

Errata

TMS320F28004x 实时 MCU 器件勘误表

器件修订版本 D、B、A、0



摘要

本文档介绍了功能规格的已知例外情况（公告）。本文档也包含了使用说明。在使用说明中介绍了器件行为可能与假定或记录的行为不匹配的情况。这可能包括影响器件性能或功能正确性的行为。

内容

1 使用说明和公告模型	3
1.1 使用说明汇总表	3
1.2 公告汇总表	3
2 命名规则、封装编号法和修订版本标识	5
2.1 器件和开发支持工具命名规则	5
2.2 支持的器件	5
2.3 封装编号法和修订版本标识	6
3 器件修订版本 D 使用说明和公告	9
3.1 器件修订版本 D 使用说明	9
3.2 器件修订版本 D 公告	12
4 器件修订版本 B 使用说明和公告	44
4.1 器件修订版本 B 使用说明	44
4.2 器件修订版本 B 公告	44
5 器件修订版本 A 使用说明和公告	45
5.1 器修订版本 A 使用说明	45
5.2 器件修订版本 A 公告	45
6 器件修订版本 0 使用说明和公告	49
6.1 器件修订版本 0 使用说明	49
6.2 器件修订版本 0 公告	49
7 文档支持	52
8 商标	52
9 修订历史记录	52

插图清单

图 2-1. PM 和 PZ 封装的器件标识示例	6
图 2-2. RSH 封装的器件标识示例	6
图 2-3. 器件命名规则示例	8
图 3-1. 流水线中没有停滞时的问题流水线图	12
图 3-2. 指令 I1 的 E3 时隙中存在停滞时的问题流水线图	13
图 3-3. 带解决方法的流水线图	14
图 3-4. 串联终端电阻器和上拉电阻器的放置方式	26
图 3-5. 不良跳闸事件和消隐窗口过期	29
图 3-6. 可能会产生不良的 ePWM 输出	29

表格清单

表 1-1. 使用说明汇总表.....	3
表 1-2. 公告汇总表.....	3
表 2-1. 从批次追踪代码中确定器件的修订版本.....	6
表 3-1. C2000 作为带有标准模式主机的目标发送器时的数据上升时间要求.....	27
表 3-2. 公共总线电容 (C _b) 的上拉电阻 (R _p) 值.....	28
表 3-3. 受公告影响的存储器.....	35
表 3-4. ADCCTL2 寄存器.....	38

1 使用说明和公告模型

表 1-1 列出了所有使用说明和适用的器件修订版本。表 1-2 列出了所有公告、受影响的模块以及适用的器件修订版本。

1.1 使用说明汇总表

表 1-1. 使用说明汇总表

编号	标题	受影响的器件修订版本			
		0	A	B	D
节 3.1.1	PIE：背对背 PIEACK 写入和手动 CPU 中断屏蔽清除之后的伪波嵌套中断	是	是	是	是
节 3.1.2	FPU32 和 VCU 背对背存储器访问	是	是	是	是
节 3.1.3	使用嵌套中断时要小心	是	是	是	是
节 3.1.4	安全性：主要的防御层是构建芯片安全边界，从启用 JTAGLOCK 和零引脚引导至闪存功能开始	是	是	是	是

1.2 公告汇总表

表 1-2. 公告汇总表

模块	说明	受影响的器件修订版本			
		0	A	B	D
CMPSS	CMPSS：在某些情况下，COMPxLATCH 可能无法正确清除	是	是	是	是
FPU	FPU：FPU 至 CPU 寄存器移动操作之前是任何 FPU 2p 操作	是	是	是	是
SDFM	SDFM：动态更改阈值设置 (LLT、HLT)、滤波器类型或 COSR 设置将触发虚假比较器事件	是	是	是	是
SDFM	SDFM：动态更改数据滤波器设置 (例如滤波器类型或 DOSR) 将触发虚假数据响应事件	是	是	是	是
SDFM	SDFM：在三个 SD 调制器时钟周期内对 SDCPARMx 寄存器位字段 CEVT1SEL、CEVT2SEL 和 HZEN 进行两次背对背写入可能会损坏 SDFM 状态机，从而导致虚假比较器事件	是	是	是	是
SDFM	SDFM：Manchester 模式 (模式 2) 无法在多种条件下生成正确的滤波器结果	是	是	是	是
eQEP	eQEP：索引期间方向变化时位置计数器复位错误	是	是	是	是
eQEP	eQEP：GPIO 异步模式下的 eQEP 输入	是	是	是	是
V _{DD} 电源：	V _{DD} 电源：在 V _{DDIO} 加电期间，V _{DD} 也可能会上升	是	是	是	是
eCAP	eCAP：HRFRC 不受 EALLOW 保护	是	是	是	是
PLL	PLL：第一次尝试锁定时，PLL 可能无法锁定	是	是	是	是
LPM	LPM：不支持待机低功耗模式	是	是	是	是
DCC	DCC：单次模式操作可能会过早终止	是	是	是	是
BOR	BOR：2.45V 至 3.0V 之间的 VDDIO 可产生多个 XRSn 脉冲	是	是	是	是
I2C	I2C：SDA 和 SCL 漏极开路输出缓冲器问题	是	是	是	是
I2C	I2C：目标发送器模式、标准模式 SDA 时序限制	是	是	是	是
ePWM	ePWM：如果跳闸在消隐窗口结束时保持活动状态，则可能会发生 ePWM 干扰	是	是	是	是
ePWM	ePWM：在消隐窗口后的前 3 个周期内，消隐窗口不会过滤跳闸事件	是	是	是	是
ePWM	ePWM：选择异步路径后，“基于 DC 事件的 CBC 跳闸”的事件门限 (DCxEVTxLAT) 可能不会按预期延长触发脉冲	是	-	-	-
ePWM	ePWM：单次/CBC 跳闸事件 DCxEVTy.force 不设置跳闸条件	是	是	是	是
INTOSC	INTOSC：在没有 VDD 的情况下为 VDDIO 加电会导致 INTOSC 频率漂移	是	是	是	是
FSI	FSI：RX FIFO 伪波超限	是	是	是	是
闪存	闪存：在未禁用闪存预取的情况下执行 Fapi_setActiveFlashBank () 可能会导致 ITRAP	是	是	是	是
	在 DCAN FIFO 模式期间，接收到的消息可能以无序状态显示在 FIFO 缓冲器中	是	是	是	是

表 1-2. 公告汇总表（续）

模块	说明	受影响的器件修订版本			
		0	A	B	D
引导 ROM	引导 ROM：从应用程序调用 SCI 引导加载程序	是	是	是	是
Boot ROM、MPOST	Boot-ROM、MPOST：启用 MPOST 时引导时间更长	是	是	是	是
内存	存储器：在有效存储器之外进行预取	是	是	是	是
系统	系统：系统控制寄存器配置限制	是	是	是	是
ADC	ADC：如果未设置 INTxCONT（继续中断模式），中断可能会停止	是	是	是	是
ADC	ADC：使用 ADCCLK 小数分频器时 ADC 性能下降	是	是	是	是
ADC	ADC：DMA 读取过时结果	是	是	是	是
CLB	CLB：不支持涉及多个有效高级控制器 (HLC) 通道的背对背 PUSH 或 PULL 指令	是	是	是	是
模拟子系统	模拟子系统：共享基准引脚的软件配置	是	是	-	-
	某些 TMX 器件的模拟修整	是	是	-	-
PGA	PGA：不支持输出滤波器路径	是	是	-	-
ROM	ROM：不存在闪存 API 库和 FPU32 Twiddle Factor RFFT 表	是	是	-	-
REVID	REVID：某些 TMX 修订版本 A 器件的 REVID 值不正确	-	是	-	-
GPIO	GPIO：加电期间的 X2/GPIO18 引脚上拉电流	是	是	是	是
GPIO	GPIO：漏极开路配置可以驱动短高电平脉冲	是	是	是	是
GPIO	GPIO：在输入模式下超过最大 V_{IH} 时， V_{SS} 的寄生路径	-	是	-	-
GPIO	GPIO：引脚在加电期间可以驱动为高电平	是	-	-	-
GPIO	GPIO：信号门锁至 V_{SS}	是	-	-	-
LIN	LIN：当同步字段错误时，同步字段错误 (ISFE) 标志/中断未设置	是	是	是	是
XTAL	XTAL：不支持 XTALCR[SWH]=0	是	是	是	是

2 命名规则、封装编号法和修订版本标识

2.1 器件和开发支持工具命名规则

为了标示产品开发周期所处的阶段，TI 为所有 TMS320 MCU 器件和支持工具的器件型号分配了前缀。每个 TMS320™ MCU 商用产品系列成员都具有以下三个前缀之一：TMX、TMP 或 TMS（例如，TMS320F280049）。德州仪器 (TI) 为其支持工具推荐使用三种可能的前缀指示符中的两个：TMDX 和 TMDS。这些前缀代表了产品从工程原型（其中 TMX 针对器件，而 TMDX 针对工具）直到完全合格的生产器件和工具（其中 TMS 针对器件，而 TMDS 针对工具）的产品开发演变阶段。

器件开发演变流程：

TMX 试验器件不一定代表最终器件的电气规范标准，并且可能不使用生产组装流程。

TMP 原型器件不一定是最终器件模型，并且不一定符合最终电气标准规范。

TMS 完全合格的芯片模型的生产版本。

支持工具开发演变流程：

TMDX 还未经德州仪器 (TI) 完整内部质量测试的开发支持产品。

TMDS 完全合格的开发支持产品。

TMX 和 TMP 器件和 TMDX 开发支持工具供货时附带如下免责条款：

“开发的产品用于内部评估用途。”

生产器件和 TMDS 开发支持工具已进行完全特性描述，并且器件的质量和可靠性已经完全论证。TI 的标准保修证书适用。

预测显示原型器件（X 或者 P）的故障率大于标准生产器件。由于这些器件的预期最终使用故障率仍未确定，故德州仪器 (TI) 建议请勿将这些器件用于任何生产系统。请仅使用合格的生产器件。

TI 的器件命名规则还包含具有器件产品系列名称的后缀。此后缀表示封装类型（例如 PZ）和温度范围（例如 S）。

2.2 支持的器件

本文档支持以下器件：

- [TMS320F280049](#)
- [TMS320F280049-Q1](#)
- [TMS320F280049C](#)
- [TMS320F280049C-Q1](#)
- [TMS320F280048-Q1](#)
- [TMS320F280048C-Q1](#)
- [TMS320F280045](#)
- [TMS320F280041](#)
- [TMS320F280041-Q1](#)
- [TMS320F280041C](#)
- [TMS320F280041C-Q1](#)
- [TMS320F280040-Q1](#)
- [TMS320F280040C-Q1](#)

2.3 封装编号法和修订版本标识

图 2-1 和图 2-2 提供了 F28004x 器件标识示例并定义了各个标识。您可以通过封装顶部所示的符号判断器件的修订版本，如 图 2-1 和 图 2-2 所示。一些原型器件的标识可能与图示标识有所不同。图 2-3 显示了器件命名规则。

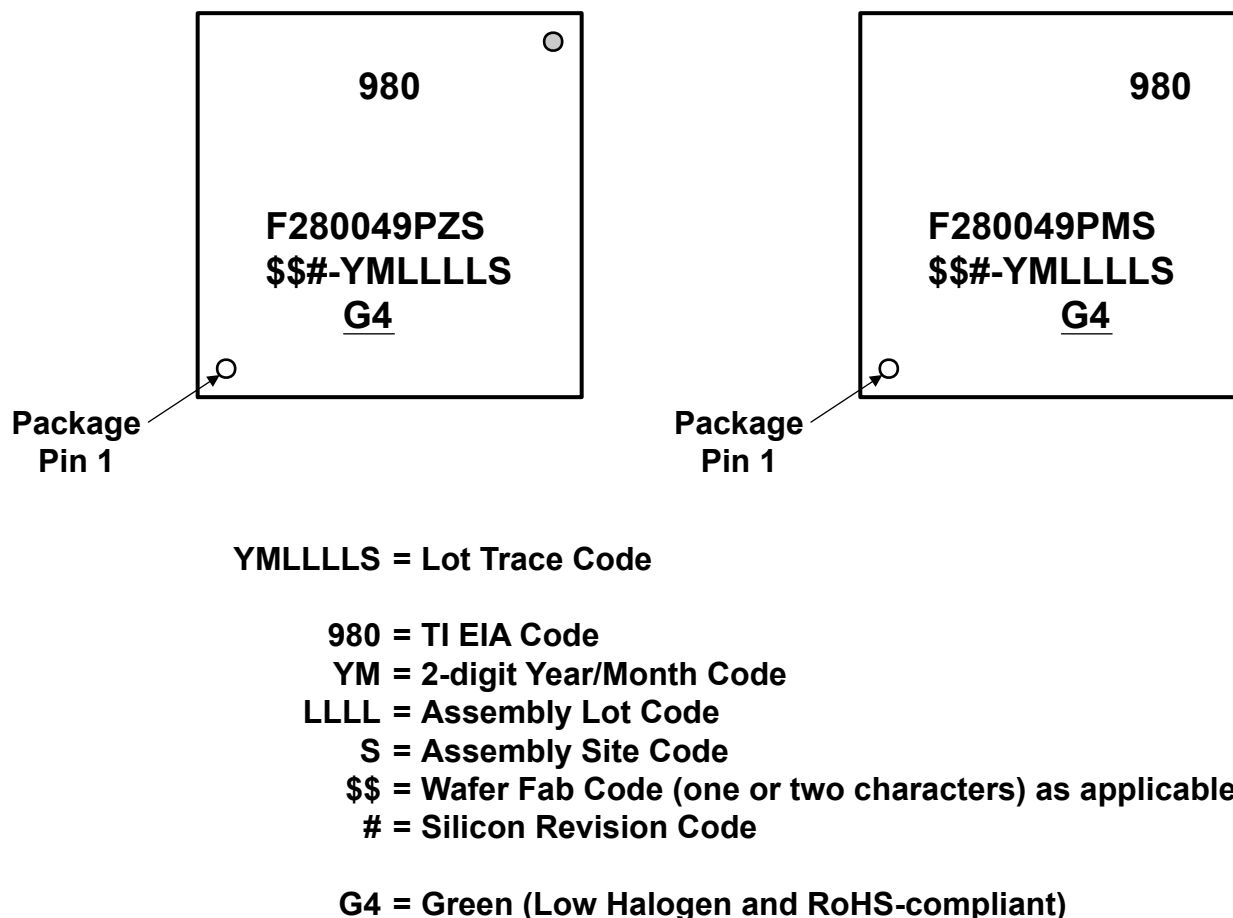


图 2-1. PM 和 PZ 封装的器件标识示例

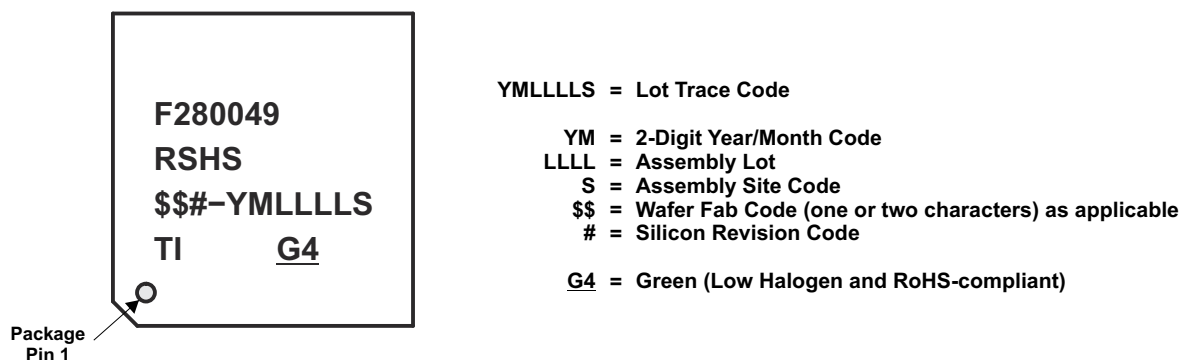


图 2-2. RSH 封装的器件标识示例

表 2-1. 从批次追踪代码中确定器件的修订版本

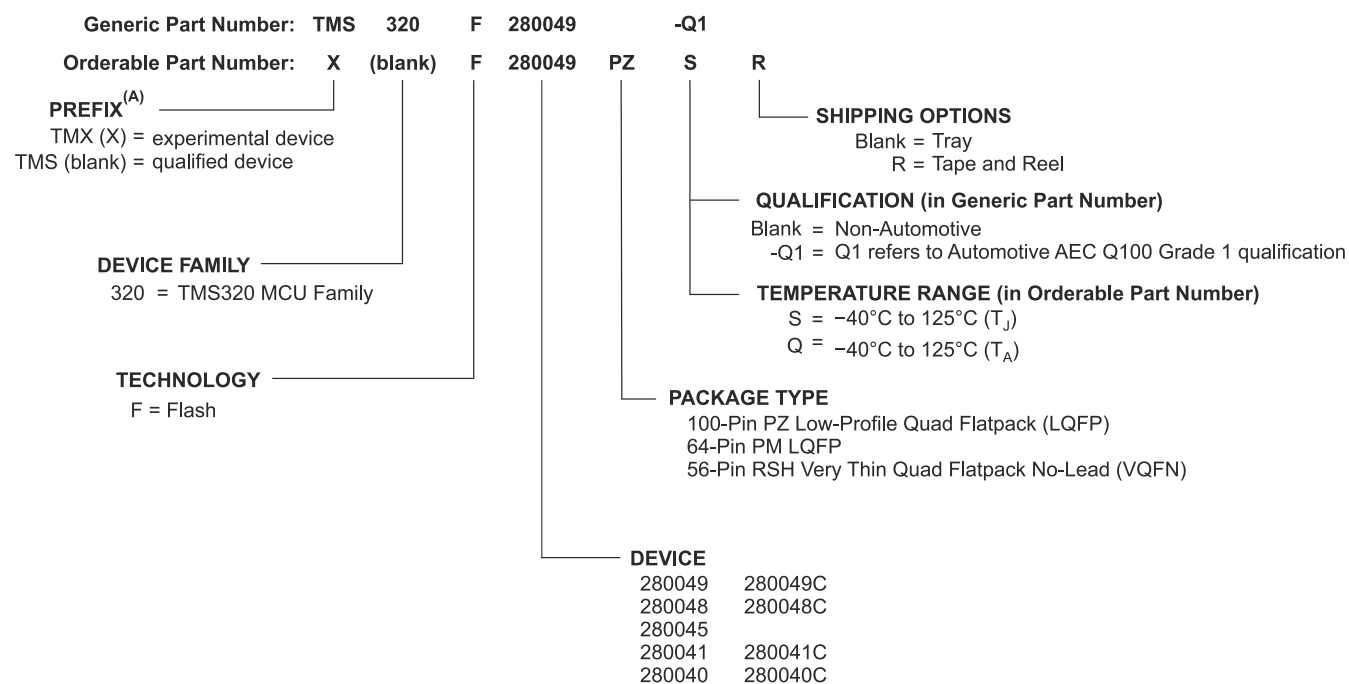
器件修订版本代码	器件修订版本	REVID ⁽¹⁾ 地址：0x5D00C	注释 ⁽²⁾
空白	0	0x0000 0000	该器件修订版本以 TMX 形式提供。
A	A	0x0000 0001	该器件修订版本以 TMX 形式提供。
B	B	0x0000 0002	该器件修订版本的代码为 TMX 和 TMS。

表 2-1. 从批次追踪代码中确定器件的修订版本（续）

器件修订版本代码	器件修订版本	REVID ⁽¹⁾ 地址：0x5D00C	注释 ⁽²⁾
D	D	0x0000 0004	该器件修订版本以 TMS 形式提供。

(1) 器件修订版本 ID

(2) 有关可订购器件型号，请参阅 [TMS320F28004x 实时微控制器](#) 数据表 中的封装信息表。



A. 可订购器件型号使用前缀 X。

图 2-3. 器件命名规则示例

3 器件修订版本 D 使用说明和公告

本节列出了此器件修订版本的使用说明和公告。

3.1 器件修订版本 D 使用说明

本节列出了适用于器件修订版本 D [和早期器件修订版本]的所有使用说明。

3.1.1 PIE : 背对背 PIEACK 写入和手动 CPU 中断屏蔽清除之后的伪波嵌套中断

受影响的版本 : 0、A、B、D

某些用于嵌套中断的代码序列允许 CPU 和 PIE 进入不一致状态,从而触发不必要的中断。进入该状态所需的条件为:

1. PIEACK 清除后,立即执行全局中断使能 (EINT 或 ASM (" CLRC INTM")) 。
2. 嵌套中断会清除其组的一个或多个 PIEIER 位。

是否触发不必要的中断,取决于系统中其他中断的配置和时序。在大多数应用中,预计这种事件很罕见或根本不存在。如果发生这种情况,不必要的中断将是嵌套中断的 PIE 组中的第一个中断、并将在嵌套中断重新启用 CPU 中断 (EINT 或 asm ("CLRC INTM")) 后触发。

权变措施: 在 PIEACK 写入和 CPU 中断使能之间添加一个 NOP (无操作)。以下显示了示例代码。

```
//Bad interrupt nesting code
PieCtrlRegs.PIEACK.all = 0xFFFF;    //Enable nesting in the PIE
EINT;                                //Enable nesting in the CPU

//Good interrupt nesting code
PieCtrlRegs.PIEACK.all = 0xFFFF;    //Enable nesting in the PIE
asm(" NOP");                          //wait for PIEACK to exit the pipeline
EINT;                                //Enable nesting in the CPU
```

3.1.2 FPU32 和 VCU 背对背存储器访问

受影响的版本：0、A、B、D

当 VCU 存储器访问和 FPU 存储器访问背靠背发生时，此使用说明适用。共有三种情形：

情形 1.背对背存储器读取：一次由 VCU 指令 (VMOV32) 执行读取，另一次由 FPU32 指令 (MOV32) 执行读取。

如果在第一次读取期间发生 R1 流水线阶段停滞，则第二次读取将锁存错误的数据。如果第一条指令在 R1 流水线阶段没有发生停滞，则第二次读取将正常进行。

指令的顺序 (FPU 后跟 VCU，或者 VCU 后跟 FPU) 无关紧要。任一读取操作所访问的存储器地址也无关紧要。

情形 1 解决方法：在两条背对背读取指令之间插入一条指令。可以使用任何指令，VCU 或 FPU 存储器读取除外。

情形 1，示例 1：

```
VMOV32  VR1,mem32      ; VCU memory read
NOP      ; Not a FPU/ VCU memory read
MOV32    R0H,mem32      ; FPU memory read
```

情形 1，示例 2：

```
VMOV32  VR1,mem32      ; VCU memory read
VMOV32  mem32, VR2      ; VCU memory write
MOV32    R0H,mem32      ; FPU memory read
```

情形 2.背对背存储器读取：一次由 VCU 指令 (VMOV32) 执行写入，另一次由 FPU 指令 (MOV32) 执行写入。

如果在第一次写入期间发生流水线停滞，则第二次写入将损坏数据。如果第一条指令在写入阶段没有发生停滞，则不会发生损坏情形。

指令的顺序 (FPU 后跟 VCU，或者 VCU 后跟 FPU) 无关紧要。任一写入操作所访问的存储器地址也无关紧要。

情形 2 解决方法：在背对背 VCU 和 FPU 写入之间插入两条指令。可以使用任何指令，VCU 或 FPU 存储器写入除外。

情形 2，示例 1：

```
VMOV32  mem32,VR0      ; VCU memory write
NOP      ; Not a FPU/VCU memory write
NOP      ; Not a FPU/VCU memory write
MOV32    mem32,R3H      ; FPU memory write
```

情形 2，示例 2：

```
VMOV32  mem32,VR0      ; VCU memory write
VMOV32  VR1, mem32      ; VCU memory read
NOP      ;
MOV32    mem32,R3H      ; FPU memory write
```

情形 3.背对背存储器写入，后跟读取或存储器读取，然后是写入。在这种情况下，两条指令之间没有交互。无需执行任何操作。

权变措施：请参阅情形 1 解决方法和情形 2 解决方法。

3.1.3 使用嵌套中断时要小心

受影响的版本：0、A、B、D

如果用户为了使用嵌套特性而在中断处理例程 (ISR) 内使用 EINT 指令启用中断，那么用户必须在退出 ISR 之前禁用中断。否则，CPU 可能会执行未确定的操作。

3.1.4 安全性：主要的防御层是构建芯片安全边界，从启用 JTAGLOCK 和零引脚引导至闪存功能开始

受影响的版本：0、A、B、D

在任何情况下都不允许未经授权的代码进入器件并执行，这是器件安全的前提条件。为此，器件提供了两项功能，担心安全性的用户应该始终启用这两项功能。

- **JTAGLOCK**

当在闪存的 USER OTP 区域中启用时，JTAGLOCK 功能会禁用对器件的资源进 JTAG 访问（例如调试器连接），阻止未经授权的一方使用 JTAG 接口将任何代码下载到器件中。启用 JTAGLOCK 后，用户仍然可以通过输入密码来允许授权方解锁，也可以通过将密码值全部设置为零来永久锁定。

- **零引脚引导至闪存**

内置于 TI ROM 中的外部引导加载程序不会对所下载的代码执行任何身份验证。在 USER OTP 中启用零引脚引导选项以及闪存引导模式，则在引导时会阻止运行所有基于引脚的外部引导加载程序选项（例如 SCI、CAN、并行），这是因为在基本引导 ROM 执行结束后，会强制使引导过程立即跳转到内部闪存。为了保障最高的安全性，可选择安全闪存启动模式。这可以在跳转到闪存代码之前通过基本引导 ROM 对闪存代码进行预检查。

如果 JTAG 被永久锁定并且启用了零引脚引导至闪存选项，则通过 JTAG 或内置引导加载程序与器件进行通信的编程工具将失效。如果需要能够执行固件升级，则用户必须将代码预存储在闪存中以安全地管理和执行更新。

3.2 器件修订版本 D 公告

本节列出了适用于器件修订版本 D [和早期器件修订版本] 的所有公告。

公告

FPU : FPU 至 CPU 寄存器移动操作之前是任何 FPU 2p 操作

受影响版本

0、A、B、D

详细信息

当多周期 (2p) FPU 指令后跟 FPU 到 CPU 寄存器转移时，此公告适用。如果 FPU 到 CPU 读取指令源寄存器与 2p 指令目标相同，则在 2p 指令完成之前读取的可能是 FPU 寄存器的值。这是因为 2p 指令依赖于流水线 E3 阶段期间结果的数据转发。如果在 E3 阶段碰巧发生流水线停滞，则读取指令不会及时转发结果。

受此公告影响的 2p 指令是 MPYF32、ADDF32、SUBF32 和 MACF32。FPU 寄存器读取的目标必须是一个 CPU 寄存器 (ACC、P、T、XAR0...XAR7)。如果寄存器读取是 FPU 到 FPU 寄存器转移，则此公告不适用。

在下面的示例中，2p 指令 MPYF32 使用 R6H 作为其目标。FPU 寄存器读取 MOV32 使用同一个寄存器 R6H 作为其源，并使用 CPU 寄存器作为目标。如果在 E3 流水线阶段发生停滞，则 MOV32 将在 MPYF32 指令完成之前读取 R6H 的值。

问题示例：

```
MPYF32 R6H, R5H, R0H ; 2p FPU instruction that writes to R6H
|| MOV32 *XAR7++, R4H
F32TOUI16R R3H, R4H ; delay slot
ADDF32 R2H, R2H, R0H
|| MOV32 *--SP, R2H ; alignment cycle
MOV32 @XAR3, R6H ; FPU register read of R6H
```

图 3-1 展示了流水线中没有停滞时的问题流水线图。

	Instruction	F1	F2	D1	D2	R1	R2	E	W		Comments
		FPU pipeline-->				R1	R2	E1	E2	E3	
I1	MPYF32 R6H, R5H, R0H MOV32 *XAR7++, R4H	I1									
I2	F32TOUI16R R3H, R4H	I2	I1								
I3	ADDF32 R3H, R2H, R0H MOV32 *--SP, R2H	I3	I2	I1							
I4	MOV32 @XAR3, R6H	I4	I3	I2	I1						
			I4	I3	I2	I1					
				I4	I3	I2	I1				
					I4	I3	I2	I1			
						I4	I3	I2	I1		
							I4	I3	I2	I1	I4 samples the result as it enters the R2 phase. The product R6H=R5H*R0H (I1) finishes computing in the E3 phase, but is forwarded as an operand to I4. This makes I4 appear to be a 2p instruction, but I4 actually takes 3p cycles to compute.
								I4	I3	I2	
									I4	I3	

图 3-1. 流水线中没有停滞时的问题流水线图

公告 (续)

FPU : FPU 至 CPU 寄存器移动操作之前是任何 FPU 2p 操作

图 3-2 展示了指令 I1 的 E3 时隙中存在停滞时的问题流水线图。

	Instruction	F1	F2	D1	D2	R1	R2	E	W		Comments
		FPU pipeline-->				R1	R2	E1	E2	E3	
I1	MPYF32 R6H, R5H, R0H MOV32 *XAR7++, R4H	I1									
I2	F32TOUI16R R3H, R4H	I2	I1								
I3	ADDF32 R3H, R2H, R0H MOV32 *--SP, R2H	I3	I2	I1							
I4	MOV32 @XAR3, R6H	I4	I3	I2	I1						
			I4	I3	I2	I1					
				I4	I3	I2	I1				
					I4	I3	I2	I1			
						I4	I3	I2	I1		
							I4	I3	I2	I1 (STALL)	I4 samples the result as it enters the R2 phase, but I1 is stalled in E3 and is unable to forward the product of R5H*R0H to I4 (R6H does not have the product yet due to a design bug). So, I4 reads the old value of R6H.
							I4	I3	I2	I1	There is no change in the pipeline as it was stalled in the previous cycle. I4 had already sampled the old value of R6H in the previous cycle.
								I4	I3	I2	Stall over

图 3-2. 指令 I1 的 E3 时隙中存在停滞时的问题流水线图

解决方法

在本例中，应将 MPYF32、ADDF32、SUBF32 和 MACF32 视为 3p 周期指令。必须将三个 NOP 或非冲突指令放置在指令的延迟时隙中。

C28x 代码生成工具 v.6.2.0 及更高版本都将生成正确的指令序列并检测汇编代码中的错误。在以前的版本 v6.0.5 (适用于 6.0.x 分支) 和 v.6.1.2 (适用于 6.1.x 分支) 中，编译器将生成正确的指令序列，但汇编器不会检测汇编代码中的错误。

解决方法示例：

```
MPYF32 R6H, R5H, R0H
|| MOV32 *XAR7++, R4H      ; 3p FPU instruction that writes to R6H
F32TOUI16R R3H, R4H      ; delay slot
ADDF32 R2H, R2H, R0H
|| MOV32 *--SP, R2H      ; delay slot
NOP                       ; alignment cycle
MOV32 @XAR3, R6H          ; FPU register read of R6H
```

图 3-3 展示了带解决方法的流水线图。

公告 (续)

FPU : FPU 至 CPU 寄存器移动操作之前是任何 FPU 2p 操作

	Instruction	F1	F2	D1	D2	R1	R2	E	W		Comments
		FPU pipeline-->				R1	R2	E1	E2	E3	
I1	MPYF32 R6H, R5H, R0H MOV32 *XAR7++, R4H	I1									
I2	F32TOUI16R R3H, R4H	I2	I1								
I3	ADDF32 R3H, R2H, R0H MOV32 *--SP, R2H	I3	I2	I1							
I4	NOP	I4	I3	I2	I1						
I5	MOV32 @XAR3, R6H	I5	I4	I3	I2	I1					
			I5	I4	I3	I2	I1				
				I5	I4	I3	I2	I1			
					I5	I4	I3	I2	I1		
						I5	I4	I3	I2	I1 (STALL)	Due to one extra NOP, I5 does not reach R2 when I1 enters E3; thus, forwarding is not needed.
						I5	I4	I3	I2	I1	There is no change due to the stall in the previous cycle.
							I5	I4	I3	I2	I1 moves out of E3 and I5 moves to R2. R6H has the result of R5H*R0H and is read by I5. There is no need to forward the result in this case.
								I5	I4	I3	

图 3-3. 带解决方案的流水线图

公告	<i>SDFM : 动态更改阈值设置 (LLT、HLT)、滤波器类型或 COSR 设置将触发虚假比较器事件</i>
受影响版本	0、A、B、D
详细信息	当 SDFM 比较器设置 (例如滤波器类型、下限/上限阈值或比较器 OSR (COSR) 设置) 在运行期间动态更改时, 将触发虚假比较器事件。如果配置得当, 虚假比较器事件将触发相应的 CPU 中断、CLA 任务、ePWM X-BAR 事件和 GPIO 输出 X-BAR 事件。
解决方法	当需要动态更改比较器设置时, 请执行以下程序, 以确保虚假比较器事件不会产生 CPU 中断、CLA 任务或 X-BAR 事件 (ePWM X-BAR/GPIO 输出 X-BAR 事件) : 1. 禁用比较器滤波器。 2. 至少延迟比较器滤波器的延时 + 3 个 SD-Cx 时钟周期。 3. 更改比较器滤波器设置, 例如滤波器类型、COSR 或下限/上限阈值。 4. 至少延迟比较器滤波器的延时 + 5 个 SD-Cx 时钟周期。 5. 启用比较器滤波器。
公告	<i>SDFM : 动态更改数据滤波器设置 (例如滤波器类型或 DOSR) 将触发虚假数据响应事件</i>
受影响版本	0、A、B、D
详细信息	当 SDFM 数据设置 (例如滤波器类型或 DOSR 设置) 在运行期间动态更改时, 将触发虚假数据滤波器就绪事件。如果配置得当, 虚假数据就绪事件将触发相应的 CPU 中断、CLA 任务和 DMA 触发器。
解决方法	当需要动态更改 SDFM 数据滤波器设置时, 请执行以下程序, 以确保不会产生虚假数据滤波器就绪事件 : 1. 禁用数据滤波器。 2. 至少延迟数据滤波器的延时 + 3 个 SD-Cx 时钟周期。 3. 更改数据滤波器设置, 如滤波器类型和 DOSR。 4. 至少延迟数据滤波器的延时 + 5 个 SD-Cx 时钟周期。 5. 启用数据滤波器。

公告	SDFM : 在三个 SD 调制器时钟周期内对 SDCPARMx 寄存器位字段 CEVT1SEL、CEVT2SEL 和 HZEN 进行两次背对背写入可能会损坏 SDFM 状态机，从而导致虚假比较器事件
受影响版本	0、A、B、D
详细信息	在三个 SD 调制器时钟周期内对 SDCPARMx 寄存器位字段 CEVT1SEL、CEVT2SEL 和 HZEN 进行背对背写入可能会损坏 SDFM 状态机，从而导致虚假比较器事件，如果配置得当，这可能触发 CPU 中断、CLA 任务、ePWM XBAR 事件和 GPIO 输出 X-BAR 事件。
解决方法	避免在三个 SD 调制器时钟周期内背对背写入或在一次寄存器写操作中配置 SDCPARMx 寄存器位字段。
公告	SDFM : Manchester 模式 (模式 2) 无法在多种条件下生成正确的滤波器结果
受影响版本	0、A、B、D
详细信息	Manchester 解码算法在 1024 SDx_Dy 信号转换的校准窗口中，使用 SYSCLK 对 Manchester 位流进行采样。通过 Manchester 位流推算出的时钟用于在后续校准窗口周期内对数据进行采样。 有多种情况可能导致滤波器结果出现较大误差： - SDx_Dy 上的任何单个噪声事件都有可能损坏解码后的 Manchester 时钟，并导致以不正确的频率对后续数据进行采样。 - 如果 Manchester 位流时钟速率接近 SYSCLK 的精确整数倍，则当 Manchester 流与内部 SYSCLK 的相位在下一个 1024 转换校准窗口生效之前发生相位漂移时，偶尔会有一个 Manchester 位被跳过。还需要考虑 Manchester 时钟占空比偏差为 50% 的情况，以确保较长的 Manchester 脉冲不是 SYSCLK 的整数倍。如果 SD 调制器或该器件的时钟源存在较大变化范围，这种情况可能无法避免，原因是宽范围的禁止频率会带来难题。 - 如果 Manchester 上升沿与下降沿之间的边沿延迟差异（即位流占空比）大于一个 SYSCLK 周期，则 SDFM 时钟解码算法可能会错误地将时钟周期识别得比实际更短。
解决方法	可用的解决方法如下： - 避免使用 Manchester 模式并考虑使用模式 0，以便在噪声条件下提供出色的滤波器性能。这是推荐的解决方法。 - 避免 Manchester 位流上出现任何噪声，并避免所选 Manchester 时钟源与 SYSCLK 形成整数倍关系。调制器和此器件必须使用精密时钟源。 - 确保上升沿和下降沿延迟（即高电平脉冲和低电平脉冲）的长度彼此相差不超过一个 SYSCLK 周期。 - 设计一种应用级算法，使其能够抵御偶尔出现的错误 SDFM 结果。

公告

eQEP: 索引期间方向变化时位置计数器复位错误

受影响版本

0、A、B、D

详细信息

在使用 $PCRM = 0$ 的配置时，若索引输入处于活动状态时方向发生变化，则位置计数器 (QPOSCNT) 可能会错误地复位，从而导致计数器值发生意外变化。若如此，可能会与位置计数器的预期值相差多达 ± 4 个数值，并随后意外设置错误标志。

在使用 $PCRM = 0$ 的配置时[即发生索引事件时位置计数器复位 (QEPCTL[PCRM] = 00)]，若在正向运动期间发生索引事件，则位置计数器在下一个 eQEP 时钟时复位为 0。若在反向移动期间发生索引事件，则位置计数器将在下一个 eQEP 时钟时复位为 QPOSMAX 寄存器中的值。eQEP 外围设备会记录第一个索引标记 (QEPSTS[FIMF]) 的出现情况以及 QEPSTS 寄存器中第一个索引事件标记 (QEPSTS[FIDF]) 方向。此外，其还会记录第一个索引标记上的正交边沿以便在索引事件复位操作中使用相同的相对正交跃迁。

如果在索引脉冲活动时方向发生改变，则该模块仍将继续查找相对正交跃迁以执行位置计数器复位操作。若如此，位置计数器的值会发生意外变化。

未同时发生方向改变的下一个索引事件将正确复位计数器并按预期工作。

解决方法

在索引处于活动状态时，若方向发生变化且位置计数器值的相应变化可能会影响应用程序，请勿使用 $PCRM = 0$ 的配置。

若适用于此应用程序，则位置计数器复位所运用的其他选项[例如索引事件初始化(IEI)]不会出现此问题。

公告

eQEP : GPIO 异步模式下的 eQEP 输入

受影响版本

0、A、B、D

详细信息

如果通过 GPxQSELn 寄存器将任一 eQEP 输入引脚配置为 GPIO 异步输入模式，则 eQEP 模块可能无法正常运行，原因是 eQEP 外设假定其模块输入端存在与 SYSCLKOUT 的外部同步。例如，QPOSCNT 可能无法正确复位或锁存，并且输入引脚上的脉冲可能丢失。

为确保 eQEP 模块正常运行，应通过 GPxQSELn 寄存器将输入 GPIO 引脚配置为同步输入模式 (无论是否经过认证)，这也是 GPxQSEL 寄存器在复位时的默认状态。TI 提供的所有现有 eQEP 外设示例还将 GPIO 输入配置为同步输入模式。

异步模式不得用于 eQEP 模块输入引脚。

解决方法

将配置为 eQEP 引脚的 GPIO 输入配置为非异步模式 (除 “11b = 异步” 之外的任何 GPxQSELn 寄存器选项)。

公告	V_{DD} 电源：在 V_{DDIO} 加电期间，V_{DD} 也可能会上升
受影响版本	0、A、B、D
详细信息	当 V_{DD} 电源电压低于约 0.5V 时，存在从 V_{DDIO} 到 V_{DD} 的泄漏电流。这会导致 V_{DD} 电压在 V_{DDIO} 加电时上升至约 0.5V。当器件配置为使用内部 VREG (VREGENZ 连接至 V_{SS}) 或外部 1.2V 稳压器 (VREGENZ 连接至 V_{DDIO})，就会看到这种情况。在内部 VREG 模式下，V_{DDIO} 与 V_{DD} 加电之间存在显著延迟 (约 1ms)，或者 V_{DDIO} 的斜升时间远超 1ms。 如果外部 1.2V 或内部 1.2V 电源开始斜升，这将不会影响器件功能。如需了解电源时序要求，请参阅 TMS320F28004x 实时微控制器 数据表。
解决方法	如果 V_{DD} 上的这种早期电压对系统级监控电路构成问题，则应尽可能地减小 3.3V V_{DDIO} 与 1.2V V_{DD} 导轨加电斜升之间的延迟。如果使用内部 VREG，请将 3.3V V_{DDIO} 电源的斜升时间缩短至 1ms 或更短。

公告	eCAP : HRFRC 不受 EALLOW 保护
受影响版本	0、A、B、D
详细信息	HRFRC 寄存器不受 EALLOW 保护。无需发出 EALLOW 和 EDIS 指令来写入该寄存器。 要在 HRFRC 受 EALLOW 保护的其他器件上启用软件重用，建议使用 EALLOW 和 EDIS。
解决方法	无

公告	PLL：第一次尝试锁定时，PLL 可能无法锁定
受影响版本	0、A、B、D
详细信息	在器件加电时，PLL 可能无法正常启动。PLLSTS[LOCKS] 位已置位，但 PLL 不会产生时钟。 一旦 PLL 正常启动，就可以禁用并重新启用 PLL，而不会出现任何问题，并且它会保持锁定。然而，PLL 锁定问题可能会在后续的加电周期中再次发生。 如果 SYSPLL 未正常启动且被选为 CPU 时钟源，CPU 将停止执行指令。 这一瞬态问题的发生率很低。在首次出现之后，该问题在系统中可能不会再次被观察到。实施此解决方法可降低该问题的发生率。
解决方法	TI 建议：在加电后首次配置 PLL 时，连续执行锁定序列，直到 PLL 进入锁定状态。锁定序列为：禁用 PLL、启动 PLL、等待将 LOCKS 位置位，以及使用双时钟比较器 (DCC) 验证 PLL 频率。观察到 PLL 正常运行后，可以选择它作为 CPU 时钟源。 TI 建议使用 C2000Ware SysCtl_setClock() 函数来设置 PLL 时钟，该函数还包括该解决方法实现。 有关 DCC 用法的详细信息，请参见 C2000Ware SysCtl_IsPLLValid() 函数。 如果器件无响应，也可以由监控器在系统级别复位该器件，从而应用该权变措施。

公告	闪存：在未禁用闪存预取的情况下执行 <code>Fapi_setActiveFlashBank()</code> 可能会导致 ITRAP
受影响版本	0、A、B、D
详细信息	闪存预取机制应仅通过从 RAM 执行的代码禁用，并且只能在没有对闪存进行任何有效获取访问时才能禁用。<code>Fapi_setActiveFlashBank()</code> 会禁用预取机制。当预取禁用过程正在进行时，任何闪存执行都可能会导致 ITRAP；因此，应在执行 <code>Fapi_setActiveFlashBank()</code> 之前禁用预取机制。
解决方法	在调用 <code>Fapi_setActiveFlashBank()</code> 之前禁用预取机制。可以在执行 <code>Fapi_setActiveFlashBank()</code> 后启用预取机制。用于禁用并启用预取机制的代码应从 RAM 中执行。在实施此解决方法后，请注意：由于预取机制已被禁用，<code>Fapi_setActiveFlashBank()</code> 可以从闪存中执行，而非从擦除/编程操作所针对的存储体执行。 C2000Ware_5_02_00_00 (及更高版本) 中提供的闪存 API 使用示例展示了此解决方法的实现情况。 TMS320F28003x 路径示例： C2000Ware_5_02_00_00\driverlib\f28003x\examples\flash\flashapi_ex1_programming.c 在本例中，<code>Flash_DisablePrefetch_SW_Workaround()</code> 从 RAM 中执行，以在调用 <code>Fapi_setActiveFlashBank()</code> 之前禁用预取机制。<code>Flash_EnablePrefetch_SW_Workaround()</code> 从 RAM 中执行，以在调用 <code>Fapi_setActiveFlashBank()</code> 之后启用预取机制。

公告 **LPM：不支持待机低功耗模式****受影响版本**
0、A、B、D**详细信息**
待机低功耗模式不受支持。**解决方法**
IDLE 或 HALT 低功耗模式可用于降低功耗。有关实现这些模式的信息，请参阅 [TMS320F28004x 实时微控制器技术参考手册](#)。
如果使用 IDLE 模式，可以通过软件选择以下一种或全部方法来额外降低功耗：

- 降低 SYSCLK 频率：
 - 通过配置 SYSPLLCTL1[PLLCLKEN]= 0，将 SYSCLK 时钟源更改为 OSCCLK。
 - 将 SYSCLKDIVSEL 寄存器更改为更高的分频值。
- 通过 PCLKCRx 寄存器禁用外设时钟。

公告 **DCC : 单次模式操作可能会过早终止**

受影响版本
0、A、B、D

详细信息
在单次触发模式下，可能会过早设置 DCCSTATUS[DONE] 或 DCCSTATUS[ERROR]。发生这种情况时，DCC 结果无效。

解决方法
以下任何条件都会过早终止 DCC 运行。如果满足以下 *任何* 条件，TI 建议重新运行 DCC。

- DCCSTATUS[DONE] = 1 且 (DCCCNT1 > 0 或 DCCCNT0 > 0 或 DCCVALID0 > 0)
- DCCSTATUS[ERROR] = 1 且 DCCCNT1 > 0 且 DCCVALID0 > 0

公告	BOR : 2.45V 至 3.0V 之间的 VDDIO 可产生多个 XRSn 脉冲
受影响版本	0、A、B、D
详细信息	当 VDDIO 电源电压介于 2.45V 和 3.0V 之间时，BOR 会生成重复的 XRSn 置为有效和置为无效。建议不要直接将 XRSn 引脚用作系统中任何其他器件的复位。 即使在这些 XRSn 脉冲发生时，F28004x BOR 也能有效地在内部将器件保持在已知的复位状态。在 VDDIO 供电电压升至 3.0 V 以上之前，该器件不会转移应用代码或引导加载程序，所有其他引脚都将保持其复位状态。
解决方法	1. 上电、断电和 BOR 事件期间，忽略额外的 XRSn 切换。额外的 XRSn 脉冲将不会影响 F28004x 器件本身的运行。 2. 如果 XRSn 脉冲会导致其他系统元件出现不良的系统行为，则请勿使用 XRSn 驱动其他器件。这些应用可使用外部电压监控器。 3. 对于需要在正常上电和断电期间避免这些脉冲的应用： a. 上电：请遵循 $t_{VDDIO-RAMP}$ 要求（位于 TMS320F28004x 实时微控制器 数据表的建议运行条件表中）；不会出现额外的 XRSn 低电平脉冲。 b. 断电：为避免 XRSn 在断电期间失效，在对电源进行设计时应确保 VDDIO 在 25 μs 内支持 3.0V 至 2.45V 的电压范围。如果 XRSn 上的电压上升是可以接受的，则可以计算 XRSn 上实现的 RC 电路的时间常数，以确保电压不会上升到系统指定的阈值以上。

公告

IC2 : SDA 和 SCL 漏极开路输出缓冲器问题

受影响版本

0、A、B、D

详细信息

SDA 和 SCL 输出通过推挽三态输出缓冲器而非 I2C 要求的漏极开路输出缓冲器来实现。虽然推挽三态输出缓冲器可以用作漏极开路输出，但内部时序偏斜问题会导致输出在禁用前驱动逻辑高电平约 0ns 至 5ns 的持续时间。仅当 SCL 或 SDA 输出从驱动的低电平状态转换为高阻抗状态，且相应的 I2C 输出有充足的内部时序偏斜时，才会出现意外的高电平脉冲。

这一短暂的高电平脉冲会在 I2C 信号走线中注入能量，从而导致 I2C 信号由于多次传输线路反射而持续一段时间的响铃。这种响铃不会导致 SDA 信号出现问题，因为它仅在预计 SDA 会改变逻辑电平时发生，并且响铃在接收器件锁存的数据之前有时间衰减。响铃幅度可能足以在此响铃期间的前几纳秒内多次超过 SCL 输入缓冲器开关阈值，从而可能导致时钟毛刺。如果响铃幅度在前 50ns 内衰减，则不会造成问题，因为系统会要求 I2C 器件对其 SCL 输入进行滤波以消除时钟毛刺。因此，设计 PCB 信号走线以将振铃的持续时间限制在 50ns 以内非常重要。一种可能的解决方案是，在 SCL 和 SDA 端子附近插入串联端接电阻器，以衰减传输线路反射。

从机开始其 ACK 周期时，此问题还可能导致 SDA 输出在意外高电平脉冲的持续时间内与从机 SDA 输出发生争用。发生这种情况是因为在发生意外的高电平脉冲之前，从机可能已经将 SDA 驱动为低电平。由于这段短暂的争用期而在 SDA 上产生的毛刺不会导致任何 I2C 协议问题，但峰值电流会对两个 I2C 器件施加不必要的应力，并可能导致电源噪声增加。因此，需要在相应的 SDA 端子附近放置一个串联端接电阻器，以限制短暂争用期间的电流。

当连接到支持时钟延展的 I2C 从机器件时，SCL 上也可能会出现类似的争用问题。发生这种情况是因为在发生意外的高电平脉冲之前，从机已将 SCL 驱动为低电平。由于这段短暂的争用期而在 SCL 上产生的毛刺不会导致任何 I2C 协议问题，因为 I2C 器件需要在其 SCL 输入端应用毛刺滤波器。但是，峰值电流会对两个 I2C 器件施加不必要的应力，并可能导致电源噪声增加。因此，需要在相应的 SCL 端子附近放置一个串联端接电阻器，以限制短暂争用期间的电流。

如果连接了另一个主器件，SCL 和 SDA 输出端的意外高电平脉冲可能会导致在时钟同步和仲裁期间发生争用。上面介绍的串联端接电阻器也将限制此用例中的争用电流，而不会产生任何 I2C 协议问题。

解决方法

在 SCL 和 SDA 信号端插入串联终端电阻器，并将其放在 SCL 和 SDA 终端附近。SCL 和 SDA 上拉电阻器也应放置在 SCL 和 SDA 终端附近。串联终端电阻器和上拉电阻器的放置方式应按 [图 3-4](#) 所示进行连接。

公告(续)

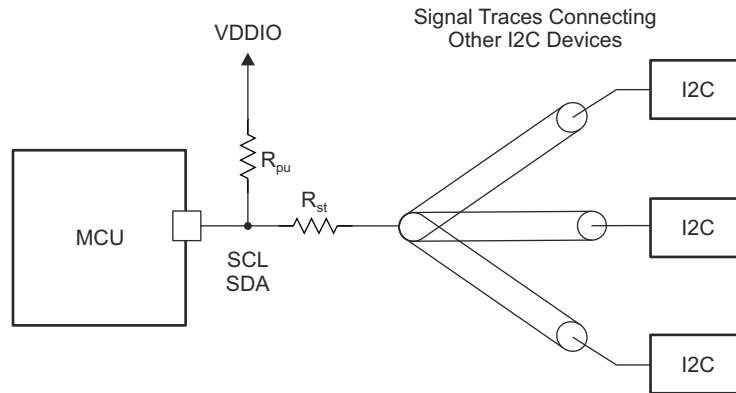
IC2 : SDA 和 SCL 漏极开路输出缓冲器问题

图 3-4. 串联终端电阻器和上拉电阻器的放置方式

公告

IC2：目标发送器模式、标准模式 SDA 时序限制

受影响版本

0、A、B、D

详细信息

MCU 上的 I2C 外设是一款快速模式器件；当与标准模式主机一同使用时，其将对 SCL（时钟）线路执行时钟延展。

I2C 规范对于标准模式系统中使用的快速模式器件有一项要求，即在释放 SCL 线路之前满足 $t_{\text{SU:DAT}}$ （数据设置时间）+ $t_{\text{r(max)}}$ （上升时间）。请参阅 NXP 半导体 *I²C 总线规范和用户手册(UM10204)* 中“标准、快速和快速模式 Plus I²C 总线器件的 SDA 和 SCL 总线特性”表的脚注 4。

然而，在上述情况下，C2000 I2C 时钟会将 SCL 线延展到一个固定量 = $6 * f_{\text{mod}}$ 时钟（C2000 的 I2C 时钟速率）。当 C2000™ 微控制器充当标准模式主机的目标发送器时，如果 SDA 的 t_{r} 过长，则 C2000 可能会在数据 (SDA) 就绪之前释放时钟线路 (SCL)。

NXP Semiconductors *I²C 总线规范和用户手册(UM10204)* 中的“上拉电阻器大小调整”部分提供了有关根据上升时间 (t_{r}) 和总线电容 (C_{b}) 选择适当 pu 电阻器 (R_{p}) 的更多详细信息，如 [方程式 1](#) 所示。

$$R_{\text{p(max)}} = \frac{t_{\text{r}}}{0.8473 \times C_{\text{b}}} \quad (1)$$

解决方法

1. 通过使用强上拉电阻来减小 t_{r}

为了确保满足 $t_{\text{SU:DAT}} + t_{\text{r(max)}}$ ，用户可以在 SDA 线路上配置上拉电阻，以便根据其系统中的 f_{mod} 时钟值，满足 [表 3-1](#) 的“SDA 数据上升时间要求”列中列出的限制。这将确保在 C2000 释放 SCL 信号时，SDA 线路上存在的数据有效。

[表 3-2](#) 给出了在给定 f_{mod} 时钟 (MHz) 和 C_{b} （总线电容）下的建议 R_{p} 电阻值。对于其他 C_{b} 值，请使用 [方程式 1](#) 来计算系统中所需的 R_{p} 值。

表 3-1. C2000 作为带有标准模式主机的目标发送器时的数据上升时间要求

f_{mod} 时钟 (MHz)	f_{mod} 周期 (ns)	C2000 I2C 的 SCL 时钟延展延迟 (ns) : ($6 * f_{\text{mod}}$ 时钟)	数据建立时间 (ns) : $t_{\text{SU:DAT}}$ (标准模式)	SDA 数据上升时间要求 (ns) : t_{r}
7	142.9	857	250	607
8	125	750		500
9	111	666		416
10	100	600		350
11	90.9	545		295
12	83.3	500		250

公告 (续)

IC2：目标发送器模式、标准模式 SDA 时序限制
表 3-2. 公共总线电容 (C_b) 的上拉电阻 (R_p) 值

f_{mod} 时钟 (MHz)	SDA 数据上升时间要求 (ns) : t_r	R_p (k Ω), 对于 $C_b = 100$ pF	R_p (k Ω), 对于 $C_b = 200$ pF	R_p (k Ω), 对于 $C_b = 300$ pF	R_p (k Ω), 对于 $C_b = 400$ pF
7	607	7.1	3.5	2.3	1.7
8	500	5.9	2.9	1.9	1.4
9	416	4.9	2.4	1.6	1.2
10	350	4.1	2.0	1.3	1.0
11	295	3.4	1.7	1.1	0.8
12	250	2.9	1.4	0.9	0.7

2. $T_r = 1000\text{ns}$

由于一般 I2C 使用方面的限制、因此不推荐使用此权变措施、请尽可能使用权变措施 1。

如果系统在 SDA 线路上需要 1000 ns 的上升时间、则可以将 C2000 I2C f_{mod} 时钟配置为 4.8 MHz，以便时钟延展 ($6 * f_{\text{mod}}$ 时钟) 满足该要求。由此可得， $t_r = (1/4.8 \text{ MHz}) * 6 = 1000 \text{ ns}$ 。此权变措施仅在 C2000 I2C 是 I2C 总线上目标的有效系统中有效。请注意，4.8 MHz 超出了数据表中 f_{mod} 时钟 7 MHz 至 12 MHz 的所需范围。在 4.8 MHz 时使用 f_{mod} ，尽管其已超出数据表的所需范围，也将适用于标准模式主机总线上的目标模式下的 C2000 I2C。除了此权变措施中列出的配置外，在任何其他配置中使用 $f_{\text{mod}} = 4.8 \text{ MHz}$ 都将导致违反其他时序参数的要求，因此不允许如此操作。

公告

ePWM：如果跳闸在消隐窗口结束时保持活动状态，则可能会发生 **ePWM** 干扰

受影响版本

0、A、B、D

详细信息

消隐窗口通常用于屏蔽转换期间将导致系统误跳闸的任何 PWM 跳闸事件。如果在消隐窗口周期结束后，ePWM 跳闸事件在少于三个 ePWM 时钟内保持活动状态，则 ePWM 输出上可能会存在不良干扰脉冲。

图 3-5 展示了可能会导致不良 ePWM 输出的时间段。

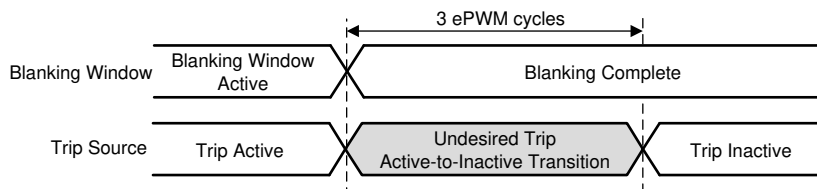


图 3-5. 不良跳闸事件和消隐窗口过期

图 3-6 展示了如果跳闸事件在消隐窗口关闭之前的 1 个周期或关闭之后的 3 个周期内结束，可能出现的两个 ePWM 输出。

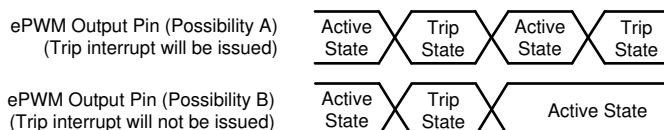


图 3-6. 可能会产生不良的 ePWM 输出

解决方法

扩展或减小消隐窗口，以避免任何不良的跳闸操作。

公告

ePWM：在启动消隐窗口后的前 3 个周期内，消隐窗口不会过滤跳闸事件

受影响版本

0、A、B、D

详细信息

在启动消隐窗口后的前 3 个周期内，消隐窗口不会消隐跳闸事件。DCEVTFILT 可能继续反映 DCxEVType 信号的变化。如果启用 DCEVTFILT，则可能会影响配置的后端子系统（例如跳闸区域子模块、TZ 中断、ADC SOC 或 PWM 输出）。

解决方法

在需要消隐之前的 3 个周期启动消隐窗口。如果在周期边界需要消隐窗口，则在下一周期开始前的 3 个周期启动消隐窗口。这是因为消隐窗口在整个期间边界中持续存在。

公告	ePWM : 单次/CBC 跳闸事件 DCxEVTy.force 不设置跳闸条件
受影响版本	0、A、B、D
详细信息	DCxEVTy.force 信号可用作单次跳闸事件和逐周期跳闸事件的输入。设置 DCxCTL[EVT1/2FRCSYNCSEL] = 1 (通过异步路径传递 DCxEVT 信号) 时, 不会正确设置单次或逐周期跳闸事件条件, 并且跳闸事件会丢失。
解决方法	异步路径 (通过异步路径传递 DCxEVT 信号) 要通过异步路径锁存单次或逐周期跳闸条件, 必须进行以下配置 : 1. EVT1/2FRCSYNCSEL = 1 2. EVT1/2LATSEL = 1 同步路径 (通过 SYNC 路径传递 DCxEVT 信号) 1. EVT1/2FRCSYNCSEL = 0 对于具有 DCxCTL.EVT1/2FRCSYNCSEL 和 DCxCTL.EVT1/2LATSEL 的器件, 存在两个选项。否则, 同步路径是唯一可用于单次或逐周期跳闸事件的选项。

公告

INTOSC : 在没有 VDD 的情况下为 VDDIO 加电会导致 INTOSC 频率漂移

受影响版本

0、A、B、D

详细信息

[TMS320F28004x 实时微控制器](#) 数据表 (SPRS945D) 的“D”修订版本更新了电源时序要求。在使用外部电源为 VDD 供电时，修订版本“C”和数据表的早期修订版本并未要求 VDDIO 和 VDD 同时加电和断电。

如果 VDDIO 已加电而 VDD 尚未加电，INTOSC1 和 INTOSC2 将会出现累积性的持续频率向下漂移。当处于高温环境时，如果 VDDIO 加电而 VDD 未加电，累积漂移的速率会更大。

由于这种漂移，INTOSC1 和 INTOSC2 内部振荡器频率可能下降到数据表中指定的最小值以下。这将影响那些使用 INTOSC2 作为 PLL 时钟源的应用，使系统以低于预期的频率运行。

解决方法

1. 使用内部 VREG 或内部 DCDC，这将确保在存在 VDDIO 时为 VDD 供电。
2. 使用外部 VDD 源时，始终保持 VDDIO 和 VDD 处于通电状态。
3. 使用外部 X1 和 X2 晶体振荡器作为 PLL 时钟源。该晶体振荡器没有与 VDDIO 和 VDD 电源时序相关的任何漂移。

公告 **FSI : RX FIFO 伪波超限****受影响版本**
0、A、B、D**详细信息**
当 FIFO 的最后一个位置被写入时，就会触发缓冲区溢出。**解决方法**
有两种可能的解决方法。

1. 在发送模块与接收模块之间建立通信时，应使读取第一个数据字前接收到的最大数据字之值为 15 (而不是 16)。在这种情况下，缓冲区溢出行为是可靠的。
2. 如果应用必须在读取第一个数据字之前填充接收缓冲区中的所有 16 个数据字 (即包含 16 个字的 NWORD 数据包)，则可以使用以下序列：
 - 忽略 RX 缓冲器溢出 RX_EVT_STS.BUF_OVERRUN。
 - 在 RX_EVT_STS.FRAME_DONE 时，读取 RX_BUF_PTR_STS.CURR_WORD_CNT 并检查其值是否为 16。
 - 使用 DMA 或软件将数据移出 RX 缓冲区。
 - 读取 RX_BUF_PTR_STS.CURR_WORD_CNT 并检查其值是否为 0。
 - 通过向 RX_EVT_CLR.FRAME_DONE 写入 1 来清除 RX_EVT_STS.FRAME_DONE 标志。

公告	在 DCAN FIFO 模式下，所接收到的消息可能以无序状态存放在 FIFO 缓冲器中
受影响版本	0、A、B、D
详细信息	在 DCAN FIFO 模式下，所接收到的、采用相同仲裁和掩码 ID 的消息应按照其被接收的先后顺序存放在 FIFO 中。然后 CPU 通过 IF1/IF2 接口寄存器从 FIFO 检索所接收到的消息。某些消息并不会按照其被接收的先后顺序，存放在 FIFO 中。如果消息的排列顺序对应应用程序进行处理至关重要，则此行为将阻止正确使用 DCAN FIFO 模式。
解决方法	使用 DMA 通过 IF3 寄存器读出 FIFO 。每次报文被接收到 FIFO 时，数据也会被复制到 IF3 寄存器中，并向 DMA 模块生成 DMA 请求以读出数据。

公告	引导 ROM : 从应用程序调用 SCI 引导加载程序
受影响版本	0、A、B、D
详细信息	ROM SCI 引导加载程序使用自动波特率来锁定波特率。SCI 波特率在两个寄存器 (SCILBAUD 和 SCIHBAUD) 之间拆分。ROM SCI 引导加载程序期望 SCIHBAUD 包含其默认复位值 0。如果 ROM SCI 引导加载程序被一个已将 SCIHBAUD 内容修改为非零值的应用程序调用, 则 SCI 将无法自动执行波特率锁定, 并且 SCI 引导加载程序将无法执行。
解决方法	在调用 ROM SCI 引导加载程序之前, 将 SCIHBAUD 清除为零。
公告	Boot-ROM、MPOST : 启用 MPOST 时引导时间更长
受影响版本	0、A、B、D
详细信息	当通过用户 OTP 启用 MPOST 功能时, boot-ROM 执行时间约为 5 秒。这是因为 boot-ROM 使用 10MHz 的 INTOSC 时钟来执行 MPOST, 从而导致引导时间更长。
解决方法	避免使用 boot-ROM 支持的 MPOST, 并在应用程序初始化过程中执行必要的存储器自检。

公告

内存：在有效存储器之外进行预取

受影响版本

0、A、B、D

详细信息

C28x CPU 预取指令的范围超出其流水线中当前正在活动的指令范围。如果预取发生在有效存储器结束之后，则 CPU 可能会接收到无效的操作码。

解决方法

M1、GS3 - 预取队列的深度为 8 x16 字。因此，代码不应在有效存储器末尾的 8 个字以内。可以在两个有效存储器块之间实现跨边界预取。

示例 1：M1 在地址 0x7FF 处结束，后面不跟随另一个存储器块。M1 中代码的存储地址不应超过 0x7F7。地址 0x7F8-0x7FF 不应用于代码。

示例 2：M0 结束于地址 0x3FF，有效存储器 (M1) 紧随其后。M0 中的代码可存储在 0x3FF 及以下的地址。代码也可以交叉进入 M1，最高到地址 0x7F7 (含地址 0x7F7)。

闪存 - 预取队列的深度为 16 x16 字。因此，代码不应在有效存储器末尾的 16 个字以内；否则，将生成闪存 ECC 不可纠正的错误。

表 3-3. 受公告影响的存储器

存储器类型	受影响的地址
M1	0x0000 07F8-0x0000 07FF
GS3	0x0001 3FF8 - 0x0001 3FFF
闪存	0x0009 FFF0 - 0x0009 FFFF

公告 **系统：系统控制寄存器配置限制****受影响版本**
0、A、B、D**详细信息**

在写入 CLKSRCCTL1 寄存器时，在写入操作之间多次未延迟，则系统可能会发生挂起的情况，并且只能通过外部 XRSn 复位或 Watchdog 复位才能恢复。在 SYSCLK 与 OSCCLKSRCSEL 所选时钟之间的时钟比有问题时，就会出现这种情况，然而，并非每次都会出现这个问题。

如果在使用调试器时遇到此问题，那么在点击暂停后，程序计数器将位于 Boot ROM 复位向量处。

实施此权变措施，可避免由 SYSCLK 与 OSCCLK 时钟比引起的这种情况。

解决方法

请参阅 [TMS320F28004x 实时微控制器技术参考手册 \(SPRU133\)](#) 的“系统控制寄存器配置限制”一节。

公告

ADC：如果未设置 *INTxCONT*（继续中断模式），中断可能会停止

受影响版本

0、A、B、D

详细信息

在 *ADCINTSELxNx[INTxCONT]* = 0 时，设置 *ADCINTFLG* 后，中断将停止，并且不会发生其他 ADC 中断。

若在 *ADCINTFLGCLR* 寄存器进行软件写入的同时发生 ADC 中断，则 *ADCINTFLG* 将意外保持为设置状态，阻止将来发生 ADC 中断。

解决方法

1. 使用“继续中断”模式，则 *ADCINTFLG* 无法阻止其他 ADC 中断：

```
ADCINTSEL1N2[INT1CONT] = 1;  
ADCINTSEL1N2[INT2CONT] = 1;  
ADCINTSEL3N4[INT3CONT] = 1;  
ADCINTSEL3N4[INT4CONT] = 1;
```

2. 为了避免发生这种情况，请确保下一次发生 ADC 中断之前，始终有足够的时间为 ADC ISR 提供服务并清除 *ADCINTFLG*。
3. 清除 *ADCINTFLG* 时，请检查 ISR 中是否存在溢出情况。在写入到 *ADCINTFLGCLR* 后立即检查 *ADCINTOVF*；如果已设置，则再次写入 *ADCINTFLGCLR* 以确保 *ADCINTFLG* 已被清除。若再设置 *ADCINTOVF* 寄存器，则表示已丢失 ADC 转换中断。

```
AdcaRegs.ADCINTFLGCLR.bit.ADCINT1 = 1;           //clear INT1 flag  
if(1 == AdcaRegs.ADCINTOVF.bit.ADCINT1)          //ADCINT overflow  
{  
    AdcaRegs.ADCINTFLGCLR.bit.ADCINT1 = 1;        //clear INT1 again  
    // If the ADCINTOVF condition will be ignored by the application  
    // then clear the flag here by writing 1 to ADCINTOVFCLR.  
    // If there is a ADCINTOVF handling routine, then either insert  
    // that code and clear the ADCINTOVF flag here or do not clear  
    // the ADCINTOVF here so the external routine will detect the  
    // condition.  
    // AdcaRegs.ADCINTOVFCLR.bit.ADCINT1 = 1;    // clear OVF  
}
```

公告

ADC : 使用 ADCCLK 小数分频器时 ADC 性能下降

受影响版本

0、A、B、D

详细信息

已证明使用分数 SYSCLK 转 ADCCLK 分频器 (由 ADCCTL2.PRESCALE 字段控制) 会导致该器件的 ADC 性能下降。请参阅 [表 3-4](#)。

表 3-4. ADCCTL2 寄存器

性能下降			
位	字段	值	说明
3-0	预分频	0001	ADCCLK = SYSCLK/1.5
		0003	ADCCLK = SYSCLK/2.5
		...	
性能正常			
位	字段	值	说明
3-0	预分频	0000	ADCCLK = SYSCLK/1.0
		0002	ADCCLK = SYSCLK/2.0
		...	

解决方法

使用偶数预分频时钟分频器值。偶数预分频值会产生整数时钟分频，而不会影响 ADC 性能。

公告

ADC : DMA 读取过时结果

受影响版本

0、A、B、D

详细信息

ADCINT 标志可在 ADCRESULT 值锁存前置位 (请参阅 t_{LAT} 和 $t_{INT(LATE)}$ 列，位于 [TMS320F28004x 实时微控制器](#) 数据表的 ADC 时序表中)。DMA 可在 ADCINT 触发置位后仅三个周期就读取 ADCRESULT 值。因此，若满足以下所有条件，当用户期望获得最新结果时，DMA 可能读取到之前的 ADCRESULT 值：

- ADC 处于延迟中断模式。
- ADC 在 $t_{INT(LATE)}$ 比 t_{LAT} 提前 3 个或更多周期的模式下工作 (ADCCTL2 [PRESCALE] > 2)。
- DMA 由 ADCINT 信号触发。
- DMA 在未先读取任何其他值的情况下，立即读取与该 ADCINT 信号关联的 ADCRESULT 值。
- DMA 在收到 ADCINT 触发时处于空闲状态。

仅上述 DMA 读取可能导致读取到陈旧数据；以下非 DMA 方法将始终读取预期数据：

- ADCINT 标志触发 CLA 任务。
- ADCINT 标志触发 CPU ISR。
- CPU 轮询 ADCINT 标志。

解决方法

从 ADCINT 标志触发两个 DMA 通道。第一个通道作为虚拟事务。这将产生足够的延迟，使得第二个通道总能读取到最新的 ADC 结果。

公告	CLB：不支持涉及多个有效高级控制器 (HLC) 通道的背对背 PUSH 或 PULL 指令
受影响版本	0、A、B、D
详细信息	如果使用具有多个活动 HLC 通道的背对背 PUSH 或 PULL 指令，此问题只会影响 HLC 操作。
解决方法	对 HLC 指令进行重新排序，以便每个 PUSH 指令都后跟一条非 PUSH 指令。对于 PULL 指令也是如此。如果只有一个 HLC 通道处于活动状态，则不会发生此问题。

公告	CMPSS : 在某些情况下, COMPxLATCH 可能无法正确清除
受影响版本	0、A、B、D
详细信息	CMPSS 锁存路径旨在在本地锁存器 (COMPxLATCH) 中保持跳闸状态, 直到由软件 (通过 COMPSTSCLR) 或 PWMSYNC 清除。 在信号经过数字滤波器数字化和鉴定后, COMPxLATCH 由比较器输出间接设置。比较器输出到达 COMPxLATCH 的预期最大延时可以用 CMPSS 模块时钟周期表示为: $\text{LATENCY} = 1 + (1 \times \text{FILTER_PRESCALE}) + (\text{FILTER_THRESH} \times \text{FILTER_PRESCALE})$ 当 COMPxLATCH 由软件或 PWMSYNC 清除时, 锁存本身根据需要清除, 但 COMPxLATCH 之前的数据路径可能不会反映额外延时数的模块时钟周期的比较器输出值。如果在 COMPxLATCH 被清除时数字滤波器输出解析为逻辑 1, 则锁存将在下一个时钟周期再次设置。
解决方法	在清除 COMPxLATCH 之前, 让数字滤波器输出解析为逻辑 0。 如果软件清除了 COMPxLATCH, 那么在清除锁存之前可以通过 COMPSTS 寄存器确认数字滤波器的输出状态。对于较大的延时值会产生不可容忍的延时的情况, 可以通过重新初始化数字滤波器 (通过 CTRIPxFILCTL) 来刷新滤波器 FIFO。 如果 COMPxLATCH 被 PWMSYNC 清除, 设计用户应用程序时应使得比较器跳闸条件在 PWMSYNC 生成前能至少清除延时周期。

公告

GPIO : 加电期间的 X2/GPIO18 引脚上拉电流

受影响版本

0、A、B、D

详细信息

在加电期间，将在 X2/GPIO18 上观察到约为 1.8mA 的上拉电流。在 XRSn 释放 (转换为高电平) 时，此引脚将正确恢复为输入模式并根据引脚说明运行。

解决方法

无

公告

GPIO : 漏极开路配置可以驱动短高电平脉冲

受影响版本

0、A、B、D

详细信息

每个 GPIO 都可以使用 GPxODR 寄存器配置为漏极开路模式。但是，内部器件时序问题可能会导致 GPIO 在进入或退出高阻抗状态期间将逻辑高电平驱动至高达 0 - 10 ns。

如果另一个驱动器同步驱动低电平，这种不必要的高电平可能会导致 GPIO 与线路上的另一个漏极开路驱动器发生争用。我们不希望出现此争用，因为它会对两个器件施加应力，并导致信号上出现短暂的中间电压电平。如果接收器逻辑中没有足够的逻辑滤波来滤除此短暂脉冲，则该中间电压电平可能会被错误地解读为高电平。

解决方法

如果存在争用问题，请勿使用 GPIO 的漏极开路功能；请改为在软件中仿真漏极开路模式。通过将 GPIO 数据 (GPxDAT) 设置为静态 0 并切换 GPIO 方向位 (GPxDIR) 来启用和禁用驱动低电平，可以实现漏极开路仿真。有关实现示例，请参阅以下代码。

```
void main(void)
{ ...

    // GPIO configuration
    EALLOW;
    GpioCtrlRegs.GPxPUD.bit.GPIOx = 1;    // disable pullup
    GpioCtrlRegs.GPxODR.bit.GPIOx = 0;    // disable open-drain mode
    GpioCtrlRegs.GPxODR.bit.GPIOx = 0;    // set GPIO to drive static 0 before
    GpioDataRegs.GPxCLEAR.bit.GPIOx = 1;  // enabling output
    EDIS;
    ...

    // application code
    ...

    // To drive 0, set GPIO direction as output
    GpioCtrlRegs.GPxDIR.bit.GPIOx = 1;

    // To tri-state the GPIO(logic 1),set GPIO as input
    GpioCtrlRegs.GPxDIR.bit.GPIOx = 0;
}
```

公告**LIN: 当同步字段错误时，同步字段错误 (ISFE) 标志/中断未设置****受影响版本**

0、A、B、D

详细信息

在 LIN 通信期间，如果接收到的同步字段（在 RX 上）是错误的（即同步字段接收到除 0x55 之外的任何值），则 LIN 不会设置 SCIFLR.ISFE 寄存器中的 ISFE 标志或触发 ISFE 中断。在未接收到数据或设置 RX 接收中断的情况下，通信会终止。应用程序无法检测同步字段中的错误。该应用程序可以检测同步字段是否为完全空或是否未在给定容差范围内接收到同步字段（如 [TMS320F28004x 实时微控制器技术参考手册](#) 中所述），但无法检测同步字段值中的任何错误。

解决方法

方法 1：如果未在特定时间内设置该标志，则继续轮询 SCIFLR.RXRDY 标志并触发超时。

将以下步骤作为指南执行：

1. 轮询要设置的 SCIFLR.BUSY 标志。
2. 一旦 BUSY 标志变为高电平，轮询 SCIFLR.RXRDY 标志。在此循环中同时还有一个 SW 超时，如果 RXRDY 标志未在用户定义的时间间隔内设置，则会超时并退出循环。

方法 2：将 CPU 计时器配置为在 RX 中断未被触发时中断。此方法不使用 CPU 带宽。

将以下步骤作为指南执行：

1. 配置 XINT 以在 LINRX 从高电平变为低电平时触发 ISR（表示 LIN 繁忙）。
2. 在 XINT ISR 内部配置 CPU 计时器，随后该计时器开始对帧完成计时。
3. 如果使用正确的同步字段正确接收到帧，则将触发 LIN RX ISR。您可以在其中关闭计时器，从而不会得到错误的超时。
4. 如果帧接收不正确，则不会触发 LIN RX ISR，而会触发 CPU 计时器 ISR（发生超时），这表示同步字段中存在错误。

公告

XTAL : 不支持 XTALCR[SWH]=0

受影响版本

全部

详细信息

《技术参考手册》的早期版本将 XTALCR[2] 记录为“SWH”，其有效设置为 0 (2 至 10MHz 范围) 或 1 (10 至 20MHz 范围)。在后续版本中，该位字段被标记为“保留”，且不应由应用程序予以修改。

数据表 XTAL 的有效电气频率范围为 10 至 20MHz，此范围自器件初始发布以来保持不变。

在器件版本 0、A 和 B 中，无论 XTALCR[2] (SWH) 设置为哪个值，XTAL 都能工作，但若设置为 0，启动时间可能会更长。

在器件修订版本 D 中，如果 XTALCR[2] (SWH) 设置为 0，则 XTAL 无法工作

解决方法

将 XTALCR[2] 保持为硬件默认设置值 1。

4 器件修订版本 B 使用说明和公告

本节列出了此器件修订版本的使用说明和公告。

4.1 器件修订版本 B 使用说明

在后续的器件修订版本中，可参阅器件修订版本适用的使用说明。有关更多详细信息，请参阅[器件修订版本 D 的使用说明](#)。

4.2 器件修订版本 B 公告

较晚的器件修订版本中已收录适用器件修订版本的公告。有关更多详细信息，请参阅[器件修订版本 D 公告](#)。

5 器件修订版本 A 使用说明和公告

本节列出了此器件修订版本的使用说明和公告。

5.1 器修订版本 A 使用说明

在后续的器件修订版本中，可参阅器件修订版本适用的使用说明。有关更多详细信息，请参阅[器件修订版本 D 的使用说明](#)。

5.2 器件修订版本 A 公告

较晚的器件修订版本中已收录适用器件修订版本的公告。有关更多详细信息，请参阅[器件修订版本 D 公告](#)。

公告	模拟子系统：共享基准引脚的软件配置
受影响版本	0、A
详细信息	F28004x 器件系列中较小的引脚数封装具有组合 VREFHI 引脚。ANAREFPP 寄存器中提供了软件配置位，用于禁用所有参考基准，但有一个组合参考基准除外。此举使得内部参考模式在这些情况下能够正常工作。在生产 (TMS) 器件上，引导 ROM 将写入这些位，并且用户无需采取进一步操作。但在某些 TMX 器件上，不会执行该写入操作。
解决方法	对于 TMX 器件，用户应执行一次以下写入操作，然后再尝试为内部参考基准模式配置参考基准： 100 引脚 PZ 封装：应将值 0x0002 写入 ANAREFPP。 64 引脚 PM 封装：应将值 0x0003 写入 ANAREFPP。 56 引脚 RSH 封装：应将值 0x0003 写入 ANAREFPP。

公告	某些 TMX 器件的模拟修整
受影响版本	0、A
详细信息	某些 TMX 样本可能未编写模拟修整程序。这可能会降低 ADC、缓冲 DAC、内部振荡器、PGA 和内部稳压器的性能。这些修整寄存器中的所有零值将产生以下影响。

TRIM	寄存器	等于零的修整值的影响
ADC 偏移	AdcaRegs.ADCOFFTRIM AdcbRegs.ADCOFFTRIM AdccRegs.ADCOFFTRIM	ADC 失调电压误差规格的性能下降。
ADC 基准	AnalogSubsysRegs.ANAREFTRIMA AnalogSubsysRegs.ANAREFTRIMB AnalogSubsysRegs.ANAREFTRIMC	ADC 对于所有规格的性能都会下降。没有可用的解决方法。
ADC 线性度	AdcaRegs.ADCINLTRIM2-3 AdcbRegs.ADCINLTRIM2-3 AdccRegs.ADCINLTRIM2-3	ADC 的 INL 和 DNL 规格已降级。没有可用的解决方法。
内部振荡器	AnalogSubsysRegs.INTOSC1TRIM AnalogSubsysRegs.INTOSC2TRIM	内部振荡器的频率精度降低和温度漂移。
缓冲 DAC 偏移	DacaRegs.DACTRIM DacbRegs.DACTRIM	缓冲 DAC 的失调电压误差规格下降。没有可用的解决方法。

公告 (续)

某些 TMX 器件的模拟修整

TRIM	寄存器	等于零的修整值的影响
PGA 增益和偏移	PGAGAIN3TRIM PGAGAIN6TRIM PGAGAIN12TRIM PGAGAIN24TRIM	PGA 增益和失调电压误差规格的性能下降。 没有可用的解决方法。

解决方法

以下解决方法可以用来提高性能，但它仍然可能不符合数据表规格。

可以按照 [TMS320F28004x 实时微控制器技术参考手册](#) 的“ADC 零偏移校准”一节中的说明，生成缺少的 **ADC 偏移修整**。

如果**内部振荡器修整**值全为零，用户可以在 1 (最小值) 到 1023 (最大值) 之间调整振荡器修整寄存器的最低 10 位，同时通过 XCLOCKOUT 引脚观察系统时钟。

公告	PGA：不支持输出滤波器路径
受影响版本	0、A
详细信息	PGA 模块包含一个嵌入式串联电阻器信号路径 (R_{FILTER})，用于在 PGA_OF 引脚上部署低通滤波器。不应使用此 R_{FILTER} 信号路径，也不应在受影响的修订版本上启用该路径。 PGA_OF 引脚上与 R_{FILTER} 共享的备选功能不受影响。例如，PGA1_OF 上的 ADC 输入信号 A2 和 B6 仍然可用。
解决方法	无
公告	ROM：不存在闪存 API 库和 FPU32 Twiddle Factor RFFT 表
受影响版本	0、A
详细信息	在受影响的修订版本中，Flash API 库以及用于 1024 点 RFFT 表的 FPU32 旋转因子并未存在于 ROM 中。有关 ROM 内容的详细信息，请参阅 TMS320F28004x 实时微控制器技术参考手册 中“ROM 代码和外设引导”一章的“内存映射”部分。
解决方法	无
公告	REVID：某些 TMX 修订版本 A 器件的 REVID 值不正确
受影响版本	A
详细信息	某些早期的 TMX 修订版本 A 器件在 REVID (地址 0x0005_D00C) 中存在不正确的值。REVID 错误地指示修订版本 0 值 (0x0000_0000)，而不是正确的修订版本 A 值 (0x0000_0001)。使用 REVID 来区分修订版本 0 和修订版本 A 的软件将不会按预期工作。此勘误表不会造成其他功能影响。不在软件中使用 REVID 的应用程序，将能与任何其他修订版本 A 器件一样正常工作。 受影响的批次追踪代码： • 65AVVDW • 66ALSXW
解决方法	封装上的器件标识正确无误，且可用于标识器件修订版本。

公告**GPIO：在输入模式下超过最大 V_{IH} 时， V_{SS} 的寄生路径****受影响版本****A****详细信息**

如果向下列 GPIO 引脚施加高于最大 V_{IH} 值 ($V_{DDIO} + 0.3V$) 的电压，则可能会开启一条从该引脚到 V_{SS} 的内部寄生路径。此寄生电流会影响引脚的功能运行。这种情况在高温下更有可能发生。当该 IO 被驱动至 V_{IL} 以下时，该寄生路径将被移除。该通路被消除后，除非再次发生过压事件，否则该路径不会重新激活。

- GPIO16
- GPIO17
- GPIO24
- GPIO25
- GPIO26
- GPIO27
- GPIO35 (TDI)
- GPIO37 (TDO)
- GPIO40
- GPIO41
- GPIO42
- GPIO43

配置为仅输出模式的引脚（该引脚上没有其他驱动器）将不会在引脚出现过压情形，因此不受此公告影响。

配置为输入或双向模式的引脚可通过三种主要方式出现过压情况：

1. 该输入由低阻抗驱动器驱动，由于阻抗不匹配且没有补偿终端，可能会在输入端产生较大的过冲。
2. 输入端受到外部噪声源带来的较大瞬态影响，使引脚处电压升高到超过 $V_{DDIO} + 0.3V$ 。
3. 该输入端由使用不同稳压器供电的器件驱动。当从另一个电压域接收电压时，系统设计应始终保持电压低于最大 V_{IH} 。但是，由于噪声事件或电源时序不当，电压暂时高于 $V_{DDIO} + 0.3V$ 的可能性会增加，则本公告仍将适用。

解决方法

如果上述任一情况适用于某个输入引脚或双向引脚，则应在信号与输入引脚之间插入一个串联电阻器。该串联电阻器应放置在靠近输入引脚的位置。

如果过压是由过冲引起（即上述第 1 种或第 2 种情况），则应使用 $100\ \Omega$ 或更大的串联电阻器。

如果过压可能呈现持续性（即上述第 3 种情况），则应使用 $220\ \Omega$ 或更大的串联电阻器。

6 器件修订版本 0 使用说明和公告

本节列出了此器件修订版本的使用说明和公告。

6.1 器件修订版本 0 使用说明

在后续的器件修订版本中，可参阅器件修订版本适用的使用说明。有关更多详细信息，请参阅[器件修订版本 D 的使用说明](#)。

6.2 器件修订版本 0 公告

较晚的器件修订版本中已收录适用器件修订版本的公告。有关更多详细信息，请参阅[器件修订版本 D 公告](#)和[器件修订版本 A 公告](#)。

公告	<i>GPIO：引脚在加电期间可以驱动为高电平</i>
受影响版本	0
详细信息	在加电期间，以下引脚将暂时处于输出模式并将驱动为高电平。在 XRSn 释放（转换为高电平）时，这些引脚将正确恢复为输入模式并根据引脚说明运行。 • GPIO0 • GPIO1 • GPIO2 • GPIO3 • GPIO4 • GPIO5 • GPIO11 • GPIO13 • GPIO29 • GPIO33
解决方法	无

公告 **GPIO : 信号门锁至 V_{SS}**

受影响版本 0

详细信息

下方所列引脚上的 ESD 结构可能会在正常运行期间被意外开启，从而将这些引脚拉低至 V_{SS} 。在这种情况下，每个输出引脚的 V_{DDIO} 电源上会产生约 40mA 的额外电流。

- GPIO16
- GPIO17
- GPIO24
- GPIO25
- GPIO26
- GPIO27
- GPIO35 (TDI)
- GPIO37 (TDO)
- GPIO40
- GPIO41
- GPIO42
- GPIO43

在正常工作条件下，尚未观察到这种情况在 70°C 以下发生。这种情况可能发生在输入或输出模式下，以及任何多路复用功能中。具有轻负载引脚和快速开关信号的设计更可能遇到这种情况。如果在较小的引脚数封装中进行切换，未进行键合的引脚也会进入门锁状态。

通过在较低温度下切换该 IO，可以结束门锁状态。

解决方法

四种解决方法选项包括：

1. 避免在受影响的修订版本上使用这些引脚。
2. 避免在受影响的修订版本上执行高温操作。
3. 如果引脚配置为输入或输出：

在这些引脚和接地端之间放置一个 300pF 或更大的电容器，并尽可能靠近该器件放置。这将减慢快速信号，从而避免触发该情况。更大的电容在滤除瞬态方面更有效，但必须与这些引脚的系统级时序要求进行权衡。

与选项 4 结合使用时，可以针对这些输入引脚使用较小的电容器。

4. 如果该引脚配置为输入：

将一个电阻器与电路板上的任何其他元件串联，以使驱动器与该电阻器的总电阻达到 1k Ω 或更高。这样做的目标是消除引脚处出现的快速电压瞬态。如果 ESD 结构因噪声而被激活，这也会限制直流电流。

公告	ePWM：选择异步路径后，“基于 DC 事件的 CBC 跳闸”的事件门锁 (DCxEVTxLAT) 可能不会按预期延长触发脉冲
受影响版本	0
详细信息	在置为无效时，DCxEVTxLAT 可能会丢失异步输入的已捕获触发事件。当异步触发器被置为无效时，预期触发器会保持该值，直到发生清除事件。由于该触发信号相对于触发器的时钟是异步的，因此该触发器可能在触发信号置为无效期间被清除。这将导致事件锁存功能丧失。
解决方法	无

7 文档支持

欲了解有关 F28004x 器件的更多信息，请参阅以下文档：

- [TMS320F28004x 实时微控制器数据表](#)
- [TMS320F28004x 实时微控制器技术参考手册](#)

8 商标

TMS320™ is a trademark of Texas Instruments.

C2000™ is a trademark of Texas Instruments.

所有商标均为其各自所有者的财产。

9 修订历史记录

Changes from February 23, 2024 to August 31, 2026 (from Revision H (February 2024) to Revision I (August 2026))

	Page
• 通篇：添加了器件修订版本 D。.....	1
• PM 和 PZ 封装的器件标识示例图：更新了 F280049PZS 图。.....	6
• 增加了 闪存：在未禁用闪存预取的情况下执行 <code>Fapi_setActiveFlashBank()</code> 可能会导致 ITRAP 公告。.....	21
• 增加了 ePWM：单次/CBC 跳闸事件 <code>DCxEVTy.force</code> 不设置跳闸条件公告。.....	30
• 更新了解决方法.....	36
• 添加了 CMPSS：在某些情况下 <code>COMPxLATCH</code> 可能无法正确清除的公告。.....	40
• 添加了 XTAL：不支持 <code>XTALCR[SWH]=0</code> 公告。.....	43

重要通知和免责声明

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、与某特定用途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安全、安保法规或其他要求。

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 产品的相关应用。严禁以其他方式对这些资源进行复制或展示。您无权使用任何其他 TI 知识产权或任何第三方知识产权。对于因您对这些资源的使用而对 TI 及其代表造成的任何索赔、损害、成本、损失和债务，您将全额赔偿，TI 对此概不负责。

TI 提供的产品受 [TI 销售条款](#)、[TI 通用质量指南](#) 或 [ti.com](#) 上其他适用条款或 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。除非德州仪器 (TI) 明确将某产品指定为定制产品或客户特定产品，否则其产品均为按确定价格收入目录的标准通用器件。

TI 反对并拒绝您可能提出的任何其他或不同的条款。

版权所有 © 2026，德州仪器 (TI) 公司

最后更新日期：2025 年 10 月