

Errata

MSPM0L110x、MSPM0L13xx 微控制器



摘要

本文档介绍了功能规格的已知例外情况（公告）。

内容

1 勘误注释.....	1
2 功能公告.....	1
3 预编程软件公告.....	3
4 仅调试公告.....	3
5 编译器修复公告.....	3
6 器件命名规则.....	3
6.1 器件编号法和修订版本标识.....	4
7 公告说明.....	5
8 修订历史记录.....	33

1 勘误注释

在阅读勘误表和条目之前，应考虑以下注意事项：

- ADC_ERR_04 不再影响器件
- FLASH_ERR_04 不再影响器件

2 功能公告

影响器件运行、功能或参数的公告。

✓ 复选标记表示指定版本中存在该问题。

勘误编号	修订版 C	修订版 D
ADC_ERR_01	✓	✓
ADC_ERR_02	✓	✓
ADC_ERR_03	✓	✓
ADC_ERR_04	✓	
ADC_ERR_05	✓	✓
ADC_ERR_06	✓	✓
ADC_ERR_11	✓	✓
COMP_ERR_01	✓	✓
COMP_ERR_02	✓	✓
COMP_ERR_03	✓	✓
COMP_ERR_05	✓	✓
CPU_ERR_01	✓	✓
CPU_ERR_02	✓	✓
CPU_ERR_03	✓	✓
CPU_ERR_04	✓	✓
CRC/CRCP_ERR_01	✓	✓

勘误编号	修订版 C	修订版 D
FCC_ERR_01	✓	✓
FLASH_ERR_02	✓	✓
FLASH_ERR_03	✓	✓
FLASH_ERR_04	✓	
FLASH_ERR_05	✓	✓
FLASH_ERR_06	✓	✓
FLASH_ERR_08	✓	✓
GPIO_ERR_01	✓	✓
GPIO_ERR_02	✓	✓
GPIO_ERR_03	✓	✓
GPIO_ERR_04	✓	✓
GPIO_ERR_06	✓	✓
I2C_ERR_01	✓	✓
I2C_ERR_02	✓	✓
I2C_ERR_03	✓	✓
I2C_ERR_04	✓	✓
I2C_ERR_05	✓	✓
I2C_ERR_06	✓	✓
I2C_ERR_07	✓	✓
I2C_ERR_08	✓	✓
I2C_ERR_09	✓	✓
I2C_ERR_10	✓	✓
I2C_ERR_13	✓	✓
I2C_ERR_15	✓	✓
PMCU_ERR_01	✓	✓
PMCU_ERR_02	✓	✓
PMCU_ERR_03	✓	✓
PMCU_ERR_10	✓	✓
PMCU_ERR_13	✓	✓
PWREN_ERR_01	✓	✓
RST_ERR_01	✓	✓
SPI_ERR_01	✓	✓
SPI_ERR_03	✓	✓
SPI_ERR_04	✓	✓
SPI_ERR_05	✓	✓
SPI_ERR_06	✓	✓
SPI_ERR_07	✓	✓
SPI_ERR_08	✓	✓
SPI_ERR_09	✓	✓
SPI_ERR_10	✓	✓
SYSCTL_ERR_03	✓	✓
SYSCTL_ERR_05	✓	✓
SYSCTL_ERR_06	✓	✓
SYSCTL_ERR_11	✓	✓
SYSOSC_ERR_01	✓	✓
SYSOSC_ERR_02	✓	✓

勘误编号	修订版 C	修订版 D
SYSOSC_ERR_04	✓	✓
SYSOSC_ERR_05	✓	✓
SYSOSC_ERR_06	✓	✓
TIMER_ERR_01	✓	✓
TIMER_ERR_04	✓	✓
TIMER_ERR_06	✓	✓
TIMER_ERR_07	✓	✓
UART_ERR_01	✓	✓
UART_ERR_02	✓	✓
UART_ERR_04	✓	✓
UART_ERR_05	✓	✓
UART_ERR_06	✓	✓
UART_ERR_07	✓	✓
UART_ERR_08	✓	✓
UART_ERR_09	✓	✓
UART_ERR_10	✓	✓
UART_ERR_11	✓	✓
VREF_ERR_01	✓	✓

3 预编程软件公告

影响出厂编程软件的公告。

✓ 复选标记表示指定版本中存在该问题。

4 仅调试公告

仅影响调试操作的公告。

✓ 复选标记表示指定版本中存在该问题。

5 编译器修复公告

由编译器权变措施解决的公告。请参阅每个公告，以了解 IDE 和编译器版本及权变措施。

✓ 复选标记表示指定版本中存在该问题。

6 器件命名规则

为了标示产品开发周期所处的阶段，TI 为所有 MSP MCU 器件的器件型号分配了前缀。每个 MSP MCU 商用系列产品都具有以下两个前缀之一：MSP 或 XMS。这些前缀代表了产品开发的发展阶段，即从工程原型 (XMS) 直到完全合格的生产器件 (MSP)。

XMS - 实验器件，不一定代表最终器件的电气规格

MSP - 完全合格的生产器件

支持工具命名前缀：

X：还未经德州仪器 (TI) 完整内部质量测试的开发支持产品。

null：完全合格的开发支持产品。

XMS 器件和 X 开发支持工具在供货时附带如下免责条款：

“开发中的产品用于内部评估用途。”

MSP 器件的特性已经全部明确，并且器件的质量和可靠性已经完全论证。TI 的标准保修证书对该器件适用。

预测显示原型器件 (XMS) 的故障率大于标准生产器件。由于这些器件的预计最终使用故障率尚不确定，德州仪器 (TI) 建议不要将它们用于任何生产系统。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。此后缀表示温度范围、封装类型和配送形式。

6.1 器件编号法和修订版本标识

下面的封装图展示了封装编号法方案，表 6-1 定义了器件修订版到版本 ID 的映射。

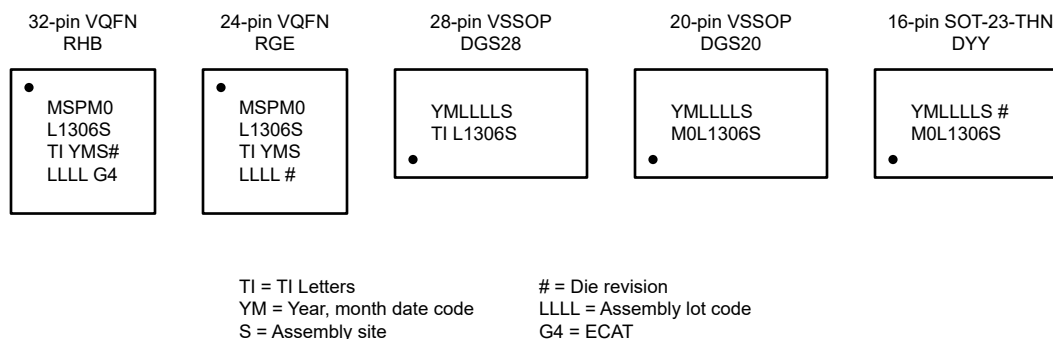


图 6-1. 封装符号

表 6-1. 芯片修订版本

修订版字母	版本 (在器件的出厂常量存储器中)
C	1
D	1

修订版字母表示产品硬件修订版本。根据修订字母，本文档中的公告标记为“适用于”或“不适用于”给定器件。该字母映射到器件存储器中存储的整数，可用于使用应用软件或已连接的调试探针来查找修订版本。

7 公告说明

ADC_ERR_01 **ADC 模块**

类别

功能

功能

ADC 在 STANDBY1 模式下无法触发快速时钟

说明

当器件在 STANDBY1 模式下运行时，ADC 模块通过事件系统触发异步快速时钟请求时（例如，当定时器等事件发布者通过 ADC 的事件订阅器端口向 ADC 生成事件时），可能无法正确发起异步快速时钟请求。

权变措施

通过事件结构触发 ADC 转换时，使用 STANDBY0 或更高的功率模式。

ADC_ERR_02 **ADC 模块**

类别

功能

功能

当 ADC 处于重复模式时，低功耗模式下的电流增加

说明

当 ADC 在低功耗模式 (LPM) 下配置为带 EVENT 触发器的重复序列或重复单次模式时，ADC 模块将消耗额外电流，直至禁用。所看到的电流值将与数据表中的 SLEEP0 电流特性相当。

权变措施

权变措施 1：将 ADC 配置为单次或序列模式（无重复），并在转换后使用中断为下一次触发重新启用 ADC。权变措施 2：进入 LPM 模式时，复位 ADC 或更改 ADC 的配置。建议在进入 LPM 时不要使用重复模式。

ADC_ERR_03 **ADC 模块**

类别

功能

功能

当将 VSSA 用作 ADC 接地基准时，ADC SNR 和 DNL 性能会更差。

说明

当 ADC 以 VSS 作为接地基准时，VSS 接地端的噪声源将影响 ADC 结果，从而导致 SNR 和 DNL 性能降低。

权变措施

权变措施 1：使用 VREF+ 和 VREF- 引脚作为 ADC 基准。通过 VREF- 引脚，ADC 将具有其模拟接地基准，从而减少外部噪声源的影响。权变措施 2：减少 VSS 上的噪声源；常见的噪声源是 CPU 或数字开关噪声

ADC_ERR_04	ADC 模块
类别	功能
功能	当 ADC 处于重复模式时，低功耗模式下的电流增加
说明	当 ADC 在低功耗模式 (LPM) 下配置为带 EVENT 触发器的重复序列或重复单次模式时，ADC 模块将消耗额外电流，直至禁用。
权变措施	将 ADC 配置为单次或序列模式 (无重复)，并在转换后使用中断为下一次触发重新启用 ADC。
ADC_ERR_05	ADC 模块
类别	功能
功能	启用 IP 之前生成的 HW 事件，ADC 触发器将保持在队列中
说明	当 ADC 收到 HW 事件触发信号时，即使 CTL0.ENC = 0x0，待处理的事件也将进入 ADC 触发队列。一旦 ADC 设置为 CTL1.TRIGSRC = 0x1 (HW 触发) 且 CTL0.ENC = 1，排队的 HW 触发信号将启动 ADC 采样和转换过程。即使 CTL1.TRIGSRC = 0x0 (SW 触发)，待处理的 HW 请求也可进入队列，但仅在 CTL1.TRIGSRC = 0x1 && CTL0.ENC = 0x1 时，才会启动 ADC 采样。
权变措施	仅当需要使用 HW 触发时，才配置 ADC F_SUB；否则，待处理的请求可能会排入队列。如果在 SW 和 HW 触发模式之间切换，ADC (RSTCTL) 上的复位将清除所有待处理的队列，但需要重新配置 ADC。
ADC_ERR_06	ADC 模块
类别	功能
功能	ADC 输出代码跳变，从而使 DNL/INL 规格降低
说明	出现转换误差时，ADC 的数字输出代码会出现 +/- 64LSB 的固定跳变，而 ADC 输入电压则不会发生相应变化。 在最坏情况 (- 40C) 下，在 12 位模式下，12M 个转换样本中的误差率为 1。 在 0C 下，误差率为 1/24M 在 55C 下，误差率为 1/60M (使用的 VDD 电压和基准对勘误差率没有影响)
权变措施	根据具体应用需求，适用的权变措施可能会有所不同，但建议在软件中采用以下权变措施。适用的权变措施的选择应由系统设计人员根据实际情况自行判断。

ADC_ERR_06

(续)

ADC 模块

权变措施 1：当 ADC 结果超出应用阈值 (通过 ADC 窗口比较器或软件阈值) 时，触发或等待另一个 ADC 结果，然后再做出关键的系统决策

权变措施 2：在后处理过程中，剔除与中值或预期值相差太大的 ADC 值。预期值应基于系统中实际采样的平均值，剔除阈值应基于测量的系统噪声大小。

权变措施 3：使用 ADC 样本平均法，以尽可能降低任何单次错误转换结果的影响。

ADC_ERR_11

ADC 模块

类别

功能

功能

当处于 MEMCTL.TRIG = 0x1 的重复模式时，ADC Busy 保持高电平，ADC ENC 不会被清零

说明

当处于 MEMCTL.TRIG = 0x1 的重复单次或重复序列模式时 (下一次转换需要触发)。当 ADC 正在等待新的触发信号时，ADC 保持 BUSY 状态。这会导致在显式写入 0 时 ENC 位不会被清除。

权变措施

权变措施 1：复位 ADC 外设 ADCx.RSTCTL = 0xB1000003，以清除 BUSY 状态和 ENC 位。需要重新配置

解决方法 2：将 ADC 设置为单模式 ADCx.CTL1.CONSEQ = 0x0 或 0x1。然后将最后一个存储器控制寄存器 MEMCTL[Y].TRIG 设置为 0x0 (下一次转换自动)。ADC 将对下一个通道进行采样，但不会在之后重复，这将允许 BUSY 状态和 ENC 清除。

COMP_ERR_01

COMP 模块

类别

功能

功能

比较器输出在 STANDBY0 模式下持续切换

说明

当比较器反相输入多路复用器 (IMSEL) 设置为 0 (COMP0_IN0-)，且器件工作模式设置为 STANDBY0 时，无论向比较器施加的输入电压如何，比较器输出都可能意外切换。

权变措施

如果在 STANDBY0 模式下使用比较器，并向负极施加外部电压，则应使用比较器的通道 1 (IMSEL = 1)。

如果比较器处于 STANDBY0 状态，并施加内部基准电压 (DAC8、VREF 等)，则将 IMSEL 设置为 0 以外的值 (IMSEL 0)。

COMP_ERR_02	COMP 模块
类别	功能
功能	当 DACCODEx 用于滞后时，COMP 输出会切换
说明	若使用 COMP 模块中的 8 位 DAC 作为 COMP 的输入，且 DAC 输出在 DACCODEx 中的值之间切换，则 COMP 输出会切换，如同立即越过新的参考点。 无论 COMPx.CTL2.DACCTL 位的设置如何，都会出现这种情况。 这种情况最常见于利用两个 DACCODEx 代码为 COMP 模块提供自定义或不对称迟滞的应用中。
权变措施	利用 COMPx.CTL1.HYST 寄存器位中提供的既定迟滞值。
COMP_ERR_03	COMP 模块
类别	功能
功能	使用输入交换功能时，COMP 迟滞功能不起作用
说明	使用 COMP 模块的迟滞功能并交换 COMP 的输入时 (COMPx.CTL1.EXCH = 1)，COMP 模块变得不稳定。
权变措施	在输入交换功能中使用 COMP 模块时，请勿使用内部迟滞方法。
COMP_ERR_05	COMP 模块
类别	功能
功能	启用比较器时，比较器输出将设置上升和下降中断
说明	启用比较器时，比较器将设置上升和下降。
权变措施	1. 使用 ICLR 位清除 CPU 中断。 ICLR 不能用于清除一般事件。按照以下步骤清除 COMP 一般事件 (以下是 DriverLib 函数，您可以通过查看我们 MSPM0 SDK 中的函数内容来查看位操作) a. 在启用 COMP 之前，使用一些虚拟 ID 来配置 COMP 发布者。 DL_COMP_setPublisherChanID(COMP_0_INST, 0); // 删除实际发布者

COMP_ERR_05

(续)

COMP 模块

- b. DL_COMP_enableEvent(COMP_0_INST, (DL_COMP_EVENT_OUTPUT_EDGE)); // 在 IMASK 中启用 COMP 事件
- c. DL_COMP_enable(COMP_0_INST); // 启用 COMP 模块，该步骤会清除 RIS 中的事件。
- d. DL_COMP_disableEvent(COMP_0_INST, (DL_COMP_EVENT_OUTPUT_EDGE)); // 通过在 IMASK 中清除来禁用 COMP 事件
- e. DL_COMP_setPublisherChanID(COMP_0_INST, COMP_0_INST_PUB_CH); // 配置实际发布者
- f. DL_COMP_enableEvent(COMP_0_INST, (DL_COMP_EVENT_OUTPUT_EDGE)); // 在 IMASK 中重新启用 COMP 事件，或在了解首次中断因启用比较器造成的情况下，在启用比较器后读取中断。

CPU_ERR_01

CPU 模块

类别

功能

功能

在主闪存和其他闪存区域之间切换时，CPU 高速缓存内容可能会损坏。

说明

在访问主闪存存储器和其他非易失性内存区域（如 NONMAIN 或 Factory 区域）之间切换时，可能发生高速缓存损坏。

权变措施

- 使用以下步骤安全访问主内存以外的区域：
1. 设置 CPUSS.CTL.ICACHE = 0x0，以禁用高速缓存。
 2. 从 SHUTDOWN 存储器读取 SYSTCL.SOCLOCK.SHUTDNSTORE0
 3. 对 NONMAIN 或 Factory 区域存储器执行所需的访问。
 4. 设置 CPUSS.CTL.ICACHE = 0x1 以重新启用高速缓存

CPU_ERR_02

CPU 模块

类别

功能

功能

禁用 CPUSS 预取/缓存具有限制

说明

如果存在待处理的闪存访问，CPU 预取/缓存禁用将不会生效

权变措施

禁用高速缓存和预取器（顺序无强制要求），然后对 SYSTCL 中的关断存储器 (SHUTDNSTORE) 发出存储器访问（更多参考信息，请查看器件 TRM 中的 SHUTDNSTORE 寄存器）。

存储器访问完成后，预取器/高速缓存将被禁用。

示例：

CPUSS->CTL = (CPUSS_CTL_PREFETCH_DISABLE |

CPU_ERR_02

(续)

CPU 模块

```
CPUSS_CTL_ICACHE_DISABLE); //禁用指令缓存和预取
DL_SYSCTL_setShutdownStorageByte(DL_SYSCTL_SHUTDOWN_STORAGE_BYTE_X
, data) // 将一个字节保存到 SHUTDOWN 存储器
```

CPU_ERR_03**CPU 模块****类别**

功能

功能

在转换到低功耗模式时，预取器可能会获取错误的指令

说明

转换到低功耗模式且存在待处理的预取时，预取器可能会错误地获取不正确的数据（全 0）。当器件唤醒时，如果预取器和高速缓存未被 ISR 代码覆盖，则从闪存执行的主代码可能会损坏。例如，如果 ISR 位于 SRAM 中，则从闪存预取的不正确的数据不会被覆盖。当 ISR 返回损坏的数据时，CPU 可能会提取预取器中的数据，从而导致指令不正确。硬件事件唤醒是将唤醒器件但不刷新预取器的进程的另一个示例。

权变措施

进入低功耗模式之前禁用预取器。

示例：

```
CPUSS->CTL &= 0x6; // 禁用预取器，保持其他设置
SYSCTL.SOCLOCK.SHUTDNSTORE0 // 从 SHUTDOWN 存储器读取
__WFI(); // or __WFE(); 该函数调用转换到低功耗模式
CPUSS->CTL |= 0x1; // 启用预取器
```

CPU_ERR_04**CPU 模块****类别**

功能

功能

高速缓存无法从导致硬故障的 FLASH 地址返回正确的数据

说明

如果器件因某个 FLASH 地址进入硬故障状态，在器件从该硬故障恢复并尝试从其高速缓存（存储来自该 FLASH 地址的数据）中检索信息后，返回值变为零。此外，如果访问相同的 FLASH 地址，则不会触发新的硬故障。

权变措施

通过更改 CPUSS.CTL.ICACHE 位禁用然后重新启用高速缓存。

示例：

```
CPUSS->CTL &= ~(CPUSS_CTL_ICACHE_ENABLE); ///禁用指令缓存
CPUSS->CTL |= CPUSS_CTL_ICACHE_ENABLE; ///启用指令缓存
```

CRC/ CRCP_ERR_01

CRC/CRCP 模块

类别

功能

功能

在挂起的 LPM 下，无法触发 DMA 访问 CRC/CRCP 模块

说明

在 RUN0/1/2 和 SLEEP0/1/2 模式下，DMA 可以正常访问 CRC/CRCP 模块。在 STOP/STANDBY 模式下，任何触发 DMA 的操作都会使器件进入挂起的低功耗状态，在这种模式下，DMA 无法访问某些器件的 CRC/CRCP 模块。

权变措施

1.如果需要通过 DMA 访问 CRC/CRCP，请不要使进入 STOP/STANDBY 模式。2.在 STOP/STANDBY 模式下，使用其他唤醒功能（例如中断）使器件退出 STOP/STANDBY 模式，然后触发 DMA 访问 CRC/CRCP，或使用 CPU 代替 DMA 来访问 CRC/CRCP。

FCC_ERR_01

FCC 模块

类别

功能

功能

当 BUSCLK 以 LFCLK 为源时，FCC 行为异常

说明

当 BUSCLK 以 LFCLK 为源时，FCC 无法按预期工作。表现为 FCC 完成信号未置为有效，或者报告的 FCC 值不正确。

权变措施

无权变措施

FLASH_ERR_02

FLASH 模块

类别

功能

功能

可以使用密码重新启用 NONMAIN 中的调试禁用

说明

如果在 NONMAIN 配置中禁用了调试 (DEBUGACCESS = 0x5566)，则仍可以使用编程的密码访问器件（如果未明显编程，则使用默认密码）。

权变措施

权变措施 1：

将 DEBUGACCESS 设置为使用密码启用调试选项 (DEBUGACCESS = 0xCCDD)，并在 PWDDEBUGLOCK 字段中提供唯一的密码。为了提高安全性，建议使用加密随机的器件唯一密码。这将允许使用正确的 128 位密码进行调试访问，但仍可允许某些调试命令以及对 CFG-AP 和 SEC-AP 的访问。

权变措施 2：

通过禁用 SWDP_MODE 来完全禁用物理 SW 调试端口。这可完全防止对器件进行任何调试访问或发送申请，但可能会影响故障分析和返回流程。

FLASH_ERR_03 *FLASH 模块*

类别	功能
功能	在 2 个等待状态下访问闪存，然后访问无效引导代码，将导致下一次闪存访问也出现违规
说明	在 2 个等待状态下访问闪存，然后访问 BOOTCODE，将导致下一次闪存访问也出现违规。
权变措施	请勿尝试在启动阶段后访问引导代码区域。否则，在引导代码违规和下一次正确闪存访问之间需要间隔 4 个时钟周期。

FLASH_ERR_04 *FLASH 模块*

类别	功能
功能	如果错误未发生在主闪存区域，则 SYSCTL_DEDERRADDR 中会报告错误地址。
说明	
权变措施	如果 SYSCTL_DEDERRADDR 的返回地址返回 0x00Cxxxxx，请使用 0x41000000 进行“OR”操作，以获取 NONMAIN 或 Factory 区域返回地址的正确地址。例如，如果 SYSCTL_DEDERRADDR = 0x00C4013C，则实际地址为 0x41C4013C。 如果 SYSCTL_DEDERRADDR 的返回地址返回 0x00Dxxxxx，请使用 0x41000000 进行“OR”操作，以获取数据银行返回地址的正确地址。例如，如果 SYSCTL_DEDERRADDR = 0x00D0012A，则实际地址为 0x00D0012A。

FLASH_ERR_05 *FLASH 模块*

类别	功能
功能	DEDERRADDR 可能具有不正确的复位值
说明	SYSCTL->DEDERRADDR 的复位值可能返回 0x00C4013C，而不是正确的 0x00000000。错误发生在出厂微调区域，并不表示出现故障，可适当忽略。在器件上对 NONMAIN 进行编程后，复位值通常会发生变化。
权变措施	接受 0x00C4013C 作为另一个复位值，因此引导后的默认值可能为 0x00000000 或 0x00C4013C。返回值超出了器件 MAIN 闪存的范围，因此该返回值不可能来自实际的 FLASH DED 状态。

FLASH_ERR_06 *闪存模块*

类别	功能
----	----

FLASH_ERR_06

(续)

闪存模块

功能

CPU 和 DMA 无法同时访问闪存

详细信息

CPU 和 DMA 无法同时访问闪存；这种同时访问会导致从闪存读取不正确的数据。

权变措施

请勿同时通过 CPU 和 DMA 访问闪存。在进行编程/擦除操作、读验证/空白验证操作以及从闪存读取 DMA 数据等典型闪存操作时，软件需确保 CPU 不会在闪存繁忙时访问闪存。为此，可在闪存操作正在进行时将代码放入 SRAM，或将闪存移入 SRAM 以读取 DMA 需要读取的数据。

FLASH_ERR_08

FLASH 模块

类别

功能

功能

不会为典型的无效存储器区域生成硬故障

说明

在尝试访问如下所示的非法存储器地址空间时，不会生成硬故障：1.0x010053FF - 0x20000000 2.0x40BFFFFFF - 0x41C00000 3.0x41C007FF - 0x41C40000

权变措施

否

GPIO_ERR_01

GPIO 模块

类别

功能

功能

在 STANDBY 模式下 GPIO 唤醒边沿可能丢失

说明

通过单个 GPIO 边沿唤醒一次后，在 STANDBY/STOP/SLEEP 模式下可能会错过后续 GPIO 唤醒边沿。

情况 1：

STANDBY0 唤醒——如果 MCU 设置为 STANDBY/STOP/SLEEP 模式，且 IO 处于“唤醒”状态，然后将 IO 设回“非唤醒”状态 < 3 个 LFCLK 周期，然后再次唤醒，则将检测不到下一个唤醒边沿。

情况 2：

STANDBY1 唤醒——如果使用 GPIO 边沿唤醒，且器件返回 STANDBY1 时 GPIO 脉冲仍处于活动状态，则器件将不会检测任何后续唤醒边沿。

权变措施

情况 1：

确保在器件处于激活模式时 GPIO 被置为无效
或

GPIO_ERR_01

(续)

GPIO 模块

确保 GPIO 唤醒脉冲超过 3 个 LFCLK 周期

情况 2 :

将 GPIO 唤醒沿设置为下降沿和上升沿

或

确保 GPIO 唤醒脉冲在进入 STANDBY1 之前未激活

GPIO_ERR_02**GPIO 模块**

类别

功能

功能

在 PA2 引脚上注入负电流后产生高电流消耗

说明

流入 PA2 引脚的负电流注入可能会导致器件中出现意外且持续的高电流消耗。

如果在 ROSC 模式下使用 PA2 (在 ROSC 和 VSS 之间安装了 100k Ω 电阻器) , 则此勘误表不适用。

权变措施

在 PA2 引脚和任何连接的电路之间放置一个典型值为 100 Ω 的串联电阻器, 并将该电阻器放置在 PA2 引脚附近。该电阻旨在限制 PA2 引脚上的快速瞬变, 并足以防止发生这种情况。

或

避免使用 PA2 引脚并改用备用引脚。

GPIO_ERR_03**GPIO 模块**

类别

功能

功能

调试器读取 GPIO EVENT0 IIDX 时, 中断被清除。

说明

在调试器读取 GPIO EVENT0 的 IIDX 时, 被视为 CPU 读取, 中断被清除。

权变措施

在调试期间, 可通过软件读取 RIS 来读取 event0 的 IIDX。

GPIO_ERR_04**GPIO 模块**

类别

功能

功能

配置全局快速唤醒可防止 GPIO 引脚将数据发送到 DIN 寄存器。

GPIO_ERR_04

(续)

GPIO 模块

说明

当配置 CTL 寄存器的仅快速唤醒位并在运行模式下强制将数据输入 GPIO 引脚时，器件将唤醒，但 GPIO 引脚上的数据不会出现在 DIN 寄存器中。这是因为 CTL 寄存器配置阻止任何数据从 GPIO 引脚流向 DIN 寄存器。

权变措施

当预期 GPIO 引脚上的数据进入 DIN 寄存器时，应避免使用 GPIO 仅快速唤醒功能。

GPIO_ERR_06

GPIO 模块

类别

功能

功能

当 DMA 传输正在进行时，对 GPIO DOUT、DOUTSET 和 DOUTCLR 寄存器的写入可能会丢失。

说明

DMA 无法访问 GPIO DOUT、DOUTSET 和 DOUTCLR 寄存器。由于实现中的错误，当并发 DMA 传输正在进行时，CPU 对 GPIO DOUT、DOUTSET 和 DOUTCLR 的访问也将被阻止。

权变措施

在应用程序代码中，软件不应写入 DOUT、DOUTSET 和 DOUTCLR 寄存器，而是应对 DOUTTGL 寄存器执行等效写入（有关 CPU 写入 DOUTTGL 寄存器的限制，请参阅权变措施 GPIO_ERR_05）。

在下面的伪代码中，“pins”表示待配置的 GPIO 模块中引脚的位向量。

```
DL_GPIO_setPins(GPIO_Regs* gpio, uint32_t pins)
{
    gpio->DOUTTGL31_0 = ~(gpio->DOUT31_0) & pins;
}
```

```
DL_GPIO_clearPins(GPIO_Regs* gpio, uint32_t pins)
{
    gpio->DOUTTGL31_0 = gpio->DOUT31_0 & pins;
}
```

```
DL_GPIO_writePins(GPIO_Regs* gpio, uint32_t pins)
{
    gpio->DOUTTGL31_0 = ~(gpio->DOUT31_0) & pins;
    gpio->DOUTTGL31_0 = gpio->DOUT31_0 & (~pins);
}
```

```
DL_GPIO_writePinsVal(GPIO_Regs* gpio, uint32_t pinsMask, uint32_t pinsVal)
{
    uint32_t doutVal = gpio->DOUT31_0;
    doutVal &= ~pinsMask;
    doutVal |= (pinsVal & pinsMask);
    gpio->DOUTTGL31_0 = ~(gpio->DOUT31_0) & doutVal;
}
```

GPIO_ERR_06

(续)

GPIO 模块

```
gpio->DOUTTGL31_0 = gpio->DOUT31_0 & (~doutVal);
}
```

I2C_ERR_01**I2C 模块**

类别

功能

功能

当发出 SMBUS 快速命令时，I2C 模块可能会在 SBMUS 模式下保持 SDA 线

说明

当 I2C 模块处于目标模式并配置为 SBMUS 时，如果总线控制器向器件发出 SMBUS 快速命令 (I2C 启动条件，然后是 7 位地址、1 位 R/W 信号、1 位 ACK 和 I2C STOP 条件)，且 R/W 位设置为读取，则 I2C 模块可能会在总线控制器尝试发出 I2C STOP 条件信号的同时，尝试将 SDA 线路拉低，从而阻止 STOP 条件成功完成。

权变措施

在完成地址 ACK 之前，将数据加载到 I2C 模块的发送 FIFO 中，并将 MSB 设置为 1，以防止 I2C 模块将 SDA 线路驱动为低电平。这将允许总线控制器成功发出 STOP 条件并完成 SMBUS 快速命令。

I2C_ERR_02**I2C 模块**

类别

功能

功能

I2C 快速命令读取模式仅在特定条件下有效

说明

I2C 快速命令读取模式仅在 TXFIFO 的数据 MSB 设为 1 且禁用时钟延展时有效。

权变措施

在 TXFIFO 中提供 MSB 设为 1 的虚拟数据，并禁用时钟延展 (CLKSTRETCH = 0)。

I2C_ERR_03**I2C 模块**

类别

功能

功能

I2C 外设模式在使用 MFCLK 源时无法唤醒器件

说明

如果 I2C 模块配置为外设模式
且 I2C 采用 MFCLK (中频时钟) 时钟
且器件置于 STOP2 或 STANDBY0/1 功耗模式，
则 I2C 无法在接收数据时唤醒器件。

I2C_ERR_03 (续) I2C 模块

权变措施

若需在 I2C 外设模式下通过接收数据实现低功耗唤醒，则需将 I2C 的时钟源设置为 BUSCLK，而非 MFCLK。

I2C_ERR_04 I2C 模块

类别

功能

功能

当 SCL 为低电平且 SDA 为高电平时，目标 i2c 无法释放延展。

说明

1：SCL 线路接地并释放，器件不限期将 SCL 拉至低电平。

2：时钟后延展、超时和释放；如果线路上存在另一个时钟低电平，器件会不限期地将 SCL 拉至低电平。

权变措施

如果 I2C 目标应用在低功耗模式下不需要使用异步快速时钟请求进行数据接收，则建议默认禁用 SWUEN，包括在复位或功率周期期间也是一样。在这种情况下，不会出现错误描述 1 和 2。

如果 I2C 目标应用需要使用异步快速时钟请求在低功耗模式下接收数据，请在进入低功耗模式之前启用 SWUEN，并在退出低功耗模式后清除 SWUEN。即使在这种情况下，当 I2C 目标处于低功耗模式时，也可能会出现错误描述 1 和 2，如果总线上的另一个器件引发连续时钟延展或超时，它也会不限期延长 SCL 线路。为了从这种情况中恢复，请在 I2C 目标器件上启用低电平超时中断，在低电平超时 ISR 内复位并重新初始化 I2C 模块。

I2C_ERR_05 I2C 模块

类别

功能

功能

如果我们在正在进行的事务期间切换 ACTIVE 位，I2C SDA 可能会卡在零电平的位置

说明

如果在正在进行的传输期间切换 ACTIVE 位，则其状态机将复位。但是，由 I2C 控制器驱动的 SDA 和 SCL 输出将不会复位。存在 SDA 为 0 且 I2C 控制器已进入 IDLE 状态的情况，在这种情况下，I2C 控制器将无法从 IDLE 状态向前移动或更新 SDA 值。设置 I2C 目标的 BUSBUSY (切换 ACTIVE 位会导致在线路上检测到启动)，并且 BUSBUSY 不会被清除，因为 I2C 控制器将无法驱动 STOP 来将其清除。

权变措施

在事务正在进行期间不要切换 ACTIVE 位。

I2C_ERR_06 I2C 模块

类别

功能

功能

当 I2C 时钟低于 24kHz 及更低时，SMBus 高电平超时功能会失败

I2C_ERR_06 (续) I2C 模块

说明

当 I2C 时钟速率低于 24kHz 及更低 (20kHz、10kHz) 时, SMBus 高电平超时功能会失败。根据 SMBUS 规格, 活动事务期间 SCL 高电平时间的上限为 50us。从将 START MMR 位写入 SCL 低电平所需的总时间为 60us, 即 >50us。它将触发超时事件, 并使 I2C 控制器进入 IDLE 状态, 而不会在传输本身开始时完成事务。以下是详细说明。对于 SCL 配置为 20kHz 的情况, SCL 低电平周期和高电平周期分别为 30us 和 20us。首先, 在高电平超时计数器开始递减的同时开始 MMR 位写入。然后, 从开始 MMR 位写入到 SDA 变为低电平 (启动条件) 需要一个 SCL 低电平 (30us)。接下来, 从 SDA 变为低电平 (启动条件) 到 SCL 变为低电平 (数据传输开始) 需要另一个 SCL 低电平周期 (30us), 此时应该停止高电平超时计数器。从计数器开始到结束总共需要 60us 的时间。但是, 由于高电平超时计数器的上限 (50us), 尽管 I2C 事务会正常工作, 而且不出现问题, 但仍将触发超时事件。

权变措施

当 I2C 时钟低于 24kHz 及更低时, 请勿使用 SMBus 高电平超时功能。

I2C_ERR_07 I2C 模块

类别

功能

功能

背对背控制器控制寄存器的写入可导致 I2C 无法启动。

说明

背对背 CTR 寄存器的写入可导致后续的 CTR.START 不会正确地引起启动条件。

权变措施

以单次写入的方式写入所有 CTR 位 (包括 CTR.START), 或者在 CTR 写入和 CTR.START 写入之间稍作等待。

I2C_ERR_08 I2C 模块

类别

功能

功能

RXDONE 中断后直接读取 FIFO 会导致读取到错误数据。

说明

当 RXDONE 中断发生时, FIFO 无法更新为最新数据。

权变措施

等待 2 个 I2C CLK 周期, 让 FIFO 确保获得最新数据。I2C CLK 基于 I2C 寄存器中的 CLKSEL 寄存器。

I2C_ERR_09 I2C 模块

类别

功能

功能

如果以低速运行 I2C, 则 ISR 读取时可能不会及时更新起始地址匹配状态。

I2C_ERR_09 (续) I2C 模块

说明	如果以低于 100kHz 的 I2C 速度运行，可能无法及时将 ADDRMATCH 位 (TSR 寄存器中的地址匹配) 置位来通过中断读取。
权变措施	如果在 I2C 上以低于 100kHz 的速度运行，请等待至少 1 个 I2C CLK 周期，然后再读取 ADDRMATCH 位。

I2C_ERR_10 I2C 模块

类别	功能
功能	启用 I2C 忙碌状态，防止进入低功耗模式。
说明	在 I2C 目标模式下，如果没有 STOP 位，I2C 忙碌状态会在事务后保持高电平。
权变措施	对 I2C 控制器进行编程，以发送 STOP 位并且不为最后一个字节发送 NACK。可以通过 STOP 条件终止任何 I2C 传输，以保持正确的忙碌状态和异步时钟请求行为 (用于重新进入低功耗模式)。

I2C_ERR_13 I2C 模块

类别	功能
功能	轮询 I2C BUSY 位可能无法保证控制器传输完成
说明	在设置 CCTR.BURSTRUN 位来启动 I2C 控制器传输后，大约需要 3 个 I2C 功能时钟周期才能将 BUSY 状态置为有效。如果在设置 CCTR.BURSTRUN 后立即轮询 BUSY 位以等待传输完成，可能会在 BUSY 状态尚未置位时就完成状态检查。在 CLKDIV 值较高 (导致 I2C 功能时钟较慢) 或编译器优化级别较高的情况下，更有可能发生该问题。
权变措施	在轮询 BUSY 状态之前添加软件延迟。软件延迟 = $3 \times \text{CPU CLK} / \text{I2C 功能时钟} = 3 \times \text{CPU CLK} / (\text{CLKSEL} / \text{CLKDIV})$ 例如，时钟分频器 (CLKDIV) 为 8、时钟源为 4MHz(MFCLK)，CPU CLK 为 32MHz 时：软件延迟 = $3 \times 32\text{MHz} / (4\text{MHz} / 8) = 192$ 个 CPU 周期

I2C_ERR_15 I2C 模块

类别	功能
功能	I2C SDA 上的干扰会导致 I2C 外设低功耗模式下处于活动状态
说明	I2C SDA 上 4us 内的干扰会导致 I2C 外设开关在低功耗模式 (STOP/STANDBY) 下进入活动状态，并且无法返回低功耗模式。

I2C_ERR_15 (续) I2C 模块

权变措施

1.增加 I2C 总线上的电容，但确保其总电容不超过 I2C 标准允许的值。2.增加 I2C 总线的电压。3.将 I2C 线移至远离噪声源。4.定期唤醒器件以检查 I2C 外设状态，如果其未处于 IDLE 状态，则复位 I2C 并将其重新初始化。

PMCU_ERR_01

GPIO 模块

类别

功能

功能

PA2/ROSC 引脚上的漏电流

说明

在不是 ROSC 的功能配置中使用引脚 PA2/ROSC 时，如果该引脚被驱动为高电平，则可能会观察到较大的漏电流。这是由于通过大约 17kΩ 的阻抗意外接地。

此勘误表不影响 ROSC 功能。

权变措施

如果 PA2/ROSC 用于 ROSC 的替代功能，则必须考虑当引脚从外部或内部被驱动为高电平时能量预算中的额外漏电流。

如果引脚通过电阻器从外部上拉，则由于此勘误表创建了分压器电路，外部电压将低于预期。

PMCU_ERR_02

GPIO 模块

类别

功能

功能

器件启动期间的瞬态电压输出

说明

在不是 ROSC 的功能配置中使用引脚 PA2/ROSC 时，在启动期间可以在 PA2/ROSC 引脚上观察到意外的瞬态电压脉冲。

此勘误表不影响 ROSC 功能。

权变措施

没有可用的解决方法。

PMCU_ERR_03

BOR 模块

类别

功能

功能

BOR1、BOR2 和 BOR3 在待机模式下不工作

PMCU_ERR_03

(续)

BOR 模块

说明

用户可选的备用 BOR 阈值 (BOR1、BOR2、BOR3) 在器件处于待机模式下运行时不起作用，因此器件在超过这些阈值时不会正确复位。

权变措施

在待机模式下运行时，请勿使用 BOR1、BOR2 或 BOR3。将器件配置为在进入待机模式之前使用默认的 BOR0 电平。退出待机模式后，可以重新启用备用 BOR 电平。

PMCU_ERR_10

PMCU 模块

类别

功能

功能

在某些工作条件下，VBOOST 可能会存在更高的延迟

说明

模拟多路复用的 VBOOST 在 $VDD < 1.8V$ 时具有较大的延迟，这会延迟 HFXT、COMP、SYSOSC (FCL 外部 R)、OPA 和 GPAMP 等其他模块的稳定时间。

权变措施

使用 GENCLKCFG[23:22]=0x2，保持 $VDD \geq 1.8V$ 并在 ONALWAYS 模式下使用 VBOOST。

PMCU_ERR_13

PMCU 模块

类别

功能

功能

MCU 可能会在从 STOP2 和 STANDBY0 唤醒时卡住

说明

如果在器件转换到 STOP2 或 STANDBY 模式时预取访问处于挂起状态，则当器件唤醒时，挂起的预取会阻止器件恢复正常执行。如果 WFI 指令未按字对齐，且闪存等待状态为 2，则会发生勘误表所述问题。在这种情况下，既不会处理 DMA 传输，也不会处理挂起的中断。

权变措施

用户应禁用预取并发出关断存储内存读取，从而防止发出新的预取并允许挂起的预取完成。

PWREN_ERR_01

GPIO 模块

类别

功能

功能

禁用 PWREN 寄存器后仍可访问外设寄存器

PWREN_ERR_01

(续)

GPIO 模块**说明**

将 PWREN 寄存器置 0 禁用外设电源时，读取外设寄存器可能会保留数据值。当 PWREN 寄存器为 0 时，读取或写入寄存器不会产生任何影响，因为外设不会产生任何影响。

以下外设受到影响：比较器 (COMP)、运算放大器 (OPA)、定时器 A、定时器 G、通用输入/输出 (GPIO) 和窗口看门狗定时器 (WWDG)。

权变措施

当外设的 PWREN 寄存器设置为 0 时，相关寄存器的值应忽略或视为无效。

RST_ERR_01**RST 模块****类别**

功能

功能

当 LFXT 或 LFCLK_IN 丢失后，NRST 释放沿与 LFOSCGOOD 上升沿同时出现时，器件仍处于复位状态

说明

在一种特定的边角情况下，如果 NRST 信号的释放沿（低电平到高电平的过渡）与 LFOSCGOOD 的上升沿同时出现，器件将无法检测到 NRST 的失效情况，并会无限期地保持在复位状态。可能触发该问题的条件：当 LFCLK 源从 LFCLK_IN 或 LFXT 切换至 LFOSC 时，LFOSC 将被重新启用，且在重新启用事件发生后的 250us（最小）至 1.5ms（最大）内，LFOSCGOOD 状态将转为高电平。如果施加的 NRST 低电平脉冲使其释放沿（低电平到高电平的过渡）与 LFOSCGOOD 的上升沿在 50ns 的时间窗口内重合，则无法检测到 NRST 信号的释放，器件将保持在复位状态。

权变措施

将 NRST 低电平脉冲的宽度保持在 2ms 以上，以避免此问题

SPI_ERR_01**SPI 模块****类别**

功能

功能

SPI 奇偶校验位不起作用

说明

SPI 硬件奇偶校验模式不起作用。

权变措施

请勿通过 SPI 模块 CTL1 寄存器中的 PTEN 或 PREN 位启用硬件奇偶校验。奇偶校验的计算与校验可由应用软件实现。

SPI_ERR_03**SPI 模块****类别**

功能

SPI_ERR_03 (续) SPI 模块

功能

当配置为外设时，启用 CSCLR 将导致接收到的数据在 SPH=0 模式下产生右移

说明

在外设模式下启用 CSCLR 时，如果当 CS 处于活动或无效时 SCK 线路上存在干扰，则接收到的数据将在下一个第一帧中具有 1 位右移。此问题在 SPH=0 时的 Motorola SPI 帧格式中出现，并将影响多外设模式，该模式在 CS 无效期间会切换 SCK。

权变措施

1. 设置 CSCLR=0h。
2. 在 SPH=0 模式下设置 CSCLR=1h 时，始终丢弃第一个帧。

SPI_ERR_04 SPI 模块

类别

功能

功能

当 SPI 外设处于仅接收模式时，在每个帧接收后进行 IDLE/BUSY 状态切换。

说明

如果 SPI 外设处于仅接收模式，则在 SPI 连续接收数据 (SPI_PHASE=1) 时，每个帧接收后都会切换 IDLE 中断和 BUSY 状态。此处没有数据加载到外设 TXFIFO，TXFIFO 为空。

权变措施

不要使用 SPI 外设的仅接收模式。将 SPI 外设设置为传输和接收模式。您无需在 TX FIFO 中为 SPI 设置任何数据。

SPI_ERR_05 SPI 模块

类别

功能

功能

无论 RXFIFO 数据如何，都将设置 SPI 外设接收超时中断

说明

当使用 SPI 超时中断时，即使在接收到最终 SPI CLK 后，RXTIMEOUT 也可以继续递减，这可能会导致错误的 RXTIMEOUT。

权变措施

接收到最后一个数据包后禁用 RXTIMEOUT (可以在 ISR 中完成) 并在 SPI 通信再次开始时重新启用。

SPI_ERR_06 SPI 模块

类别

功能

功能

调试暂停置为有效时，IDLE/BUSY 状态不反映 SPI IP 的正确状态

SPI_ERR_06 (续) SPI 模块

说明

IDLE/BUSY 与暂停无关，它仅控制 RXFIFO/TXFIFO 写入/读取选通。因此，如果控制器正在发送数据，尽管未在 FIFO 中为其设置门锁，但正在设置 BUSY。在暂停期间，POCI 线路会在线路上传输先前传输的数据

权变措施

当 SPI IP 暂停时，请勿使用 IDLE/BUSY 状态。

SPI_ERR_07 SPI 模块

类别

功能

功能

如果同时在 SPI 外设对 TXFIFO 进行读取/写入，则可能不会生成 SPI 下溢事件

说明

当 SPI.CTL0.SPH = 0 且器件配置为 SPI 外设时。

如果在从 SPI 控制器发出读取请求时对 TXFIFO 进行了写入，则由于同时发生读取/写入请求，可能不会生成下溢事件。

权变措施

当 SPI 控制器对器件寻址时，确保 TXFIFO 不为空，这可以通过预加载数据来实现，以避免对同一 TXFIFO 地址进行写入和读取。或者，可以使用数据检查策略（如 CRC）来验证数据包是否已正确发送，在 CRC 不匹配时，可以重新发送数据。

SPI_ERR_08 SPI 模块

类别

功能

功能

启用 DMA 时，SPI_DMA_DONE_RX 中断将影响 RX 中断

说明

如果 SPI_DMA_DONE_RX 中断标志置为有效，但中断未启用，则下次不会触发 RX 中断。

权变措施

启用 DMA_DONE_RX 中断。或者，手动清除 SPI_DMA_DONE_RX 中断标志。

SPI_ERR_09 SPI TX FIFO 中断可只在模块触发一次

类别

功能

功能

说明

SPI TX FIFO 触发器仅设置一次，如果 TX FIFO 未超过阈值并从中断服务例程返回，则即使 FIFO 低于阈值级别，RIS 也不会重新触发。

权变措施

请将“TX FIFO 阈值水平”配置为“TX FIFO 为空”。

SPI_ERR_10

SPI 模块

类别

功能

功能

DMA 未对最新 SPI 触发计数进行采样

说明

如果 SPI 配置为在发生接收超时 (RTOUT) 事件时触发 DMA 传输，且 DMA 通道在触发请求时正忙于为另一个传输提供服务，则在 DMA 应答请求之前 SPI 接收到的任何额外字节都不会包含在传输中。DMA 仅传输发出触发请求时接收到的字节数，使随后接收到的字节仍保留在 RX FIFO 中未被读取。

权变措施

将 DMA_TRIG_RX 配置为 RX 触发条件与 RTOUT 结合使用。当 RX FIFO 达到配置的阈值水平时，就会触发 RX 触发，从而确保在 DMA 服务开始之前，FIFO 中已存在全部预期字节数。这可保证无论 DMA 通道是否延迟应答请求，DMA 传输计数均准确无误。

SYSCTL_ERR_03

SYSCTL 模块

类别

功能

功能

在执行 SYSRESET 或对 SYSSTATUSCLR 进行写入后，DEDERRADDR 仍然存在

详细信息

在执行 SYSRESET 或对 SYSSTATUSCLR 进行写入后，DEDERRADDR 寄存器仍然存在。仅当发生新的 FLASHDED 错误时，才会覆盖其值。这种行为不符合技术参考手册 (TRM)，手册中规定其初始复位值为零。

权变措施

无权变措施

SYSCTL_ERR_05

LFCLK 模块

类别

功能

功能

LFCLK 在退出关断模式时无法正常工作

说明

如果 LFCLK_IN 引脚配置为具有上拉的通用输入 (或) LFCLK_IN 功能，则在该配置中，退出关断模式将导致 LFCLK 卡住。

权变措施

选择以下任一选项：1. 在该 LFCLK_IN I/O 上启用下拉而不是上拉 2. 避免将其配置为输入

SYSCTL_ERR_06

CLK_OUT 模块

类别

功能

SYSCTL_ERR_06

(续)

CLK_OUT 模块**功能**

当用户禁用 CLK_OUT 而异步时钟被选为 CLK_OUT 源时，外部时钟输出 (CLK_OUT) 上可能会出现干扰

说明

如果 CLK_OUT 的时钟源与当前总线时钟异步 (例如，总线时钟为 SYSOSC、而 CLK_OUT 的时钟源被选为 LFCLK)，则在这种情况下，当 CLK_OUT 引脚在启用一段时间后被禁用时，CLK_OUT 引脚上可能会出现干扰

权变措施

为避免外部引脚的干扰输出，请按照以下顺序操作：CLK_OUT 启用情况：IOMUX 启用配置应在 CLK_OUT 启用配置之后。CLK_OUT 禁用情况：IOMUX 禁用配置应在 CLK_OUT 禁用配置之前。

SYSCTL_ERR_11 SYSCTL 模块**类别**

功能

功能

当器件处于 RUN2/SLEEP2 模式且启用 FCL 时，会发生意外的 BOR 复位周期

说明

当 FCL 启用且器件在 RUN2/SLEEP2 模式下运行时，BOR1/2/3 在 NMI 之后会出现意外的 BOR

权变措施

使用 BOR1/2/3 监控器时，应避免启用 FCL 并避免使器件在 RUN2 或 SLEEP2 模式下运行。

SYSOSC_ERR_01 SYSOSC 模块**类别**

功能

功能

将 SYSOSC FCL 与 STOP1 模式一起使用时的 MFCLK 漂移

说明

如果启用了 MFCLK 并且 SYSOSC 使用频率改正环路 (FCL) 模式并使用 STOP1 低功耗工作模式，则当 SYSOSC 从 4MHz 转换回 32MHz 时 (在退出 STOP1 进入 RUN 模式时或发生强制 SYSOSC 进入 32MHz 的异步快速时钟请求时)，MFCLK 可能会漂移两个周期。

权变措施

使用 STOP0 模式而不是 STOP1 模式，使用 STOP0 模式时没有 MFCLK 漂移。

或

在使用 STOP1 时，在 FCL 模式下不使用 SYSOSC (使 FCL 保持禁用状态)。

SYSOSC_ERR_02 **SYSOSC 模块**

类别

功能

功能

在 LPM 下接收到异步时钟请求 (在 FCL 模式下禁用了 SYSOSC) 时, MFCLK 不工作

说明

在以下情况下, MFCLK 不会开始切换:

1. 启用 FCL 模式, 然后启用 MFCLK
 2. 进入禁用 SYSOSC 的低功耗模式(SLEEP2/STOP2/STANDBY0/STANDBY1)。
 3. 从一些使用 MFCLK 作为功能时钟的外设接收到异步请求。
- 接收到异步请求时, SYSOSC 将被启用, ulpclk 将变为 32MHz。但 MFCLK 会断开, 并且它根本不会切换, 因为器件的设置仍然为 LPM。

权变措施

如果 SYSOSC 正在使用 FCL 模式 - 当您进入 LPM 模式 (通常会关闭 SYSOSC) 时, 请勿启用外设的 MFCLK。

SYSOSC_ERR_04 **SYSOSC 模块**

类别

功能

功能

在 FCL ON 模式下, SYSOSC 精度降低

说明

当使用内部振荡器 (SYSOSC) 的 FCL ON 模式时, 精度可能会降低多达 +/-3%。精度下降是由 4MHz FCL 采样时钟与在 4MHz 谐波下运行的其他外设产生的系统噪声之间发生同步所致。

权变措施

目前没有解决方法可以完全消除系统中的噪声。但是, 如果使用 FCL ON 模式, 可通过让外设运行于非 4MHz 倍数频率, 将精度降低程度降至最低。

SYSOSC_ERR_05 **SYSOSC 模块**

类别

功能

功能

启用 FCL 时, SYSOSC 在从 LPM 进行异步快速唤醒期间的运行频率可能低于基频

说明

FCL=enabled 时, 在低功耗模式下且存在活动的异步快速时钟请求期间, SYSOSC 的运行频率可能比其基频最多低 10%。当器件处于低功耗模式时会发生这种情况, 原因是即使 FCL = enabled, 也始终会应用 FCL = Disabled 微调, 而不是使用 FCL= enabled 微调。器件唤醒、退出 LPM 并处理请求后, SYSOSC 将应用正确的微调, 并以其预期的目标频率恢复正常的 FCL = Enabled 运行。

大多数用例不会产生实质性影响, 用户仍可根据需要继续同时使用 FCL 和异步快速时钟请

SYSOSC_ERR_05

(续)

SYSOSC 模块

求。此勘误表会产生实质性影响的主要应用是 UART 为异步快速时钟请求源的应用，因为这种临时移位可能会影响有效波特率，直至器件被完全唤醒。

权变措施

1. 保持 FCL = disabled
2. 如果需要 FCL = Enabled，则禁用异步快速时钟请求。

SYSOSC_ERR_06**SYSOSC 模块****类别**

功能

功能

在 LPM 下接收到异步时钟请求（在 FCL 模式下禁用了 SYSOSC）时，MFCLK 会以 32MHz 而非 4MHz 工作

说明

在以下情况时，MFCLK 会错误地配置为 32MHz：

1. 启用 FCL 模式，然后启用 MFCLK
 2. 进入禁用 SYSOSC 的低功耗模式 (SLEEP2/STOP2/STANDBY0/STANDBY1)
 3. 从某些使用 MFCLK 作为功能时钟的外设接收到异步请求
- 收到异步请求后，将启用 SYSOSC，且 ULPClk 变为 32MHz。随后，MFCLK 错误地配置为 32MHz。

权变措施

如果 SYSOSC 正在使用 FCL 模式：当您进入 LPM 模式（通常会关闭 SYSOSC）时，请勿启用外设的 MFCLK。

TIMER_ERR_01**TIMx 模块****类别**

功能

功能

使用硬件事件启动计时器时，捕获模式捕获的值不正确

说明

在捕获模式下使用任何定时器实例时，使用零 (ZCOND) 或负载 (LCOND) 条件启动定时器会导致定时器捕获零值或负载值，而非捕获相应 TIMx.CC 寄存器中的值。这会影响周期和脉宽捕获等周期性用例。

权变措施

使用以下软件流程计算周期或脉冲宽度。有关权变措施示例，请参阅 MSPM0-SDK 中的 `timx_timer_mode_capture_duty_and_period`。

1. 通过设置为 0h 禁用 ZCOND 或 LCOND。
2. 当捕获发生时，捕获值将被正确捕获到 TIMx.CC 中
3. 将 TIMx.CTR 设置为重载值（负载或 0），以重新启动定时器。

TIMER_ERR_04 ***TIMER 模块***

类别

功能

功能

如果接近零事件，则可能会错过计时器的重新启用

说明

在单次模式下使用计时器时，如果接近零事件，则可能会错过计时器的重新启用。对计时器使能位进行硬件更新将需要一个功能时钟周期。例如，如果计时器的时钟源为 32.768kHz，时钟分频器为 3，则需要 ~100us 才能将使能位正确设置为 0。

权变措施

在重新启用计时器之前等待 1 个功能时钟周期，或者可以先禁用计时器，然后再重新启用。

通过 `CTRCTL.EN = 0` 禁用计数器，然后通过 `CTRCTL.EN = 1` 重新启用

TIMER_ERR_06 ***TIMER 模块***

类别

功能

功能

向 `CLKEN` 位写入 0 不会禁用计数器

说明

向计数器时钟控制寄存器 (`CCLKCTL`) 时钟使能位 (`CLKEN`) 写入 0 不会停止定时器。

权变措施

通过向计数器控制 (`CTRCTL`) 使能 (`EN`) 位写入 0 来停止定时器。

TIMER_ERR_07 ***TIMG 模块***

类别

功能

功能

初始重复计数器的周期比下一个重复少 1 个

说明

使用计时器重复计数器模式时，第一次重复的计数将比后续重复的计数少 1，因为以下重复计数器将包括 0 和加载值之间的转换。例如，如果 `TIMx.RCLD = 0x3`，则第一个重复计数器上将出现 3 个可观察到的零事件，并在以下重复计数器序列上显示 4 个可观察到的零事件。

权变措施

将初始 `RCLD` 值设置为比预期的 `RCLD` 大 1，然后在重复计数器归零事件 (`REPC`) 的 ISR 中将 `RCLD` 设置为预期的 `RCLD` 值。

例如，如果打算重复 4 次，请将初始 `RCLD` 值设置为 `RCLD = 0x5`，然后在 `REPC` 中断的计时器 ISR 中设置 `RCLD = 0x4`。现在，所有计时器重复将具有相同数量的零/加载事件。

UART_ERR_01	UART 模块
类别	功能
功能	在切换到 STANDBY1 模式时，未检测到 UART 启动条件
说明	器件处于 STANDBY1 模式时，由 UART 传输启动的异步快速时钟请求提供服务后，器件将返回 STANDBY1 模式。如果在转换回 STANDBY1 模式期间开始另一次 UART 传输，则器件无法正确检测到并接收数据。
权变措施	当预计存在重复的 UART 启动条件时，使用 STANDBY0 模式或更高的低功耗模式。
UART_ERR_02	UART 模块
类别	功能
功能	仅启用 TXE 时，不设置 UART 传输结束中断
说明	当器件设置为仅传输 (CTL0.TXE = 1, CTL0.RXE = 0) 时，不会触发 UART 传输结束 (EOT) 中断。当器件设置为传输和接收 (CTL0.TXE = 1, CTL0.RXE = 1) 时，EOT 会成功触发
权变措施	当使用 UART 传输结束中断时，设置 CTL0.TXE 和 CTL0.RXE 位。请注意，您不需要将引脚分配为 UART 接收。
UART_ERR_04	UART 模块
类别	功能
功能	当时钟从 SYSOSC 转换到 LFOSC 时，通过快速时钟请求接收到的错误 UART 数据会被禁用
说明	场景： 1.已选择 LFCLK 作为 UART 的功能时钟。 2.波特率为 9600，配置了 3 倍过采样 3.UART 快速时钟请求已被禁用 。如果在 UART RX 传输过程中 ULPCLK 从 SYSOSC 更改为 LFOSC，则会观察到一个位读取不正确
权变措施	在 LPM 模式下使用 UART 时启用 UART 快速时钟请求。

UART_ERR_05 *UART 模块*

类别

功能

功能

UART 模块中调试暂停功能的限制

说明

所有 Tx FIFO 元素都在通信进入暂停状态之前发出，预计完成现有帧并暂停。

权变措施

调试暂停置为有效后，请确保数据不会写入 TX FIFO。

UART_ERR_06 *UART 模块*

类别

功能

功能

UART 9 位模式下的 RTOUT/忙碌/异步异常行为

说明

在多节点场景中，UART 接收超时 (RTOUT) 无法正常工作，其中一个 UART 将用作控制器，其他 UART 节点作为外设，在 9 位 UART 模式下为每个外设配置不同的地址。第一个 UART 控制器与 UART 外设 1 通信，通过发送外设 1 的地址作为第一个字节，然后发送数据，外设 1 已看到地址匹配并接收到数据。控制器处理好外设 1 后，外设 1 在配置的超时期间后不设置 RTOUT、如果控制器立即开始与另一个 UART 外设 (外设 2) 的通信，该外设 1 在总线上配置了不同的地址。外设 1 RTOUT 计数器在与外设 2 和外设 1 通信过程中复位，仅在 UART 控制器完成与外设 2 的通信后才设置其 RTOUT。在 BUSY 和异步请求中观察到类似行为。即使与总线上的其他外设的通信时地址不匹配，控制器也正在设置忙碌和 Async 请求。

权变措施

请勿在单个控制器连接到多个外设的多节点 UART 通信中使用 RTOUT/ BUSY / 异步时钟请求行为。

UART_ERR_07 *UART 模块*

类别

功能

功能

在 IDLE LINE MODE 下，RTOUT 计数器不会按预期计数

说明

在 UART 中的 IDLE LINE MODE 下，RTOUT 计数器会卡住，即使线路处于 IDLE 状态且 FIFO 有一些元素也是如此。这意味着 RTOUT 中断在 IDLE LINE MODE 下将不起作用。如果地址不匹配，当在 Rx 线上看到切换时，将重新加载 RTOUT 计数器。在多响应器场景中，当命令发出位置和其他某个响应器之间正在通信时，这可能会导致产生 RTOUT 事件的无限延迟。

权变措施

在 IDLELINE 模式/多节点 UART 应用中使用 UART 模块时，请勿启用 RTOUT 功能。

UART_ERR_08	UART 模块
类别	功能
功能	STAT BUSY 并不代表 UART 模块的正确状态
说明	即使 UART 模块被禁用并且 TXFIFO 中有可用数据，STAT BUSY 也会保持高电平。
权变措施	轮询 TXFIFO 状态和 CTL0.ENABLE 寄存器位以识别 BUSY 状态。
UART_ERR_09	UART 模块
类别	功能
功能	当以较慢的 UART 速度运行时，UART ADDR_MATCH 可能无法在读取时及时设置。
说明	在地址匹配中断期间，当代码跳转到 ISR 并读取 FIFO 时。由于地址匹配中断在 STOP 位之前产生，UART 无法接收作为地址发送到 RX 线上的数据。
权变措施	在读取数据之前等待 1 个 UART CLK 周期，以便设置 ADDR_MATCH。
UART_ERR_10	UART 模块
类别	功能
功能	对于 UART IrDA 模式，BUSY 位设置会延迟
说明	在 IrDA 模式下，UART.STAT.BUSY 位会在 IrDA 启动脉冲的第二个边沿设置；这意味着整个位传输将在正确设置 BUSY 状态之前完成。在此期间，如果软件轮询 BUSY 位，即使 IrDA 启动脉冲正在进行，也会出现 UART 不繁忙的错误指示。BUSY 状态将受到 UART 波特率的影响，UART 传输速度越慢，正确设置 BUSY 之前的时间就越长。
权变措施	检查 BUSY 状态之前位传输长度的延迟。或者，依次检查 UART.STAT.BUSY == 0x0 和 UART.STAT.BUSY == 0x1 是另一种使动态延迟独立于波特率或其他 ISR 的权变措施。
UART_ERR_11	UART 模块
类别	功能
功能	在 STOP 位事务期间，UART 接收超时比预期更早开始计数

UART_ERR_11

(续)

UART 模块

说明

在 STOP 位事务期间，接收超时将在 STOP 位事务的中间开始计数，如果 RXTOSSEL 设置太小，这可能会导致意外的 RTOUT 中断。例如，如果波特率为 1Mbps 且 RXTOSSEL 设置为 1，则预期的 RTOUT 应该在 STOP 位事务后 1us 发生，而不是 RTOUT 中断设置为 0.5us。

权变措施

UART.IFLS.RXTOSSEL 寄存器选择在接收超时 (RTOUT) 中断触发之前的位时间。RXTOSSEL 值需要大于 1 才能防止提前中断。接收超时时间的计算公式如下：接收超时 = (RXTOSSEL - 0.5)/波特率

VREF_ERR_01

VREF 模块

类别

功能

功能

禁用 VREF 后，VREF READY 位不会被清除

说明

在 SYSRST 之后首次启用 VREF 模块时，VREF READY 位可用于其预期功能。如果在应用中禁用 VREF 模块，VREF READY 位不会被清除。由于此勘误表，VREF 模块的后续启用无法使用 VREF READY 位来指示 VREF 模块是否稳定。

权变措施

在应用中重新启用 VREF 模块时，在使用 VREF 模块之前，利用 TIMER 模块等待最大 VREF 启动时间。有关 VREF 启动时间，请参阅器件数据表。

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

Changes from AUGUST 30, 2025 to JULY 30, 2026 (from Revision D (August 2025) to Revision E (July 2026))

Page

• 已更新 ADC_ERR_02 说明.....	5
• 已更新 ADC_ERR_02 权变措施.....	5
• 已更新 ADC_ERR_05 权变措施.....	6
• 已更新 ADC_ERR_05 说明.....	6
• 已更新 ADC_ERR_06 权变措施.....	6
• 已更新 ADC_ERR_06 说明.....	6
• 已更新 ADC_ERR_11 模块.....	7
• 已更新 ADC_ERR_11 功能.....	7
• 已更新 ADC_ERR_11 说明.....	7
• 已更新 ADC_ERR_11 权变措施.....	7
• 已更新 CPU_ERR_01 功能.....	9
• 已更新 CPU_ERR_01 说明.....	9
• 已更新 CPU_ERR_01 权变措施.....	9
• 已更新 CPU_ERR_02 功能.....	9
• 已更新 CPU_ERR_02 说明.....	9
• 已更新 CPU_ERR_02 权变措施.....	9

• 已更新 CPU_ERR_03 功能.....	10
• 已更新 CPU_ERR_03 说明.....	10
• 已更新 CPU_ERR_03 权变措施.....	10
• 已更新 CPU_ERR_04 类别.....	10
• 已更新 CPU_ERR_04 模块.....	10
• 已更新 CPU_ERR_04 功能.....	10
• 已更新 CPU_ERR_04 说明.....	10
• 已更新 CPU_ERR_04 权变措施.....	10
• 已更新 CRC/CRCP_ERR_01 类别.....	11
• 已更新 CRC/CRCP_ERR_01 模块.....	11
• 已更新 CRC/CRCP_ERR_01 功能.....	11
• 已更新 CRC/CRCP_ERR_01 说明.....	11
• 已更新 CRC/CRCP_ERR_01 权变措施.....	11
• 已更新 FCC_ERR_01 类别.....	11
• 已更新 FCC_ERR_01 权变措施.....	11
• 已更新 FCC_ERR_01 功能.....	11
• 已更新 FCC_ERR_01 说明.....	11
• 已更新 FCC_ERR_01 模块.....	11
• 已更新 FLASH_ERR_02 功能.....	11
• 已更新 FLASH_ERR_02 说明.....	11
• 已更新 FLASH_ERR_02 权变措施.....	11
• 添加了 FLASH_ERR_03.....	12
• 已更新 FLASH_ERR_08 模块.....	13
• 已更新 FLASH_ERR_08 功能.....	13
• 已更新 FLASH_ERR_08 说明.....	13
• 已更新 FLASH_ERR_08 权变措施.....	13
• 已更新 GPIO_ERR_04 模块.....	14
• 已更新 GPIO_ERR_04 类别.....	14
• 已更新 GPIO_ERR_04 功能.....	14
• 已更新 GPIO_ERR_04 权变措施.....	14
• 已更新 GPIO_ERR_04 说明.....	14
• 已更新 GPIO_ERR_06 类别.....	15
• 已更新 GPIO_ERR_06 模块.....	15
• 已更新 GPIO_ERR_06 功能.....	15
• 已更新 GPIO_ERR_06 说明.....	15
• 已更新 GPIO_ERR_06 权变措施.....	15
• 已更新 I2C_ERR_09 说明.....	18
• 已更新 I2C_ERR_09 权变措施.....	18
• 已更新 I2C_ERR_13 模块.....	19
• 已更新 I2C_ERR_13 功能.....	19
• 已更新 I2C_ERR_13 权变措施.....	19
• 已更新 I2C_ERR_13 说明.....	19
• 已更新 I2C_ERR_15 类别.....	19
• 已更新 I2C_ERR_15 模块.....	19
• 已更新 I2C_ERR_15 功能.....	19
• 已更新 I2C_ERR_15 权变措施.....	19
• 已更新 I2C_ERR_15 说明.....	19
• 添加了 PMCU_ERR_10.....	21
• 已更新 PMCU_ERR_13 功能.....	21
• 已更新 PMCU_ERR_13 说明.....	21
• 已更新 PMCU_ERR_13 权变措施.....	21

• 已更新 RST_ERR_01 功能.....	22
• 已更新 RST_ERR_01 说明.....	22
• 已更新 RST_ERR_01 权变措施.....	22
• 已更新 SPI_ERR_03 权变措施.....	22
• 已更新 SPI_ERR_07 说明.....	24
• 已更新 SPI_ERR_07 权变措施.....	24
• 已更新 SPI_ERR_08 模块.....	24
• 已更新 SPI_ERR_08 功能.....	24
• 已更新 SPI_ERR_08 说明.....	24
• 已更新 SPI_ERR_08 权变措施.....	24
• 已更新 SPI_ERR_09 模块.....	24
• 已更新 SPI_ERR_09 说明.....	24
• 已更新 SPI_ERR_09 权变措施.....	24
• 已更新 SPI_ERR_10 类别.....	25
• 已更新 SPI_ERR_10 模块.....	25
• 已更新 SPI_ERR_10 功能.....	25
• 已更新 SPI_ERR_10 说明.....	25
• 已更新 SPI_ERR_10 权变措施.....	25
• 添加了 SYSCTL_ERR_03.....	25
• 已更新 SYSCTL_ERR_05 类别.....	25
• 已更新 SYSCTL_ERR_05 模块.....	25
• 已更新 SYSCTL_ERR_05 功能.....	25
• 已更新 SYSCTL_ERR_05 说明.....	25
• 已更新 SYSCTL_ERR_05 权变措施.....	25
• 已更新 SYSCTL_ERR_06 类别.....	25
• 已更新 SYSCTL_ERR_06 模块.....	25
• 已更新 SYSCTL_ERR_06 功能.....	25
• 已更新 SYSCTL_ERR_06 说明.....	25
• 已更新 SYSCTL_ERR_06 权变措施.....	25
• 已更新 SYSCTL_ERR_11 类别.....	26
• 已更新 SYSCTL_ERR_11 模块.....	26
• 已更新 SYSCTL_ERR_11 功能.....	26
• 已更新 SYSCTL_ERR_11 权变措施.....	26
• 已更新 SYSCTL_ERR_11 说明.....	26
• 已更新 SYSOSC_ERR_04 类别.....	27
• 已更新 SYSOSC_ERR_04 模块.....	27
• 已更新 SYSOSC_ERR_04 权变措施.....	27
• 已更新 SYSOSC_ERR_04 功能.....	27
• 已更新 SYSOSC_ERR_04 说明.....	27
• 已更新 SYSOSC_ERR_05 类别.....	27
• 已更新 SYSOSC_ERR_05 模块.....	27
• 已更新 SYSOSC_ERR_05 功能.....	27
• 已更新 SYSOSC_ERR_05 权变措施.....	27
• 已更新 SYSOSC_ERR_05 说明.....	27
• 已更新 SYSOSC_ERR_06 类别.....	28
• 已更新 SYSOSC_ERR_06 模块.....	28
• 已更新 SYSOSC_ERR_06 功能.....	28
• 已更新 SYSOSC_ERR_06 说明.....	28
• 已更新 SYSOSC_ERR_06 权变措施.....	28
• 已更新 TIMER_ERR_04 说明.....	29
• 已更新 TIMER_ERR_04 权变措施.....	29

• 已更新 TIMER_ERR_06 模块.....	29
• 添加了 TIMER_ERR_07.....	29
• 已更新 UART_ERR_10 模块.....	32
• 已更新 UART_ERR_10 功能.....	32
• 已更新 UART_ERR_10 说明.....	32
• 已更新 UART_ERR_10 权变措施.....	32
• 已更新 UART_ERR_11 模块.....	32
• 已更新 UART_ERR_11 功能.....	32
• 已更新 UART_ERR_11 说明.....	32
• 已更新 UART_ERR_11 权变措施.....	32

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