

工频数字电源系统中的实时锁相技术

*Igor AN**Connected MCU Central FAE Shanghai*

摘要

在工频电源系统中，有许多应用需要进行锁相控制，如 PFC 应用中的将输入电流与电网电压进行锁相，太阳能逆变中的输出电流与电网电压进行锁相等。在各种锁相技术中，基于市电正弦波过零点检测的锁相控制技术被广泛应，但是过零点检测频率较低，同时软件算法又相对复杂，使此方法的实时性不高，锁相控制系统的带宽较低，锁相速度慢。本文基于简单的正弦余弦正交方法，引入了一种实时性很高的锁相控制算法，能够克服过零点检测锁相方法的缺点，在实际应用中获得了非常好的锁相控制性能。

同时，利用该锁相算法实现的锁相功能，更方便系统对电网侧的一些异常工况进行检测，迅速准确的检测出异常并报警，如太阳能逆变中的孤岛检测、PFC 应用中的电压丢失、缺相等。

关键词： 数字锁相、PLL、PFC、太阳能并网逆变器、TI、德州仪器、C2000

目录

1	前言.....	3
2	锁相的基本原理.....	3
3	过零点锁相算法.....	3
3.1	基本原理.....	3
3.2	过零点锁相法的局限	4
4	实时锁相算法	4
4.1	获取相位差	5
4.2	锁相环设计	6
5	算法在 C2000 高性能 MCU 上的实现	8
	参考文献	10

配图

图 1.	输出正弦的频率变化	3
图 2.	过零侦测锁相控制原理.....	4
图 3.	$f_w=120\text{Hz}$, $A_w=0.8$, $T_s=416\mu\text{s}$ 的频率特性	6
图 4.	基于正交法的锁相控制系统.....	6
图 5.	锁相控制系统传递函数方框图	7
图 6.	系统带宽为 25Hz 时的频率特性	7
图 7.	TMS320F28004x 框图	8
图 8.	TMU 加速下 Park 变换运算汇编程序示例	9

表格

表 1, TMS320F28004x TMU 性能列表	9
-----------------------------------	---

1 前言

近年来，电力系统对用电设备的谐波性能有了更严格的要求，所以更多的谐波校正电源被设计开发出来（PFC，APF...）。同时，太阳能、风能等新能源的应用日益广泛，特别是太阳能发电系统，已经成为取代传统发电方式的重要手段。在太阳能发电系统中，其并网逆变器的性能直接影响了太阳能发电的效能，决定着能否成功并网，能否实现功率因数可调，并最大限度的降低并网输出电流的谐波等主要功能。

无论是 PFC、APF 或是太阳能逆变器等这类工频电源应用中，锁相控制技术为其关键技术之一。但是，目前最为流行的基于过零点侦测的锁相控制技术，其锁相调节系统的带宽太低，锁相速度太慢，当电网频率变化时，不能够很快的跟踪，并且会向电网注入瞬时谐波，影响其功率因数。同时，在太阳能逆变器的主动式孤岛侦测中，往往会定时修改参考频率，也会使输出电流波形出现畸变。实际上，过零点侦测的主要缺点是因为其侦测的频率与市电频率相同，这导致了锁相控制系统的采样频率过低，闭环控制的带宽就不能太高。所以，要改善锁相系统的控制性能，首先要从改善锁相控制系统的采样频率入手。

2 锁相的基本原理

锁相的目的就是要输出的正弦与参考正弦同频、同相。但是，对于任意相位的正弦波来说，必须通过实时的改变其周波频率、相位，最终才能实现与参考正弦同频同相。如下图所示：

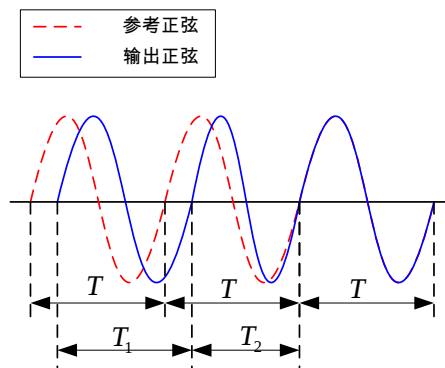


图 1. 输出正弦的频率变化

从图 1 可以看到，输出正弦在第二个周波通过改变频率至 $\frac{1}{T_2}$ ，就能够在第 3 个周波开始时，将相位差调整为 0；而在第 3 个周波开始与参考正弦同频同相运行，这就是数字锁相的基本思想。也就是先锁频率，再锁相位。

3 过零点锁相算法

3.1 基本原理

过零点锁相法的基本实现思想是，通过硬件电路将市电过零点信息提取出来，一般的会用方波翻转信号来表示过零点，处理器采集该方波在软件中标明对应的时刻为过零点时刻，再通过算法实现锁

相。由于每个市电周期只有两个过零点时刻，所以利用过零点方法进行锁相，环路的最高相应速度只能做到市电周期的一半。利用过零点锁相的一般实现办法，是将过零点信号进行处理，得出相位差信息，并通过改变频率的办法实现频率和相位的跟踪，即柔性频率跟踪。

上述基于过零点侦测的锁相方式，由于过分依赖硬件过零点，所以硬件过零点检测电路引入的延迟和干扰，对锁相效果会造成较大的不良影响，增加了系统不稳定的因素和硬件电路设计、布板的难度。同时，此种方式必须在每个周波开始时，计算出相位差，由此信息通过控制器控制确定如何调整下一个周期输出正弦信号的频率，进一步降低了锁相环路的带宽。

柔性频率跟踪锁相法的控制框图如下图 2 所示：

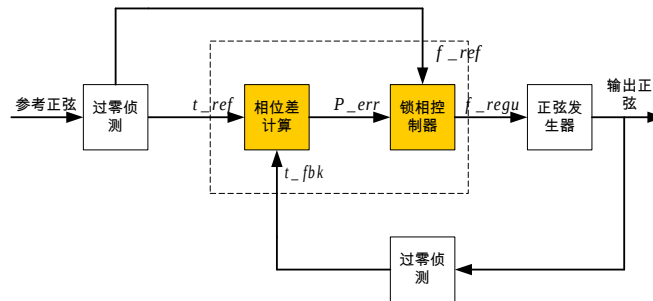


图 2. 过零侦测锁相控制原理

图 2 中： t_{ref} 是参考过零时刻； t_{fbk} 是输出过零时刻； P_{err} 是相位差； f_{ref} 是参考正弦频率； f_{regu} 是锁相控制器输出的调整频率，即输出正弦在下一个周波的频率。在具体的实现中，一般用硬件产生过零信号，并用 MCU（如 TI C2000）中的 Capture 单元去捕获参考正弦和输出正弦的过零点，并在软件中来处理锁相调节。

3.2 过零点锁相法的局限

从以上分析可以看出，过零点锁相调节的带宽受限于过零侦测的频率，因为每个市电周波只有一个正过零，也就决定了每个市电周波只能得到 1 次 P_{err} ，其采样频率和市电频率相同（如 50Hz），所以， f_{regu} 只能每个市电周波更新 1 次，这等效于整个闭环回路有一纯延迟环节 e^{-Ts} ，其延迟的时间为 1 个市电周期。所以，为了保证锁相系统的稳定性，闭环系统的带宽不能太高，一般小于 5Hz。

4 实时锁相算法

实时锁相算法，顾名思义就是时时刻刻都在检测相位差，时时刻刻都在进行锁相控制。相比较过零点锁相法，其最主要优点就是可以提供更高的带宽。

实时锁相法的基本思想是，利用采样，通过一定算法实时检测得到相位差，并利用其建立闭环控制环路，对信号相位进行控制。因此，实现实时锁相的第一步，就是要检测计算得到实时相位差。

相位差的概念其实包含两层含义：其一为频率差，其二为相位差。

只是实际实现中，一般的，对于大多数应用，尤其是工频应用中，频率是基本固定的，而且对于需要锁相的信号而言，其频率也一般是固定好的，不会经常变换，因此，算法核心需要解决的问题是相位差的问题。

4.1 获取相位差

为了得到相位差信息，首先假定被控信号的频率已经与参考信号一致，请对照图 1 参考理解。

假设 $\sin(\omega_1 t + \varphi_1)$ 为被控的输出正弦信号，则与之正交的余弦信号表达式为 $\cos(\omega_1 t + \varphi_1)$ ，另外，假设 $\sin(\omega_2 t + \varphi_2)$ 为锁相源参考正弦信号。

将输出正交余弦信号乘以参考正弦信号，可以得到如下的关系式：

$$\cos(\omega_1 t + \varphi_1) \sin(\omega_2 t + \varphi_2) = \frac{1}{2} \{ \sin[(\omega_1 + \omega_2)t + (\varphi_1 + \varphi_2)] + \sin[(\omega_1 - \omega_2)t + (\varphi_1 - \varphi_2)] \}$$

如前述，频率变化微小，因此输出信号频率已经设定为采集到的锁相源频率，即： $\omega_1 = \omega_2$ ，则：

$$\cos(\omega_1 t + \varphi_1) \sin(\omega_2 t + \varphi_2) = \frac{1}{2} \{ \underbrace{\sin[(2\omega_1)t + (\varphi_1 + \varphi_2)]}_{\text{(高频分量)}} + \underbrace{\sin(\varphi_1 - \varphi_2)}_{\text{(低频分量)}} \} \quad \text{式 1}$$

由式 1，当 $\varphi_1 - \varphi_2$ 较小时，有：

$$\sin(\varphi_1 - \varphi_2) \approx (\varphi_1 - \varphi_2)$$

故式一可变为：

$$\cos(\omega_1 t + \varphi_1) \sin(\omega_2 t + \varphi_2) \approx \frac{1}{2} \sin[(2\omega_1)t + (\varphi_1 + \varphi_2)] + \frac{1}{2} (\varphi_1 - \varphi_2) \quad \text{式 2}$$

从式 2 看出，低频分量所表示的就是我们感兴趣的相位差信息，只要将其从式 2 中提取出来就即可。

为了提取相位差信息，需要将式 2 中的高频分量去除。可以直接选取适当滤波器进行滤除，但为了保证较高带宽，低通滤波器是不够好的选择，这里推荐使用 Notch 滤波器或者梳状滤波器来对倍频信号进行滤除。

常用 Notch Filter 离散传递函数如式 3 所示：

$$G_{\text{Notch}}(z) = \frac{(1+A_w)(1-2B_w z^{-1}+z^{-2})}{2[1-B_w(1+A_w)z^{-1}+A_w z^{-2}]} \quad \text{式 3}$$

其中：

A_w 为通带带宽因子

T_s 为采样时间

B_w 为选择频率因子， $B_w = \cos(2\pi T_s f_w)$

f_w 为 Notch 中心频率

A_w 越大时，则阻带越窄，反之越宽，其经验值为 0.8。图 3 给出了中心频率为 120Hz 时的 Notch 滤波器的频率特性，可以看出，在中心频率处，其幅频特性衰减非常大，并且在中心频率附近处也有一定的衰减幅度，同时在远离中心频率时，衰减很小，具有非常好的选择特性。另外，在远离中心频率时，相频特性的相角滞后也比较小，保证了低频分量的采样延迟较小。

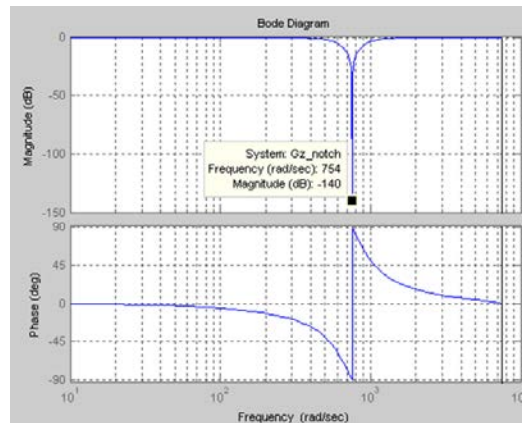


图 3. $f_w=120\text{Hz}$, $A_w=0.8$, $T_s=416\mu\text{s}$ 的频率特性

经 Notch 滤除后的信号，只剩下相位差信息，实现了最初的目的。

从上述分析可知，将采样得到的正弦信号，进行余弦正交计算就可以得到需要的相位差信息。

4.2 锁相环设计

以上我们得到了实时获得相位差的方法，那么就可以基于此方法来设计锁相控制器，由于采样频率的提高，使得整个锁相控制系统的带宽大大提高，锁相速度也大幅度提升。图 4 为基于正交法的锁相控制系统框图。

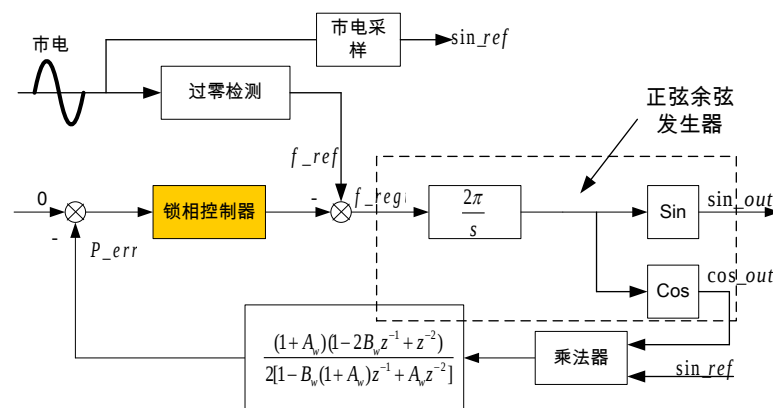


图 4. 基于正交法的锁相控制系统

图中的过零检测环节，可以选择与基于过零点侦测锁相中使用的相同的硬件过零点检测法，也可以完全采用软件过零检测的方式。由于在正交法锁相中，过零点的检测只是为了计算频率信息，因此对过零点的精度要求大幅下降，因此相较前种方法，引入误差的概率大大降低。

另外，在选择锁相控制器时，可选择 PI 调节器，但如果想要增强高频段的衰减，可在 PI 调节器中增加 1 个高频极点。如式 4 所示：

$$G_{c_PLL}(s) = \frac{K(s+a)}{s(s+b)}$$

式 4

则锁相控制系统的等效传递函数方框图如下：

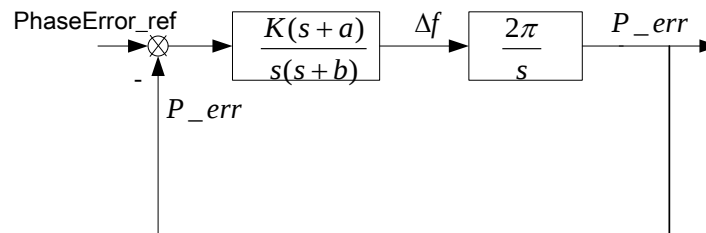


图 5. 锁相控制系统传递函数方框图

虽然理论上能够使锁相系统的带宽无限提高，但是并不是带宽越大越好。因为，实际中的市电并不是标准的正弦，其含有多次谐波的分量，这就使得这些谐波会叠加到相位差的计算中，必须依靠环路的滤波特性将其滤除。若在选择锁相控制器时，系统的带宽过大，则锁相控制器的滤波特性就不足，导致产生的正弦也会叠加各次谐波，THD 增加。所以，要在满足快速的同时，也要满足波形质量的要求，一般将带宽设置在 50Hz 以内，这样对于 3 次以上的谐波都有一定的滤波特性。图 6 为带宽在 25Hz 时的系统开环频率特性。

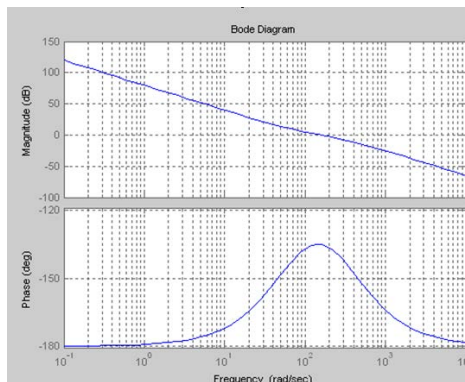


图 6. 系统带宽为 25Hz 时的频率特性

另外，由于实际中的采样具有一定的延时，同时，控制回路也会有一定的延迟，导致最终的输出波形会有滞后的情况，在这种情况下，应该对系统的延迟进行补偿。例如可将环路的给定设置为非 0，而是带有一定的偏移。

最后需要指出，Notch 滤波器的陷波滤波频率选择特性只在一定范围内效果最佳，当被滤波信号的频率波动超过了系统阻带的宽度，则 Notch 滤波器会快速失效，失去陷波滤波效果，无法有效的滤除相位差获取算法中的高频分量，也就无法有效获得相位差信息，最终导致锁相环无法正常工作。实际应用中，对于 50Hz 和 60Hz 需要设置不同的 Notch 滤波器的参数，当市电频率超出对应频率系统范围时，则应该加以保护。

5 算法在 C2000 高性能 MCU 上的实现

从上述分析推导过程可以看出，整个锁相环算法中包含大量的正弦、余弦计算，并且整个运算过程对实时性要求较高。结合整个控制系统功能（如：PFC，逆变器等）的其他实时控制环路（电压环，电流环等），在具体实现上，对 MCU 的性能提出了较高的要求，如高速 ADC 采样能力，PWM 信号的更新速率，以及 CPU 的实时计算效率、能力等。

Texas Instruments 公司的 C2000 系列 MCU 充分满足上述要求，尤其是新一代 C2000 芯片中高性价比的 TMS320F28004x[3]系列，非常适合上述控制系统类似的应用场景，其系统框图如下图：

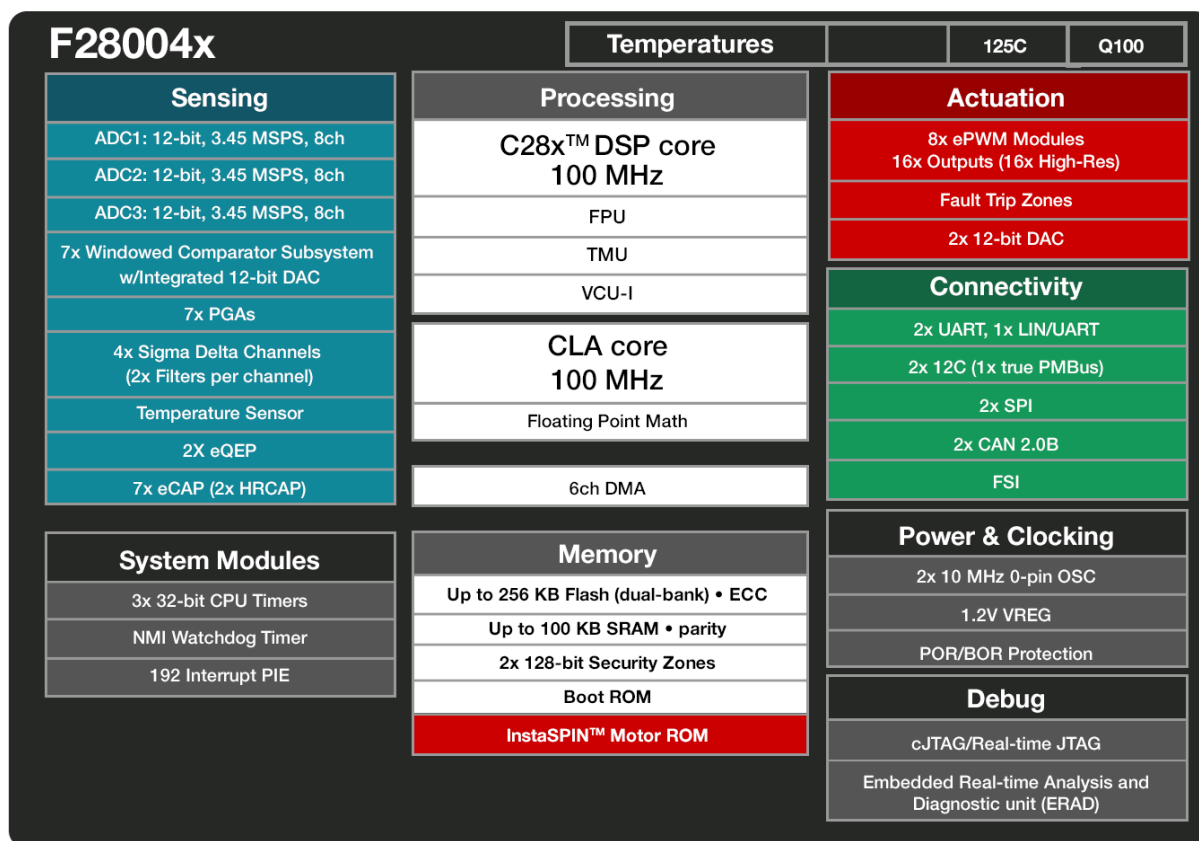


图 7. TMS320F28004x 框图

该芯片包含三组独立 12 位 ADC 采样模块，16 路 Type4 ePWM 模块，使之继承并进一步升级了 C2000 强大的 ADC 和 PWM 功能。同时，最新加入了 TMU（三角函数运算硬件加速单元），使得正弦、余弦、

除法等运算的运算速度大幅提升，TMU 硬件运算加速单元支持的运算硬件指令如下表 1 详述，而图 8 是一个 Park 变换使用 TMU 时所需要的汇编代码示例。

表 1, TMS320F28004x TMU 性能列表

INSTRUCTIONS	C EQUIVALENT OPERATION	PIPELINE CYCLES
MPY2PIF32 RaH,RbH	$a = b * 2\pi$	2/3
DIV2PIF32 RaH,RbH	$a = b / 2\pi$	2/3
DIVF32 RaH,RbH,RcH	$a = b/c$	5
SQRTF32 RaH,RbH	$a = \text{sqrt}(b)$	5
SINPUF32 RaH,RbH	$a = \sin(b*2\pi)$	4
COSPUF32 RaH,RbH	$a = \cos(b*2\pi)$	4
ATANPUF32 RaH,RbH	$a = \text{atan}(b)/2\pi$	4
QUADF32 RaH,RbH,RcH,RdH	Operation to assist in calculating ATANPU2	5

由图 8 可以看到，通常在一个 32 位浮点处理器上需要 80-100 个时钟周期执行的 Park 变换，在使用 TMU 的情况下，仅仅需要 13 个时钟周期即可完成，减少了 80%左右的执行时间，大大提高了核心代码的执行效率。

$$\text{Park} \begin{bmatrix} i_d \\ i_q \\ i_o \end{bmatrix} = \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} X \begin{bmatrix} i_\alpha \\ i_\beta \\ i_o \end{bmatrix}$$

```
// Park Transform:
Id = Ia * cos(AnglePU*2pi) + Ib * sin(AnglePU*2pi)
Iq = Ib * cos(AnglePU*2pi) - Ia * sin(AnglePU*2pi)
```

```
MOV32    R0,@PUAngle    ; R0 - Per Unit Angle
PUSINF32 R1,R0           ; R1 - sin(AnglePU*2pi)
PUCOSF32 R2,R0           ; R2 - cos(AnglePU*2pi)
MOV32    R3,@Ia          ; R3 - Ia
MOV32    R4,@Ib          ; R4 - Ib
MPYF32   R6,R4,R1        ; R6 - Ib * sin(AnglePU*2pi)
MPYF32   R7,R3,R2        ; R7 - Ia * cos(AnglePU*2pi)
MPYF32   R0,R4,R2        ; R0 = Ib * cos(AnglePU*2pi)
MPYF32   R1,R3,R1        ; R1 = Ia * sin(AnglePU*2pi)
ADDF32   R7,R7,R6        ; R7 = Ia * cos(AnglePU*2pi) + Ib * sin(AnglePU*2pi)
SUBF32   R1,R0,R1        ; R1 = Ib * cos(AnglePU*2pi) - Ia * sin(AnglePU*2pi)
MOV32    @Id,R7          ; Id = R7
MOV32    @Iq,R1          ; Iq = R1
; 13 instructions, 13 cycles (65ns @200MHz)
```

图 8. TMU 加速下 Park 变换运算汇编程序示例

本文介绍的锁相算法已经在 C2000 controlSUITE 中的 Solar Kit 例程中应用并验证，其对应的公开源码文件所在位置为：

“~\controlSUITE\libs\app_libs\solar\v1.2\IQ\include\SPLL_1ph_IQ.h”；

可以通过下面路径中的工程

“~\controlSUITE\development_kits\SolarExplorer_v1.1\SolarExplorer_PVInverter_F2803x”
打开，结合工程应用来理解锁相环在整个控制环路中的应用。

参考文献

- [1]*Operation of a Phase Locked Loop System Under Distorted Utility Conditions.* Vikram Kaura, Member, IEEE, and Vladimir Blasko, Member, IEEE.
- [2]*Fast Design Algorithms for FIR Notch Filters.* Miroslav Vlek and Ladislav Jires Czech Technical University Faculty of Electrical Engineering Technick 2, 166 27 Praha 6 Czech Republic.
- [3]*TMS320F28004x Piccolo™ Microcontrollers*

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 产品的功能而设计。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，该等许可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示的保证或陈述，包括但不限于对准确性或完整性、产权保证、无屡发故障保证，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 © 2018 德州仪器半导体技术（上海）有限公司