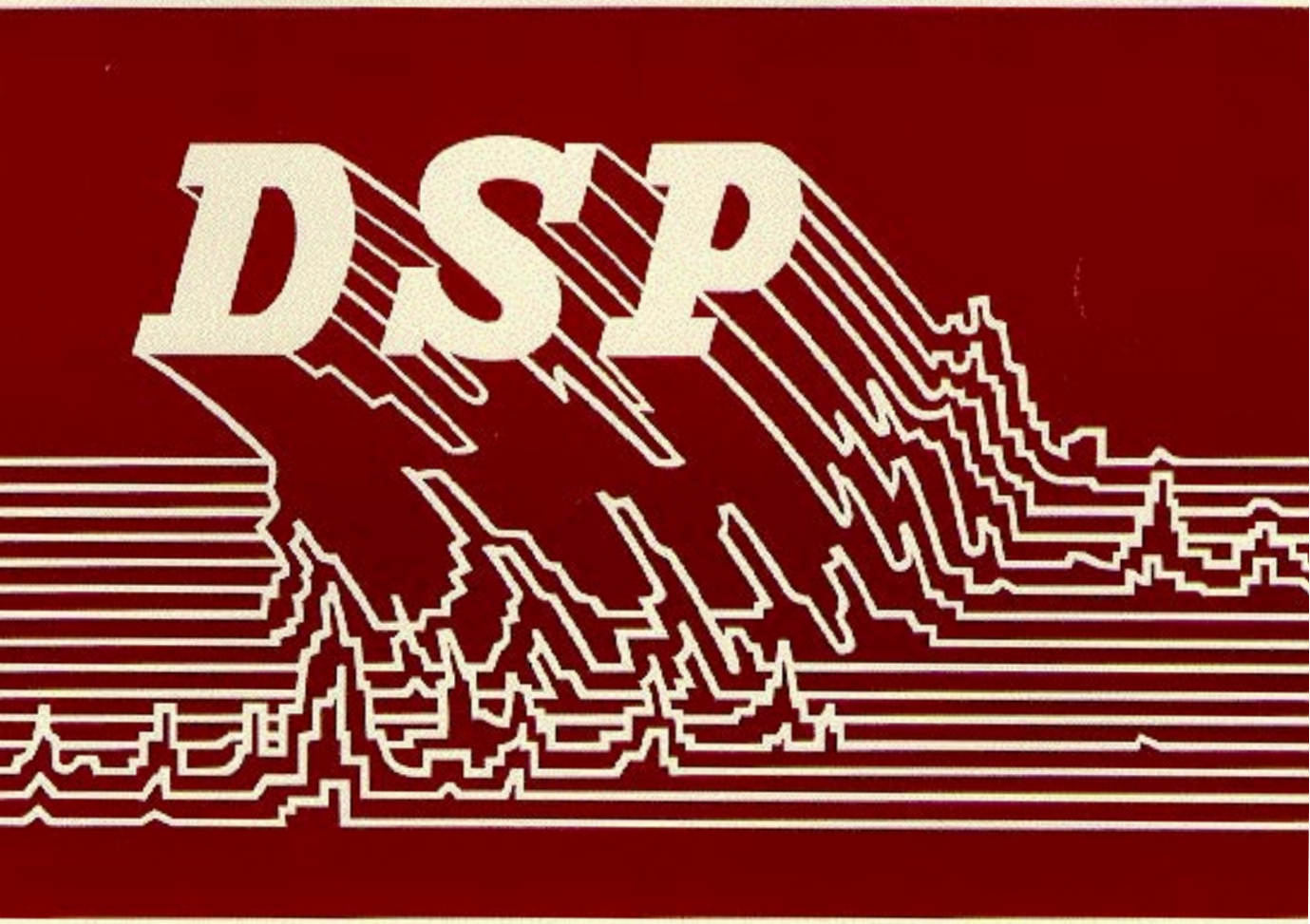
APR2/D

Digital Stereo 10-Band Graphic Equalizer Using the DSP56001



MOTOROLA

Digital Stereo 10-Band Graphic Equalizer Using The DSP56001

INTRODUCTION

A stereo 10-band graphic equalizer implemented with the DSP56001 is discussed in this application note. The theory behind the infinite impulse response (IIR) algorithm used to perform the bandpass filtering is examined briefly. The connection between the analog passive filter and the digital IIR filter is presented. Similar analytical techniques are employed to characterize the filter response in both the analog and digital domains. Exact algebraic expressions are derived relating center frequency (f_0), quality factor (Q), gain (G), and phase angle (ϕ) to the IIR coefficients. For frequencies much lower than one-half of the sample frequency, the gain and phase equations reduce to a simple form (symmetric over the logarithm of frequency), which is equivalent to those describing the resistor-capacitor-inductor (RCL) network of Figure 1(a). The IIR coefficients are easily obtained by evaluating these formulas based on a quality factor, center frequency, and center frequency (resonant frequency) gain (G_0). A graphic equalizer is composed of parallel filters with identical quality factor and center frequencies based on equal intervals of the log of frequency. Intervals differing by a factor of two (octaves) and ranging from 31 Hz to 16 kHz were chosen, thus covering most of the audio spectrum.

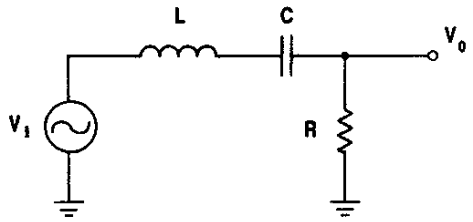


Figure 1(a). Analog Passive RCL Bandpass Network

The effect of coefficient quantization is shown to be severe for word size less than 24 bits. An approximate formula is derived to calculate the allowed quantized frequency bands as a function of the coefficient word length. The lowest allowed band for 16-bit coefficient word length is 54.8 Hz (using the transfer function of Equation (4)); whereas, it is 3.4 Hz for 24-bit cofficient word length. These results agree with those predicted by the "Filter Design & Analysis System" filter design software.[1]

A hardware interface to a SONY 650ESD compact disk player utilizing the DSP56000/1's synchronous serial interface (SSI) port is described, thus yielding an all-digital graphic-equalizer system (i.e., analog-to-digital converter (ADC) and digital-to-analog converter (DAC) are not needed). This project demonstrates the use of the SSI port for receiving and transmitting data, the implementation of a set of parallel second-order IIR filters, and an example of a low-cost, memory-port bootstrap EPROM/DSP56001 system.

FILTER ANALYSIS

The fundamental filter used is a bandpass, single-response pole, second-order IIR filter. (Even though two poles appear inside the unit circle in the z-plane, only one pole lies between 0 and π, the region of valid operation.) The filter center frequency, f_0, and the bandwidth, Δf, can be adjusted through software control. The primary advantage of this second-order digital filter is the minimal number of instructions (a total of four adds and multiplies) needed to implement the algorithm.

THE PASSIVE SERIES RESONANT NETWORK

To describe the characteristics of the digital filter, the equivalent analog passive RCL bandpass network (Figure 1(a)) will be examined. By straightforward voltage divider analysis, the transfer function can be written as follows:[2]

$$\frac{V_o}{V_i} = \frac{R}{R + j(\Omega L - 1/\Omega C)} \tag{1a}$$

where $\Omega = 2\pi f$. The gain is the magnitude of Equation (1a):

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

where $\Omega_0 = (LC)^{-1/2}$ and $Q = \Omega_0 L/R$. The phase angle, ϕ, is found by taking the ratio of the imaginary to real parts of the transfer function of Equation (1a):

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

The equivalent s-plane expression is calculated by substituting $s = j\Omega$:

$$H(s) = \frac{Rs}{Rs + Ls^2 + 1/C} \tag{2}$$

An op-amp active-filter circuit with essentially the same response as the passive RCL network is shown in Figure 1(b).[3] This active filter has several advantages over the passive network in that it eliminates the inductor; it is essentially isolated from input and output loading and can provide signal gain to the system.

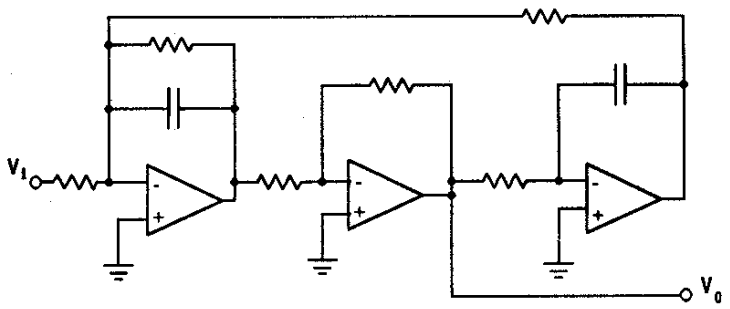


Figure 1(b). Op-Amp Active Bandpass Filter

A digital-transfer-function representation of Equation (2) may be obtained by applying the bilinear transformation:[4,5]

$$s = \frac{2}{T}\left(\frac{1-z^{-1}}{1+z^{-1}}\right) \tag{3a}$$

where $z = e^{j\theta}$, $\theta = \omega T$, and T is the sample period (see Figure 2). Equation (3a) can also be expressed as follows:

$$\Omega = \frac{2}{T}\tan(\theta/2) \tag{3b}$$

using the definitions of s and z. Substituting Equation (3a) into Equation (2) yields the z-plane transfer function:

$$H(z) = \frac{\alpha(1-z^{-2})}{1/2 - \gamma z^{-1} + \beta z^{-2}} \tag{4}$$

where the coefficients, α, β, and γ, are related to R, C, and L by

$$\alpha = \frac{RT/2}{T^2/2C + RT + 2L} \tag{5a}$$

$$\gamma = \frac{2L - T^2/2C}{T^2/2C + RT + 2L} \tag{5b}$$

$$\beta = \frac{T^2/4C - RT/2 + L}{T^2/2C + RT + 2L} \tag{5c}$$

The nonlinear relationship between the analog domain frequency, Ω, and the digital domain frequency, ω, as shown by Equation (3b), is often referred to as the frequency warp.[6] As the frequency starts from zero, both Ω and ω are approximately equal, since the $\tan\theta \approx \theta$ for small angles. However, as Ω approaches infinity, ω approaches $2\pi f_s/2$. In this particular

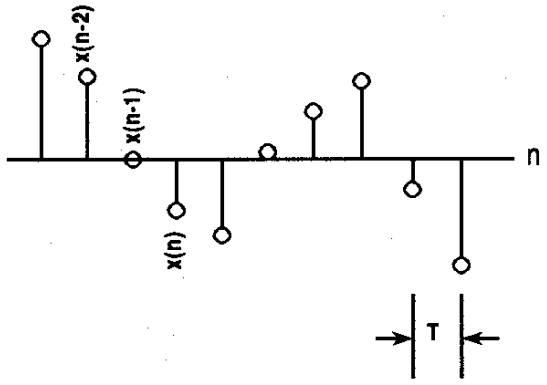


Figure 2. Standard Sampled Data, x(n)

application, most of the interesting and useful frequencies satisfy the small angle approximation (SAA) where $\theta < \pi/4$. Thus, using the SAA simplifies the digital analysis, and a direct correspondance to the analog RCL network of Figure 1(a) is established. Although the SAA indeed simplifies the analysis, it must be used very carefully in derivations because the nature of this IIR filter depends on very small differences of numbers. If not used correctly, the SAA can mask out these differences and give totally erroneous results.

THE DIFFERENCE EQUATION

To implement the transfer function from Equation (4) as an IIR filter, it is first necessary to transform it to a difference equation in the discrete time domain. In this form, the filter can be directly implemented in software. Applying the inverse z-transform operator, Z^{-1}, to Equation (4), yields the following:[7]

$$Z^{-1}\{H(z)\} = Z^{-1}\{Y(z)/X(z)\} \tag{6a}$$

$$Z^{-1}\{Y(z)[1/2 - \gamma z^{-1} + \beta z^{-2}]\} = Z^{-1}\{X(z)[\alpha(1 - z^{-2})]\} \tag{6b}$$

The time-delay property of the z-transform can be stated as follows:

$$X(z)z^{-m} = Z\{[x(n-m)]\} \tag{7}$$

where n is the discrete time index variable associated with continuous time sampling at a rate T (see Figure 2). Evaluating Equation (6b), using the property of Equation (7), gives the final IIR difference equation:

$$y(n) = 2\{\alpha[x(n) - x(n-2)] + \gamma y(n-1) - \beta y(n-2)\} \tag{8}$$

The coefficients, α, β, and γ, in the difference equation (Equation (8)) are used to adjust the filter response (gain and phase as a function of frequency). The representation of the time-varying data is based on standard notation used in digital filter theory.[8,9,10] Thus, x(n) is the current sampled data represented as an N-bit signed fraction; x(n − 1) is the previous data word; and x(n − 2) is the data word previous to x(n − 1). The n is the time index where it is assumed that the sample period, T, is constant and is related to the sample frequency, f_s, by $T = 1/f_s$. For example, the time between x(n) and x(n − 2) is 2T. Sampled values of the input signal are only collected at integral multiples of T (i.e., the x(n)'s are standard sampled/digitized data).

The y(n) is similar to the input data, x(n), but is instead the output data from the difference equation algorithm. As before, y(n) is the current output value; y(n − 1) is the previous value; and y(n − 2) is the value previous to y(n − 1). Even though it is assumed that the input data is a signed fraction (a number between one and minus one), the y(n)'s can be greater than one (or less than minus one) unless scaling is performed to prevent this overflow condition. The choice of fractional values is not essential to proper behavior of the IIR algorithm, but it is a great convenience to both the analysis and the software implementation on the DSP56001.

The coefficients, α, β, and γ, in the difference equation (Equation (8)) are also fractional values (i.e., between one and minus one). As it will later be shown (Equation 15 with $G_0 = 1$), scaling at the output can be controlled by imposing the following condition on two of these three coefficients:

$$\alpha = 1/4 - \beta/2 \tag{9}$$

This formula guarantees that, at the center frequency, the gain is one and the phase difference is zero. In this case, the bandpass filter acts as an attenuation filter (and phase shifter) for all frequencies other than the center frequency. Limiting the gain to one and the input to a fraction scaled to a maximum of one does not always prevent overflow at the output. For example, a square wave with an amplitude excursion from -1 to 1 has a fundamental sine component with an amplitude of $4/\pi$, which is greater than one by about 30 percent. Therefore, care must exercised when specifying gain and dynamic range for filters to prevent distortion.

RESPONSE OF THE DIGITAL FILTER

The gain and phase response can be calculated solely from Equation (4). (The advantage of complex numbers is demonstrated in that both gain and phase information are present in the transfer function.) By definition, the gain is the absolute magnitude of H(z). In the RCL circuit, the gain is simply the ratio of the resistance to the magnitude of the total complex impedance. The ratio of the real to imaginary components of impedance is equal to the tangent of the phase. Likewise, the ratio of real to imaginary components of H(z) is equal to the tangent of the phase for the digital case.

Euler's identity is implemented to ease the calculation of gain, $G(\omega)$, and phase, $\theta(\omega)$:

$$e^{j\theta} = \cos\theta + j\sin\theta \tag{10}$$

The transfer function (Equation (4)) then becomes:

$$H(e^{j\theta}) = \frac{\alpha(e^{j2\theta} - 1)}{\frac{1}{2}e^{j2\theta} - \gamma e^{j\theta} + \beta} = \frac{\alpha(\cos 2\theta - 1) + j\sin 2\theta}{(\frac{1}{2}\cos 2\theta - \gamma\cos\theta + \beta) + j(\frac{1}{2}\sin 2\theta - \gamma\sin\theta)} \tag{11}$$

where $\theta = \omega T$. For example, $\theta = \pi/2$ would correspond to $f = f_s/4$ (since $\omega = 2\pi f$ and $T = 1/f_s$). If $f_s = 44.1$ kHz (the standard for compact disc digital audio), then $\theta = \pi/2$ would be a frequency of 11.025 kHz.

The filter gain is found by evaluating the following expression:

$$G(\omega) = [H(e^{j\theta})H^*(e^{j\theta})]^{1/2} \tag{12}$$

and, after some algebraic and trigonometric manipulations, becomes:

$$G(\omega) = \frac{2\alpha\sin\theta}{\{[(\frac{1}{2} - \beta)\sin\theta]^2 + [(\frac{1}{2} + \beta)(\cos\theta - \cos\theta_0)]^2\}^{1/2}} \tag{13}$$

where

$$\cos\theta_0 = \gamma/(\frac{1}{2} + \beta) \tag{14}$$

is the filter center frequency.

Examination of the gain in Equation (13) shows several important features:

- The gain, $G(\omega)$, is proportional to α.
- The gain at the center frequency, ω_0, is

$$G_0 = 2\alpha/(\frac{1}{2} - \beta) \tag{15}$$

as previously noted in Equation (9).

- The bandwidth is adjusted by β (that also effects the center frequency as shown by Equation (14)).
- Equation (13) is symmetric (neglecting the zero at $\theta=\pi$) on a logarithmic scale. This characteristic of the gain can be seen more easily by taking the SAA of Equation (13) where

$$\sin\theta \approx \theta \tag{16}$$

and

$$\cos\theta \approx 1-\theta^2/2 \tag{17}$$

so that

$$G_a(\omega) = \frac{G_0}{\left[1+\left(\frac{\frac{1}{2}+\beta}{\frac{1}{2}-\beta}\right)^2\left(\frac{\theta_0^2-\theta^2}{2\theta}\right)^2\right]^{\frac{1}{2}}} \tag{18}$$

Subsitution of $\theta=k\theta_0$ or $\theta=\theta_0/k$ yields equivalent values of gain, thus proving the gain is symmetric over the log of frequency. The subscript "a" denotes that the SAA was used in that expression.

The phase shift, $\phi(\omega)$, is found from the ratio of the imaginary to real part of H(z) from Equation (11):

$$\tan\phi = \frac{\mathrm{Im}[H(e^{j\theta})]}{\mathrm{Re}[H(e^{j\theta})]} \tag{19}$$

After same algebraic and trigonometric manipulations, Equation (19) can be written as

$$\tan\phi = \frac{(\frac{1}{2}+\beta)(\cos\theta-\cos\theta_0)}{(\frac{1}{2}-\beta)\sin\theta} \tag{20}$$

Applying the SAA simplifies the previous result:

$$\tan\phi_a = \frac{(\frac{1}{2}+\beta)(\theta_0^2-\theta^2)}{(\frac{1}{2}-\beta)2\theta} \tag{21}$$

The SAA can be used to approximate the filter center frequency, θ_0, from Equation (14):

$$\theta_{0_a} = \left[\frac{1+2\beta-2\gamma}{\frac{1}{2}+\beta}\right]^{\frac{1}{2}} \tag{22}$$

The SAA is accurate within a few precent for angles up to $\pi/4$. (This SAA corresponds to a filter frequency of $f<f_s/8$.)

The bandwidth of the filter is most easily determined from Equation (18). Generally, two frequencies are considered, one on each side of the center frequency, θ_0. The gain at each of the frequencies, θ_1 and θ_2, is equivalent and is commonly chosen so that the value of gain is $G_0/\sqrt{2}$. Since $20\log(1/\sqrt{2})=-3$, the bandwidth can be defined as $\Delta\theta=\theta_2-\theta_1$, where $G(\theta_1)=G(\theta_2)=G_0/\sqrt{2}=-3$ dB of the center frequency gain. As previously noted, $\theta_1=\theta_0/k$ and $\theta_2=k\theta_0$ for a filter symmetric about the center frequency over the log of frequency.

The Q of the filter in such a case is as follows:

$$Q = \frac{\theta_0}{\Delta\theta} = \frac{\theta_0}{k\theta_0 - \theta_0/k} = \frac{k}{k^2-1} \tag{23}$$

where $k>1$. Since, by definition, the bandwidth is determined at the frequencies corresponding to a gain of $G_0/\sqrt{2}$, using Equation (18), the following term is equal to one:

$$\left(\frac{1/2+\beta_a}{1/2-\beta_a}\right)^2 \quad \left(\frac{\theta_1^2-\theta_0^2}{2\theta_1}\right)^2 = 1 \tag{24}$$

and using Equation (23) to solve for β_a in terms of Q yields

$$\frac{1/2+\beta_a}{1/2-\beta_a} = \frac{2}{\theta_0}\left(\frac{k}{k^2-1}\right) = 2Q/\theta_0 \tag{25}$$

Rearranging terms results in the final form:

$$\beta_a = \frac{Q-\theta_0/2}{2Q+\theta_0} \tag{26}$$

where $\theta_0 = 2\pi f_0/f_s$. The subscript "a" is used to denote that the SAA was used in this derivation (i.e., by definition of Q from Equation (18)).

Solving for γ in Equation (14) gives

$$\gamma = (1/2+\beta)\cos\theta_0 \tag{27}$$

For unity gain at the center frequency, α (Equation (15)) becomes

$$\alpha = (1/2-\beta)/2 \tag{28}$$

Equation (18) now simplifies to the following equation for gain:

$$G_a(\omega) = \left[1+Q^2\left(\frac{\theta_0^2-\theta^2}{\theta_0\theta}\right)^2\right]^{-1/2} \tag{29}$$

Equation (21) becomes

$$\phi_a(\omega) = \tan^{-1}\left[Q\left(\frac{\theta_0^2-\theta^2}{\theta_0\theta}\right)\right] \tag{30}$$

Equations (13), (14), (15), and (20) provide a complete, theoretically accurate, concise description of the digital filter response described by the difference equation (Equation (8)). The coefficients, α and γ, can be found from β, θ_0, and G_0. β must be determined from the gain (Equation (13)) by picking $G(\theta_1)$ and θ_1, then finding the value of β from the equality. These four equations are exact and can be used over the entire frequency range from 0 to π.

Equations (26) through (30) provide a simplified set of formulas that are reasonably accurate for $f<f_s/8$ and for unity gain at the center frequency.

ANALYSIS CONSTRAINTS OF THE BILINEAR TRANSFORMATION

THE PASSIVE SERIES RESONANT NETWORK shows how to determine the IIR coefficients from the RCL values of a passive network filter based on the bilinear transformation. This technique is very powerful, especially if the frequencies of interest are much lower than the sample frequency (as is often true in digital audio applications) so that the SAA can be used. The resonant and cutoff frequency and quality factor, Q, of most RCL networks are known or can be easily determined. **THE DIFFERENCE EQUATION** discusses how to convert the transfer function to a difference equation, which is the final form (for software implementation) of the digital IIR filter. The relationship connecting the R, L, and C values of the analog filter to the coefficients α, β, and γ of the digital filter (from Equations 5a thru 5c) holds true only for frequencies where the SAA is valid. This frequency range makes up the linear region of the bilinear transformation where (from Equation 3b) $\tan\theta \approx \theta$. In this case, a direct correspondence between the response of almost any RCL network and an IIR filter's coefficients can be established, as previously demonstrated for the bandpass filter network. This technique lends itself to audio applications because the response of a network is usually described over the log of frequency. The audio range is basically logarithmic, thus the SAA applies to most of the range of interest since $f_s/8$ is very close to $f_s/2$ on a log scale (see Figures 11 and 12).

COEFFICIENT QUANTIZATION

Coefficient quantization is an effect that depends solely on the word length of the filter coefficients. Equations (13), (14), and (15) yield values for α, β, and γ for given values of center frequency and bandwidth. These formulas are exact. However, the word size of the variables used to represent the coefficients in the filter algorithm are of finite length. Therefore, only certain discrete values of center frequency, bandwidth, and resonant frequency gain are obtainable.

To analyze the effects of coefficient quantization in this particular digital filter, let N be the number of bits used to represent data in the algorithm. Assuming that the coefficients are fractions, the smallest number that can be represented is therefore (see reference 2):

$$\delta = 2^{-(N-1)} \tag{31}$$

Using Equation (31), β and γ can be represented as follows:

$$\gamma = 1 - n\delta \tag{32a}$$

$$\beta = \tfrac{1}{2} - m\delta \tag{32b}$$

since $\beta < \frac{1}{2}$ and $|\gamma K|$. This can be easily seen by evaluating the zeros of the transfer function (Equation 4) and then calculating the magnitude of that complex number. The resulting value is the distance from the origin to the pole in the complex plane and is equal to 2β. Now, since the poles must lie within the unit circle[11] $\beta < \frac{1}{2}$. Using Equation (14), it can be seen that $|\gamma| < 1$. The integers n and n take on values from 1 to 2^{N-1}. Equation (22) can be written as

$$\theta_0 = \left[\frac{1 + 2(\tfrac{1}{2} - m\delta) - 2(1 - n\delta)}{\tfrac{1}{2} + (\tfrac{1}{2} - m\delta)} \right]^{1/2} = \left[\frac{2(n-m)\delta}{1 - m\delta} \right]^{1/2} \tag{33}$$

By inspection, the lowest nonzero value of θ_0 is with n = 2 and m = 1. The lowest obtainable frequency is then

$$f_0 = \theta_0 f_s/2\pi = \frac{1}{2\pi}\left[2\left(\frac{\delta}{1-\delta}\right)\right]^{1/2} = f_s 2^{-N/2}/\pi \quad (34)$$

Assuming f_s = 44.1 kHz, the lowest obtainable frequency for 16 bits is 54.8 Hz; for 24 bits, it is 3.4 Hz. Clearly, 16 bits does not yield the coefficient accuracy needed to implement filter responses in the low-frequency bands (i.e., 20 to 200 Hz) for audio applications. The 24-bit word length of the DSP56001 is more than enough to ignore coefficient-quantization errors.

HARDWARE DESCRIPTION

The basic hardware description of the SSI and compact disk player interface and a layout of the system are presented in the following paragraphs.

SSI AND COMPACT DISK PLAYER INTERFACE

Figure 3(a) shows the data signals tapped from the compact disk player (CDP). Three of the four signals are clocks. The data line is cut to form two data signals, DATOUT and DATIN. The format of the data is stereo multiplexed (i.e., left-right-left-right. . .). The data length for each channel is 24 bits: 16 of which are significant; the upper eight are sign extended. Since the audio sample frequency used on the CDPs is 44.1 kHz, the bit clock (BITCLK) runs at 48 × 44.1 kHz = 2.1168 MHz. The word clock (WRDCLK) runs at 1/24 of the BITCLK, which is 88.2 kHz; the left-right clock (LRCLK) runs at 44.1 kHz. These signals are tapped from the CDP before the up-samplig section. The format of these signals varies among manufacturers and models; therefore, this particular application should only be used as an example.

The SSI consists of six pins, which are the upper bits of port C (PC3 through PC8):

PC3	SC0	Serial Control 0
PC4	SC1	Serial Control 1
PC5	SC2	Serial Control 2
PC6	SCK	SSI Serial Clock
PC7	SRD	SSI Receive Data
PC8	STD	SSI Transmit Data

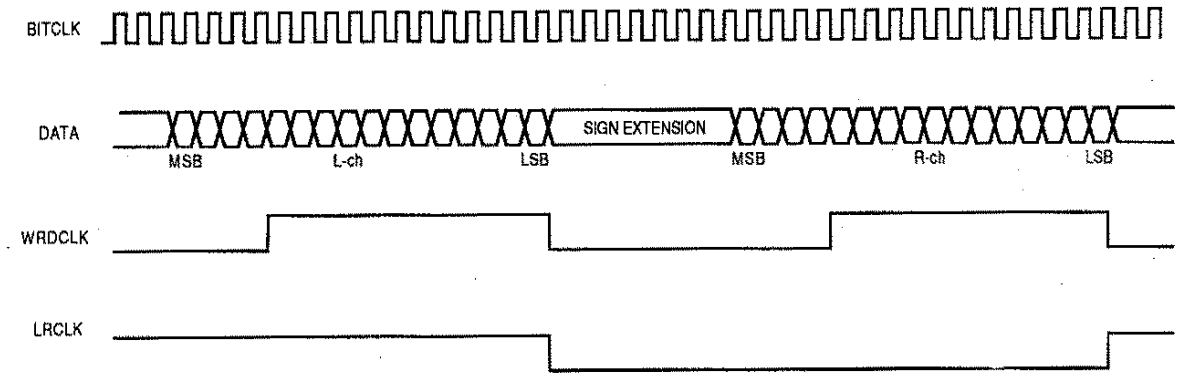


Figure 3(a). Signals Tapped from CDP

Four memory-mapped hardware registers control the configuration and operation of the SSI port:

Port C Control Register	[X:$FFE1]
Control Register A	[X:$FFEC]
Control Register B	[X:$FFED]
SSI Status Register	[X:$FFEE]

The port C control register (PCC) is used to select port C pins as SSI or general-purpose input/output (I/O). If all of the SSI pins are to be used, bits 3 through 8 are set (i.e., PCC = $1F8). The DPS56000/1 code for setting the PCC is as follows:

```
MOVEP       #$1F8,X:FFE1
```

where X:$FFE1 is the memory address of the PCC register. In this particular example, shown in Figure 3(b), the CDP's WRDCLK is connected to SC2, which is configured as a frame sync. A frame then consists of one 24-bit word. The LRCLK is connected to SC0 as a serial input flag whose status can be determined by polling the SSI status register, bit 0 (IF0), shown in Figure 4.

The immediate value of #$6000 is loaded into control register A for this particular implementation. This sets the word length to 24, ignores the clock prescaler (because an external clock is used), and selects the normal mode.

Control register B is loaded with #$B200, which chooses synchronous mode, enables the receive and transmit shift registers, arms the receive interrupt, and sets the serial pins to inputs (SC0, SC1, SC2, and SCK). Note that the frame sync length is set to a word frame, which may at first appear to violate the SSI operation. However, since the frame sync is external, the actual sampling of the frame occurs during the first SSI clock cycle of the frame sync clock.

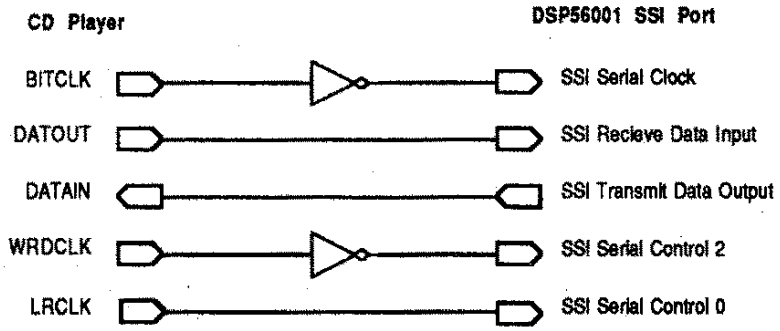


Figure 3(b). CDP to SSI Interface

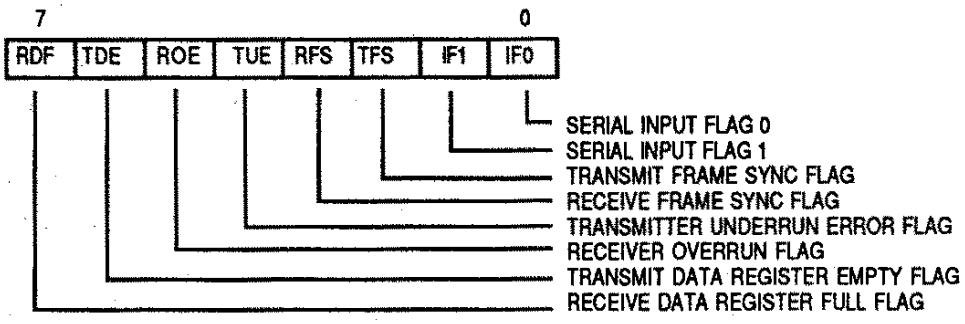


Figure 4. Read-Only SSI Status Register (X:$FFEE)

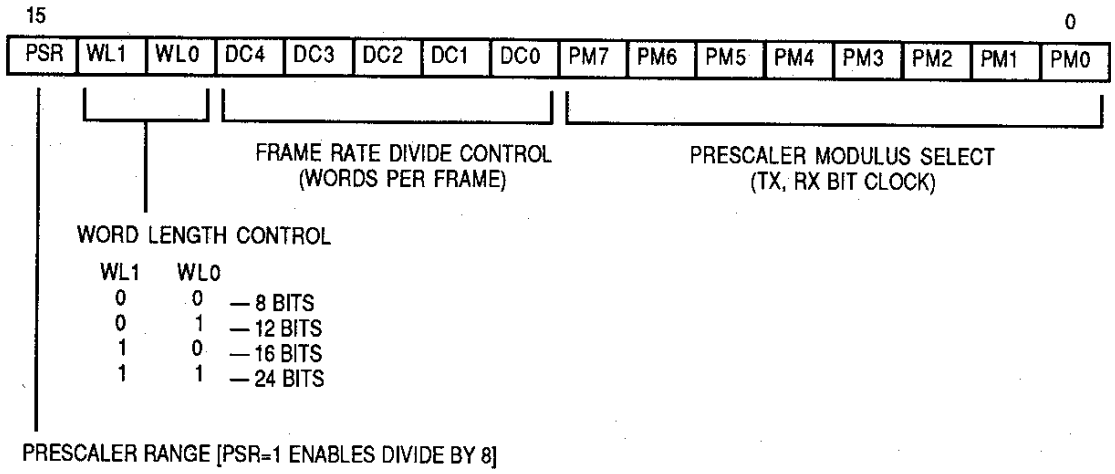


Figure 5. SSI Control Register A (X:$FFEC)

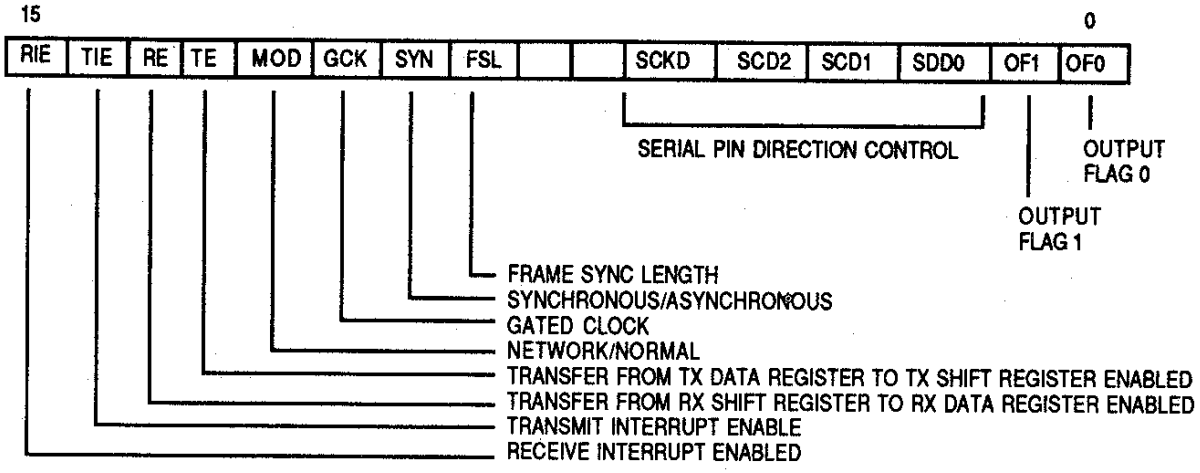


Figure 6. SSI Control Register B (X:$FFED)

BLOCK DIAGRAM

The block diagram of Figure 7 shows the basic hardware layout of the system. The EPROM's data is loaded into the DSP56001 P:RAM upon powerup (or reset). Slide potentiometers, one for each frequency band, are multiplexed into an 8-bit ADC. The value of the voltage from each slide potentiometer is proportional to the gain used to multiply the particular bandpass filter response. Figure 8 is a complete schematic diagram of the system. The analog multiplexers used are three 4051, 8:1 MUXs. The address is generated by latching a 7-bit address with a 74LS374 octal D-type latch off the DSP56001's data bus. The ADC read enable is generated by ANDing the read ($\overline{RD}$) and data strobe ($\overline{DS}$) from the DSP56001; the MUX select enable is generated by ANDing write ($\overline{WR}$) and data strobe ($\overline{DS}$). With this particular system, only 20 slide potentiometers are used for setting frequency response; one is used for master volume. Therefore, a total of 21 channels of the MUX are used, resulting in only three of the four 4051s being used.

Data received at the SSI port initiates an interrupt every 11.3 ms (the inverse of 88.2 kHz). This is the rate at which 24 bits are read, corresponding to a complete word of data from the left or the right channels. SC0 (the LRCLK) is then polled to determine which channel

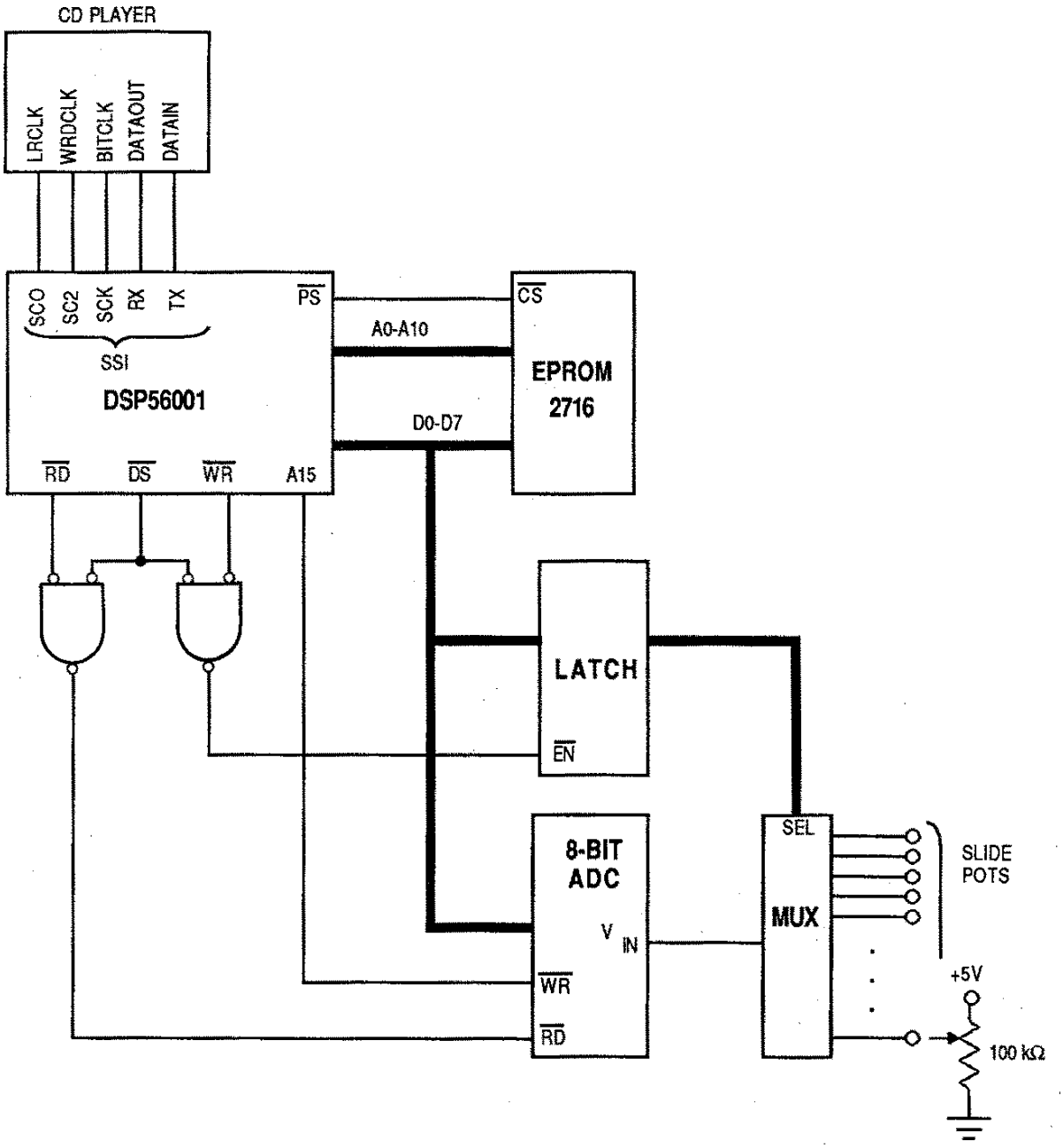


Figure 7. Block Diagram of DPS56001 Graphic Equalizer

needs processing. (In an optimized configuration, the LRCLK can also be used as the frame sync, using the network mode, where a frame consists of two 24-bit words, thus eliminating the need for the WRDCLK.) The received word is processed by the 10 parallel IIR filters, corresponding to the 10 left slide potentiometers or the 10 right slide potentiometers, and then transmitted back to the CDP. The CDP's DAC receives the data a total of two words later; thus, a latency of two sample periods has been introduced. This latency does not cause any undesirable effects since everything becomes delayed by the same amount. Jumpering the DATIN to DATOUT will completely bypass the DSP56001 system and eliminate the delay (useful as the standard bypass mode of an audio equalizer). Equivalently,

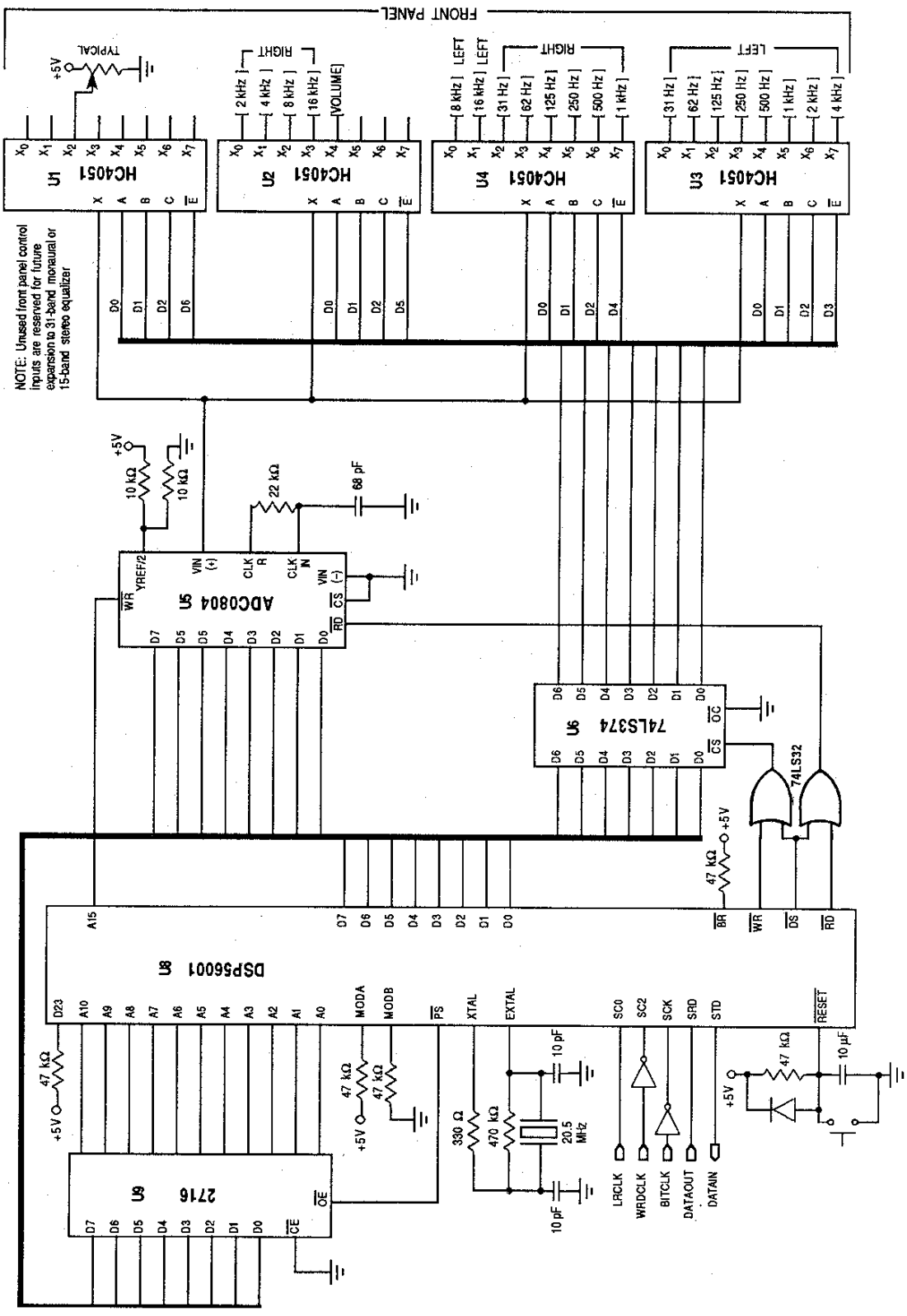


Figure 8. Schematic Diagram of DSP56001 Graphic Equalizer

later; thus, a latency of two sample periods has been introduced. This latency does not cause any undesirable effects since everything becomes delayed by the same amount. Jumpering the DATIN to DATOUT will completely bypass the DSP56001 system and eliminate the delay (useful as the standard bypass mode of an audio equalizer). Equivalently, the bypass mode can be achieved by receiving the data at the SSI and then transmitting the same data without processing (also a useful technique for debugging).

ALGORITHM AND SOFTWARE

The following paragraphs discuss the filter algorithm and the final implementation of the DSP56000/1 code.

FILTER ALGORITHM

Figure 9(a) shows a high-level implementation of the difference formula of Equation (8). The array indexes I, J, and K are modulo 3 (i.e., they will only contain the values 0, 1, and 2). The current value and previous two values of data are stored in a cyclic buffer (arrays X and Y). Figure 9(b) shows the input array, X, for the first four sample times. The output array, Y, is treated in a similar fashion.

```
        I = 0

        A = ALPHA       X(0) = 0      Y(0) = 0
        B = BETA        X(1) = 0      Y(0) = 0
        C = GAMMA       X(2) = 0      Y(2) = 0

100     READ ADC

        X(I) = ADC

        J = I-2         IF J < 0 THEN J = J+3
        K = I-1         IF K < 0 THEN K = K+3

        Y(I) = 2 * (A * (X(I) - X(J)) + C * Y(K) - B * Y(J))

        OUTPUT Y(I) TO DAC

        I = I + 1       IF I > 2 THEN I = 0
        GOTO 100
```

Figure 9(a). IIR Filter Algorithm (Equation (8)) Coded in a High-Level Language

t = 1	
I = 0	x(n) = ADC1
J = 1	x(n-2) = 0
K = 2	x(n-1) = 0

t = 2	
K = 0	x(n-1) = ADC1
I = 1	x(n) = ADC2
J = 2	x(n-2) = 0

t = 3	
J = 0	x(n-2) = ADC1
K = 1	x(n-1) = ADC2
I = 2	x(n) = ADC3

t = 4	
I = 0	x(n) = ADC4
J = 1	x(n-2) = ADC2
K = 2	x(n-1) = ADC3

Figure 9(b). IIR Data Arrays During First Four Iterations

The equivalent DSP56000 code to perform the IIR filter difference equation is shown as follows:

```
MOVE    X:ADC,B                                         ;Read external ADC.
MOVE    B,Y:(R5)                                        ;Store ADC value in
                                                        ; cyclic buffer as x(n).
CLR     B           X:(R0)+,X0  Y:(R4)+,Y0              ;Get β and y(n-2).
MAC     -X0,Y0,B    X:(R0)+,X0  Y:(R5)+,Y0              ;B=B-βy(n-2).
MAC     X0,Y0,B                 Y:(R5),Y0               ;B=B+αx(n).
MAC     -X0,Y0,B    X:(R0)+,X0  Y:(R4)+,Y0              ;B=B-αx(n-2).
MACR    X0,Y0,B                                         ;B=B+γy(n-1).
MOVE    B,Y:(R4)-                                       ;Store result in cyclic
                                                        ; buffer as y(n).
MOVE    B,X:DAC                                         ;Write y(n) to external
                                                        ; DAC.
```

The code would often be an interrupt routine where the interrupt is activated by an external sample rate clock (such as a CDP). The first two instructions read the data from a memory-mapped ADC and store the data in the x(n) array. The next five instructions perform the four multiply/accumulates necessary to implement the difference equation. The last two instructions store the result in y(n) and the memory-mapped DAC. The index registers, R0, R4, and R5, point to the coefficient table, y(n), and x(n), respectively (see Figure 10). Index registers, R0, R4, and R5, are modulo 3 as are the indexes I, J, and K in the high-level language example.

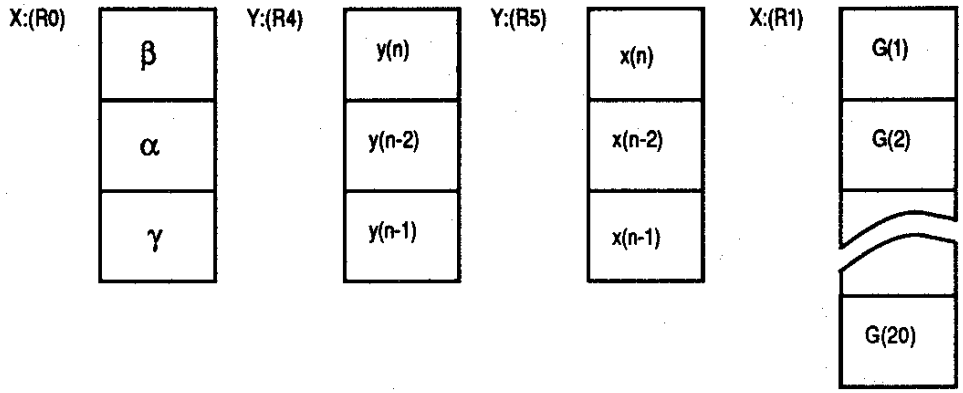


Figure 10. IIR Filter Data Structures

The theoretically minimum execution time is determined by the number of terms in the difference equation (four in this case), which equate to the number of multiply/accumulates necessary to calculate y(n). The actual execution time depends how the data is stored, indexed, and manipulated, which is largely application dependent. The following code demonstrates the same filter done in four instructions (neglecting I/O). The tradeoff is in the number of data ALU registers used (see reference 5).

```
MOVE    X:ADC,Y1

MPY     -X0,Y0,B    X:(R0)-,X0  B,Y0                    ;B=-βy(n-2)
MAC     X0,Y0,B     X:(R0)-,X0  Y1,Y:(R5)+              ;B=B+γy(n-1)
MAC     X0,Y1,B                 Y:(R5),Y1               ;B=B+αx(n)
MACR    -X0,Y1,B    X:(R0)-,X0                          ;B=B-αx(n-2)

MOVE    B,X:DAC
```

Register Y1 is now used in addition to X0 and Y0. Also, X0, Y0, and B must not be modified before the next call to this routine. Numerous ways exist to code the difference equation; it cannot be done in fewer instructions then the number of terms. The details are then up to tradeoff considerations for the remainder of the software system.

FINAL IMPLEMENTATION OF THE DSP56000/1 CODE

To construct a graphic equalizer, a set of 10 IIR bandpass filters are used in parallel for each of the stereo audio channels. Using 1 kHz as a starting point, successively dividing down by two to 31 Hz, and successively multiplying by two up to 16 kHz, a 10-band octave response is generated. The coefficients for the lower eight bands are easily determined by means of Equations (26) through (30). Equations (13), (14), (15), and (20) are used for the highest two bands. Unity gain at the center frequency is chosen, and a quality factor, Q, is arbitrarily selected to be 1.4. Table 1 lists the 10 sets of coefficients used to generate the response curves shown in Figure 11. The pole/zero plot of Figure 12 shows the location of the poles for all 10 bands. The response curve is not the final response of the system, but shows the response of each individual filter superimposed on the same graph. The total audio response would be the summation of each filter, where the gain of each can independently vary from g_{LO} to g_{HI} (where $g<0$ represents a phase change of π).

Table 1. 10-Band Bandpass IIR Coefficients

Center Frequency	IIR Coefficients		
	α	β	γ
31	0.000787462865	0.498425074	0.998415336
62	0.00157244917	0.496855102	0.996816209
125	0.00316016172	0.493679677	0.993522095
250	0.00628062774	0.487438745	0.986812425
500	0.0124054279	0.475189144	0.972715729
1000	0.0242101804	0.451579639	0.941937749
2000	0.0461841095	0.407631781	0.871031797
4000	0.0845577687	0.330884463	0.699565951
8000	0.1199464	0.2601072	0.3176087
16000	0.159603	0.1800994	−0.4435172

The primary difference between the analog filter versus the digital response is the zero at $f_s/2$. Anything above this frequency is not allowed, because it would violate the basic sampling theorem that says all frequencies must be lower than half of f_s to prevent aliasing. This zero causes the higher frequency bands to be asymmetric over the logarithm of frequency. The SAA formulas derived previously must be used cautiously, or not at all, at these higher frequencies ($\theta>\pi/4$ or $f>f_s/8$). The exact formulas (Equations (13), (14), (15), and (20)) should be used at the high frequencies. Mathematically, this zero is the consequence of mapping $s=j\Omega$, the infinite axis, into the finite unit circle of the z-plane (Equation (3)). This zero at $\theta=\pi$ is the same zero found at $\Omega=$ infinity with the analog filter. As evidenced by the frequency warping (Equation (3b)), both the analog frequency, Ω, and the digital frequency, ω, start at zero, but as Ω approaches infinity, f approaches $f_s/2$. However, with the SAA, both the analog and digital frequencies are equal, since $\tan\theta\approx\theta$ for θ small.

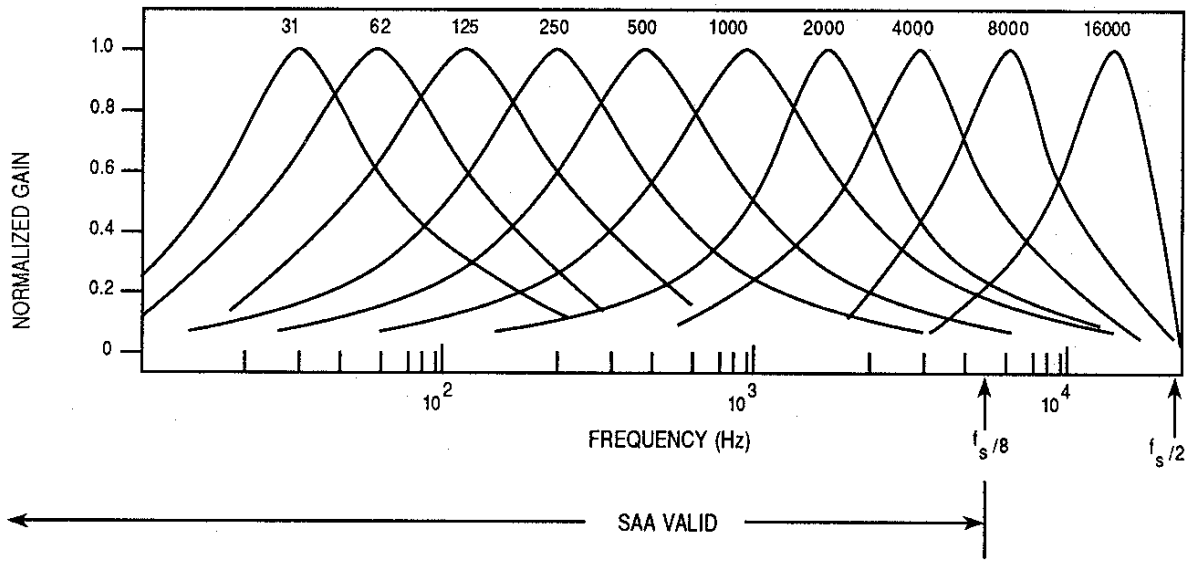


Figure 11. 10-Band Bandpass IIR Filter Response

Figure 13 shows the total algorithmic system for the graphic equalizer. Each block labeled F_i is one IIR filter; g_i is the gain coefficient set by the i^{th} potentiometer that scales F_i. Because of the direct through-path, the condition of all g_i's equal to zero produces a flat response (the output is due only to the direct path from the input). The passthrough gain at 2^{-2} is required to scale the g_i's so that g_i's a fraction. All g_i's equal to one will essentially produce a gain of five ($\sim +14$ dB). However, this gain will not be exactly flat since ripple is produced in the response curve due to the finite number and quality factor, Q, of each band (also true of analog equalizers). All g_i's set to -0.2 will produce an overall gain of 0.2 (~ -14 dB); again there will be ripple in the response.

The scaling of the data, shown in Figure 13, is done for several reasons. First, the multiplication of the input data by 2^8 normalizes the data (the input is 16-bit data sign-extended to 24 bits). This technique will minimize word-length and roundoff error effects in the IIR blocks. Second, the left shift by two (2^2) after volume scaling compensates for the pass-through gain of 2^{-2} (normalizes the data) so that anything above 24 bits will be limited rather than truncated. The final multiplication by 1/256 (2^{-8}) shifts the data back to its original 16-bit (sign-extended to 24-bit) format.

The execution time for each IIR block (as implemented in this example) is 0.7 μs. For all 10 filters of one channel, the total time is 7 μs. Adding in approximately 4 μs for volume scaling, limiting, receiving and transmitting to the SSI, etc. gives a total execution time of 11 μs. Since the interrupt period is $(1/f_s \times 1/2) = 11.3$ μs, the control-panel polling portion of the software is executed for 0.3 μs for every 11.3 μs. The ratio represents a duty cycle of two percent for slide potentiometer input and 98 percent for audio processing. However, this data cycle poses no problem in that the front panel settings are changed slowly with respect to the 44.1-kHz sample rate of the digital audio.

As discussed previously, the minimum number of instructions (thus, the minimum execution time) is four since the filters have only four coefficients (i.e., terms). Although the example described in this application note is not the minimum design, it was chosen for its simplicity. There are numerous ways to optimize the algorithm (at the expense of simplicity, memory, etc.) to approach the theoretical limit of 0.4 μs per IIR filter. The difference equation (Equation (8)), which is a direct form 1 representation, can be expressed in other forms such as direct form 2 (or the canonical direct).[12] Other forms have advantages, such as better stability due to roundoff or overflow, or may use less memory due

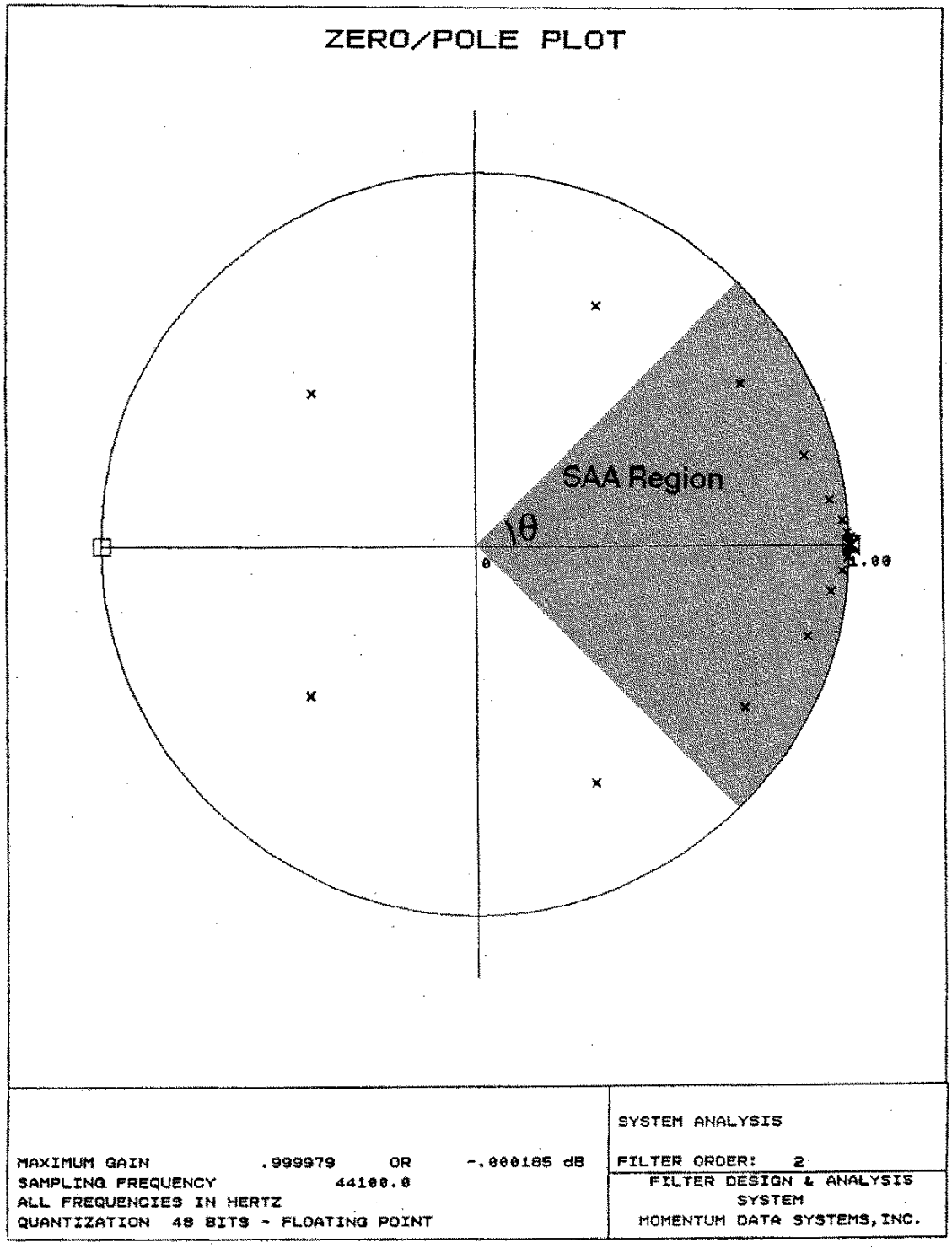


Figure 12. 10-Band Pole/Zero Plot

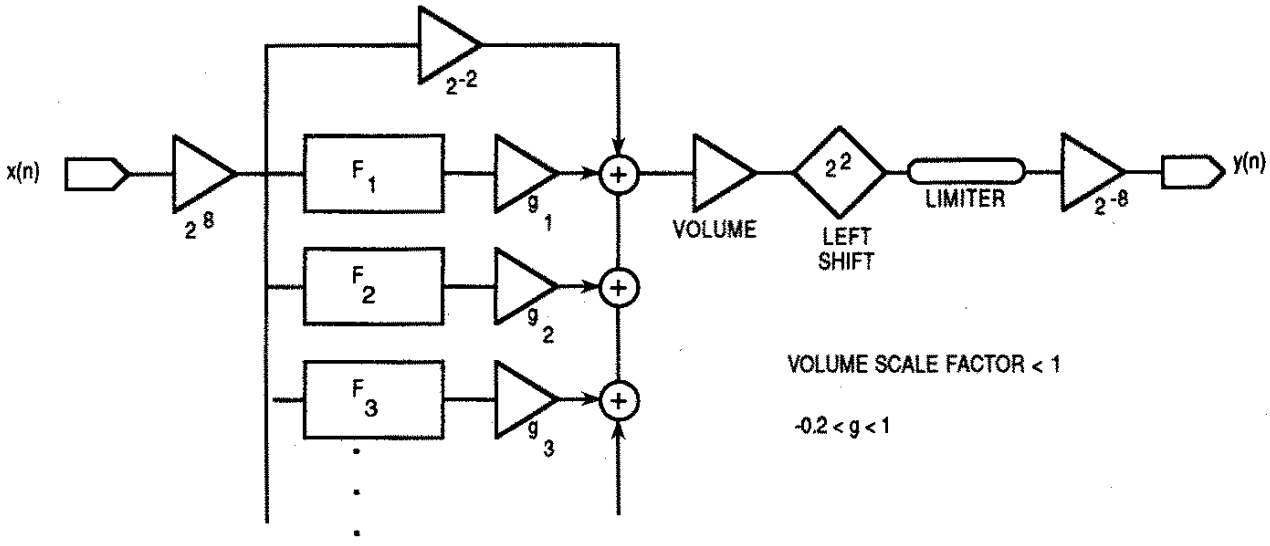


Figure 13. Data-Flow Diagram Showing Implementation of IIR Graphic Equalizer with the DSP56001

to storing filter states rather than storing input and output states. Regardless of the method used to implement the transfer function, the optimized execution time is simply the number of multiply/adds needed to calculate the difference equation. In practice, it may not be possible to approach this minimum because of addressing and data manipulation constraints. Because of the parallel moves allowed with the DSP56000/1, it is much easier to solve these indexing and data-move tasks. Generally, a good IIR design is one in which the total number of instructions is one plus the number of terms in the difference equation. This design allows one instruction cycle to set up data before beginning the MAC operations.

Flowcharts of the basic software algorithm that comprises the graphic equalizer are shown in Figure 14. The software is composed of two independent routines: the slide potentiometer scan routine and the sample/processing interrupt routine. The scan routine (Figure 14(a)) scans 21 slide potentiometers, reduces the 8-bit value to 5 bits (to reduce the size of the lookup table), and uses that value as an index into a gain table. The value from the gain table is stored as the g_i's for use by the sample/process routine. The slide potentiometer scan routine executes for approximately two percent of the total time.

Figure 14(b) shows the sample/process routine, which is executed via the SSI receive interrupt every 11.3 μs. This routine consists of two nearly identical sections, one for left-channel and one for right-channel servicing. The channel requested is determined by the SC0 flag on the SSI port, which is tied directly to the CDP's LRCLK. Data is received from the SSI port, processed by the filter algorithm (previously shown in Figure 13), scaled by the volume control, and transmitted back to the CDP, all in less than the sample time window of 11.3 μs. This routine is executed for 98 percent of the time.

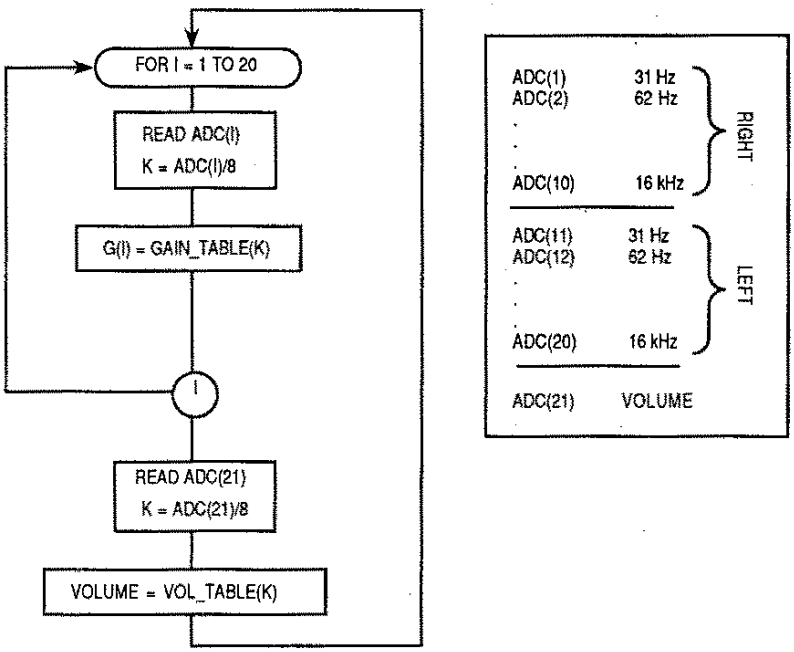


Figure 14(a). Algorithm Flowchart for Slide Potentiometer Scan Routine (Main Program)

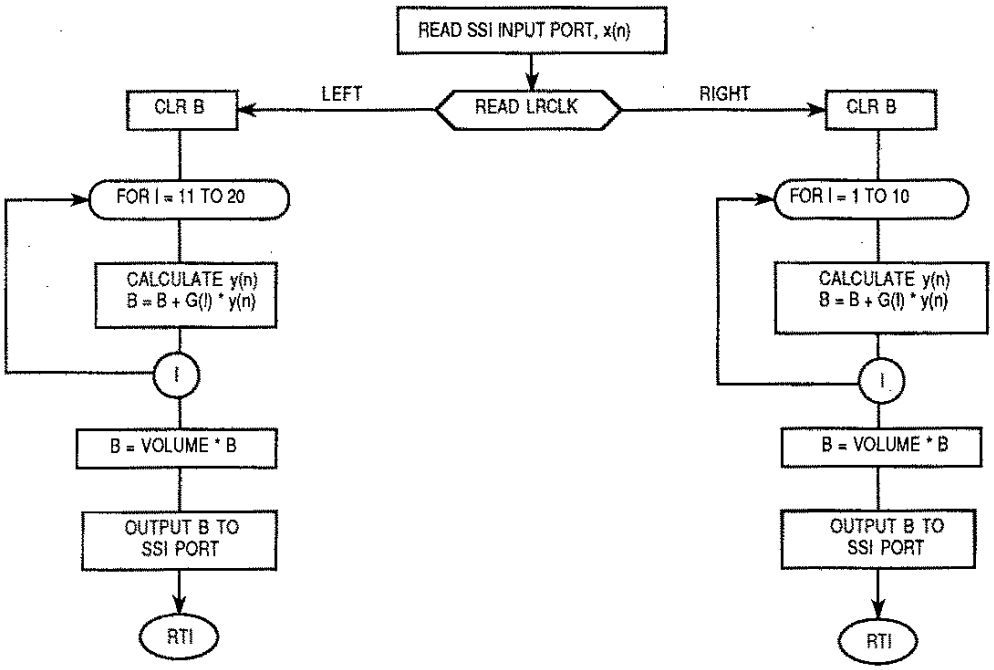


Figure 14(b). Algorithm Flowchart for CDP Interface and IIR Filter Processing (Interrupt Routine)

FOOTNOTES

[1]"Filter Design & Analysis System," chp. 2, p. 15.

[2]Brophy, *Basic Electronics*, pp. 88-92.

[3]Lancaster, *Active Filter Cookbook*, pp. 10-18.

[4]Oppenheim and Schafer, *Digital Signal Processing*, pp. 206-211.

[5]Rabiner and Gold, *Theory and Application*, pp. 219-224.

[6]Oppenheim and Schafer, *Digital Signal Processing*, p. 210.

[7]Strawn, *Digital Audio Signal Processing*, pp. 117-126.

[8]Oppenheim and Schafer, *Digital Signal Processing*, pp. 8-15.

[9]Rabiner and Gold, *Theory and Application*, pp. 9-16.

[10]Strawn, *Digital Audio Signal Processing*, pp. 33-35.

[11]*Ibid.*, p. 116.

[12]Rabiner and Gold, *Theory and Application*, pp. 41-43.

REFERENCES

1. Brophy, J. J. *Basic Electronics for Scientists*. New York: McGraw-Hill, 1966.

2. Chrysafis, A. "Fractional and Integer Arithmetic." Motorola Application Note, forthcoming.

3. "Filter Design & Analysis System." Version 1.3, Momentum Data Systems, 1985.

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

5. Lindsley, B. "Digital Filters on DSP56000/1." Motorola Application Note, forthcoming.

6. Oppenheim, A. V., and Schafer, R.W. *Digital Signal Processing*. New Jersey: Prentice-Hall, 1975.

7. Rabiner, L. R., and Gold, B. *Theory and Application of Digital Signal Processing*. New Jersey: Prentice-Hall, 1975.

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

```
1                               page    132
2
3                       ;*********************************************************
4                       ;*********************************************************
5                       ;*                                                       *
6                       ;*              DUAL 10-BAND IIR BAND-                   *
7                       ;*           PASS FILTER GRAPHIC EQUALIZER               *
8                       ;*                                                       *
9                       ;*********************************************************
10                      ;*********************************************************
11
12
13                      ;*********************************
14                      ;*  SSI and other I/O EQUATES    *
15                      ;*********************************
16
17      00000040        START   EQU     $0040
18      0000FFFF        M_IPR   EQU     $FFFF
19      0000FFFE        M_BCR   EQU     $FFFE
20      0000FFEC        M_CRA   EQU     $FFEC
21      0000FFED        M_CRB   EQU     $FFED
22      0000FFE1        M_PCC   EQU     $FFE1
23      0000FFEE        M_SR    EQU     $FFEE
24      0000FFEF        M_TX    EQU     $FFEF
25      0000FFEF        M_RX    EQU     $FFEF
26
27
28                      ;*********************************
29                      ;*  RESET VECTOR                 *
30                      ;*********************************
31
32      P:0000                  ORG     P:$0000
33      P:0000 0C0040           JMP     START
34
35
36                      ;*********************************
37                      ;*  SSI RCV INTERRUPT VECTOR     *
38                      ;*********************************
39
40      P:000C                  ORG     P:$000C
41      P:000C 0BF080           JSR     LRTEST
                0000F1
42
43
44                      ;*********************************
45                      ;*  MAIN PROGRAM                 *
46                      ;*********************************
47
48      P:0040                  ORG     P:START
49
```

```
50                      ;------------------
51                      ;    Mask Interrupts
52                      ;------------------
53      P:0040 0003F8           ORI     #$03,MR
54
55                      ;----------------------
56                      ;    Initialize SSI Port
57                      ;----------------------
58      P:0041 08F4BE           MOVEP             #$3300,X:M_BCR        ;3 wait states for ADC and MUX.
               003300
59      P:0043 08F4BF           MOVEP             #$3000,X:M_IPR        ;SSI RCV INT priority level.
               003000
60      P:0045 08F4AC           MOVEP             #$6000,X:M_CRA        ;Set SSI word length = 24.
               006000
61      P:0047 08F4AD           MOVEP             #$3200,X:M_CRB        ;Set SSI to synchronous,
               003200
62                                                                      ;and enable RE and TE.
63      P:0049 08F4A1           MOVEP             #$01FF,X:M_PCC        ;Turn on SSI Port.
               0001FF
64
65                      ;-----------------------------------------
66                      ;  Move constants from P:mem to X:mem and Y:mem
67                      ;-----------------------------------------
68
69      P:004B 62F400           MOVE              #$18D,R2              ;** Filter Coefficients.
               00018D
70      P:004D 332000           MOVE              #$20,R3
71                                                                      ;Table located at X:$20.
72      P:004E 061E80           DO      #30,COLOOP                      ;Length of table = 30.
               000051
73      P:0050 07DA84           MOVE              P:(R2)+,X0            ;Order is ... beta, alpha,
74      P:0051 445B00           MOVE              X0,X:(R3)+            ;gamma for each band.
75                      COLOOP
76
77      P:0052 62F400           MOVE              #$1AB,R2              ;** Filter Gain Look-up.
               0001AB
78      P:0054 334000           MOVE              #$40,R3
79                                                                      ;Table located at X:$40.
80      P:0055 062080           DO      #32,FGLOOP                      ;Length of table = 32 (5 bits).
               000058
81      P:0057 07DA84           MOVE              P:(R2)+,X0            ;Minimum value is -.2, maximum
82      P:0058 445B00           MOVE              X0,X:(R3)+            ;value is 0.999, center value is 0.
83                      FGLOOP
84
85      P:0059 62F400           MOVE              #$1CB,R2              ;** Volume Gain Look-up.
               0001CB
86      P:005B 336000           MOVE              #$60,R3
87                                                                      ;Table located at X:$60.
88      P:005C 062080           DO      #32,VGLOOP                      ;Length of table = 32 (5 bits).
               00005F
89      P:005E 07DA84           MOVE              P:(R2)+,X0            ;Minimum value is 0, maximum
90      P:005F 445B00           MOVE              X0,X:(R3)+            ;value is 0.9999.
91                      VGLOOP
92
93      P:0060 62F400           MOVE              #$1EB,R2              ;** Mux Sel Address.
               0001EB
```

```
94        P:0062 330000          MOVE            #0,R3
95                                                                ;Table located at Y:$00.
96        P:0063 061580          DO      #21,MSLOOP               ;Length of table = 21.
                 000066
97        P:0065 07DA84          MOVE            P:(R2)+,X0        ;These are the MUX select addresses
98        P:0066 4C5B00          MOVE                      X0,Y:(R3)+ ;for each of the 21 slide pots.
99                               MSLOOP
100
101
102                              ;-----------------------
103                              ;    Set runtime variables
104                              ;-----------------------
105       P:0067 44F400          MOVE            #$200000,X0       ;Constants used for data scaling.
                 200000
106       P:0069 447000          MOVE            X0,X:>$1D
                 00001D
107       P:006B 44F400          MOVE            #>$80,X0
                 000080
108       P:006D 447000          MOVE            X0,X:>$1E
                 00001E
109
110
111                              ;---------------------------------
112                              ;    Clear x(n) and y(n) table arrays
113                              ;---------------------------------
114       P:006F 200013          CLR     A                         ;Clear Y:$20 to Y:$BF.
115       P:0070 322000          MOVE            #$20,R2
116                                                               ;This area is used for runtime
117       P:0071 06A080          DO      #$A0,XYNCLR               ;tables, x(n) and y(n).
                 000073
118       P:0073 5E5A00          MOVE                      A,Y:(R2)+ ;X:$20..$22 - x(n) left chan.
119                              XYNCLR                              ;X:$40..$42 - y(n) left chan. 31 Hz.
120                                                                 ;X:$44..$46 - y(n) left chan. 62 Hz.
121                                                                 ; ....
122                                                                 ;X:$60..$62 - y(n) left chan. 16 kHz.
123
124                                                                 ;X:$30..$32 - x(n) right chan.
125                                                                 ;X:$80..$82 - y(n) right chan. 31 Hz.
126                                                                 ;X:$84..$86 - y(n) right chan. 62 Hz.
127                                                                 ; ....
128                                                                 ;X:$A0..$A2 - y(n) right chan. 16 kHz.
129
130
131                              ;-----------------------------
132                              ;    Clear g(n) and SP(n) arrays
133                              ;-----------------------------
134       P:0074 200013          CLR     A                         ;Clear X:$00 to X:$14.
135       P:0075 320000          MOVE            #0,R2             ;This is gain array which contains
136                                                                 ;fractional gain value from look-up
137       P:0076 061580          DO      #21,GNCLR                 ;table used to scale filtered band.
                 000078
138       P:0078 565A00          MOVE            A,X:(R2)+
139                              GNCLR
140       P:0079 328000          MOVE            #$80,R2           ;Clear X:$80 to X:$94.
141                                                                 ;This is 8-bit value read from
142       P:007A 061580          DO      #21,SWNCLR                ;ADC slide pots.  This value when
```

```
                00007C
143     P:007C 565A00           MOVE            A,X:(R2)+               ;reduced to 5 bits is used as an
144                     SWNCLR                                          ;index into gain lookup table at
145                                                                     ;X:$40.  The value from lookup table
146                                                                     ;is a 24-bit fraction which is stored
147                                                                     ;at X:$00..$X:$14.
148
149
150                     ;-----------------------------------------------
151                     ;    Setup Register Defaults for Interrupt Routines
152                     ;-----------------------------------------------
153
154     P:007D 344000           MOVE            #$40,R4                 ;** Yi(n):L-ch
155     P:007E 3C0400           MOVE            #4,N4
156     P:007F 368000           MOVE            #$80,R6                 ;** Yi(n):R-ch
157     P:0080 3E0400           MOVE            #4,N6
158     P:0081 352000           MOVE            #$20,R5                 ;** X(n):L-ch
159     P:0082 0502A5           MOVE            #2,M5
160     P:0083 373000           MOVE            #$30,R7                 ;** X(n):R-ch
161     P:0084 0502A7           MOVE            #2,M7
162
163     P:0085 302000           MOVE            #$20,R0                 ;** IIR Coeff
164     P:0086 051DA0           MOVE            #29,M0
165     P:0087 330000           MOVE            #0,R3                   ;** Gain Coeff
166     P:0088 0514A3           MOVE            #20,M3
167     P:0089 3B8000           MOVE            #$80,N3
168
169
170                     ;----------------------
171                     ;    Init SSI Interrupt
172                     ;----------------------
173     P:008A 08F4AD           MOVEP           #$8200,X:M_CRB          ;Enable SSI (RIE) interrupt.
                00B200
174     P:008C 00FCB8           ANDI    #$FC,MR                         ;Unmask all interrupts.
175
176
177                     ;---------------------------------
178                     ;    Main Loop to Monitor Slide Pots
179                     ;---------------------------------
180
181     P:008D 324000   LOOP1   MOVE            #$40,R2                 ;R2 points to gain lookup table.
182
183     P:008E 061580           DO      #21,BPCHAN                      ;Scan all 21 pots.
                0000AF
184     P:0090 4FE300           MOVE                            Y:(R3),Y1 ;MUX select address of pot.
185
186     P:0091 4F7000           MOVE                            Y1,Y:$8000 ;Select MUX channel.
                008000
187     P:0093 06C880           DO      #200,ADC_RD1                    ;Wait for analog MUX to stabilize.
                000095
188     P:0095 000000           NOP
189                     ADC_RD1
190
191     P:0096 5FF000           MOVE                            Y:$1000,B ;WR strobe to ADC (starts data
                001000
192     P:0098 000000           NOP                                     ;conversion).
```

```
193     P:0099 000000            NOP                                    ;Note: A15 tied to WR of ADC.
194     P:009A 5FF000            MOVE                        Y:$8000,B
               008000
195     P:009C 06F481            DO      #500,ADC_RD2                   ;Wait for conversion ADC conversion.
               00009E
196     P:009E 000000            NOP
197                    ADC_RD2
198
199     P:009F 20001B            CLR     B
200     P:00A0 5FF000            MOVE                        Y:$8000,B  ;Read slide pot data from ADC.
               008000
201
202     P:00A2 45F400            MOVE            #>$FF,X1               ;Mask off upper 16 bits.
               0000FF
203     P:00A4 20006E            AND     X1,B
204     P:00A5 21A500            MOVE            B1,X1                  ;X1 now contains 8-bit pot value.
205
206     P:00A6 47EB00            MOVE            X:(R3+N3),Y1           ;Previous pot value.
207     P:00A7 20007C            SUB     Y1,B
208     P:00A8 20002E            ABS     B                              ;If absolute value of difference
209     P:00A9 47F400            MOVE            #>9,Y1                 ;is less than 9, then skip.
               000009
210
211     P:00AB 20007D            CMP     Y1,B                           ;Note: 9 is rather arbitrary.
212     P:00AC 0AF0AB            JMI     SKIP
               0000AF
213
214     P:00AE 456B00            MOVE            X1,X:(R3+N3)           ;If greater than 9, than update
215     P:00AF 205B00  SKIP      MOVE            (R3)+                  ;X:($80+pot_index).
216                    BPCHAN                                           ;This comparsion eliminates jitter
217     P:00B0 000000            NOP                                    ;about a point.
218                                                                     ;End of 21 pot scan.
219
220
221     P:00B1 061480            DO      #20,BPCHAN2                    ;For all pots except volume control.
               0000D0
222
223     P:00B3 45EB00            MOVE            X:(R3+N3),X1           ;Load X1 with slide pot value.
224     P:00B4 20AF00            MOVE            X1,B
225
226     P:00B5 20002A            ASR     B                              ;Reduce to 5-bit value for gain
227     P:00B6 20002A            ASR     B                              ;table lookup.
228     P:00B7 20002A            ASR     B
229
230     P:00B8 21BA00            MOVE            B1,N2
231     P:00B9 000000            NOP
232     P:00BA 45EA00            MOVE            X:(R2+N2),X1           ;Load X1 with fractional value
233                                                                     ;from table lookup.
234     P:00BB 57E300            MOVE             X:(R3),B               ;Compare gain fraction to previous
235     P:00BC 200060            CMP     X1,B                           ;value in X:(R3).
236     P:00BD 0AF0AA            JEQ     NOCHNG                         ;Skip if no change.
               0000CF
237     P:00BF 0AF0A7            JGT     NSLOPE                         ;If new value is greater than
               0000C9
238                                                                     ;previous value, go to negative
239                                                                     ;slope routine.
```

```
240
241     P:00C1 47F400  PSLOPE   MOVE            #0.0001,Y1          ;Positive slope routine.
               000347
242     P:00C3 200078  PRAMP    ADD     Y1,B                        ;Increment previous gain value
243     P:00C4 576300           MOVE            B,X:(R3)            ;by 0.0001 towards latest value
244                                                                 ;read from slide pot.
245     P:00C5 20006D           CMP     X1,B                        ;Continue updating this value
246     P:00C6 0E90C3           JLT     PRAMP                       ;until previous value exceeds
247                                                                 ;new value.
248     P:00C7 0AF080           JMP     NOCHNG                      ;Exit positive slope routine.
               0000CF
249                                                                 ;Note: In the course of this loop,
250                                                                 ;the SSI interrupt will occur many
251                                                                 ;times, so that the band-pass
252                                                                 ;filter response gain will be ramped
253                                                                 ;smoothly to its new value.  Thus,
254                                                                 ;clicking noises generated from a
255                                                                 ;coarse 5-bit gain table will be
256                                                                 ;eliminated.
257
258     P:00C9 47F400  NSLOPE   MOVE            #-0.0001,Y1         ;Negative slope routine.
               FFFCB9
259     P:00CB 200078  NRAMP    ADD     Y1,B                        ;Same as above but negative ramp
260     P:00CC 576300           MOVE            B,X:(R3)            ;to new gain value.
261     P:00CD 20006D           CMP     X1,B
262     P:00CE 0E70CB           JGT     NRAMP
263
264     P:00CF 456300  NOCHNG   MOVE            X1,X:(R3)           ;Update gain table with latest value
265     P:00D0 205B00           MOVE            (R3)+               ;read from slide pot.
266                    BPCHAN2
267                                                                 ;Continue for the all 20 of the
268                                                                 ;band-pass slide pots.
269
270     P:00D1 326000  VOLUME   MOVE            #$60,R2             ;Pot 21 (volume) is treated
271     P:00D2 57F000           MOVE            X:$94,B             ;seperately since it uses a different
               000094
272     P:00D4 20002A           ASR     B                           ;gain lookup table.
273     P:00D5 20002A           ASR     B                           ;Reduce 8-bit value from ADC volume
274     P:00D6 20002A           ASR     B                           ;slide pot to 5-bits.
275
276     P:00D7 21BA00           MOVE            B1,N2               ;Use this value for index into
277     P:00D8 000000           NOP                                 ;lookup table.
278     P:00D9 45EA00           MOVE            X:(R2+N2),X1        ;X1 now contains volume gain
279                                                                 ;fraction.
280     P:00DA 5F9E00           MOVE                        Y:$1E,B ;Y:$1E is previous value of volume
281     P:00DB 20006D           CMP     X1,B                        ;gain.
282     P:00DC 0AF0AA           JEQ     NOCHNG2                     ;If it has not changed, jump.
               0000EE
283     P:00DE 0AF0A7           JGT     NSLOPE2                     ;If it is different, decide whether
               0000E8
284                                                                 ;to ramp negative or positive.
285     P:00E0 47F400  PSLOPE2  MOVE            #0.00005,Y1         ;Positive slope routine for volume.
               0001A3
286     P:00E2 200078  PRAMP2   ADD     Y1,B                        ;Increment previous value by 0.00005
287     P:00E3 5F1E00           MOVE                        B,Y:$1E ;towards new value until it has
288     P:00E4 20006D           CMP     X1,B                        ;passed new value.  Then exit loop.
```

```
289       P:00E5 0E90E2           JLT     PRAMP2                          ;Note: As before, this loop will be
290                                                                       ;interrupted many times by the SSI
291                                                                       ;receive flag full.  The volume gain
292                                                                       ;stored at Y:$1E will ramp smoothly
293                                                                       ;towards its new value.
294       P:00E6 0AF080           JMP     NOCHNG2
                 0000EE
295
296       P:00E8 47F400 NSLOPE2   MOVE            #-0.00005,Y1            ;Negative slope routine for volume.
                 FFFE5D
297       P:00EA 200078 NRAMP2    ADD     Y1,B                            ;Same as before, but ramps in the
298       P:00EB 5F1E00           MOVE                        B,Y:$1E     ;opposite direction.
299       P:00EC 20006D           CMP     X1,B
300       P:00ED 0E70EA           JGT     NRAMP2
301
302
303       P:00EE 4D1E00 NOCHNG2   MOVE                        X1,Y:$1E    ;Update volume gain value with
304       P:00EF 205B00           MOVE            (R3)+                   ;newest value read from slide pot.
305
306       P:00F0 0C008D           JMP     LOOP1                           ;Do everything all over again...
307                                                                       ;continuing slide pot scan loop.
308
309
310                 ;*********************************
311                 ;   MAIN INTERRUPT ROUTINE      *
312                 ;*********************************
313
314       P:00F1 0AAEA0 LRTEST    JSET    #0,X:M_SR,RIGHT                 ;Check SC0 (LRCLK from CDP) to
                 000114
315                                                                       ;determine which channel to process.
316
317                 ;*********************************
318                 ;   LEFT CHANNEL SERVICE        *
319                 ;*********************************
320
321                 ;Save registers
322
323       P:00F3 491F00 LEFT      MOVE            B,L:$1F                 ;Save register B.
324
325                 ;Recieve data
326       P:00F4 084F2F           MOVEP           X:M_RX,B                ;Read SSI data.
327
328       P:00F5 21E600           MOVE            B,Y0                    ;Copy SSI data to Y0.
329       P:00F6 0502A4           MOVE            #2,M4                   ;Set y(n) modulo for 3 words.
330       P:00F7 310A00           MOVE            #10,R1                  ;Set R1 index to filter gain values
331                                                                       ;for the left channel.
332
333       P:00F8 449E13           CLR     A       X:$1E,X0                ;X:$1E = $000080
334       P:00F9 205CD8           MPY     X0,Y0,B (R4)+                   ;Scale input data
335       P:00FA 596500           MOVE                        B0,Y:(R5)   ;by 2^16.
336       P:00FB 0008F8           ORI     #$08,MR                         ;Set scale mode to
337                                                                       ;scale up (left
338                                                                       ;shift) when data is
339                                                                       ;moved from B to X0.
```

```
340                     ;---------------------
341                     ;    All 10 Filters ...
342                     ;---------------------
343     P:00FC 060A80           DO      #10,LFBAND                      ;For all 10 bands of
            000104
344                                                                     ;left channel.
345     P:00FE F0981B           CLR     B         X:(R0)+,X0  Y:(R4)+,Y0 ;X0=beta;Y0=y(n-2).
346     P:00FF F088DE           MAC     -X0,Y0,B  X:(R0)+,X0  Y:(R5)+,Y0 ;X0=alpha;Y0=x(n).
347     P:0100 4ED5DA           MAC     X0,Y0,B               Y:(R5)-,Y0 ;X0=alpha;Y0=x(n-2).
348     P:0101 F098DE           MAC     -X0,Y0,B  X:(R0)+,X0  Y:(R4)+,Y0 ;X0=gamma;Y0=y(n-1).
349     P:0102 46D9DB           MACR    X0,Y0,B   X:(R1)+,Y0             ;Y0=gain for scaling.
350     P:0103 1B5C00           MOVE              B,X0        B,Y:(R4)+  ;X0=filter reponse.
351     P:0104 204CD2           MAC     X0,Y0,A   (R4)+N4                ;A=scaled response.
352                     LFBAND                                          ;Continue for all
353                                                                     ;10 left chan bands.
354
355     P:0105 00F7B8           ANDI    #$F7,MR                         ;Turn off scale mode.
356     P:0106 052BA4           MOVE              #43,M4                ;Set Yi(n) modulo to wrap around to
357                                                                     ;start of entire y(n) buffer.
358                     ;-----------------
359                     ;    Volume Scaling
360                     ;-----------------
361     P:0107 449D00           MOVE              X:$1D,X0              ;X:$1D=$200000.
362     P:0108 4ED000           MOVE                          Y:(R5)+,Y0 ;Y0=x(n).
363     P:0109 4E9ED2           MAC     X0,Y0,A               Y:$1E,Y0  ;scale x(n) down
364     P:010A 21C400           MOVE              A,X0                  ;by 2^-2.  Add this
365                                                                     ;in to the total
366                                                                     ;filter response.
367     P:010B 2000D8           MPY     X0,Y0,B                       ;Y0=volume gain.
368                                                                     ;Scale total left
369                                                                     ;data by volume.
370     P:010C 204C3A           ASL     B         (R4)+N4
371     P:010D 20003A           ASL     B                             ;Scale result by
372                                                                     ;2^2.
373     P:010E 18F400           MOVE              B,X0        #>$8000,Y0 ;Now, move B to X0
            008000
374     P:0110 2000D8           MPY     X0,Y0,B                       ;to force limiting.
375                                                                     ;Scale result down
376                     ;---------------------------                     ;by 2^-8.
377                     ;    Output Data to CD Player
378                     ;---------------------------
379
380     P:0111 08CF2F   LXMIT   MOVEP             B,X:M_TX              ;Write result to SSI.
381
382     P:0112 499F00           MOVE              L:$1F,B               ;Retrieve B register and return.
383     P:0113 000004           RTI
384
385
386                     ;*********************************
387                     ;   RIGHT CHANNEL SERVICE       *
388                     ;*********************************
389
390     P:0114 491F00   RIGHT   MOVE              B,L:$1F               ;Right channel process
391                                                                     ;identical to Left channel,
392     P:0115 084F2F           MOVEP             X:M_RX,B              ;except right channel index
```

```
393     P:0116 21E600          MOVE            B,Y0                ;registers (R6 and R7), and
394     P:0117 0502A6          MOVE            #2,M6               ;first 10 gain table values
395     P:0118 310000          MOVE            #0,R1               ;are used instead.
396
397     P:0119 449E13          CLR     A       X:$1E,X0
398     P:011A 205ED8          MPY     X0,Y0,B (R6)+
399     P:011B 596700          MOVE                        B0,Y:(R7)
400     P:011C 0008F8          ORI     #$08,MR
401
402                    ;---------------------
403                    ;    All 10 Filters ...
404                    ;---------------------
405     P:011D 060A80          DO      #10,RFBAND
            000125
406     P:011F F0D81B          CLR     B       X:(R0)+,X0  Y:(R6)+,Y0
407     P:0120 F0F8DE          MAC     -X0,Y0,B X:(R0)+,X0 Y:(R7)+,Y0
408     P:0121 4ED7DA          MAC     X0,Y0,B             Y:(R7)-,Y0
409     P:0122 F0D8DE          MAC     -X0,Y0,B X:(R0)+,X0 Y:(R6)+,Y0
410     P:0123 46D9DB          MACR    X0,Y0,B X:(R1)+,Y0
411     P:0124 185E00          MOVE            B,X0        B,Y:(R6)+
412     P:0125 204ED2          MAC     X0,Y0,A (R6)+N6
413                    RFBAND
414
415     P:0126 00F7B8          ANDI    #$F7,MR
416     P:0127 0528A6          MOVE            #43,M6
47
418                    ;-----------------
419                    ;    Volume Scaling
420                    ;-----------------
421     P:0128 449D00          MOVE            X:$1D,X0
422     P:0129 4EDF00          MOVE                        Y:(R7)+,Y0
423     P:012A 4E9ED2          MAC     X0,Y0,A             Y:$1E,Y0
424     P:012B 21C400          MOVE            A,X0
425     P:012C 2000D8          MPY     X0,Y0,B
426
427     P:012D 204E3A          ASL     B       (R6)+N6
428     P:012E 20003A          ASL     B
429
430     P:012F 18F400          MOVE            B,X0        #>$8000,Y0
            008000
431     P:0131 2000D8          MPY     X0,Y0,B
432
433                    ;--------------------------
434                    ;    Output Data to CD Player
435                    ;--------------------------
436
437     P:0132 08CF2F  RXMIT   MOVEP           B,X:M_TX
438
439     P:0133 499F00          MOVE            L:$1F,B
440     P:0134 000004          RTI
441
442
```

```
                        ;*********************************
                        ;* DATA VARIABLES and CONSTANTS *
                        ;*********************************

P:018D                          ORG     P:$18D

                        ;-------------------
                        ;    IIR Coefficients
                        ;-------------------

                        ; 31 Hz
                                DC      .4984587                ;beta
                                DC      .0007706594             ;alpha
                                DC      .9984491                ;gamma
                        ; 62 Hz
                                DC      .496876
                                DC      .001562013
                                DC      .9968368
                        ; 125 Hz
                                DC      .4937405
                                DC      .003129769
                                DC      .9935817
                        ; 250 Hz
                                DC      .4876357
                                DC      .006182143
                                DC      .9870087
                        ; 500 Hz
                                DC      .4757282
                                DC      .01213592
                                DC      .9732514
                        ; 1000 Hz
                                DC      .4531951
                                DC      .02340247
                                DC      .9435273
                        ; 2000 Hz
                                DC      .4128511
                                DC      .04357446
                                DC      .8760584
                        ; 4000 Hz
                                DC      .3474929
                                DC      .07625358
                                DC      .7136286
                        ; 8000 Hz
                                DC      .2601072
                                DC      .1199464
                                DC      .3176087
                        ; 16000 Hz
                                DC      .180994
                                DC      .159503
                                DC      -.4435172
```

```
495                 ;------------------------------
496                 ;    Filter Gain (G) Coefficients
497                 ;------------------------------
498
499                         DC      -0.200
500                         DC      -0.187
501                         DC      -0.171
502                         DC      -0.160
503                         DC      -0.150
504                         DC      -0.137
505                         DC      -0.114
506                         DC      -0.103
507
508                         DC      -0.092
509                         DC      -0.080
510                         DC      -0.067
511                         DC      -0.051
512                         DC      -0.039
513                         DC      -0.027
514                         DC      -0.015
515                         DC      0.000
516
517                         DC      0.000
518                         DC      0.030
519                         DC      0.060
520                         DC      0.090
521                         DC      0.120
522                         DC      0.150
523                         DC      0.180
524                         DC      0.210
525
526                         DC      0.250
527                         DC      0.290
528                         DC      0.340
529                         DC      0.380
530                         DC      0.460
531                         DC      0.540
532                         DC      0.750
533                         DC      0.999
534
535                 ;------------------------------
536                 ;    Volume Gain (V) Coefficients
537                 ;------------------------------
538
539                         DC      0.0000
540                         DC      0.0000
541                         DC      0.0002
542                         DC      0.0005
543                         DC      0.0010
544                         DC      0.0030
545                         DC      0.0100
546                         DC      0.0150
547
548                         DC      0.0200
549                         DC      0.0300
550                         DC      0.0400
```

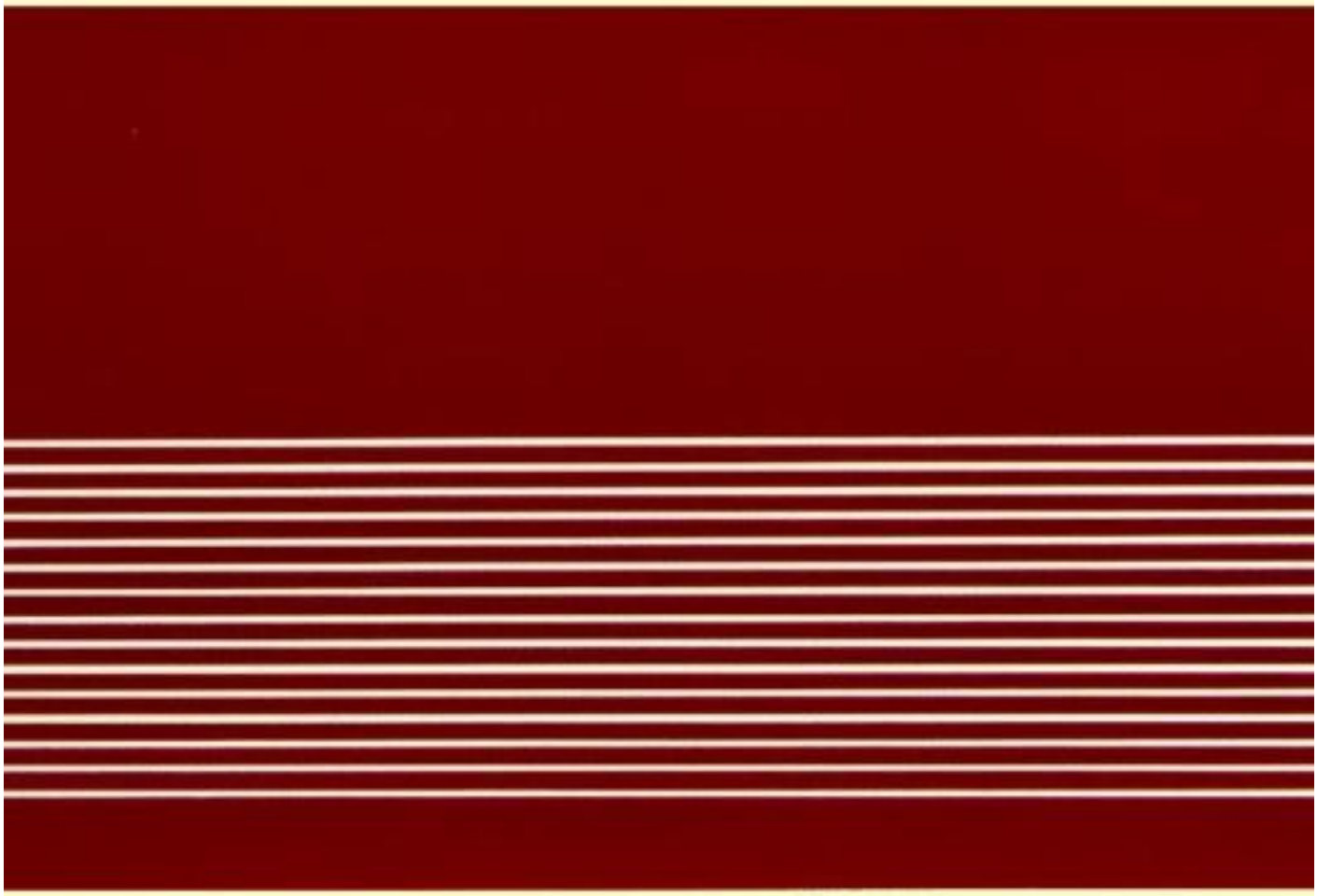
```
                        DC      0.0600
                        DC      0.0800
                        DC     0.1000
                        DC      0.1200
                        DC      0.1500

                        DC      0.2000
                        DC      0.2500
                        DC      0.3000
                        DC      0.3500
                        DC      0.4000
                        DC      0.4500
                        DC      0.5000
                        DC      0.6000

                        DC      0.7000
                        DC      0.8000
                        DC      0.9000
                        DC      0.9999
                        DC      0.9999
                        DC      0.9999
                        DC      0.9999
                        DC      0.9999


                    ;--------------------
                    ;  Slide Pot Addresses
                    ;--------------------
                            DC      $70
                            DC      $71
                            DC      $72
                            DC      $73
                            DC      $74
                            DC      $75
                            DC      $76
                            DC      $77
                            DC      $68
                            DC      $69
                            DC      $6A
                            DC      $6B
                            DC      $6C
                            DC      $6D
                            DC      $6E
                            DC      $6F
                            DC      $58
                            DC      $59
                            DC      $5A
                            DC      $5B
                            DC      $5C

                            END
0    Errors
0    Warnings
```



Literature Distribution Centers:
USA: Motorola Literature Distribution; P.O. Box 20912; Phoenix, Arizona 85036
EUROPE: Motorola Ltd.; European Literature Centre; 88 Tanners Drive, Blakelands, Milton Keynes, MK14 5BP, England.
JAPAN: Nippon Motorola Ltd.; 4-32-1, Nishi-Gotanda, Shinagawa-ku, Tokyo 141 Japan.
ASIA-PACIFIC: Motorola Semiconductors H.K. Ltd.; Silicon Harbour Center, No. 2 Dai King Street, Tai Po Industrial Estate, Tai Po, N.T., Hong Kong.

MOTOROLA

APR2/D