Configure the Coefficients for Digital Biquad Filters in TLV320AIC3xxx Family

Hui Li
HPA-AIP-Audio Converters

1 Introduction of the Digital Biquad Filter

In digital signal processing, a Digital Biquad Filter is a second-order recursive linear filter with two poles and two zeros. “Biquad” is an abbreviation of “biquadratic”, which refers to the fact that in the Z-domain, its transfer function is the ratio of two quadratic functions:

$$H(z) = \frac{b_0 + b_1 \cdot z^{-1} + b_2 \cdot z^{-2}}{a_0 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2}} \quad (1)$$

High-order recursive filters can be highly sensitive to quantization of their coefficients, and can easily become unstable. This is much less of a problem with first and second-order filters. Therefore, high-order filters are typically implemented as serially-cascaded biquad filters.

2 Define Parameters

2.1 User-Defined Parameters

- Fs: Sampling frequency
- f_0: Center frequency or corner frequency or shelf midpoint frequency, depending on which type of filters you use
- dBgain: (only for peaking and shelving filters)
- Q: Quality factor (except in peaking EQ, *A·Q* is the qualify factor)
- BW: Bandwidth in octaves
- S: Shelf slope (for shelving EQ only)

2.2 Intermediate Parameters

$$A = 10^{\frac{dBgain}{40}}$$

$$w_0 = 2 \cdot \pi \cdot \frac{f_0}{Fs}$$

$$\cos(w_0)$$

$$\sin(w_0)$$

$$\alpha = \frac{\sin(w_0)}{2 \cdot Q} = \sin(w_0) \cdot \sinh\left(\frac{\ln 2}{2} \cdot BW \cdot \frac{w_0}{\sin(w_0)}\right) = \frac{\sin(w_0)}{2} \cdot \sqrt{\left(A + \frac{1}{A}\right) \cdot \left(\frac{1}{S} - 1\right) + 2} \quad (2)$$

3 Transfer Functions for Analog Filters (S-domain)

All filter transfer functions were derived using analog prototypes. Transfer functions of a few commonly-used 2nd order analog filters are represented below:

3.1 2nd Low Pass Filter

$$H(s) = \frac{1}{s^2 + \frac{s}{Q} + 1} \tag{3}$$

3.2 2nd High Pass Filter

$$H(s) = \frac{s^2}{s^2 + \frac{s}{Q} + 1} \tag{4}$$

3.3 2nd Peaking EQ

$$H(s) = \frac{s^2 + s \cdot \frac{A}{Q} + 1}{s^2 + \frac{s}{A \cdot Q} + 1} \tag{5}$$

4 Bilinear Transformation From S-domain to Z-domain

The transfer functions of digital biquad filters are represented in Z-domain, whereas transfer functions of analog filters are represented in S-domain. Therefore, bilinear transformation (BLT) is used to convert the transfer function from S-domain to Z-domain. A table of simplified substitutes is provided below:

S-Domain	Z-Domain
1	$(1 + 2z^{-1} + z^{-2}) \cdot (1 - \cos(w_0))$
S	$(1 - z^{-2}) \cdot \sin(w_0)$
S^2	$(1 - 2z^{-1} + z^{-2}) \cdot (1 + \cos(w_0))$
$1 + S^2$	$2 \cdot (1 - 2\cos(w_0) \cdot z^{-1} + z^{-2})$

By plugging these substitutes into the analog filter prototypes, we can mathematically derive the continuous coefficients [b_0, b_1, b_2, a_0, a_1, a_2] for digital biquad filters:

$$H(z) = \frac{b_0 + b_1 \cdot z^{-1} + b_2 \cdot z^{-2}}{a_0 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2}} \tag{6}$$

4.1 2nd Order Low Pass Filter:

$$b_0 = \frac{1 - \cos(w_0)}{2}$$
$$b1 = 1 - \cos(w_0)$$
$$b2 = \frac{1 - \cos(w_0)}{2}$$
$$a0 = 1 + \alpha$$
$$a1 = -2 \cdot \cos(w_0)$$
$$a2 = 1 - \alpha \tag{7}$$

4.2 2nd Order High Pass Filter:

$$
\begin{aligned}
b_0 &= \frac{1+\cos(w_0)}{2} \\
b1 &= 1+\cos(w_0) \\
b2 &= \frac{1+\cos(w_0)}{2} \\
a0 &= 1+\alpha \\
a1 &= -2\cdot\cos(w_0) \\
a2 &= 1-\alpha
\end{aligned}
\qquad (8)
$$

4.3 2nd Order Peaking EQ:

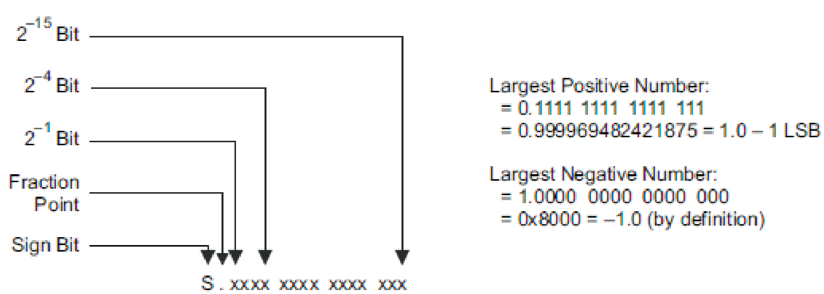
$$
\begin{aligned}
b_0 &= 1+\alpha\cdot A \\
b1 &= -2\cdot\cos(w_0) \\
b2 &= 1-\alpha\cdot A \\
a0 &= 1+\frac{\alpha}{A} \\
a1 &= -2\cdot\cos(w_0) \\
a2 &= 1-\frac{\alpha}{A}
\end{aligned}
\qquad (9)
$$

5 Transfer Function of Digital Biquad Filters in AIC3xxx Family

The AIC3xxx family offers a variety of processing blocks which implement various signal processing capabilities. These processing blocks give users the freedom to choose any type of signal processing block they may use. Depending on the selected processing blocks, different types and orders of digital filtering are available. Due to the fact that high-order digital filters can be highly sensitive to quantization of coefficients, serially-cascaded digital biquad filters can be used to achieve high-order filtering while still maintaining the system's stability. The transfer function of each of the Biquad Filters in AIC3xxx family is given by:

$$
H(z) = \frac{N_0 + 2\cdot N_1\cdot z^{-1} + N_2\cdot z^{-2}}{2^{15} - 2\cdot D_1\cdot z^{-1} - D_2\cdot z^{-2}} \qquad (10)
$$

The coefficients of these biquad filters [N_0, N_1, N_2, D_1, D_2] are each 16 bits wide, in 2s-complement format, and occupy two consecutive 8-bit registers in the register space. Specifically, the filter coefficients are in 1.15 format with a range from -1 (0x8000) to 0.999969482421875 (0x7FFF), as shown below:



6 How to Convert From Continuous Coefficients to 16-bit Quantized Coefficients

Continuous Coefficient [b_0, b_1, b_2, a_0, a_1, a_2] to 16-bit Quantized Coefficients [N_0, N_1, N_2, D_1, D_2]

The transfer function derived from bilinear transformation has continuous coefficients [b_0, b_1, b_2, a_0, a_1, a_2]. However the coefficients of digital biquad filters in the selected processing block [N_0, N_1, N_2, D_1, D_2] need to be in 1.15 format with a range from 0x8000 to 0x7FFF in order to be compatible with the system. Therefore a procedure needs to be introduced to explain how we convert the continuous coefficients to 1.15 format digitized ones. Starting from:

$$H(z) = \frac{b_0 + b_1 \cdot z^{-1} + b_2 \cdot z^{-2}}{a_0 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2}} \tag{11}$$

6.1 *Step 1: Normalize a_0 to 1 so that we have 5 coefficients instead of 6*

$$H(z) = \frac{(b_0/a_0) + (b_1/a_0) \cdot z^{-1} + (b_2/a_0) \cdot z^{-2}}{1 + (a_1/a_0) \cdot z^{-1} + (a_2/a_0) \cdot z^{-2}} \tag{12}$$

6.2 *Step 2: Write out the corresponding continuous coefficients for N_0, N_1, N_2*

$$N_0\_real = (b_0/a_0)$$
$$N_1\_real = \frac{b_1/a_0}{2}$$
$$N_2\_real = (b_2/a_0) \tag{13}$$

6.3 *Scale the coefficients so that none of N_0_real, N_1_real, and N_2_real is greater than 1*

$$Factor = \max(N_0\_real,\ N_1\_real,\ N_2\_real)$$

$$H(z) = Factor \cdot \frac{\frac{b_0/a_0}{Factor} + 2 \cdot \frac{b_1/a_0}{2 \cdot Factor} \cdot z^{-1} + \frac{b_2/a_0}{Factor} \cdot z^{-2}}{1 + (a_1/a_0) \cdot z^{-1} + (a_2/a_0) \cdot z^{-2}} \tag{14}$$

NOTE: Do nothing if all N_0_real, N_1_real and N_2_real are already less than 1.

6.4 *Step 4: Convert scaled continuous coefficients to quantized coefficients*

$$NB = \text{Number of bits};\ Range = 2^{NB-1} - 1$$
$$N_0 = \frac{b_0/a_0}{Factor} \cdot Range$$
$$N_1 = \frac{b_1/a_0}{2 \cdot Factor} \cdot Range$$
$$N_2 = \frac{b_2/a_0}{Factor} \cdot Range$$
$$D_0 = 1 \cdot (Range + 1)(\text{Not required to be calculated, Hard-coded in codec})$$
$$D_1 = \left(\frac{a_1}{-2 \cdot a_0}\right) \cdot Range$$
$$D_2 = \left(-\frac{a_2}{a_0}\right) \cdot Range \tag{15}$$

6.5 Step 5: Round to the nearest integer and then convert it to HEX

For positive integer, convert it directly to HEX.

For negative integer, say -29322. Take the positive (29322) and convert it to binary number, which is 0111001010001010. Take its 2's compliment, which is 1000110101110110. Finally convert it to HEX, which is 0x8D76.

7 How to program the digital biquad filters on ADC side of AIC3xxx family

Coefficients of Digital Biquad Filters can be implemented by running a command script. The script segment below shows how to program the coefficients for a Low-pass 2nd order Butterworth Filter with N_0 = 0x97, N_1 = 0x97, N_2 = 0x97, D_1 = 78e4, D_2 = 8d77

```
-----------------------------------------------------------------------------------------------------
# Configure Digital Biquad Filters on ADC side (One channel)
-----------------------------------------------------------------------------------------------------
# Select Page 0
w 30 00 00
# PRB_R5 selected (Page 0, Register 61)
w 30 3d 05
-----------------------------------------------------------------------------------------------------
# Select Page 4
w 30 00 04
# Write coefficient N0, N1, N2, D1, D2 for Biquad Filter A (Page 4, Register 14 - 23)
w 30 0e 00 97 00 97 00 97 78 e4 8d 77
# Write default coefficients for Biquad Filter B (page 4, Register 24 - 33)
w 30 18 7f ff 00 00 00 00 00 00 00 00
# Write default coefficients for Biquad Filter C (page 4, Register 34 - 43)
w 30 22 7f ff 00 00 00 00 00 00 00 00
# Write default coefficients for Biquad Filter D (page 4, Register 44 - 53)
w 30 2c 7f ff 00 00 00 00 00 00 00 00
# Write default coefficients for Biquad Filter E (page 4, Register 54 – 63)
w 30 36 7f ff 00 00 00 00 00 00 00 00
---------------------------------------------------------------------------------------------------
```

NOTE: If more than 1 biquad filters are used, replace the default coefficients with actual coefficients calculated with the method introduced in this application note.

8 How to program the digital biquad filters on DAC side of AIC3xxx family

```
------------------------------------------------------------------------------------------------
# Configure Digital Biquad Filters on DAC side (Stereo)
------------------------------------------------------------------------------------------------
# Select Page 0
w 30 00 00


# PRB_P2 selected (Page 0, Register 60)
w 30 3c 02
------------------------------------------------------------------------------------------------
# Select Page 8
w 30 00 08
```

Left DAC channel

```
#Write coefficient N0, N1, N2, D1, D2 for Biquad Filter A (Page 8, Register 2 - 11)
w 30 02 00 97 00 97 00 97 78 e4 8d 77

#Write default coefficients for Biquad Filter B (page 8, Register 12 - 21)
w 30 0c 7f ff 00 00 00 00 00 00 00 00

#Write default coefficients for Biquad Filter C (page 8, Register 22 - 31)
w 30 16 7f ff 00 00 00 00 00 00 00 00

#Write default coefficients for Biquad Filter D (page 8, Register 32 - 41)
w 30 20 7f ff 00 00 00 00 00 00 00 00

#Write default coefficients for Biquad Filter E (page 8, Register 42 - 51)
w 30 2a 7f ff 00 00 00 00 00 00 00 00

#Write default coefficients for Biquad Filter F (page 8, Register 52 - 61)
w 30 34 7f ff 00 00 00 00 00 00 00 00
-------------------------------------------------------------------------------------------
```

Right DAC channel

```
#Write coefficient   for Biquad Filter A (Page 8, Register 66 - 75)
w 30 42 00 97 00 97 00 97 78 e4 8d 77

#Write default coefficients for Biquad Filter B (page 8, Register 76 - 85)
w 30 4c 7f ff 00 00 00 00 00 00 00 00

#Write default coefficients for Biquad Filter C (page 8, Register 86 - 95)
w 30 56 7f ff 00 00 00 00 00 00 00 00

#Write default coefficients for Biquad Filter D (page 8, Register 96 - 105)
w 30 60 7f ff 00 00 00 00 00 00 00 00

#Write default coefficients for Biquad Filter E (page 8, Register 106 - 115)
w 30 6a 7f ff 00 00 00 00 00 00 00 00

#Write default coefficients for Biquad Filter F (page 8, Register 116 - 125)
w 30 74 7f ff 00 00 00 00 00 00 00 00
```

NOTE: If more than 1 biquad filters are used, replace the default coefficients with actual coefficients for both left and right channels.

9 How to enable the adaptive filter feature in AIC3xxx family

In non-adaptive filtering system, the user-programmable filter coefficients are locked and cannot be accessed for either read or write once the device is powered up.

However, the AIC3xxx family offers an adaptive filter mode. Setting page 8 / register 1, bit D2 = 1 turns on double buffering of the coefficients. In this mode, filter coefficients can be updated through the host and activated without stopping and restarting the DAC. This enables advanced adaptive filtering application. When the DAC is running and the adaptive filtering mode is turned on, setting page 8 / register 1, bit D0 = 1 switches the coefficient buffers at the next start of a sampling period. This bit is set back to 0 after the switch occurs. At the same time, page 8/ register 1, bit D1 toggles.

The script segment below shows how to implement adaptive filtering for a Low-pass 2nd order Butterworth filter with N_0 = 0x97, N_1 = 0x97, N_2 = 0x97, D_1 = 78e4, D_2 = 8d77

```
---------------------------------------------------------------------------------------------------
--
# Configure Processing Blocks
---------------------------------------------------------------------------------------------------
--
# Select page 0
w 30 00 00

# PRB_P2 selected (Page 0, Register 60)
w 30 3c 02

# Select page 8, Enable Adaptive Filtering for DAC
w 30 00 08 04
---------------------------------------------------------------------------------------------------
--
# First, write to buffer A's registers
# Biquad Filter A, Left Channel (Page 8, Register 2 - 11)
w 30 02 00 97 00 97 00 97 78 e4 8d 77

# Biquad Filter A, Right Channel (Page 8, Register 66 - 75)
w 30 42 00 97 00 97 00 97 78 e4 8d 77

# Second, switch buffers
w 30 00 08 05

#Third, poll page 8 / Register 1 and wait for bit D0 to clear
f 30 01 x1x0

#Finally, write again to Buffer A's registers
# Biquad Filter A, Left Channel (Page 8, Register 2 - 11)
w 30 02 00 97 00 97 00 97 78 e4 8d 77

# Biquad Filter A, Right Channel (Page 8, Register 66 - 75)
w 30 42 00 97 00 97 00 97 78 e4 8d 77
```

10 References

1. TLV320AIC3100, Low-Power Audio Codec with Audio Processing and Mono Class-D Amplifier data sheet (SLAS667)
2. Coefficient RAM Access Mechanism application report (SLAA425)
3. Design and Configuration Guide for the TLV320AIC32x4 Audio Codec application note (SLAA404B)
4. http://en.wikipedia.org/wiki/Bilinear_transform
5. http://en.wikipedia.org/wiki/Two's_complement
6. http://en.wikipedia.org/wiki/Digital_biquad_filter
7. http://www.musicdsp.org/files/Audio-EQ-Cookbook.txt

Appendix A Appendix A: Low-pass 2nd order Butterworth Filter with Fs = 48 KHz, Fc = 600 Hz, Gain = 0 dB, Q = 0.707

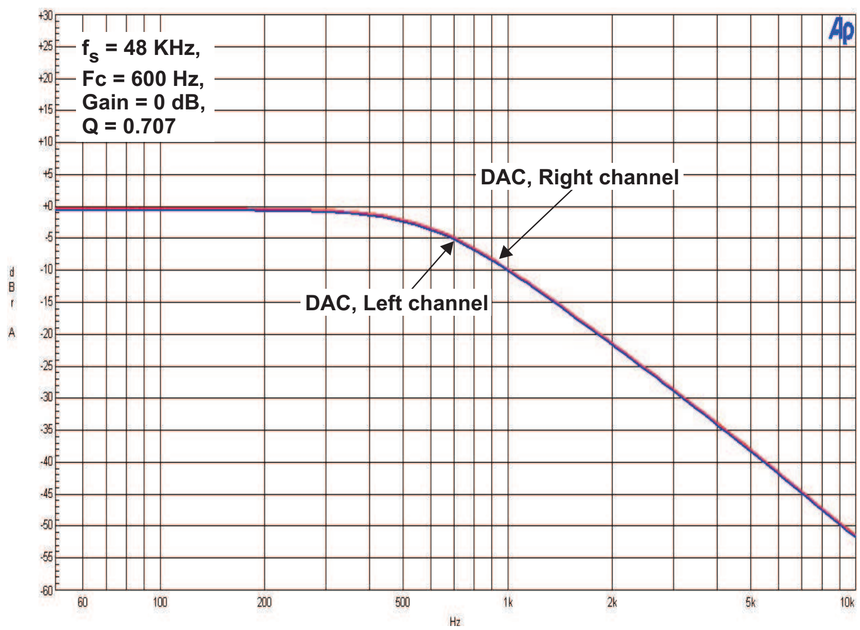
A.1 MATLAB script to calculate [N_0, N_1, N_2, D_1, D_2]

```
%user-defined parameters:
fs = 48000;  %sampling frequency
f0 = 600;
dBgain = 0;
Q = 0.707; % Q is constant for 2nd order Butterworth

%intermediate parameters
wo = 2*pi*f0/fs;
cosW = cos(wo);
sinW = sin(wo);
alpha = sinW/(2*Q);

%low pass filter coefficients
b0 = 10^(dBgain/20)*((1 - cosW)/2);
b1 = 10^(dBgain/20)*(1 - cosW);
b2 = 10^(dBgain/20)*((1 - cosW)/2);
a0 = (1+alpha);
a1 = -2*cosW;
a2 = (1-alpha);

%Normalize so that A0 = 1
B0 = b0/a0;
B1 = b1/(2*a0);
B2 = b2/a0;
A1 = a1/(-2*a0);
A2 = a2/(-a0);

Mx = max(abs([B0, B1, B2]));
if Mx > 1
    B0new = B0/Mx;
    B1new = B1/Mx;
    B2new = B2/Mx;
else
    B0new = B0;
    B1new = B1;
    B2new = B2;
end

NB = 16; % number of bits
Range = 2^(NB-1)-1;

N0 = floor(B0new*Range);
N1 = floor(B1new*Range);
N2 = floor(B2new*Range);

D1 = floor(A1*Range);
D2 = floor(A2*Range);
```

A.2 Command script to program AIC3xxx EVM

```
# Select page 0
w 30 00 00

# PRB_P2 selected
w 30 3c 02

# Select page 8, Enable Adaptive Filtering for DAC
w 30 00 08 04

# First, write to buffer A's registers
# Biquad Filter A, Left Channel (Page 8, Register 2 - 11)
w 30 02 00 2f 00 2f 00 2f 78 e4 8d 77

# Biquad Filter A, Right Channel (Page 8, Register 66 - 75)
w 30 42 00 2f 00 2f 00 2f 78 e4 8d 77

# Second, switch buffers
w 30 00 08 05

#Third, poll page 8 / Register 1 and wait for bit D0 to clear
f 30 01 x1x0

#Finally, write again to Buffer A's registers
# Biquad Filter A, Left Channel (Page 8, Register 2 - 11)
w 30 02 00 2f 00 2f 00 2f 78 e4 8d 77

# Biquad Filter A, Right Channel (Page 8, Register 66 - 75)
w 30 42 00 2f 00 2f 00 2f 78 e4 8d 77
```

A.3 Frequency Response



Appendix B High-pass 2nd order Butterworth Filter with Fs = 48 KHz, Fc = 600 Hz, Gain = 0 dB, Q = 0.707

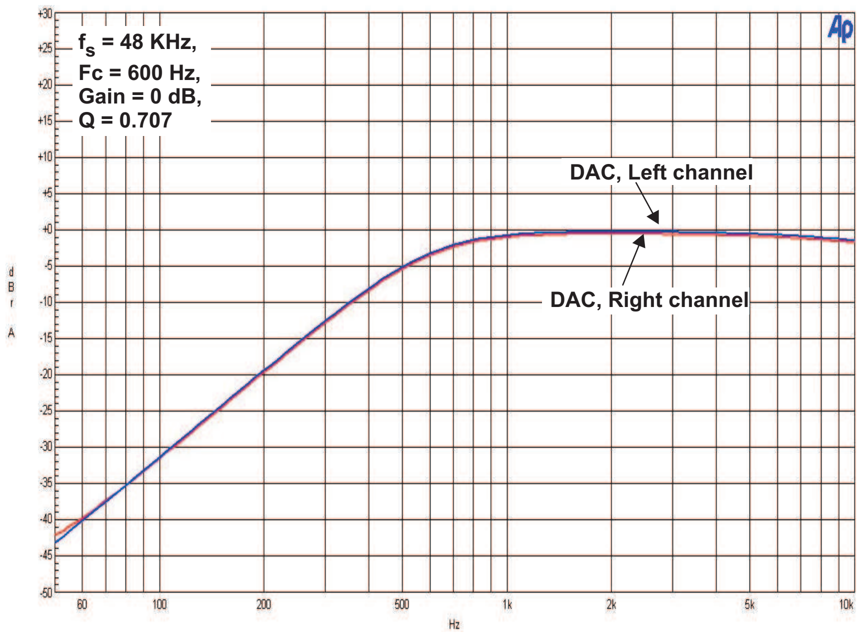
B.1 MATLAB script to calculate [N_0, N_1, N_2, D_1, D_2]

```
%user-defined parameters:
fs = 48000;  %sampling frequency
f0 = 600;
dBgain = 0;
Q = 0.707; %Q is a constant for 2nd order Butterworth

%intermediate parameters
wo = 2*pi*f0/fs;
cosW = cos(wo);
sinW = sin(wo);
alpha = sinW/(2*Q);

%High pass filter coefficients
b0 = 10^(dBgain/20)*((1 + cosW)/2);
b1 = 10^(dBgain/20)*(-(1+cosW));
b2 = 10^(dBgain/20)*((1 + cosW)/2);
a0 = 1 + alpha;
a1 = -2*cosW;
a2 = 1 - alpha;

%Normalize so that A0 = 1
B0 = b0/a0;
B1 = b1/(2*a0);
B2 = b2/a0;
A1 = a1/(-2*a0);
A2 = a2/(-a0);

Mx = max(abs([B0, B1, B2]));
if Mx > 1

B0new = B0/Mx;
B1new = B1/Mx;
B2new = B2/Mx;

else
B0new = B0;
B1new = B1;
B2new = B2;
end

NB = 16; % number of bits
Range = 2^(NB-1)-1;

N0 = floor(B0new*Range);
N1 = floor(B1new*Range);
N2 = floor(B2new*Range);
D1 = floor(A1*Range);
D2 = floor(A2*Range);
```

B.2 Command script to program AIC3xxx EVM

```
# Select page 0
w 30 00 00

# PRB_P2 selected
w 30 3c 02

# Select page 8, Enable Adaptive Filtering for DAC
w 30 00 08 04

# First, write to buffer A's registers
# Biquad Filter A, Left Channel (Page 8, Register 2 - 11)
w 30 02 79 14 86 ec 79 14 78 e4 8d 77

# Biquad Filter A, Right Channel (Page 8, Register 66 - 75)
w 30 42 79 14 86 ec 79 14 78 e4 8d 77

# Second, switch buffers
w 30 00 08 05

#Third, poll page 8 / Register 1 and wait for bit D0 to clear
f 30 01 x1x0

#Finally, write again to Buffer A's registers
# Biquad Filter A, Left Channel (Page 8, Register 2 - 11)
w 30 02 79 14 86 ec 79 14 78 e4 8d 77

# Biquad Filter A, Right Channel (Page 8, Register 66 - 75)
w 30 42 79 14 86 ec 79 14 78 e4 8d 77
```

B.3 Frequency Response



Appendix C Peaking EQ Filter with Fs = 48 KHz, Fc = 600 Hz, Gain = 10 dB and BW = 100 Hz

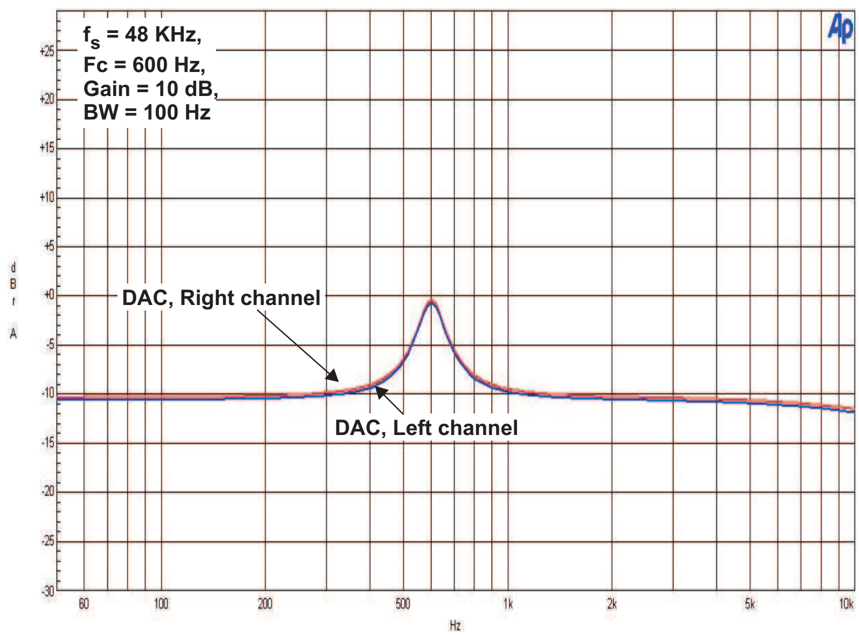
C.1 MATLAB script to calculate [N_0, N_1, N_2, D_1, D_2]

```
%user-defined parameters:
fs = 48000; %sampling frequency
f0 = 600;
dBgain =10;  %10 dB
Q = 1;

%intermediate parameters
wo = 2*pi*f0/fs;
cosW = cos(wo);
sinW = sin(wo);
A = 10^(dBgain/40);
alpha = sinW/(2*Q*A);

% %Peaking EQ coefficients
b0 = 10^(dBgain/20)*(1 + alpha*A);
b1 = 10^(dBgain/20)*(-2*cos(wo));
b2 = 10^(dBgain/20)*(1 - alpha*A);
a0 = 1 + (alpha/A);
a1 = -2*cos(wo);
a2 = 1 - (alpha/A);

%Normalize so that A0 = 1
B0 = b0/a0;
B1 = b1/(2*a0);
B2 = b2/a0;
A1 = a1/(-2*a0);
A2 = a2/(-a0);

Mx = max(abs([B0, B1, B2]));
if Mx > 1

B0new = B0/Mx;
B1new = B1/Mx;
B2new = B2/Mx;

else
B0new = B0;
B1new = B1;
B2new = B2;
end

NB = 16; % number of bits
Range = 2^(NB-1)-1;

N0 = floor(B0new*Range);
N1 = floor(B1new*Range);
N2 = floor(B2new*Range);
D1 = floor(A1*Range);
D2 = floor(A2*Range);
```

C.2 Command script to program AIC3xxx EVM

```
# Select page 0
w 30 00 00

# PRB_P2 selected
w 30 3c 02

# Select page 8, Enable Adaptive Filtering for DAC
w 30 00 08 04

# First, write to buffer A's registers
# Biquad Filter A, Left Channel (Page 8, Register 2 - 11)
w 30 02 7f ff 82 fd 7a ce 7e c5 81 ac

# Biquad Filter A, Right Channel (Page 8, Register 66 - 75)
w 30 42 7f ff 82 fd 7a ce 7e c5 81 ac

# Second, switch buffers
w 30 00 08 05

#Third, poll page 8 / Register 1 and wait for bit D0 to clear
f 30 01 x1x0

#Finally, write again to Buffer A's registers
# Biquad Filter A, Left Channel (Page 8, Register 2 - 11)
w 30 02 7f ff 82 fd 7a ce 7e c5 81 ac

# Biquad Filter A, Right Channel (Page 8, Register 66 - 75)
w 30 42 7f ff 82 fd 7a ce 7e c5 81 ac
```

C.3 Frequency Response



IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of TI information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

TI products are not authorized for use in safety-critical applications (such as life support) where a failure of the TI product would reasonably be expected to cause severe personal injury or death, unless officers of the parties have executed an agreement specifically governing such use. Buyers represent that they have all necessary expertise in the safety and regulatory ramifications of their applications, and acknowledge and agree that they are solely responsible for all legal, regulatory and safety-related requirements concerning their products and any use of TI products in such safety-critical applications, notwithstanding any applications-related information or support that may be provided by TI. Further, Buyers must fully indemnify TI and its representatives against any damages arising out of the use of TI products in such safety-critical applications.

TI products are neither designed nor intended for use in military/aerospace applications or environments unless the TI products are specifically designated by TI as military-grade or "enhanced plastic." Only products designated by TI as military-grade meet military specifications. Buyers acknowledge and agree that any such use of TI products which TI has not designated as military-grade is solely at the Buyer's risk, and that they are solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI products are neither designed nor intended for use in automotive applications or environments unless the specific TI products are designated by TI as compliant with ISO/TS 16949 requirements. Buyers acknowledge and agree that, if they use any non-designated products in automotive applications, TI will not be responsible for any failure to meet such requirements.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

Products		**Applications**	
Amplifiers	amplifier.ti.com	Audio	www.ti.com/audio
Data Converters	dataconverter.ti.com	Automotive	www.ti.com/automotive
DLP® Products	www.dlp.com	Communications and Telecom	www.ti.com/communications
DSP	dsp.ti.com	Computers and Peripherals	www.ti.com/computers
Clocks and Timers	www.ti.com/clocks	Consumer Electronics	www.ti.com/consumer-apps
Interface	interface.ti.com	Energy	www.ti.com/energy
Logic	logic.ti.com	Industrial	www.ti.com/industrial
Power Mgmt	power.ti.com	Medical	www.ti.com/medical
Microcontrollers	microcontroller.ti.com	Security	www.ti.com/security
RFID	www.ti-rfid.com	Space, Avionics & Defense	www.ti.com/space-avionics-defense
RF/IF and ZigBee® Solutions	www.ti.com/lprf	Video and Imaging	www.ti.com/video
		Wireless	www.ti.com/wireless-apps

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265