2-23) to the*
pole of the filter

Figure 3-25 *Internal Node Transfer Function, $H_w(z)$, of Canonic (Direct Form II) Network*
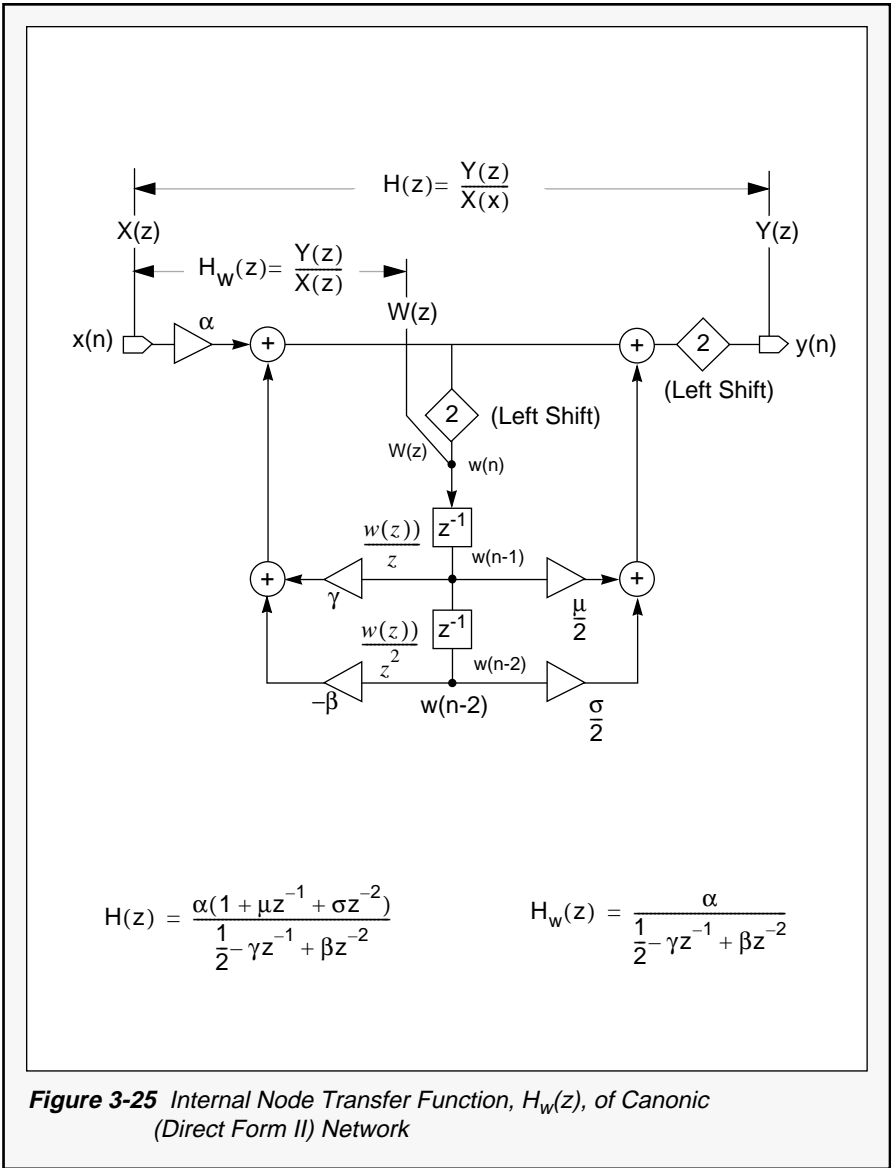
Since this result does not depend on the numerator of the filter transfer function, H(z), the result is valid for all four basic filter types. However, as shown by the example given for a bandpass network in Figure 3-26, g_o may exceed unity by a large amount, especially for filters having poles at frequencies much less than $f_s/2$ where $\sin \theta p \ll 1$. To compensate, the input must be scaled down by an amount equal to $1/g_o$ or guaranteed not to exceed $1/g_o$ before arriving at the filter. This scaling aspect of the canonic-form network is a disadvantage, but this network has the advantage of being implemented in one less instruction than the other filter realizations for the lowpass, highpass, and bandstop filters; of course, less memory is required since only two intermediate variables are stored.

In general, the behavior of systems at internal nodes can be unexpected. For instance, it is interesting to note that the frequency at which the gain at the internal node of the canonic IIR filter section peaks is not the same as the frequency at which the gain of the filter peaks as given by the poles of the filter. From Figure 2-13, this behavior is expressed by:

$$\frac{\sqrt{2\beta}}{\frac{1}{2}+\beta} = \frac{\cos\theta_o}{\cos\theta_p} \qquad \text{Eqn. 3-6}$$

The canonic (direct form II) network has trade-offs that must be carefully understood and analyzed for the particular application.

Gain at w(n) is:
$$H_w(z) = \frac{W(z)}{X(z)} = \frac{\alpha}{\frac{1}{2} - \gamma z^{-1} + \beta z^{-2}}$$

$$G_w(\theta) = \sqrt{H_w(e^{j\theta})H_w(e^{-j\theta})}$$

$$= \frac{\alpha}{\sqrt{\left(\frac{1}{2}-\beta\right)^2 \sin^2\theta + \left(\frac{1}{2}+\beta\right)^2 (\cos\theta - \cos\theta_0)^2}}$$

where: $\gamma = (\frac{1}{2} + \beta) \cos\theta_0$ has been used.

Peak Gain is: $\left.\frac{d}{d\theta}G_w(\theta)\right|_{\theta = \theta_m} = 0$

Frequency of peak gain is: $\quad \cos\theta_m = \xi\cos\theta_0 \qquad\qquad$ for $\xi\cos\theta_0 \leq 1$

$$= 1 \qquad\qquad\qquad \text{otherwise}$$

where: $\quad \xi \equiv \dfrac{\left(\frac{1}{2}+\beta\right)^2}{2\beta}$

$$g_0 \equiv G_w(\theta_m) = \frac{\alpha}{\left(\frac{1}{2}-\beta\right)\sqrt{1 - \frac{\gamma^2}{2\beta}}} \qquad\qquad \text{for}\, \xi\cos\theta_0 \leq 1$$

$$= \frac{\alpha}{\left(\frac{1}{2}+\beta\right)(1 - \cos\theta_0)} \qquad\qquad \text{otherwise}$$

EXAMPLE: Maximum Internal Node Gain for Bandpass Filter

$$g_0 = \frac{1}{2\sin\theta p} \qquad\qquad \text{for}\,\xi\cos\theta_0 \leq 1$$

where $\gamma^2 = 2\beta^2 \cos^2\theta_p$ has been used and $\alpha = (1/2-\beta)/2$ for a bandpass filter. If $\sin\theta_p > 1/2$, then $g_0 < 1$; otherwise, an overflow (i.e., $g_0 > 1$) may occur at the internal node, w(n), unless the input is scaled down by $1/g_0$.

Figure 3-26 *Internal Node Gain Analysis of Second-Order Canonic Form*

Difference Equation

$$y(n) = 2\left\{\frac{1}{2}w(n) + \frac{\mu}{2}w(n-1) + \frac{\sigma}{2}w(n-2)\right\}$$

$$w(n) = 2\{\alpha x(n) + \gamma w(n-1) - \beta w(n-2)\}$$

Data Structures

X:(R0)

w(n-1)
w(n-2)

Y:(R4)

γ
$-\beta$
$\mu/2$
$\sigma/2$
α

DSP56001 Code

```
                                      ;Y1=x(n) (Input)
                                      ;X0=α
MPY     X0,Y1,A  X:(R0)+,X0 Y:(R4)+,Y0  ;A=αx(n)
MAC     X0,Y0,A  X:(R0),X1  Y:(R4)+,Y0  ;A=A+γw(n-1)
MACR    X1,Y0,A  X0,X:(R0)- Y:(R4)+,Y0  ;A=A-βw(n-2)
MAC     X0,Y0,A  A,X:(R0)   Y:(R4)+,Y0  ;A=1/2w(n)+μ/2w(n-1)
MACR    X1,Y0,A  A,X0       Y:(R4)+,Y0  ;A=A+σ/2w(n-2). X0=2A
                                      ;(assumes scaling mode
                                      ;is set).X0 is Output.
```

Total Instruction Cycles

5 Icyc @ 20 MHz = 500ns

Figure 3-27 *Internal Node Gain Analysis of Second-Order Canonic Form*

3.13 Implementation on the DSP56001

Figure 3-27 shows the DSP56001 code and data structures for implementation of the single-section canonic-form network. Note that the modifier register M4 is set equal to 4 to allow circular operation for addressing coefficient data. M0 is set to FFFF to turn off circular addressing for w(n-1) and w(n-2). ■
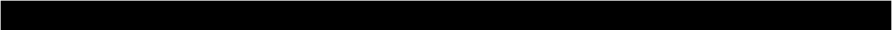
Single-Section Transpose Form

"The modifier register M4 is initially set to a value of 4 so that a circular buffer can be conveniently used to address the coefficient data."

A third realization of IIR filters is the transpose form (direct form I) shown in Figure 4-1. This network implementation can be derived directly from the direct-form difference equation (see Figure 4-1) or by taking the transpose of the canonic network (see References 10 and 11). The transpose realization is characterized by three accumulator operations. One reason for the popularity of this realization is that, like the canonic realization, it only requires two memory locations; however, unlike the canonic realization, it is much less prone to overflow at internal nodes. The disadvantage is that this realization requires more instructions to implement.

4.14 Gain Evaluation of Internal Nodes

Using the same techniques used to calculate $H_w(z)$ for the canonic realization, $H_u(z)$ and $H_v(z)$ are found as shown in Figure 4-2 and Figure 4-3. The resulting expressions, unlike the canonic-form results, depend on the numerator of the transfer function; thus, the internal gains, $G_u(\theta)$ and $G_v(\theta)$, have different forms for the different filter types. In the case of the bandpass filter, these results simplify significantly so that a closed-form

expression for the maximum gain at the internal nodes can be derived by calculating the maxima of the gain functions. For the bandpass and bandstop networks, the maximum of $G_u(\theta)$ and $G_v(\theta)$ is $g_m = \beta + 1/2$. Since, $\beta < 1/2$, then $g_m < 1$ so that no overflow occurs at these nodes in the bandpass or bandstop case.

Figure 4-4 contains example plots of $G_u(\theta)$ and $G_v(\theta)$ for the second-order transpose-form lowpass filter with various values of cutoff frequency, θ_c, and damping factor, d. In most cases, $G_u(\theta)$ and $G_v(\theta)$ never exceed the maximum value of $G(\theta)$, so that, if the total gain does not exceed unity, the internal nodes will not exceed unity (i.e., no overflow).

4.15 Implementation on the DSP56001

The DSP56001 code and data structures for a single-section second-order transpose-form network are shown in Figure 4-5. Referring to the network diagram of Figure 4-1, the diamond blocks enclosing 1/2 are represented in the code by an accumulator shift right (ASR) instruction. The 2 that is enclosed by a diamond block can be implemented by the automatic scaling mode (equivalent to a left shift of the accumulator) feature of the DSP56001. The modifier register M4 is initially set to a value of 4 so that a circular buffer can be conveniently used to address the coefficient data. Note that this network requires more instructions than either of the previous two network forms.

Network Diagram

Direct-Form Difference Equation

$$y(n) = 2\{\alpha[x(n) + \mu x(n-1) + \sigma x(n-2)] + \gamma y(n-1) - \beta y(n-2)\}$$

$$= 2\{\alpha x(n) + \alpha\mu x(n-1) + \gamma y(n-1) + [\alpha\sigma x(n-2) - \beta y(n-2)]\}$$

$$= 2\left\{\alpha x(n) + [\alpha\mu x(n-1) + \gamma y(n-1) + \frac{1}{2}u(n-2)]\right\}$$

$$= 2\left\{\alpha x(n) + \frac{1}{2}v(n-1)\right\}$$

Transpose-Form Difference Equation

$$y(n) = 2\left\{\alpha x(n) + \frac{1}{2}v(n-1)\right\}$$

$$\text{where} \quad v(n) = 2\left\{\alpha\mu x(n) + \gamma y(n) + \frac{1}{2}u(n-1)\right\}$$

$$\text{and} \quad u(n) = 2\{\alpha\sigma x(n) - \beta y(n)\}$$

Figure 4-1 *Network Diagram for Transpose-Form (Direct Form I)*
Implementation of a Single-Section Second-Order Filter
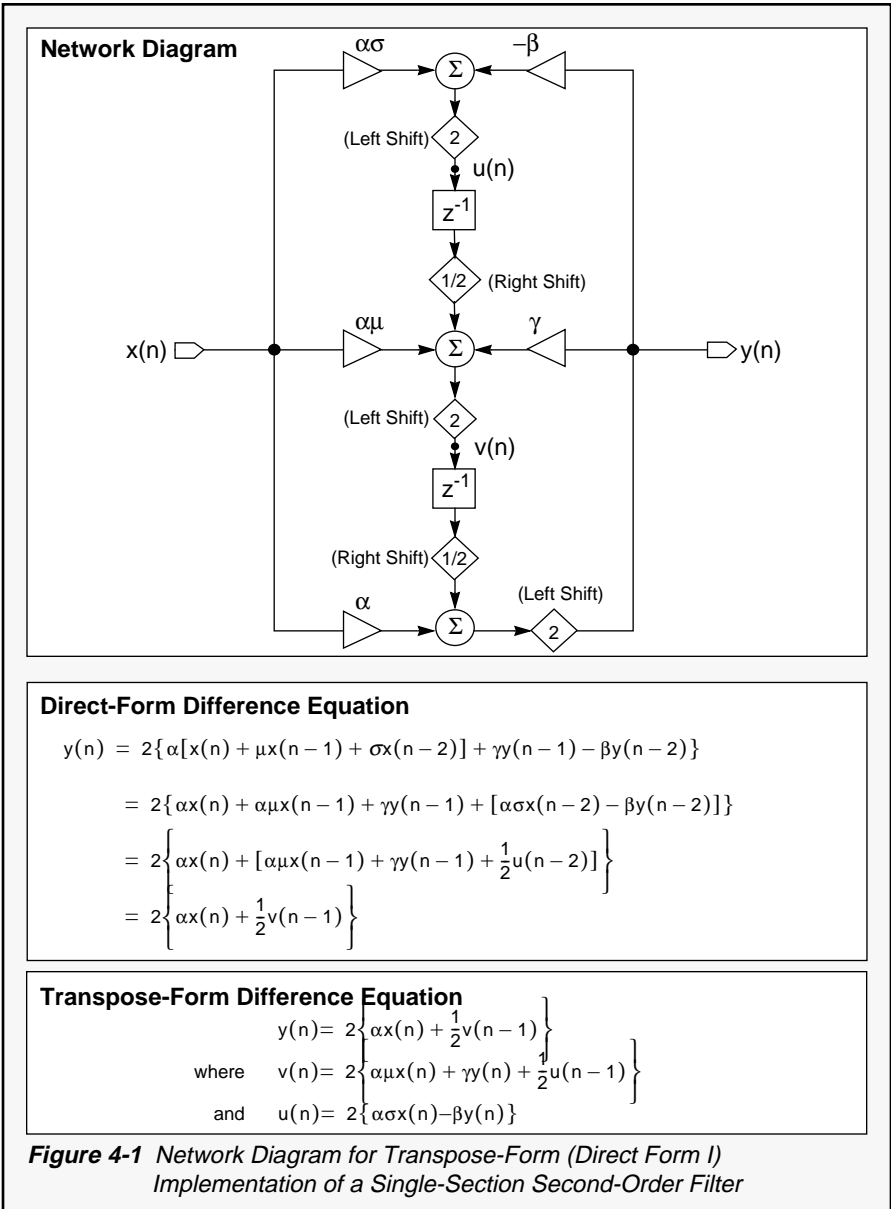
GAIN AT INTERNAL NODE, u(n)

Transfer function of u(n) is $\quad U(z) = 2[\alpha\sigma X(z) - \beta Y(z)]$

$$= H_u(z)X(z)$$

where $\quad H_u(z) = 2[\alpha\sigma - \beta H(z)]$

$$= 2\left[\frac{\alpha\sigma(1/2 - \gamma z^{-1} + \beta z^{-2}) - \alpha\beta(1 + \mu z^{-1} + \sigma z^{-2})}{1/2 - \gamma z^{-1} + \beta z^{-2}}\right]$$

$$= 2\left[\frac{2\alpha(A - Bz^{-1})}{1/2 - \gamma z^{-1} + \beta z^{-2}}\right]$$

where $H_u(z)$ is the transfer function from input node, x(n), to internal mode, u(n), and $A = \sigma/2 - \beta$ and $B = \sigma\gamma + \mu\beta$.

Gain of u(n) is $\qquad G_u(\theta) = \left|H_u(z)\right|_{z = e^{j\theta}}$

$$= \frac{2\alpha\sqrt{A^2 + B^2 - 2AB\cos\theta}}{\sqrt{(1/2 - \beta)^2 \sin^2\theta + (1/2 + \beta)^2(\cos\theta - \cos\theta_0)^2}}$$

BANDPASS EXAMPLE

Bandpass Coefficients are $\quad \sigma = -1 \qquad \mu = 0 \qquad \alpha = (1/2 - \beta)/2$

so that $\quad A = -(1/2 + \beta)$

$$B = -\gamma$$

$$= -(1/2 + \beta)\cos\theta_0$$

and $\quad G_u(\theta) = \dfrac{(1/2 - \beta)(1/2 + \beta)\sqrt{\sin^2\theta + (\cos\theta - \cos\theta_0)^2}}{\sqrt{(1/2 - \beta)^2 \sin^2\theta + (1/2 + \beta)^2(\cos\theta - \cos\theta_0)^2}}$

Peak Gain is $\qquad g_m = G_u(\theta_m)$

which is found by evaluating $\quad \dfrac{d}{d\theta}G_u(\theta)\big|_{\theta = \theta_m} = 0$

after evaluating derivative, $\quad \theta_m = \theta_0$

so that $g_m = \dfrac{(1/2 - \beta)(1/2 + \beta)\sin\theta_0}{(1/2 - \beta\sin\theta_0)} = 1/2 + \beta < 1$

Figure 4-2 *Gain Evaluation at First Internal Node, u(n), of Transpose Network*

GAIN AT INTERNAL NODE, v(n)

Transfer function of v(n) is
$$V(z) = [Y(z) - 2\alpha X(z)]/z^{-1}$$
$$= H_v(z)X(z)$$

where $\quad Y(z) = 2[\frac{1}{2}V(z)z^{-1} + \alpha X(z)] \qquad H_v(z) = \dfrac{H(z) - 2\alpha}{z^{-1}}$

$$\text{and} \quad = \frac{\alpha[(1 + \mu z^{-1} + \sigma z^{-2}) - 2(1/2 + \gamma z^{-1} + \beta z^{-2})]}{z^{-1}(1/2 - \gamma z^{-1} + \beta z^{-2})}$$

$$= \frac{2\alpha(C + Az^{-1})}{1/2 - \gamma z^{-1} + \beta z^{-2}}$$

where Hv(z) is the transfer function from input node, x(n), to internal mode, v(n), and
A=σ/2-b and C = μ/2 + g.

Gain of v(n) is
$$G_v(\theta) = \left| H_v(z) \right|_{z = e^{j\theta}}$$

$$= \frac{2\alpha\sqrt{A^2 + C^2 + 2AC\cos\theta}}{\sqrt{(1/2 - \beta)^2 \sin^2\theta + (1/2 + \beta)^2(\cos\theta - \cos\theta_0)^2}}$$

BANDPASS EXAMPLE

Bandpass Coefficients are $\qquad \sigma = -1 \qquad \mu = 0 \qquad \alpha = (1/2 - \beta)/2$

$$\text{so that} \quad A = -(1/2 + \beta)$$
$$C = \gamma$$
$$= (1/2 + \beta)\cos\theta_0$$

$$\text{and} \quad G_v(\theta) = \frac{(1/2 - \beta)(1/2 + \beta)\sqrt{\sin^2\theta + (\cos\theta - \cos\theta_0)^2}}{\sqrt{(1/2 - \beta)^2 \sin^2\theta + (1/2 + \beta)^2(\cos\theta - \cos\theta_0)^2}}$$

Peak Gain is $\qquad g_m = G_v(\theta_m)$

which is found by evaluating $\quad \dfrac{d}{d\theta}G_v(\theta)|_{\theta = \theta_m} = 0$

after evaluating derivative, $\quad \theta_m = \theta_0$

$$\text{so that} \quad g_m = \frac{(1/2 - \beta)(1/2 + \beta)\sin\theta_0}{(1/2 - \beta)\sin\theta_0} = 1/2 + \beta < 1$$

(since β< 1/2

Figure 4-3 *Gain Evaluation at Second Internal Node, v(n), of Transpose Network*
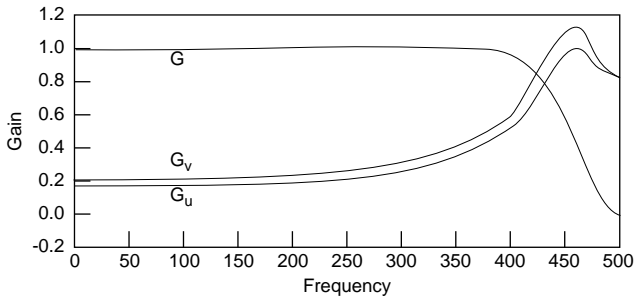
(a) f_c = 450 Hz and d=√2
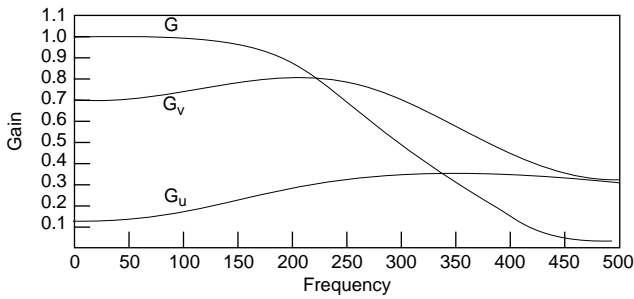
(b) f_c = 250 Hz and d=√2
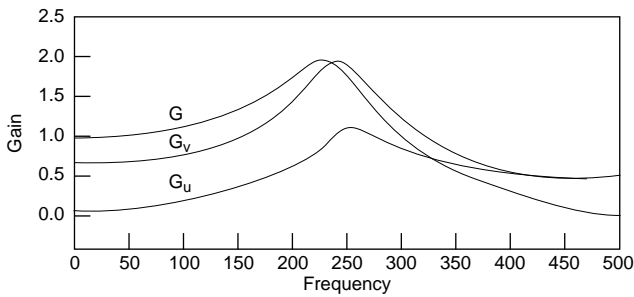
(c) f_c = 250 Hz and d= 1/2

Figure 4-4 *Total Gain and Gain at Internal Nodes of Lowpass Transpose Filter Network Figure 4-1 for f_s = 1000Hz*

Difference Equation

$$y(n) = 2\left\{\alpha x(n) + \frac{1}{2}v(n-1)\right\}$$

$$v(n) = 2\left\{\alpha\mu x(n) + \gamma y(n) + \frac{1}{2}u(n-1)\right\}$$

$$u(n) = 2\{\alpha\sigma x(n) - \beta y(n-1)\}$$

Data Structures

X:(R0) | v(n-1)

X:(R1) | u(n-1)

Y:(R4)

| $\alpha\mu$ |
| γ |
| $\alpha\sigma$ |
| $-\beta$ |
| α |
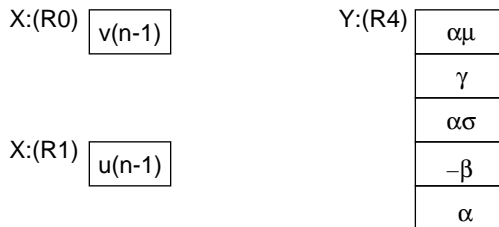
DSP56001 Code

```
                                  ;Y1=x(n) (Input)
                                  ;Y0=α . A=v(n-1)/2
MACR Y0,Y1,A X:(R1),B  Y:(R4)+,Y0 ;A=A+αx(n)
ASR  B       A,X0                 ;y(n)=X0=2A (scale mode on)
MAC  Y0,Y1,B           Y:(R4)+,Y0 ;B=u(n-1)/2+αμx(n)
MACR X0,Y0,B           Y:(R4)+,Y0 ;B=B+γy(n)
MPY  Y0,Y1,B B,X:(R0)  Y:(R4)+,Y0 ;y(n)=2B. B=ασx(n)
MACR X0,Y0,B X:(R0),A             ;B=B- βy(n)
ASR  A       B,X:(R1)  Y:(R4)+,Y0 ;A=v(n)/2. u(n)=2B
                                  ;Y0=α. y(n)=X0 (output)
```

Total Instruction Cycles

7 Icyc @ 20 MHz = 700ns

Figure 4-5 *DSP56001 Code and Data Structures for Single-Section Second-Order Transpose Form*

Cascaded Direct Form

By placing any of the direct-form second-order filter networks from Figure 2-10, Figure 2-13, Figure 2-16, and Figure 2-19 in series (i.e., connect the y(n) of one to the x(n) of the next), a cascaded filter is created. The resulting order N of the network is two times the number of second-order sections. An odd-order net-work can be made simply by adding one first-order section in the chain. In general, to achieve a particular response, the filter parameters associated with each second-order section are different, since generating a predefined total response requires that each section has a different response. This fact becomes more ob-vious when discussing the special case of Butterworth lowpass filters.

5.16 Butterworth Lowpass Filter

The Butterworth filter response is maximally flat in the passband at the expense of phase linearity and steep-ness of attenuation slope in the transition band. For lowpass or highpass cascaded second-order sections, all sections have the same center frequency (not the case for bandpass filters). For this reason, it is easy to design since all that remains to be determined are the

damping factors, dk, of each individual kth section. The damping coefficients, dk, are calculated from a simple formula for any order N of response.

The s-domain transfer function for the Nth-order lowpass Butterworth filter is:

$$H(s) = \left(\frac{1}{(s/\Omega_c)^2 + d_1(s/\Omega_c) + 1}\right)\left(\frac{1}{(s/\Omega_c)^2 + d_{2(1)}(s/\Omega_c) + 1}\right)\cdots$$

$$= \prod_{k=1}^{N/2} \frac{1}{(s/\Omega_c)^2 + d_k(s/\Omega_c) + 1} \qquad \text{Eqn. 5-7}$$

where: d_k is the kth damping coefficient,

s $= j\Omega$

Ω_c is the common cutoff frequency (see Reference 14).

Only filter orders of even N will be considered in this discussion to minimize the complexity of mathematical results. The analysis can be extended to include odd values of N by inserting an additional term of $[(s/\Omega_c) + 1]^{-1}$ in Eqn. 5-7. Eqn. 5-7 can be generalized if Ω_c Of each second-order section is an arbitrary value (corresponding to a different cutoff frequency, Ω_k, of each section). However, since this discussion is limited to Butterworth polynomials, Eqn. 5-7 will serve as the basis of all following derivations. A filter of order N has N/2 second-order

sections. The second-order section of a Butterworth filter can be derived from the simple RCL network of Figure 1-1. However, the Butterworth damping factors are predetermined values, which can be shown to yield a maximally flat passband response (see Reference 14). The Butterworth damping coefficients are given by the following equation:

$$d_k = 2\sin\frac{(2k-1)\pi}{2N} \qquad\qquad \text{Eqn. 5-8}$$

Eqn. 5-8 is the characteristic equation that determines a Butterworth filter response. Note that for a single-section ($k = 1$) second-order ($N = 2$) lowpass filter, $d_k = 2 \sin (\pi/4) = \sqrt{2}$ as expected for a maximally flat response. For a fourth-order filter with two second-order sections, $d_1 = 2 \sin (\pi/8)$, where $k = 1$ and $N = 4$, and $d_2 = 2 \sin (3\pi/8)$, where $k = 2$ and $N = 4$.

Eqn. 5-7 represents an all-pole response (the only zeros are at plus and minus infinity in the analog s-domain). The poles of a second-order section are the roots of the quadratic denominator as given by:

$$p_{k1} = -d_k/2 - j\left(1 - d_k^2/4\right)^{1/2} \qquad\qquad \text{(a)}$$

$$\text{and}$$

$$p_{k2} = d_k/2 + j\left(1 - d_k^2/4\right)^{1/2} \qquad\qquad \text{(b)}$$

$$\text{Eqn. 5-9}$$

Using , Eqn. 5-7 becomes:

$$H(s) = \prod_{k=1}^{N/2} \frac{1}{[(s/\Omega_c)-p_k][(s/\Omega_c)-p_k^*]} \qquad \text{Eqn. 5-10}$$

where: $p_k = p_{k1}$ and
$p_k^* = p_{k2}$ (complex conjugate of p_k)

Eqn. 5-10 is useful in that the response of the system can be analyzed entirely by studying the poles of the polynomial. However, for purposes of transforming to the z-domain, Eqn. 5-7 can be used as previously shown in Figure 2-10.

To examine the gain and phase response (physically measurable quantities) of the lowpass Butterworth filter, the transfer function, H(s), of Eqn. 5-7 will be converted into a polar representation. The magnitude of H(s) is the gain, $G(\Omega)$; the angle between the real and imaginary components of H(s), $\phi(\Omega)$, is the arctangent of the phase shift introduced by the filter:

$$G(\Omega) = \sqrt{H(s)H^*(s)|_{s=j\Omega}}$$

$$= \prod_{k=1}^{N/2} \frac{1}{\sqrt{\left[(\Omega/\Omega_c)^2 - 1\right]^2 + (d_k\Omega/\Omega_c)^2}} \qquad \text{Eqn. 5-11}$$

$$\phi(\Omega) \; = \; \sum_{k=1}^{N/2} \tan^{-1} \frac{d_k \Omega/\Omega_c}{(\Omega/\Omega_c)^2 - 1} \qquad \text{Eqn. 5-12}$$

Eqn. 5-11, and Eqn. 5-12 describe the response characteristics of an Nth-order lowpass Butterworth filter in the continuous frequency analog domain.Since the quantity of interest is usually 20 log $G(\Omega)$, Eqn. 5-11 can be transformed into a sum (over the second-order sections) of 20 log (G_k), where G_k is the gain of the kth section. Similarly, the total phase is just the sum of the phase contribution by each section from Eqn. 5-12.

The bilinear transformation is used to convert the continuous frequency domain transfer function into the digital domain representation, where θ, the normalized digital domain frequency equal to $2\pi f/f_s$, can be thought of as the ratio of frequency to sampling frequency scaled by 2π. Substituting s from Eqn. 2-13 and Ω_c from Eqn. 2-14 into the kth section of Eqn. 5-7 yields the digital domain form of the Butterworth lowpass filter (for the kth second-order section):

$$H_k(z) \; = \; \frac{\alpha_k(1 + 2z^{-1} + z^{-2})}{\frac{1}{2} - \gamma_k z^{-1} + \beta_k z^{-2}} \qquad \text{Eqn. 5-13}$$

$$\alpha_k = \left[\tan^2(\theta_c/2)\right]/A_k(c) \tag{a}$$

$$\beta_k = \left[1 - d_k\tan(\theta_c/2) + \tan^2(\theta_c/2)\right]/A_k(c) \tag{b}$$

$$\gamma_k = 2\left[1 - \tan^2(\theta_c/2)\right]/A_k(\theta c) \tag{c}$$

$$A_k(\theta_c) = 2\left[1 + d_k\tan(\theta_c/2) + \tan^2(\theta_c/2)\right] \tag{d}$$

Eqn. 5-14

(a)-(d) provides a complete description of the digital lowpass N^{th}-order Butterworth filter. Given θ_c and d_k, these formulas allow precise calculation of the digital coefficients, α_k, β_k, and γ_k, used to implement each kth second-order section of the filter.

(a)-(d) can be further simplified into the following set of formulas:

$$\beta_k = \frac{1 - (d_k/2)\sin(\theta_c)}{2[1 + (d_k/2)\sin(\theta_c)]} \tag{a}$$

$$\gamma_k = (1/2 + \beta_k)\cos(\theta_c) \tag{b}$$

$$\alpha_k = (1/2 + \beta_k - \gamma_k)/4 \tag{c}$$

Eqn. 5-15

where: d_k is given by Eqn. 5-8

θ_c is the digital domain cutoff frequency (actual operating cutoff frequency of the digital filter)
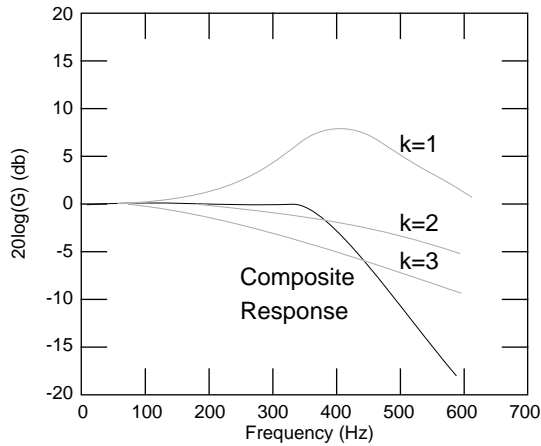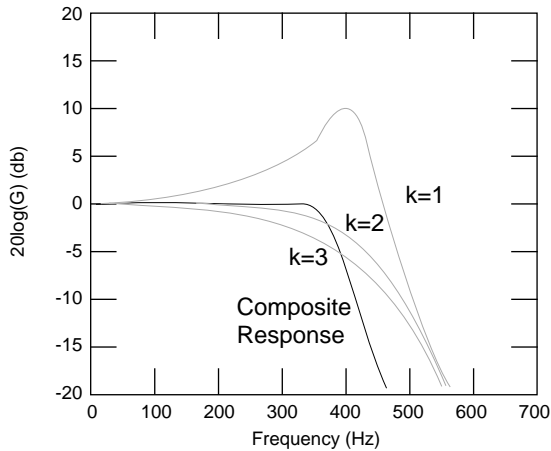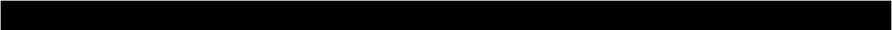
Figure 5-28 *Composite Response of Cascaded Second-Order Sections for Lowpass Butterworth Filter (k^{th} Section Damping Factor According to Eqn. 5-8)*

Figure 5-28 shows an example of a sixth-order low-

pass Butterworth filter (three second-order sections) in both the analog domain and digital domain. Note that the gain of the first section (k = 1) is greater than unity near the cutoff frequency but that the overall composite response never exceeds unity. This fact allows for easy implementation of the Butterworth filter in cascaded direct form (i.e., scaling of sections is not needed as long as the sections are implemented in the order of decreasing k). Overflow at the output of any section is then guaranteed not to occur (the gain of the filter never exceeds unity). Note that the digital response (see Figure 5-28) is identical to the analog response but warped from the right along the frequency axis. Imagine the zero at plus infinity in the analog response mapping into the zero at $f_s/2$ in the digital case. Also note that, because of this mapping, the digital response falls off faster than the -12 dB/octave of the analog filter when the cutoff is near $f_s/2$.

The previous analysis is nearly identical to the case of the highpass filter except the coefficients (see Figure 3-24) have slightly different values. Since the bandstop case is just the sum of a lowpass and highpass case, it can be analyzed by these techniques. The bandpass case, however, is more difficult and requires considerably more work (see Reference 14) because the center frequency of each section is now different and the formula for calculating these frequencies is not as simple as the formulas for the previous filter types. In addition, to complicate matters further, scaling between sections becomes more of a problem since the off-

set of the center of each section reduces the final center response, which must be compensated at some point in the filter network. For cases such as higher order Butterworth bandpass filter designs, commercially available filter design packages such as FDAS are useful. The use of FDAS is discussed in **SECTION 6 Filter Design And Analysis System (FDAS)** and **SECTION 7 Fir Filters.**

5.17 Cascaded Direct-Form Network

Figure 5-29 shows the cascaded direct-form network and data structures for the DSP56001 code implementation of Figure 5-30. By cascading network diagrams presented in **SECTION 2 Second-Order Direct-Form IIR Digital Filter Sections**, the set of delays at the output of one section can be combined with the set of delays at the input of the next section, thus reducing the total number of delays by almost a factor of two. For this reason, the cascaded direct-form network becomes canonic as the filter order N increases. The DSP56001 code (see Figure 5-30) shows an example of reading data from a user-supplied memory-mapped analog-to-digital converter (ADC) and writing it to a memory-mapped digital-to-analog converter (DAC). The number of sections (nsec) in these examples is three; thus, the filter order is six. The total instruction time for this filter structure is 600 nsec + 800 ns, including the data I/O moves (but excluding the interrupt overhead).

Network Diagram

$x(n)$ α_1 2 α_2 2 α_3 2 $y(n)$

z^{-1} $\alpha_1\mu_1$ γ_1 $\alpha_2\mu_2$ γ_2 $\alpha_3\mu_3$ γ_3

z^{-1} $\alpha_1\sigma_1$ $-\beta_1$ $\alpha_2\sigma_2$ $-\beta_2$ $\alpha_3\sigma_3$ $-\beta_3$
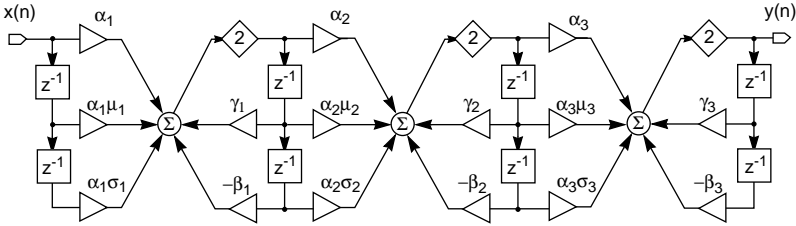
Difference Equations

$$y_1(n) = 2\{\alpha_i[x_i(n) + \mu_i x_i(n+1) + \sigma_i x_i(n-2)] + \gamma_i x_i(n-1) - \beta_i y_i(n-2)\}$$
$$x_{i+1}(n-k) = y_i(n-k)$$

Data Structures

coeff   xbuf

X:(R0)

| $-\beta_1$ |
| α_1 |
| $\alpha_1\sigma_1$ |
| $\alpha_1\mu_1$ |
| γ_1 |
| $-\beta_2$ |
| α_2 |
| $\alpha_2\sigma_2$ |
| $\alpha_2\mu_2$ |
| γ_2 |
| $-\beta_3$ |
| α_3 |
| $\alpha_3\sigma_3$ |
| $\alpha_3\mu_3$ |
| γ_3 |

Y:(R4)

$$
\begin{array}{l}
x_1(n-2) \\
x_1(n-1) \\
x_2(n-2) \\
x_2(n-1) \\
x_3(n-2) \\
x_3(n-1) \\
y(n-2) \\
y(n-1)
\end{array}
=
\begin{array}{l}
x(n-2) \\
x(n-1) \\
y_1(n-2) \\
y_1(n-1) \\
y_2(n-2) \\
y_2(n-1) \\
y_3(n-2) \\
y_3(n-1)
\end{array}
$$

Figure 5-29 *Network Diagram for Cascaded Direct-Form Filter and Data Structures Used in Code Implementation (Three-Section Example)*

DSP56001 CODE

```
    MOVE #coeff, R0                             ;Pointer to coefficients
    MOVE #1,M4                                  ;Modulo of length 2 for xbuf
    MOVE          X:(R0)+, X1                   ;β₁
    MOVE          X:(R0)+, X0                   ;α₁
    MOVE #2,N4
    MOVE          (R4)+                         ;Point to next xbuf entry
                                                ;Input:Y1=x(n)
    DO   X: nsec, Sectn                         ;Loop on number of sections
    MPY  X0,Y1,A  X:(R0)+,X0   Y:(R4)+,Y0       ;A=αᵢxᵢ(n)
    MAC  X0,Y0,A  X:(R0)+,X0   Y:(R4)+N4,Y0     ;A=A+αᵢσᵢxᵢ(n-2)
    MAC  X0,Y0,A  X:(R0)+,X0   Y:(R4)+,Y0       ;A=A+αᵢμᵢxᵢ(n-1)
    MAC  X0,Y0,A              Y:(R4)-N4,Y0      ;A=A+γᵢyᵢ(n-1)
    MAC  -X1,Y0,A X:(R0)+,X1  Y1,Y:(R4)+N4      ;A=A-βᵢyᵢ(n-2). Save x(n)
    MOVE A,Y1     X:(R0)+,X0                    ;yᵢ(n)=2A, assumes scale mode on

Sectn                                          ;X1=βᵢ₊₁. X0=αᵢ₊₁

                                               ;Output:y(n)=Y1
    MOVE          X:Buflen, M4                  ;Filter Order +1
    NOP
    MOVE                     Y1,Y:(R4)+N4       ;Save y(n)
```

Total Instruction Cycles

(6*nsec+10)Icyc @ 20 MHz = (0.6*nsec+1.0)ns

NOTE: nsec is number of sections.

Figure 5-30 *DSP56001 Code for Cascaded Direct-Form Filter*

SECTION 6

Filter Design and Analysis System

"The filter example used is a sixth-order Butterworth lowpass filter with a cutoff frequency of approximately 225 Hz and a sample frequency of 1000 Hz."
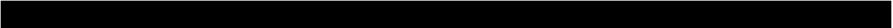
The following paragraphs discuss the design of a cascaded filter in both the direct-form and canonic implementations, using a software package, QEDesign (formerly FDAS[1]), available from Momentum Data Systems, Inc. The filter example used is a sixth-order Butterworth lowpass filter with a cutoff frequency of approximately 225 Hz and a sample frequency of 1000 Hz. Figure 6-6 is the log magnitude (gain) plot from the system output. Figure 6-7 is the phase as a function of frequency in wrapped format ($-\pi$ wraps to $+\pi$). In addition, Figure 6-8 is a zero/pole plot, and Figure 6-9 is the group delay, which is the negative of the derivative of the phase with respect to frequency.

FDAS will also generate an impulse response, step response, and a linear magnitude plot. The results of the design are written to a file, FDAS.OUT, which contains much useful information. The coefficient data is written to COEFF.FLT. The DSP56001 code generator (MGEN) reads the COEFF.FLT file and generates a DSP56001 assembly source file, COEFF.ASM, which

---

1. All references to FDAS in this application note refer to the software package QEDesign.

---

can be assembled by the DSP56001 assembler or linker software. Examples of these files are shown in Figure 6-10 to Figure 6-10.

6.18 Canonic Implementation

Figure 6-10 is the output file associated with the FDAS design session, containing information on the analog s-domain equivalent filter as well as the final digital coefficients (listed again in Figure 6-6), which have been properly scaled to prevent overflow at the internal nodes and outputs of each cascaded section. This procedure is done automatically by the program in a matter of seconds. Executable code is generated by MGEN (also from Momentum Data Systems, Inc.), as shown in Figure 6-7. The code internal to each cascaded section is five instructions long; thus, 500 ns (assuming a DSP56001 clock of 20 MHz) is added to the execution time for each additional second-order section.

Passband Ripple in -dB   -.5000
Stopband Ripple in -dB   -20.0000
Passband Cutoff Frequency   200.000
Stopband Cutoff Frequency   300.000
Sampling Frequency   1000.00
All Frequencies in Hertz
Quantization 24 Bits Fixed Point Fractional

Lowpass Filter
IIR Design-Bilinear Transformation
Butterworth
Filter Order: 6

Filter Design & Analysis System
Momentum Data Systems, Inc.
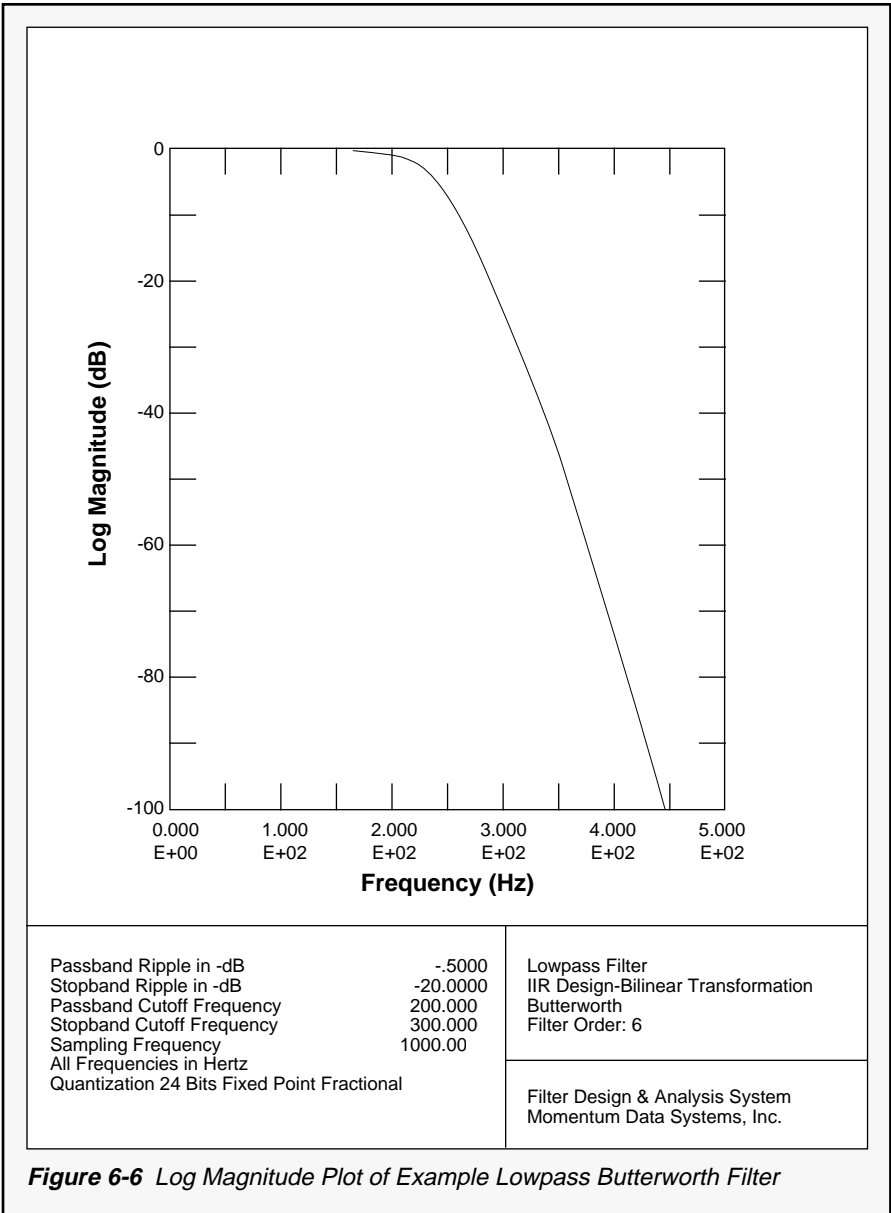
Figure 6-6 *Log Magnitude Plot of Example Lowpass Butterworth Filter*
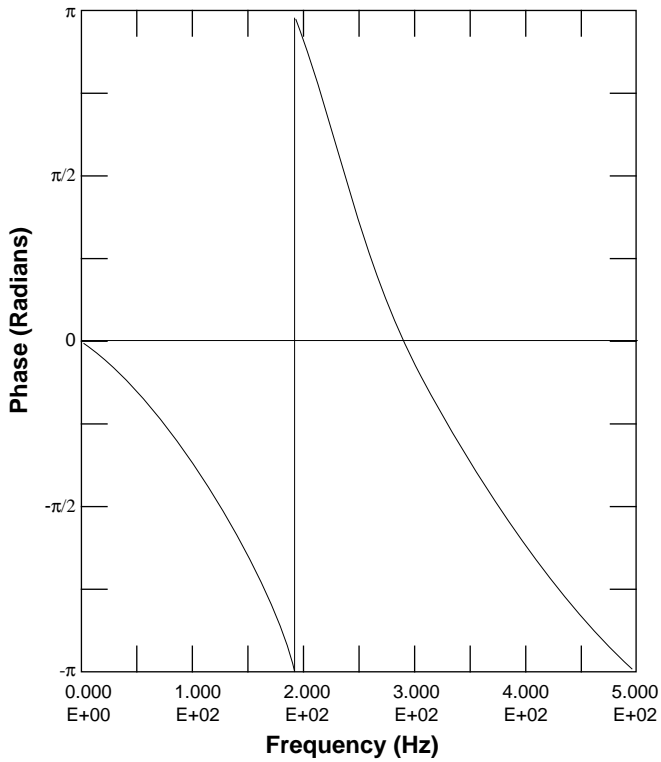
Phase (Radians)

π

π/2

0

-π/2

-π

0.000
E+00

1.000
E+02

2.000
E+02

3.000
E+02

4.000
E+02

5.000
E+02

Frequency (Hz)

Passband Ripple in -dB -.5000 Stopband Ripple in -dB -20.0000 Passband Cutoff Frequency 200.000 Stopband Cutoff Frequency 300.000 Sampling Frequency 1000.00 All Frequencies in Hertz Quantization 24 Bits Fixed Point Fractional	Lowpass Filter IIR Design-Bilinear Transformation Butterworth Filter Order: 6
	Filter Design & Analysis System Momentum Data Systems, Inc.

Figure 6-7 *Phase Versus Frequency Plot for Example Filter*

Passband Ripple in -dB -.5000	Lowpass Filter
Stopband Ripple in -dB -20.0000	IIR Design-Bilinear Transformation
Passband Cutoff Frequency 200.000	Butterworth
Stopband Cutoff Frequency 300.000	Filter Order: 6
Sampling Frequency 1000.00	
All Frequencies in Hertz	
Quantization 24 Bits Fixed Point Fractional	Filter Design & Analysis System
	Momentum Data Systems, Inc.

Figure 6-8 *Zero/Pole Plot of Sixth-Order Lowpass Example Filter*

Figure panel contains:

Y-axis (Seconds):
- 6.3 E-03
- 4.7 E-03
- 3.1 E-03
- 1.5 E-03
- 0.0

X-axis (Frequency (Hz)):
- 0.000 E+00
- 1.000 E+02
- 2.000 E+02
- 3.000 E+02
- 4.000 E+02
- 5.000 E+02

Passband Ripple in -dB -.5000	Lowpass Filter
Stopband Ripple in -dB -20.0000	IIR Design-Bilinear Transformation
Passband Cutoff Frequency 200.000	Butterworth
Stopband Cutoff Frequency 300.000	Filter Order: 6
Sampling Frequency 1000.00	
All Frequencies in Hertz	
Quantization 24 Bits Fixed Point Fractional	Filter Design & Analysis System
	Momentum Data Systems, Inc.

Figure 6-9 *Group Delay Versus Frequency for FDAS IIR Example*

```
Filter  Type                          Low Pass
Analog  Filter  Type                  Butterworth
Passband  Ripple In -dB                  -.5000
Stopband  Ripple In -dB                -20.0000
Passband  Cutoff  Frequency           200.000    Hertz
Stopband  Cutoff  Frequency           300.000    Hertz
Sampling   Frequency                  1000.00    Hertz
Filter Order: 6
Filter  Design  Method: Bilinear Transformation

Coefficients of Hd(z) are Quantized to 24 bits
Quantization Type: Fixed Point Fractional
Coefficients Scaled For Cascade Form II



             NORMALIZED  ANALOG TRANSFER FUNCTION  T(s)

   Numerator  Coefficients                    Denominator Coefficients
***********************************  **********************************
  S**2 TERM        S TERM      CONST TERM   S**2 TERM    S TERM     CONST TERM
.000000E+00     .000000E+00  .100000E+01  .100000E+01.193185E+01 .100000E+01
.000000E+00     .000000E+00  .100000E+01  .100000E+01.141421E+01 .100000E+01
.000000E+00     .000000E+00  .100000E+01  .100000E+01.517638E+01 .100000E+01
     INITIAL GAIN  1.00000000



             UNNORMALIZED  ANALOG  TRANSFER  FUNCTION  T(s)

   Numerator   Coefficients                   Denominator Coefficients
*************************************** **********************************
  S**2 TERM        S TERM      CONST TERM  S**2 TERM   S TERM       CONST TERM
.000000E+00     .000000E+00  .299809E+07  .100000E+01.334500E+04 .299809E+07
.000000E+00     .000000E+00  .299809E+07  .100000E+01.244871E+04 .299809E+07
.000000E+00     .000000E+00  .299809E+07  .100000E+01.896290E+03 .299809E+07
     INITIAL GAIN  1.00000000



             DIGITAL   TRANSFER   FUNCTION  Hd(z)

   Numerator  Coefficients                    Denominator Coefficients
***************************************  *********************************
  Z**2           Z TERM      CONST TERM  Z**2      Z TERM       CONST TERM
.2520353794     .5040707588  .252035379  1.00000  -.1463916302  .0225079060
.2364231348     .4728463888  .236423134  1.00000  -.1684520245  .1765935421
.3606387377     .7212774754  .360638737  1.00000  -.2279487848  .5921629667
     INITIAL GAIN .876116210
```

Figure 6-10 *FDAS.OUT File of Example Filter for Cascaded
            Canonic Implementation*   *(sheet 1 of 2)*

```
                    ZEROES   OF   TRANSFER   FUNCTION Hd(z)

  Real Part      Imaginary Part  Real Part     Imaginary Part   Radius
-.100000000E+01 .00000000E+00  -.100000000E+01  .00000000E+00  .100000000E+01
-.999290168E+00 .00000000E+00  -.100071034E+01  .00000000E+00  .999290168E+00
-.100000000E+01 .00000000E+00  -.100000000E+01  .00000000E+00  .100000000E+01


                    POLES   OF   TRANSFER   FUNCTION Hd(z)

  Real Part      Imaginary Part  Real Part     Imaginary Part   Radius
.731958151E-01  .130959072E+00  .731958151E-01 -.130959072E+00  .150026351E+00
.842260122E-01  .411703195E+00  .842260122E-01 -.411703195E+00  .420230344E+00
.113974392E+00  .761034036E+00  .113974392E+00 -.761034036E+00  .769521258E+00


       NUMERATOR   COEFFICIENTS - HIGHEST   ORDER FIRST (in  z)

.214893785E-01  .128936282E+00  .322340720E+00   .429787634E+00  .322340720E+00
.128936282E+00  .214893785E-01


       DENOMINATOR   COEFFICIENTS - HIGHEST   ORDER FIRST (in  z)


 .100000000E+01  -.542792439E+00  .887692610E+00 .267088214E+00 .143235132E+0
-.184597152E-01   .235370025E-02


IMPULSE RESPONSE MIN = -.179722E+00, MAX = .452734E+00
```

Figure 6-5 *FDAS.OUT File of Example Filter for Cascaded*
Canonic Implementation *(sheet 2 of 2)*

```
FILTER COEFFICIENT FILE
IIR DESIGN
FILTER  TYPE              LOW PASS
ANALOG  FILTER TYPE       BUTTERWORTH
PASSBAND RIPPLE IN -dB        -.5000
STOPBAND RIPPLE IN -dB      -20.0000
PASSBAND CUTOFF FREQUENCY  .200000E+03 HERTZ
STOPAND CUTOFF FREOUENCY   .300000E+03 HERTZ
SAMPLING FREQUENCY         .100000E+04 HERTZ
FILTER DESIGN METHOD: BILINEAR TRANSFORMATION
FILTER ORDER          6 0006h
NUMBER OF SECTIONS       3 0003h
NO. OF QUANTIZED BITS  24 0018h
QUANTIZATION TYPE - FRACTIONAL FIXED POINT
COEFFICIENTS SCALED FOR CASCADE FORM 11
        0 00000000   /* shift count for overall gain         */
  7349395 00702493   /* overall  gain                        */
        0 00000000   /* shift count for section 1 value      */
  2114226 002042B2   /* section1 coefficient B0              */
  4228452 00408564   /* section1 coefficient B1              */
  2114226 002042B2   /* section1 coefficient B2              */
  1228022 0012BCF6   /* section1 coefficient A1              */
  -188810 FFFD1E76   /* section1 coefficient A2              */
        0 00000000   /* shift count for section 2 values     */
  1983261 001E431D   /* section2 coefficient B0              */
  3966523 003C863B   /* section2 coefficient B1              */
  1983261 001E431D   /* section2 coefficient B2              */
  1413078 00158FD6   /* section2 coefficient A1              */
 -1481374 FFE96562   /* section2 coefficient A2              */
        0 00000000   /* shift count for section 3 values     */
  3025257 002E2969   /* section3 coefficient B0              */
  6050514 005C52D2   /* section3 coefficient B1              */
  3025257 002E2969   /* section3 coefficient B2              */
  1912173 001D2D6D   /* section3 coefficient A1              */
 -4967423 FFB43401   /* section3 coefficient A2              */
.2520353794097900D+00 3FD0215900000000  .25203538E+00  /* section 1 B0  */
.504D707588195801D+00 3FE0215900000000  .50407076E+00  /* section 1 B1  */
.2520353794097900D+00 3FDD215900000000  .25203538E+00  /* section 1 B2  */
.1463916301727295D+00 3FC2BCF600000000  .14639171E+00  /* section 1 A1  */
-.2250790596008301D-01 BF970C5000000000 -.22507921E-01  /* section 1 A2  */
.23642313480377O2D+00 3FCE431D00000000  .23642325E+00  /* section 2 B0  */
.4728463888168335D+00 3FDE431D80000000  .47284651E+00  /* section 2 B1  */
.2364231348037720D+00 3FCE431D00000000  .23642325E+00  /* section 2 B2  */
.1684520244598389D+00 03FC58FD60000000  .168452D4E+D0  /* section 2 A1  */
.17659354Z0989990D+00 BFC69A9E000000D0 -.17659356E+00  /* section 2 A2  */
.3606387376785278D+00 3FD714B480000000  .36063876E+00  /* section 3 B0  */
.7212774753570557D+00 3FE714B480000000  .72127753E+00  /* section 3 Bl  */
.3606387376785278D+00 3FD714B480000000  .36063876E+00  /* section 3 B2  */
.2279487848281860D+00 3FCD2D6DD0000000  .22794882E*00  /* section 3 A1  */
-.5921629667282104D+00 BFE2F2FFC0000000 -.59216306E+00  /* section 3 A2  */
```

Figure 6-6 *COEFF.OUT File of Example Filter Design-Scaled for Cascaded Canonic Implementation*

```
1              COEFF    ident 1,0
2                       include 'head2.asm'
3
4                       page 132,66,            0,10
5                       pt      cex,            mex
6
7      ; This program implements an IIR filter in cascaded canonic sections
8      ; The coefficients of each section are scaled for cascaded canonic sections
9
10
11  00FFFF   datin   equ  $ffff                 ;location in Y memory of input file
12  00FFFF   datout  equ  $ffff                 ;location in Y memory of output file
13  00FFF0   m_scr   equ  $fff0                 ;sci control register
14  00FFF1   m_ssr   equ  $fff1                 ;sci status register
15  00FFF2   m_sccr  equ  $fff2                 ;sci clock control register
16  00FFE1   m_pcc   equ  $ffe1                 ;port c control register
17  00FFFF   m_ipr   equ  $ffff                 ;interrupt priority register
18  000140   xx      equ  @cvi(20480/64*1.000)) ;timer interrupt value
19  FFD8F1   m_tim   equ  -9999                 ;board timer interrupt value
20  000003   nsec    equ  3                     ;number of second order sectiona
21                   include 'cascade2.asm'
22          ;
23          ; This code segment implements cascaded biquad sections in canonic form
24          ;
25          ;
26          cascade2 macro nsec
27  m       ;
28  m       ;assumes each section's coefficients are divided by 2
29  m
30  m               mpy  y0,y1,a    x:(r0)+,y0   y:(r4)+,y0  ;x0=1st section w(n-2),y0=a12/2
31  m               do   #nsec,_ends                         ;do each section
32  m               mac  x0,y0,a    x:(r0)-,x1   y:(r4)+,y0  ;x1=w(n-1) ,y0=ai1/2
33  m               macr x1,y0,a    x1,x:(r0)+   y:(r4)+,y0  ;push w(n-2),y0=bi2/2
34  m               mac  x0,y0,a    a,x:(r0)     y:(r4)+,y0  ;push w(n) to w(n-1),y0=bi1/2
35  m               mac  x1,y0,a    x:(r0)+,x0   y:(r4)+,y0  ;get this iter w(n),y0=bi0/2
36  m               mac  x0,y0,a    x:(r0)+,x0   y:(r4)+,y0  ;next iter:x0=w(n-2),y0=ai2/2
37  m       _ends
38  m               endm
```

Figure 6-7 COEFF.ASM File Generated by MGEN for Example Design and Cascaded Canonic Implementation

(Sheet 1 of 4)

```
39
40
41        ;
42        ;multiple shift left macro
43
44        mshl    macro   scount
45 m              if      scount
46 m              rep     #scount
47 m              asl     a
48 m              endif
49 m              endm
50 X:0000         org     x:0
51 X:0000 states  ds      2*nsec
52 Y:0000         org     y:0
53 Y:0000 coef
54 d Y:0000 FE8F38        dc    $FE8F3B        ;a(*,2)/2       =-.01125395  section number 1
55 d Y:0001 095E7B        dc    $095E7B        ;a(*,1)/2       = .07315582  section number 1
56 d Y:0002 102159        dc    $102159        ;b(*,2)/2       = .12601769  section number 1
57 d Y:0003 2042B2        dc    $2042B2        ;b(*,1)/2       = .25203538  section number 1
58 d Y:0004 D02159        dc    $D02159        ;b(*,0)/2-0.5   =-.37398231  section number 1
59 d Y:0005 F4B2B1        dc    $F4B2B1        ;a(*,2)/2       =-.08829677  section number 2
60 d Y:0006 0AC7EB        dc    $0AC7EB        ;a(*,1)/2       = .08422601  section number 2
61 d Y:0007 0F218E        dc    $0F218E        ;b(*,2)/2       = .11821157  section number 2
62 d Y:0008 1F4313        dc    $1E431D        ;b(*,1)/2       = .23642319  section number 2
63 d Y:0009 CF218E        dc    $CF218E        ;b(*,0)/2-0.5   =-.38178843  section number 2
64 d Y:000A DA1A01        dc    $DA1A01        ;a(*,2)/2       =-.29608148  section number 3
65 d Y:000B 0E96B6        dc    $0E96B6        ;a(*,1)/2       = .11397439  section number 3
66 d Y:000c 1714B4        dc    $1714B4        ;b(*,2)/2       = .18031937  section number 3
67 d Y:000D 2E2969        dc    $2E2969        ;b(*,1)/2       = .36063874  section number 3
68 d Y:000E D714B4        dc    $0714B4        ;b(*,0)/2-0.5   =-.31968063  section number 3
69 Y:00c8               org   y:200
70 d Y:00C8 381249 igain   dc    3674697
71 000000       scount  equ   0                ;final shift count
72
73                      include 'bodyZ.asm'
74 000040       start   equ   $40              ;origin for user program
75
76 P:0000         org   p:$0                   ;origin for reset vector
```

Figure 6-7 *COEFF.ASM File Generated by MGEN for Example Design and Cascaded Canonic*
Implementation   *(Sheet 2 of 4)*

```
77    P:0000    0C0040    jmp       start                ;jump to 'start' on system reset
78                                                        ;origin for timer interrupt vector
79    P:001c              org       p:$1c
80    P:001c    0BF080    jsr       filter               ;jump to 'filter' on timer interrupt
                000051
81
82    P:0040              org       p:start              ;origin for user program
83
84    P:0040    0003F8    ori       #3,mr                ;disable all interrupts
85    P:0041    08F4B2    movep     #(xx-1),x:m_sccr     ;cd=xx-1 for divide by xx
                00013F
86    P:0043    08F4A1    movep     #$7,x:m_pcc          ;set cc(2:0) to turn on timer
                000007
87    P:0045    08F4B0    movep     #$2000,x:m_scr       ;enable timer interrupts
                002000
88    P:0047    08F4BF    movep     #$c000,x:m_ipr       ;set interrupt priority for sci
                00C000
89
90    P:0049    300000    move      #states,r0           ;initialize internal state storage
91    P:004A    200013    clr       a                    ;* set memory to zero
92    P:004B    0606A0    rep       #nsec*2              ;*
93    P:004C    565800    move      a,x:(r0)+            ;*
94
95    P:004D    4FF000    move      y:igain,yl           ;yl=initial gain/2
                0000C8
96    P:004F    00FCB8    andi      #$fc,mr              ;allow interrupts
97    P:0050    0C0050    jmp       *                    ;wait for interrupt
98
99              filter
100   P:0051    0008F8    ori       #$08,mr              ;set scaling node
101   P:0052    300000    move      #states,r0           ;point to filter states
102   P:0053    340000    move      #coef,r4             ;point to filter coefficients
103   P:0054    09463F    movep     y:datin,y0           ;get sample
104
105                       cascade2  nsec                 ;do cascaded biquads
106 +          ;
107 +          ;assumes each section's coefficients are divided by 2
108 +          ;
```

Figure 6-7 COEFF.ASM File Generated by MGEN for Example Design and Cascaded Canonic Implementation *(Sheet 3 of 4)*

```
109 +   P:0055  F098B0          mpy   y0,y1,a   x:(r0)+ x0    y:(r4)+,y0   ;x0=1stsection w(n-2),y0=ai2/2
110 +   P:0056  060380          do    #nsec,_ends                          ;do each section
                00005C
111 +   P:0058  F490D2          mac   x0,y0,a   x:(r0)- ,x1   y:(r4)+,y0   ;x1=w(n-1)y0=ai1/2
112 +   P:0059  F418E3          macr  x1,y0,a   x1,x:(r0)+    y:(r4)+,y0   ;push w(n-1) to w(n 2),y0=bi2/2
113 +   P:005A  F800D2          mac   x0,y0,a   a,x:(r0)      y:(r4)+,y0   ;push w(n) to w(n-1),y0=bi1/2
114 +   P:005B  F098E2          mac   x1,y0,a   x:(r0)+,x0    y:(r4)+,y0   ;get this iter w(n),y0=bi0/2
115 +   P:005C  F098D2          mac   x0,y0,a   x:(r0)+,x0    y:(r4)+,y0   ;next iter:x0=w(n-2),yo=ai2/2
116                      _ends
117                             mshl scount
118 +                          if    scount
                               endif
121 +   P:005D  200011          rnd   a                                    ;round result
122
123     P:005E  09CE3F          movep a,y:datout                           ;output sample
124     P:005F  000004          rti
125
126                             end
        0 Errors
        0 Warnings
```

Figure 6-7 *COEFF.ASM File Generated by MGEN for Example Design and Cascaded Canonic Implementation* *(Sheet 4 of 4)*

6.2 Transpose Implementation (Direct Form I)

Figure 6-8 is the output file from FDAS for a transpose-form implementation. As before, it contains information on the analog s-domain equivalent filter as well as the final digital coefficients (listed again in Figure 6-9), which have been properly scaled to prevent overflow at the internal nodes and outputs of each cascaded section. Because of the stability of the internal node gain of the transpose form and because of the response of each second-order Butterworth section, scaling was not done by the program because it was not needed. Executable code shown in Figure 6-10 is again generated by MGEN. The code internal to each cascaded section is seven instructions long; thus, 700 ns (assuming a DSP56001 clock of 20 MHz) is added to the execution time for each additional second-order section.

```
FILTER  TYPE                 LOW PASS
ANALOG  FILTER  TYPE         BUTTERWORTH
PASSBAND RIPPLE IN -dB          -.5000
STOPBAND RlPPLE lN -dB         -20.0000
PASSBAND  CUTOFF FREQUENCY      200.000   HERTZ
STOPBAUD  CUTOFF FREQUENCY      300.000   HERTZ
SAMPLING  FREQUENCY           1000.00    HERTZ
FILTER ORDER: 6
FILTER  DESIGN  METHOD: BILINEAR TRANSFORMATION
COEFFICIENTS OF Hd(Z) ARE QUANTIZED TO 24 BITS
QUANTIZATION TYPE: FIXED  POINT  FRACTIONAL
COEFFICIENTS SCALED FOR CASCADE FORM I (TRANSPOSE FORM)


            NORMALIZED  ANALOG  TRANSFER  FUNCTION   T(s)

    NUMERATOR COEFFICIENTS                DENOMINATOR COEFFICIENTS
 **********************************    **********************************
  S**2 TERM   S TERM    CONST TERM     S**2 TERM     S TERM    CONST TERM
.000000E+00 .000000E+00 .100000E+01   .100000E+01.193185E+01   .100000E+01
.000000E+00 .000000E+00 .100000E+01   .100000E+01.141421E+01   .100000E+01
.000000E+00 .000000E+00 .100000E+01   .100000E+01.517638E+00   .1000000E+01
    INITIAL GAIN 1.00000000



            UNNORMALIZED  ANALOG  TRANSFER  FUNCTION   T(s)

    NUMERATOR  COEFFICIENTS               DENOMINATOR COEFFICIENTS
 **********************************    **********************************
  S**2 TERM  S TERM     CONST  TERM    S**2 TERM   S TERM      CONST TERM
.000000E+00.000000E+00  .299809E+07    .100000E+01.334500E+04   .299809E+07
.000000E+00.000000E+00  .299809E+07    .100000E+01.244871E+04   .299809E+07
.000000E+00.000000E+00  .299809E+07    .100000E+01.896290E+03   .299809E+07
    INITIAL GAIN  1.00000000


            DIGITAL    TRANSFER  FUNCTION   Hd(z)
    NUMERATOR  COEFFICIENTS               DENOMINATOR  COEFFICIENTS
 **********************************    **********************************
  Z**2 TERM  Z TERM     CONST TERM    Z**2 TERM   Z TERM      CONST TERM
.2190289497.4380580187 .2190289497   1.000000 -.1463916302 .0225079060
.2520353794.5040707588 .2520353794   1.000000 -.1684520245 .1765935421
.3410534859.6821070910 .3410534859   1.000000 -.2279487848 .5921629667
    INITIAL GAIN 1.00000000


            ZEROES   OF  TRANSFER  FUNCTION  Hd(z)
  REAL PART     IMAGINARY  PART    REAL PART    IMAGINARY  PART    RADIUS
-.999262530E+00 .000000000E+00 -.100073501E+01 .000000000E+00 .999262530E+00
-.100000000E+01 .000000000E+00 -.100000000E+01 .000000000E+00 .100000000E+01
-.999408962E+00 .000000000E+00 -.100059139E+01 .000000000E+00 .999408962E+00
```

Figure 6-8 FDAS.OUT File of Example Filter for Cascaded Transpose–
Form Implementation   (sheet 1 of 2)

```
                POLES OF TRANSFER FUNCTION Hd(z)
  Real Part   Imaginary Part   Real Part    Imaginary Part    Radius
-.731958151E-01  -.130959072E+00  -.731958151E-01  -.130959072E+00  -.150026351E+00
-.842260122E-01  -.411703195E+00  -.842260122E-01  -.411703195E+00  -.420230344E+00
-.113974392E+00  -.761034036E+00  -.113974392E-00  -.761034036E+00  -.769521258E+00


      NUMERATOR  COEFFICIENTS - HIGHEST  ORDER  FIRST (in z)

.188271907E-01   .112963161E+00.282407928E+00    .376543916E+00  .282407928E+OO
.112963161E+00   .188271907E-01


     DENOMINATOR   COEFFICIENTS - HIGHEST  ORDER  FIRST (in z)

.10000000E+01   -.542792439E+00.887692610E+00    -.267088214E+00 .143235132E+00
.184597152E-01   .235370025E-02



IMPULSE RESPONSE  MIN = -.179722E+00, MAX =  .452734E+00
```

Figure 6-8 *FDAS.OUT File of Example Filter for Cascaded Transpose–*
            Form Implementation   *(sheet 2 of 2)*

```
FILTER COEFFICIENT FILE
IIR DESIGN
FILTER TYPE           LOW PASS
ANALOG FILTER TYPE      BUTTERWORTH
PASSBAND RIPPLE IN -dB        -.5000
STOPBAND RIPPLE IN -dB       -20.0000
PASSBAND CUTOFF FREQUENCY    .200000E+03 HERTZ
STOPBAND CUTOFF FREQUENCY    .300000E+03 HERTZ
SAMPLING FREQUENCY           .100000E+04 HERTZ
FILTER DESIGN METHOD: BILINEAR TRANSFORMATION
FILTER ORDER          6 0006h
NUMBER OF SECTIONS     3 0003h
NO. OF QUANTIZED BITS  24 0018h
QUANTIZATION TYPE - FRACTIONAL FIXED POINT
COEFFICIENTS SCALED FOR CASCADE FORM I
       1  00000001        /* shift count for overall gain        */
 4194304  00400000        /* overall gain                        */
       0  00000000        /* shift count for section 1 values    */
 1837348  001C0924        /* section 1 coefficient B0            */
 3674697  00381249        /* section 1 coefficient B1            */
 1837348  001C0924        /* section 1 coefficient B2            */
 1228022  00128CF6        /* section 1 coefficient A1            */
 -188810  FFFD1E76        /* section 1 coefficient A2            */
       0  00000000        /* shift count for section 2 values    */
 2114226  00204292        /* section 2 coefficient B0            */
 4228452  00408564        /* section 2 coefficient B1            */
 2114226  00204282        /* section 2 coefficient B2            */
 1413078  00158FD6        /* section 2 coefficient A1            */
-1481374  FFE96562        /* section 2 coefficient A2            */
       0  00000000        /* shift count for section 3 values    */
 2860964  002BA7A4        /* section 3 coefficient B0            */
 5721929  00574F49        /* section 3 coefficient B1            */
 2860964  002BA7A4        /* section 3 coefficient B2            */
 1912173  001D2D6D        /* section 3 coefficient Al            */
-4967423  FFB43401        /* section 3 coefficient A2            */
 .2190289497375488D+00  3FCC092400000000  .21902905E+00 /* section 1 B0  */
 .4380580186843872D+00  3FDC092480000000  .43805810E+00 /* section 1 B1  */
 .2190289497375488D+00  3FCC092400000000  .21902905E+00 /* section 1 B2  */
 .146391630172 n95D+00  3FC2BCF600000000  .14639171E+00 /* section 1 A1  */
-.2250790596008301D-01  BF970C5000000000 -.22507921E-01 /* section 1 A2  */
 .2520353794097900D+00  3FD0215900000000  .25203538E+00 /* section 2 B0  */
 .5040707588195801D+00  3FE0215900000000  .50407076E+00 /* section 2 Bl  */
 .2520353794097900D+00  3FD0215900000000  .25ZD3538E+00 /* section 2 B2  */
 .1684520244598389D+00  3FC58FD600000000  .16845204E+00 /* section 2 Al  */
-.1765935420989990D+00  BFC69A9E00000000 -.17659356E+00 /* section 2 A2  */
 .34105348587036130D+00 3FDSD3D200000000  .34105356E+00 /* section 3 B0  */
 .6821070909500122D+00  3FESD3DZ40000000  .68210712E+00 /* section 3 Bl  */
 .3410534858703613D+00  3FDSD3D200000000  .34105356E+00 /* section 3 B2  */
 .2279487848281860D+00  3FCD2D6D00000000  .22794882E+00 /* section 3 Al  */
-.5921629667282104D+00  BFE2F2FFC0000000 -.59216306E+00 /* section 3 A2  */
```

Figure 6-9 *COEFF.FLT File for Example Filer Design–Scaled for Cascaded Transpose Form*

```
                COEFF    ident 1,0
                         include 'headl.asm'

                         page132,66,0,10
                         optcex, mex
                ;
                ; This program implements an IIR filter in cascaded transpose sections
                ; The coefficients of each section are scaled for cascaded transpose sections
                ;
00FFFF          datin    equ      $ffff                      ;location in Y memory of input file
00FFFF          datout   equ      $ffff                      ;location in Y memory of output file
00FFF0          m_scr    equ      $fff0                      ;sci control register
00FFF1          m_ssr    equ      $fff1                      ;sci status register
00FFF2          m_sccr   equ      $fff2                      ;sci clock control register
00FFE1          m_pcc    equ      $ffe1                      ;port c control register
00FFFF          m_ipr    equ      $ffff                      ;interrupt priority register
000140          xx       equ      @cvi(20480/(64*1.000))     ;timer interrupt value
FFD8F1          m_tim    equ      -9999                      ;board timer interrupt value
000003          nsec     equ      3                          ;number of second order sections
                         include 'cascadel.asm'
                ;
                ;This code segment implements cascaded biquad sections in transpose form
                ;
                cascadel macro   nsec

                ;assumes each section's coefficients are divided by 2

                         do      #nsec,_ends                         ;do each section
m                        macr    y0,y1,a    x:(r1),b    y:(r4)+,y0   ;a=x(n)*bi0/2+wil/2,b=wi2,y0=bil/2
m                        asr     b          a,x0                     ;b=wi2/2,x0=y(n)
m                        mac     y0,y0,b    y:(r4)+,y0               ;b=x(n)*bil/2+wi2/2,y0=ail/2
m                        macr    x0,y0,b    y:(r4)+,y0               ;b=b+y(n)*ail/2,y0=bi2/2
m                        mpy     y0,y1,b    b,x:(r0)+   y:(r4)+,y0   ;b=x(n)*bi2/2,save wil,y0=ai2
m                        macr    x0,y0,b    x:(r0),a    a,y1         ;b=b+y(n)*ai2/2,a=next iter wil,
m                                                                   ;y1=output of section i
```

Figure 6-10 COEFF.ASM File Generated by MGEN for Example Design and Cascaded Transpose–Form Implementation *(sheet 1 of 4)*

```
38  m                              asr   a    b,x:(r1)+    y:(r4)+,y0   ;a=next iter wi1/2,save wi2,
39  m                                                                   ;y0=next iter bi0
40  m                   _ends       endm
41  m
42
43
44        X:0000                   org   x:0
45        X:0000       state1       dsm   nsec
46        X:0004       state2       dsm   nsec
47        Y:0000                    org   y:0
48                     coef
49  d     Y:0000  0E0492            dc    $0E0492           ;b(*,0)/2   = .10951447section number 1
50  d     Y:0001  1C0924            dc    $1C0924           ;b(*,1)/2   = .21902901 section number 1
51  d     Y:0002  095E7B            dc    $095E7B           ;a(*,1)/2   = .07319582 section number 1
52  d     Y:0003  0E0492            dc    $0E0492           ;b(*,2)/2   = .10951447 section number 1
53  d     Y:0004  FE8F3B            dc    $FE8F3B           ;a(*,2)/2   =-.01125395 section number 1
54  d     Y:0005  102159            dc    $102159           ;b(*,0)/2   = .12601769 section number 2
55  d     Y:0006  2042B2            dc    $2042B2           ;b(*,1)/2   = .25203538 section number 2
56  d     Y:0007  0AC7EB            dc    $0AC7EB           ;a(*,1)/2   = .08422601 section number 2
57  d     Y:0008  102159            dc    $102159           ;b(*,2)/2   = .12601769 section number 2
58  d     Y:0009  F4B2B1            dc    $F4B2B1           ;a(*,2)/2   =-.08829677 section number 2
59  d     Y:000A  1SD3D2            dc    $15D3D2           ;b(*,0)/2   = .17052674 section number 3
60  d     Y:000B  2BA7A4            dc    $2BA7A4           ;b(*,1)/2   = .34105355 section number 3
61  d     Y:000c  0E96B6            dc    $0E9686           ;a(*,1)/2   = .11397439 section number 3
62  d     Y:000D  1SD3D2            dc    $15D3D2           ;b(*,2)/2   = .17052674 section number 3
63  d     Y:000E  DA1A01            dc    $DA1A01           ;a(*,2)/2   =-.29608148 section number 3
64                                  include 'body1.asm'
65
66        000040       start        equ   $40               ;origin for user program
67
68        P:0000                    org   p:$0
69        P:0000  0C0040            jmp   start             ;jump to 'start' on system reset
70
71        P:001C                    org   p:$1c             ;origin for timer interrupt vector
72        P:001C  0BF080            jsr   filter            ;jump to 'filter' on timer interrupt
                  000057
73
74        P:0040                    org   p:start           ;origin for user program
```

Figure 6-10 COEFF.ASM File Generated by MGEN for Example Design and Cascaded Transpose–Form Implementation

(sheet 2 of 4)

```
75
76   P:0040   0003F8         ori     #3,mr                               ;disable all interrupts
77   P:0041   08F4B2         movep   #(xx-1),x:m_sccr                    ;cd=xx 1 for divide by xx
              00013F
78   P:0043   08F4A1         movep   #$7,x:m_pcc                         ;set cc(2:0) to turn on timer
              000007
79   P:0045   08F4B0         movep   #$2000,x:m_scr                      ;enable timer interrupts
              002000
80   P:0047   08F4BF         movep   #$c000,x:m_ipr                      ;set interrupt priority for sci
              00C000
81
82
83   P:0049   300000         move    #state1,r0                          ;point to filter state1
84   P:004A   310400         move    #state2,r1                          ;point to filter state2
85   P:004B   340000         move    #coef,r4                            ;point to filter coefficients
86   P:004C   0502A0         move    #nsec-1,m0                          ;addressing modulo nsec
87   P:004D   050EA4         move    #5*nsec-1,m4                        ;addressing modulo 5*nsec
88   P:004E   0502A1         move    #nsec-1,m1                          ;addressing modulo nsec
89   P:004F   200013         clr     a                                   ;initialize internal state storage
90   P:0050   0603A0         rep     #nsec                               ;*    zero state1
91   P:0051   565800         move    a,x:(r0)+                           ;*
92   P:0052   0603A0         rep     #nsec                               ;*    zero state2
93   P:0053   565900         move    a,x:(r1)+                           ;*
94
95   P:0054   F88000         move    x:(r0),a     y:(r4)+,y0             ;a=wl (initially zero) ,y0=b10/2
96
97   P:0055   00FCB8         andi    #$fc,mr                             ;allow interrupts
98   P:0056   0C0056         jmp     *                                   ;wait for interrupt
99
100           filter
101  P:0057   0008F8         ori     #$08,mr                             ;set scaling mode
102  P:0058   09473F         movep   y:datin,y1                          ;get sample
103
104                          cascade1 nsec                               ;do cascaded biquads
105 +                ;
106 +                ;    assumes each section's coefficients are divided by 2
107 +                ;
```

Figure 6-10 *COEFF.ASM File Generated by MGEN for Example Design and Cascaded Transpose–Form Implementation* *(sheet 3 of 4)*

```
108 + P:0059  060380    do    #nsec,_ends                    ;do each section
              000061
109 + P:005B  FC81B3    macr  y0,y1,a x:(r1),b  y:(r4)+,y0    ;a=x(n)*bi0/2+wi1/2,b=wi2,y0=bi1/2
110 + P:005C  21C42A    asr   b         a,x0                  ;b=wi2/2,x0=y(n)
111 + P:005D  4EDCBA    mac   y0,y1,b           y:(r4)+,y0    ;b=x(n)*bi1/2+wi2/2,y0=ail/2
112 + P:005E  4EDCDB    macr  x0,y0,b           y:(r4)+,y0    ;b=b+y(n)*ail/2,y0=bi2/2
113 + P:005F  FC18B8    mpy   y0,y1,b b,x:(r0)+ y:(r4)+,y0    ;b=x(n)*bi2/2,save wil,y0=ai2
114 + P:0060  19A0DB    macr  x0,y0,b x:(r0),a  a,y1          ;b=b+y(n)*ai2/2,a=next iter wil,
115 +                                                         ;y1=output of section i
116 + P:0061  FC1922    asr   a       b,x:(r1)+  y:(r4)+,y0   ;a=next iter wil/2,save wi2,
117 +                                                         ;y0=next iter bi0
118 +          _ends
119
120 + P:0D62  09C73F    movep y1,y:datout                     ;output sample
121           _endp
122
123 + P:0063  000004    rti
124                      end
0 Errors
0 Warnings
```

FIR FILTERS

"The filter can be efficiently implemented on the DSP56001 by using modulo addressing to implement the shifting and parallel data moves to load the multiplier-accumulator."

If phase distortion is of secondary importance to magnitude response, the desired filter can generally be implemented with less memory, less computational complexity, and at the lowest cost using IIR structures. On the other hand, in applications requiring linear phase in the passband and a specific magnitude response, the specified filter is generally best implemented using FIR structures. Examples of applications requiring linear phase are as follows:

4. Communication systems such as modems or Integrated Services Digital Networks (ISDN) in which the data pulse shape and relative timing in the channel must be preserved

5. Ideal differentiators which provide a 90-degree phase shift at all frequencies in addition to a constant group delay

6. Hilbert transformers used to demodulate complex signals such as those used in high-speed modems

7. Hi-fidelity audio systems in which phase distortion of recorded music must be minimized to reproduce the original sound with as much fidelity as possible

8. System synthesis in which the system impulse response is known as *a priori*. FIR filters are also important because they are all-zero filters (i.e., no feedback) and are therefore guaranteed to be stable.

7.19 Linear-Phase FIR Filter Structure

The basic structure of a FIR filter is simply a tapped delay line in which the output from each tap is summed to generate the filter output. This is shown in Figure 7-31. This structure can be represented mathematically as:

$$y(n) = \sum_{i=0}^{N-1} h(i)x(n-i) \qquad \text{Eqn. 7-16}$$

where: $x(n)$ *is the most recent (t = nT) input signal sample*

$x(n-i)$ *is signal samples delayed by i sample periods (iT)*

$h(i)$ *is the tap weights (or filter coefficients)*

$y(n)$ *is the filter output at time t = nT*

From this structure, it is easy to see why the filter is termed finite—the impulse response of the filter will be identically zero after N sample periods because an impulse input (i.e., $x(n) = 1$, $x(n-i) = 0$ for $i \neq 0$) will have traversed the entire delay line at time $t = NT$. That is, the impulse response of FIR filters will only last for a finite period of time. This response is in contrast to IIR filters which will "ring" in response to an impulse for an infinite period of time. The values of the coefficients represent the impulse response of the FIR filter as can be seen by evaluating Eqn. 7-16 over N sample periods for a single-unit input pulse at time $t = 0$.

Figure 7-31 *FIR Structure*
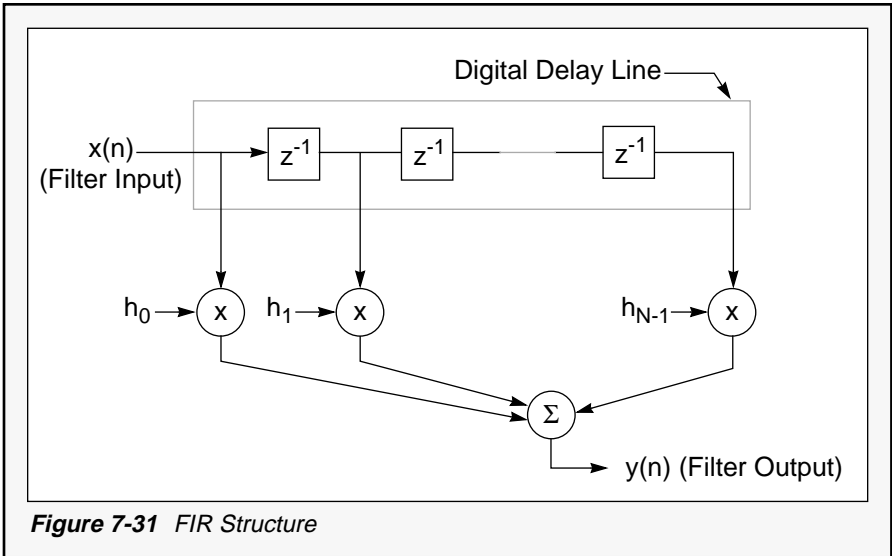
There are no feedback terms in the structure (i.e., Eqn. 7-16 has no denominator) but rather only N zeros. By taking the z-transform of Eqn. 7-16, the transfer function of the filter becomes:

$$H(z) \; = \; \frac{Y(z)}{X(z)} \; = \; \sum_{i \, = \, 0}^{N-1} h(i)z^{-1} \qquad \text{Eqn. 7-17}$$

Eqn. 7-17 is a polynomial in z of order N. The roots of this polynomial are the N zeros of the filter.

The same procedures used to calculate the magnitude and phase response of IIR filters can be applied to FIR filters. Accordingly, the gain, $G(\theta)$, can be obtained by substituting $z = e^{j\theta}$ into Eqn. 7-17 (where $\theta = 2\pi/f_s$ is the normalized digital frequency), then taking the absolute value so that:

$$G(\theta) = \left| \sum_{i=0}^{N-1} h(i)e^{-ji\theta} \right|$$

<div align="right">Eqn. 7-18</div>

$$= \sum_{i=0}^{N-1} [h(i)\cos i\theta - jh(i)\sin i\theta]$$

The phase response, $\phi(\theta)$, is found by taking the inverse tangent of the ratio of the imaginary to real components of $G(\theta)$ so that:

$$\phi(\theta)=\tan^{-1}\left[\frac{-\sum_{i=0}^{N-1} h(i)\sin i\theta}{\sum_{i=0}^{N-1} h(i)\cos i\theta} \right]$$

<div align="right">Eqn. 7-19</div>

where: *h(i) is implicitly assumed to be real*

Intuitively, it can be reasoned that, for any pulse to retain its general shape and relative timing (i.e., pulse width and time delay to its peak value) after passing through a filter, the delay of each frequency component making up the pulse must be the same so that each component recombines in phase to reconstruct the original shape. In terms of phase, this implies that the phase delay must be linearly related to frequency or:

$$\phi(\theta) = -\tau\theta$$

<div align="right">Eqn. 7-20</div>

This relationship is illustrated in Figure 7-2 in which a pulse having two components, $\sin \omega t$ and $\sin 2\omega t$, passes through a linear-phase filter having a delay of two cycles per Hz and a constant magnitude response equal to unity. That is, the fundamental is

delayed by two cycles (4π) and the first harmonic is delayed by four cycles (8π). The group delay of a system, τ_g, is defined by taking the derivative of $\phi(\theta)$:

$$\tau_g \equiv -\frac{d\phi}{d\theta}$$

Eqn. 7-21

Figure 7-32 *Signal Data through a FIR Filter*
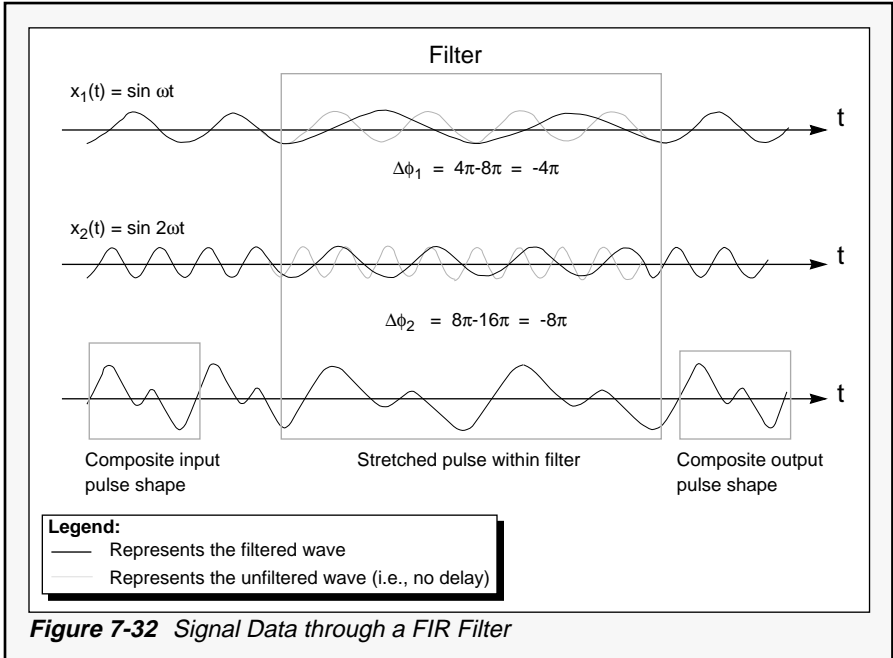
Since θ is the normalized frequency, τ_g in Eqn. 7-21 is a dimensionless quantity and can be related to the group delay in seconds by dividing by the sample frequency, f_s. For a linear-phase system, τ_g is independent of θ and is equal to τ. This fact can be seen by substituting Eqn. 7-20 into Eqn. 7-21.

In this example, τ_g is two cycles per Hz. Note that the pulse shape within the filter has been retained

but is twice the width of the original pulse; that is the change in the pulse width is equal to the group delay. The negative sign indicates the phase is retarded or delayed (i.e., a causal system).

The impact of the requirement of linear phase on the design of a FIR filter can be seen by substituting Eqn. 7-19 into Eqn. 7-20 so that:

$$\tan \theta \tau_g = \left[\frac{\displaystyle\sum_{i=0}^{N-1} h(i)\sin i\theta}{\displaystyle\sum_{i=0}^{N-1} h(i)\cos i\theta} \right] \qquad \text{Eqn. 7-22}$$

which can be written as:

$$\sum_{i=0}^{N-1} h(i)(\cos i\theta \sin \theta \tau_g - \sin i\theta \cos \theta \tau_g) = 0$$

or

$$\sum_{i=0}^{N-1} h(i)\sin[\theta(\tau_g - i)] = 0 \qquad \text{Eqn. 7-23}$$

The solution to is the constraint on τ_g for a FIR filter to be linear phase. The solution to can be found by expanding the left-hand side (LHS) as follows:

$$LHS = h_0 \sin\theta\tau_g + h_1 \sin\theta(\tau_g - 1) + h_2 \sin\theta(\tau_g - 2) + \ldots$$

$$+h_{N-3}\sin\theta(\tau_g - N + 3) + h_{N-2}\sin\theta(\tau_g - N + 2) + h_{N-1}\sin\theta(\tau_g - N + 1)$$

so that if:

$$\tau_g = \frac{(N-1)}{2} \qquad \text{Eqn. 7-24}$$

then for every positive argument there will be a corresponding negative argument. For example, the argument for the h_1 term becomes:

$$\theta\left(\frac{N-1}{2} - 1\right) = \theta\left(\frac{N-3}{2}\right)$$

which is the negative of the argument for the h_{N-2} term, i.e.,
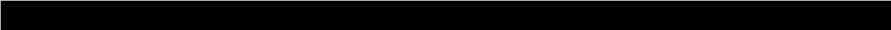
$$\theta\left(\frac{N-1}{2} - N + 2\right) = -\theta\left(\frac{N-3}{2}\right)$$

so that if:

$$h(i) = h(N - 1 - i) \quad \text{for} \quad 0 \leq i \leq N - 1 \qquad \text{Eqn. 7-25}$$

then Eqn. 7-9 and Eqn. 7-25 represent the solution to . By substituting Eqn. 7-9 into Eqn. 7-20, the phase of a linear-phase FIR filter is given by:

$$\phi(\theta) = -\left(\frac{N-1}{2}\right)\theta \qquad \text{Eqn. 7-26}$$

Therefore, a nonrecursive filter (FIR), unlike a recursive filter (IIR), can have a constant time delay for all frequencies over the entire range (from 0 to $f_s/2$). It is only necessary that the coefficients (and therefore impulse response) be symmetrical about the midpoint between samples (N-2)/2 and N/2 for even N or about sample (N-1)/2 for odd N (see Eqn. 7-25). When this symmetry exists, τ_g for the filter will be $\left(\dfrac{N-1}{2}\right)$ T seconds.

7.20 Linear-Phase FIR Filter Design Using the Frequency Sampling Method

Implementing a FIR filter in DSP hardware such as the DSP56001 is a relatively simple task. Determining a set of coefficients that describe a given impulse response of a filter is also a straightforward procedure. However, deriving the optimal coefficients necessary to obtain a particular response in the frequency domain is not always as easy. The following paragraphs introduce a simple method to determine a set of coefficients based on a desired arbitrary frequency response (often referred to as the frequency sampling method). This method has the distinct advantage of being done in real time. However, the most efficient determination of coefficients is best done by utilizing a software filter

design system such as FDAS to perform numerical curve-fitting and optimization, a procedure that necessitates using a computer. For example, the inverse Fourier transform integral is used to determine the FIR coefficients from a response specification in the frequency domain, which generally requires a numerical integration procedure. The inverse discrete Fourier transform (IDFT), which can be implemented using the inverse fast Fourier transform (IFFT) algorithm, is discussed in the following paragraphs. One reason for choosing this approach is to demonstrate a method that can be used to determine coefficients in real time since the fast Fourier transform (FFT) can be implemented in the same DSP56001 hardware as the FIR filter.

The question is "How must the filter be specified in the frequency domain so that Eqn. 7-25 and Eqn. 7-26 are realized?". That is, starting with the definition of the filter in the frequency domain, a method for calculating the filter coefficients is required. Beginning with Eqn. 7-17, the z-transform of the FIR filter, and evaluating this transfer function on the unit circle at N equally spaced normalized frequencies (i.e., $z = e^{j2\pi lk/N}$) produces:

$$H(k) = H(e^{j2\pi k/N}) = \sum_{i=0}^{N-1} h(i)e^{-j2\pi ki/N}$$

$$\text{Eqn. 7-27}$$

where: $0 \leq k \leq N\text{-}1$

h(i) can be solved in terms of the frequency response at the discrete frequencies, H(k), by multiplying both

sides of by $e^{j2\pi km/N}$ and summing over k as follows:

$$\sum_{k=0}^{N-1} H(k)e^{j2\pi km/N} = \sum_{k=0}^{N-1}\sum_{i=0}^{N-1} h(i)e^{-j2\pi ki/N}e^{j2\pi km/N}$$

$$= \sum_{i=0}^{N-1} h(i) \sum_{k=0}^{N-1} e^{-j2\pi ki/N}e^{j2\pi km/N}$$

$$= \sum_{i=0}^{N-1} h(i)N\delta_{im}$$

$$= Nh(m)$$

Eqn. 7-28

where: δ_{im} is the Kronecker delta, which is equal to one when $i = m$ but is zero otherwise

can be used to find $h(i)$ by simply setting $i = m$.

$$h(i) = \frac{1}{N}\sum_{k=0}^{N-1} H(k)e^{j2\pi ki/N}$$ Eqn. 7-29

Eqn. 7-29 is the IDFT of the filter response. In general, the discrete Fourier coefficients are complex; therefore, H(k) can be represented as:

$$H(k) = A(k)e^{j\phi(k)}$$

where: $A(k) \equiv |H(k)|$
so that Eqn. 7-29 becomes:

$$h(i) = \frac{1}{N} \sum_{k=0}^{N-1} A(k)e^{j\phi(k)}e^{j2\pi ki/N} \qquad \text{Eqn. 7-30}$$

At this point, the linear-phase constraints, Eqn. 7-25 and Eqn. 7-26, can be applied. The constraint that the coefficients be real and symmetrical when h(i) is complex (see Reference 1) implies that:

$$h(i) = h*(N-1-i)$$

where: * signifies the complex conjugate

therefore:

$$h*(N-1-i) = \frac{1}{N} \sum_{k=0}^{N-1} A(k)e^{-j\phi(k)}e^{-j2\pi k(N-1-i)/N}$$

$$= \frac{1}{N} \sum_{k=0}^{N-1} A(k)e^{-j[\phi(k)+2\pi k(N-1)/N]}e^{j2\pi ki/N}$$

Eqn. 7-31

will be identical to Eqn. 7-30 if:

$$\phi(k) = -\phi(k) - \frac{2\pi k(N-1)}{N} + 2\pi r \qquad \text{for r=0,1,2,...}$$

or

$$\phi(k) = \pi r - \pi k \frac{(N-1)}{N} \qquad \text{Eqn. 7-32}$$

What are the constraints on r and A(k) which will guarantee a purely real response for N even? Substituting Eqn. 7-32 into Eqn. 7-30 yields:

$$h(i) = \frac{1}{N} \sum_{k=0}^{N-1} A(k)e^{j[\pi r - \pi k(N-1)/N + 2\pi ki/N]}$$

Expanding Eqn. 7-33 for even N yields:

$$\text{\`i)} = \frac{1}{N}\{A(0)e^{j\pi r} + A(1)e^{j[\pi r - \pi(N-1)/N + 2\pi i/N]} + .$$

$$+A(N/2)e^{j[\pi r - \pi(N-1)/2 + \pi i]} + \ldots$$

$$+A(N-1)e^{j[\pi r - \pi(N-1)(N-1)/N + 2\pi(N-1)i/N]}\}$$

Consider the A(k) and A(N-k) terms; if r = 0 for the A(k) term and r = 1 for the A(N-k) term, then the argument for the A(N-k) term is the negative of the argument for the A(k) term, given that N is even. That is:

$$e^{j[\pi - \pi(N-k)(N-1)/N + 2\pi(N-k)i/N]}$$

$$= e^{j[\pi k(N-1)/N - 2\pi ki/N]}e^{j[\pi - \pi(N-1) + 2\pi i]}$$

$$= e^{-j[-\pi k(N-1)/N + 2\pi ki/N]}e^{-j[\pi(N-2) + 2\pi i]}$$

$$= e^{-j[-\pi k(N-1)/N - 2\pi ki/N]}$$

Therefore, if A(k) = A(N-k), then all imaginary components in Eqn. 7-33 will cancel and h(i) will be

purely real, which is the desired result. In general, for N even, the formulas for the frequency sampling method can be reduced to:

$$h(i) = \frac{1}{N}\left\{A(0)+ \sum_{k=1}^{N-1} A(k)e^{j[\pi r - \pi k(N-1)/N + 2\pi ki/N]}\right\}$$

$$= \frac{1}{N}\left\{A(0)+ \sum_{k=1}^{N/2-1} Ak[e^{j[\pi r - \pi k(N-1)/N + 2\pi ki/N]}+ e^{j[\pi - \pi(N-k)(N-1)/N + 2\pi(N-k)i/N]}]\right\}$$

$$= \frac{1}{N}\left\{A(0) + \sum_{k=1}^{N/2-1} Ak[e^{j-\pi kN-1/N + 2\pi ki/N}+ e^{-j[-\pi k(N-1)/N + 2\pi ki/N]}]\right\}$$

$$= \frac{1}{N}\left\{A(0) + \sum_{k=1}^{N/2-1} A(k)[e^{-j\pi k}e^{j[\pi k/N + 2\pi ki/N]}+ e^{j\pi k}e^{-j[\pi k/N + 2\pi ki/N]}]\right\}$$

$$= \frac{1}{N}\left\{A(0) + 2 \sum_{k=1}^{N/2-1} A(k)(-1)^k\cos\frac{\pi k}{N}(1 + 2i)\right\}$$

Eqn. 7-34

where: r= 0 for 0 ≤ k < N/2

r = 1 for N/2 < k ≤ N-1

given

$$A(k) = A(N-k) \qquad \text{for } 1 \le k \le \frac{N}{2}\text{-}1$$

with

$$A(N/2) = 0$$

and

$$\phi(k) = -\pi k \frac{N-1}{N} \qquad \text{for } 0 \le k \le \frac{N}{2}\text{ -1}$$

$$\phi(k) = \pi - \pi k \frac{N-1}{N} \qquad \text{for } \frac{N}{2} + \le k \le \text{N-1}$$

In summary, if a filter with an even number of symmetrical real coefficients is desired, then the phase must be linear, and the frequency response must be symmetrical about N/2 and zero at N/2 (see Reference 1).

(a) Arbitrary Input Specification for Frequency Response Magnitude

(b) Necessary Phase Specifications for Frequency Response Magnitude

Figure 7-33 *Arbitrary Filter Example*
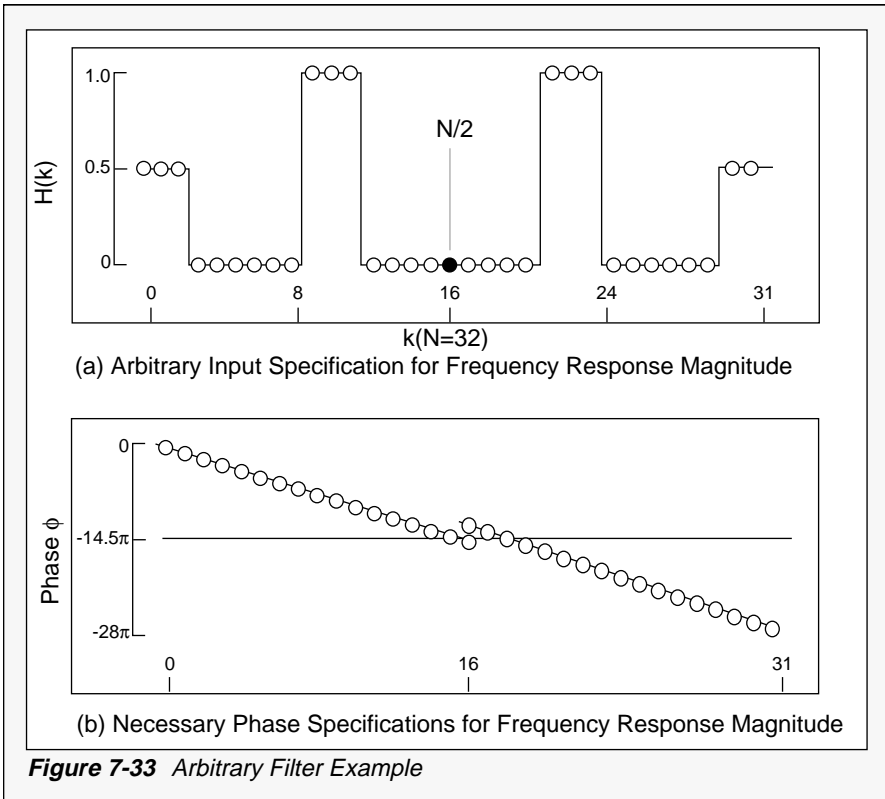
As an example, consider the arbitrary filter speci-
fied in Figure 7-34. In this example, N = 32 and an
arbitrary lowpass and bandpass combination is
specified. Figure 7-34 shows the result of trans-
forming the polar coordinate filter specification into
rectangular coordinates, yielding real and imagi-
nary components of the transfer function. This
transformation is accomplished by treating the
magnitude of H(k) as the length of a vector in polar
coordinates and the phase as the angle. The x
component (real part) is the product of the length
(magnitude of H(k)) and the cosine of the phase.

Likewise, the y component (imaginary part) is the product of the length and the sine of the phase angle. This transformation to rectangular coordinates is necessary to perform the IDFT (or IFFT) calculation:

$$h(n) = \frac{1}{N} \sum_{k=0}^{N-1} H(k)e^{jk\theta n} \qquad \text{Eqn. 7-35}$$

where: *H(k) is a complex number and θn = 2πn/N*

When the filter's transfer function, H(k), is described as in Figure 7-34 and Figure 7-34 (i.e., the real part symmetric about N/2 and the imaginary part asymmetric about N/2), h(n) is strictly real. If h(n) were complex, the FIR filter would be much more difficult to implement since twice as many terms would be present. Figure 7-35 shows the results of the IDFT applied to the example arbitrary filter specification.

Note the symmetry of the coefficients in Figure 7-35 (symmetric about (N-1)/2). This symmetry is to be expected for an even number of coefficients. Eqn. 7-36 has N-1 roots since it is a polynomial of order N-1. These roots are plotted in Figure 7-36 and must be found from a numerical algorithm such as the Newton-Raphson root-finding recursion relation or the Mueller method (see Reference 18). h(n) can now be used to specify the filter response in the continuous frequency domain by setting $\theta = 2\pi f/f_s$.

Figure 7-34 *Response Transformed from Polar Coordinates*
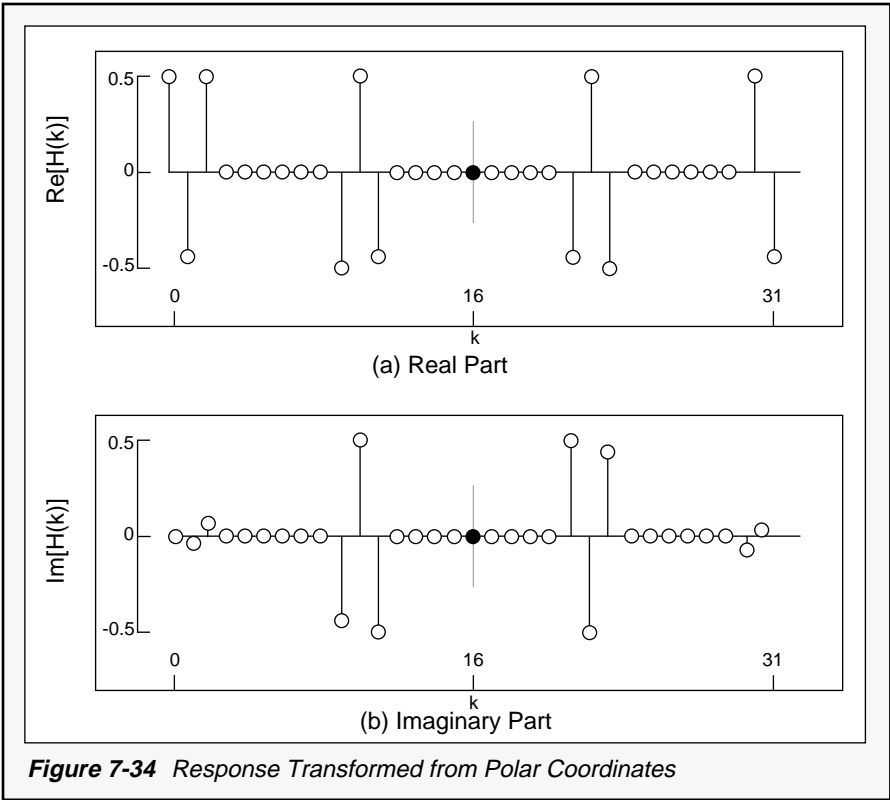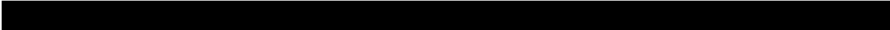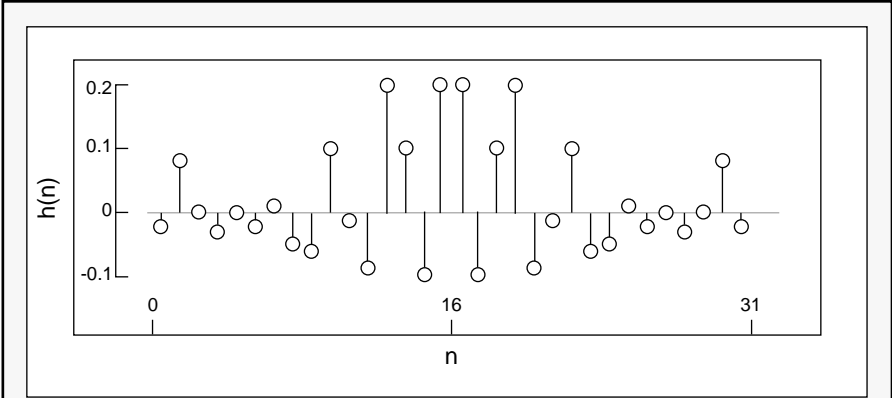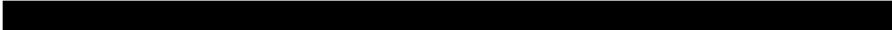
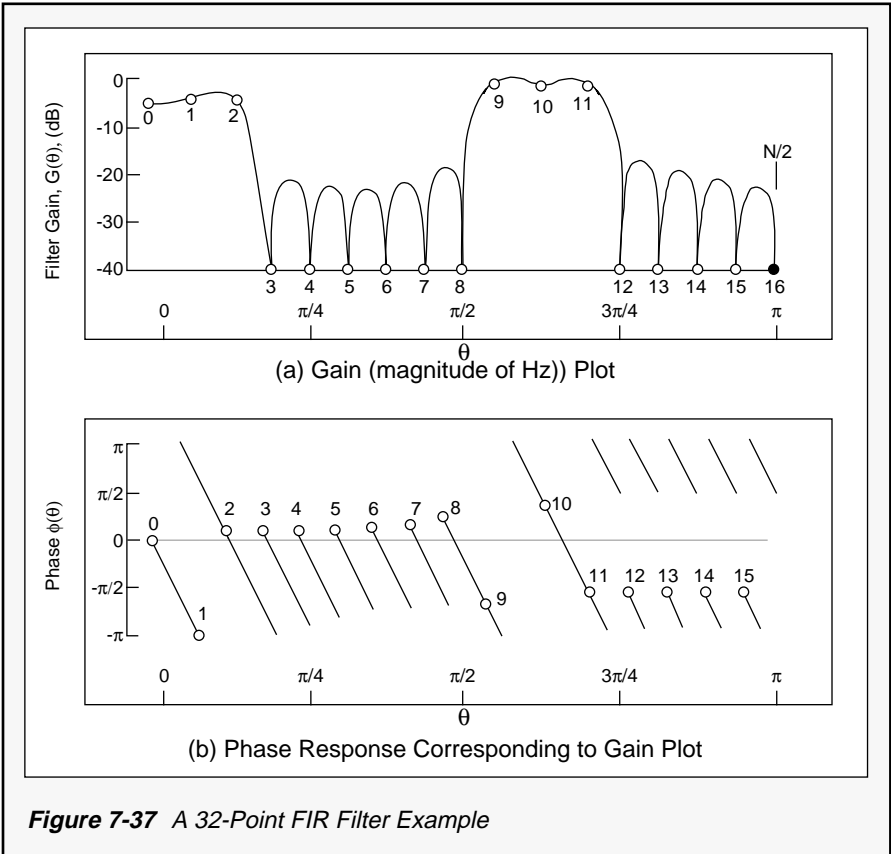**Figure 7-35** *FIR Coefficients from Eqn. 7-17 for Filter Example*
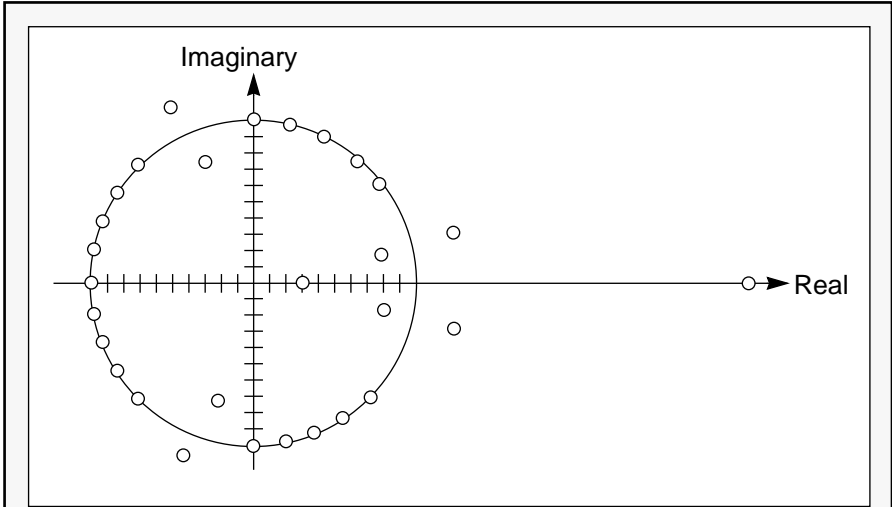
Figure 7-36 *Roots (Zeros) of Eqn. 7-36 for Filter Example*

The transfer function has exactly the same form as the DFT of the h(n), but now the frequency is continuous up to $f_s/2$:

$$H(\theta) = \sum_{n=0}^{N-1} h(n)e^{-jn\theta}$$

Eqn. 7-36

where: $\theta = 2\pi f/f_s$

(a) Gain (magnitude of Hz)) Plot

(b) Phase Response Corresponding to Gain Plot

Figure 7-37 *A 32-Point FIR Filter Example*

The continuous frequency gain and phase response of the 32-coefficient example filter, plotted in Figure 7-37, are generated as follows:

$$G(\theta) = |H(\theta)| = [H(\theta)H^*(\theta)]^{1/2}$$

$$= \left\{ \left[\sum_{n=0}^{N-1} h(n)\cos n\theta \right]^2 + \left[\sum_{n=0}^{N-1} h(n)\sin n\theta \right]^2 \right\}^{1/2}$$

Eqn. 7-37

and

$$\phi(\theta) = (-\tan^{-1})\left\{ \frac{\displaystyle\sum_{n=0}^{N-1} h(n)\sin n\theta}{\displaystyle\sum_{n=0}^{N-1} h(n)\cos n\theta} \right\}$$

Eqn. 7-38

where: *h(n) is the value obtained from*
or equivalently from Eqn. 7-19

Note that the gain plot (see Figure 7-37) exactly intersects the discrete frequency points originally specified in Figure 7-34. Clearly, $G(\theta)$ and $\phi(\theta)$ have many discontinuities. Due to the symmetry of linear-phase FIR filters, analytic expressions can be found for $H(\omega)$ as shown in Eqn. 7-42 and Eqn. 7-43.

For comparison, Figure 7-38 shows the log magnitude response of the example filter with larger values of N. The stopband attenuation is not improved as much as expected when the number of coefficients is increased; however, the sharpness of the transition band is significantly enhanced because the approach discussed has implicitly
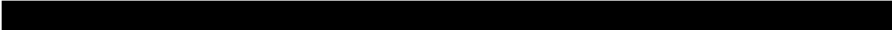
assumed a rectangular window function. A smooth-
ing window, w(n), can be used to improve this
situation (see Reference 10).

Figure 7-38 *Log Magnitude Response of Filter Example with Larger N Values*
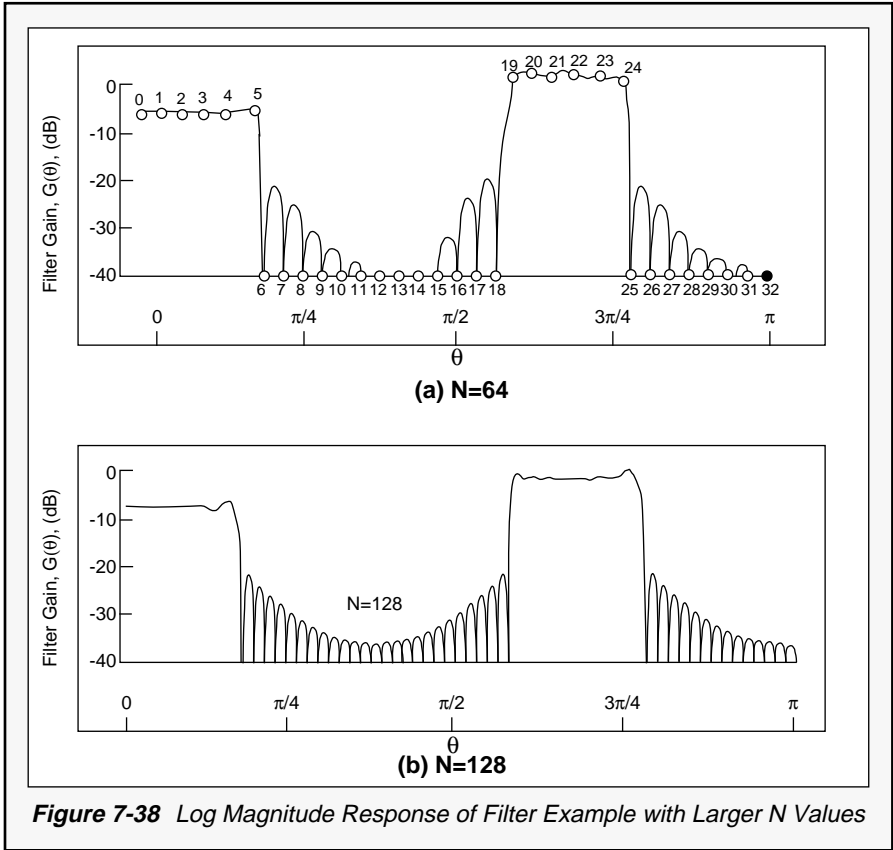
If h(n) is modified by a window function, w(n), $h_w(n)$
= h(n)w(n) for $0 \leq n \leq$ N-1, the gain of Eqn. 7-37
(shown in Figure 7-34(a)) can be greatly improved.
The window function, w(n), goes to zero at both
ends (n = 0 and n = N) and is unity at the center; w(n)

is symmetrical so as not to disrupt the linear-phase characteristics of the filter. Figure 7-34(b) uses a window function described by $w(n) = \sin^2[\pi n/(N-1)]$ (also known as a Hanning window) to demonstrate the sensitivity of the gain to windowing. w(n) is basically an envelope function used to taper the ends of h(n) smoothly. The rounding of the transition-band edges in Figure 7-34(b) is the trade-off for windowing; however, in most applications, this trade-off is well worthwhile.

(a) Rectangular Window

(b) Sine^2 (Hanning) Window

Figure 7-39 *Window Function Effects on Filter Example*
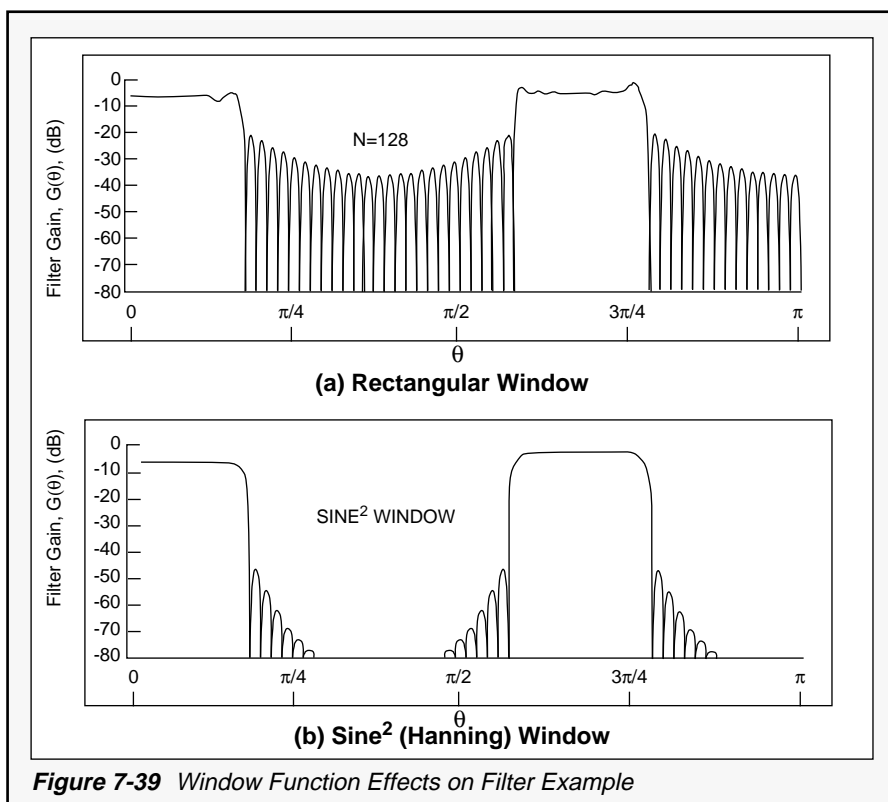
Window functions have a powerful effect on the stopband attenuation as well as the passband transition slope.The best passband transition slope performance is achieved with the rectangular window, but this window results in very poor stopband performance and often severe passband fluctuation (or ripple). All window functions have the effect of increasing the stopband attenuation and reducing the passband ripple at the expense of increasing the width of the transition region. However, for most window functions, the passband ripple is relatively insensitive to N. Windowing is described in virtually any DSP textbook (see REFERENCES). Also, windowing is discussed with practical examples in *Implementation of Fast Fourier Transforms on Motorola's DSP56000/DSP56001 and DSP96002 Digital Signal Processors* (see Reference 19).

7.21 FIR Filter Design Using FDAS

Figure 7-40 shows an example (log magnitude and impulse response) of a bandpass filter generated with the FDAS software package using the Kaiser window. A totally different approach to FIR filter design, the equiripple method, is based on finding an optimum approximation to the ideal or desired response, $D(\theta)$. An optimum approximation can be found because of the inherent symmetry of the coefficients of linear-phase filters.

Figure 7-40 *FDAS Output for FIR Bandpass Filter Example with a Kaiser Window*

Recall that the continuous frequency response of a FIR filter can be found by setting $z = e^{j\theta}$ in the z-transform so that:

$$H(\theta) = \sum_{n=0}^{N-1} h(n)e^{-j\theta n} \qquad \text{Eqn. 7-39}$$

If the linear-phase factor is factored out, Eqn. 7-39 can be written as:

$$H(\theta) = e^{-j\theta(N-1)/2} \sum_{n=0}^{N-1} h(n)e^{j\theta[(N-1)/2-n]}$$

$$= e^{-j\theta(N-1)/2}\{h(0)e^{j\theta(N-1)/2} + h(1)e^{j\theta[((N-1)/2)-1]} + \dots$$

$$h(N-1))e^{-j\theta(N-1)/2} + h(N-2)e^{-j\theta[((N-1)/2)-1]}\}$$

Eqn. 7-40

Using Euler's identity, Eqn. 7-40 can be grouped into sine and cosine terms as follows:

$$H(\theta) = e^{-j\theta(N-1)/2}\{[h(0) + h(N-1)]\cos\left[\theta\frac{(N-1)}{2}\right]$$

$$+jh0-hN-1\sin\left[\theta\frac{N-1}{2}\right] + \left[h(1) + h(N-2)\right]\cos\left[\theta\frac{(N-1)}{2}-1\right]$$

$$+j[h(1)-h(N-2)]\sin\left[\theta\frac{N-1}{2}-1\right]$$

Eqn. 7-41

If it is assumed that linear phase is achieved by even symmetry (i.e., h(n) = h(N-1-n) and that N is even, reduces to:

$$H(\theta) = e^{-j\theta(N-1)/2} \sum_{n=0}^{N/2-1} 2h(n)\cos\left[\theta\left(\frac{N-2}{2} - n\right)\right]$$

which, with a change of variable, k = N/2-n-1, can be written as follows:

$$H(\theta) = e^{-j\theta(N-1)/2} \sum_{n=0}^{N/2-1} 2h(n)\cos\left[\theta\left(\frac{N-2}{2} - n\right)\right]$$

Eqn. 7-42

which is in the following form:

$$H(\theta) = A(\theta)e^{jP\theta}$$

Eqn. 7-43

where: *A(θ) is a real value amplitude function*
        P(θ) is a linear-phase function

The linear functions, A(θ) and P(θ), are to be con-trasted with the inherently nonlinear absolute value and arctangent functions in and Eqn. 7-38. For all four types of linear-phase filters (N even or odd and symmetry even or odd), A(θ) can be expressed as a sum of cosines (see Reference 21).

Since an ideal filter cannot be realized, an approxima-tion must be used. If the desired frequency response, D(θ), can be specified in terms of a deviation, δ, from

the ideal response, then the error function in Eqn. 7-44 can be minimized:

$$\|E(\theta)\| = \max_{\theta}|D(\theta) - A(\theta)|$$ Eqn. 7-44

by finding the best $A(\theta)$. Because $A(\theta)$ can be expressed as a finite sum of cosines as shown in Eqn. 7-42, it can be shown (see Reference 18) that the optimal $A(\theta)$ will be unique and will have at least N/2 + 1 extremal frequencies, where extremal frequencies are points such that for:

$$\theta_1 < \theta_2 < ... < \theta_{N/2} < \theta_{N/2+1}$$

$$E(\theta_e) = -E(\theta_{e+1}) \qquad \text{for} e = 1, 2, ..., \frac{N}{2}+1$$
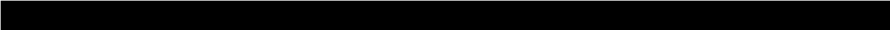
and

$$\left|E(\theta_e)\right| = \max_{\theta}[E(\theta)]$$

Thus, the best approximation will exhibit an equiripple error function. The problem reduces to finding the extremal frequencies since, once they are found, the coefficients, 2h(N/2-k-1), can be found by solving the set of linear equations:

$$D(\theta)\pm\delta = A(\theta_e) = \sum_{k=0}^{N/2-1} 2h\left(\frac{N}{2}-k-1\right)\cos\left[\theta_e\left(k+\frac{1}{2}\right)\right]$$

Eqn. 7-45

The Remez exchange algorithm is used to systematically find the extremal frequencies (see

Reference 5). Basically, a guess is made for the initial N/2 + 1 extremal frequencies. (Usually, this guess consists of N/2 + 1 equally spaced frequencies in the Nyquist range.) Using this guess, is solved for the coefficients and δ. Using these coefficients, $A(\theta)$ is calculated for all frequencies and the extrema, and frequencies at which the extrema are attained are determined. If the extrema are all equal and equal to or less than that specified in the initial filter specification, the problem is solved. However, if this is not the case, the frequencies at which the extrema were attained are used as the next guess. Note that the final extremal frequencies do not have to be equally spaced. Clearly, the equiripple design approach is calculation intensive.

What is the benefit of equiripple designs over window designs? In general, equiripple designs require fewer taps for straightforward requirements. When the specification requires a sharp cutoff and/or a large stopband attenuation or a narrow bandpass, the equiripple approach may fail to converge. In general, when *N* is decreased, an equiripple design tends to maintain its transition band while sacrificing stopband attenuation; window designs tend to do the opposite. Of the window alternatives, the Kaiser window is preferred for designing filters because the passband ripple and stopband attenuation can be varied relatively independent of the transition width (see Reference 1). For spectral analysis, the Blackman-Harris window is preferred (see Reference 10).
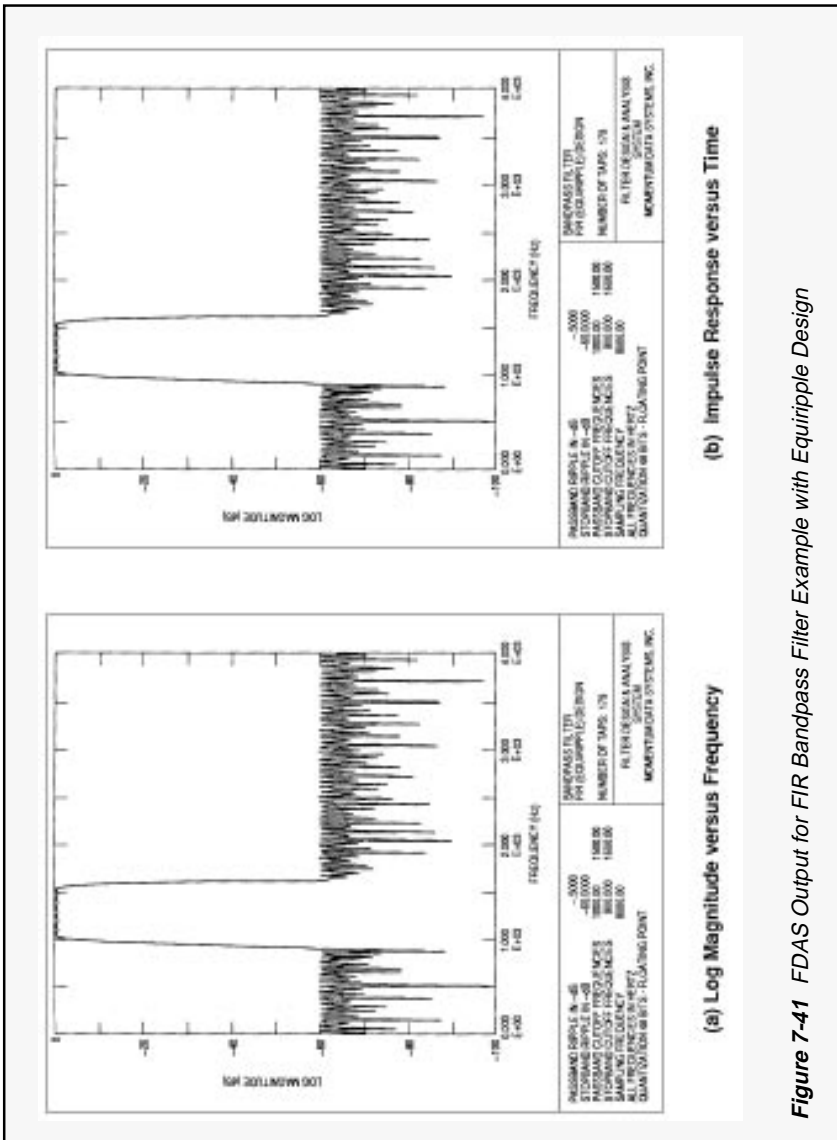
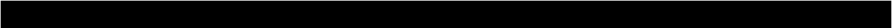**Figure 7-41** *FDAS Output for FIR Bandpass Filter Example with Equiripple Design*

Figure 7-41 shows an example of an equiripple design generated from the FDAS software package. This example is the same as that used for the Kaiser window example of Figure 7-40. The number of coefficients in the equiripple design is far less than that generated by the Kaiser window method (179 versus 291). However, the passband ripple is larger.

7.22 FIR Implementation on the DSP56001

The DSP56001 has several architectural features that make it ideally suited for implementing FIR filters:

1. Dual Harvard architecture uses two data memories with dedicated buses and address generation units, allowing two addresses to be generated in a single cycle. If one address is pointing to data and another address is pointing to coefficients, a word of data and a coefficient can be fetched in a single cycle.

2. Modulo addressing makes the shifting of data unnecessary. If an address pointer is incremented (or decremented) with the modulo modifier in effect, data shifting can be accomplished by just "backing up" the address register by one to overwrite the data that would normally be shifted out. This procedure allows very efficient addressing of operands without wasting time shifting the data or reinitializing pointers.

3. Hardware DO loops execute without overhead once the loop is started. After a three-cycle initialization of the DO loop, the body of the loop executes as if it were straight-line code. Since the DO loop does not require any overhead cycles for each pass, the need for straight-line code is eliminated.

For a four-coefficient example of a linear-phase FIR filter, the input-output difference equation can be found by expanding Eqn. 7-16:

$$y(n) = h_0 x(n) + h_1 x(n-1) + h_2 x(n-2) + h_3 x(n-3)$$

<div align="right">Eqn. 7-46</div>

This difference equation can be realized with the discrete-time four-tap filter example shown in Figure 7-41. The filter can be efficiently implemented on the DSP56001 by using modulo addressing to implement the shifting and parallel data moves to load the multiplier-accumulator. The filter network is shown in Figure 7-41(a); the memory map for the filter inputs and coefficients is shown in Figure 7-41(b). The following DSP56001 code is used to implement the direct form FIR filter:

```
CLR   A        X0,X:(R0)+    Y:(R4)+,Y0  ;Save input sample, fetch coef.
REP   #NTAPS-1                           ;Repeat next instruction.
MAC   X0,Y0,A  X:(R0)+,X 0  Y:(R4)+,Y0  ;FIR Filters.
MACR  X0,Y0,A  (R0)-                     ;Round result and adjust R0.
```

Register R0 points to the input variable buffer, M0 is set to three (modulo 4), R4 points to the coefficient buffer, and M4 is set to three. The input sample is in X0.

The CLR instruction clears accumulator A and performs parallel data moves. The data move saves the most recent input value to the filter (assumed to be in XO) into the location occupied by the oldest data in the shift register and moves the first coefficient in the filter (h_0) into the data ALU.

The REP instruction repeats the next instruction NTAPS-1 times. Since there are four taps in this filter, the next instruction is repeated three times.
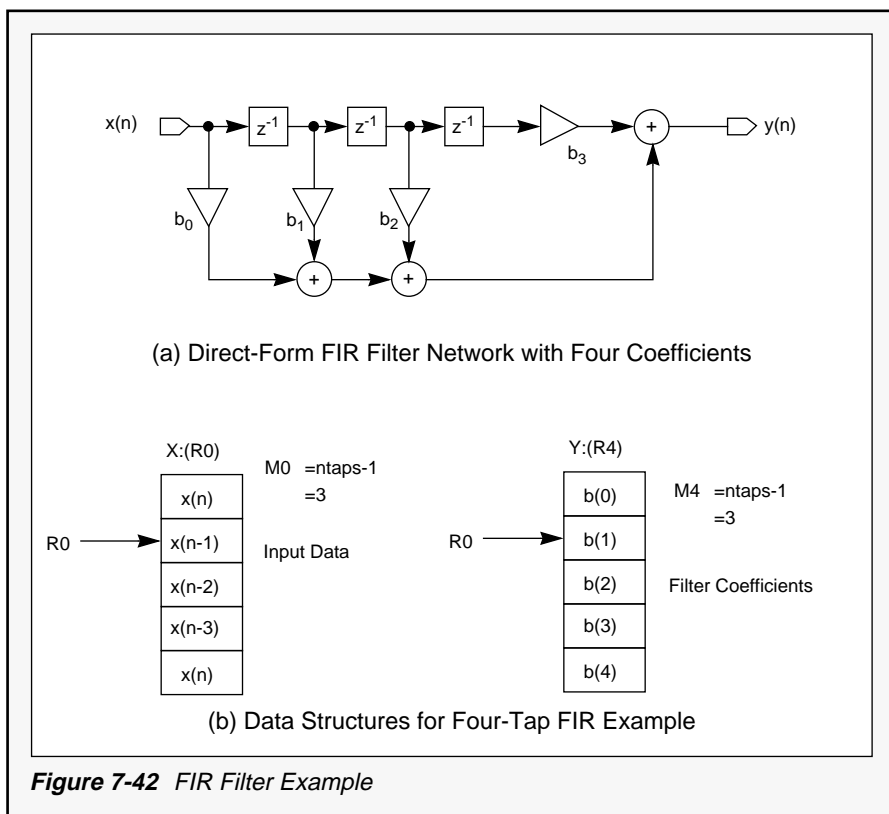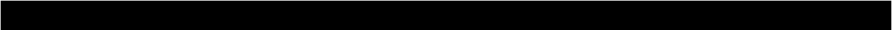

(a) Direct-Form FIR Filter Network with Four Coefficients

(b) Data Structures for Four-Tap FIR Example

Figure 7-42 *FIR Filter Example*

The MAC instruction multiplies the data in X0 by the coefficient in Y0 and adds the result to accumulator A in a single cycle. The data move in this instruction loads the next input data variable into X0 and the next coefficient into Y0. Both address registers R0 and R4 are incremented. The MACR instruction calculates the final tap of the filter, rounds the result using convergent rounding, and address register R0 is decremented.

Address register R4 is incremented once before the REP instruction and three times due to the REP instruction for a total of four increments. Since the modulus for R4 is four, the value of R4 wraps around pointing back to the first coefficient.

The operation of RO is similar. The value input to the filter is saved, and then RO is incremented pointing to the first state. The REP instruction increments RO three times. Since the modulo on RO is four and RO is incremented four times, the value in RO wraps around pointing to the new input sample. When the MACR instruction is executed, the value of RO is decremented, pointing to the old value, x(n-3). The next sample time overwrites the value of x(n-3) with the new input sample, x(n). Thus, the shifting of the input data is accomplished by simply adjusting the address pointer, and the modular addressing wraps the pointer around at the ends of the input data buffer. Instruction cycle counts for this filter are NTAPS+3. Thus, for a four-tap filter, seven instructions are required. ∎
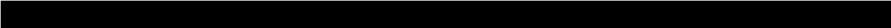
REFERENCES

4. Antoniou, Andreas. *Digital Filters: Analysis and Design* New York, NY: McGraw-Hill, 1974.

5. Berlin, Howard M. *Design of Active Filters with Experiments.* Indianapolis, IN: Howard W. Sams & Co., Inc., 1977.

6. Bogner and Constantinides, eds. *Introduction to Digital Filtering.* New York, NY: John Wiley & Sons, 1975.

7. Brophy, J. J. *Basic Electronics for Scientists.* New York, NY: McGraw-Hill, 1 966.

8. Cheney, E. W. *Introduction to Approximation Theory.* New York, NY: McGraw-Hill, 1966.

9. Chrysafis, A. *Fractional and Integer Arithmetic Using the DSP56000 Family of General-Purpose Digital Signal Processors* (APR3/D). Motorola, Inc., 1988.

10. *Digital Stereo 10-Band Graphic Equalizer Using the DSP56001* (APR2/D). Motorola, Inc., 1988.

11. "Digital Filters on DSP56000/DSP56001." Motorola Technical Bulletin, 1988.

12. "Filter Design & Analysis System." Version 1.3, Momentum Data Systems, 1 985.

13. Harris, Fredric J. "On the Use of Windows for Harmonic Analysis with the Discrete Fourier Transform." *Proc. of IEEE,* vol. 66, no. 1, 1978, pp.51-83.

14. Jackson, Leland B. *Digital Filters and Signal Processing.* Boston, MA: Kluwer Academic Publishers, 1986.

15. Lancaster, D. *Active Filter Cookbook.* Indianapolis, IN: Howard W. Sams & Co., Inc., 1975.

16. McClellan, J. H. and T. W. Parks. "A Unified Approach to the Design of Optimum FIR Linear-Phase Digital Filters." *IEEE Trans. Circuits Systems CT-20,* 1973, pp. 697-701.

17. Moschytzm, G. S. and P. Horn. *Active Filter Design Handbook.* New York, NY: John Wiley & Sons, 1981.

18. Oppenheim, A. V. and R. W. Schafer. *Digital Signal Processing.* Englewood Cliffs, NJ: Prentice-Hall, 1975.

19. Parks, T. W. and J. H. McClellan. "Chebyshev Approximation for Nonrecursive Digital Filters with Linear Phase." *IEEE Trans. Circuit Theory CT- 19,* 1972, pp. 189-194.

20. Proakis, John G. et al. *Introduction to Digital Signal Processing.* New York, NY: MacMillan, 1988.

21. Rabiner, L. R. and B. Gold. *Theory and Application of Digital Signal Processing.* Englewood Cliffs, NJ: Prentice-Hall, 1975.

22. Sohie, Guy R. L. *Implementation of Fast Fourier Transform on Motorola's DSP56000/DSP56001 and DSP96002 Digital Signal Processors* (APR4/D). Motorola, Inc., 1989.

23. Strawn, J., et al. *Digital Audio Signal Processing—An Anthology.* William Kaufmann, 1985.

24. Williams, Arthur B. *Electronic Filter Design Handbook.* New York, NY: McGraw-Hill, 1981. ∎