

C2000 F28004x 系列 MCU PLL 锁相 失败问题解决方案

JOHNSON CHEN/ EP FAE

摘要

C2000 第三代产品 F28004x 系列 MCU 由于其出色的性价比，因此被广泛的应用于汽车、工业、家电等市场。

C2000 F28004x 系列 MCU 有可能出现非常小概率且随机的 PLL 锁相失败问题，本文档将详细介绍 F28004x 系列 MCU PLL 锁相失败问题的解决方案。

本解决方案适用于整个 F28004x 系列产品，包括 F280049、F280048、F280045、F280041 和 F280040。

Contents

1	介绍.....	4
2	解决方案.....	5
3	解决方案使用方法.....	8
4	注意事项.....	8
5	参考文献.....	8

Figures

图 1.	PLL 勘误表问题	4
图 2.	DCC 勘误表问题	6

1 介绍

如图一所示，F28004x 系列 MCU PLL 存在不能锁相的可能性。PLL 不能锁相问题只有在 Power On Reset 时可能出现，而且出现的概率非常低为 PPM 级别，只有极少数的芯片可能会出现，这部分芯片出现 PLL 锁相失败问题也是随机性的。问题出现后，通常芯片重新上电故障会消失，因此问题较难分析和复现。

PLL 上电复位后，一旦锁相成功，在芯片掉电之前，PLL 不会再出现锁相失败问题。

Advisory	<i>PLL: PLL May Not Lock on the First Lock Attempt</i>
Revision(s) Affected	0, A, B
Details	<p>The PLL may not start properly at device power up. The PLLSTS[LOCKS] bit is set, but the PLL does not produce a clock.</p> <p>Once the PLL has started properly, the PLL can be disabled and reenabled with no issues and will stay locked. However, the PLL lock problem could reoccur on a subsequent power-up cycle.</p> <p>If the SYSPLL has not started properly and is selected as the CPU clock source, the CPU will stop executing instructions.</p> <p>The occurrence rate of this transient issue is low. After an initial occurrence, this issue may not be subsequently observed in the system again. Implementation of the workaround reduces the rate of occurrence.</p>

图 1. PLL 勘误表问题

这个问题出现后，CPU 将运行在外部(XTAL)时钟或者内部时钟(INTOSC1 或者 INTOSC2)的频率上。例如客户目标是 CPU 工作在 100MHz，软件配置选择外部无源时钟 20MHz XTAL 作为时钟源，如果锁相失败，CPU 将工作在 20MHz。因此当 PLL 锁相失败问题出现后，客户系统将无法正常工作。如果 F28004x 是用于环路控制，由于实际的 PWM 周期是期望的 PWM 周期的 5 倍，还有可能出现因为功率器件导通时间过长，从而损坏功能器件的可能性。由此可见如果出现 PLL 锁相失败问题，将会带来非常严重的系统性失效。

本文将详细介绍 F28004x 系列 MCU PLL 锁相失败问题的解决方案。

2 解决方案

本方案会通过多次锁相的方法并配合片上的看门狗 (Watchdog Timer) 和双时钟比较 (DCC) 模块来解决 PLL 锁相失败的问题。

由于过多的锁相尝试次数会带来较长的系统启动时间，因此对于启动时间较关键的应用会带来挑战。因为 PLL 本身是 PPM 的小概率事件，本方案会在尽量减少启动时间的前提下，来解决这个问题。

下面为具体实现方法：

步骤一：

在不同版本的 C2000Ware 里面的 f28004x_sysctrl.c 中 InitSysPll (Uint16 clock_source, Uint16 imult, Uint16 fmult, Uint16 divsel) 函数里面，尝试多次锁相函数有如下一些问题：

1. 有的版本不包含 EALLOW, 虽然尝试多次锁相，但实际效果相当于只锁了一次。
2. 有的版本使用的是 ClkCfgRegs.SYSPLLCTL1.bit.PLLCLKEN = 0 而不是 ClkCfgRegs.SYSPLLCTL1.bit.PLLEN = 0。
3. 有的版本等待时间是 33 个 CPU CLK (asm(" RPT #33 || NOP");)，而芯片手册要求最少 60 个 CPU CLK。

以上这些问题都会导致多次锁相失败，从 C2000Ware_3_04_00_00 开始已经使用下面红色部分代码，如果使用的 C2000Ware 版本早于 3_04_00_00 会有以上这些问题，因此需要确保使用红色部分代码。

```

for(retries = 0; (retries < PLL_RETRIES); retries++)
{
    EALLOW;
    ClkCfgRegs.SYSPLLCTL1.bit.PLLEN = 0;
    asm(" RPT #60 || NOP");
    //
    // Program PLL multipliers
    //
    ClkCfgRegs.SYSPLLMULT.all = temp_syspllmult;

```

//

步骤二:

使用 DCC 来检测 PLL 输出时钟是否正常。如图二所示，由于 DCC 本身有勘误表描述的问题，因此需要加入软件代码来解决 DCC 勘误表的问题，附件代码中 IsPLLValid(Uint16 oscSource, Uint16 imult, Uint16 fmult) 函数为具体的解决 DCC 勘误表问题的代码。

Advisory	<i>DCC: Single-Shot-Mode Operation May End Prematurely</i>
Revision(s) Affected	0, A, B
Details	In single-shot mode, DCCSTATUS[DONE] or DCCSTATUS[ERROR] may be prematurely set. When this occurs, DCC results are invalid.
Workaround(s)	Any of the following conditions ends DCC operation prematurely. TI recommends rerunning DCC if <i>any</i> of the below conditions are met. <ul style="list-style-type: none"> • DCCSTATUS[DONE] = 1 and (DCCCNT1 > 0 or DCCCNT0 > 0 or DCCVALID0 > 0) • DCCSTATUS[ERROR] = 1 and DCCCNT1 > 0 and DCCVALID0 > 0

图 2. DCC 勘误表问题

步骤三:

为防止出现尝试多次锁相后，依然无法成功锁相，从而导致 CPU 频率变成时钟源频率的问题。

在锁相不成功时，将使能看门狗复位，从而使芯片重新复位工作。

因此在 f28004x_sysctrl.c 中 InitSysPll(Uint16 clock_source, Uint16 imult, Uint16 fmult, Uint16 divsel) 函数中，加入下面红色部分代码。

```

if(status)
{
    EALLOW;
    //
    // Set divider to produce slower output frequency to limit current increase
    //
    if(divsel != PLLCLK_BY_126)

```

```

{
    ClkCfgRegs.SYSCLKDIVSEL.bit.PLLSYSCLKDIV = divsel + 1;
}
else
{
    ClkCfgRegs.SYSCLKDIVSEL.bit.PLLSYSCLKDIV = divsel;
}
//
// Enable PLLSYSCLK is fed from system PLL clock
//
ClkCfgRegs.SYSPLLCTL1.bit.PLLCLKEN = 1;
//
// Small 100 cycle delay
//
asm(" RPT #100 || NOP");
//
// Set the divider to user value
//
ClkCfgRegs.SYSCLKDIVSEL.bit.PLLSYSCLKDIV = divsel;
EDIS;
}
else
{
    EALLOW;
    WdRegs.SCSR.all = 0x0;           //Configure for watchdog reset
    WdRegs.WDCR.all = 0x28;        //Run at max frequency
    WdRegs.WDCR.bit.WDCHK = 0;    //Immediate reset CPU
    EDIS;
}

```

3 解决方案使用方法

整个完整代码实现可以下载 PLL_Solution_F28004x.c，解决方案使用方法如下：

步骤一：用 PLL_Solution_F28004x.c 里面的 InitSysPll(Uint16 clock_source, Uint16 imult, Uint16 fmult, Uint16 divsel) 函数替换 f28004x_sysctrl.c 里面的同一个函数。

步骤二：用 PLL_Solution_F28004x.c 里面的 IsPLLValid(Uint16 oscSource, Uint16 imult, Uint16 fmult) 函数替换 f28004x_sysctrl.c 里面的同一个函数。

4 注意事项

本文以 Bit Field 软件结构的写法为例，介绍了 PLL 锁相失败问题的具体解决方案。此解决方案原理同样适用于 Driverlib 软件结构。

5 参考文献

1. TMS320F28004x MCUs Silicon Errata (SPRZ439)
2. TMS320F28004x Real-Time Microcontrollers Technical Reference Manual (SPRUI33)

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 © 2022，德州仪器 (TI) 公司