

## TLV320AIC3xxx 滤波器的设置与实时调节

Kevin Dang

Sales and Marketing/Shenzhen

### ABSTRACT

AIC 系列音频编解码器比如 TLV320AIC3101/3104/3106/3204/3206 等内部都有固化的微程序，这些微程序实现了滤波器，AGC, DRC, Tone Generator 等功能。这些滤波器以 Biquad 双二阶滤波器的架构来实现，可以配置为几乎所有常见的滤波器类型，在实际应用中有着非常重要的作用。本文介绍这些滤波器的配置方法，以及怎样在产品中怎样计算所需要的数值来实时调整这些滤波器。

### Contents

<b>1</b>	<b>Biquad 双二阶滤波器简介 .....</b>	<b>2</b>
<b>2</b>	<b>音频系统中的 BQ 滤波器应用 .....</b>	<b>3</b>
<b>3</b>	<b>利用工具 TIBQ 生成滤波器参数 .....</b>	<b>3</b>
<b>4</b>	<b>参数的配置和实时切换 .....</b>	<b>4</b>
	4.1 参数的配置 .....	4
	4.2 参数的实时切换 .....	5
<b>5</b>	<b>用程序计算 EQ 的参数 .....</b>	<b>6</b>
<b>6</b>	<b>参考文献 .....</b>	<b>7</b>

### Figures

<b>Figure 1</b>	<b>Biquad 滤波器的逻辑框图 .....</b>	<b>2</b>
<b>Figure 2</b>	<b>AIC 系列 CODEC 的 Biquad 滤波器的传输函数 .....</b>	<b>2</b>
<b>Figure 3</b>	<b>AIC 系列 CODEC 的 Biquad 滤波器的具体实现 .....</b>	<b>2</b>
<b>Figure 4</b>	<b>TIBQ 操作界面说明 .....</b>	<b>3</b>
<b>Figure 5</b>	<b>TIBQ 计算结果输出 .....</b>	<b>4</b>
<b>Figure 6</b>	<b>手册中系数地址表格示例 .....</b>	<b>5</b>
<b>Figure 7</b>	<b>双缓冲切换控制寄存器 .....</b>	<b>5</b>

# 1 Biquad 双二阶滤波器简介

线性非时变数字系统的滤波器一般表达为:

$$H(z) = \frac{B(z)}{A(z)} = \frac{b_0 + b_1z^{-1} + b_2z^{-2} + \dots + b_Nz^{-N}}{1 + a_1z^{-1} + a_2z^{-2} + \dots + a_Mz^{-M}}$$

Biquad 滤波器因为分子分母都为两阶的等式所以称为双二阶滤波器,

$$H(z) = \frac{B(z)}{A(z)} = \frac{b_0 + b_1z^{-1} + b_2z^{-2}}{1 + a_1z^{-1} + a_2z^{-2}}$$

作为通用数字滤波器的实现, 双二阶滤波器可以实现常见的所有滤波器的功能比如高通、低通、带通等等, AIC 系列 CODEC 使用了简单方便的 Direct Form I 形式的实现方式, 逻辑表达上如下图所示:

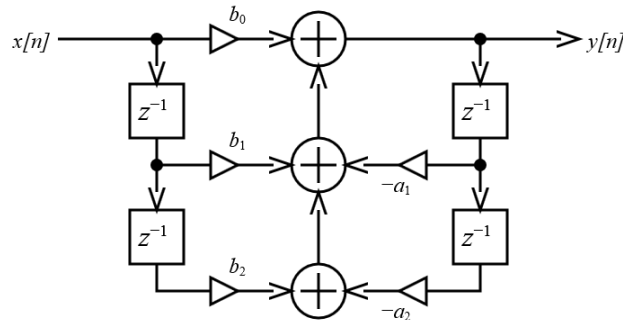


Figure 1 Biquad 滤波器的逻辑框图

为了便于实现整数运算, 系数将转换为整数, 所以 24 位系数对应 1.23 格式, a<sub>0</sub> 变换为 2<sup>23</sup>, 16 位系数对应 1.15 格式, a<sub>0</sub> 变换为 2<sup>15</sup>, 为尽可能提高系数的精度, N1 和 D1 实际使用的时候将乘以 2 将为系数 N1/D1 提供多一位的精度:

$$H(z) = \frac{N_0 + 2 * N_1z^{-1} + N_2z^{-2}}{2^{23} - 2 * D_1z^{-1} - D_2z^{-2}} \qquad H(z) = \frac{N_0 + 2 * N_1z^{-1} + N_2z^{-2}}{2^{15} - 2 * D_1z^{-1} - D_2z^{-2}}$$

Figure 2 AIC 系列 CODEC 的 Biquad 滤波器的传输函数

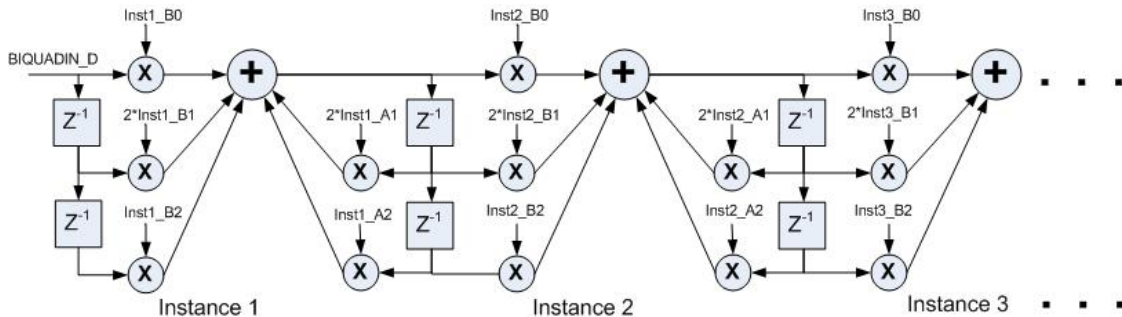


Figure 3 AIC 系列 CODEC 的 Biquad 滤波器的具体实现

所以如果使用通用的滤波器设计工具来计算参数，则需要一系列的转换来把系数 **a** 和 **b** 转换为这里实际使用的 **D** 和 **N**。

## 2 音频系统中的 BQ 滤波器应用

滤波器在音频系统中，主要用于对声学器件的频率补偿，噪声衰减，音色调整，相位纠正等。

一般麦克风和音箱的频响曲线都不平坦，甚至存在较大的高点或者低点，比如最常见的低频灵敏度不足，音箱 **f0** 点灵敏度过高，侧出音喇叭存在 **fH** 频点灵敏度过高等问题，都需要用 **EQ** 或者 **Shelf** 滤波器来进行提升或者衰减，以控制频响的平坦度在需要的范围内。另外一种频率补偿，属于消除无效频率，比如过低的频率，喇叭或者音箱几乎无法表现出来，只能增加线圈电流，不能转换为声音，这种频率一般就采用高通滤波器去除。同样适用于这种情况的有倒相孔的噪声。

噪声衰减上，一种是固定噪声，比如现有硬件电路上无法去除的工频干扰，则可以用 **Notch** 滤波器滤掉对应的频率。另外，很多常驻噪声比如风声等，都在比较低的频率，所以如果是用于语音通讯的时候，则也可以用高通滤波器比如 **fc** 为 **300Hz** 的二阶高通滤波器将这些频率进行衰减。

音色调整，则是比较主观的使用 **EQ** 对不同的频率进行调节。相位纠正一般需要专业的设备来测量，用于室内音箱分布的时候考虑音箱的位置和声音的反射进行相位的补偿。

## 3 利用工具 TIBQ 生成滤波器参数

为了简化设计，TI 提供了两个工具可以用于 AIC 系列 CODEC 的滤波器设计。TIBQ(COEFFICIENT-CALC)和用于 miniDSP CODEC 的 PurePath Studio. 这两个工具都可以从 TI 官网下载。

TIBQ 设计专用于 Biquad 滤波器的参数计算所以用起来简单方便。但是没有说明文档可以用，这里对其用法简单说明一下。下面是 TIBQ 运行的界面：

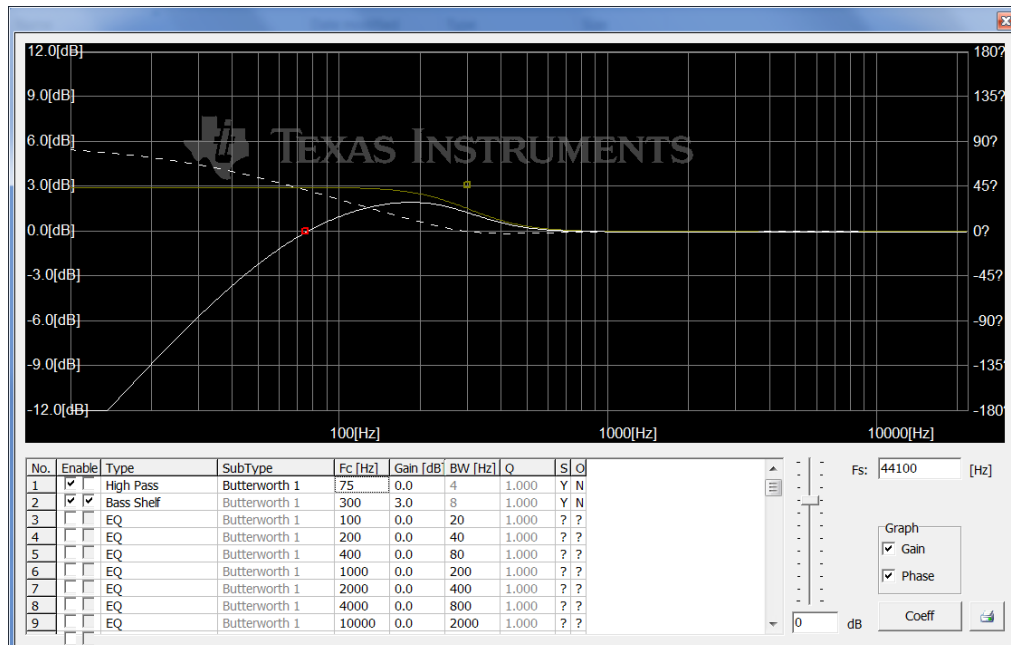
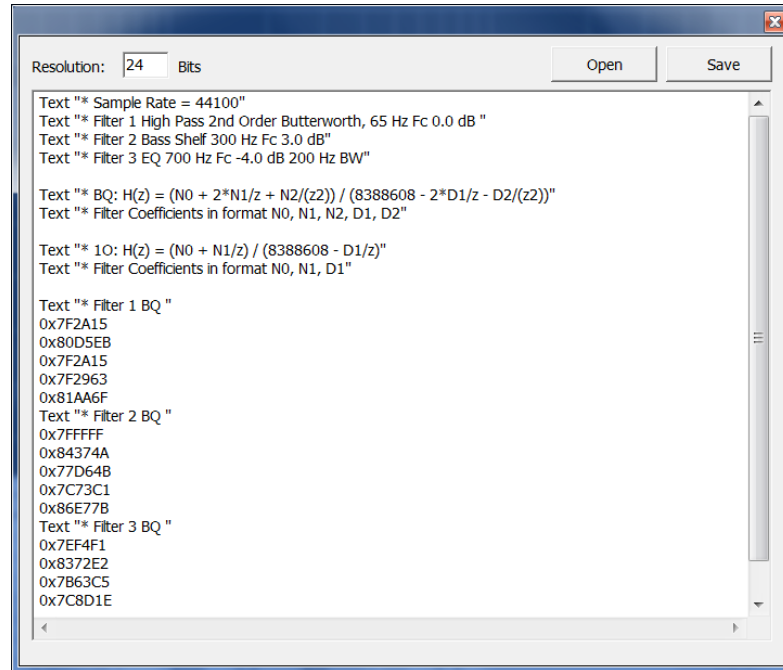


Figure 4 TIBQ 操作界面说明

首先在  $F_s$  的位置设定采样率，然后在左下列表打开关闭滤波器，也可以左键双击波特图空白位置增加新的滤波器。**Enable** 对应的两个选项，第一个表示是否打开对应滤波器，第二个表示是否单独显示对应滤波器的频响曲线，比如这个图中的 **Bass Shelf** 滤波器可以看到对应草绿色的频响曲线。最后面的 **S** 表示是否稳定，**O** 表示是否溢出。**AIC** 系列 **CODEC** 滤波器为 16 位或者 24 位，可调增益为  $\pm 12\text{dB}$ 。如果前面有额外预增益调节则可以改变这个实际范围，在右侧的增益调整可以设置这个预增益来进行叠加计算来判断是否溢出。已经设定的滤波器的频点和增益可以通过鼠标直接拖动对应的点来自由调整。

设置完滤波器之后，可以点“**Coeff**”按键来查看具体的系数情况，如下图所示：



```

Resolution: 24 Bits
Open Save

Text "* Sample Rate = 44100"
Text "* Filter 1 High Pass 2nd Order Butterworth, 65 Hz Fc 0.0 dB "
Text "* Filter 2 Bass Shelf 300 Hz Fc 3.0 dB"
Text "* Filter 3 EQ 700 Hz Fc -4.0 dB 200 Hz BW"

Text "* BQ: H(z) = (N0 + 2*N1/z + N2/(z^2)) / (8388608 - 2*D1/z - D2/(z^2))"
Text "* Filter Coefficients in format N0, N1, N2, D1, D2"

Text "* IO: H(z) = (N0 + N1/z) / (8388608 - D1/z)"
Text "* Filter Coefficients in format N0, N1, D1"

Text "* Filter 1 BQ "
0x7F2A15
0x80D5EB
0x7F2A15
0x7F2963
0x81AA6F
Text "* Filter 2 BQ "
0x7FFFFF
0x84374A
0x77D64B
0x7C73C1
0x86E77B
Text "* Filter 3 BQ "
0x7EF4F1
0x8372E2
0x7B63C5
0x7C8D1E

```

Figure 5 TIBQ 计算结果输出

一般 **AIC31** 系列为 16 位系数，而 **AIC32** 系列为 24 位系数，具体可以查看对应的数据手册。在 **Resolution** 位置设定对应的系数位数并回车，下面的窗口就显示出对应的滤波器情况，传输函数和对应的参数的数值。这些数值即可用于写入 **CODEC** 内部对应的系数寄存器。数值的排列和窗口上面所列的传输函数是对应的，比如对于两阶滤波器，有 **N0**, **N1**, **N2** 和 **D1**, **D2** 分别对应列出的 5 组数值。而对于一阶的滤波器，**N2** 和 **D2** 为 0，所以只有三个系数，对应 **N0**, **N1** 和 **D1**。这些数值的定义和 **CODEC** 的数据手册中的定义是相对应的。

## 4 参数的配置和实时切换

### 4.1 参数的配置

参数配置需要注意的主要是要在 **CODEC** 没有运行的时候进行设置。**CODEC** 内部使用小型 **DSP** 执行微代码，在运行的时候，**DSP** 使用的系数是没有办法修改的。所以在设置系数的时候，要保证 **DSP** 是没有运行的。**DSP** 的电源和 **ADC/DAC** 的电源是相连的，所以要断掉 **DSP** 的电源，则需要关闭对应的 **ADC/DAC**。或者在初始化的时候，在开启 **ADC/DAC** 之前配置好系数。

如果已经在运行，想要关断 DSP/ADC/DAC，则需要遵守正确的上电和掉电次序才能保证 DSP 正常工作，避免产生噪声。掉电次序为静音→关闭 ADC/DAC→读写系数→关闭 PLL，重新上电一般打开 ADC/DAC→打开 PLL→取消静音。

系数的地址在数据手册可以查到，AIC310x 系列因为滤波器比较少，比如录音和播放是分时复用的滤波器，每个通道可能只有 2 个串联的滤波器可以用，所以系数的地址直接在寄存器列表中列出来。而 AIC311x 和 AIC32 系列，滤波器很多，而且还有其它功能比如 3D, AGC, DRC, Beep Generator 等，所以系数非常多，滤波器的系数则在介绍 Processing Block (PRB) 之后有一个表格，里面列出了各滤波器的系数地址。如下图，比如左声道滤波器 A 的系数 N0 是从 P44 寄存器 12 开始的 3 个字节：

Filter	Coefficient	Left DAC Channel	Right DAC Channel
BIQUAD A	N0	C1 (Page 44, Registers 12,13,14)	C33 (Page 45, Registers 20,21,22)
	N1	C2 (Page 44, Registers 16,17,18)	C34 (Page 45, Registers 24,25,26)
	N2	C3 (Page 44, Registers 20,21,22)	C35 (Page 45, Registers 28,29,30)
	D1	C4 (Page 44, Registers 24,25,26)	C36 (Page 45, Registers 32,33,34)
	D2	C5 (Page 44, Registers 28,29,30)	C37 (Page 45, Registers 36,37,38)

Figure 6 手册中系数地址表格示例

写系数 N0 只需要把前面用 TIBQ 生成的 N0 的数值分为 3 个字节写到这 3 个寄存器即可。

## 4.2 参数的实时切换

但常常在改变参数时是不希望关闭 ADC/DAC 再开启的，比如要在播放音乐的时候调整 EQ，那么这个时候就适合使用实时切换系数避免掉电引起的声音断续。AIC311x 和 AIC32 系列 CODEC 提供了双缓冲的机制来实现系数的实时切换，系数可以使用两组存储空间缓冲 A 和缓冲 B，在 DSP 运行的时候，DSP 使用其中一个缓冲比如 Buffer A，这个正在使用的缓冲 A 不能被 I2C 读写，而另外一个没有被 DSP 使用的缓冲 B 可以被 I2C 读写。当启动切换缓冲的命令的时候，DSP 自动切换使用另外一个缓冲 B，之前使用的缓冲 A 被释放变得可以被 I2C 读写。AIC 提供了一个友好的机制来简化操作，就是当 DSP 运行的时候，无论 I2C 读写的哪个缓冲区，AIC 总是将其自动映射到 DSP 没有使用的空闲缓冲。

使用双缓冲机制的滤波器这里称为 Adaptive Filter。缓冲区控制的寄存器用于开关这种模式和执行缓冲区切换。比如 AIC3204 的控制寄存器，D2 位为开关，D0 为切换命令，D1 表示当前 DSP 正在使用的缓冲：

### 5.4.2 Page 8 / Register 1: ADC Adaptive Filter Configuration Register - 0x08 / 0x01 (P8\_R1)

BIT	READ/ WRITE	RESET VALUE	DESCRIPTION
D7-D3	R	0000 0	Reserved. Write only default values
D2	R/W	0	ADC Adaptive Filtering Control 0: Adaptive Filtering disabled for ADC 1: Adaptive Filtering enabled for ADC
D1	R	0	ADC Adaptive Filter Buffer Control Flag 0: In adaptive filter mode, ADC accesses ADC Coefficient Buffer-A and control interface accesses ADC Coefficient Buffer-B 1: In adaptive filter mode, ADC accesses ADC Coefficient Buffer-B and control interface accesses ADC Coefficient Buffer-A
D0	R/W	0	ADC Adaptive Filter Buffer Switch control 0: ADC Coefficient Buffers will not be switched at next frame boundary 1: ADC Coefficient Buffers will be switched at next frame boundary, if in adaptive filtering mode. This will self clear on switching.

Figure 7 双缓冲切换控制寄存器



D0 位这个切换命令，同时也是自清零的，指示命令是否执行完成。需要注意的是，这个缓冲的切换是由 I2S 接口的 WCLK 时钟来驱动的，所以在切换的时候必须保证 I2S 是工作的。如果不能确定 I2S 是否在工作，则可以判断 D0 位是否在一定时间内比如 30 毫秒内自动清零，如果设置为 1 之后 30 毫秒内没有清零，则有可能 I2S 是没有工作的，适合使用断电的方式设置参数。

所以这里总结起来，在运行的时候进行系数的切换需要做以下动作：

写系数（任意缓冲区比如缓冲区 A）→向 Adaptive Filter Configuration Register 写 0x05（保持这个模式是开的）→每 10 毫秒读一次控制寄存器并判断 D0 位是否清零，如果没有清零，继续完成 3 次循环→如果 D0 位仍然没有清零，则执行断电写参数并重新上电；→如果 D0 位已经清零，则再次执行写系数操作（任意缓冲区比如缓冲区 A）。此操作适用于所有模块的系数修改比如 AGC, DRC, Beep 等。

## 5 用程序计算 EQ 的参数

最后举例说明在程序中怎样实时计算 EQ 的参数。通常 EQ 需要根据给定的频率带宽和增益来实时调整的，其他类型的滤波器一般没有这种需求。在文档 SLAA447 中，作者给出了公式的推导方法和 Matlab 的工具生成的参数的转换方法，没有介绍怎样用代码将这些东西实现并联系起来，这里不在重复 SLAA447 中讲的变换过程，直接给出经过测试验证正确的代码和简单注释供参考，从代码和注释来看，转换的过程更加清晰和直观。

```
#include <stdio.h>
#include <math.h>
int main(int argc, char **argv){
    double res = 8388607; /* 这是一个常量: (2^(B-1)-1); 24位系数: 8388607, 16位: 32767 */
    double dBVal = 3; /* 增益 */
    double SRVal = 48000; /* 采样率 */
    double F0Val = 300; /* 频点 */
    double BWVal = 100; /* 带宽 */
    double A, omega, sn, cs, alpha; /* 中间量, s域到z域双线性变换使用 */
    double a0, a1, a2, b0, b1, b2; /* 系数, 计算用中间量 */
    double absMax = 0; /* 将b0, b1, b2转为小数时使用 */
    int N0, N1, N2, D1, D2; /* 保存计算的系数结果 */

    A = pow(10, (dBVal)/40);
    omega = 2 * M_PI * (F0Val)/(SRVal);
    sn = sin(omega);
    cs = cos(omega);
    alpha = A*sn/(2*F0Val/BWVal); /* 对于Peak EQ, Q = Q/A */

    b0 = 1 + (alpha*A); /* 这里开始根据s域到z域的映射计算z域的系数 */
    b1 = -1 * cs; /* 这里在转换的时候本应当乘以-2, 但是CODEC内部已经乘以2, 所以这里只能乘以1 */
    b2 = 1 - (alpha*A);
    a0 = 1 + (alpha/A);
    a1 = -1 * cs; /* 同上, 这里本应乘以-2 */
    a2 = 1 - (alpha/A);
    b0 /= a0; /* 这里开始将a0变换为1 */
    if(fabs(b0) > absMax) absMax = fabs(b0); /* 获取b0, b1, b2的最大绝对值 */
    b1 /= a0;
    if(fabs(b1) > absMax) absMax = fabs(b1);
    b2 /= a0;
    if(fabs(b2) > absMax) absMax = fabs(b2);
    a1 /= (-1*a0); /* a1, a2乘以-1是因为CODEC滤波器的设计, 参见第一章的逻辑方框图 */
    a2 /= (-1*a0);
}
```

```

if(absMax > 1.0) { /* 确保b0, b1, b2为纯小数 */
    b0 /= absMax;
    b1 /= absMax;
    b2 /= absMax;
}
/* 转换为整数并打印结果。在显示的时候, 因为最高位是符号位, 所以如果超出位宽24/16位则需要去掉纯符号位, 比如FF81AA09实际数值取0x81AA09就是用于写到CODEC的系数数值 */
N0 = floor(b0*res);
N1 = floor(b1*res);
N2 = floor(b2*res);
D1 = floor(a1*res);
D2 = floor(a2*res);
printf("N0: %04X, N1: %04X, N2:%04X\r\nD1: %04X, D2: %04X\r\n", N0, N1, N2, D1, D2);
return 0;
}
    
```

## 6 参考文献

1. *Configure the Coefficients for Digital Biquad Filters in TLV320AIC3xxx Family* (SLAA447)
2. [https://en.wikipedia.org/wiki/Digital\\_biquad\\_filter](https://en.wikipedia.org/wiki/Digital_biquad_filter)
3. *TLV320AIC3204 Application Reference Guide* (SLAA557)
4. *TLV320AIC3101/4/6 datasheet*.

## 有关 TI 设计信息和资源的重要通知

德州仪器 (TI) 公司提供的技术、应用或其他设计建议、服务或信息，包括但不限于与评估模块有关的参考设计和材料（总称“TI 资源”），旨在帮助设计人员开发整合了 TI 产品的应用；如果您（个人，或如果是代表贵公司，则为贵公司）以任何方式下载、访问或使用了任何特定的 TI 资源，即表示贵方同意仅为该等目标，按照本通知的条款进行使用。

TI 所提供的 TI 资源，并未扩大或以其他方式修改 TI 对 TI 产品的公开适用的质保及质保免责声明；也未导致 TI 承担任何额外的义务或责任。TI 有权对其 TI 资源进行纠正、增强、改进和其他修改。

您理解并同意，在设计应用时应自行实施独立的分析、评价和判断，且应全权负责并确保应用的安全性，以及您的应用（包括应用中使用的 TI 产品）应符合所有适用的法律法规及其他相关要求。您就您的应用声明，您具备制订和实施下列保障措施所需的一切必要专业知识，能够 (1) 预见故障的危险后果，(2) 监视故障及其后果，以及 (3) 降低可能导致危险的故障几率并采取适当措施。您同意，在使用或分发包含 TI 产品的任何应用前，您将彻底测试该等应用和该等应用所用 TI 产品的功能。除特定 TI 资源的公开文档中明确列出的测试外，TI 未进行任何其他测试。

您只有在为开发包含该等 TI 资源所列 TI 产品的应用时，才被授权使用、复制和修改任何相关单项 TI 资源。但并未依据禁止反言原则或其他法律授予您任何 TI 知识产权的任何其他明示或默示的许可，也未授予您 TI 或第三方的任何技术或知识产权的许可，该等产权包括但不限于任何专利权、版权、屏蔽作品权或与使用 TI 产品或服务的任何整合、机器制作、流程相关的其他知识产权。涉及或参考了第三方产品或服务的信息不构成使用此类产品或服务的许可或与其相关的保证或认可。使用 TI 资源可能需要您向第三方获得对该等第三方专利或其他知识产权的许可。

TI 资源系“按原样”提供。TI 兹免除对 TI 资源及其使用作出所有其他明确或默认的保证或陈述，包括但不限于对准确性或完整性、产权保证、无复发故障保证，以及适销性、适合特定用途和不侵犯任何第三方知识产权的任何默认保证。

TI 不负责任何申索，包括但不限于因组合产品所致或与之有关的申索，也不为您辩护或赔偿，即使该等产品组合已列于 TI 资源或其他地方。对因 TI 资源或其使用引起或与之有关的任何实际的、直接的、特殊的、附带的、间接的、惩罚性的、偶发的、从属或惩戒性损害赔偿，不管 TI 是否获悉可能会产生上述损害赔偿，TI 概不负责。

您同意向 TI 及其代表全额赔偿因您不遵守本通知条款和条件而引起的任何损害、费用、损失和/或责任。

本通知适用于 TI 资源。另有其他条款适用于某些类型的材料、TI 产品和服务的使用和采购。这些条款包括但不限于适用于 TI 的半导体产品 (<http://www.ti.com/sc/docs/stdterms.htm>)、[评估模块](http://www.ti.com/sc/docs/sampters.htm)和样品 (<http://www.ti.com/sc/docs/sampters.htm>) 的标准条款。

邮寄地址：上海市浦东新区世纪大道 1568 号中建大厦 32 楼，邮政编码：200122  
Copyright © 2017 德州仪器半导体技术（上海）有限公司