Application Note

避免 UCD3138 器件锁定

Xuemei Lu

摘要

当器件无法再重新编程时就会发生锁定。锁定可能是由不正确的下载流程或存在故障的固件引起的。本应用手册介绍了可能导致 UCD3138 器件发生锁定的原因以及避免锁定的方法。

内容

插图清单

商标

所有商标均为其各自所有者的财产。

English Document: SLUAAP5

Copyright © 2023 Texas Instruments Incorporated

1 为何会发生锁定

当存在无法清除的有效校验和时，可能会发生锁定。不同的 UCD3138 器件具有不同数量的校验和。

UCD3138(A) 器件具有单个块。校验和有两个位置，每个校验和为 4 个字节。

- 0x7fc - 引导校验和，即从地址 0 到 0x7fB 的 pflash 字节之和
- 0x7ffc - pflash 总体校验和，即从地址 0 到 0x7ffB 的 pflash 字节之和

图 1-1 是显示 UCD3138(A) ROM 如何处理校验和验证的流程图。只要两个校验和中有一个有效，就会跳转到地址 0 来执行 pflash 中的代码。

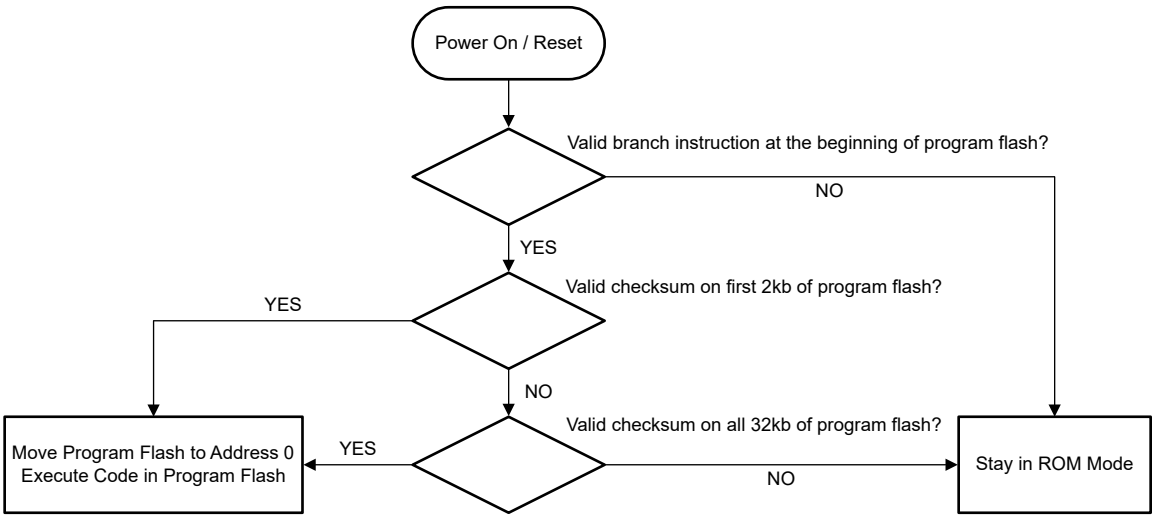

图 1-1. UCD3138(A) ROM 中的校验和验证流程图

UCD3138064(A) 器件具有块 1 和块 2。校验和有四个位置，每个校验和为 4 个字节。

- 0x7fc - 块 1 引导校验和，即从地址 0 到 0x7fB 的 pflash 字节之和
- 0x7ffc - 块 1 总体校验和，即从地址 0 到 0x7ffB 的 pflash 字节之和
- 0x87fc - 块 2 引导校验和，即从地址 0x8000 到 0x87fB 的 pflash 字节之和
- 0xfffc - 组合块 1 和块 2 的 64K 程序总体校验和，或仅块 2 的总体校验和

图 1-2 是显示 UCD3138064(A) ROM 如何处理校验和验证的流程图。

English Document: SLUAAP5
Copyright © 2023 Texas Instruments Incorporated

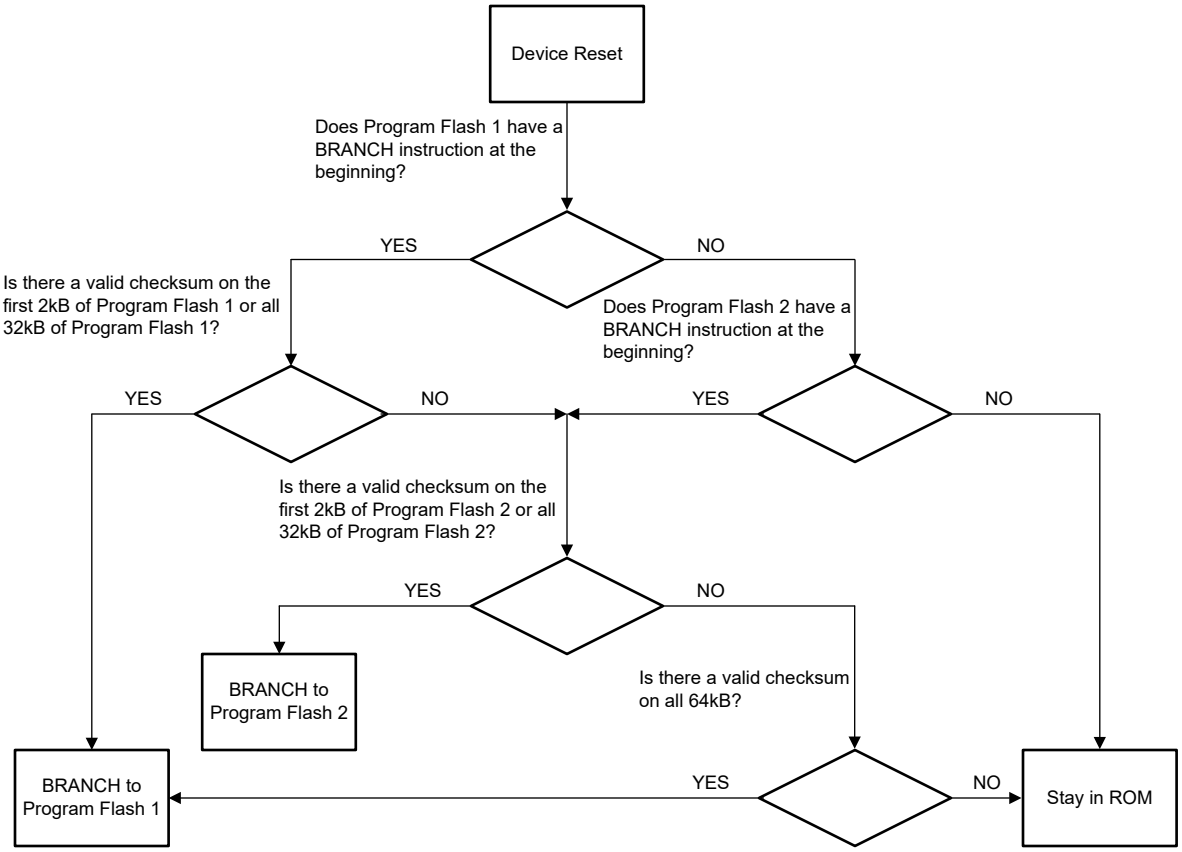

图 1-2. UCD3138064(A) ROM 中的校验和验证流程图

UCD3138128(A) 具有块 0、块 1、块 2 和块 3。校验和有四个位置，每个校验和为 8 个字节。

- 0x7f8 - 块 0 引导校验和，即从地址 0 到 0x7f7 的 pflash 字之和
- 0x7ff8 - 块 0 总体校验和，即从地址 0 到 0x7ff7 的 pflash 字之和
- 0xfff8 - 组合块 0 和块 1 的 64K 程序总体校验和，即从地址 0 到 0xfff7 的 pflash 字之和
- 0x1fff8 - 128K 程序总体校验和，或组合块 2 和块 3 的 64K 程序总体校验和

图 1-3 是显示 UCD3138128(A) ROM 如何处理校验和验证的流程图。

Copyright © 2023 Texas Instruments Incorporated

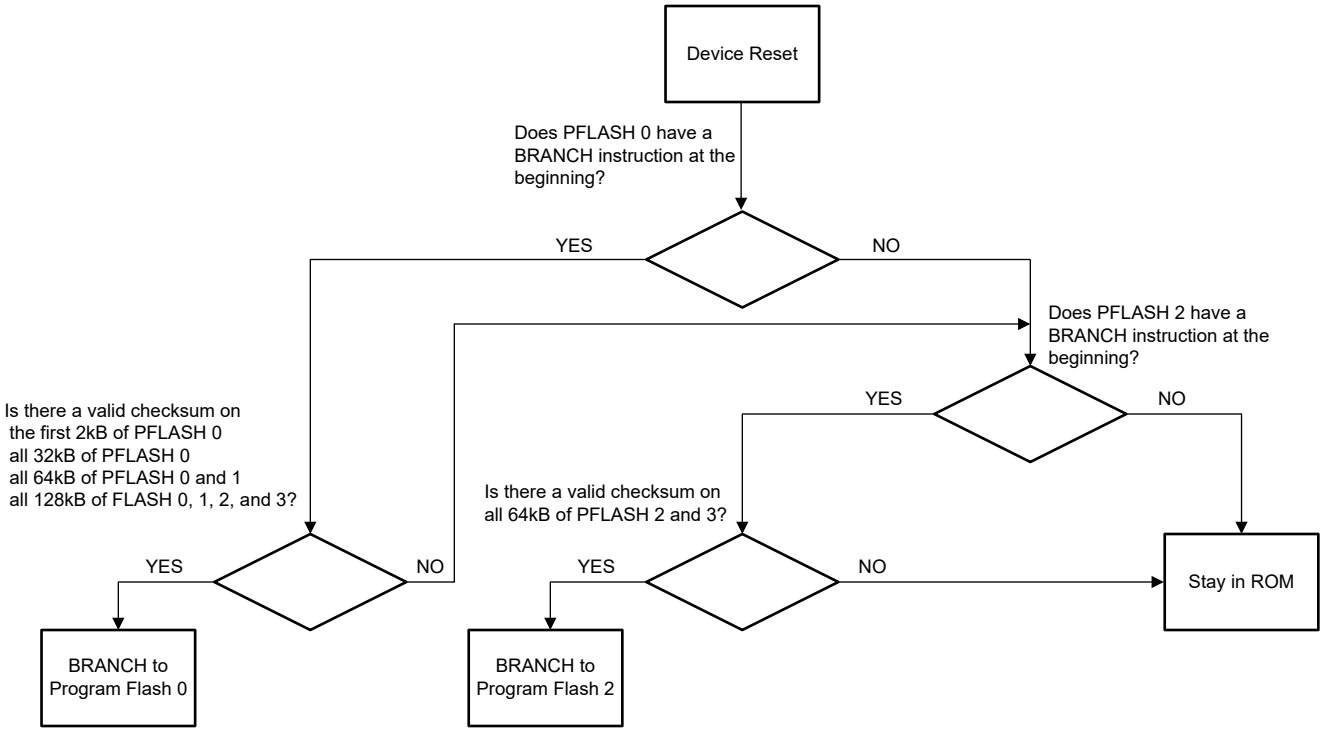

图 1-3. UCD3138128(A) ROM 中的校验和验证流程图

2 锁定的原因

2.1 load.asm 中的代码错误

在 load.asm 中，如果删除了以下红线，但保留了两条蓝线，则会导致器件持续复位，因为 r4 没有有效值。在这种情况下，器件会永久锁定，并且这 5 条线将完全删除或保留。

```
B_fast_interrupt
         .align 4
         .sect ".text"
             .state32
c_int00
;            B       c_int00
             LDR           r13, c_sup_stack_top ; initialize supervisor stack pointer
;            LDR           r4,c_mfbalr1_half0 ;point r4 at program flash base address register
;            MOV           r0,#0x62 ;make block size 32K, address 0, read only
;            STRH    r0,[r4]; store it there
             LDR           r0,c_mfbalr2_half0_load ;set up data flash for write only
             STRH    r0,[r4,#8] ;put it into mfbalr2
```

2.2 TI GUI 误操作

如果应用中没有引导加载程序，但选择了 *Boot support*，则地址 0x7f4（或 0x7f8）处可以存在用于引导的有效校验和。由于未使用引导加载程序，因此 0x7f4（或 0x7f8）处的校验和无法清除，并可能导致锁定。

English Document: SLUAAP5
Copyright © 2023 Texas Instruments Incorporated

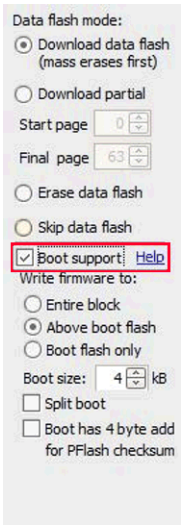

图 2-1. 引导支持选择

2.3 zero_out_integrity_word 函数失败

zero_out_integrity_word 函数用于清除校验和的指定位置。当 zero_out_integrity_word 函数未在 32 位模式下进行编译时，可能会发生锁定。此函数通常位于名为 zero_out_integrity_word.c 的独立文件中不同工程的函数名或文件名可能略有不同，但通常如下所示。

```
#define program_flash_integrity_word (*((volatile unsigned long *) 0x7ffc))
//last word in flash, when executing from Flash.  used to store integrity code

void zero_out_integrity_word(void)
{
    DecRegs.FLASHILOCK.all = 0x42DC157E;// Write key to Program Flash Interlock Register
    DecRegs.MFBALR1.all = MFBALRX_BYTE0_BLOCK_SIZE_32K; //enable program flash write
program_flash_integrity_word = 0;
   DecRegs.MFBALR1.all = MFBALRX_BYTE0_BLOCK_SIZE_32K + MFBALRX_BYTE0_RONLY;

    while(DecRegs.PFLASHCTRL.bit.BUSY != 0)
    {
        ; //do nothing while it programs
}

    return;
}
```

要检查此函数/文件是否在 32 位模式下编译：

- 右键点击 zero_out_integrity_word.c > *Properties* > *CCS Build* > *ARM Compiler* > *Processor Option* > *Designate code state*。
- 如果当前为 16，则更改为 32（如图 2-2 所示。）
- 重建工程。

请注意，此选项通过 CCS 中的配置完成，所以当变更 CCS 版本或编译器版本时，zero_out_integrity_word 函数可能会失败。

English Document: SLUAAP5
Copyright © 2023 Texas Instruments Incorporated

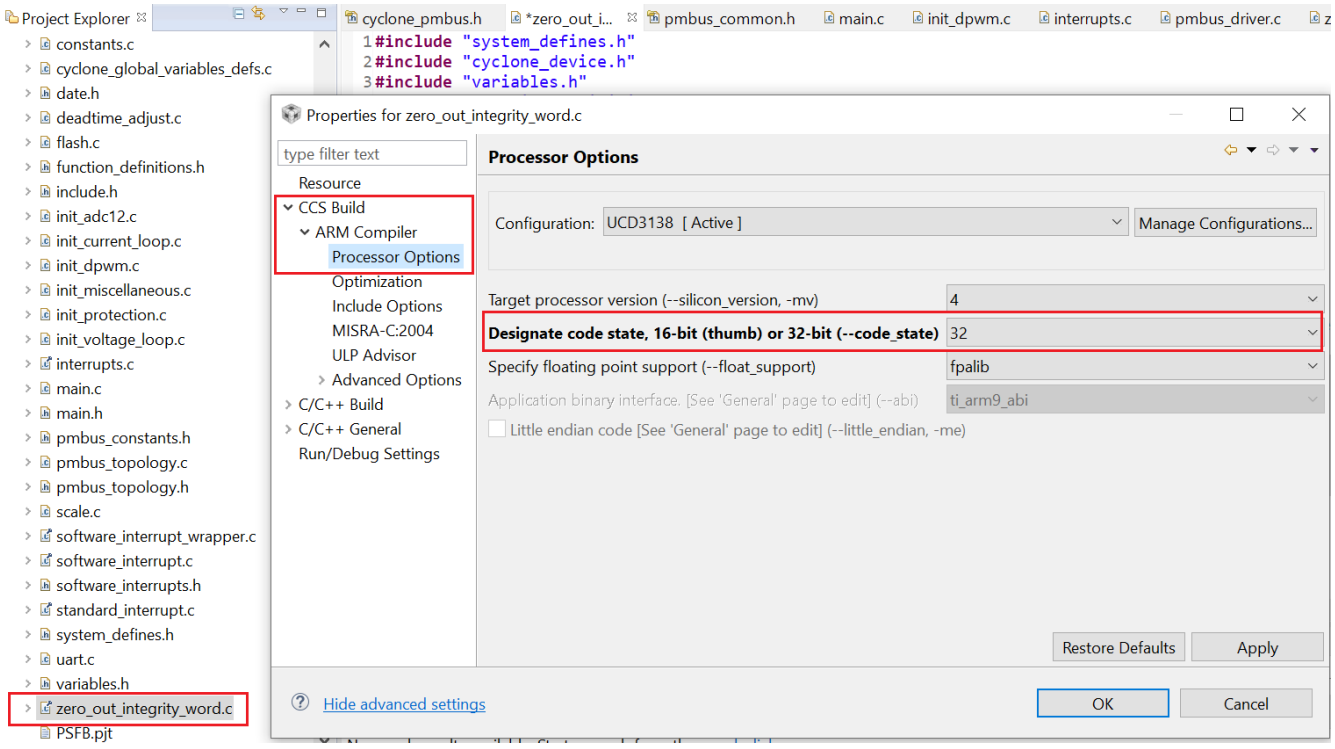

图 2-2. 指定代码状态

另一种选择是使用预处理器指令 #pragma 告知编译器在 32 位模式下编译指定函数。下面是一个示例：

```
#define program_flash_integrity_word (*((volatile unsigned long *) 0x7ffc))
//last word in flash, when executing from Flash.  used to store integrity code

#pragma CODE_STATE(zero_out_integrity_word, 32) // 16 = thumb mode, 32 = ARM mode
void zero_out_integrity_word(void)
{
    DecRegs.FLASHILOCK.all = 0x42DC157E;// Write key to Program Flash Interlock Register
    DecRegs.MFBALR1.all = MFBALRX_BYTE0_BLOCK_SIZE_32K; //enable program flash write
    program_flash_integrity_word = 0;
   DecRegs.MFBALR1.all = MFBALRX_BYTE0_BLOCK_SIZE_32K + MFBALRX_BYTE0_RONLY;
    while(DecRegs.PFLASHCTRL.bit.BUSY != 0)
    {
        ; //do nothing while it programs
}

    return;
}
```

2.4 PMBus 通信失败

当 zero_out_integrity_word 函数正常运行时可能会发生锁定，而 PMBus 通信失败的原因在于 zero_out_integrity_word 通常由 PMBus 命令触发。

要进行调试，请在应用代码的开头放置后门，如下所示。这是为了确保器件能够在满足后门条件时恢复。

```
void main()
{
    volatile unsigned int dummy;
    if(GioRegs.FAULTIN.bit.FLT3_IN == 0)// Re-Check pin assignment
    {
        clear_integrity_word();
}
}
```


English Document: SLUAAP5
Copyright © 2023 Texas Instruments Incorporated

2.5 意外情况

如果在使用旧 GUI 编程期间出现意外情况，则可能会发生锁定（例如，电源关闭或电线松动）。在使用引导校验和的情况下尤其如此。这是因为旧 GUI 程序按顺序从低地址转为高地址。如果发生这种情况，请使用最新的 GUI 工具 Fusion Digital Power Studio。该工具最后对校验和进行编程，从而使器件有机会在复位时恢复。

3 如何避免锁定

要避免锁定，请执行以下三个步骤：

步骤 1：如果固件可能无法正常运行，请勿在对 UCD 器件进行编程时写入校验和。虽然编程后校验和仍可跳转到 pflash 执行，但 UCD 器件始终可以在复位时返回到 ROM 模式。请遵循图 3-1 所示配置。

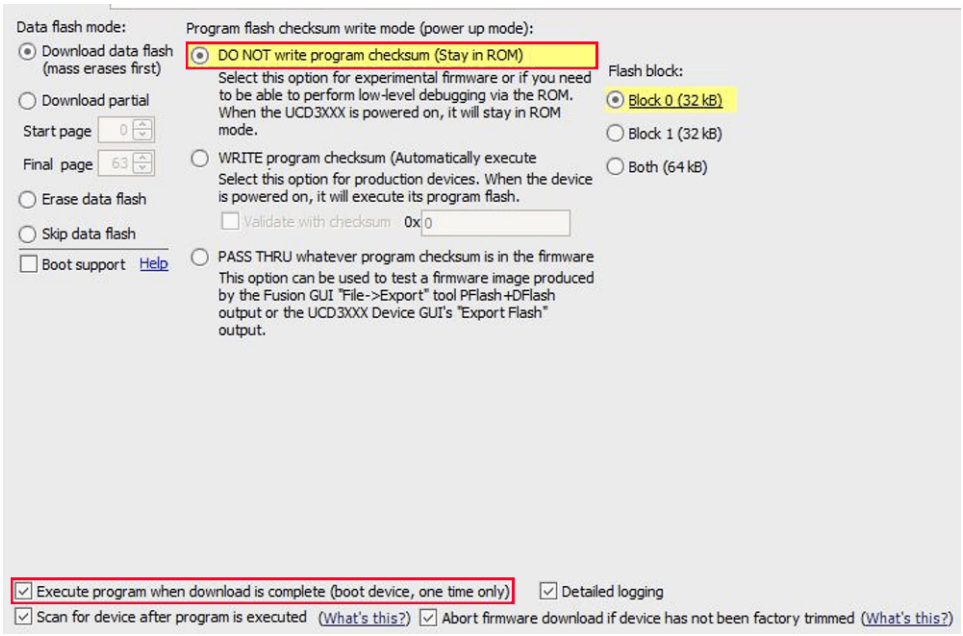

图 3-1. 无校验和编程

步骤 2：确认已正确清除校验和的预期位置。点击命令 *Command Program to jump to ROM (Send Byte 0xD9 to address xx)*，将 UCD 发送回 ROM 模式。

图 3-2. 跳转至 ROM 模式

步骤 3：转至 *Checksums* 标签。选择 *Dump* 按钮来通过配置 *Block Configuration* 读取每个校验和。检查是否清除了预期的校验和位置。如果已正确清除，该字段将显示为零。

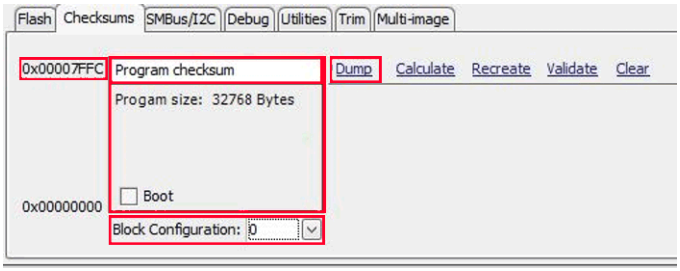

图 3-3. 校验和值验证

Copyright © 2023 Texas Instruments Incorporated

检查代码，确认是否已清除正确的校验和。按预期清除后，使用校验和对 UCD 器件进行编程。

4 使用 JTAG 解锁

如果器件无法重新编程，但固件和存储器调试器的读写功能正常运行，则可以通过 JTAG 解锁。

4.1 启用 JTAG 功能

JTAG 端口主要包括 4 个引脚：TCK/TDI/TDO/TMS。如果禁用 JTAG 功能，这 4 个引脚可在 GPIO 模式下正常运行，这在应用中是正常情况。要启用 JTAG 功能，请检查以下寄存器的配置，并在必要时进行重新配置。将 IOMUX 寄存器设置为 0 以便启用 JTAG，并确保 TCK/TDI/TDO/TMS 引脚均不在 GPIO 模式下运行。这可以通过 UCD3xxx 器件 GUI 中的存储器调试器来完成。

```
MiscAnalogRegs.IOMUX.all = 0; //enable JTAG

        MiscAnalogRegs.GLBIOEN.bit.TCK_IO_EN = 0;
        MiscAnalogRegs.GLBIOEN.bit.TDI_IO_EN = 0;
        MiscAnalogRegs.GLBIOEN.bit.TDO_IO_EN = 0;
        MiscAnalogRegs.GLBIOEN.bit.TMS_IO_EN = 0;
```

4.2 CCS 中的新目标配置

在 CCS 顶部的菜单中，点击 *View* > *Target Configurations*。在目标配置窗格中，右键点击“User Defined”，选择 *New target configuration*。在文件名中填写有意义的配置名称（例如，仿真器类型）和 UCD 型号，如图 4-1 所示。点击 *Finish*。

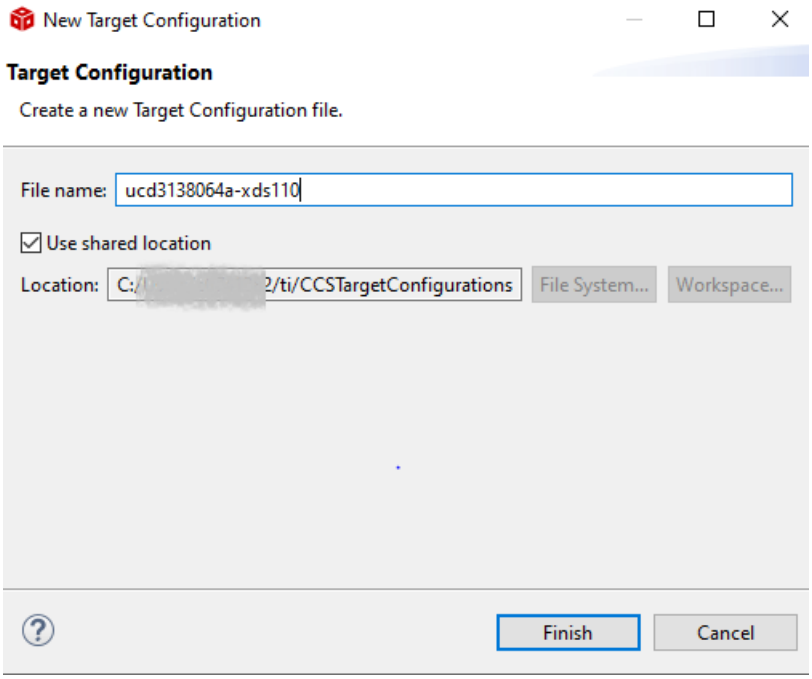

图 4-1. 新目标配置

在 *Connection* 和 *Board or Device* 项中，分别选择正确的仿真器类型和 UCD 型号。下面以 JTAG XDS110 和 UCD3138064A 为例展示。

English Document: SLUAAP5
Copyright © 2023 Texas Instruments Incorporated

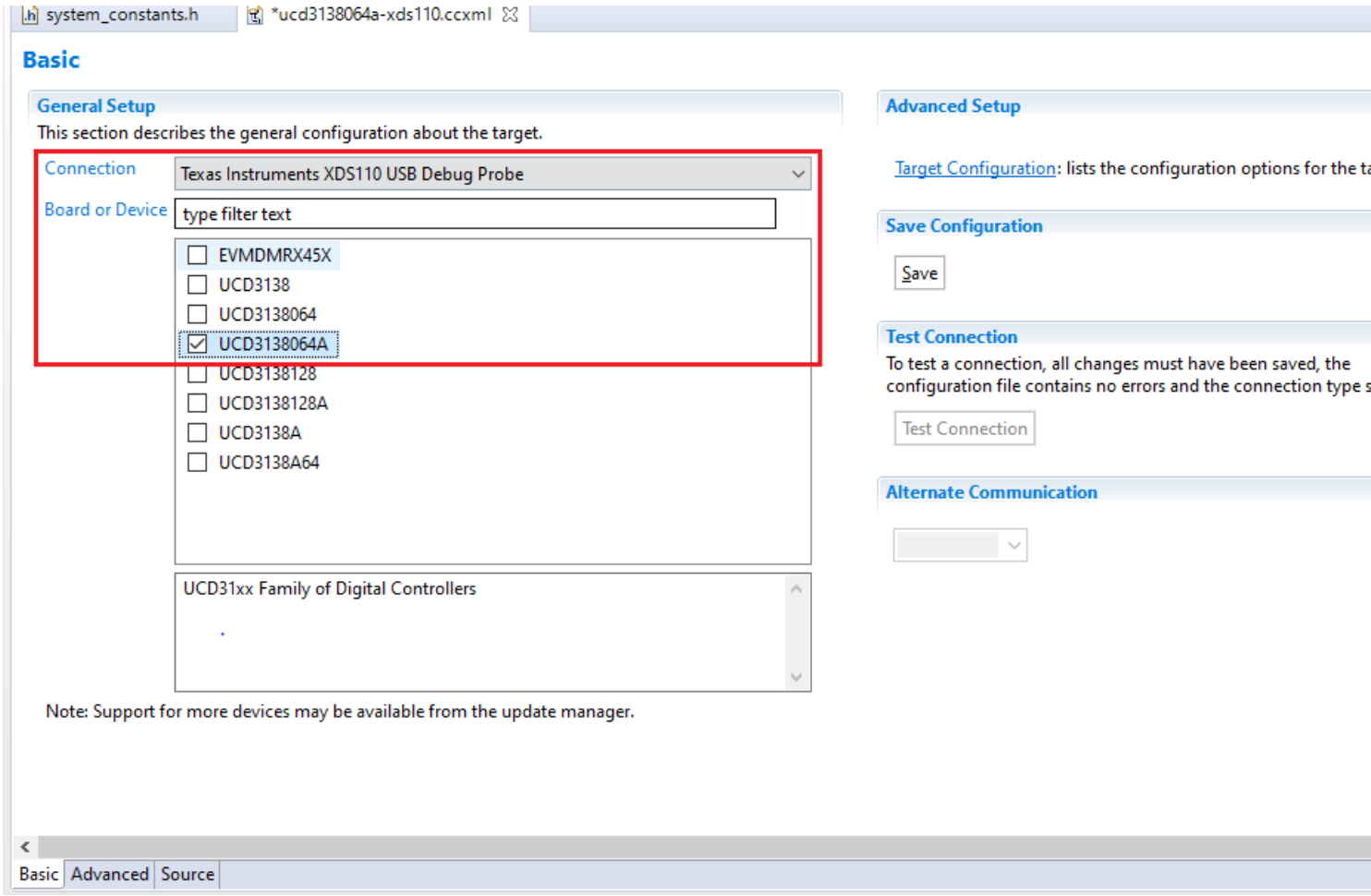

图 4-2. 仿真器类型和 UCD 型号选择

在“Advanced”选项卡中，将 TCLK 频率从 5.5MHz 更改为 1.0MHz。使用大于 1MHz 但小于 5.5MHz 的值进行实验。点击“Save”。

English Document: SLUAAP5
Copyright © 2023 Texas Instruments Incorporated

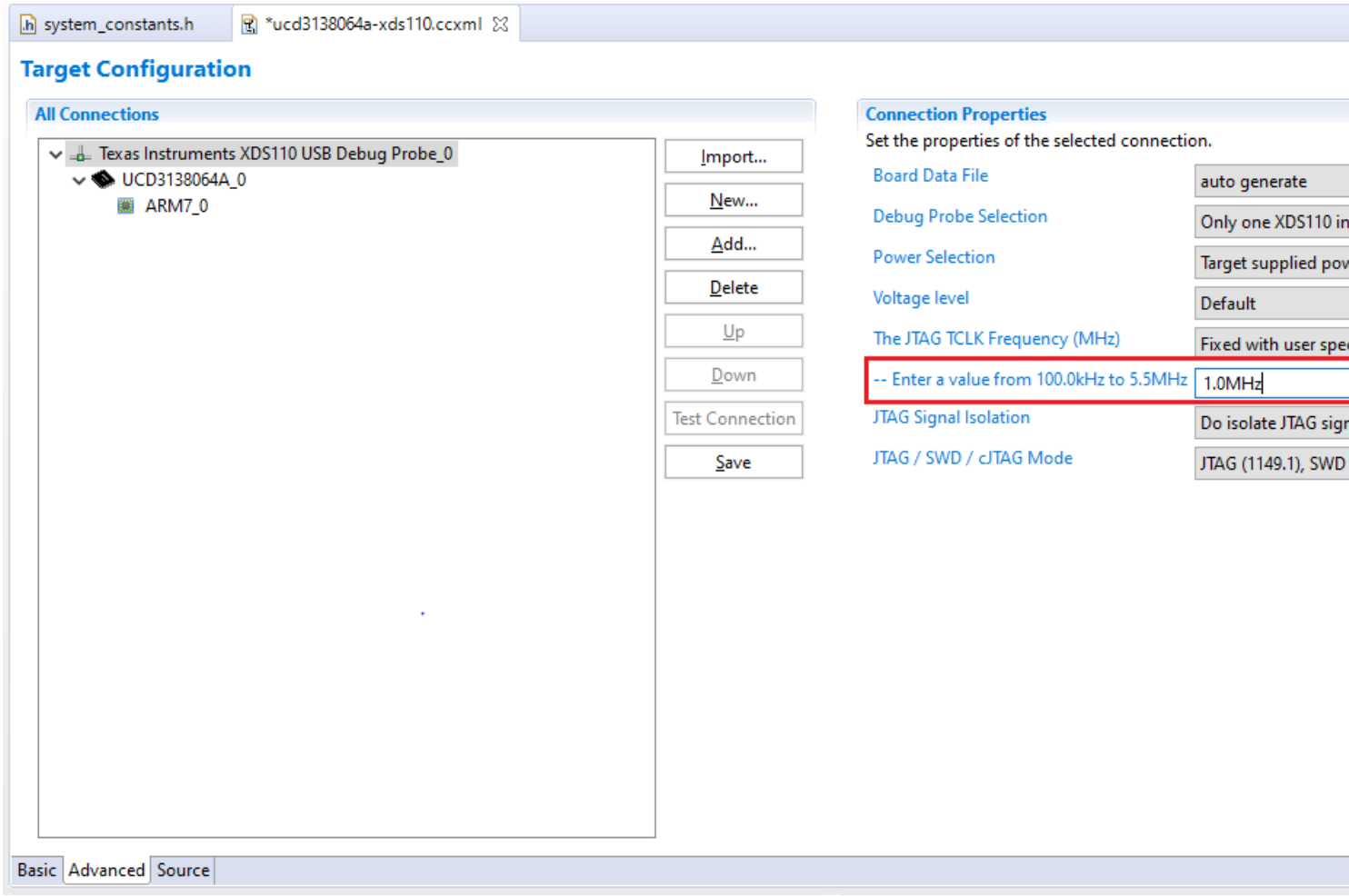

图 4-3. TCLK 频率配置

点击 *Test Connection*。测试完成后，向下滚动至弹出窗口的底部并观察是否显示如图 4-4 所示的消息。如果未显示该消息，则说明物理连接或设置存在问题。

English Document: SLUAAP5
Copyright © 2023 Texas Instruments Incorporated

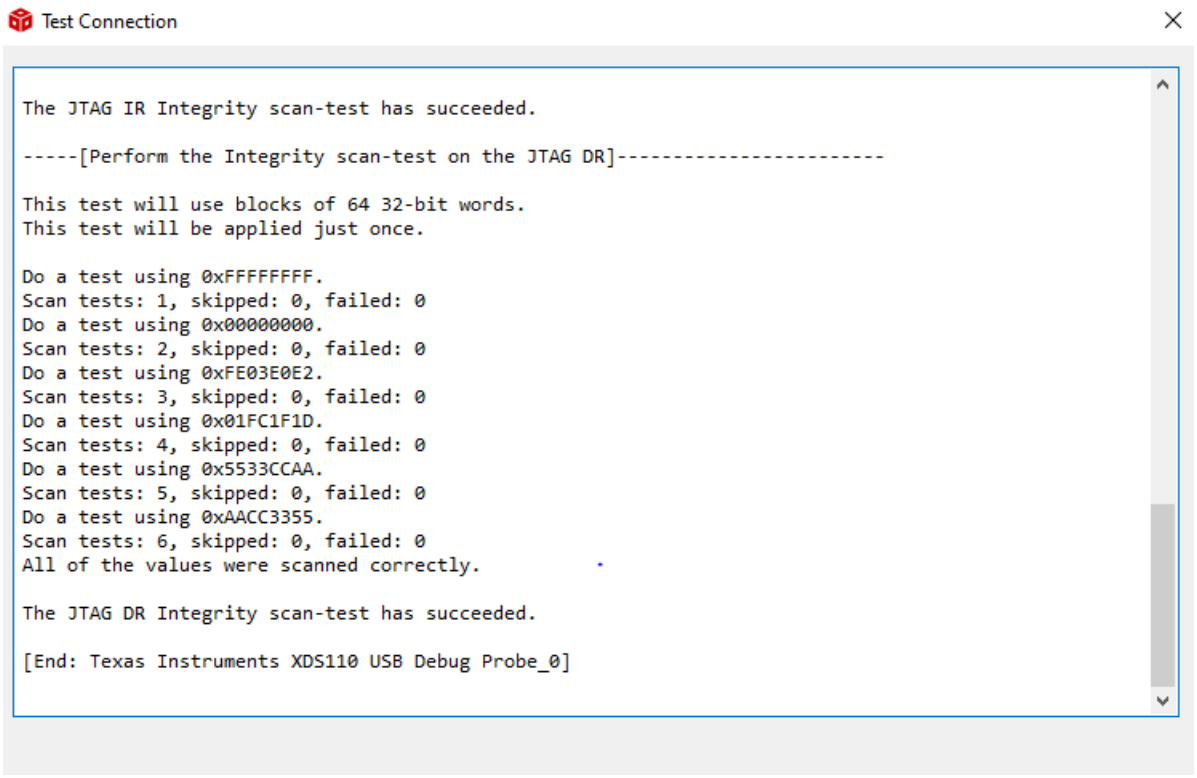

图 4-4. JTAG 连接成功

4.3 清除闪存

右键点击 ucd3138064a-xds110.ccxml 并选择 *Launch Selected Configuration*。此时会显示如图 4-5 所示的调试窗口。在顶部菜单上，点击 *Run* > *Connect Target*。

采用以下步骤：

- 转到 *Registers* 选项卡，清除 MFBALR1 寄存器中的 RONLY 位。
- 将 FLASHILOCK 寄存器的值设置为 0x42DC157E。
- 设置 PFLASHCTRL1 中的 MASS_ERASE 位。这会擦除 UCD3138064A 中的块 1。
- 退出调试模式并复位器件。此器件现在处于 ROM 模式。

Copyright © 2023 Texas Instruments Incorporated

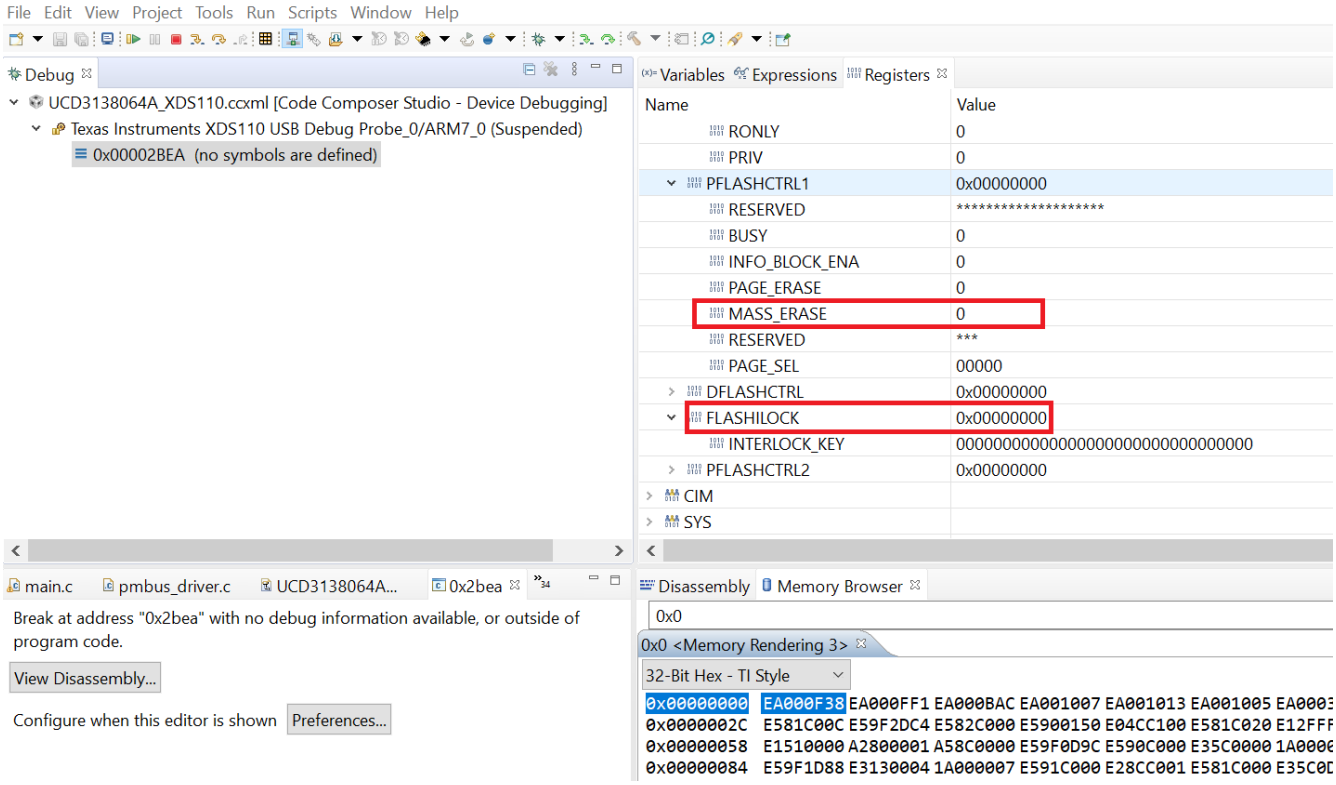

图 4-5. 调试模式

5 总结

当存在无法清除的有效校验和时，可能会发生锁定，这可能是由不正确的下载流程或存在故障的固件引起的。为了避免锁定，请在使用校验和进行编程之前检查列出的可能原因并按照说明进行操作。如果器件已锁定，但固件和 PMBus 正常运行，则器件能够通过 JTAG 恢复；否则便没有挽救方法。

6 参考文献

- 德州仪器 (TI)，*Fusion Digital Power Studio*。
- 德州仪器 (TI)，*UCD3138064 编程手册*。
- 德州仪器 (TI)，*UCD3138A64/UCD3138128 编程手册*。

English Document: SLUAAP5
Copyright © 2023 Texas Instruments Incorporated

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出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