



Matthew Pate

## 摘要

性能、效率、灵活性和保护性 - 这些是电力电子技术 (如电机控制、数字电源、可再生能源、照明和电动汽车) 的重要属性。20 多年与客户合作开发实时控制应用的经验, C2000™ 实时微控制器 (MCU) 平台使开发人员能够经济高效地满足上述所有标准, 同时还能使他们的设计脱颖而出。本应用报告旨在对使实时控制系统脱颖而出的器件进行更深入的研究, 并提供了后续评估步骤。

## 内容

<b>1 C2000 和实时控制</b>	4
1.1 入门资源	6
1.2 处理	6
1.3 控制	7
1.4 检测	8
1.5 接口	9
1.6 功能安全	10
<b>2 传感关键技术</b>	11
2.1 模拟信号的精确数字域表示	11
2.2 优化模拟输入的采集时间与电路复杂度	13
2.3 使用单引脚基准对双阈值进行基于硬件的监视	15
2.4 解决 ADC 采样期间的容差和老化效应问题	17
2.5 使用 C2000 可配置逻辑块实现旋转传感解决方案	18
2.6 跨隔离边界的智能传感	20
2.7 在高带宽控制拓扑中启用内部周期更新	22
2.8 无需信号调节即可准确监控实时控制系统事件	24
<b>3 处理关键技术</b>	25
3.1 三角函数加速器	25
3.2 快速板载整数除法	27
3.3 对双精度浮点运算的硬件支持	29
3.4 通过独立的处理单元增加控制环带宽	31
3.5 灵活的系统互连: C2000 X-Bar	32
3.6 使用非线性 PID 控制改善控制性能	34
3.7 了解实时控制应用中的闪存性能	35
3.8 使用 C28x DSP 内核进行确定性程序执行	38
3.9 高效的实时固件更新 (LFU) 和固件无线 (FOTA) 更新	40
<b>4 控制关键技术</b>	43
4.1 使用 C2000 HRPWM 减少控制系统中的极限环振荡	43
4.2 具有可配置死区的电流控制拓扑击穿预防措施	45
4.3 使用 C2000 可配置逻辑块进行片上硬件定制	46
4.4 快速检测过电流及欠电流和欠电压	49
4.5 通过高分辨率相位控制提高系统功率密度	51
4.6 高频、多相和变频拓扑中的安全和优化的 PWM 更新	53
4.7 解决分散型控制系统中跨多个控制器的事件同步问题	55
<b>5 接口关键技术</b>	57
5.1 对 C2000 外设的直接主机控制	57
5.2 通过 AES 引擎保护外部通信和固件更新	59

5.3 跨隔离边界的分布式实时控制.....	60
5.4 使用嵌入式图形发生器 (EPG) 进行自定义测试和数据图形生成.....	62
<b>6 安全关键技术.....</b>	<b>63</b>
6.1 作为控制环路一部分的非侵入式运行时间监测和诊断.....	63
6.2 C28x CPU 硬件内置自检.....	65
6.3 嵌入式片上存储器的零 CPU 开销循环冗余校验.....	66
6.4 代码执行前的引导代码身份验证.....	67
<b>7 参考文献.....</b>	<b>68</b>
7.1 器件列表.....	68
7.2 硬件/软件资源.....	68
7.3 文档.....	69
<b>8 修订历史记录.....</b>	<b>69</b>

## 插图清单

图 1-1. 常见的 C2000 实时控制应用.....	4
图 1-2. C2000 实时 MCU 器件.....	4
图 1-3. 实时信号链组成.....	5
图 1-4. C2000 处理.....	6
图 1-5. PWM 方框图.....	7
图 1-6. C2000 模拟集成.....	8
图 1-7. C2000 MCU 支持的接口.....	9
图 1-8. 功能安全启用器.....	10
图 2-1. 单端输入模型.....	13
图 2-2. 迟滞控制器中的阈值电平.....	15
图 2-3. CMPSS 方框图.....	15
图 2-4. TMS320F2837xD 上的 ADC 后处理块.....	17
图 2-5. 具有 T-Format 绝对位置编码器接口的工业伺服驱动器.....	19
图 2-6. QepDiv 输入和输出图.....	20
图 2-7. C2000 SDFM 内部的滤波器和解调器.....	21
图 2-8. TMS320F2837xD MCU 上带有主滤波器和次滤波器块的 SDFM 模块.....	21
图 2-9. 实时信号链.....	22
图 2-10. PWM 周期内更新.....	23
图 2-11. CMPSS 原始输出与滤波输出.....	24
图 3-1. 派克变换.....	25
图 3-2. TMU 对通用变换的改进.....	25
图 3-3. 截断除法函数.....	27
图 3-4. 取整除法函数.....	28
图 3-5. 欧几里德除法函数.....	28
图 3-6. C28x 和 CLA 与 ADC 和 ePWM 模块之间的连接.....	31
图 3-7. TMS320F2837xD MCU 上的输入 X-Bar.....	32
图 3-8. TMS320F2837xD MCU 上的本地多路复用器和逻辑或.....	33
图 3-9. TMS320F2837xD MCU 上的 X-Bar 源和目标.....	33
图 3-10. 非线性 PID 方框图.....	34
图 3-11. 线性和非线性 PID 的响应时间比较.....	35
图 3-12. C2000 闪存预取模块.....	36
图 3-13. C28x 管线可视化图.....	39
图 3-14. C28x CPU 可屏蔽中断的标准操作.....	39
图 4-1. HRPWM 功能与传统 PWM 生成方法比较.....	43
图 4-2. 同步升压控制器.....	45
图 4-3. 具有可配置死区的 COMP 模块的逐周期跳闸动作.....	46
图 4-4. C2000 MCU 架构中的 CLB 集成.....	47
图 4-5. SysConfig 中的 CLB 配置工具.....	48
图 4-6. CMPSS 可视化图.....	49
图 4-7. 双有源电桥方框图.....	51
图 4-8. 影子到活动加载操作.....	54
图 4-9. 非同步与同步事件触发.....	55
图 4-10. 主控器件和节点器件实现.....	55
图 4-11. 菊花链网络同步和时序.....	56

图 5-1. 适用于 FSI 应用的 HIC 桥.....	58
图 5-2. 适用于位置编码器应用的 HIC 桥.....	58
图 5-3. AES 方框图.....	59
图 5-4. 全双工 3 线 FSI 实现.....	60
图 5-5. FSI 偏斜补偿.....	61
图 5-6. 带偏移的时钟.....	62
图 5-7. 串行流和时钟.....	62
图 6-1. ERAD 方框图.....	63
图 6-2. 栈溢出保护.....	64
图 6-3. TMS320F2838xD 器件上的 BGCRC 实现.....	66
图 6-4. CMAC 工作原理.....	67

## 表格清单

表 2-1. TMS320F28379D 16 位 ADC 规格.....	12
表 2-2. 采集时间配置范围 ( 每个通道 ) .....	14
表 2-3. 随时间变化的典型电阻容差和系统影响.....	17
表 2-4. 采样延迟导致的潜在相位错误.....	23
表 3-1. TMU 支持的指令汇总.....	26
表 3-2. 使用和不使用 FID 模块的整数除法.....	28
表 3-3. FPU64 和 FPU32 之间的周期比较.....	30
表 3-4. 数据类型大小 C28x 与 Arm.....	30
表 3-5. 通过预取功能实现的有效闪存访问时间.....	37
表 4-1. PWM 与 HRPWM 的分辨率比较.....	44
表 4-2. 故障检测与跳闸方法的比较.....	49
表 4-3. 相移要求满足 DAB 拓扑中 1% 的输出容差.....	52
表 6-1. HWBIST 支持的诊断覆盖率 ( 因器件而异 ) .....	65

## 商标

C2000™ and Code Composer Studio™ are trademarks of Texas Instruments.

Bosch™ is a trademark of Robert Bosch GmbH, Germany.

Arm® is a registered trademark of Arm Limited (or its subsidiaries) in the US and/or elsewhere.

所有商标均为其各自所有者的财产。

## 1 C2000 和实时控制

从 1997 年开始，德州仪器 (TI) 在单个器件上集成了闪存、模数转换器 (ADC)、数字信号处理器 (DSP) 和脉宽调制 (PWM) 单元。首个 C2000 实时控制 MCU 就此诞生。

二十多年来，此器件系列不断壮大，如今在工业和汽车应用等众多应用中可以找到数百万个 C2000，如电机控制、光伏逆变器、数字电源、电动车辆等（请参阅图 1-1）。有一条线将上述所有应用联合在一起；它们的实时性以及对于实时控制器的需求。

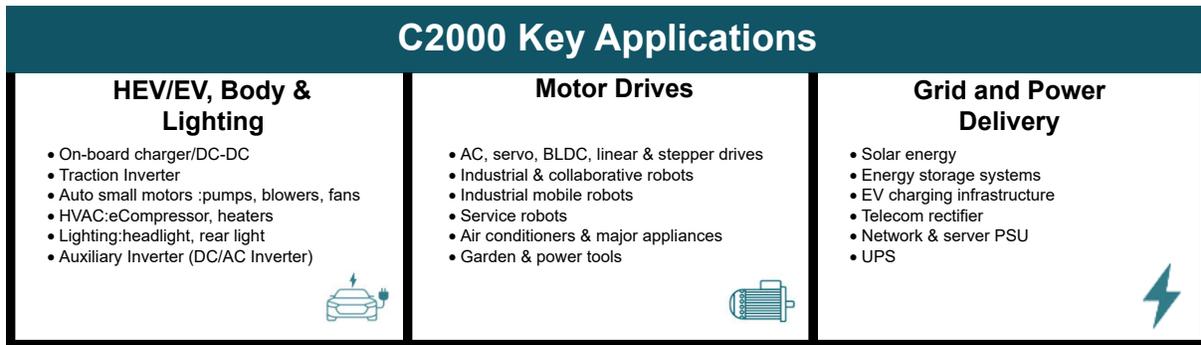


图 1-1. 常见的 C2000 实时控制应用

实时控制系统通常由四个主要元素组成：请参阅图 1-2：

- 感测**：或反馈采集。该应用需要以精确的方式并在非常精确的时刻测量多个关键参数（电压、电流、电机转速、温度）。
- 处理**：使用感测信息将控制算法应用于输入数据并计算下一个输出命令。
- 控制**：该命令通常通过驱动电力电子系统的 PWM 单元施加到系统，例如，电机转动得更快，流向太阳能安装系统的电流减少，汽车正在加速。
- 接口**：器件与其他外部元件通信的功能。尽管不必参与系统控制，但与其他系统组件的通信也必须与主控制回路共存。

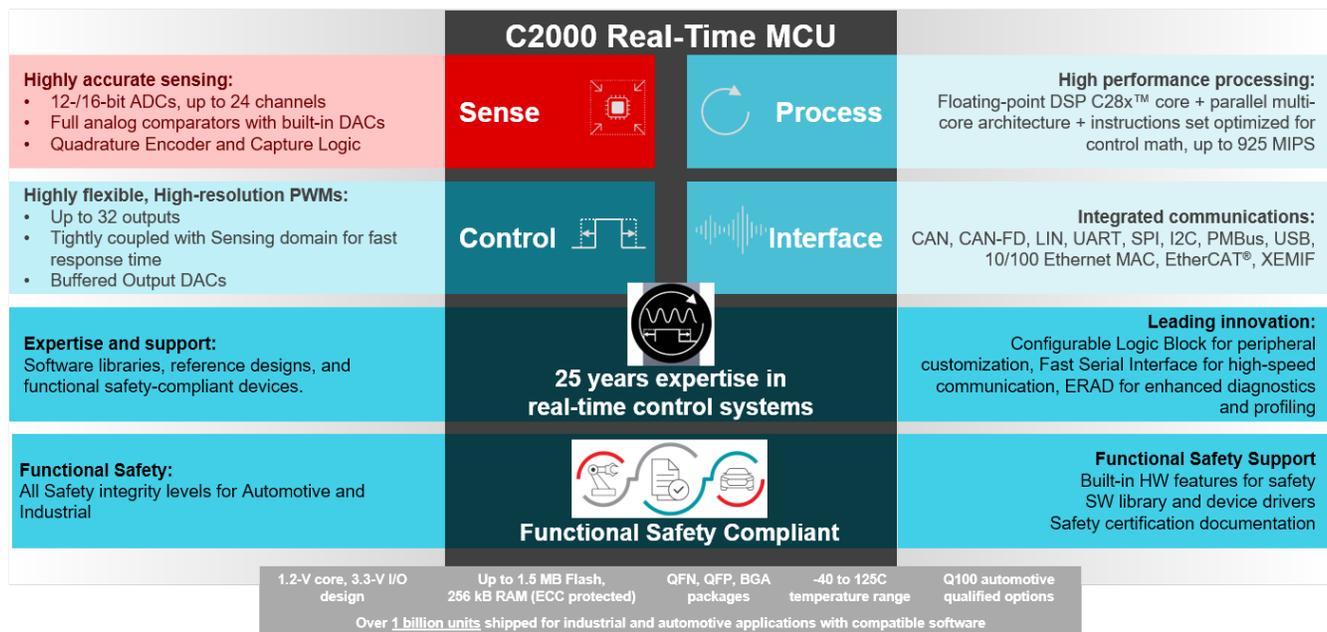


图 1-2. C2000 实时 MCU 器件

实时控制的关键是尽可能缩短感测、处理和控制的间隔时间：此时间定义为实时信号链。图 1-3 说明了为何整个过程对于理解实时控制器的整体系统性能至关重要，而不是简单地看一下主处理单元完成控制算法所需的时间（第 4 步）。

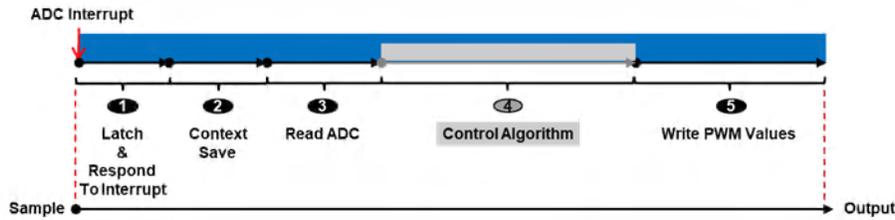


图 1-3. 实时信号链组成

许多基准测试仅关注完成第 4 步所需的时间，通常以每秒百万指令 (MIPS) 表示，而设计人员的实时挑战是尽可能缩短采样到输出的间隔时间：实时 MCU 架构选择至关重要。

C2000 实时 MCU 是可扩展、超低延迟的实时控制器平台，旨在提高电力电子产品的效率，例如高功率密度、高开关频率、[GaN 和 SiC 技术](#)，并在设计时考虑优异的实时信号链性能，而且可以提供比基于 [Arm®](#) 的架构高两倍的实时信号链性能。

以下各节从 CPU 周期数的角度展示了这些优势，以便于比较：

- [三角函数加速器](#)
- [了解实时控制应用中的闪存性能](#)

有关 C2000 为实时信号链带来的优势（包括软件基准测试）的更多详细信息，请参阅 [信号链基准 - 如何优化 C2000™ MCU 的实时性能](#)。

下一部分将重点介绍让 C2000 MCU 能够提供优异实时性的每个关键要素。凭借 25 年的专业知识（已转化为业内最大的功率变换参考设计产品），并提供可靠的 [可直接用于生产环境的软件](#)和 [开源硬件文档](#)，设计人员现在可以进行创新并构建未来能源系统。C2000 实时 MCU 继续通过软件兼容的器件平台从低端扩展到高端。若要随时了解 C2000 实时 MCU 提供的新创新，请参阅 [TI.com 首页](#)并注册 [TI 新闻稿](#)。

## 1.1 入门资源

Hardware	Software/Tools	Training
<ul style="list-style-type: none"> <li>选择器件                             <ul style="list-style-type: none"> <li>器件选型指南</li> <li>外设参考指南</li> </ul> </li> <li>选择 C2000 EVM                             <ul style="list-style-type: none"> <li>Launchpad：低成本评估板</li> <li>controlCARD：功能齐全的开发板</li> <li>购买一个 EVM 开始使用</li> </ul> </li> <li>参考设计</li> </ul>	<ul style="list-style-type: none"> <li>软件概要</li> <li>工具链                             <ul style="list-style-type: none"> <li>Code Composer Studio                                     <ul style="list-style-type: none"> <li>独立下载</li> <li>基于云</li> </ul> </li> <li>SysConfig 和 PinMux 支持</li> </ul> </li> <li>C2000ware SDK - 底层驱动程序和高度优化的库                             <ul style="list-style-type: none"> <li>Digital Power SDK</li> <li>带 InstaSpin 的电机控制 SDK</li> </ul> </li> <li>基于模块的评估                             <ul style="list-style-type: none"> <li>MathWorks Embedded Coder</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>入门</li> <li>C2000 Academy</li> <li>电机控制研讨会系列</li> <li>数字电源培训系列</li> <li>电动汽车培训模块</li> <li>安全概述</li> </ul>

## 1.2 处理

如图 1-4 中所示，C2000 实时 MCU 使用 C28x DSP ( 数字信号处理器 ) 内核作为主处理单元。同时支持 32 位浮点或定点运算，具有专门为实时控制应用定制的指令。下面概述了此子系统的其他组件：

- 控制律加速器 (CLA)：基于状态机的 32 位浮点协处理器，能够独立于 C28X DSP 内核执行代码运行
- C28x 扩展指令：
  - 浮点单元 (FPU)：支持 32 位浮点运算，在部分器件上支持 64 位浮点。
  - 三角数学单元 (TMU)：提供内在指令以支持常见的三角数学函数，这些函数通常在变换和扭矩环路计算中找到。
  - Viterbi 和 CRC 单元 (VCU)：减少复杂数学方程式中的 Viterbi 和循环冗余校验码 (CRC) 运算的周期数。

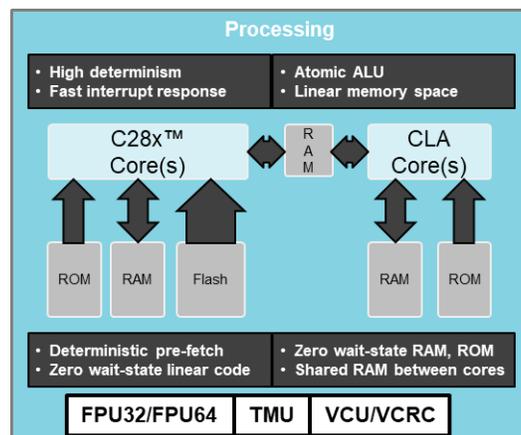


图 1-4. C2000 处理

### 1.3 控制

控制子系统包括将在受控状态下激励系统的模块。通常，这是通过脉宽调制 (PWM) 输出完成的。也可以通过片上数模转换器 (DAC) 的输出，或通用输入/输出 (GPIO) 引脚来完成。

- PWM - C2000 实时 MCU 上的机理驱动模块。负责驱动大多数电力电子系统中的外部场效应晶体管 (FET)。支持用于波形的占空比、周期和死区控制的标准和高分辨率模式
  - 特定 C2000 上的所有 PWM 都具有全局装载能力，从而确保多个开关控件之间的同步更新
  - 高分辨率 (约 150ps) 边沿位置有助于减少控制系统中的限制循环。可用于占空比、相位和死区位置。
  - 与模拟比较器紧密耦合，以提供过压/电流保护，适用于诸如峰值电流模式控制之类的系统的 CPU 独立占空比控制，以及适用于诸如峰值电流模式控制之类的系统的 CPU 独立占空比控制。
- 缓冲 DAC - 12 位 DAC，能够驱动定义的外部负载。通常用于在模拟域中产生偏置电压。
- 可配置逻辑块 (CLB) - 在硬件域的内部信号节点上运行的一组查找表和状态机逻辑。可以用于实现增强的系统性能的终点或中间步骤。

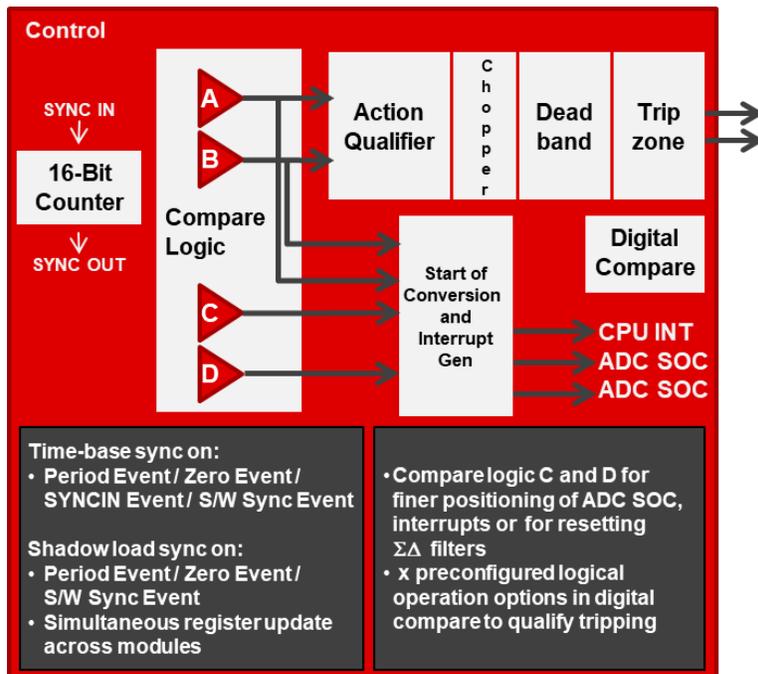


图 1-5. PWM 方框图

## 1.4 检测

采样子系统包括一些可将受控外部系统（模拟域）的状态转换为 C2000 实时控制器所使用数据（数字域）的模块。通常，该任务由 MCU 上的模数转换器 (ADC) 执行，但也可以由外部 ADC 的比较器或解调器来处理。该域中还包括其他单元转换器，例如正交编码器和时间脉冲测量器件。

- **ADC** - 多个 **12 或 16 位 ADC**，主要用于将受控系统的电压或电流（通过分流器）转换为数字域。支持内部基准电压或外部基准电压，采样率分别为 **3.5MSPS**（12 位）和 **1MSPS**（16 位），可将系统条件快速转换为控制系统可以动作的信息。
- **比较器 (COMP)** - 多个片上比较器通过比较系统电压与内部基准点（由内部 12 位 DAC 生成）来提供系统保护和逐周期 PWM 控制。直接连接到 **ePWM 模块**可以在不需要 CPU 干预的情况下以最快的速度改变输出状态。
- $\Sigma$ - $\Delta$  解调器 - 用于解码外部  $\Sigma$ - $\Delta$  ADC 输出的串行位流的片上逻辑。通常用于跨越大功率系统的隔离边界。包括硬件阈值检测。
- **正交编码器脉冲测量器 (eQEP)** - 对来自各种编码器的脉冲进行计数，以确定电机轴位置。可与 **CLB 模块** 配合使用，从而实现硬件中的客户编码器解决方案。
- **时间捕获 (eCAP)** - 测量外部脉冲事件之间的持续时间，用于评估霍尔传感器。标准和高分辨率模式均可用。

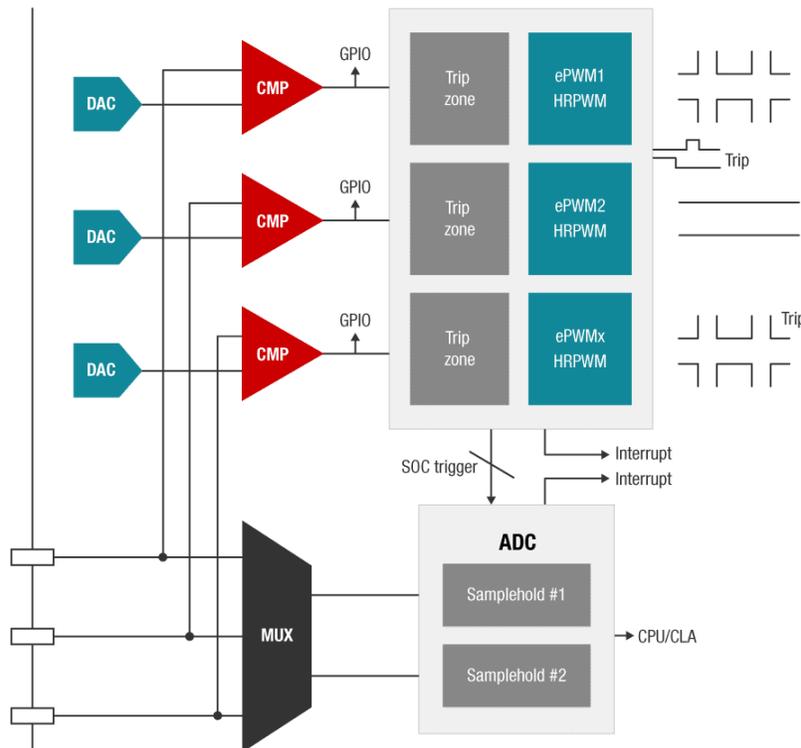


图 1-6. C2000 模拟集成

## 1.5 接口

从系统集成的角度来看，几乎总是需要一个额外的模块，尽管该模块通常与系统的实时控制环路并不相关。接口模块与 CPU 的良好集成对于避免可能影响控制环路的高开销至关重要。从串行数据流到多通道输入，以及从业界通用选项到专有格式，接口子系统均支持各种通信选项。

- 控制器局域网 (CAN) - CAN 模块支持 Bosch™ CAN 协议标准。
- 外部存储器接口 (EMIF) - 通常用于支持与 SDRAM 以及宽总线外设连接的并行数据总线。
- EtherCAT 从站控制器 (EtherCAT) - 此模块允许 C2000 MCU 充当 EtherCAT 网络中的从站节点。
- 以太网 - 10/100Mbps 以太网控制器和用于通过此总线进行外部通信的物理接口。
- 快速串行接口 (FSI) - 2 或 3 线单工串行数据的发送或接收。设计用于满足高速 (100Mbps) 需求以及跨越隔离边界时引入的可变延迟。
- 主机接口控制器 (HIC) - 让其他器件能够控制 C2000 外设或与其交互
- 集成电路总线 (I2C) - I2C 总线的接口/控制器
- 串行外设接口 (SPI) - 标准串行外设接口总线的接口/控制器。
- 通用异步接收器/发送器 (UART) - 通用异步接收器发送器总线的接口/控制器
- 通用串行总线 (USB) - 用于连接标准 USB 网络的 USB 2.0 MAC 和 PHY。

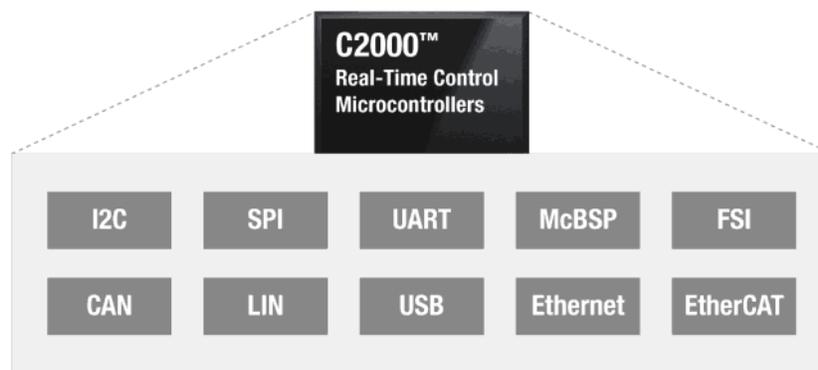


图 1-7. C2000 MCU 支持的接口

### 备注

外设计数以及功能可能会因器件而异。有关器件上外设数量的完整列表，请参阅每节末尾引用的数据表。有关特定器件支持的功能集，请参阅 [C2000 实时控制外设参考指南](#)。

## 1.6 功能安全

根据国际电工委员会 (IEC) 的定义，安全是指不存在因财产或环境损害而直接或间接导致人身伤害或健康损害的不可接受的风险。IEC 将功能安全定义为整体安全的一部分，这取决于系统或器件是否能够对其输入做出正确的响应。

凭借 TUV SUD 定义并独立评估其有效性的 300 多种安全机制，C2000 MCU 提供所需的诊断覆盖范围，以满足元件级 SIL 2/ASIL B 的随机硬件能力。功能安全手册提供了有关安全机制的详细信息，以及实现元件之间不干扰和避免相关故障的技术，以帮助客户开发符合 SIL 3/ASIL D 要求的合规系统。

可以在[功能安全启用器](#)中查看 C2000 器件中的功能安全启用器示例。

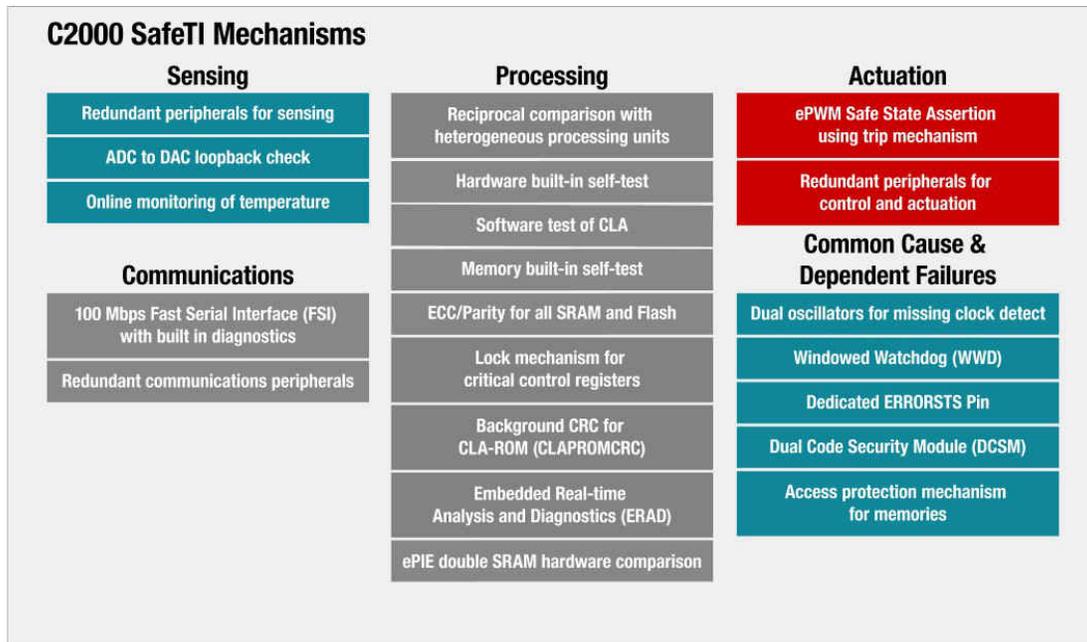


图 1-8. 功能安全启用器

有关更多详细信息，请参阅以下技术文档：

- [C2000™ 实时微控制器的工业功能安全](#)
- [C2000™ 实时微控制器的汽车功能安全](#)

## 2 传感关键技术

### 2.1 模拟信号的精确数字域表示

#### 2.1.1 价值主张

许多 MCU 都集成了 ADC 作为其采样子系统的一部分。ADC 准确地将模拟域转换为数字空间的能力是 MCU 实现正确控制系统的关键因素之一。借助 C2000 MCU ADC 的数据表规格，可在系统实现之前正确评估系统的性能。

#### 2.1.2 深入介绍

选择实时控制系统的 MCU 时，第一步是相对简单的过程；即将 MCU 的器件与系统需求进行对比。还需要考虑内存大小、CPU 速度、使用的通信标准、模拟内容、I/O 数量等等。在考虑模拟模块（如 ADC）的适合性时，可以很容易地根据采样率、输入数量和电平做出决定。然而，在实践中，要做出此决定还需要考虑的内容还有很多。

人们在选择 ADC 时通常仅基于最高等级规格书，但在开发过程中，由于 ADC 本身的原因，系统性能受到限制：

- 系统是否会使用模拟输入进行频率分析？那样的话，在选择带有片上 ADC 的 MCU 时，必须考虑诸如 SNR 和 THD 之类的交流规格。
- 整体准确性是一个关键问题吗？INL、增益和偏移量等直流规格是要考虑的关键参数。

ADC 规格及其与系统相关性的简要概述：

- 交流规格：与转换器如何准确地解析来自其他噪声源的信号的基频音调相关参数。包括 SNR、SINAD、THD 和 SFDR，均以 dB 为单位。还包括 ENOB，它是被转换成比特数的 SINAD。通常，在选择 ADC 时会考虑基于 SINAD 的 SINAD 和 ENOB，其重要性将取决于最终应用。
- 直流规格：与转换器精度相关的参数，因为它应用于表示数字域中的模拟输入。包括增益、偏移量、DNL 和 INL。增益、偏移量和 INL 的加权和通常被称为“总体未调整误差”（[方程式 1](#)）。该方程式通常用于确定这些参数对转换精度的实际影响。

$$\sum \sqrt{(\text{Err}_{\text{gain}})^2 + (\text{Err}_{\text{offset}})^2 + (\text{Err}_{\text{INL}})^2} \quad (1)$$

其中

- $\text{Err}_{\text{gain}}$  是 ADC 的最大增益误差（以 LSB 为单位）
- $\text{Err}_{\text{offset}}$  是 ADC 的最大偏移量误差（以 LSB 为单位）
- $\text{Err}_{\text{INL}}$  是 ADC 的最大 INL 误差（以 LSB 为单位）

有关 C2000 ADC 的具体规格和参数，请参阅表 2-1，[此处](#)是数据表中指向该表的动态链接。

C2000 器件在数据表中列出的所有参数的最后一个方面是参数本身的隐含含义。对于具有最小/最大值的参数，在器件的整个工作范围和使用寿命内，这些参数都是有保证的。典型值 (TYP) 列对于所有参数也很重要，因为它表示参数在其工作范围内的平均性能。

**表 2-1. TMS320F28379D 16 位 ADC 规格**

参数	测试条件：	最小值	典型值	最大值	单位
ADC 转换周期		29.6		31	ADCCLK
上电时间 (将 ADCPWDNZ 设置为第一次转换后)				500	μs
增益误差		-64	±9	64	LSB
偏移量误差		-16	±9	16	LSB
通道间增益误差			±6		LSB
通道间偏移量误差			±3		LSB
ADC 间增益误差	所有 ADC 的 V <sub>REFHI</sub> 和 V <sub>REFLO</sub> 均相同		±6		LSB
ADC 间偏移量误差	所有 ADC 的 V <sub>REFHI</sub> 和 V <sub>REFLO</sub> 均相同		±3		LSB
DNL		> -1	±0.5	1	LSB
INL		-3	±1.5	3	LSB
SNR	V <sub>REFHI</sub> = 2.5V, f <sub>in</sub> = 10kHz		87.6		dB
THD	V <sub>REFHI</sub> = 2.5V, f <sub>in</sub> = 10kHz		-93.5		dB
SFDR	V <sub>REFHI</sub> = 2.5V, f <sub>in</sub> = 10kHz		95.4		dB
SINAD	V <sub>REFHI</sub> = 2.5V, f <sub>in</sub> = 10kHz		86.6		dB
ENOB	V <sub>REFHI</sub> = 2.5V, f <sub>in</sub> = 10kHz, 单个 ADC		14.1		位
	V <sub>REFHI</sub> = 2.5V, f <sub>in</sub> = 10kHz, 同步 ADC		14.1		
	V <sub>REFHI</sub> = 2.5V, f <sub>in</sub> = 10kHz, 异步 ADC		不支持		
PSRR	V <sub>DDA</sub> = 3.3V 直流 + 200mV 直流至正弦 (1kHz 时)		77		dB
PSRR	V <sub>DDA</sub> = 3.3V 直流 + 200mV 正弦 (800kHz 时)		74		dB
CMRR	DC 到 1MHz		60		dB
V <sub>REFHI</sub> 输入电流			190		μA
ADC 间隔	V <sub>REFHI</sub> = 2.5V, 同步 ADC	-2		2	LSB
	V <sub>REFHI</sub> = 2.5V, 异步 ADC		不支持		

### 2.1.3 器件列表

- [TMS320F2838xD/S](#)
- [TMS320F2837xD/S](#)
- [TMS320F2807x](#)
- [TMS320F28004x](#)
- [TMS320F28003x](#)
- [TMS320F28002x](#)
- [TMS320F280013x](#)
- [TMS320F280015x](#)

### 2.1.4 硬件平台和软件示例

所有用于其特定 C2000 MCU 的 controlCARD 都已通过验证，可以再现片上 ADC 的 DS 规格

- [TMDSCNCD28388D](#)
- [TMDSCNCD28379D](#)
- [TMDSCNCD280049C](#)
- [TMDSCNCD280025](#)
- [TMDSCNCD2800137](#)
- [TMDSCNCD2800157](#)

## 2.1.5 文档

- 德州仪器 (TI) : [具有连接管理器的 TMS320F2838x 实时微控制器数据表](#) ( 请参阅 C28x 模拟外设部分 )
- 德州仪器 (TI) : [TMS320F2837xD 双核微控制器数据表](#) ( 请参阅模拟外设部分 )
- 德州仪器 (TI) : [TMS320F28004x 微控制器数据表](#) ( 请参阅模拟外设部分 )
- 德州仪器 (TI) : [TMS320F28003x 微控制器数据表](#) ( 请参阅模拟外设部分 )
- 德州仪器 (TI) : [TMS320F28002x 微控制器数据表](#) ( 请参阅模拟外设部分 )
- 德州仪器 (TI) : [TMS320F280013x 微控制器数据表](#) ( 请参阅模拟外设部分 )
- 德州仪器 (TI) : [TMS320F280015x 微控制器数据表](#) ( 请参阅模拟外设部分 )

## 2.2 优化模拟输入的采集时间与电路复杂度

### 2.2.1 价值主张

控制系统需要与各种反馈和监测源接口相连接。这些系统中的信号源驱动容性输入电路的能力各不相同，这些电路与模数转换器 (ADC) 的采样保持 (S+H) 输入电路中的典型电路类似。C2000 器件上的 ADC 允许在宽范围内为每个输入通道分别配置 S+H 的采集时间。这使得系统可以同时与混合的高性能和低成本信号源接口相连接。

### 2.2.2 深入介绍

ADC 的输入通常被建模为开关电容电路，其中 ADC 中的保持电容  $C_h$  需要在采集期间从未知电压充电到接近输入电压的值。图 2-1 显示了一个来自 TMS320F2837xD 器件的示例。

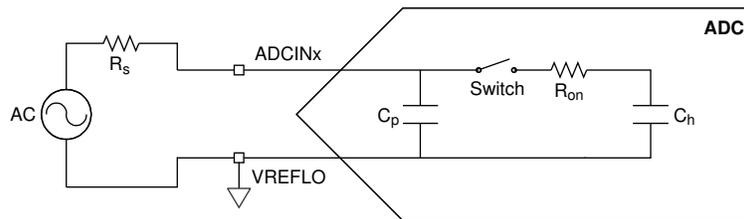


图 2-1. 单端输入模型

对  $C_h$  充电所需的采集时间取决于无源器件的外部阻抗、任何缓冲器或传感器的带宽、内部 ADC 输入寄生电容以及 ADC 的分辨率。

系统设计人员可以在外部电路成本及复杂性与建立速度之间进行各种权衡，例如：

- 添加/升级用于驱动 ADC 输入的运算放大器缓冲器**：通过更好地将电荷转移到 ADC 内的采样保持电容来缩短采集时间
- 增加 ADC 输入端上的电阻和/或电容的数量**：通过增加额外的低通滤波来降低噪声，但这会以增加采集时间为代价
- 容许较低的精度**：或者，使用较小的采集窗口来减少采样时间，但这会降低精度/分辨率。

考虑到以上所有可能的折衷，很难选择一个适合系统中所有模拟输入的单个采集时间。C2000 ADC 允许为每个通道选择一个单独的采集窗口，从而为系统设计人员提供了极大的灵活性，使他们可以根据自己的需要，在速率、信号调理电路成本及精度之间进行权衡。

采集窗口 ( 由 ADC SOC 配置寄存器的 ACQPS 字段控制 ) 还可以在较大范围的值内以较小的阶跃幅度进行配置, 如表 2-2 所示。

**表 2-2. 采集时间配置范围 ( 每个通道 )**

C2000 MCU	器件 SYSCLK	最小 S+H 时间	最大 S+H 时间	S+H 时间配置分辨率
TMS320F28004x 和 TMS320F28002x	100MHz	80ns	5.1μs	10.00ns
TMS320F2807x、 TMS320F28003x 、TMS320F280013x、 TMS320F280015x	120MHz	75ns	4.3μs	8.33ns
TMS320F2837xD 和 TMS320F2837xS	200MHz	75ns	2.6μs	5.00ns

ADC 输入的建模方法有多种, 德州仪器 (TI) 可提供免费工具来帮助设计 ADC 输入驱动器电路, 并可提供有关正确选择前端组件的指导视频。

### 2.2.3 器件列表

- [TMS320F2838xD/S](#)
- [TMS320F2837xD/S](#)
- [TMS320F2807x](#)
- [TMS320F28004x](#)
- [TMS320F28003x](#)
- [TMS320F28002x](#)
- [TMS320F280013x](#)
- [TMS320F280015x](#)

### 2.2.4 硬件平台和软件示例

- [TMDSCNCDF28388D](#)
- [TMDSCNCDF28379D](#)
- [TMDSCNCDF280049C](#)
- [TMDSCNCD280039C](#)
- [TMDSCNCD280025](#)
- [TMDSCNCD2800137](#)
- [TMDSCNCD2800157](#)
- [F2838xD ADC 软件示例](#)
- [F2837xD ADC 软件示例](#)
- [F28004x ADC 软件示例](#)
- [F28002x ADC 软件示例](#)
- [F280013x ADC 软件示例](#)
- [F280015x ADC 软件示例](#)

### 2.2.5 文档

- [C2000 ADC 的充电共享驱动电路](#)
- [C2000 MCU 的 ADC 输入电路评估](#)
- [SAR ADC 输入驱动器设计](#)
- [TI 精密实验室 - ADC : SAR ADC 前端组件选择简介](#)
- [TMS320F2838xD 实时微控制器技术参考手册](#) ( 更多信息, 请参阅 [选择采集窗口持续时间](#) 部分 )
- [TMS320F2837xD 实时微控制器技术参考手册](#) ( 更多信息, 请参阅 [选择采集窗口持续时间](#) 部分 )
- [TMS320F28004x 实时微控制器技术参考手册](#) ( 更多信息, 请参阅 [选择采集窗口持续时间](#) 部分 )
- [TMS320F28003x 实时微控制器技术参考手册](#) ( 更多信息, 请参阅 [选择采集窗口持续时间](#) 部分 )
- [TMS320F28002x 实时微控制器技术参考手册](#) ( 更多信息, 请参阅 [选择采集窗口持续时间](#) 部分 )
- [TMS320F280013x 实时微控制器技术参考手册](#) ( 更多信息, 请参阅 [选择采集窗口持续时间](#) 部分 )

- [TMS320F280015x 实时微控制器技术参考手册](#) (更多信息, 请参阅[选择采集窗口持续时间](#)部分)

## 2.3 使用单引脚基准对双阈值进行基于硬件的监视

### 2.3.1 价值主张

通过为每个比较器子系统 (CMPSS) 模块提供对两个嵌入式电压比较器的单个引脚访问, C2000 MCU 有助于降低使用多个比较器监视反馈信号的成本和复杂性。

### 2.3.2 深入介绍

控制系统通常使用电压比较器来监视阈值交叉事件的反馈信号。这些交叉事件可以表示从额定条件到临界条件的各种状态。有时可能会由多个比较器监视单个反馈信号, 以触发每个状态的自定义响应。

考虑使用两个阈值电平定义驱动开关行为的简单迟滞控制器 (图 2-2) :

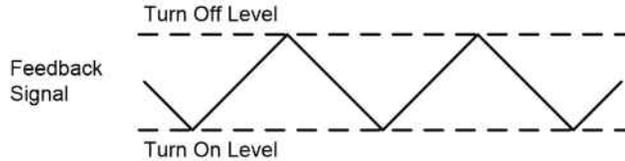


图 2-2. 迟滞控制器中的阈值电平

1. 低电平“下限”阈值定义了应启动的时间,
2. 高水平“上限”阈值定义了应关断的时间

可以使用一个 CMPSS 引脚实现针对该迟滞控制器的基于比较器的监视和触发方案, 如图 2-3 所示。同样, 可以使用另一个 CMPSS 引脚检测过压和欠压故障情况, 以进行系统保护。

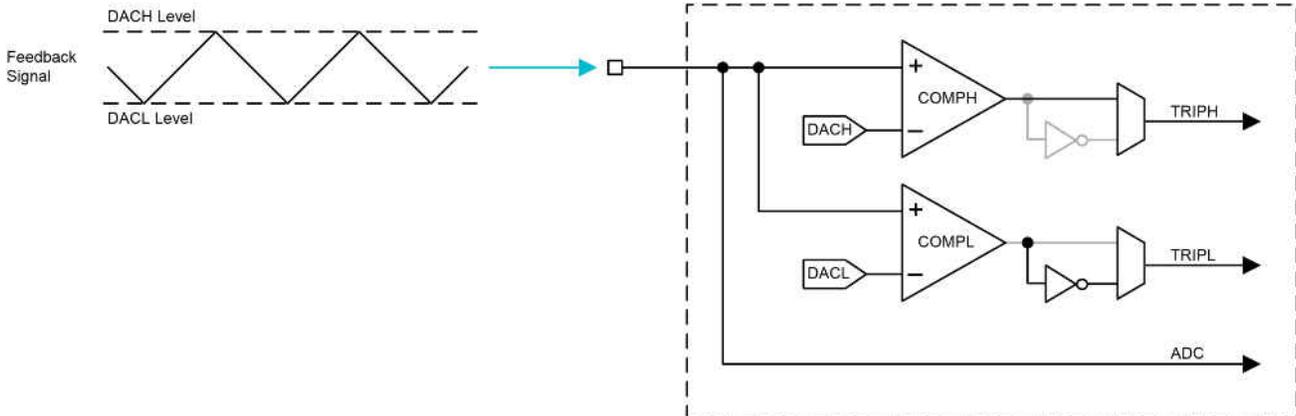


图 2-3. CMPSS 方框图

每个 CMPSS 比较器均具有各自的电压参考值 DAC、输出调节逻辑以及用于独立操作的独特跳闸信号。此外, 每个 CMPSS 引脚还分配给一个 ADC 通道, 该通道可与比较器监视并行地对引脚电压进行采样。这些 ADC 采样可用于影响复杂的系统行为, 并用作电压监控的冗余形式。

与其他具有专用引脚功能的嵌入式解决方案相比, CMPSS 引脚的多功能性在资源优化方面具有显著优势。与需要局部资源 (如电源和基准电压) 的分立解决方案相比, 优化优势甚至更大。通过利用每个 C2000 MCU 引脚提供的全部资源, 可以降低系统成本和复杂性。

### 2.3.3 器件列表

- [TMS320F2838xD/S](#)
- [TMS320F2837xD/S](#)
- [TMS320F2807x](#)
- [TMS320F28004x](#)
- [TMS320F28003x](#)

- [TMS320F28002x](#)
- [TMS320F280013x](#)
- [TMS320F280015x](#)

#### 2.3.4 硬件平台和软件示例

- [TIDM-DC-DC-BUCK](#)
- [TIDM-DC-DC-BUCK 示例软件](#)
- [TIDM-02002](#)
- [TIDM-02002 示例软件](#)
- [TIDM-1022](#)
- [TIDM-1022 软件示例](#)
- [TMDXIDDK379D](#)
- [TMDXIDDK379D 软件示例](#)

#### 2.3.5 文档

- [带连接管理器的 TMS320F2838x 实时微控制器数据表](#) (更多信息, 请参阅[比较器子系统 \(CMPSS\)](#) 一章)
- [TMS320F2837xD 双核微控制器数据表](#) (更多信息, 请参阅[比较器子系统 \(CMPSS\)](#) 部分)
- [TMS320F28004x 微控制器数据表](#) (更多信息, 请参阅[比较器子系统 \(CMPSS\)](#) 部分)
- [TMS320F28003x 微控制器数据表](#) (更多信息, 请参阅[比较器子系统 \(CMPSS\)](#) 部分)
- [TMS320F28002x 实时微控制器数据表](#) (更多信息, 请参阅[比较器子系统 \(CMPSS\)](#) 部分)
- [TMS320F280013x 实时微控制器数据表](#) (更多信息, 请参阅[比较器子系统 \(CMPSS\)](#) 部分)
- [TMS320F280015x 实时微控制器数据表](#) (更多信息, 请参阅[比较器子系统 \(CMPSS\)](#) 部分)
- [TMS320F2838xD 实时微控制器技术参考手册](#) (更多信息, 请参阅[比较器子系统 \(CMPSS\)](#) 一章)
- [TMS320F2837xD 实时微控制器技术参考手册](#) (更多信息, 请参阅[比较器子系统 \(CMPSS\)](#) 一章)
- [TMS320F28004x 实时微控制器技术参考手册](#) (更多信息, 请参阅[比较器子系统 \(CMPSS\)](#) 一章)
- [TMS320F28003x 实时微控制器技术参考手册](#) (更多信息, 请参阅[比较器子系统 \(CMPSS\)](#) 一章)
- [TMS320F28002x 实时微控制器技术参考手册](#) (更多信息, 请参阅[比较器子系统 \(CMPSS\)](#) 一章)
- [TMS320F280013x 实时微控制器技术参考手册](#) (请参阅[比较器子系统 \(CMPSS\)](#) 一章)
- [TMS320F280015x 实时微控制器技术参考手册](#) (请参阅[比较器子系统 \(CMPSS\)](#) 一章)

## 2.4 解决 ADC 采样期间的容差和老化效应问题

### 2.4.1 价值主张

ADC 结果在应用于特定系统的控制算法之前，通常需要进行数学处理。这通常是通过 CPU 的一些额外操作来完成的，从而增加了系统的延迟，并为此类操作增加了 CPU 的负载。C2000 MCU 能够在硬件中对此进行校正，而不会占用 CPU 开销且不会影响 ADC 采样率。

### 2.4.2 深入介绍

在使用 ADC 结果进行控制计算之前，通常需要去除由外部因素（例如器件容差或布局差异）引入的任何已知偏移量（表 2-3）。尽管上述问题的各个方面可以通过 PCB 布局或选择更高容差/稳定的电阻器来部分解决，但始终存在与理想情况的偏差。

表 2-3. 随时间变化的典型电阻容差和系统影响

生命周期阶段	总容差	相关的 12 位错误
购买	±0.05%	±2LSB
组装后	±0.5%	±20LSB
贮存后/潮湿	±0.75%	±30LSB
温度系数和 EOL	±1.00%	± 40LSB

C2000 MCU 实现了一个集成的硬件模块，可以校正与 ADC 转换过程相关的最多 10 位有符号值，从而节省了宝贵的系统循环时间。由于保持了 ADC 采样率，并且没有使用 CPU 周期来执行校正，系统的周期值实际上增加了一倍。此外还内置了饱和度。

有关偏差校正的实现及 ADC 后处理块中包含的其他模块，请参阅图 2-4。

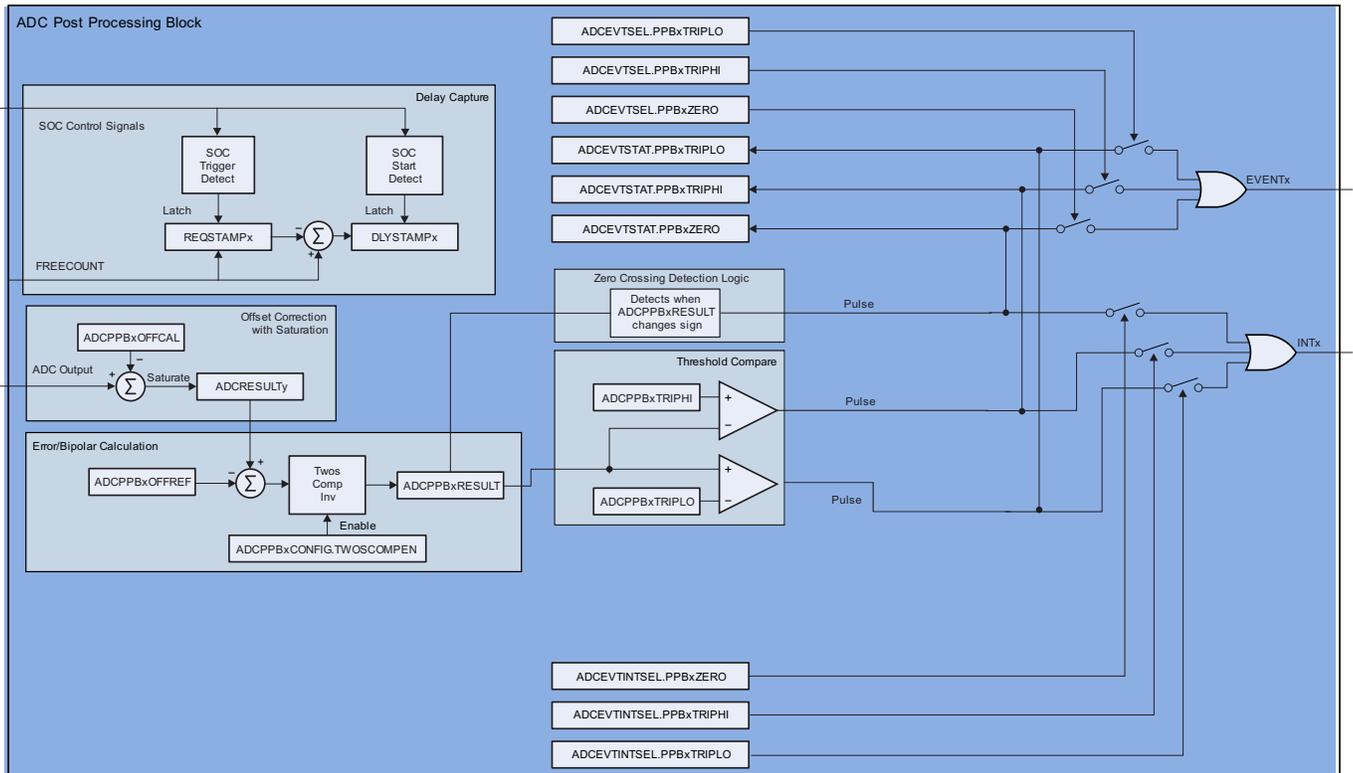


图 2-4. TMS320F2837xD 上的 ADC 后处理块

### 2.4.3 器件列表

- TMS320F2838xD/S
- TMS320F2837xD/S

- [TMS320F2807x](#)
- [TMS320F28004x](#)
- [TMS320F28003x](#)
- [TMS320F28002x](#)
- [TMS320F280013x](#)
- [TMS320F280015x](#)

#### 2.4.4 硬件平台和软件示例

- [TMDSCNCDF28388D](#)
- [TMDSCNCDF28379D](#)
- [TMDSCNCDF280049C](#)
- [TMDSCNCD280039C](#)
- [TMDSCNCD280025](#)
- [TMDSCNCD2800137](#)
- [TMDSCNCD2800157](#)
- [F2838xD ADC PPB 软件示例](#)
- [F2837xD ADC PPB 软件示例](#)
- [F28004x ADC PPB 软件示例](#)
- [F28003x ADC PPB 软件示例](#)
- [F28002x ADC PPB 软件示例](#)
- [F280013x ADC PPB 软件示例](#)
- [F280015x ADC PPB 软件示例](#)

#### 2.4.5 文档

- [TMS320F2838xD 实时微控制器技术参考手册](#) - 请参阅 [模数转换器 \(ADC\)](#) 一章的 [后处理块](#) 部分
- [TMS320F2837xD 实时微控制器技术参考手册](#) - 请参阅 [模数转换器 \(ADC\)](#) 一章的 [后处理块](#) 部分
- [TMS320F28004x 实时微控制器技术参考手册](#) - 请参阅 [模数转换器 \(ADC\)](#) 一章的 [后处理块](#) 部分
- [TMS320F28003x 实时微控制器技术参考手册](#) - 请参阅 [模数转换器 \(ADC\)](#) 一章的 [后处理块](#) 部分
- [TMS320F28002x 实时微控制器技术参考手册](#) - 请参阅 [模数转换器 \(ADC\)](#) 一章的 [后处理块](#) 部分
- [TMS320F280013x 实时微控制器技术参考手册](#) - 请参阅 [模数转换器 \(ADC\)](#) 一章的 [后处理块](#) 部分
- [TMS320F280015x 实时微控制器技术参考手册](#) - 请参阅 [模数转换器 \(ADC\)](#) 一章的 [后处理块](#) 部分

## 2.5 使用 C2000 可配置逻辑块实现旋转传感解决方案

### 2.5.1 价值主张

如“片上硬件定制 (节 4.3)”一文中所述，C2000 可配置逻辑块 (CLB) 技术使系统设计人员能够在 C2000 器件内实现新逻辑并消除外部电路。实时控制系统的一个特定示例是与数字旋转编码器的集成接口；使用 CLB，开发人员可以自由地在 C2000 MCU 内部集成业界通用的编码器通信协议或特定于应用的定制协议，而无需额外的外部电路。

## 2.5.2 深入介绍

数字旋转编码器是一种将轴的位置转换为数字信号的器件。编码器主要有两种类型：

- 绝对编码器 (图 2-5)：绝对编码器的输出指示由特定协议所定义发送回主机的消息中的当前角度位置。
- 增量编码器：增量编码器的输出以一系列调制脉冲的形式提供信息，通常由系统主机将其进一步处理成速度、距离和位置等信息。

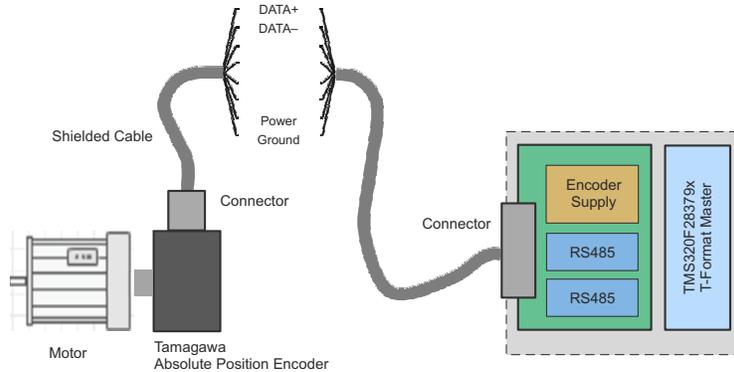


图 2-5. 具有 T-Format 绝对位置编码器接口的工业伺服驱动器

C2000 CLB 技术支持集成解决方案，能够与最流行的数字旋转位置编码器接口相连接，从而无需使用外部现场可编程门阵列 (FPGA) 或专用集成电路 (ASIC)。PositionManager BoosterPack 插件模块是一个灵活的低电压平台，旨在评估各种编码器接口，并设计为与多个 C2000 LaunchPad 开发套件 (如 LAUNCHXL-F28379D、LAUNCHXL-F280049C 或 LAUNCHXL-F280025C) 配合使用。

以下编码器示例来自电机控制 SDK 软件包，可用于评估。计划在未来对其进行更新以添加 BiSS-C 和 EnDAT22 协议的示例。

- T-Format 绝对编码器接口：

Tamagawa T-Format 协议是一种流行的数字双向绝对编码器接口。TIDM-1011 随附的易用型库和示例软件演示了 Tamagawa 的 T-Format 标准。在此示例中，我们使用片上资源 (例如 CLB、SPI 和 GPIO) 将 T-Format 绝对编码器接口集成到了 C2000 中，如图 2-5 所示。此 TI 设计包括以下功能：

- T-Format 编码器接口库和 CRC 库用于实现协议命令
  - 演示项目 (具有完整的源代码访问权限) 用于执行 T-Format 命令
  - 由集成在 Code Composer Studio 内的 C2000 CLB 工具提供完全支持
- 高带宽电流环路应用中的 T-Format 评估：

使用最新的 C2000 MCU (例如 TMS320F2837x 和 TMS320F28004x)，可以实现快速电流环路 (FCL) 算法，该算法使用与传统的磁场定向控制 (FOC) 方法相同的外部硬件来提供高电流环路带宽。TI 已在这些 MCU 上开发了 FCL 算法，并在 DesignDRIVE IDDK 平台上实现了该算法。T-Format 编码器接口已集成到 C2000 器件上 FCL 算法的评估实现中。

使用快速电流环对 PMSM 进行快速响应控制研究了电流环路的实时频率响应分析，并验证了 TIDM-1011 中实现的 T-format 编码器接口的接口逻辑。可以使用 QEP 编码器或 T-format 编码器关闭此示例中的位置环，并且在两种情况下都可以实现 FCL。

- 脉冲序列输出 (PTO) QepDiv 和 PulseGen :

增量旋转编码器输出脉冲序列，以指示被监测的轴已实现移动。系统主机通常会处理此脉冲序列，以确定速度、距离和位置等信息。在 QepDiv 实现中，位置信息从编码器发送到 C2000 MCU 上的增强型正交编码器脉冲 (eQEP) 模块。在 PulseGen 情况下，将生成自定义脉冲流以满足系统需求。PTO-QepDiv 示例演示了如何使用 CLB 从这些 eQEP 输入生成分离的脉冲流，如 图 2-6 所示。分离的脉冲流继而将被发送到系统中的另一个器件。[C2000 Position Manager PTO API 参考指南](#) 介绍了 QepDiv 和 PulseGen 示例。

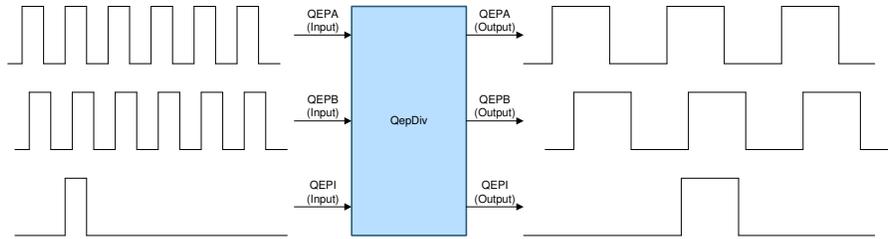


图 2-6. QepDiv 输入和输出图

### 2.5.3 器件列表

- [TMS320F2838xD/S](#)
- [TMS320F2837xD/S](#)
- [TMS320F2807x](#)
- [TMS320F28004x](#)
- [TMS320F28003x](#)
- [TMS320F28002x](#)

### 2.5.4 硬件平台和软件示例

- [基于 C2000™ MCU 的 Tamagawa T-Format 绝对编码器主接口参考设计](#)
- [C2000 Position Manager PTO API 参考指南](#)
- [适用于 C2000 MCU 的 MotorControl 软件开发套件 \(SDK\)](#)
- [Position Manager BoosterPack \(BOOSTXL-POSMGR\)](#)
- [适用于工业电机控制的 C2000™ DesignDRIVE 开发套件 \(TMDXIDDK379D\)](#)

### 2.5.5 文档

- 培训：[C2000 可配置逻辑块 \(CLB\) 工具如何在我的设计中集成自定义逻辑](#)
- [CLB 工具用户指南](#)
- [使用 C2000 可配置逻辑块 \(CLB\) 进行设计](#)
- [《如何将自定义逻辑从 FPGA/CPLD 迁移到 C2000 微控制器》](#)
- [快速电流环路库](#)

## 2.6 跨隔离边界的智能传感

### 2.6.1 价值主张

高压控制系统通常采用  $\Sigma$ - $\Delta$  ADC，以便轻松地将高压域的模拟信息传送到 MCU 所在的较低电压域。在使用数据流之前，必须先通过滤波器和解调器对其进行处理。此逻辑存在于 C2000 MCU 上，称为  $\Sigma$ - $\Delta$  滤波器模块 (SDFM)。位于 C2000 MCU SDFM 内部的高低电平比较器可以在无需 CPU 干预的情况下驱动 PWM，从而节省了宝贵的时间来更好地控制系统。

### 2.6.2 深入介绍

$\Sigma$ - $\Delta$  型 ADC 是通过设计一个过采样架构来实现的。ADC 本身是一个 single-bit 设计，对相关的信号进行过采样以产生更高阶的输出。这种转换器的调制产生二进制输出，因此产生串行数据流。然后，该串行数据流将被发送到 C2000 MCU 上的滤波器/解调器，为采样信号重构更高阶数字表示 (图 2-7)。

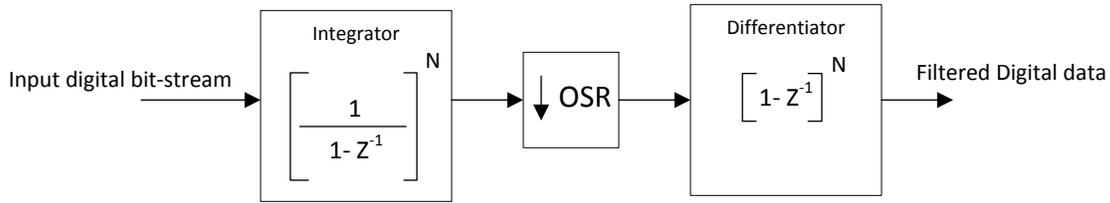


图 2-7. C2000 SDFM 内部的滤波器和解调器

此时，将向 MCU 生成一个中断，通知其他域有新数据要处理和操作。

C2000 MCU 上的 SDFM 模块的独特之处在于，不仅可以在滤波后读取转换后的数据，而且可以基于该数据自动切换 PWM。每个 SDFM 模块包含四个通道。每个通道内部有两个滤波器：一个用于产生 SDFM 数据的主滤波器，一个包含上限和下限比较器的次级滤波器 (图 2-8)。这使得系统无需等待 CPU 干预即可控制 PWM 信号，从而既降低了延迟，又降低了总体 CPU 利用率。

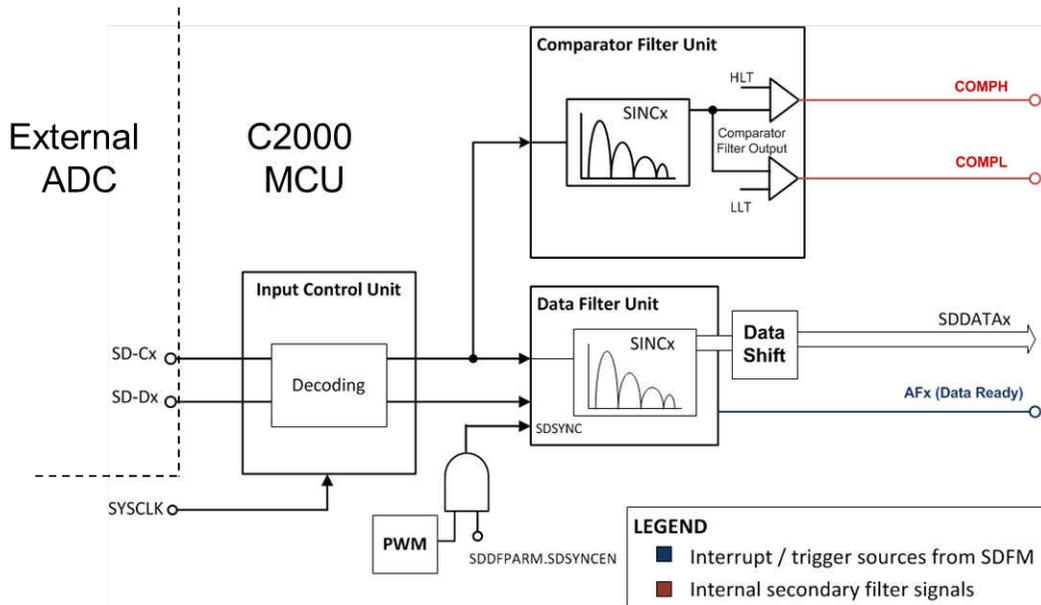


图 2-8. TMS320F2837xD MCU 上带有主滤波器和次滤波器块的 SDFM 模块

### 2.6.3 器件列表

- [TMS320F2838xD/S](#)
- [TMS320F2837xD/S](#)
- [TMS320F2807x](#)
- [TMS320F28004x](#)
- [TMS320F28003x](#)

### 2.6.4 硬件平台和软件示例

- [TMDSCNCD28388D controlCARD 评估模块](#)
- [LAUNCHXL-F28379D LaunchPad](#)
- [LAUNCHXL-F280049C LaunchPad](#)
- [TMDSCNCD280039C controlCARD 评估模块](#)
- [F2838xD/S 的 SDFM\\_filter\\_sync\\_cpuread 软件示例](#)
- [F2838xD/S 的 SDFM\\_pwm\\_sync\\_cpuread 软件示例](#)

## 2.6.5 文档

- [TMS320F2838x 实时微控制器技术参考手册](#) (更多信息, 请参阅 *SDFM* 一章)
- [TMS320F2837xD 实时微控制器技术参考手册](#) (更多信息, 请参阅 *SDFM* 一章)
- [TMS320F28004x 实时微控制器技术参考手册](#) (更多信息, 请参阅 *SDFM* 一章)
- [TMS320F28003x 实时微控制器技术参考手册](#) (更多信息, 请参阅 *SDFM* 一章)
- [实时控制器获得新的连接功能](#)

## 2.7 在高带宽控制拓扑中启用内部周期更新

### 2.7.1 价值主张

设计实时控制系统时, 最重要的考虑因素之一是系统可以达到的最小采样输出延迟 (图 2-9)。从提供对实时系统的稳定控制的基础到实现更高的系统效率, 每个 C2000 实时微控制器内部的片上 ADC 均经过设计, 以通过各种方式尽可能缩短系统的有效延时时间。除了分立式 ADC 的基础, 并与 C2000 片上 ADC 进行比较, 由于数十年来参与实时控制, 其功能具有多样性。

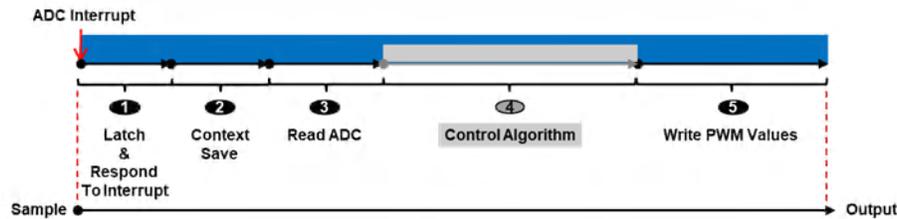


图 2-9. 实时信号链

### 2.7.2 深入介绍

在为系统选择分立式 ADC 时, 要考虑许多因素: 原始转换器速度、直流误差项 (例如增益和偏移量) 和交流误差项 (例如 SNR、THD) (如节 2.1 所示)。除了这些参数之外, 还包括系统集成, 相对于控制环路的转换速度以及可以建立或破坏实时控制设计的取样能力等方面。

**系统集成:** 集成 ADC 的一个优势是缺少外部通信总线, 这不仅增加了通信通道的延时时间, 而且还增加了总线布线的噪声考虑, 从而增加了系统的复杂性。ADC 转换完成后, 尽可能缩短系统控制方案的延迟至关重要。所有 CPU 内核、CLA 内核以及 DMA 均可访问 ADC 结果。

由于在通知 CPU ADC 转换已完成时引入了延迟, 因此引入了另一个常见延迟。C2000 ADC 会实施可选的早期中断, 以利用 ADC 从采样开始到模数转换完成的间隔时间。由于集成了 ADC, 因此 C2000 MCU 可以同步至模拟采样阶段的末尾或整个转换完成。这使 C28x CPU 或 CLA 可以继续与 ADC 转换并行执行其他任务或计算, 并在可用时立即使用 ADC 转换。由于 CPU 使用的转换值在时间上更接近实际信号状态, 因此不仅有效缩短了浪费周期, 而且系统控制更加稳定。

**转换速度：**评估任何分立或集成 ADC 的共同点是其转换速度。ADC 将采样的模拟信号转换为数字表示形式所需的时间。这在高带宽系统中尤其重要，因为系统周期内更新与等待控制环路的下一个完整周期相比更好（图 2-10）。（快速电流环路示例）中显示了这种情况。采样完成后，C2000 MCU 上的 ADC 能够在短至 260ns 的时间内生成转换。这为 C28x 或 CLA 内核留出了足够的时间来在下一个控制环路周期之前计算新的 PWM 频率，从而允许进行周期内更新。ePWM 模块在这里也发挥了作用，它允许通过写入有效寄存器和影子寄存器来更新 PWM，这将使更新一直持续到下一个完整周期。

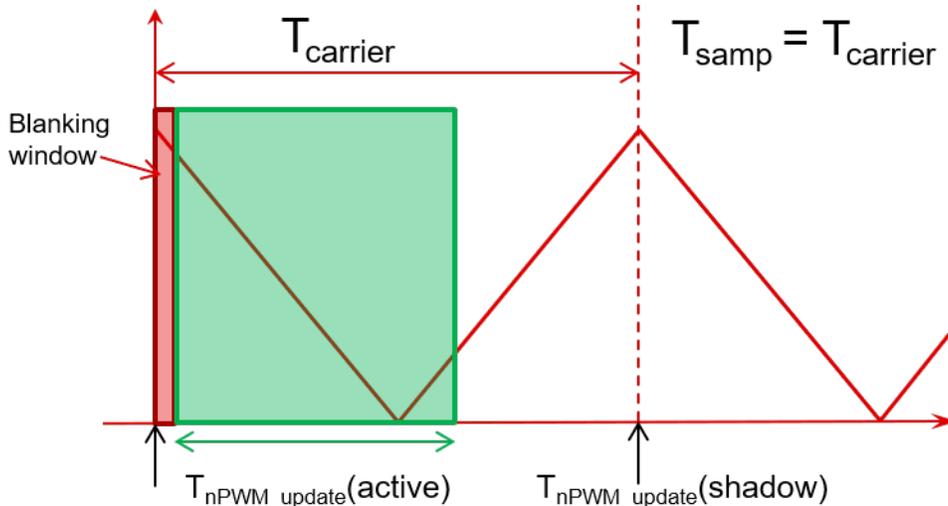


图 2-10. PWM 周期内更新

**同步采样：**本指南中引用的每个 C2000 器件至少具有 2 个 ADC（如果没有更多的 ADC）。查看常见的实时控制拓扑，可以同时采样电机电流。对于使用旋转变压器进行位置感测的系统，这将允许在同一时间点对正弦和余弦反馈进行采样。表 2-4 显示了未在同一时间点对此类输入进行采样时对系统的潜在影响

表 2-4. 采样延迟导致的潜在相位错误

控制环路频率	500ns 采样延迟时的相位误差	1µs 采样延迟时的相位误差
10kHz	1.8 度	3.6 度
20kHz	3.6 度	7.2 度
50kHz	8.9 度	17.8 度

### 2.7.3 器件列表

- TMS320F2838xD/S
- TMS320F2837xD/S
- TMS320F2807x
- TMS320F28004x
- TMS320F28003x
- TMS320F28002x
- TMS320F280013x
- TMS320F280015x

### 2.7.4 硬件平台和软件示例

- TIDM-02007 在单个 MCU 上使用快速电流环路 (FCL) 和 SFRA 的双轴电机驱动器参考设计
- BOOSTXL-3PHGANINV 具有基于采样电阻的内嵌式电机相电流采样的 48V 三相逆变器评估模块
- C2000Ware 电机控制 SDK 中的 BOOSTXL-3PHGANINV 示例
- TMDXIDDK379D，适用于工业电机控制的 C2000 DesignDRIVE 开发套件

### 2.7.5 文档

- 伺服中快速电流环路 (FCL) 的性能分析

- [基于快速电流环路的 PMSM 快速响应控制](#)

## 2.8 无需信号调节即可准确监控实时控制系统事件

### 2.8.1 价值主张

如节 4.4 所述，快速检测过冲和下冲对于实时控制系统的运行至关重要。然而在现实中，所关注的瞬态通常具有耦合噪声，这可能会引起误跳闸和关断，从而导致系统功能出现可靠性问题。

通常用于解决此问题的一种方法是使用外部无源 RC 滤波器从目标信号中去除高频噪声。该方法具有 BOM 成本、PCB 空间和减慢目标信号的缺点。C2000 MCU 上的 CMPSS 模块具有集成在器件中的可配置滤波器，无需外部 RC 滤波器即可滤除噪声事件。

### 2.8.2 深入介绍

CMPSS 滤波器遵循多数票决原则。它从比较器捕获高/低跳闸事件的窗口，并根据阈值设置情况将输出设置为高或低。窗口捕获的跳闸事件数量和用于确定输出的阈值设置均可配置。由于具有可配置性，应用程序可在抑制多少噪声和延迟多少最终行程之间进行权衡。图 2-11 简要说明了处于工作状态的滤波器。更多详细信息，请参阅 TRM 中的“CMPSS”一章。

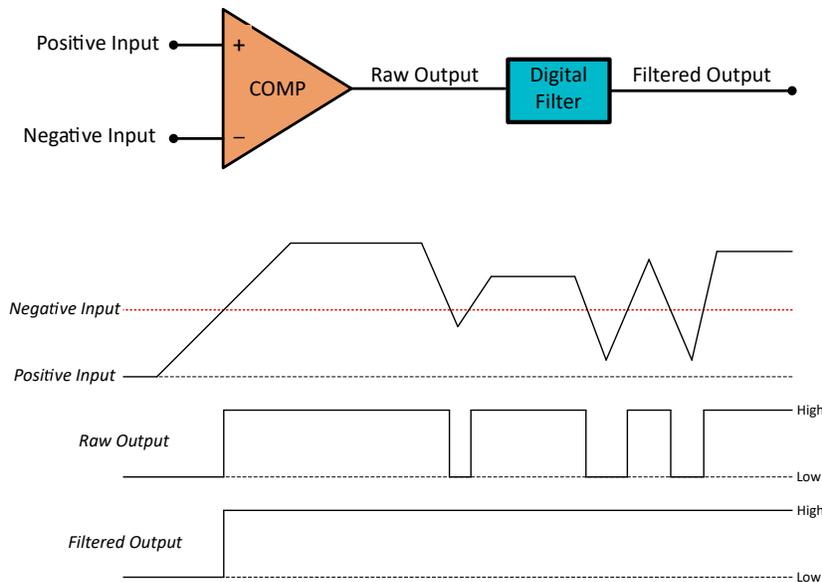


图 2-11. CMPSS 原始输出与滤波输出

### 2.8.3 器件列表

- [TMS320F2838xD/S](#)
- [TMS320F2837xD/S](#)
- [TMS320F2807x](#)
- [TMS320F28004x](#)
- [TMS320F28003x](#)
- [TMS320F28002x](#)
- [TMS320F280013x](#)
- [TMS320F280015x](#)

### 2.8.4 硬件平台和软件示例

- [TMDSCNCF28388D](#)
- [TMDSCNCF28379D](#)
- [TMDSCNCF280049C](#)
- [TMDSCNCF280039C](#)
- [TMDSCNCF280025](#)

- [TMDSCNCD2800137](#)
- [TMDSCNCD2800157](#)
- [F2838xD CMPSS 数字滤波器软件示例](#)
- [F2837xD CMPSS 数字滤波器软件示例](#)
- [F28004x CMPSS 数字滤波器软件示例](#)
- [F28003x CMPSS 数字滤波器软件示例](#)
- [F28002x CMPSS 数字滤波器软件示例](#)
- [F280013x CMPSS 数字滤波器软件示例](#)
- [F280015x CMPSS 数字滤波器软件示例](#)

### 2.8.5 文档

- [TMS320F2838x 实时微控制器技术参考手册](#) (更多信息, 请参阅 [CMPSS](#) 一章)
- [TMS320F2837xD 实时微控制器技术参考手册](#) (更多信息, 请参阅 [CMPSS](#) 一章)
- [TMS320F28004x 实时微控制器技术参考手册](#) (详细信息, 请参阅 [CMPSS](#) 一章)
- [TMS320F28003x 实时微控制器技术参考手册](#) (更多信息, 请参阅 [CMPSS](#) 一章)
- [TMS320F28002x 实时微控制器技术参考手册](#) (详细信息, 请参阅 [CMPSS](#) 一章)
- [TMS320F280013x 实时微控制器技术参考手册](#) (更多信息, 请参阅 [CMPSS](#) 一章)
- [TMS320F280015x 实时微控制器技术参考手册](#) (更多信息, 请参阅 [CMPSS](#) 一章)

## 3 处理关键技术

### 3.1 三角函数加速器

#### 3.1.1 价值主张

三角函数在实时控制系统中有着广泛的应用, 无论是在电力应用还是电机控制领域。派克变换 (图 3-1)、空间矢量生成和旋转变压器角度便是其中一些依赖三角数学的示例。C2000 MCU 上的三角数学单元 (TMU) 支持一个扩展的指令集, 旨在进行基于 32 位浮点三角的计算。这将为三角函数带来更快的性能和更小的代码量。

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

图 3-1. 派克变换

#### 3.1.2 深入介绍

实时控制中许多常见的数学技术都依赖于三角函数的使用: 正弦、余弦和反正切无一不是其中的一些例子。TMU 在 C28x 内核中为这些函数及其逆函数添加了专用指令, 以取代标准 C 库调用。如图 3-2 所示, 使用基于 TMU 的指令与其对应的 Fast RTS 指令相比, 可以显著减少周期计数。

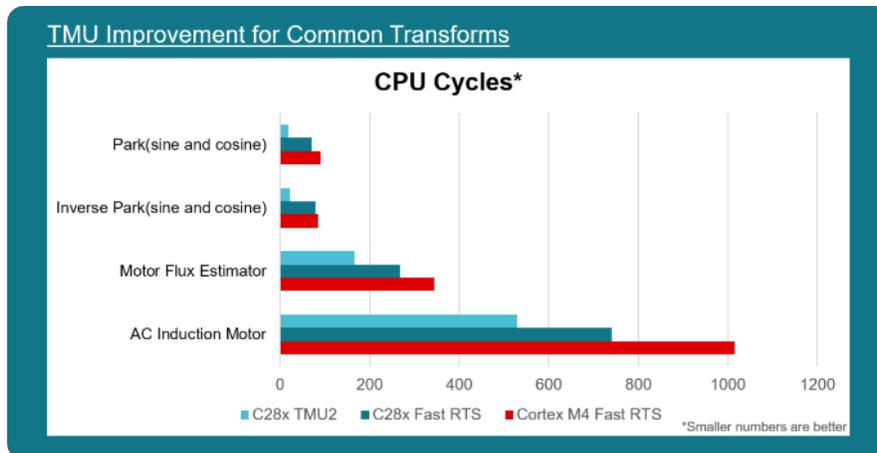


图 3-2. TMU 对通用变换的改进

平方根和浮点除法也受单一指令支持。这些函数通常与前面列出的三角函数结合使用。表 3-1 列出了支持的指令及其周期数的完整列表。这些指令由 C 编译器在具有 TMU 的器件上自动插入。

**表 3-1. TMU 支持的指令汇总**

运算	C 等效运算	C28x 管线周期
乘以 $2\pi$	$a = b * 2\pi$	2 个周期 + 正弦/余弦函数
除以 $2\pi$	$a = b / 2\pi$	2 个周期 + 正弦/余弦函数
除以	$a = b / c$	5 个周期
平方根	$a = \text{sqrt}(b)$	5 个周期
单位正弦	$a = \sin(b * 2\pi)$	4 个周期
单位余弦	$a = \cos(b * 2\pi)$	4 个周期
单位反正切	$a = \text{atan}(b) / 2\pi$	4 个周期
反正切 2 和象限运算	用于协助计算 ATANPU2 的运算	5 个周期

#### 备注

虽然下面列出的所有 C2000 MCU 都具有 TMU 模块，但是用于生成目标代码的 C 编译器必须选择正确的选项才能使用此硬件。通过 TMU 支持下拉菜单，以及使用“C2000 Compiler → Optimizations”下浮点模式的“relaxed”设置，可以在 C2000 编译器的“Processor Options”中对此进行控制。如果特定于 TMU 的函数优于全局设置，则可以在 C 源代码中以内联函数对其进行显式调用。

### 3.1.3 器件列表

- [TMS320F2838xD/S](#)
- [TMS320F2837xD/S](#)
- [TMS320F2807x](#)
- [TMS320F28004x](#)
- [TMS320F28003x](#)
- [TMS320F28002x](#)
- [TMS320F280013x](#)
- [TMS320F280015x](#)

### 3.1.4 硬件平台和软件示例

- TIDM-1007，交错式 CCM 图腾柱无桥功率因数校正 (PFC) 参考设计
- TIDM-HV-1PH-DCAC，具有电压源和并网模式的单相逆变器参考设计
- TMDXIDDK379D，适用于工业电机控制的 C2000 DesignDRIVE 开发套件
- TMDSHVMTRINSPIN，用于 F280049C 器件 lab7 和 lab8 且带有 InstaSPIN FOC 和 InstaSPIN MOTION 的高压电机控制套件

### 3.1.5 文档

- 增强 C2000™ MCU 系列器件的性能和功能
- 信号链基准 - 演示了 C2000™ MCU 经过优化的实时性能
- TMS320C28x 扩展指令集技术参考手册

## 3.2 快速板载整数除法

### 3.2.1 价值主张

与其他标准算术运算相比，精确的整数除法通常更复杂，并且 MCU 中的周期数高得多。另外，在控制算法中，需要执行更多的线性除法运算（例如欧几里得和模数），这往往会消耗额外的 CPU 开销。为解决此问题，FID 模块通过专用硬件支持这些特殊除法定义，从而实现控制算法，而无需与这些方法相关的典型额外开销。最后，FID 有最优的计算周期来支持各种大小和带符号的操作数。

### 3.2.2 深入介绍

根据编程语言和计算机科学文献，除法和模数运算有多种定义，这些定义中的每一个都提供了可以在应用上下文中使用并带来益处的不同数学特性。截断除法是在诸如 C 之类的许多编程语言中广泛使用的标准除法定义。在此定义中，余数始终具有分子的符号。截断除法的传递函数如图 3-3 中所示，从中可以看出该函数是非周期性的，因为零点周围有一个“平台”。

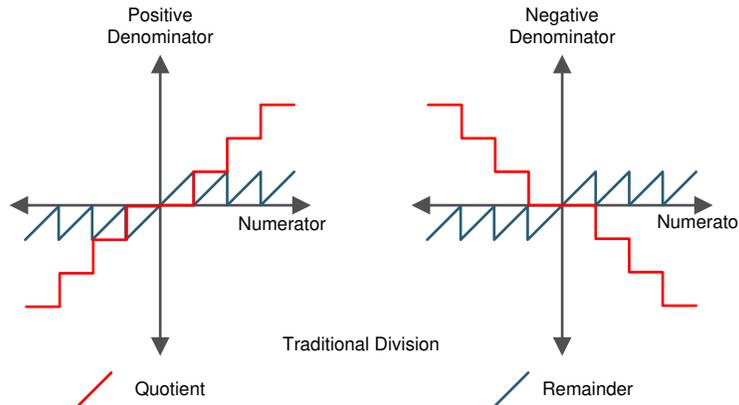


图 3-3. 截断除法函数

由于零点周围是非线性的，因此该函数并非控制应用中的优先选择。因此，除法和模数运算的非常规定义有时会因为更好的线性和周期性而成为优选方案。取整/模数（图 3-4）函数和欧几里德除法（图 3-5）函数的传递函数如下所示。在模除法函数中，余数将始终具有分母的符号，因此该函数在零点附近呈线性。在欧几里德除法中，余数将始终为正，因此除法函数在零点附近呈线性，而且模数函数呈周期性。

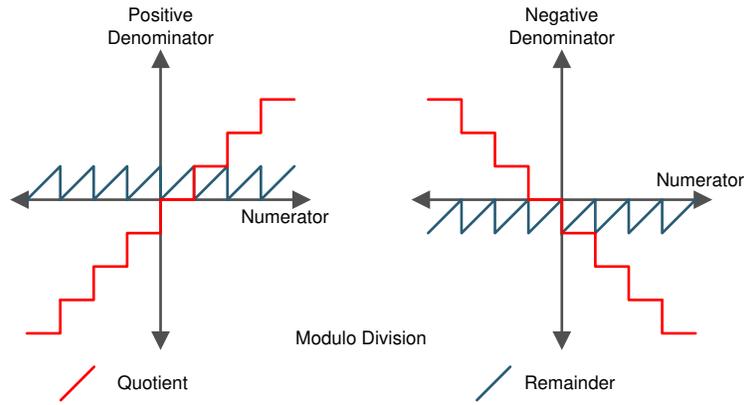


图 3-4. 取整除法函数

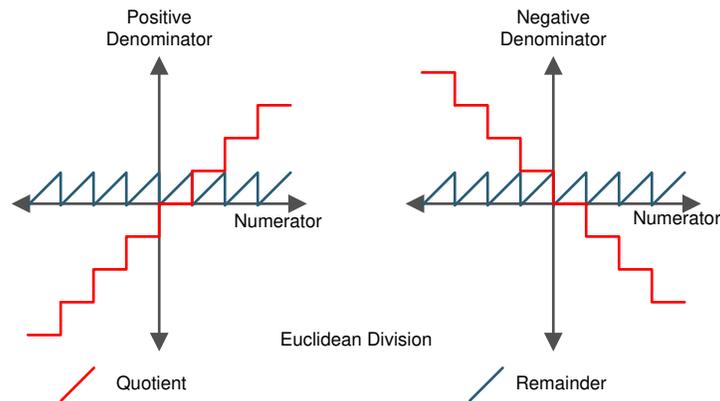


图 3-5. 欧几里德除法函数

C28x CPU 添加了专用指令，以使应用能够在硬件中有效地实现上述除法和模数定义。这些用于启用整数除法的新指令可中断，具有极低的延迟，并支持不同类型的运算（`ui32/ui32`、`i32/ui32`、`i64/i32`、`ui64/ui32`、`ui64/ui64`、`i64/i64` 等）。表 3-2 列出了不同类型的除法运算的周期数和使用快速整数除法单元实现的操作数的大小，并将其与没有 FID 模块的周期数进行了比较。从表中可以明显看出，对于不同类型的除法运算，FID 单元的性能提高了几倍，这有助于更大程度地减少控制回路计算的等待时间。

表 3-2. 使用和不使用 FID 模块的整数除法

除法运算	在 C28x 上使用不带 FASTINTDIV 硬件的 C 运算符 “/”	结合使用固有函数与 FASTINTDIV 硬件 + C28x	改善系数
i16/i16 传统	52	16	3.3
i16/i16 欧几里德	56	14	4.0
i16/i16 模数	56	14	4.0
u16/u16	56	14	4.0
i32/i32 传统	59	13	4.5
i32/i32 欧几里德	63	14	4.5
i32/i32 模数	63	14	4.5
i32/u32 传统	37	14	2.6
i32/u32 模数	41	14	2.9
u32/u32	37	12	3.1
i32/i16 传统	60	18	3.3
i32/i16 欧几里德	64	16	4.0
i32/i16 模数	64	16	4.0
u32/u16	38	13	2.9

**表 3-2. 使用和不使用 FID 模块的整数除法 (continued)**

除法运算	在 C28x 上使用不带 FASTINTDIV 硬件的 C 运算符 “/”	结合使用固有函数与 FASTINTDIV 硬件 + C28x	改善系数
i64/i64 传统 <sup>(1)</sup>	78-2631	42	1.9-62.6
i64/i64 欧几里德 <sup>(1)</sup>	82-2635	42	2.0-62.7
i64/i64 模数 <sup>(1)</sup>	82-2635	42	2.0-62.7
i64/u64 传统 <sup>(1)</sup>	54-2605	42	1.3-62.0
i64/u64 欧几里德 <sup>(1)</sup>	58-2609	42	1.4-62.1
i64/u6 模数 <sup>(1)</sup>	58-2609	42	1.4-62.1
u64/u64/ <sup>(1)</sup>	53-2548	42	1.3-60.7

(1) FASTINTDIV 硬件会实现 64 位整数除法，具有更优固定周期数，以实现快速确定性行为。没有此类硬件加速功能的 MCU 使用未针对除法优化的通用 CPU 指令来实现 64 位整数除法，或使用基于分子和分母值优化执行的算法技术。例如，如果分子和分母的值小于 32 位，则软件将执行 32 位除法。因此，周期数可能会显著变化，并且对于较大的分子和分母值，总周期远高于 FASTINTDIV 加速器可达到的周期

### 3.2.3 器件列表

- [TMS320F2838xD/S](#)
- [TMS320F28003x](#)
- [TMS320F28002x](#)

### 3.2.4 硬件平台和软件平台

- [F2838xD C2000Ware 中的快速整数除法示例](#)
- [F28002x C2000Ware 中的快速整数除法示例](#)
- [TMDSCNCD28388D 控制卡](#)
- [TMDSCNCD280039C 控制卡](#)
- [LAUNCHXL-F280025C LaunchPad](#)

### 3.2.5 文档

- [快速整数除法 - C2000 产品系列特性](#)
- [TMS320C28x 优化 C/C++ 编译器用户指南](#)
- [TMS320C28x 汇编语言工具用户指南](#)

## 3.3 对双精度浮点运算的硬件支持

### 3.3.1 价值主张

FPU64 模块可通过提供对 IEEE-754 单精度和双精度浮点运算的原生硬件支持来扩展 C28x CPU 的计算能力。这使需要在应用中使用双精度浮点的客户能够在不看到 CPU 周期负载显著增加的情况下实现它们。

### 3.3.2 深入介绍

当 32 位精度（例如单精度浮点）不足以用于应用时，FPU64 模块就非常有用。在许多实时控制应用中都是这样，因此需要 64 位精度（双精度浮点数）。这通常需要付出一定代价，因为器件上的本机硬件不支持双精度浮点运算。当运行双精度浮点算法时，用户会看到 CPU 周期数显著增加。通过纳入硬件支持，可以避免这种增加。

表 3-3 比较了 FPU64 与 FPU (单精度浮点硬件) 上的双精度运算的性能。

表 3-3. FPU64 和 FPU32 之间的周期比较

浮点运算	FPU64 (cycles) -- fp_mode=relaxed	FPU (cycles) -- fp_mode=strict	FPU (cycles) -- fp_mode=relaxed FPU64 disabled	FPU (cycles) -- fp_mode=strict FPU64 disabled
32 位除法	8	234	8	234
64 位除法	27	27	2222	2222

上表中的结果 (经过配置并关闭优化, 从 RAM 运行代码) 表明, FPU64 上的双精度浮点除法的性能比 FPU32 上的单精度浮点除法的性能所需的成本更加昂贵。在单精度情况下, 当浮点模式 (fp\_mode) 设置为 “relaxed” (宽松) 时, 编译器会生成硬件指令以执行单精度除法, 但会略微降低精度。当浮点模式设为 “strict” 时, 编译器不会生成硬件指令, 而会调用 RTS 库来保持精确性, 但会以周期数为代价。

在双精度情况下, 浮点模式无关紧要, 因为 FPU64 实现是准确的。只要启用了 FPU64, 编译器就会生成 FPU64 支持的硬件指令以执行双精度除法。如果禁用了 FPU64, 则编译器不会生成 FPU64 硬件指令, 而会调用 RTS 库。即使在 “relaxed” (宽松) 模式下, 64 位浮点除法也无法利用 32 位浮点硬件, 这就是周期数仍然为 2222 的原因。

具有 FPU64 的器除了增加了八个用于双精度浮点运算的浮点结果扩展寄存器之外, 还使用与 FPU 相同的寄存器。除了支持 64 位双精度浮点指令外, FPU64 增强还支持所有现有的 FPU 单精度浮点指令。FPU64 64 位指令在 1 到 3 个流水线周期中运行, 有些指令还支持并行移动操作。

使用 FPU64 很简单。用户只需使用双精度浮点变量和运算编写 C 代码, 使用 TI C28x C/C++ 编译器 v18.9.0.STS (或更高版本) 编译即可, 并使用编译器开关 `--float_support = fpu64`。这将生成 C28x 本机双精度浮点指令。如果用户不熟悉 C28x CPU 上标准 C 数据类型的大小, 参阅表 3-4 会有所帮助。想要为 FPU64 编写手动优化的汇编的用户也可以轻松地完成。单精度汇编指令和双精度浮点汇编指令大致是一一对应的。有关完整的指令集和详细信息, 请参阅文档部分中引用的文档。下述小节列出了软件示例, 这些示例指向 DSP 和 Math 双精度浮点手动优化的汇编例程, 用户可以用来进行应用开发。

表 3-4. 数据类型大小 C28x 与 Arm

数据类型	C28x 位长度 (EABI)	Arm 位长度
char	16	8
short	16	16
int	16	32
long	32	32
long long	64	64
float	32	32
double	64 <sup>(1)</sup>	64
long double	64	64
指针	32	32

(1) C28x COFF 将 double (双精度) 视为 32 位

随着 MATLAB Embedded Coder 等自动代码生成工具的出现, 许多客户正在迁移到这些生成工具以自动生成代码。由于 MATLAB 默认使用双精度浮点, 因此可以更容易地将代码从仿真环境移植到嵌入式环境, 而不必因为精度降低而重新验证系统的运行性能。这是使用 FPU64 的另一个关键优势。

### 3.3.3 器件列表

- [TMS320F2838xD/S](#)

### 3.3.4 硬件平台和软件示例

- [TMDSCNCD28388D 控制卡评估模块](#)
- [C2000Ware 中 FPUFastRTS 库内的 64 位浮点示例](#)
- [C2000Ware 中 DSP FPU 库内的 64 位浮点示例](#)

### 3.3.5 文档

- [TMS320C28x 扩展指令集技术参考手册](#)

## 3.4 通过独立的处理单元增加控制环带宽

### 3.4.1 价值主张

在任何控制应用中，对系统进行采样（传感）、应用控制功能（处理）以及将激励应用于外部系统（驱动）所消耗的时间都是一个主要的关注点。创建 CLA 的目的是为了解决在增加系统总吞吐量的同时最大程度地减少时间这一需求。

### 3.4.2 深入介绍

下面列出的器件上的 CLA 是与 C28x 主内核完全并行的处理器。虽然 C28x 内核是一个更传统的处理器，可以执行指令和处理中断，而 CLA 是一个任务驱动的状态机。CLA 采用 32 位浮点架构。

由于控制系统的实际情况，在某些情况下，采样子系统有新数据要处理。利用 MCU 的其他功能对这些事件进行高级规划可能很难进行时间分配，这些将会将延迟引入受控制的系统以及 CPU 其他必须执行的其他功能。

作为任务驱动的状态机，CLA 始终在空闲状态下等待事件（例如 ADC 转换）以处理该数据并启动系统。此外，CLA 可以完全访问关键控制外设，因此可以完全实现独立于 C28x CPU 的控制系统（图 3-6）。

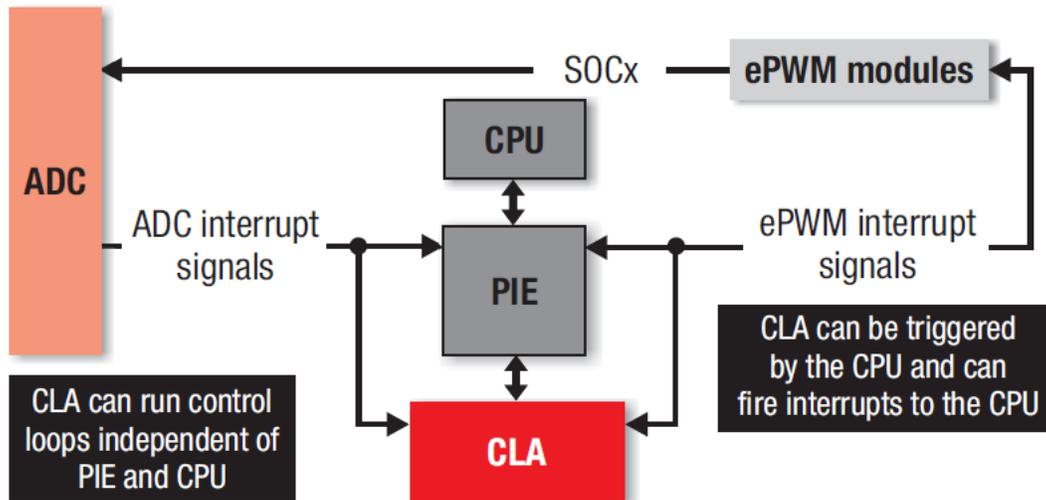


图 3-6. C28x 和 CLA 与 ADC 和 ePWM 模块之间的连接

出于以下几点原因，上述架构益处颇多：

- 处理数据几乎没有延迟，这通常是由 C28x 主内核的上下文切换以及将新的外部激励应用于系统引起的。
- 对当前的 C28x 程序执行没有中断或影响。
- 具有在同一 MCU 器件上独立运行并行控制系统的可行性。

CLA 具有自己的代码专用存储区和用于在它和系统中的 C28x CPU 之间传递信息的共享内存。CLA 在 Code Composer Studio™ 中受自己的 C 编译器支持。

### 3.4.3 器件列表

- [TMS320F2838xD/S](#)
- [TMS320F2837xD/S](#)
- [TMS320F2807x](#)
- [TMS320F28004x](#)
- [TMS320F28003x](#)

### 3.4.4 硬件平台和软件示例

- 谷底开关升压功率因数校正 (PFC) 参考设计
- 用于工业电机控制的 C2000 DesignDRIVE 开发套件

### 3.4.5 文档

- 展示 TI 的 C2000™ CLA 独特功能的软件示例
- CLA 实践技术讲座
- CLA 在谷底开关升压功率因数校正 (PFC) 参考设计中的应用
- E2E 上的 CLA 常见问题解答
- 增强 C2000™ MCU 系列器件的性能和功能
- 展示 TI 的 C2000™ CLA 独特功能的软件示例

## 3.5 灵活的系统互连：C2000 X-Bar

### 3.5.1 价值主张

三个片上信号交叉开关 (X-bar)：输入、输出和 ePWM 在硬件中提供了必要的机制，可以跨不同的控制系统实现有效地连接多个子系统。其关键优势包括更低的系统延迟、更简单的 PCB 布线和一致的时序。

### 3.5.2 深入介绍

在实时控制 MCU 中，传感和驱动子系统之间存在着明显的相互依赖关系。无论是进入到器件的传入信号、片上逻辑 (比较器、SDFM、ADC 等) 生成的信号，还是输出信号；在系统中路由通过这些信号可能非常具有挑战性。片上 X-bar 提供了一种灵活的机制，可以在 MCU 内部的硬件中实现这一点。这些模块为系统提供了三个主要优势：

- 简单地将外部信号路由到芯片中：任何 GPIO 都可以进入多个片上模块 (图 3-7)。例如，借助该特性，eCAP 模块可选择使用任何输入引脚作为其信号源，或者选择要传入到 CPU/CLA 的任何输入引脚作为外部中断。这也为印刷电路板 (PCB) 的信号布线和布局提供了灵活性。因为引脚不是针对一组功能进行固定配置的。

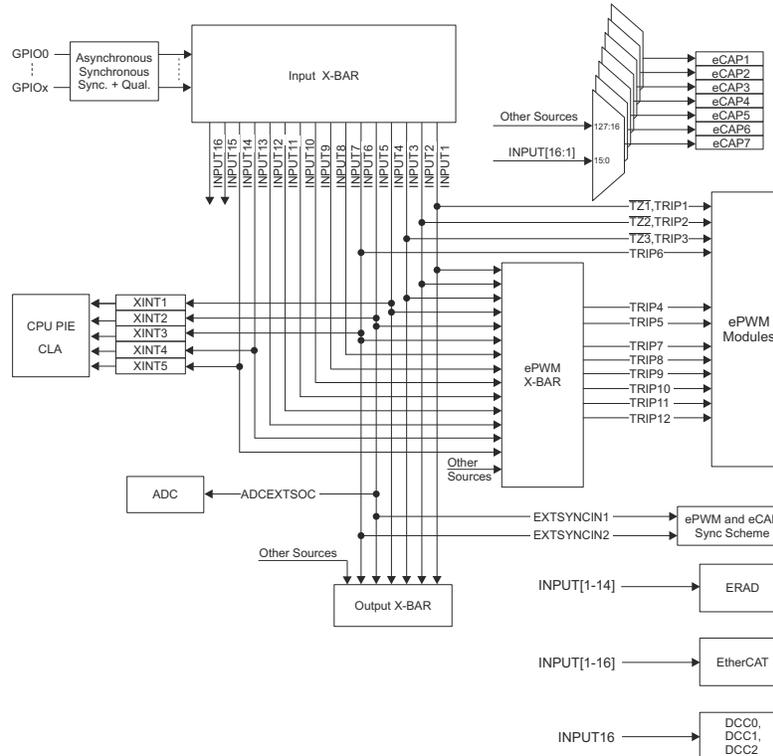


图 3-7. TMS320F2837xD MCU 上的输入 X-Bar

- 更低的系统延迟：对于简单的点对点传输，使用 X-bar 时，C28x 主 CPU 或 CLA 不需要使用指令将信号从一个域路由到另一个域。在每个 X-bar 的内部，还存在一个简单的逻辑或 (图 3-8)，它允许将硬件中 X-bar 的任

何输入组合在一起。由于多路复用器的选择全部由软件控制并且可以根据需要动态更改，因此这不仅节省了处理器的周期，而且还提高了系统的灵活性。

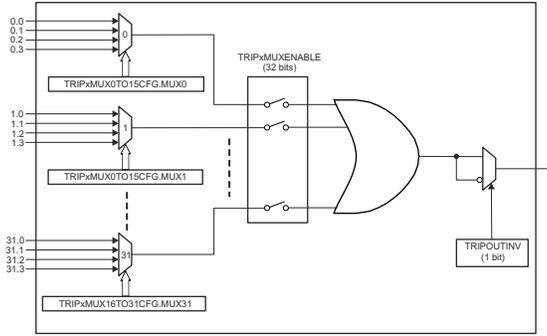


图 3-8. TMS320F2837xD MCU 上的本地多路复用器和逻辑或

- 一致的时序：与上述特性类似，由于没有 CPU 参与，信号在发生时会展实时地基于系统时钟传播，并且不依赖于另一个块来允许信号通过 (图 3-9)。这将导致更可预测和可重复的系统行为。考虑到许多信号要传递到驱动子系统的性质，时序尤其重要。

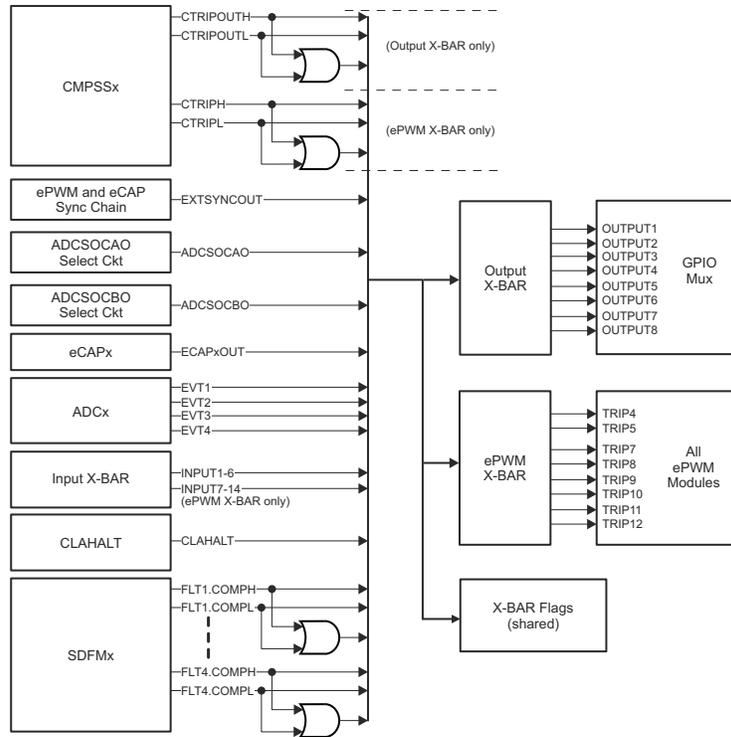


图 3-9. TMS320F2837xD MCU 上的 X-Bar 源和目标

### 3.5.3 器件列表

- [TMS320F2838xD/S](#)
- [TMS320F2837xD/S](#)
- [TMS320F2807x](#)
- [TMS320F28004x](#)
- [TMS320F28003x](#)
- [TMS320F28002x](#)
- [TMS320F280013x](#)
- [TMS320F280015x](#)

### 3.5.4 硬件平台和软件示例

- 用于工业电机控制的 [C2000 DesignDRIVE 开发套件](#)

- 使用 TI SysConfig 工具简化 X-bar 设置

### 3.5.5 文档

- [TMS320F2838xD 实时微控制器技术参考手册](#)
- [TMS320F2837xD 实时微控制器技术参考手册](#)
- [《TMS320F28004x 实时微控制器技术参考手册》](#)
- [TMS320F28003x 实时微控制器技术参考手册](#)
- [《TMS320F28002x 实时微控制器技术参考手册》](#)
- [TMS320F280013x 实时微控制器技术参考手册](#)
- [TMS320F280015x 实时微控制器技术参考手册](#)

## 3.6 使用非线性 PID 控制改善控制性能

### 3.6.1 价值主张

非线性 PID (NLPID) 提供了改善控制环路性能的能力，可实现线性控制器通常无法获得的性能。该控制器位于[数字控制库 \(DCL\)](#) 中，可轻松与现有 C2000 代码集成。

### 3.6.2 深入介绍

诸如比例、积分、微分 (PID) 之类的常规线性控制器已广泛用于包括电机控制和运动控制在内的数字电源应用。DCL 提供的非线性 PID ([DCL 培训视频 - 非线性控制](#)) 通过使用非线性定律对环路误差进行整形，从而扩展了线性 PID 的性能。非线性成形块与三个控制器路径串联，如下所示。

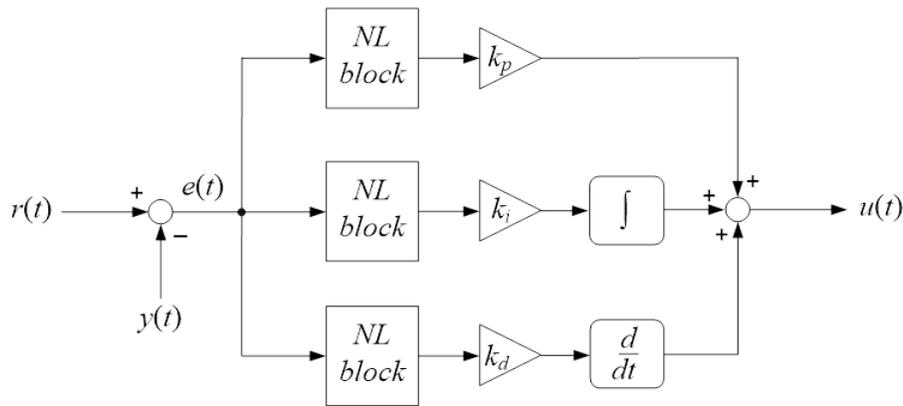


图 3-10. 非线性 PID 方框图

可以通过六个附加控制器参数 (每个非线性块中有两个) 来配置非线性动作的形状和主动性，这些参数通常以迭代方式进行调整以优化瞬态响应。与 DCL 中的其他控制器一样，NLPID 参数可以使用阴影参数集和更新函数来安全地更新。

图 3-11 显示了使用非线性控制动作可改善阶跃响应的示例。

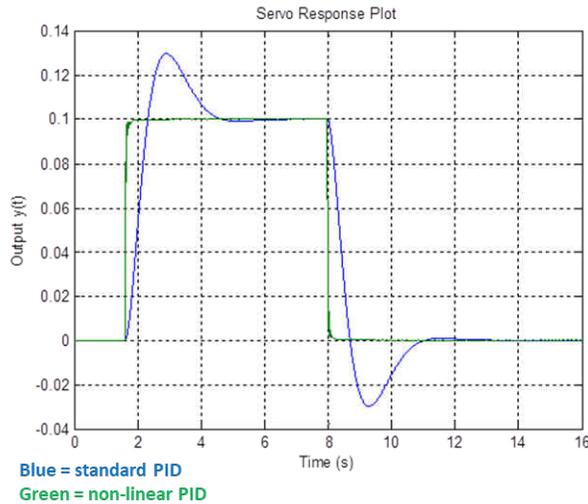


图 3-11. 线性和非线性 PID 的响应时间比较

NLPID 可在配备 Type 1 TMU 的器件（请参阅节 3.1）（例如 F280025 器件）上以最高效率执行运算。这些器件具有 CPU 指令，使得非线性控制器可在 117 个周期内完成运算执行，而没有此类指令的器件大约需要 3300 个周期。这种指令效率使非线性 PID 可以在高频应用中使用，例如开关电源和电流控制回路。

DCL 打包在 C2000Ware 中，可供 C2000 用户免费下载。该库包含 PID 控制器调优指南，可帮助用户尽可能发挥 NLPID 控制器的作用。

### 3.6.3 器件列表

- TMS320F28003x
- TMS320F28002x

### 3.6.4 硬件平台和软件示例

- C2000Ware 中的数字控制库
- F280039C controlCARD
- F280025 controlCARD

### 3.6.5 文档

- DCL 培训视频 - 非线性控制

## 3.7 了解实时控制应用中的闪存性能

### 3.7.1 价值主张

尽管闪存访问会受等待状态的影响，但与流水线缓冲区关联的 128 位宽预取逻辑使顺序代码执行性能等于 0 等待状态 (WS) RAM 的性能。考虑到常见的代码中断，相对于从 RAM 执行的代码，大多数应用的运行效率约为 80%。此外，数据总线上还有一个 128 位高速缓存，可提高数据读取性能。所有这些性能增强都带有零执行周期的错误代码校正 (ECC) 评估。

### 3.7.2 深入介绍

闪存是一种非易失性存储器，其优点在于即使在下电上电后也可以保留其内容。但是，由于其物理结构，闪存通常不如易失性存储器 (SRAM、DRAM 等) 的速度快。因此，在访问闪存时会使用等待状态以调整 MCU 时钟速率，这会影响 CPU 性能。为了大大降低其对性能的影响，C2000 FMC (闪存模块控制器) 的读取接口提供了预取模式 (图 3-12)。此模式显著提高了线性代码的性能，在实时控制系统中，线性代码通常占应用程序代码的大部分。

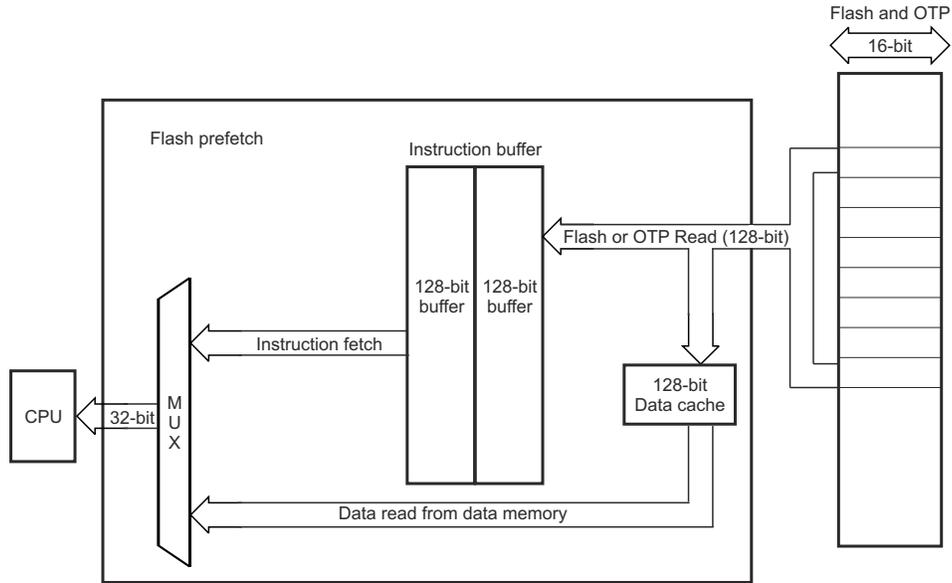


图 3-12. C2000 闪存预取模块

当在 C2000 MCU 上启用时，该模块从最后一条指令提取地址开始对线性地址增量进行前瞻性预读（128 位对齐），并将其存储在 128 位宽的 2 级深度指令预取缓冲区中。该缓冲区最多可容纳 16 个 16 位指令，并且随着 CPU 对缓冲区中已获取指令的持续使用，该缓冲区将通过预取机制在后台连续填充。因此，每个操作码的提取都不会产生等待状态，与每次等待状态提取相比，这是一个显著的性能提升。仅当程序计数器 (PC) 中断（例如分支、函数调用等）时，才会产生等待状态。

表 3-5 提供了一些可在两种不同类型 C2000 闪存器件上获得的代码效率的真实示例。

**表 3-5. 通过预取功能实现的有效闪存访问时间**

器件型号	器件属性	32 位浮点运算	16 位 If-Then-Else	ACI 电机信号链
TMS320F2838xD/S TMS320F2837xD/S	<ul style="list-style-type: none"> <li>200MHz CPU 时钟</li> <li>50MHz 闪存速度</li> <li>3 个等待状态</li> </ul>	<ul style="list-style-type: none"> <li>闪存访问效率：93%</li> <li>有效性能：188 MHz</li> </ul>	<ul style="list-style-type: none"> <li>闪存访问效率：87%</li> <li>有效性能：174MHz</li> </ul>	<ul style="list-style-type: none"> <li>闪存访问效率：92%</li> <li>有效性能：184 MHz</li> </ul>
TMS320F28004x TMS320F28002x	<ul style="list-style-type: none"> <li>100MHz CPU 时钟</li> <li>20MHz 闪存速度</li> <li>4 个等待状态</li> </ul>	<ul style="list-style-type: none"> <li>闪存访问效率：84%</li> <li>有效性能：84MHz</li> </ul>	<ul style="list-style-type: none"> <li>闪存访问效率：84%</li> <li>有效性能：84MHz</li> </ul>	<ul style="list-style-type: none"> <li>闪存访问效率：82%</li> <li>有效性能：82MHz</li> </ul>
TMS320F28003x	<ul style="list-style-type: none"> <li>120 MHz CPU 时钟</li> <li>20MHz 闪存速度</li> <li>5 个等待状态</li> </ul>	<ul style="list-style-type: none"> <li>闪存访问效率：70%</li> <li>有效性能：84MHz</li> </ul>	<ul style="list-style-type: none"> <li>闪存访问效率：78%</li> <li>有效性能：93 MHz</li> </ul>	<ul style="list-style-type: none"> <li>闪存访问效率：73%</li> <li>有效性能：87 MHz</li> </ul>
TMS320F280013x、 TMS320F280015x	<ul style="list-style-type: none"> <li>120 MHz CPU 时钟</li> <li>40 MHz 闪存速度</li> <li>2 个等待状态</li> </ul>	<ul style="list-style-type: none"> <li>闪存访问效率：96%</li> <li>有效性能：115 MHz</li> </ul>	<ul style="list-style-type: none"> <li>闪存访问效率：91%</li> <li>有效性能：109 MHz</li> </ul>	<ul style="list-style-type: none"> <li>闪存访问效率：95%</li> <li>有效性能：114 MHz</li> </ul>

**备注**

闪存等待状态数始终比访问时间少 1。

- FLASH\_wait\_states = FLASH\_access\_cycles - 1
- FLASH\_access\_cycles = round\_up(CPU MHz/FLASH MHz) 例如：= round\_up(165/50) = round\_up(3.3) = 4

到目前为止，我们已经讨论了闪存执行程序代码的效率问题，但它也有一个 128 位数据缓存。用户可以启用此功能来提高闪存数据读取性能。当 CPU 从闪存地址请求数据时，闪存包装程序会将整个 128 位的闪存数据（对齐）存储在此缓存中，而不是简单地将请求的地址数据提供给 CPU。CPU 可以访问此缓存中的剩余数据，而不会产生任何等待状态。当缓存未命中时，系统将刷新并重新填充此数据缓存。

最后，对于闪存的每 64 位还有一个纠错码 (ECC) 值。ECC 可为每 64 位提供 single-bit 校正和 dual-bit 错误检测功能。在将数据放入预取缓冲区之前，系统将对 ECC 的正确性进行评估，而不会影响前面提到的访问时间/延迟。如果检测到不可纠正的错误，则将生成不可屏蔽的中断，以停止与 CPU 的正常代码执行并行的正常代码执行。您还可以设置可纠正错误的阈值，以触发对 C28x 内核的中断。

### 3.7.3 器件列表

- [TMS320F2838xD/S](#)
- [TMS320F2837xD/S](#)
- [TMS320F2807x](#)
- [TMS320F28004x](#)
- [TMS320F28003x](#)
- [TMS320F28002x](#)
- [TMS320F280013x](#)
- [TMS320F280015x](#)

### 3.7.4 硬件平台和软件示例

受支持器件的所有代码示例均在 `SysCtrl.c` 文件内将预取缓冲区和 ECC 逻辑作为其初始化例程的一部分。

### 3.7.5 文档

- [信号链基准 - 演示了 C2000™ MCU 经过优化的实时性能](#)
- [带连接管理器的 TMS320F2838x 实时微控制器数据表](#) (更多信息, 请参阅[闪存参数部分](#))
- [TMS320F2837xD 双核微控制器数据表](#) (更多信息, 请参阅[闪存参数部分](#))
- [TMS320F28004x 微控制器数据表](#) (更多信息, 请参阅[闪存参数部分](#))
- [TMS320F28003x 实时微控制器数据表](#) (更多信息, 请参阅[闪存参数部分](#))
- [TMS320F28002x 实时微控制器数据表](#) (更多信息, 请参阅[闪存参数部分](#))
- [TMS320F280013x 实时微控制器数据表](#) (更多信息, 请参阅[闪存参数部分](#))
- [TMS320F280015x 实时微控制器数据表](#) (更多信息, 请参阅[闪存参数部分](#))

## 3.8 使用 C28x DSP 内核进行确定性程序执行

### 3.8.1 价值主张

对于任何实时控制系统, 程序执行随时间推移的一致性都是一个重要方面。无论是整个系统更新期间的性能, 还是一段时间内多次加电, 实时系统的优化都依赖于在已知时间点发生的关键系统事件。C28x MCU 定义明确的 8 级 CPU 流水线及其中断处理逻辑的互补行为提供了这种级别的确定性。

### 3.8.2 深入介绍

处理器行为随时间变化不一致的处理器会对实时控制系统造成干扰, 这可能是由于没有及时驱动系统, 或是在不正确的时间点对系统状态进行采样。此外, 实时控制系统在其程序流中很大程度上是由中断驱动的。鉴于此, 缓存内存是不符合需求的, 因为每当程序出现中断时 (在这种情况下, 经常发生), 经常需要将其丢弃。

相反, 对于程序执行以及具有指令管线的处理器来说, 大量的快速存储器是理想的, 该指令管线足够深以使指令并行化, 但也足够浅以至于在出现不连续时不会引起大量的时间损失。C28x CPU 使用 8 级管线, 如 [图 3-13](#) 所示。指令一旦进入管线的 D2 阶段, 就无法通过中断停止完全执行。相反, 当 C28x 内核接收到传入的中断时, 处于管线 D2 阶段之前的任何指令都将被清除。从中断程序返回时, 系统会从 F1 提取阶段再次开始执行。深入了解管线的细微差别不在本文的讨论范围之内, 但这些细节在 [TMS320C28x DSP CPU 和指令集参考指南](#)中均有介绍。

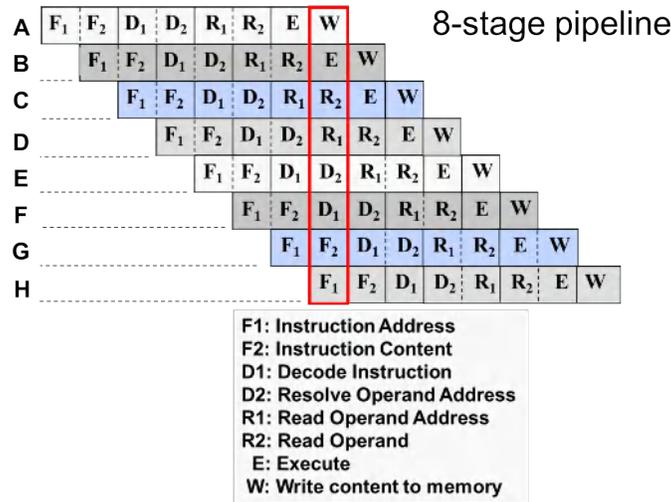


图 3-13. C28x 管线可视化图

虽然您可以看到程序执行的行为随着时间的推移是可重复的，但传入中断的行为又如何呢？图 3-14 表明这也是确定性的。请注意，除非手动更改，否则所有其他中断请求都将挂起，直到当前中断已得到完全处理。这一点很重要，因为一旦中断开始，就要保持对它的服务在时域内的一致性。C28x 内核的这两个器件都有助于维护系统的确定性代码执行。

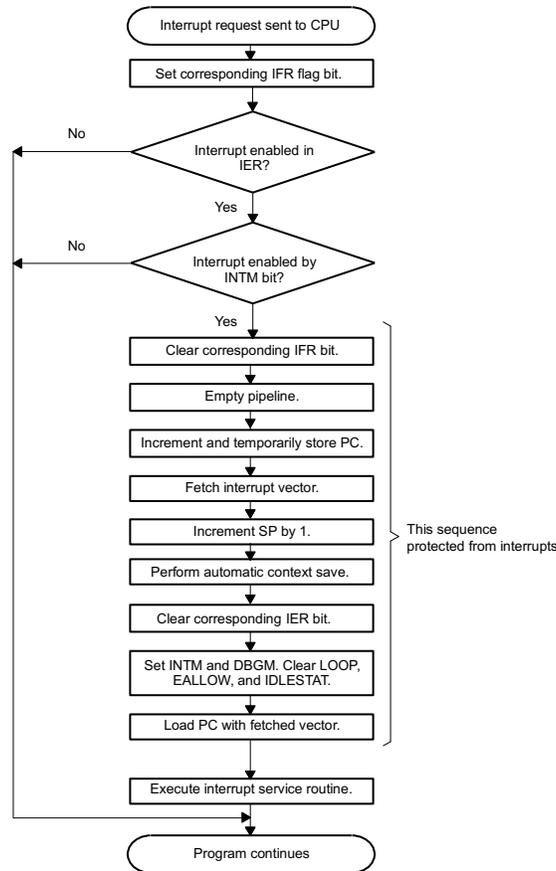


图 3-14. C28x CPU 可屏蔽中断的标准操作

### 3.8.3 器件列表

- [TMS320F2838xD/S](#)
- [TMS320F2837xD/S](#)
- [TMS320F2807x](#)
- [TMS320F28004x](#)
- [TMS320F28003x](#)
- [TMS320F28002x](#)
- [TMS320F280013x](#)
- [TMS320F280015x](#)

### 3.8.4 硬件平台和软件示例

- [F28388D controlCARD 评估模块](#)
- [C2000 MCU F28379D LaunchPad™ 开发套件](#)
- [C2000 MCU F280049C LaunchPad™ 开发套件](#)
- [F280039C controlCARD 评估模块](#)
- [F280025 controlCARD 评估模块](#)
- [F2800137 controlCARD 评估模块](#)
- [F2800157 controlCARD 评估模块](#)

### 3.8.5 文档

- [TMS320C28x CPU 和指令集参考指南](#)
- [C2000™ Academy 概述](#)

## 3.9 高效的实时固件更新 (LFU) 和固件无线 (FOTA) 更新

### 3.9.1 价值主张

服务器电源设备 (PSU) 等工业终端设备需要 LFU 支持，这样，固件更新时就不会使系统离线。汽车市场需要 FOTA 支持，车辆内任何受支持的 ECU ( 电子控制单元 ) 的固件都可以远程更新。C2000 实时 MCU 具有多闪存库支持、中断矢量表交换、RAM 块交换和软件示例等硬件功能，可以有效执行 LFU 和 FOTA。

### 3.9.2 深入介绍

术语：

由于术语并无统一标准定义 ( 至少对于 LFU 是这样 ) ，因此理解术语至关重要。使用以下定义：

- **安装** - 将固件应用或“映像”从主机控制器传输到目标器件 ( 固件将在目标上执行 )  
这还包括对目标器件闪存上的新固件进行编程。
- **激活** - 顾名思义，是指激活新的固件应用。在一些 LFU 文档中，还使用了术语“切换”
- **LFU** - 无需器件复位即可安装和激活新固件
- **FOTA** - 在旧固件运行时安装新固件，但激活新固件需要复位器件
- **交换** - 将多个物理硬件模块映射到同一地址空间的能力。硬件可以指闪存组、RAM 块、中断向量表等。

构建块：

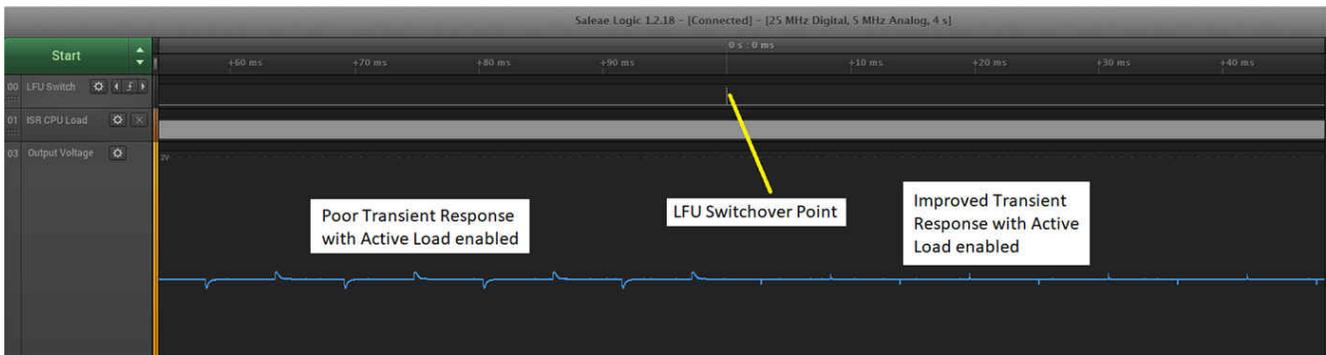
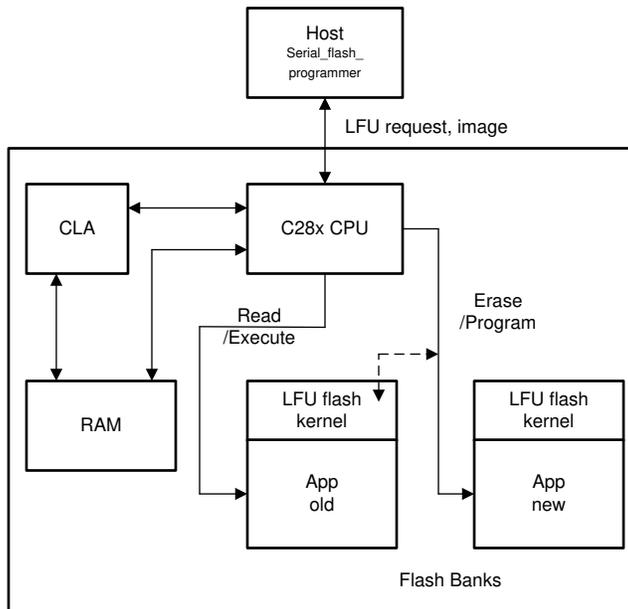
许多软件和硬件部件可用于启用 LFU 和 FOTA 等系统级功能：

- **多个闪存组** - 例如，一个闪存组包含活动固件，其他闪存组包含非活动固件
- **启用 LFU 的硬件特性** - 例如，中断矢量表交换和 RAM 块交换，可以快速激活 LFU。
- **编译器** - 一种支持 LFU 的编译器，使用户可以更轻松地将 LFU 支持集成到其解决方案中。对于 LFU，编译器使用现有的固件可执行文件作为参考来生成新的固件可执行文件，通过优化 LFU 初始化例程，可以快速高效地激活新固件可执行文件。
- **闪存内核** - 这些指的是作为示例提供的引导加载程序，位于闪存中，支持 LFU 和 FOTA 功能。闪存内核可以与主机交互以安装固件，并包含组选择逻辑，以确定在器件复位后闪存组是否包含有效固件，以及需要执行哪些固件。组选择逻辑现在也内置在 C2000 MCU 的引导 ROM 中，并可用作引导模式。

- **LFU 软件示例** - 简单示例以及参考设计和详细的用户指南，将帮助您了解所有构建块和 **LFU** 流程，从而快速将 **LFU** 集成到系统并实现出色的性能。将以 **C28x CPU** 和 **CLA** 上的 **LFU** 为例进行说明。只要稍加改动，这些示例也可以重新用于 **FOTA**。

为 **LFU** 定义了两个流程：

- 生产中：
  - 对一个或多个闪存库中的闪存内核、固件进行编程
- 在现场：
  - 选项 1 - 使用此方法，您需要了解固件更新的目标闪存组：
    - 开发人员创建新的映像（使用前一个映像作为参考映像和支持 **LFU** 的编译器），并且知道它是针对哪个目标闪存组
    - 主机启动 **LFU** - 下载对应于非活动组的映像
    - 闪存内核将映像安装到非活动库
    - 旧应用程序（具有 **LFU** 软件/硬件支持）激活新应用程序，而无需复位器件（如果执行完上一步骤）
    - 旧应用程序的存在允许在出现故障时使用回退选项
  - 选项 2 - 使用此方法，您**无需**知道固件更新的目标闪存组：
    - 开发人员创建两个新映像（使用前一个映像用作参考映像和支持 **LFU** 的编译器），无需知晓目标闪存组是哪个。例如，如果使用 2 个闪存组，新版本为 **vN**，则用户可以为闪存组 1 创建 **vN**，并以此为参考，为闪存组 0 创建 **vN-1**。用户还可以为闪存组 0 创建 **vN**，并以此为参考，为闪存组 1 构建 **vN-1**。
    - 主机启动 **LFU** - 将**两个映像**传输至目标器件
    - 闪存内核只安装针对非活动闪存组的映像，而忽略针对活动闪存组的映像
    - 旧应用程序（具有 **LFU** 软件/硬件支持）激活新应用程序，而无需复位器件（如果执行完上一步骤）
    - 旧应用程序的存在允许在出现故障时使用回退选项



有两个流程也适用于 FOTA：

- 生产中：
  - 对一个或多个闪存库中的闪存内核、固件进行编程
- 在现场：
  - 如果没有硬件支持闪存组交换，工厂需要知道要为哪个闪存组（闪存组 1 或闪存组 0）构建固件。这种方法是不可接受的，因为使用汽车 FOTA，您可以自由地从一个版本更新到其他版本（即跳过版本），因此不可能形成这样的约束
  - 因此，一种可行的方法是构建加载到闪存组 1 并从闪存组 0 运行的固件。由于不支持交换，激活过程涉及将映像从闪存组 1 复制到闪存组 0。这里的问题是，一旦固件从闪存组 1 复制到闪存组 0，两个闪存组会有相同的新固件（如需回滚，无有效备份）。由于回滚是 FOTA 的基本要求，因此需要第三个分区或闪存组
  - 开发人员创建要加载到闪存组 1 并从闪存组 0 运行的新固件
  - 主机（FOTA 主 ECU）启动 FOTA - 将映像传输到目标器件
  - 闪存内核将映像安装到闪存组 1
  - 主机启动器件复位。闪存内核将闪存组 0 中的旧映像复制到闪存组 2，以便映像在需要回滚时可用。然后闪存内核将闪存组 1 中的新映像复制到闪存组 0。然后会执行常规映像激活步骤



### 4.1.2 深入介绍

所有 PWM 控制的电源拓扑本质上都受带宽的限制即控制器将 PWM 边沿放置在尽可能接近控制律的数学解的能力。无论“四舍五入”产生的误差是多少，以输出 PWM 信号形式创建的解决方案都决定了系统可以实现的最大效率。

从这个意义上讲，其可能有助于将 PWM 视为一种具有固定分辨率的 DAC。在选择下一个可用 PWM 边沿位置时产生的任何误差都将等同于任何 DAC 固有的量化误差项。因此，PWM 模块可实现的最小时间阶跃可以转换为该等效 DAC 的分辨率的“位”。

如表 4-1 所示，与传统的 PWM 相比，C2000 MCU HRPWM 的分辨率明显提高，有效分辨率增加了约 6 位。

表 4-1. PWM 与 HRPWM 的分辨率比较

PWM 频率 (kHz)	常规分辨率 (PWM) 100MHz EPWMCLK		高分辨率 (HRPWM)	
	位	误差百分比	位	误差百分比
20	12.3	0.02	18.1	0.000
50	11	0.05	16.8	0.001
100	10	0.1	15.8	0.002
150	9.4	0.15	15.2	0.003
200	9	0.2	14.8	0.004
250	8.6	0.25	14.4	0.005
500	7.6	0.5	13.4	0.009
1000	6.6	1	12.4	0.018
1500	6.1	1.5	11.9	0.027
2000	5.6	2	11.4	0.036

### 4.1.3 器件列表

- [TMS320F2838xD/S](#)
- [TMS320F2837xD/S](#)
- [TMS320F2807x](#)
- [TMS320F28004x](#)
- [TMS320F28003x](#)
- [TMS320F28002x](#)
- [TMS320F280013x](#)
- <https://www.ti.com.cn/product/cn/TMS320F2800157>

### 4.1.4 硬件平台和软件示例

- [TIDM-02002](#)，适用于混合动力汽车/电动汽车车载充电器的双向 CLLLC 谐振双有源电桥
- [TIDA-00961](#)，基于 GaN 的高效率 1.6kW 高密度 1MHz CrM 图腾柱 PFC 转换器
- [TIDA-010054](#)，适用于 3 级电动汽车充电站的双向双有源电桥参考设计
- [针对 TMS320F28388D 的 C2000Ware HRPWM 示例](#)
- [针对 TMS320F28379D 的 C2000Ware HRPWM 示例](#)
- [针对 TMS320F280049C 的 C2000Ware HRPWM 示例](#)
- [针对 TMS320F280039C 的 C2000Ware HRPWM 示例](#)
- [针对 TMS320F280025 的 C2000Ware HRPWM 示例](#)
- [针对 TMS320F2800137 的 C2000Ware HRPWM 示例](#)
- [针对 TMS320F2800157 的 C2000Ware HRPWM 示例](#)

### 4.1.5 文档

- [TMS320F2838xD 双核实时微控制器技术参考手册](#)
- [TMS320F2837xD 双核实时微控制器技术参考手册](#)
- [《TMS320F28004x 实时微控制器技术参考手册》](#)
- [TMS320F28003x 实时微控制器技术参考手册](#)
- [《TMS320F28002x 实时微控制器技术参考手册》](#)
- [TMS320F280013x 实时微控制器技术参考手册](#)
- [TMS320F280015x 实时微控制器技术参考手册](#)
- [C2000 F2837xD 微控制器 1 日培训讲座第 1.6 节 - 控制外设](#)，跳至 14:38 处可查看 HRPWM

## 4.2 具有可配置死区的电流控制拓扑击穿预防措施

### 4.2.1 价值主张

C2000 MCU 能够在硬件中使用片上比较器来控制 PWM 占空比，从而实现电流控制技术，例如峰值电流模式控制 (PCMC)。Type 4 PWM 模块集成了可变死区控制，无需占 CPU 资源就可以防止输出短路。

### 4.2.2 深入介绍

为了提高电源效率，许多直流/直流系统都实现了同步升压控制器，其中第二个开关管取代了常规升压控制器中的前馈二极管 (图 4-2)。峰值电流模式控制是用于控制此拓扑的较常见方法之一，并且 C2000 MCU 具有一些独特的功能，可使其非常有效地实施这种类型的控制。

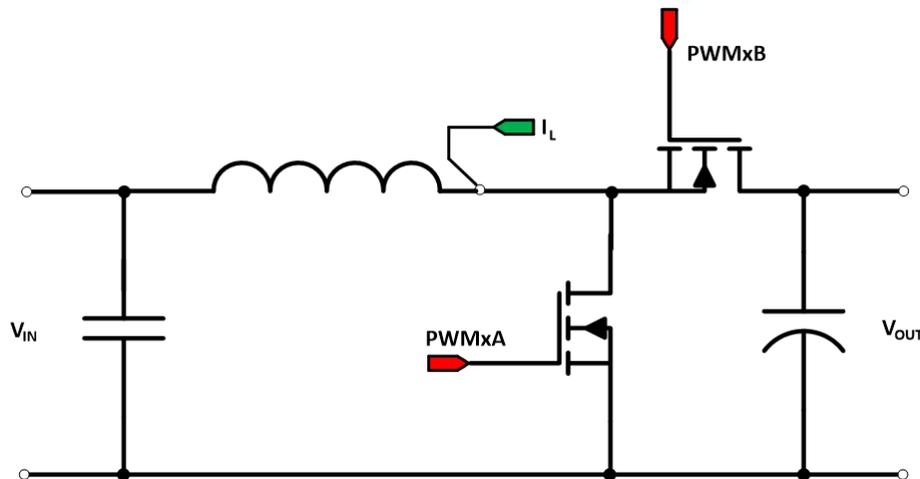


图 4-2. 同步升压控制器

添加第二个 FET 需要精确控制一次和二次开关相对于彼此的开/关时间。如果两个开关管同时打开，则会产生短路电流，不仅效率降低，而且可能对 FET 开关管造成破坏。

实现此类系统的一种准确方法是让比较器监测电感器电流，并在电流超过预定义阈值时驱动 FET。理想情况下，当一个 FET 导通时，另一个 FET 可以同时关断。但是，由于开关失配和电路板传播延迟，控制器的同时切换可能不会实现 FET 的同时切换，从而会产生前面提到的击穿现象。尽管软件中的技术有助于阻止第二个开关管的切换以避免这种情况，但在各种硬件相互依赖性以及控制环路时间受限的情况下，实现这些技术可能非常困难。

C2000 MCU 已实现了可编程的死区控制（来自比较器输出本身），以防止这种情况发生，同时保持 C28x CPU 的空载（图 4-3）。这使得一旦初始化，就可以在 CPU 域之外实现完整的 PCMC 解决方案。该逻辑存在于特定器件的所有 PWM 模块上，从而允许许多相电路具有不同的死区，例如存在多个开关对的移相全桥。

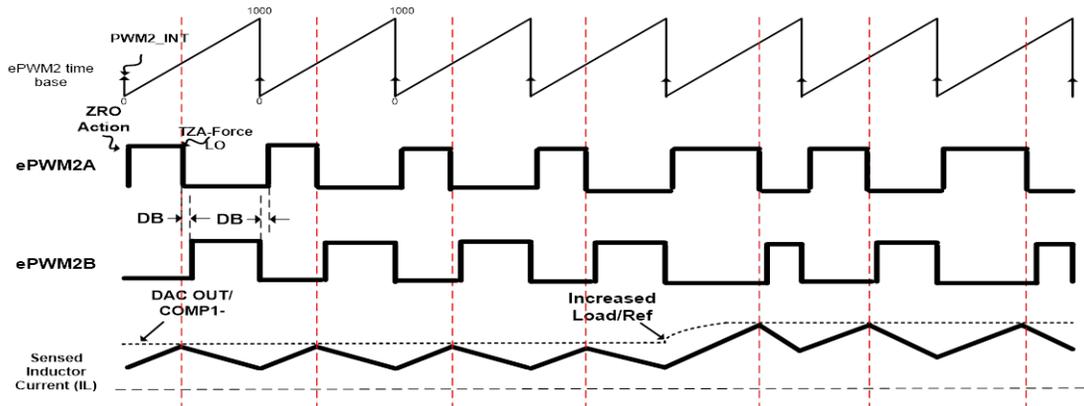


图 4-3. 具有可配置死区的 COMP 模块的逐周期跳闸动作

#### 4.2.3 器件列表

- [TMS320F2838xD/S](#)
- [TMS320F2837xD/S](#)
- [TMS320F2807x](#)
- [TMS320F28004x](#)
- [TMS320F28003x](#)
- [TMS320F28002x](#)
- [TMS320F280013x](#)
- [TMS320F280015x](#)

#### 4.2.4 文档

- [TMS320F2838xD 实时微控制器技术参考手册](#)（更多信息，请参阅 [增强型脉宽调制器 \(ePWM\)](#) 一章的 [动作限定符 \(AQ\)](#) 子模块和 [死区生成器 \(DB\)](#) 子模块部分）
- [TMS320F2837xD 实时微控制器技术参考手册](#)（更多信息，请参阅 [增强型脉宽调制器 \(ePWM\)](#) 一章的 [动作限定符 \(AQ\)](#) 子模块和 [死区生成器 \(DB\)](#) 子模块部分）
- [TMS320F28004x 实时微控制器技术参考手册](#)（更多信息，请参阅 [增强型脉宽调制器 \(ePWM\)](#) 一章的 [动作限定符 \(AQ\)](#) 子模块和 [死区生成器 \(DB\)](#) 子模块部分）
- [TMS320F28003x 实时微控制器技术参考手册](#)（更多信息，请参阅 [增强型脉宽调制器 \(ePWM\)](#) 一章的 [动作限定符 \(AQ\)](#) 子模块和 [死区生成器 \(DB\)](#) 子模块部分）
- [TMS320F28002x 实时微控制器技术参考手册](#)（更多信息，请参阅 [增强型脉宽调制器 \(ePWM\)](#) 一章的 [动作限定符 \(AQ\)](#) 子模块和 [死区生成器 \(DB\)](#) 子模块部分）
- [TMS320F280013x 实时微控制器技术参考手册](#)（更多信息，请参阅 [增强型脉宽调制器 \(ePWM\)](#) 一章的 [动作限定符 \(AQ\)](#) 子模块和 [死区生成器 \(DB\)](#) 子模块部分）
- [TMS320F280015x 实时微控制器技术参考手册](#)（更多信息，请参阅 [增强型脉宽调制器 \(ePWM\)](#) 一章的 [动作限定符 \(AQ\)](#) 子模块和 [死区生成器 \(DB\)](#) 子模块部分）
- [C2000 Academy ePWM 模块](#)

### 4.3 使用 C2000 可配置逻辑块进行片上硬件定制

#### 4.3.1 价值主张

可配置逻辑块 (CLB) 提供了一种在芯片内部创建自定义逻辑的机制。将其用于增强现有子系统、创建新的子系统以及替换片外逻辑电路都是可能的。

### 4.3.2 深入介绍

片上 CLB 提供了一种灵活的机制，可以在 MCU 内部的硬件中添加个性化的逻辑定制。无论是修改现有子系统以满足应用特定需求（实现旋转传感解决方案 节 2.5），还是创建全新的子系统，或者替换外部逻辑电路，CLB 都可以完成所有这些任务。

CLB 模块为系统提供了三个主要好处：

1. 增强现有的片上子系统：在 CLB 模块内部实现的自定义逻辑可以插入片上其他子系统（例如 ePWM 子系统）中，以增强外设的功能。这样的示例包括创建 T-format 绝对编码器 (List item.) 接口或脉冲序列输出 (List item.)。

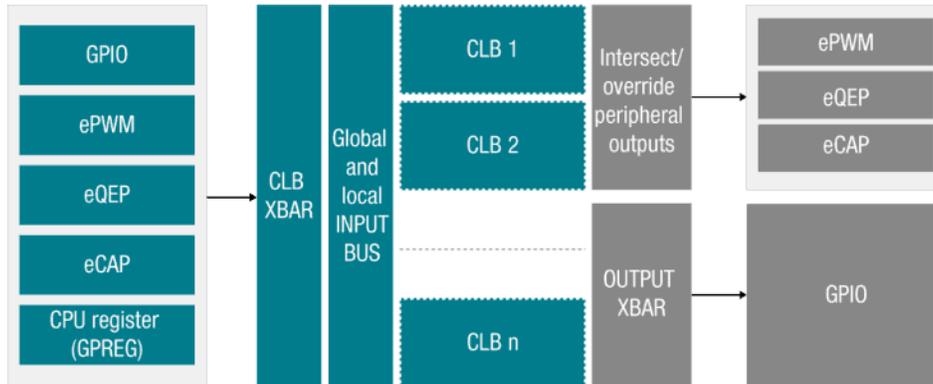


图 4-4. C2000 MCU 架构中的 CLB 集成

2. 创建新的子系统：CLB 模块可以进行组合以形成新的子系统，这些子系统可以是 C2000 MCU 中不存在的全新外设，也可以复制一个 C2000 MCU 外设并创建一个额外的子系统，例如：
  - a. 创建辅助 PWM 模块软件示例
  - b. 使用 C2000 可配置逻辑模块进行设计，如 CLB 状态机软件示例所示
3. 替换外部逻辑：在某些情况下，CLB 模块可用于代替外部器件，如实现特定于应用的自定义逻辑的 FPGA 或 CPLD。CLB 模块也可用于替换外部逻辑电路。其中一些示例包括：
  - a. 从 FPGA/CPLD 迁移到 CLB
  - b. 在 CLB 软件示例中用自定义逻辑替换外部 PWM 保护电路

虽然自定义逻辑的物理实现是通过 C2000 MCU 上的寄存器控制的，但是 TI 也提供了一些诸如 CLB 配置工具 (图 4-5) 的 GUI 工具，既可以达成逻辑实现，也可以在系统中使用之前验证仿真中的逻辑操作。

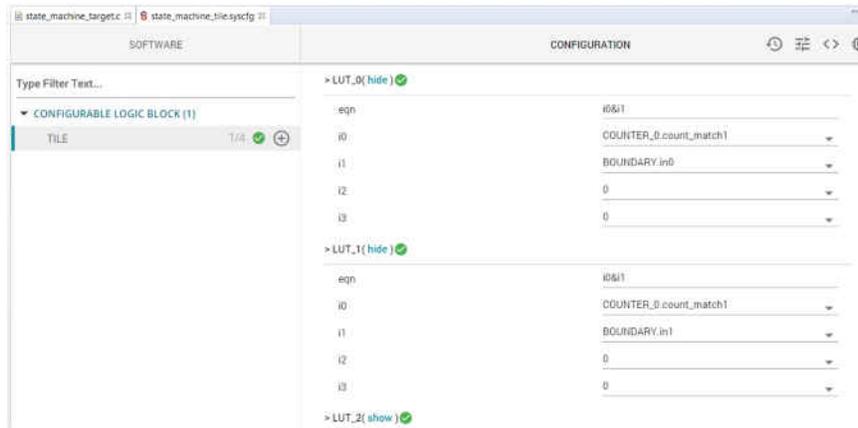


图 4-5. SysConfig 中的 CLB 配置工具

### 4.3.3 器件列表

- [TMS320F2838xD/S](#)
- [TMS320F2837xD/S](#)
- [TMS320F2807x](#)
- [TMS320F28004x](#)
- [TMS320F28003x](#)
- [TMS320F28002x](#)

### 4.3.4 硬件平台和软件示例

- [F28388D controlCARD 评估模块](#)
- [LAUNCHXL-F28379D](#)
- [LAUNCHXL-F280049C](#)
- [F280039C controlCARD 评估模块](#)
- [LAUNCHXL-F280025C](#)
- [创建辅助 PWM 模块软件示例](#)
- [CLB 状态机软件示例](#)
- [在 CLB 软件示例中将外部 PWM 保护电路替换为自定义逻辑](#)

### 4.3.5 文档

- [CLB 编程工具培训 \(视频\)](#)
- [使用 C2000™ 可配置逻辑块进行设计](#)
- [如何将自定义逻辑从 FPGA/CPLD 迁移到 C2000™ 微控制器](#)
- [CLB 工具用户指南](#)

## 4.4 快速检测过电流及欠电流和欠电压

### 4.4.1 价值主张

每个控制系统都会遇到随机事件，这些事件可能会对系统造成损坏。对这些事件的快速检测和反应对于保持系统安全并处于良好的工作状态至关重要。片上比较器可以检测这些事件并对其做出反应，所需时间仅为 ADC 和处理器所耗费时间的一小部分。

### 4.4.2 深入介绍

在大多数系统中，故障检测和保护都很重要，不仅是为了避免产生不确定的输出，而且还在于防止损坏主控板 (PCB) 之上和之外的器件。故障检测的速度以及最终 FET 输出状态的变化对系统至关重要。我们已在 C2000 MCU 上实现了将模拟和数字域集成在一起的专用子系统，以用于处理此需求：比较器子系统，简称 CMPSS (图 4-6)。

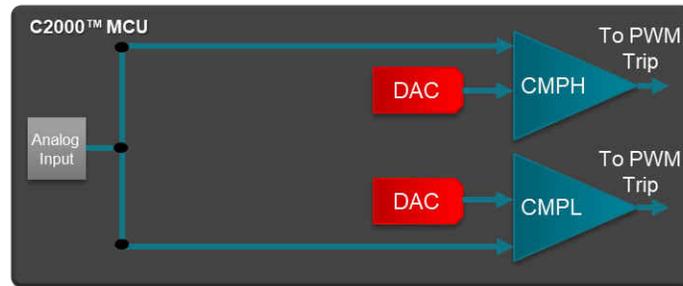


图 4-6. CMPSS 可视化图

每个 C2000 MCU 最多具有 8 个 CMPSS 模块，带有内部 DAC，这些 DAC 给出了被监测线路的反相/比较检测电平。如 图 4-6 所示，每个 CMPSS 模块都有两个比较器，用于同时进行高电平和低电平检测。与使用 ADC 进行故障检测相比，使用 CMPSS 有几个优势：

- 系统开销：在初始设置之后，使用 CMPSS 监测引脚基本上是不需要软件参与。使能比较器后，将始终根据比较值对引脚进行监测。其他技术将需要定期进行 ADC 转换和阈值检查。
- 延迟：虽然可以将 ADC 采样率简单地计入控制环路的周期，但对于故障情况，并没有一个确定的常数。因此，无论是采样点还是 ADC 本身的转换时间，都将存在检测故障的固有延迟。比较器没有此类触发要求或采样时间，它将连续监测模拟信号。
- 专用的 PWM 跳闸区输入：每个 CMPSS 模块的输出都可以直接连接到任何 PWM 的跳闸区中，并且可以在软件中配置接收信号时的动作。这个过程不需要软件开销，跟处理 ADC ISR，然后在软件中对 PWM 进行控制不一样。
- 无时钟相关性：根据定义，比较器是一个纯模拟域电路，因此不存在基于输入的输出变化状态的时钟相关性。C2000 MCU 对此进行了转发，以提供从比较器到 PWM 的异步路径。除了消除任何时钟相关性之外，这样一来，还实现了最短的故障检测到引脚状态改变时间 (表 4-2)。

表 4-2. 故障检测与跳闸方法的比较

采样方法	采样时间 (最小值)	结果就绪 (最小值)	锁存和更改 PWM 引脚 (200MHz 系统时钟)	从故障到跳闸的总时间
12 位 ADC	75ns	260ns	大约 100ns (包括 ISR)	435ns
12 位 ADC (带 PPB)	75ns	260ns	10ns	355ns
CMPSS	不适用	不适用	不适用	60ns

- 同时检测高低电平：CMPSS 模块的每个输入将信号路由到两个物理上独立的比较器，这些比较器可以同时检测过冲和下冲。

### 4.4.3 器件列表

- TMS320F2838x/D/S

- [TMS320F2837xD/S](#)
- [TMS320F2807x](#)
- [TMS320F28004x](#)
- [TMS320F28003x](#)
- [TMS320F28002x](#)
- [TMS320F280013x](#)
- [TMS320F280015x](#)

#### 4.4.4 硬件平台和软件示例

以下套件可用于实现 CMPSS 以检测超出范围的电流/电压事件

- [TIDM-02002](#)，适用于混合动力汽车/电动汽车车载充电器的双向 CLLLC 谐振双有源电桥参考设计
- [TIDM-1022](#)，谷底开关升压功率因数校正
- [TMDXIDDK379D](#)，适用于工业电机控制的 C2000 DesignDRIVE 开发套件

#### 4.4.5 文档

- [带连接管理器的 TMS320F2838x 实时微控制器数据表](#) (更多信息，请参阅[比较器子系统 \(CMPSS\)](#) 一章)
- [TMS320F2837xD 双核微控制器数据表](#) (更多信息，请参阅[比较器子系统 \(CMPSS\)](#) 部分)
- [TMS320F28004x 微控制器数据表](#) (更多信息，请参阅[比较器子系统 \(CMPSS\)](#) 部分)
- [TMS320F28003x 微控制器数据表](#) (更多信息，请参阅[比较器子系统 \(CMPSS\)](#) 部分)
- [TMS320F28002x 实时微控制器数据表](#) (更多信息，请参阅[比较器子系统 \(CMPSS\)](#) 部分)
- [TMS320F280013x 实时微控制器数据表](#) (更多信息，请参阅[比较器子系统 \(CMPSS\)](#) 部分)
- [TMS320F280015x 实时微控制器数据表](#) (更多信息，请参阅[比较器子系统 \(CMPSS\)](#) 部分)
- [TMS320F2838xD 实时微控制器技术参考手册](#) (更多信息，请参阅[比较器子系统 \(CMPSS\)](#) 一章)
- [TMS320F2837xD 实时微控制器技术参考手册](#) (更多信息，请参阅[比较器子系统 \(CMPSS\)](#) 一章)
- [TMS320F28004x 实时微控制器技术参考手册](#) (更多信息，请参阅[比较器子系统 \(CMPSS\)](#) 一章)
- [TMS320F28003x 实时微控制器技术参考手册](#) (更多信息，请参阅[比较器子系统 \(CMPSS\)](#) 一章)
- [TMS320F28002x 实时微控制器技术参考手册](#) (更多信息，请参阅[比较器子系统 \(CMPSS\)](#) 一章)
- [TMS320F280013x 实时微控制器技术参考手册](#) (请参阅[比较器子系统 \(CMPSS\)](#) 一章)
- [TMS320F280015x 实时微控制器技术参考手册](#) (请参阅[比较器子系统 \(CMPSS\)](#) 一章)

## 4.5 通过高分辨率相位控制提高系统功率密度

### 4.5.1 价值主张

在用于电动汽车的典型充电系统（例如车载充电器或非板载直流充电器）中，在输出电压和电流下仅允许产生 1% 的纹波。双有源电桥（图 4-7）是一种常用拓扑结构，可在这些充电系统中实现直流/直流转换，因为它能够实现宽电压工作范围。C2000 HRPWM 能够将相移控制在 150ps 之内，从而可以减小组件尺寸，同时仍然满足输出纹波要求。

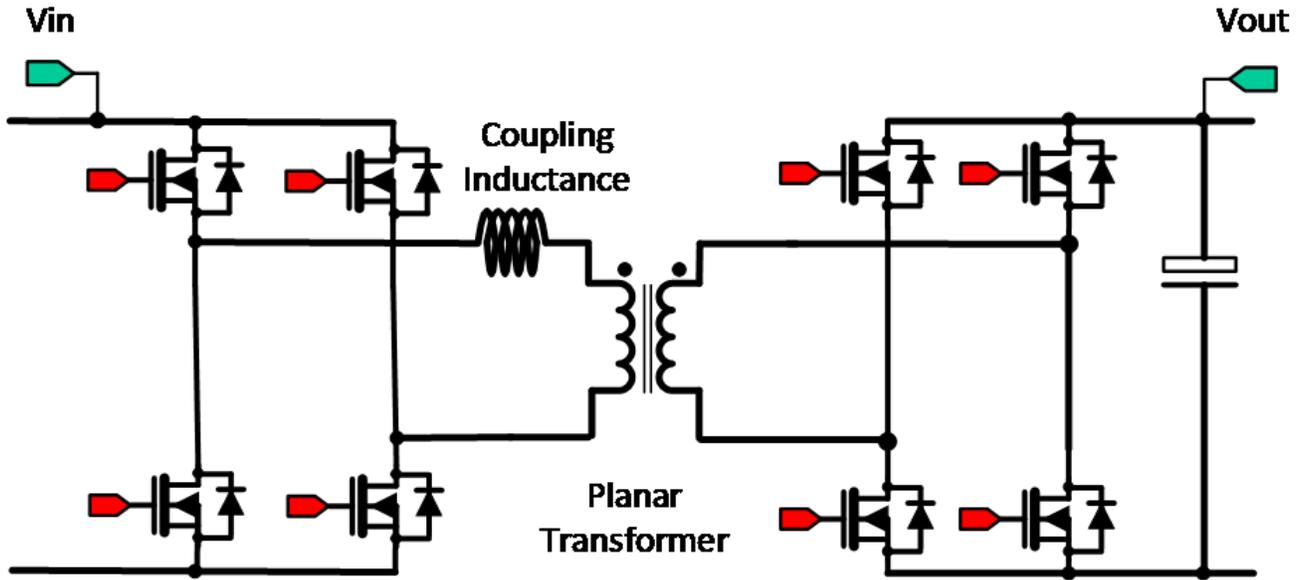


图 4-7. 双有源电桥方框图

### 4.5.2 深入介绍

单相移 DAB 中的功率传输与耦合电感 (L) 值以及开关频率 (Fs) 和相位 (θ) 有关。

$$P_{output} = \frac{nV_{in}V_{out}\theta(\pi - \theta)}{2\pi^2 F_s L}$$

其中

- n 是变压器转数
- Vin 是输入到 DAB 的电压
- Vout 是 DAB 输出的电压
- θ 是相位
- Fs 是开关频率
- L 是电感

为了提高功率密度（减小设计的整体物理尺寸），既需要较低的电感，又需要较高的开关频率。一个参考示例是 [TIDA-010054](#)，它是一个 10kW 直流/直流 DAB 转换器：对于漏电指示器，使用 100kHz PWM 开关频率和 35 μH。采用 800V 总线的三相 PFC，例如 [Vienna 整流器 \(TIDM-1000\)](#) 或 [T-Type PFC \(TIDA-010039\)](#) 用作前端。最后，对于 10kW 下的 400V 输出电压，我们可以确定相移引起的电压输出阶跃，并检查是否允许其在我们的纹波要求的 ±1% 范围内。

如表 4-3 所示，通过使用时钟频率为 MCU 时钟速率典型 PWM 生成逻辑，无法将电压调节至约 400V 的 1% 精度。假设 CPU 时钟为 100MHz，则将导致 10ns 的阶跃变化，这对于我们的输出纹波容差来说太大了。使用 HRPWM 模块，无论主 MCU 时钟如何，都可以将相移调制到 150ps 以内。更重要的是，超出相位步长要求表示能够在这些范围内以 13 倍步长进行调制，从而能够调整输出以使其与理想值更接近。

**表 4-3. 相移要求满足 DAB 拓扑中 1% 的输出容差**

Vin=800V , Pout=10kW , Fs=100kHz	输出电压 (V)	所需相移 (ns)
$\theta = 19.032$ 度	400	528.66
$\theta = 18.9249$ 度	402	525.69
$\theta = 18.7662$ 度	405	521.11
$\theta = 18.5077$ 度	408	517.10

#### 4.5.3 器件列表

- [TMS320F2838xD/S](#)
- [TMS320F2837xD/S](#)
- [TMS320F2807x](#)
- [TMS320F28004x](#)
- [TMS320F28003x](#)
- [TMS320F28002x](#)
- [TMS320F280013x](#)
- [TMS320F280015x](#)

#### 4.5.4 硬件平台和软件示例

- [TIDA-010054](#)，适用于 3 级电动汽车充电站的双向双有源电桥参考设计
- [TIDM-1000](#) 基于 Vienna 整流器的三相功率因数校正参考设计
- [TIDM-1000 Digital Power SDK](#) 中的基准软件工程
- [TIDA-010039](#) 三级三相 SiC 交流/直流转换器参考设计

#### 4.5.5 文档

- [TMS320F2838xD 实时微控制器技术参考手册](#) (更多信息，请参阅 [HRPWM](#) 一章)
- [TMS320F2837xD 实时微控制器技术参考手册](#) (更多信息，请参阅 [HRPWM](#) 一章)
- [TMS320F28004x 实时微控制器技术参考手册](#) (更多信息，请参阅 [HRPWM](#) 一章)
- [TMS320F28003x 实时微控制器技术参考手册](#) (更多信息，请参阅 [HRPWM](#) 一章)
- [TMS320F28002x 实时微控制器技术参考手册](#) (更多信息，请参阅 [HRPWM](#) 一章)
- [TMS320F280013x 实时微控制器技术参考手册](#) (更多信息，请参阅 [HRPWM](#) 一章)
- [TMS320F280015x 实时微控制器技术参考手册](#) (更多信息，请参阅 [HRPWM](#) 一章)

## 4.6 高频、多相和变频拓扑中的安全和优化的 PWM 更新

### 4.6.1 价值主张

全局加载和全局链接特性简化了需要主动管理多个 PWM 参数（例如变频和多相电源拓扑）的应用。更具体地说，全局加载特性通过提供单个选通信号，在单个 PWM 模块内或跨多个模块将 PWM 参数从影子寄存器加载到活动寄存器，从而帮助消除 PWM 干扰。通过允许跨多个 PWM 模块同时更新链接的寄存器，全局链接特性降低了 CPU 开销。

### 4.6.2 深入介绍

影子寄存器为活动寄存器提供了一个临时位置，发生加载事件时，其内容便会在 PWM 周期内的关键点传输到活动寄存器。这种影子到活动加载特性可防止寄存器被软件异步修改导致损坏和虚假操作。过去，使内容从影子寄存器传输到活动寄存器的加载事件是为每个寄存器单独配置的。然而，当全局加载模式启用 ( $GLDCTL[GLD]=1$ ) 后，对于启用该模式的所有寄存器，将由同一事件 ( $GLDCTL[GLDMODE]$ ) 触发从影子寄存器到活动寄存器的内容传输。可通过  $GLDCFG$  寄存器启用子模块（例如时基、动作限定符和死区）的关键 PWM 寄存器，以便使用全局加载功能。此外，在更新活动寄存器  $GLDCTL[GLDPRD]$  之前，还可配置需要发生的加载选通事件的数量，从而满足应用需求。

如果控制 ISR 与 PWM 开关频率异步，则还需要使用一次性加载模式特性，确保 PWM 模块中的所有寄存器仅在所有必需的 PWM 参数都更新后才更新。通过  $GLDCTL[OSHTMODE]$  寄存器启用一次性加载模式。当  $GLDCTL2[OSHTLD]$  设置为 1 时，对于配置为使用全局加载机制的寄存器，从影子寄存器到活动寄存器的内容传输将在  $GLDCTL[GLDMODE]$  选择的下一事件中发生。在影子到活动加载事件发生后，全局加载事件将被阻止，直到  $GLDCTL2[OSHTLD]$  再次设置为 1。图 4-8 是一次性加载模式操作的示例。

对于变频应用，需要在 ePWM 模块之间同时写入周期寄存器和比较寄存器。使用  $EPWMXLINK$  寄存器，可以通过在多个 PWM 模块之间创建链接来更新周期寄存器和比较寄存器。为了在此基础上进一步扩展，也可以在模块之间链接全局加载模式寄存器  $GLDCTL2$ ，该寄存器允许重新加载或强制执行一次性加载模式。这创建了一种方案，使周期寄存器和比较寄存器能够在多个 PWM 模块中以相同的值同时更新。

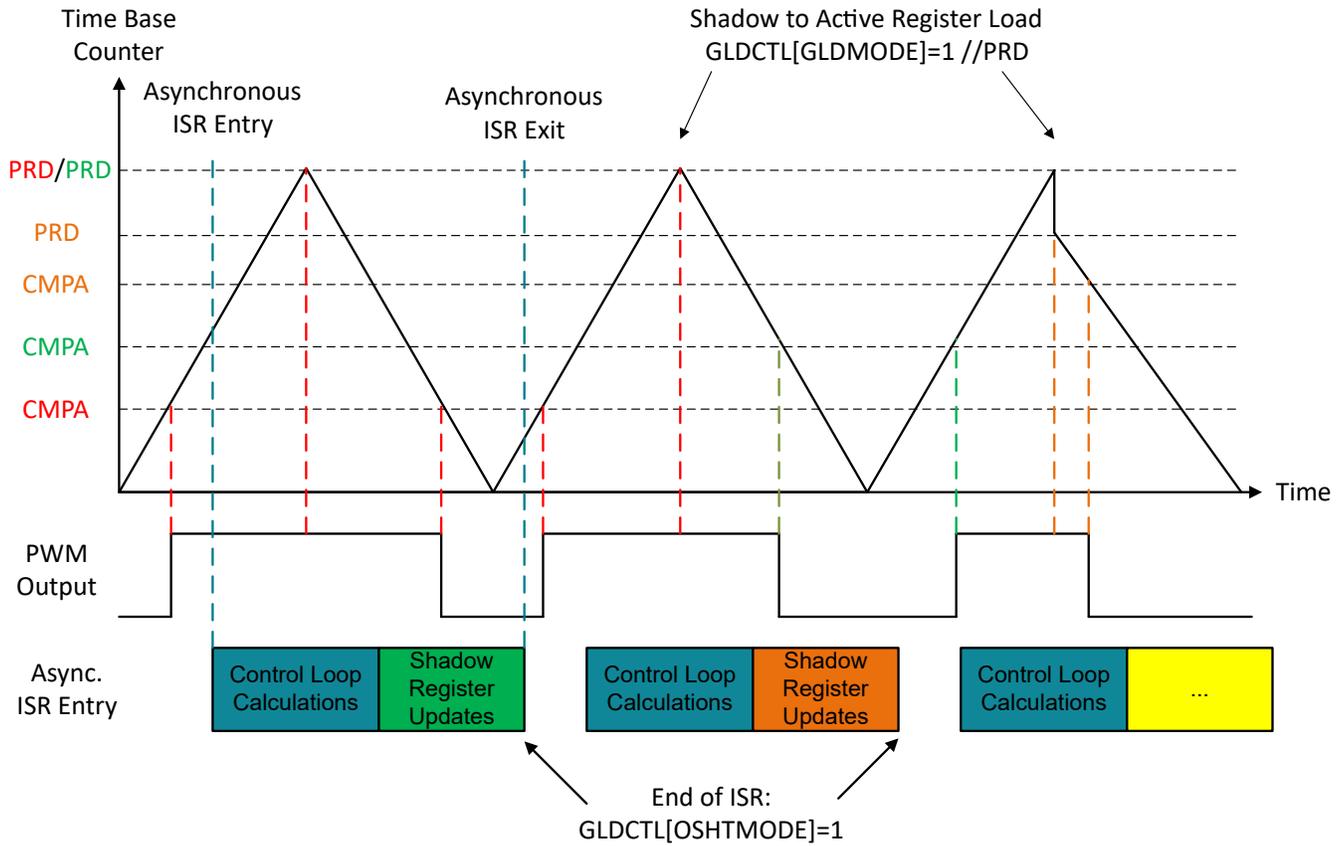


图 4-8. 影子到活动加载操作

#### 4.6.3 器件列表

- TMS320F2838xD/S
- TMS320F2837xD/S
- TMS320F2807x
- TMS320F28004x
- TMS320F28003x
- TMS320F28002x
- TMS320F280013x
- TMS320F280015x

#### 4.6.4 硬件平台和软件示例

- TIDM-1001 使用 C2000 MCU 的两相交错式 LLC 谐振转换器参考设计 (全局加载)
- TIDM-02002 HEV/EV 车载充电器的双向 CLLC 谐振双有源电桥参考设计 (全局链接)

#### 4.6.5 文档

- 利用新型 ePWM 特性进行多相控制
- ePWM 的“全局加载”部分和 TRM 的“时基 (TB) 子模块”部分

## 4.7 解决分散型控制系统中跨多个控制器的事件同步问题

### 4.7.1 价值主张

工业和汽车实时控制应用中使用的分散型架构需要一种方法来同步某些事件，同时将器件之间的触发延迟和抖动降至最低。FSI 是一种高吞吐量、低延迟的有线通信外设，可在多器件网络拓扑中实现不超过约 100ns 的时间同步事件触发。使用现有的通信端口（如 FSI）进行系统事件同步，无需在器件之间设置额外的同步信号，从而节省器件资源和额外的设计/BOM 成本。

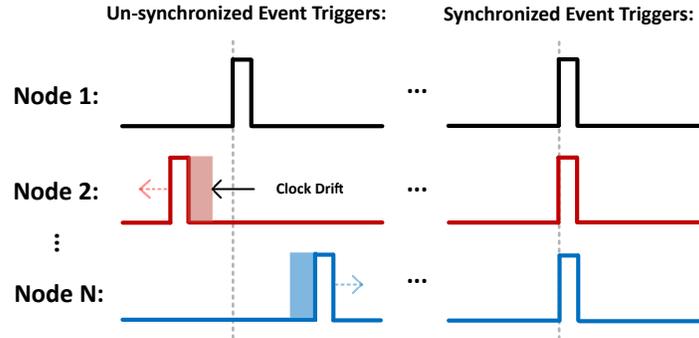


图 4-9. 非同步与同步事件触发

### 4.7.2 深入介绍

在分散型架构中，器件之间的通信接口在网络拓扑、节点器件数量、节点之间的物理距离等方面可能有所不同。由于制造不确定性、热效应、老化等原因，器件本地时钟在运行期间也可能略有偏差。无论这些变化如何，系统中所有器件同时发生特定事件通常很重要，例如 ADC 转换启动或 PWM 信号的上升/下降沿。可以使用 FSI 模块和自定义 CLB 逻辑来实现 C2000 器件之间的同步事件。

以跨器件同步等效 PWM 信号的目标为例，可以进一步探索实现方法，如图 4-10 所示。网络中的主控器件将定期向所有节点器件发送 PWM 同步请求，作为 FSI PING 帧。在菊花链拓扑中，每个节点器件将同步请求帧转发到链中的下一个器件。当节点器件接收到同步请求帧时，CLB 模块将在连接到 ePWM 的 EPWMSYNCIN 信号之前，通过一个可配置延迟在内部路由来自 FSI RX 模块的 Ping 数据包接收 (PING\_PKT\_RCVD) 信号。可配置延迟会进行校准，以便同时触发节点器件的所有 EPWMSYNCIN 信号，并匹配主控器件的 TBCTR = 0 或 PRD 事件。该实现完全基于硬件，初始化后不需要软件干预。在菊花链拓扑中，由于同步请求帧需要一些时间才能到达所有器件，与链末端的节点相比，链开头的节点需要更长的可配置延迟。这将确保主控器件和所有节点器件的 PWM 信号在运行期间保持同步。

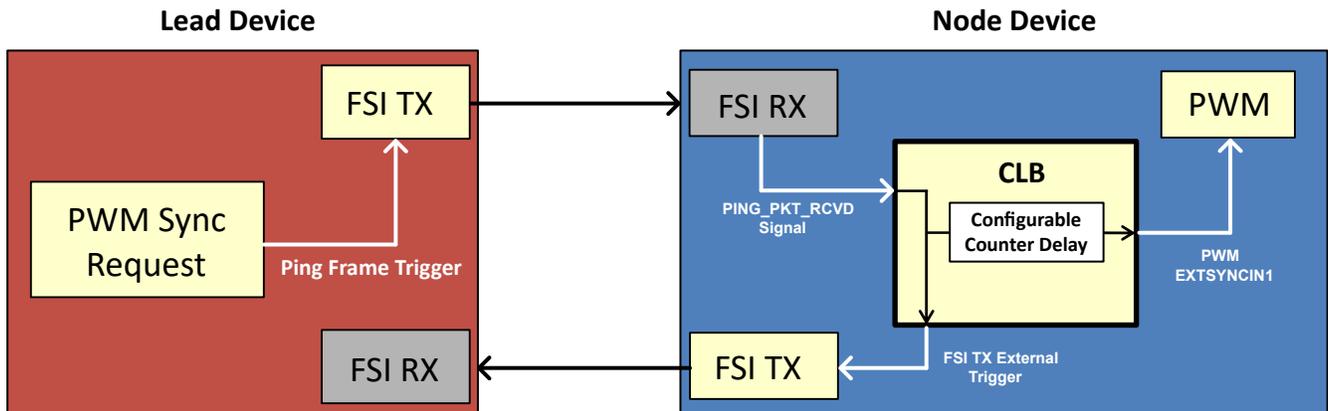


图 4-10. 主控器件和节点器件实现

图 4-11 演示了 N 节点菊花链网络的 PWM 同步序列和时序。由于器件和物理层的不确定性，每个节点的 PWM 边沿都会出现一些非常小的抖动，但随着菊花链网络中节点器件的不断增多，抖动会越来越剧烈。对于其他不需要帧转发的网络拓扑（例如星形拓扑或总线拓扑），抖动会更小。

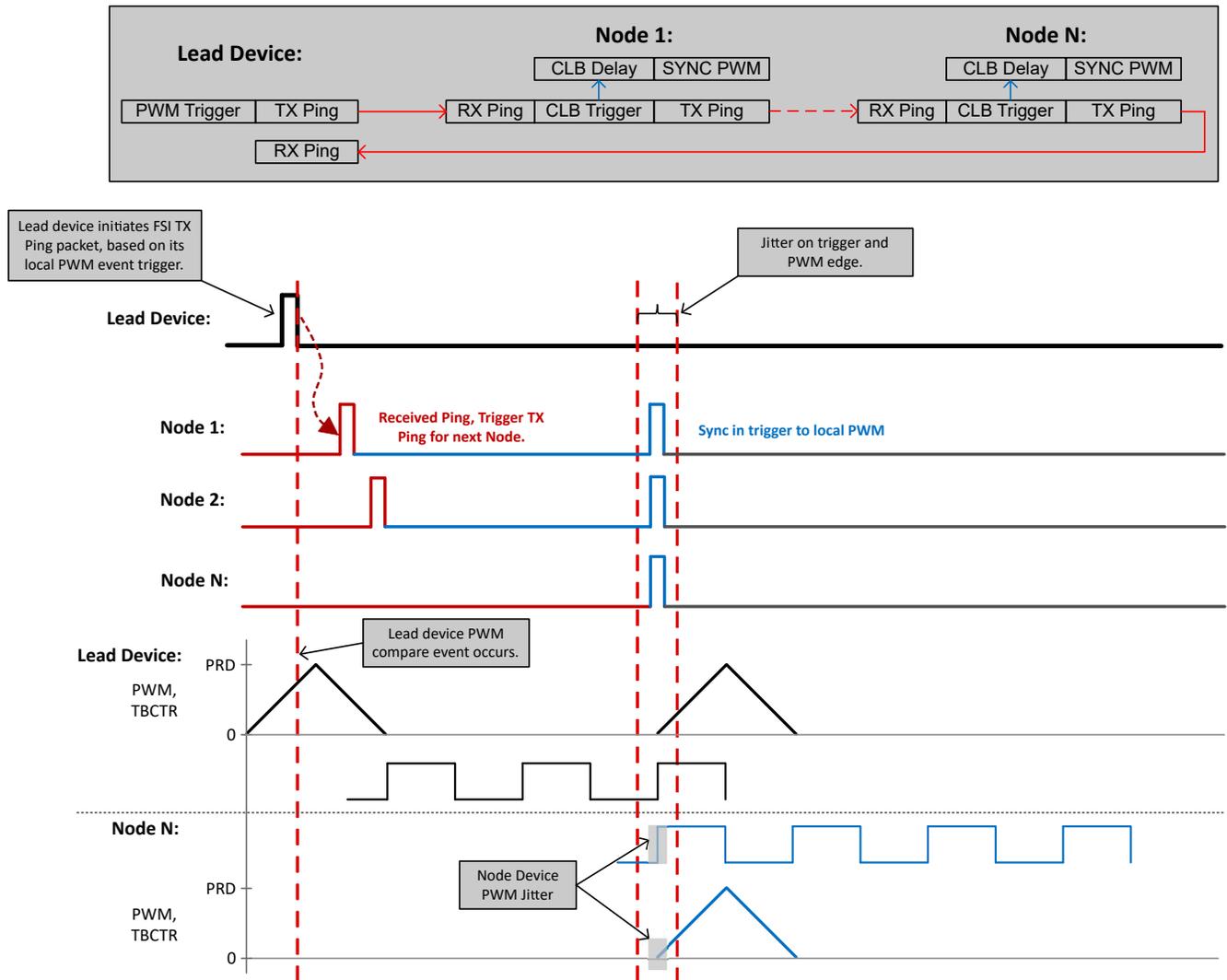


图 4-11. 菊花链网络同步和时序

有关通过 FSI 进行事件同步的更详细说明以及测试结果，请参阅 [快速串行接口 \(FSI\) 在多芯片互连中的应用](#) 中 [通过 FSI 进行事件同步](#) 部分。

#### 4.7.3 器件列表

- [TMS320F2838xD/S](#)
- [TMS320F28004x](#)
- [TMS320F28003x](#)
- [TMS320F28002x](#)

#### 4.7.4 硬件平台和软件示例

- [TMDSFSIADAEVM](#)
- [FSI 菊花链 ePWM 同步引线](#)
- [FSI 菊花链 ePWM 同步节点](#)

#### 4.7.5 文档

- [将快速串行接口 \(FSI\) 应用于应用中的多个器件](#)

## 5 接口关键技术

### 5.1 对 C2000 外设的直接主机控制

#### 5.1.1 价值主张

许多工业控制系统都实现一个主机控制器，该控制器可以监视系统的多个方面，包括在实现直接控制回路时与 C2000 实时 MCU 进行接口连接。在其他情况下，可能希望将 C2000 的差异化 IP 添加到现有系统中。C2000 实时 MCU 内的主机接口控制器 (HIC) 逻辑允许使用通用异步接口控制各种外设。

通过 HIC，主机可以利用 C2000 的不同功能，例如快速串行接口 (FSI)、可配置逻辑块 (CLB) 等，通过 ASRAM 端口将 C2000 器件用作外围扩展器。

#### 5.1.2 深入介绍

HIC 支持主机可与之连接的各种配置。每种配置在延迟、输入/输出 (IO) 引脚开销等方面都具有一定的优势和影响。

HIC 允许访问许多常见的 C2000 MCU 外围设备，例如但不限于：模数转换器、比较器、PWM、CAN 控制器、快速串行接口 (FSI) 通信和可配置逻辑块 (CLB)。

有关 HIC 功能的完整概述，请参阅 [使用 HIC 启用外设扩展应用的设计指南](#)。

以下部分详细介绍了使用 FSI 进行隔离通信以及使用 CLB 来实现绝对编码器的 HIC 的一些实际应用。

### 5.1.2.1 适用于 FSI 应用的 HIC 桥

快速串行接口 (FSI) 能够在系统中支持低延迟和稳健的带隔离高速通信。图 5-1 展示了具有异步 RAM 接口的主机如何连接 C2000 芯片，以使用 FSI 实现带隔离高速通信。

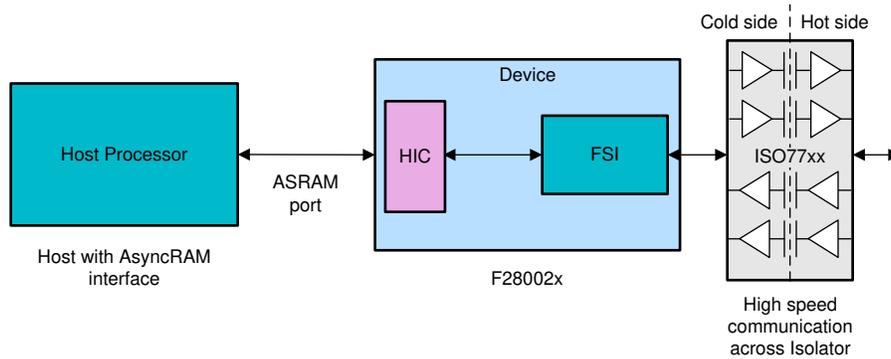


图 5-1. 适用于 FSI 应用的 HIC 桥

### 5.1.2.2 使用 CLB 的位置编码器应用的 HIC 桥

CLB 支持自定义逻辑实现，并扩展了现有的 C2000 外设集，从而不需要使用 FPGA、CPLD 或外部逻辑组件实现通用的功能，或减少对这些组件的需求。CLB 可以实现许多解决方案，其中一种是与工业驱动控制系统中的位置传感器连接的绝对编码器协议实现 (图 5-2)。有关受 CLB 支持的编码器协议，请参阅 C2000 网站上的“位置管理器技术”部分。HIC 可用作主机与位置传感器连接的桥梁，如图 5-2 所示。

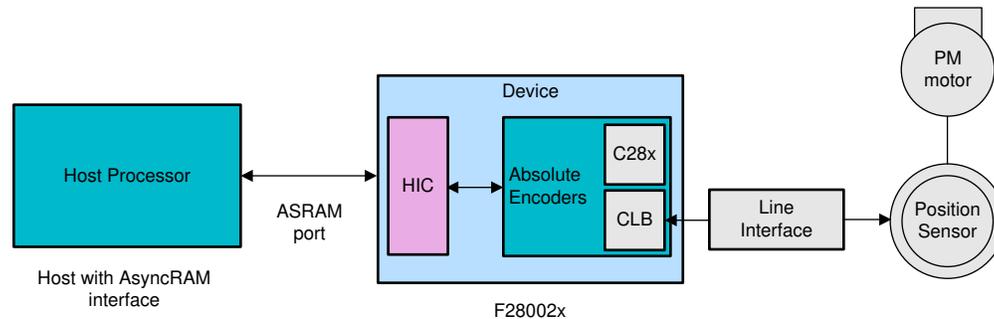


图 5-2. 适用于位置编码器应用的 HIC 桥

### 5.1.3 器件列表

- [TMS320F28003x](#)
- [TMS320F28002x](#)

### 5.1.4 硬件平台和软件示例

- [C2000Ware 中的 16 位 FSI Bridge 示例](#)
- [访问 HIC 的 16 位 EMIF 示例](#)
- [C2000Ware 中的 8 位示例](#)
- [使用 HIC 访问 ADC 的 8 位 EMIF 示例](#)

### 5.1.5 文档

- [使用 HIC 启用外设扩展应用的设计指南](#)
- [TMS320F28003x 微控制器技术参考手册](#)
- [TMS320F28003x 微控制器数据表](#)
- [TMS320F28002x 微控制器技术参考手册](#)
- [TMS320F28002x 微控制器数据表](#)

## 5.2 通过 AES 引擎保护外部通信和固件更新

### 5.2.1 价值主张

传入命令（例如固件更新）的完整性和机密性对于保护系统至关重要。在软件中执行加密算法很慢，并且需要一定水平的专业知识。TI C2000 产品已在硬件中实现了专用的高级加密标准 (AES) 引擎，以减少与消息和数据的加密、解密、标记及身份验证相关的开销。

### 5.2.2 深入介绍

TI C2000 产品上的高级加密标准 (AES) 硬件加速器能够以常见的加密/解密、身份验证以及带有关联数据的身份验证加密 (AEAD) 模式运行。它支持 128 位、192 位和 256 位的密钥大小，根据具体操作和密钥大小，吞吐量上限为 4 位/周期。利用 DMA，引擎需要很少的 CPU 干预，从而使您可以管理通信的安全性，同时使用 CPU 执行其他重要任务。

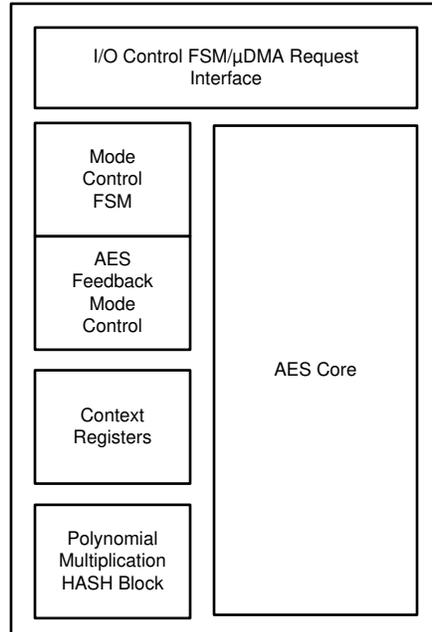


图 5-3. AES 方框图

仅就消息机密性而言，支持以下加密/解密模式：

- 电码本 (ECB)
- 密码分组链接 (CBC)
- 计数器 (CTR) 或整数计数器 (ICM)
- 密码反馈 (CFB)
- 密文窃取 (XTS)
- F8

仅就消息完整性而言，支持以下身份验证模式：

- 密码分组链接消息身份验证代码 (CBC\_MAC)
- F9

如果同时要求消息的机密性和完整性，则支持以下带有关联数据的身份验证加密 (AEAD) 模式：

- 伽罗瓦计数器模式 (GCM)
- 带 CBC-MAC 的计数器模式 (CCM)

### 5.2.3 器件列表

- [TMS320F2838xD/S](#)
- [TMS320F28003x](#)

### 5.2.4 硬件平台和软件示例

- TMDSCNCD28388D 控制卡
- AES ECB 加密
- AES ECB 解密
- AES GCM 加密
- AES GCM 解密

### 5.2.5 文档

- TMS320F2838x 微控制器技术参考手册 (更多信息, 请参阅 AES 一章)
- TMS320F28003x 微控制器技术参考手册 (更多信息, 请参阅 AES 一章)

## 5.3 跨隔离边界的分布式实时控制

### 5.3.1 价值主张

许多实时控制系统都依赖于外部通信来获取重要的系统信息, 这些信息既可以作为控制环路的输入, 也可以作为系统的监测器。由于这些系统中任何操作都需要限时完成的特点, 因此绝对对速度和数据完整性都很重要。与其他串行通信外设相比, C2000 MCU 的快速串行接口 (FSI) 满足了这一需求, 提供了高达 200Mbps 的吞吐量, 以及硬件中的额外功能, 这些功能仅需几根导线即可增加数据完整性。

### 5.3.2 深入介绍

设计多器件实时控制系统时, 有许多串行通信外设可供选择。由于处理器需要在非常短的时间内在彼此之间传递关键数据, 因此延迟是系统设计人员的主要考虑因素。

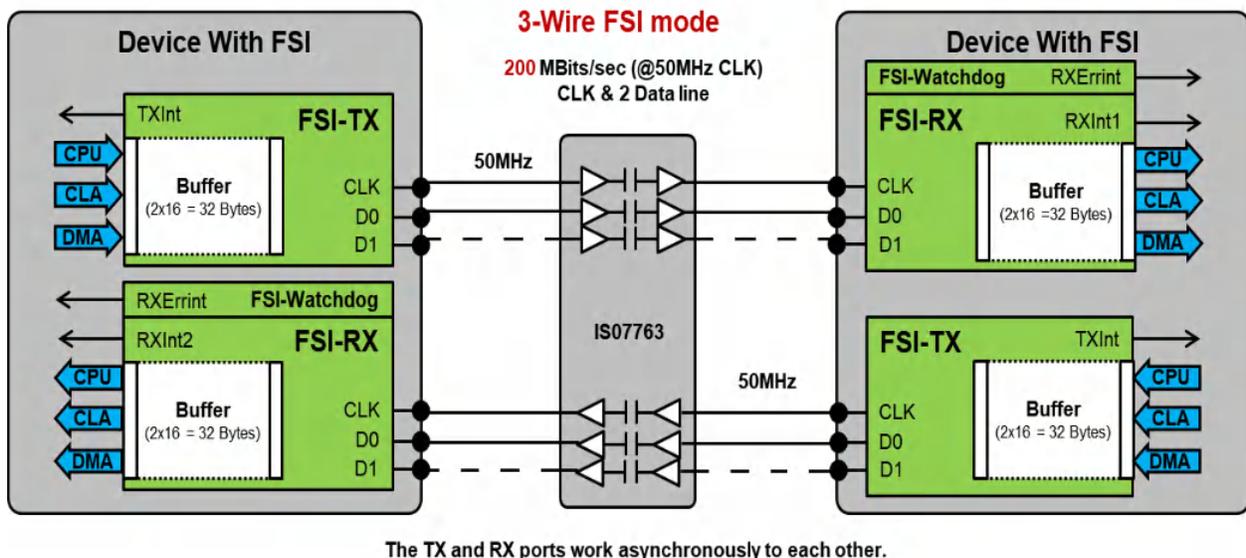


图 5-4. 全双工 3 线 FSI 实现

#### 备注

单一数据 (D0) 和时钟 (CLK) 是以 100Mbps 速率进行 FSI 通信的最小信号集, 具有最低的信号计数和隔离成本。图 5-4 显示, 信号 D1 是用于实现 FSI 的完整 200Mbps 数据速率的可选数据线。

FSI 物理接口由三根导线、一个时钟和两个数据信号组成，其中一个数据信号是可选的（请参见图 5-4）。数据在上升沿和下降沿上传输，这允许最大 50MHz FSI 时钟频率通过两条导线（CLK 和 D0）以 100Mbps 的速率，以及通过三条导线（CLK、D0 和 D1）以 200Mbps 的速率传输数据。高吞吐量以及包含有限的报头和报尾的定义数据包（帧），使得数据可以在器件之间以极小的延迟进行传输。FSI 模块由独立的发送器和接收器内核组成，它们允许在两个方向上同时进行全速通信，而不需要主从器件。在[基于快速串行接口 \(FSI\) 的分布式多轴伺服驱动参考设计](#)中，介绍了一种使用 FSI 进行分布式控制的实时系统。

FSI 在其他常用通信外设上提供的功能包括：

- 在发送器和接收器侧均采用硬件实现的 CRC，消除了软件实现的 CPU 开销
- 在接收器模块上进行延迟线控制，以补偿不同通道之间的延迟不一致
- 使用 ping 和数据帧看门狗进行断行检测
- FSI 协议没有主从概念，使得器件可以随时发送反馈而无需主器件发出请求
- 高比特率和低信号计数减少了系统中所需的隔离器数量

有许多系统拓扑结构，它们的器件必须在系统的“热”（高压）侧和“冷”（低压）侧上运行，并且必须彼此通信。在这种情况下，需使用数字隔离器将数据带过隔离栅，而跨越隔离边界的信号之间的潜在延迟可能很难在许多生产单元中预测。即使在没有隔离的系统中，也可能会由于不相等的信号布线长度而产生延迟。

接收器处的延迟线控制功能使 FSI 非常适合此应用，因为它可以通过为各个 FSI 信号增加延迟来主动补偿这种偏斜（图 5-5）。有关此差异功能的更多信息，请参阅[快速串行接口 \(FSI\) 偏斜补偿](#)。另请参阅 [TMDSFSIADAPEVM](#)，了解带数字隔离器的 FSI 的硬件评估。

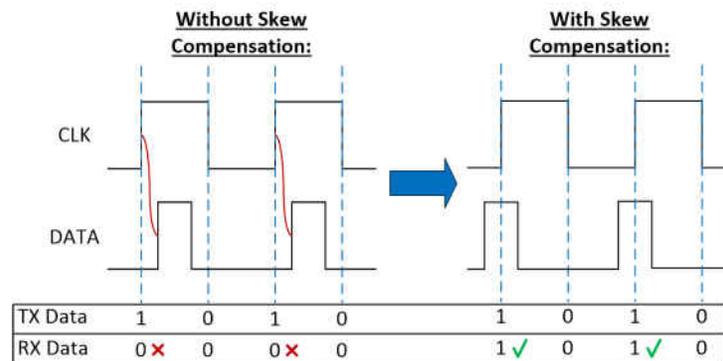


图 5-5. FSI 偏斜补偿

#### 备注

数据始终是在 FSI 时钟的上升沿和下降沿发送和接收。

尽管图 5-5 中仅显示一条数据线，但在所有 FSI 实现中都有第二条数据线选项

### 5.3.3 器件列表

- [TMS320F2838xD/S](#)
- [TMS320F28004x](#)
- [TMS320F28003x](#)
- [TMS320F28002x](#)

### 5.3.4 硬件平台和软件示例

- [TMDSFSIADAPEVM](#)
- [LAUNCHXL-F280049C LaunchPad](#)
- [LAUNCHXL-F280039C LaunchPad](#)
- [LAUNCHXL-F280025C LaunchPad](#)
- [TIDM-02006](#)

### 5.3.5 文档

- [快速串行接口 \(FSI\) 偏斜补偿](#)
- [快速串行接口 \(FSI\) 在多芯片互连中的应用](#)

## 5.4 使用嵌入式图形发生器 (EPG) 进行自定义测试和数据图形生成

### 5.4.1 价值主张

独特的图形生成可以是对实时系统的有用补充，既可用于自测目的，也可作为系统中其他 IC 的保持活动信号。如果可能，通常通过手动切换 GPIO 的 SW 例程来实现独特的图形生成。EPG 模块消除了这些应用中涉及的 CPU 开销，并为可生成的图形提供更大的灵活性。

### 5.4.2 深入介绍

嵌入式图形发生器 (EPG) 模块是可定制的图形和时钟发生器，可以服务于许多需要简单图形发生器或周期性时钟发生器的测试和应用场景。EPG 模块还可用于采集传入的串行数据流。EPG 模块允许用户设计新的时钟发生器、脉宽调制器 (PWM) 和串行通信模块等。

EPG 模块可用作独立的外设使用，也可与 CAN 等其他外设一起使用。当与 CAN 等其他外设一起使用时，EPG 可以为 C2000™ 器件内的测试和运行诊断提供信号图形。

当用于生成时钟时，EPG 可以生成：

- 独立的时钟生成和时钟分频
- 具有可编程偏移的同步时钟生成 (图 5-6)

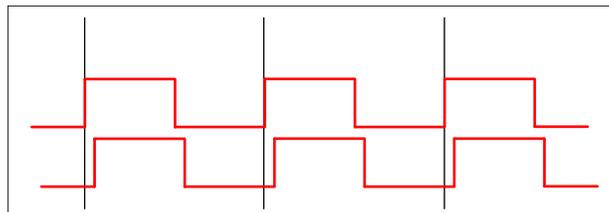


图 5-6. 带偏移的时钟

当用于图形生成时，EPG 可以生成：

- 独立的串行数据流
- 串行数据流和相关时钟 (图 5-7)
- 相对于串行数据的时钟偏移
- 相互间具有可编程偏移的同步数据流

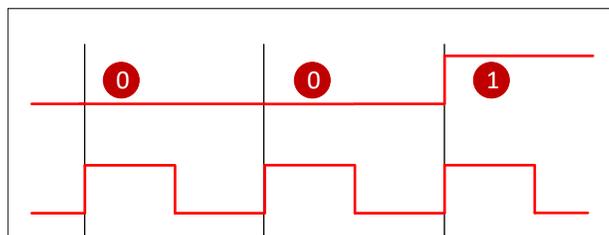


图 5-7. 串行流和时钟

### 5.4.3 器件列表

- TMS320F28003x
- TMS320F280013x
- TMS320F280015x

### 5.4.4 硬件平台和软件示例

- 生成时钟的 C2000Ware EPG 示例
- 串行数据的 C2000Ware EPG 示例
- TMS320F280039C LaunchPad
- TMS320F2800137 LaunchPad
- TMDSCNPDF2800157

### 5.4.5 文档

- 使用 C2000™ 嵌入式图形发生器 (EPG) 进行设计
- TMS320F28003x 实时微控制器 TRM
- TMS320F280013x 实时微控制器 TRM
- TMS320F280015x 实时微控制器 TRM

## 6 安全关键技术

### 6.1 作为控制环路一部分的非侵入式运行时间监测和诊断

#### 6.1.1 价值主张

在评估控制系统的性能时，CPU 资源管理通常是一个关键问题。了解不同功能所需的带宽不仅在系统开发过程中很重要，在系统最终使用时也很重要。ERAD ( 嵌入式实时和诊断 ) 模块在 C2000 MCU 上提供了此功能，减少了开发过程中花费的时间，还确保了最终应用的稳定和安全性能。

#### 6.1.2 深入介绍

ERAD 是具有增强总线比较器和系统事件计数器的硬件模块，位于 MCU 总线架构内，如图 6-1 所示。ERAD 本身可以生成系统级中断和标志，还可以馈入其他外围设备 ( 如 CLB ) 以进一步增强功能。要完全了解这两个模块之间的关系，请参阅 CLB 文档。

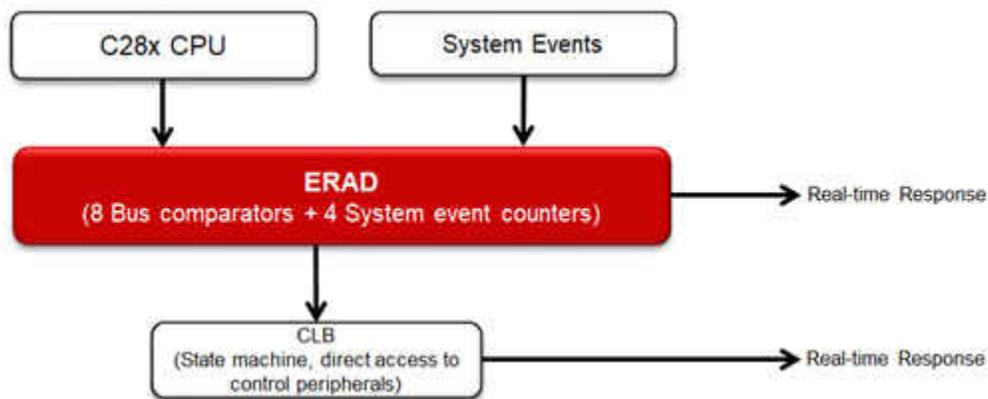


图 6-1. ERAD 方框图

增强型总线比较器监视一些关键的 CPU 内部总线和信号，这些总线传达有关 CPU 代码执行和流水线的信息。EBC 单元监视大多数关键 CPU 接口，可以进行配置以监测 CPU 接口上任何地址/数据/指令的出现并产生相应的事件。一个简单的例子是当访问目标地址并进行读取或写入时，可以产生事件。此外，可以将这些事件导出到 CLB 以定义状态机，从而跟踪这些事件的顺序。

另一方面，系统事件计数器会监视系统中的各种事件，例如中断和外围活动，这是对于任何进行监视和分析的实时系统都是一个重要的需求。所有关键系统事件（如中断、DMA 触发器）和重要的外围事件（通过中断、CLA 事件等）都可以在 ERAD 内部进行概要分析。因此，可以相对于 CPU 活动为它们添加时间戳、进行计数和度量。增强的计数器块还允许参考其他事件来测量和分析系统事件以及 CPU 活动。

### 示例：栈溢出检测

嵌入式代码开发中经常遇到的问题是检测栈溢出。ERAD 可以只使用一个增强型总线比较器单元，然后以很小的余量映射栈结束地址，从而在尝试进行写操作时产生中断。对于以下示例（图 6-2），只需使用地址为 0x99FF0 的比较器并屏蔽后 4 位进行比较，因此地址 0x99FF0 至 0x99FFF 之间的任何写访问都会产生事件/中断，然后可以根据应用的类型进行适当的处理。

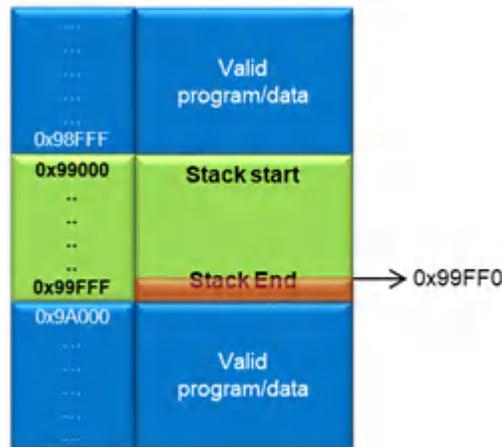


图 6-2. 栈溢出保护

### 6.1.3 器件列表

- [TMS320F2838xD/S](#)
- [TMS320F28004x](#)
- [TMS320F28003x](#)
- [TMS320F28002x](#)

### 6.1.4 硬件平台和软件示例

- [F28388D controlCARD 评估模块](#)
- [C2000 MCU F280049C LaunchPad™ 开发套件](#)
- [F280039C controlCARD 评估模块](#)
- [F280025 controlCARD 评估模块](#)

### 6.1.5 文档

- [ERAD 简介视频](#)
- [适合控制应用的嵌入式实时分析和响应](#)
- [TMS320F2838x 实时微控制器 TRM](#)
- [TMS320F28004x 实时微控制器 TRM](#)
- [TMS320F28003x 实时微控制器 TRM](#)
- [TMS320F28002x 实时微控制器 TRM](#)

## 6.2 C28x CPU 硬件内置自检

### 6.2.1 价值主张

安全应用需要对关键器件组件进行启动和运行时诊断。主 CPU 最为关键；对于 C2000 器件，C28x 实时内核最为关键。此外，如果没有对 CPU 的深入了解，用户就不能在软件中进行上述诊断。受支持的 C2000 MCU 上有一个逻辑块来自动执行这些测试，可达到高水平的诊断覆盖率，从而满足 ISO 26262 或 IEC 61508 等安全标准。

### 6.2.2 深入介绍

启动和应用期间，硬件内置自检 (HWBIST) 对 CPU (包括 FPU、TMU 和 VCU) 提供了一个非常高的晶体管级诊断覆盖。为了快速执行高质量的制造测试，这个逻辑电路采用插入器件的可测性设计 (DfT) 结构，但是使用的是一个内部测试引擎而非外部自动测试设备 (ATE)。事实证明，这项技术可以有效地在较短时间内提供高覆盖率。

HWBIST 测试由软件触发。用户可以选择运行所有测试，或根据为诊断分配的执行时间只运行这些测试的一个子集。这个时间分片测试特性使得 HWBIST 能够高效进行运行时诊断，与应用程序并行执行测试。HWBIST 执行失败 (例如检测到逻辑故障或自检未完成就超时) 会触发 NMI。HWBIST 提供的最大诊断覆盖率取决于器件，如 [HWBIST 支持的诊断覆盖率 \(因器件而异\)](#) 所示。

表 6-1. HWBIST 支持的诊断覆盖率 (因器件而异)

F2838xD/S	F2837xD/S	F2807x	F28002x
> 99% 直流	> 99% 直流	> 99% 直流	> 99% 直流

若要配置和运行 HWBIST，需要使用由 C2000 诊断软件库提供的 HWBIST API。提供的功能可以运行一个超短期或完整的 HWBIST 测试，以及在 HWBIST 中配置自检模式，从而允许注入几种不同类型的故障来检查其是否正常运行。

### 6.2.3 器件列表

- [TMS320F2838xD/S](#)
- [TMS320F2837xD/S](#)
- [TMS320F2807x](#)
- [TMS320F28003x](#)
- [TMS320F28002x](#)

### 6.2.4 硬件平台和软件示例

- [C2000Ware 中的 C2000 诊断软件库 \(F28002x、F2838x\)](#)
- [C2000 SafeTI™ 诊断软件库 \(F2837xD/S、F2807x\)](#)

### 6.2.5 文档

- [TMS320F28002x 安全手册](#)
- [TMS320F2837xD、TMS320F2837xS 和 TMS320F2807x 安全手册](#)
- [C2000™ 硬件内置自检](#)

## 6.3 嵌入式片上存储器的零 CPU 开销循环冗余校验

### 6.3.1 价值主张

在任何控制系统的开发过程中，都需要非常仔细地理解将由器件执行的代码的交互和时序。然而，无论之前如何努力，如果该代码被无意修改或破坏，系统的行为就无法再得到保证。许多系统（特别是那些与安全有关的系统）会定期检查存储器的好坏，但这以牺牲 CPU 和系统带宽为代价。C2000 MCU 上的背景 CRC (BGCRC) 模块解决了这个问题，实现了一个 CRC 引擎，可以检查存储器完整性，而不会影响 CPU 开销或系统性能。

### 6.3.2 深入介绍

C2000 MCU 中的存储块受奇偶校验或 ECC 保护，这有助于检测存储器是否损坏。但奇偶校验/ECC 特性只有在应用程序对存储器进行读取访问时才能检测到错误。因此，定期回读存储器以提前检测到此类问题非常重要。为存储器添加 CRC 校验可确认存储器内容未更改。

C2000 MCU 中的 BGCRC 模块可用于在后台执行存储器回读，因此在不占用任何 CPU 周期的情况下，确保了应用程序的不间断执行。一旦触发，它会在后台读取存储器内容，计算 CRC，并将其与编程的黄金值进行比较，如果不匹配则标记错误。它还支持清理模式，在这种模式下，它不执行 CRC 比较，但会检查奇偶校验/ECC 错误。读取发生在空闲时间（此时 CPU、CLA、DMA 等其他存储器所有者均未访问存储块），因此功能访问不受影响。

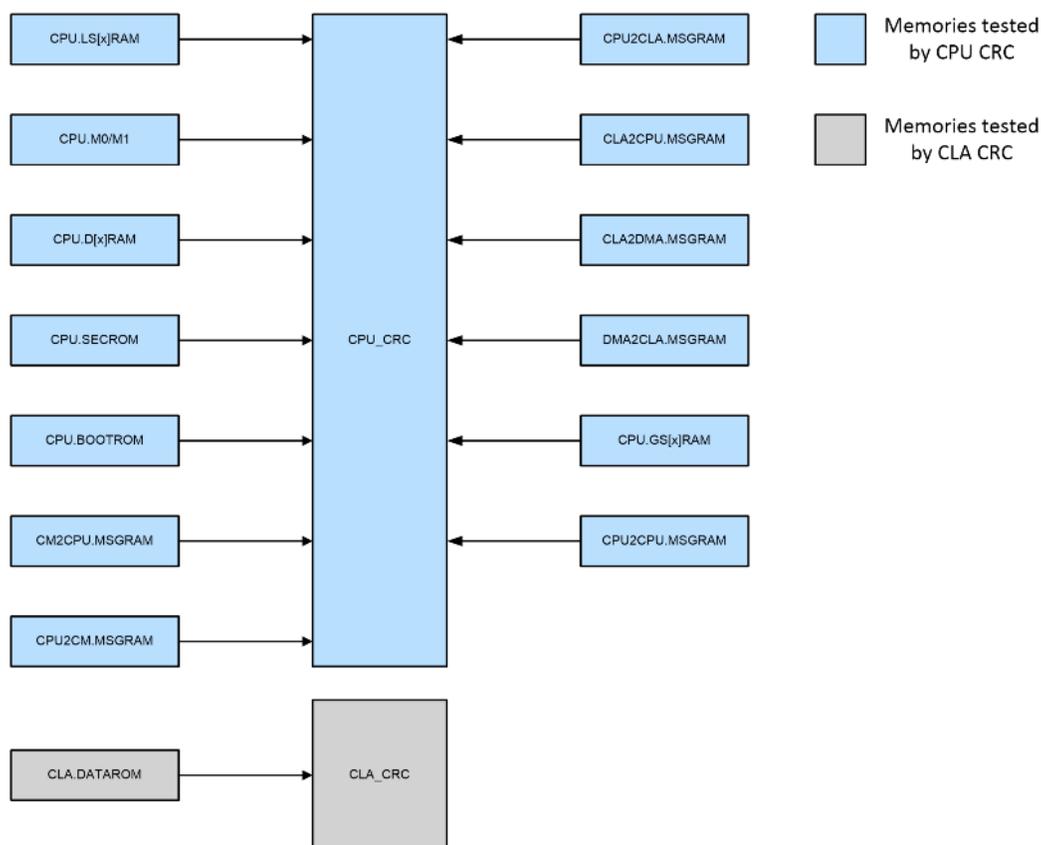


图 6-3. TMS320F2838xD 器件上的 BGCRC 实现

### 6.3.3 器件列表

- [TMS320F2838xD/S](#)
- [TMS320F28003x](#)
- [TMS320F28002x](#)

### 6.3.4 硬件平台和软件示例

- [F2838x BGCRC 示例](#)

- [F28003x BGCRC 示例](#)
- [F28002x BGCRC 示例](#)

### 6.3.5 文档

- [C2000™ 器件中的 CRC 引擎](#)

## 6.4 代码执行前的引导代码身份验证

### 6.4.1 价值主张

实时控制系统在其生命周期内通常具有多次固件 (FW) 更新。虽然可以手动完成这些更新，但使用器件上的引导加载程序远程更新固件更为常见和方便。通过远程更新，人们越来越担心确保新固件的内容不仅需要正确，而且未被外部来源操纵。C2000 MCU 上的安全启动功能提供了一种在执行代码之前验证新固件内容的方法。

### 6.4.2 深入介绍

与应用闪存引导相关的 C2000 MCU 特性之一是能够在执行闪存中的用户应用程序代码之前对其进行身份验证。这样可确保应用程序代码在被编程到闪存存储器后未被篡改，以此确定应用程序代码的完整性。

安全闪存启动是通过使用 128 位 AES-CMAC 身份验证算法实现的，该算法在会返回通过/失败状态的应用程序代码内容上运行，并且只有在身份验证成功时才继续执行应用程序代码。基于密码的消息身份验证代码 (CMAC) 是一种基于 AES 的身份验证算法，它根据输入数据块构造一种身份验证标签。输入数据块按每次 128 位，与 128 位 CMAC 密钥一起输入加密引擎/软件 (基于 CPU 子系统) (图 6-4)。

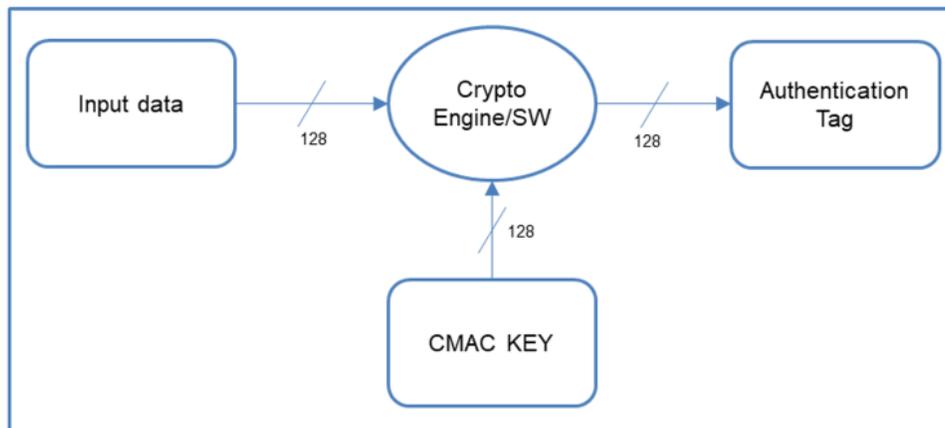


图 6-4. CMAC 工作原理

### 6.4.3 器件列表

- [TMS320F2838xD/S](#)
- [TMS320F28003x](#)
- [TMS320F280013x](#)
- [TMS320F280015x](#)

## 6.4.4 硬件平台和软件示例

- [F2838xD 的 C2000Ware 安全启动示例](#)
- [F28388D controlCARD 评估模块](#)
- [TMS320F280039C LaunchPad](#)
- [TMS320F2800137 LaunchPad](#)
- [TMDSCNCD2800157 controlCARD](#)

### 6.4.4.1 文档

- [C2000 器件上的安全启动](#)
- [TMS320F2838x 带连接管理器的实时微控制器 TRM](#)
- [TMS320F28003x 实时微控制器 TRM](#)
- [TMS320F280013x 实时微控制器 TRM](#)
- [TMS320F280015x 实时微控制器 TRM](#)

## 7 参考文献

### 7.1 器件列表

- [TMS320F2838xD/S](#)
- [TMS320F2837xD/S](#)
- [TMS320F2807x](#)
- [TMS320F28004x](#)
- [TMS320F28003x](#)
- [TMS320F28002x](#)
- [TMS320F280013x](#)
- [TMS320F280015x](#)

### 7.2 硬件/软件资源

- [TIDM-1007](#)，交错式 CCM 图腾柱无桥功率因数校正 (PFC) 参考设计
- [TIDM-HV-1PH-DCAC](#)，具有电压源和并网模式的单相逆变器参考设计
- [TMDXIDDK379D](#)，适用于工业电机控制的 C2000 DesignDRIVE 开发套件
- [TMDSHVMTRINSPIN](#)，用于 F280049C 器件 lab7 和 lab8 且带有 InstaSPIN FOC 和 InstaSPIN MOTION 的高压电机控制套件
- 基于 Vienna 整流器且采用 C2000 MCU 的三相功率因数校正参考设计
- [TIDM-02002](#)，适用于混合动力汽车/电动汽车车载充电器的双向 CLLLC 谐振双有源电桥
- [TIDA-00961](#)，基于 GaN 的高效率 1.6kW 高密度 1MHz CrM 图腾柱 PFC 转换器
- [F28388D controlCARD 评估模块](#)
- [C2000 MCU F28379D LaunchPad™ 开发套件](#)
- [C2000 MCU F280049C LaunchPad™ 开发套件](#)
- [C2000 MCU F280039C LaunchPad™ 开发套件](#)
- [C2000 MCU F280025 LaunchPad™ 开发套件](#)
- [C2000 MCU F2800137 LaunchPad™ 开发套件](#)
- 用于工业电机控制的 C2000 DesignDRIVE 开发套件
- [谷底开关升压功率因数校正 \(PFC\) 参考设计](#)
- [TMDSIDDK379D](#)
- [TMDSCNCD28379D](#)
- [TMDSCNCD280049C](#)
- [TMDSCNCD280039C](#)
- [TMDSCNCD2800137](#)
- [TMDSCNCD2800157](#)
- [TIDM-1022](#)，谷底开关升压功率因数校正

### 7.3 文档

- 德州仪器 (TI) : [C2000 实时控制外设参考指南](#)
- 德州仪器 (TI) : [TMS320F2837xD 双核实时微控制器技术参考手册](#)
- 德州仪器 (TI) : [《TMS320F28004x 实时微控制器技术参考手册》](#)
- 德州仪器 (TI) : [增强 C2000™ MCU 系列器件的性能和功能](#)
- 德州仪器 (TI) : [TMS320C28x 扩展指令集技术参考手册](#)
- 德州仪器 (TI) : [TMS320C28x CPU 和指令集参考指南](#)
- 德州仪器 (TI) : [《TMS320F28002x 实时微控制器》数据表](#)
- 德州仪器 (TI) : [《TMS320F2838x 实时微控制器技术参考手册》](#)
- 德州仪器 (TI) : [TMS320F28002x 实时微控制器技术参考手册](#)
- 德州仪器 (TI) : [使用 C2000™ 可配置逻辑块进行设计](#)
- 德州仪器 (TI) : [如何将自定义逻辑从 FPGA/CPLD 迁移到 C2000™ 微控制器](#)
- 德州仪器 (TI) : [CLB 工具用户指南](#)
- 德州仪器 (TI) : [基于快速电流环路的 PMSM 快速响应控制](#)
- 德州仪器 (TI) : [快速电流环路库](#)
- 德州仪器 (TI) : [C2000 Position Manager PTO API 参考指南](#)
- 德州仪器 (TI) : [基于快速串行接口 \(FSI\) 的分布式多轴伺服驱动器参考设计](#)
- 德州仪器 (TI) : [快速串行接口 \(FSI\) 偏斜补偿](#)
- 德州仪器 (TI) : [快速串行接口 \(FSI\) 在多芯片互连中的应用](#)
- 培训 : [C2000 可配置逻辑块 \(CLB\) 工具如何在我的设计中集成自定义逻辑](#)
- [TMS320F2838xD/S 的 ADC 规格](#)
- [TMS320F2837xD/S 的 ADC 规格](#)
- [TMS320F2807x 的 ADC 规格](#)
- [TMS320F28004x 的 ADC 规格](#)
- [C2000™ F2837xD 微控制器讲座](#)
- [CLA 实践技术讲座](#)
- [CLA 在谷底开关升压功率因数校正 \(PFC\) 参考设计中的应用](#)
- [E2E 上的 CLA 常见问题解答](#)
- [C2000 F2837xD 微控制器 1 日培训讲座第 1.6 节 - 控制外设](#)

### 8 修订历史记录

注：以前版本的页码可能与当前版本的页码不同

<b>Changes from Revision E (June 2022) to Revision F (March 2023)</b>	<b>Page</b>
• 更新了整个文档中的表格、图和交叉参考的编号格式.....	4
• 向适用器件及支持硬件工具添加了 F280013x 和 F280015x 器件.....	4
• 更新了 <a href="#">节 3.7.2</a> 。.....	35
• 更新了 <a href="#">节 3.9</a> 。.....	40
• 更新了 <a href="#">节 3.9.2</a> .....	40

## 重要声明和免责声明

TI“按原样”提供技术和可靠性数据（包括数据表）、设计资源（包括参考设计）、应用或其他设计建议、网络工具、安全信息和其他资源，不保证没有瑕疵且不做任何明示或暗示的担保，包括但不限于对适销性、某特定用途方面的适用性或不侵犯任何第三方知识产权的暗示担保。

这些资源可供使用 TI 产品进行设计的熟练开发人员使用。您将自行承担以下全部责任：(1) 针对您的应用选择合适的 TI 产品，(2) 设计、验证并测试您的应用，(3) 确保您的应用满足相应标准以及任何其他功能安全、信息安全、监管或其他要求。

这些资源如有变更，恕不另行通知。TI 授权您仅可将这些资源用于研发本资源所述的 TI 产品的应用。严禁对这些资源进行其他复制或展示。您无权使用任何其他 TI 知识产权或任何第三方知识产权。您应全额赔偿因在这些资源的使用中对 TI 及其代表造成的任何索赔、损害、成本、损失和债务，TI 对此概不负责。

TI 提供的产品受 [TI 的销售条款](#) 或 [ti.com](#) 上其他适用条款/TI 产品随附的其他适用条款的约束。TI 提供这些资源并不会扩展或以其他方式更改 TI 针对 TI 产品发布的适用的担保或担保免责声明。

TI 反对并拒绝您可能提出的任何其他或不同的条款。

邮寄地址：Texas Instruments, Post Office Box 655303, Dallas, Texas 75265

Copyright © 2023，德州仪器 (TI) 公司