针对 TMS320F28xxx DSC 的闪存编程解决方案

Tim Love and Pradeep Shinde

摘要

闪存编程过程出现在 TMS320F28xxx 数字信号控制器 (DSC) 开发周期中的所有阶段：固件调试、原型设计、生产、和现场重编程。 提供了几个解决方案来适应所有这些开发阶段的需要。 这个应用报告介绍了几个可用的解决方案以及这些解决方案用于开发周期的那个阶段。

内容

图片列表

图表列表

SPRAAL3 — http://www-s.ti.com/sc/techlit/SPRAAL3

1 简介

TMS320F28xxx DSC 的内部闪存存储器是一个巨大的优势，这是因为此存储器为非易失性内存，此类内存使得设计人员能够将应用代码存储在芯片内部，而无需连接外部内存来存储这个代码。

闪存存储器由一排内存单元（由浮栅晶体管制成）组成。 闪存的每个单元能够存储一位信息。 一个在浮动栅极上带有一个电荷的单元包含一个为 0 的值，而在浮动栅极上只有很少或者无电荷的单元包含一个为 1 的值。这项技术要求为闪存一直提供电源电压。 所有 TMS320F28xxx 器件包含 V_{DD3VFL}电压引脚，需要在此引脚上施加 3.3V 电压来进行编辑（写入）和读取闪存的操作。

由于采用了这项技术，如果要将应用代码存储在内存中，闪存必须经历一个擦除、编辑、和认证的过程。 针对这一功能所使用的算法是时间关键算法，此算法在 DSC 上从内部随机访问存储器 (RAM) 中执行。 这些算法必须被配置为适当的中央处理单元 (CPU) 频率并且不应被中断以确保闪存的正确编辑。 TI 在 [1]，[2]， [7]和 [8]中提供了闪存应用编程接口 (API) 算法。 所有在这个应用报告中讨论的闪存编程解决方案使用这些算法从您器件的接口来无缝编辑闪存。图 1显示了 a)JTAG，b) 串行，和 c) 定制解决方案所采用的闪存 API 的总体配置。

您首先要知道的是，TI 提供的闪存工具可以从网站http://www.ti.com/中下载。 按照以下路径：TI 主页→数字信号处理→处理器平台→ C2000™ 高性能 32 位控制器 → 选择F28x 代产品并点击 Flash Tools 按钮。

注: 有必要使用与 F28x 部件和其芯片版本相匹配的正确的闪存 API 版本。

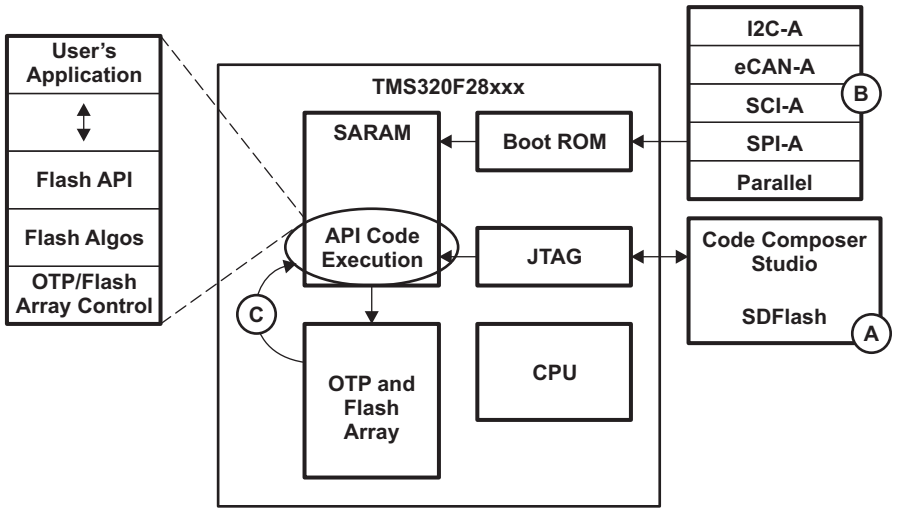

图 **1.** 闪存 **API** 代码执行

2 JTAG 解决方案

IEEE 标准 1149.1-1990，IEEE 标准测试访问端口和边界扫描架构 (JTAG) 解决方案可被应用到开发周期的所有阶段，但是主要用于固件调试和原型设计阶段，这是因为这个方法使得设计人员能够编辑闪存并随后在 Code Composer Studio™ 集成开发环境 (IDE) 中对其进行调试。 现有的几个解决方案包括 Code Composer Studio 片载闪存编程器，安全数据 (SD) 闪存、和 Flasher-C2000。 闪存编程工具取决于所使用的仿真器。

2.1 Code Composer Studio 片载闪存编程器

Code Composer Studio 片载闪存编程器是一款针对 Code Composer Studio 的插件，此编程器可实现 IDE 内的闪存编程，此 IDE 使用支持 eZdsp™ 开发板，并可与 Code Composer Studio 直接对接的仿真器。 在固件调试和原型设计阶段，这个编程器是最为便捷的 JTAG 选项，这是因为可通过 Code Composer Studio 直接访问此编程器。

可从 Code Composer Studio 的 Tools Menu (工具菜单) 中选择此编程器。 如果使用的是 Code Composer Studio 3.1 或者更老的版本，那么可从 F281x 闪存工具 [1]，F280x 闪存工具 [2]或者 Code Composer Studio 的更新导航 (Update Advisor) 中获得这个编程器。 如果使用 Code Composer Studio 3.3，这个编程器与 Code Composer Studio 的基础安装一起安装并且可通过更新导航中提供的服务通告进行更新。

图 2显示了 Code Composer Studio 片载闪存编程器图形用户接口 (GUI)。

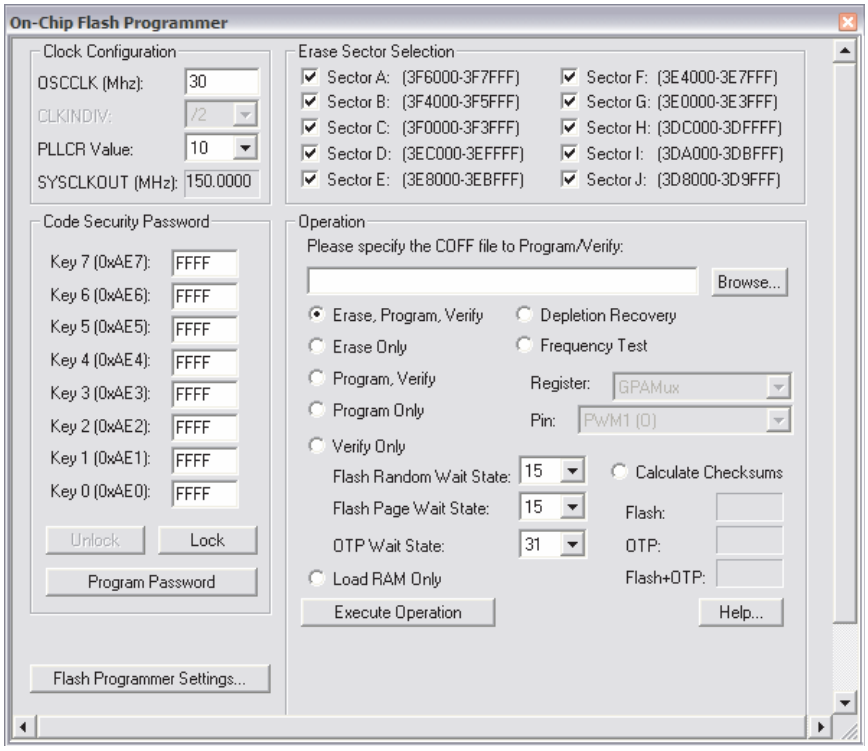

图 2. Code Composer Studio 片载闪存编程器

2.1.1 片载闪存编程器选项

片载闪存编程器有几个可供使用的选项/特性。 在 GUI 内部，有四个可视化部分以及闪存编程器设置按钮。下面的段落对每个部分的功能进行了讨论。

2.1.1.1 时钟配置

编程器的这个部分配置了用于闪存 API 算法的计时。 打开编程器时，Code Composer Studio 提示您配置图 3中显示的这些属性。

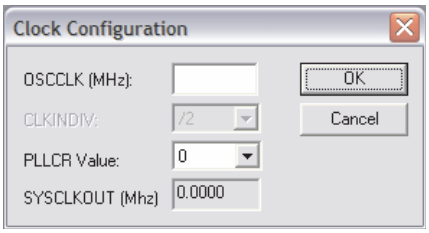

图 3. 时钟配置设置

按照这个提示，输入时钟频率将被指定并且 PLLCR 的值也将被设定。 基于这些输入，编程器计算 SYSCLKOUT 并且相应地配置算法。

2.1.1.2 闪存编程器设置

闪存编程器设置按钮打开了图 4中显示的菜单。

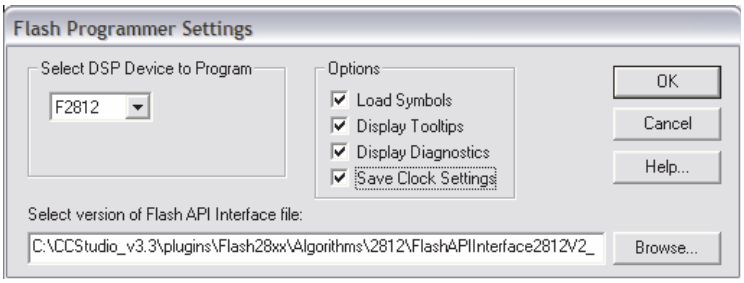

图 4. 闪存编程器设置菜单

在这菜单中选择闪存 API 文件。 在选择这个算法时，Browse（浏览）按钮将您引导至系统中可用的算法。一直选择最新的算法。

可从这个菜单选择器件。 这个是 Code Composer Studio 所配置的器件的默认值。 其它提供的选项有：

- 载入符号-在编辑闪存之后立即启用/禁用 Code Composer Studio 载入符号功能来启用调试。
- 显示工具提示-当对选项不确定时，启用/禁用编程器内的细节显示
- 显示诊断-启用/禁用显示编程状态的输出屏幕。
- 保存时钟设置-启用/禁用此选项来保存节 2.1.1.1中输入的时钟配置。

2.1.1.3 擦除扇区选择

编程器的这个部分使您能够选择擦除所有闪存存储器或者仅仅是闪存存储器的特定部分。 如果在某个扇区内编辑的代码不应被擦除，这个选项将有助于保存这个闪存内的内容。 可取消选定未使用的扇区，并且也不会擦除此扇区，这样也有助于减少总体擦除时间。

2.1.1.4 运行

编程器的操作部分包含以下几个特性：

- 指定一个 COFF 文件 - 对闪存进行编程的 .out 文件在这个部分被输入。 如果一个项目是在 Code Composer Studio 中打开并建立的，那么生成的 .out 文件被自动指定。
- 擦除、编辑、和认证操作-这些按钮可实现擦除、编辑、和认证操作同时运行或者独立执行每一个操作。
- 删除恢复-这个选项调用删除恢复算法来寻找删除的扇区并尝试恢复这些扇区。
- 频率测试-从节 2.1.1.1指定的时钟配置可在通用输入/输出 (GPIO) 选择引脚上进行校验。
- 计算校验和-执行一个闪存、一次性可编辑 (OTP)、和闪存 + OTP 的校验和。

- 只载入 RAM - 这个选项载入被指定从 RAM 运行的已初始化部分。
- 写入状态-编程期间，使用这些选项来设定闪存和 OTP 的等待状态。
- 执行操作-这个按钮将执行任何选择的选项。
- 帮助-这个按钮将打开与编程器相关的所有帮助文件文档。

2.1.1.5 代码安全密码

编程器的代码安全密码部分直接访问闪存存储器的代码安全模块 (CSM)。 CSM 由 8 个可实现闪存密码保护的 16 位存储单元。 与 CSM 有关的详细信息，请见 [10]， [11]和 [12]的《*代码安全模块 (CSM)*》部分。

编程器的这个部分包含下列元件：

- KEY0-KEY7-16 位密码单元。
- 解锁-如果 CSM 是安全的并且密码被写入 KEY0-KEY7 中，则解锁闪存。
- 锁住-如果 CSM 之前已经被写入到 KEY0-KEY7 中的密码编辑。
- 程序密码-设定写入到 KEY0-KEY7 的 CSM 密码。 对于这个过程，闪存需要被擦除。 如果程序密码按钮被按下时，闪存已经被擦除，闪存插件将提示您擦除闪存。

在使用 CSM 时，有几点需要注意。 更多细节，请见6.3 节。

2.1.2 编程示例

闪存的编程和调试过程只包含少数几步。 对于这个示例，使用了 TMS320F28335 eZdsp，Code Composer Studio 3.3，和《*在 TMS320F28xxx DSP 上运行一个来自内部闪存存储器的应用*》（文献编号：SPRA958） [3]中使用的闪存示例。 这个过程可用于所有 TMS320F28xxx DSC。

1. 使用一个 JTAG 仿真器将目标板连接至 PC 并使用适当的电源连接器为目标板供电。
2. 启动 Code Composer Studio，启动时已经在 Code Composer Studio 设置工具中选择了合适的仿真驱动器。
3. 通过先选择 Project→Open，随后选择 Project→Rebuild All 来打开并建立项目。
4. 从 Tools Menu （工具菜单）中打开片载闪存编程器。

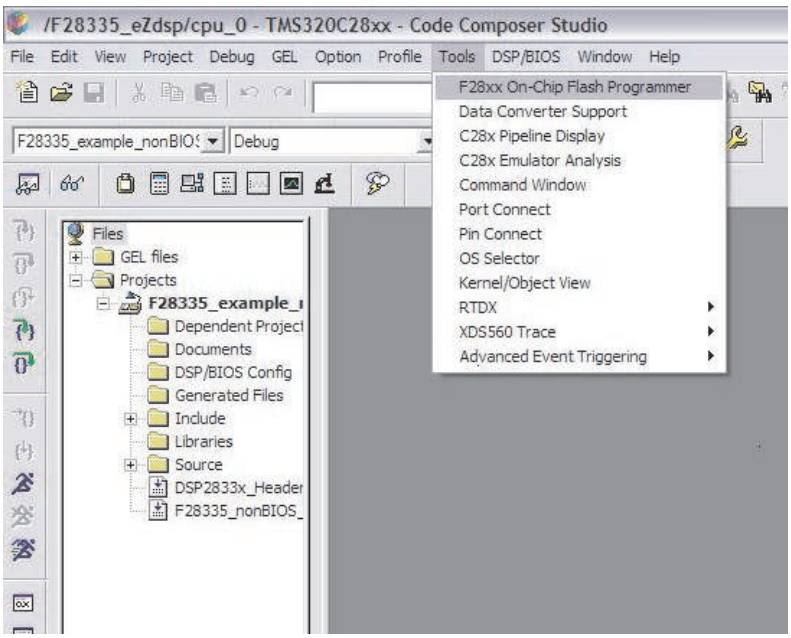

图 **5.** 调用片载闪存编程器

5. 为节 2.1.1.1中描述的目标板配置时钟设置。

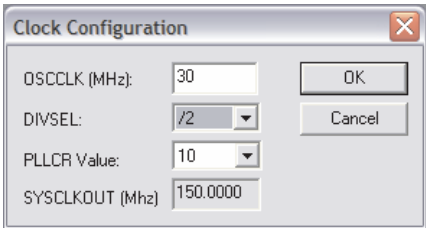

图 **6.** 示例时钟配置设置

6. 为节 2.1.1.2中描述的编程器选择正确的算法。

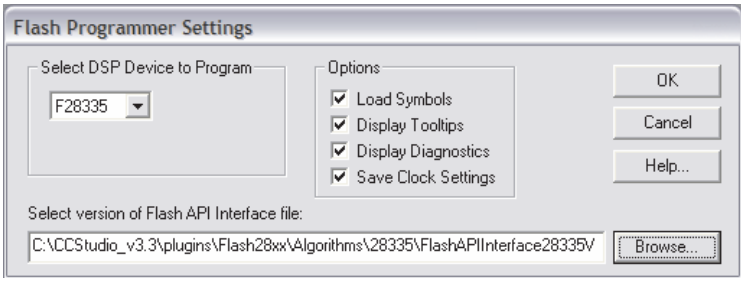

图 **7.** 示例闪存编程器设置

7. 选择节 2.1.1.3中所描述的将被擦除/编辑的扇区。
8. 缺省情况下，将执行操作选为擦出、编程、验证选项并且 .out 文件已经在项目建立时被指定。

一旦此过程结束，使用6.1 节中描述方法，通过 Code Composer Studio，程序被存储在闪存中并且电路板已经为独立运行或者调试做好准备。

2.2 SD 闪存

SD 闪存是一款免费的、独立的编程器，此编程器无需 Code Composer Studio 即可使用一个频谱数字仿真器实现编程。 由于这一特性，这个编程器也可用于一个产品的所有开发阶段。 工具、算法、示例项目、和全部对工具使用进行解释说明的文档可从 SD 闪存工具中主页获得 [4]。

图 8中显示了 SD 闪存 GUI。

图 **8. SD** 闪存

SD 闪存使用项目来配置用于 JTAG 连接、擦除、编辑、和校验选项的所有设置。 从 File 菜单从可创建一个新项目或者打开一个现有项目。 一旦项目被打开，可通过选择 Project→Settings 来配置设置。

2.2.1 目标方

图 9显示了目标方选项。 目标方标签配置针对器件选项的 JTAG 连接。

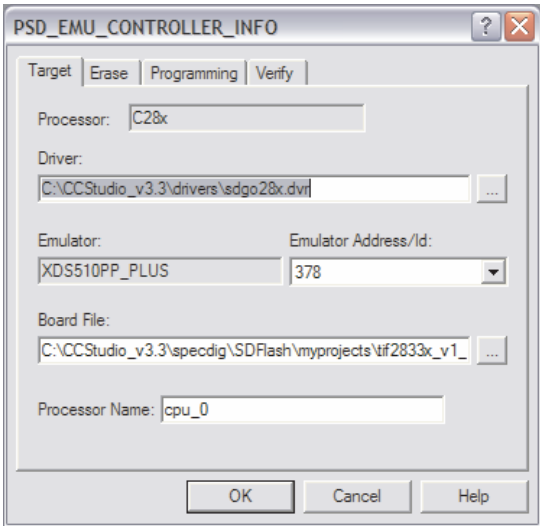

图 **9. SD** 闪存目标方选项

这个标签页内的选项如下：

- 驱动器-选择用于与器件通信的 Code Composer Studio 仿真驱动器
- 仿真器地址 / ID - 选择针对 JTAG 仿真器的地址。 针对 XDS510PP 的缺省值为 0x378，而针对 XDS510USB 的缺省值为 510。

- 电路板文件-向 SD 闪存提供与 JTAG 扫描链上的器件数量相关的信息。
- 处理器名称-用于在项目内为器件放置一个总名称。

2.2.2 擦除

图 10显示了配置所有擦除过程的擦除标签页。

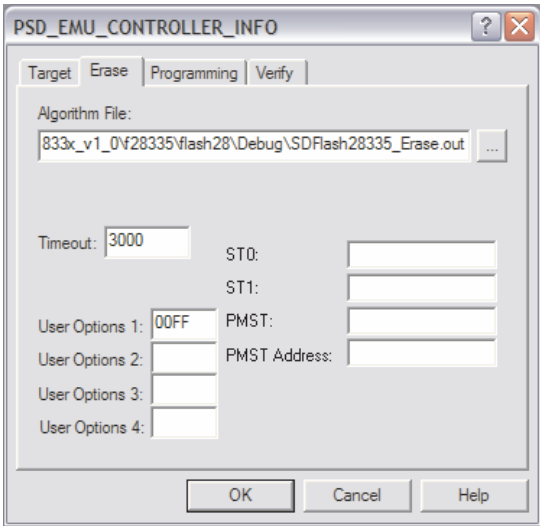

图 10. SD 闪存擦除选项

这个标签页内的选项如下：

- 算法文件-为项目指定擦除算法。 针对使用的特定器件，应该选择这个文件。
- 计时-以秒为单位为擦除过程计时。
- 用户选项 1 - 指定擦除的扇区。缺省值为 00FF 并将指定所有扇区。
- 用户选项 2 / 用户选项 4 - 未使用。
- 用户选项 3 - 运行频率切换测试。
- ST0/ST1/PMST/PMST 地址-缺省值为空白且不应被使用。

2.2.3 编程

图 11显示了配置所有编程过程选项的编程标签页。

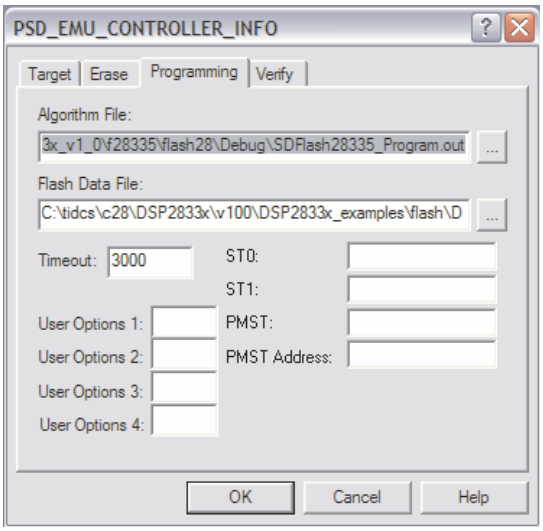

图 **11. SD** 闪存编程选项

这个标签页内的选项如下：

- 算法文件-为项目指定编程算法。 应该为所使用的特定器件选择此文件。
- 闪存数据文件-为闪存编程指定由 Code Composer Studio 项目生成的 .out 文件
- 计时-以秒为单位为编程过程计时。
- 用户选项 3 - 运行频率切换测试。
- 用户选项 1 / 用户选项 2 / 用户选项 4 - 未使用。
- ST0/ST1/PMST/PMST 地址-缺省值为空白且不应被使用。

2.2.4 验证

图 12显示了配置所有验证过程选项的校验标签页。

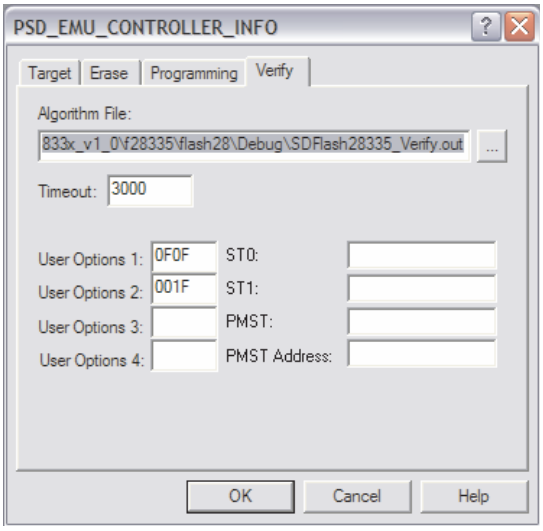

图 **12. SD** 闪存验证选项

这个标签页内的选项如下：

- 算法文件-为项目指定校验算法。 应该为所使用的特定器件选择此文件。

- 计时-以秒为单位为校验过程计时。
- 用户选项 1 / 用户选项 2 - 指定校验操作期间针对闪存和 OTP 的等待状态。
- 用户选项 3 - 运行频率切换测试。
- 用户选项 4 - 未使用。
- ST0/ST1/PMST/PMST 地址-缺省值为空白且不应被使用。

2.2.5 编程示例

闪存编程过程只包含少数几个步骤。 对于这个示例，使用了 TMS320F28335 eZdsp，SampleF28335usb.sdp（提供的 SD 闪存项目），以及《从 *TMS320F28xxx DSP 上的内部闪存存储器上运行一个应用*》（文献编号：SPRA958） [3]中的闪存示例。 这个过程可用于所有 TMS320F28xxx DSC。

1. 从 SD 闪存中下载 SD 闪存工具和算法 [4]。
2. 使用 JTAG 仿真器将目标板连接至 PC 并使用合适的电源连接器为电路板供电。
3. 打开 SD 闪存。
4. 选择 File→Open Project 来打开提供的 SD 闪存项目。
5. 按需要修改节 2.2.1，节 2.2.2，节 2.2.3，和节 2.2.4中描述的任何项目选项。 （在这个示例中，.out 文件在闪存数据文件中进行更改来使用《从 *TMS320F28xxx DSP 上的内部闪存存储器上运行一个应用*》（文献编号：SPRA958） [3]中的 .out 文件。）
6. 选择 File→Save Project As 来保存项目文件。
7. 选择 Device→Reset 来复位器件。
8. 选择 Device→Flash 来编辑闪存。 如图 13所示，将打开一个独立菜单。 可为闪存过程选择所有选项或者只选择特定选项。

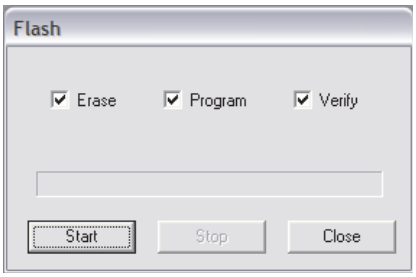

图 **13.** 闪存窗口

一旦这个过程结束，程序被存储在闪存中并且使用6.1 节描述的方法，通过 Code Composer Studio，电路板为独立运行或者调试做好准备。 与 SD 闪存一同提供的文档详细解释了这个部分中描述的步骤并且提供了与时钟速率配置，CSM，删除恢复等相关的信息。 建议阅读本文档以获得与 SD 闪存有关的全部信息。

2.3 Flasher-C2000

Flasher-C2000 是一个独立编程器，此编程器能够使用一个 Signum™ JTAGjet-TMS-C2000 仿真器进行编程而无需 Code Composer Studio，从而使这个编程器可用于一个产品的所有开发阶段。 可在 JTAGjet-TMS-C2000 主页http://www.signum.com/Signum.htm?p=c2000.htm [5]上购买此工具。 对于与这个编程器相关的进一步信息请与 Signum Systems Corp. 联系。

3 串行解决方案

SD 闪存还有通过 SCI-A 外设（使用串行通信接口 (SCI) 引导 ROM 选项）编辑闪存的免费 RS-232 算法。由于 SD 闪存是独立运行的并且唯一所需的连接为一条标准 RS-232 线缆，这个编程解决方案可用于生产和开发周期的现场编程阶段。

这个解决方案包含的图形接口与2.2 节中显示的一致并且使用一样的选项。 唯一的区别是使用 RS-232 接口而非 JTAG。 通过使用这个方法，这个闪存的编程步骤除了连接 RS-232 之外与节 2.2.5中的编程步骤一致。

1. 在 SCI-A 端口和主机 PC 之间连接一个串行线缆。
2. 将 F28xxx 器件设定为引导至 SCI-A 串行引导加载。
3. 为了访问引导 ROM，将 XMP/MC 引脚拉至低电平。
4. 断开任一有可能被连接至电路板的 JTAG 仿真器。

这些步骤的全部说明文档和其它与这个编程解决方案、工具、和示例相关的信息可从http://emulators.spectrumdigital.com/utilities/sdflash/ [4]中下载。

4 嵌入式解决方案

一些应用闪存存储器或者 OTP 块的全部或者部分重编程来重新配置应用代码或者将数据（例如，校准参数）存储在非易失性内存中。 这个操作也被称为电路内编程；在这个环境中，CPU 编辑系统上的闪存。

4.1 所需的设置

由于任何闪存操作都需要闪存 API 程序，在嵌入式编程期间闪存 API 程序必须可用。 驻留在闪存中的应用软件需要包括闪存 API。 正如之前讨论的那样，闪存例程（擦除、写入操作）必须遵守严格的时序要求。 因此，由于它较慢的速度，它们不能从闪存运行。 运行时，整个闪存 API 需要被复制到 RAM；将被写入的代码/数据也需要驻留在 RAM 中。

将闪存编程电压引脚 V_{DD3VFL}连接至 3.3V 电源轨，这一点很重要。 请注意，闪存擦除和写入操作在操作期间会汲取额外的电流。 为了获得汲取自 V_{DD}和 V_{DDIO}电源轨的电流的典型值，请见特定器件数据表的闪存参数表。 为了实现可靠运行，系统电源必须能够支持这个过多的电流需求。

4.2 闪存 API

这个部分仔细检查了闪存 API，这是因为针对不同的闪存操作，闪存 API 由 CPU 驻留和调用。

API 库包括擦除、编程和校验闪存阵列的功能。 一次可以擦除的最少数量内存是一个单一扇区。 闪存 API 擦除功能包括闪存预调节并且需要一个独立的*清除*步骤。 程序功能运行在闪存阵列和 OTP 块上。 程序功能只能将位从 1 改为 0。编程功能不能将位从 0 改回为 1。 编程功能每次运行在一个单一 16 位字上。

对于闪存 API 库的所有选项和操作流程的全部细节，请见包括在 API 压缩文件内的 API 文档。

4.3 闪存 API 清单

作为参考并实现文档完整性，从 API 文档中提取了下列数据。

将闪存 API 集成进入用户软件要求系统设计人员执行操作来满足几个关键需求。 下列清单概括了使用 API 所需的步骤。 这些步骤在指明的参考部分中进行了详细讨论。 这个清单应用到所有 F2823x 闪存 API 中并且取自 F2823x 闪存 API 文档。 这个总清单应用于所有 F28xxx 闪存 API 库并且可在每个库的各自的文档中找到。

在使用 API 前，完成以下步骤：

1. 针对您的目标运行环境修改 Flash823x_API_Config.h。
2. 将 Flash2823x_API_Library.h 包括在您的源代码中。
3. 将合适的闪存 API 库添加到您的项目中。
   - 当使用闪存 API 时，使用较大内存模型构建您的代码。
   - API 库内置了 28x 项目代码 (OBJMODE=1，AMODE=1)

在调用任一闪存 API 功能前，在您的应用中进行以下操作：

1. 初始化锁相环 (PLL) 控制寄存器 (PLLCR) 并等待 PLL 锁定。
2. 确保 PLL 没有运行在跛行模式。 如果 PLL 运行在跛行模式，不要调用任何 API 函数，这是因为器件将不能运行在适当的频率上。
3. API 必须从零等待状态内部 SARAM 中执行（可选）。 如果 API 从闪存 / OTP 复制到内部 SARAM 内存中，那么按照这个部分中的指令操作。
4. 初始化 32 位全局变量 Flash_CPUScaleFactor。
5. 初始化全局函数指针 Flash_CallbackPtr 来指向应用的回调函数。 或者，将指针设定为 NULL。
6. 在调用一个 API 函数前禁用全局中断（可选）。
7. 在进行任何 API 调用前，请首先理解在本部分中进行了详细说明的 API 限制。
8. 运行频率切换测试来确认闪存 API 适当的频率配置（可选）。

> **注:** 切换测试功能将一直执行。 您必须暂停处理器来停止这个测试。

9. 解除对 CSM 的锁定（可选）。
10. 调用 API 参考中描述的闪存 API 函数。

被调用的闪存 API 函数将进行以下操作：

- 安全装置定时器被禁用。
- 检查 PARTID（内存位置 0x882）寄存器来确保此部件为正确的器件
- 检查引导 ROM 中 0x3FFFB9 中 API 版本的内容与芯片修订版本兼容性之间的关系。
- 执行被调用的操作并且：
  - 在时间关键代码段附近禁用并且恢复全局中断（通过 INTM，DBGM，XNMICR）。
  - 如果 Flash_CallbackPtr 不为 NULL，则调用回调函数。
- 返回成功或者一个错误代码。 这些在 F2823x_API_Library.h 中进行了定义。

然后，您的代码应该进行如下操作：

1. 检查错误代码的返回状态。
2. 重新启用安全装置定时器（可选）。

4.4 闪存 API 进行的操作和不进行的操作

- API 进行的操作
  - 从零等待状态内部 SARAM 内存中执行闪存 API 代码。 F2823x 和 F2833x 器件包含零等待状态 (L0-L3) 和一个等待状态 SARAM (L4-L7)。 闪存 API 应该从 L0-L3 SARAM 中运行。
  - 针对正确的 CPU 运行频率来配置 API。
  - 按照闪存 API 文档的清单来将 API 整合至一个应用。
  - 在调用一个 API 函数前，初始化 PLLCR 并等待 PLL 锁定。
  - 初始化 API 回调函数指针 (Flash_CallbackPtr)。 如果将不使用回调函数，那么最好如 API 文档中*初始化回调函数指针*部分所描述的那样明确地将函数指针设定至 NULL。 初始化回调函数指针故障会导致代码分支至一个未定义的位置。
  - 仔细检查文档中描述的针对回调函数、中断、和安全装置的 API 限制。
- API 不进行的操作
  - 不要从闪存或者 OTP 中执行闪存 API。 如果 API 存储在闪存或者 OTP 内存中，在执行之前，它们必须首先被复制到内部零等待状态 SARAM。
  - 在一个来自闪存或 OTP 内存块的擦除、编程或者删除恢复 API 函数期间，不要执行任何会出现的中断处理例程 (ISR)。 在 API 函数完成并且推出之前，闪存和 OTP 不可用于程序执行和数据存储。
  - 不要从闪存或者 OTP 中执行 API 回调函数。 擦除、编程或者删除恢复例程期间，当回调函数被 API 调用时，闪存和 OTP 不可用于程序执行和数据存储。 只有在 API 函数完成且退出后，闪存和 OTP 才可用。
  - 不要停止正在执行的擦除、编程或者删除恢复函数（例如，不要停止 API 代码内的调试器、不要复位部件等）。
  - 在闪存和/或者 OTP 正在被擦除、编辑时或者在删除恢复期间，不要执行代码或者从闪存阵列取数据。

5 生产解决方案

由非工程技术人员完成新生产出部件的应用代码编辑并且需要更快速的完成。 根据生产量，提供了不同的解决方案。

对于小批量生产，一个针对内部编程的合适选项为可从第三方，例如 Spectrum Digital 和 Signum，获得的串行工具。 有几个不与 Code Composer Studio 相连接的独立软件工具（在 PC 上运行）。 它们通常接受项目的 COFF (.out) 文件并且与 CPU 通过 SCI/RS232 端口进行通信。 详细信息请检查这些销售商的网站。

德州仪器 (TI) 分销商 Arrow 和 Avnet 提供使用应用镜像文件来编辑器件的服务。 这一点可用于中批量生产 (>1K)。

对于大批量生产，请考虑由 BP Micro 和 Data I/O 等销售商提供的独立闪存编程器。它们经常在部件列表中添加可被编辑的全新器件。

所有这些选项可通过 F281x 闪存工具 [1]和 F280x 闪存工具 [2]获得。

6 调试、有用的文档、和注意事项

这个应用报告中介绍的闪存编程解决方案使得设计人员能够在器件上存储并且运行应用代码。 正因如此，一个设计人员能够使用 Code Composer Studio 从闪存到 JTAG 进行直接调试。 当使用闪存存储器和编程解决方案时，还需要将几个有用的文档、注意事项等考虑在内。

6.1 在 Code Composer Studio 环境中调试

在使用其中一种介绍的闪存解决方案的闪存过程完成之后，在 Code Composer Studio 内可从闪存中直接调试程序。 当从闪存调试时，可使用所有标准调试选项：运行、暂停、单步执行、复位 CPU 等。

唯一一个需要从闪存编辑的步骤就是选择 File→Load Symbols→Load Symbols Only 并选择被编程至闪存的 .out 文件。

一个对于调试的限制就是断点的使用。 由于调试发生自闪存存储器，您只能使用两个硬件断点。 通过选择 Toggle Breakpoint（切换断点）选项可以像设定软件断点一样来设定这些断点（请见图 14）。

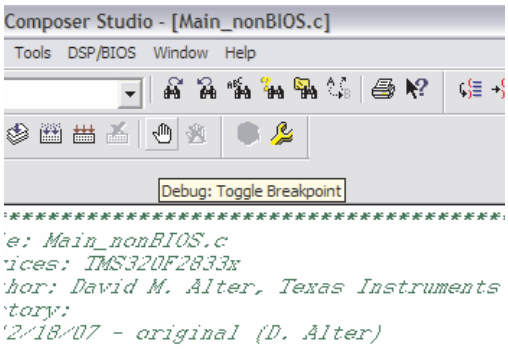

图 14. 切换断点

注: Code Composer Studio 自动为程序的末尾设定两个断点并且 CIO 防止您有任何用于调试的断点。 通过引导至 Option→Customize→Program/Project/CIO 并且选中两个名称为 Do Not Set End of Program Breakpoint At Load 和 Do Not Set CIO Breakpoint At Load 的复选框可启用这些断点。

6.2 *有用的文档*

与编程解决方案一起提供的文档对于描述如何使用工具和需要在工具内设定的选项十分有用。 然而，在工具可用前，项目必须被配置为从闪存运行。 TI 提供了一个详细的文档来描述如何将一个基于 RAM 的项目转换为一个基于闪存的项目并且提供了显示一个已转换项目的示例。 更多细节，请参阅《从 *TMS320F28xxx DSP 上的内部闪存存储器中运行一个应用*》（文献编号：SPRA958 [3]）。

TI 的其它文档涉及将代码载入到闪存，以及为了实现更快速的执行，随后在运行时间将完整代码复制到 RAM 中。 更多细节，请参阅《*在 TMS320F28xxx DSP上将编译程序段从闪存复制到 RAM*》（文献编号：SPRAAU8） [6]。 [3]还讨论了在运行时只将代码段从闪存复制到 RAM，复制的代码段只是指定的代码函数而非整个代码。

最后，TI 有一个显示 TMS320F281x 引导 ROM 的 SCI 引导选项用法的文档，这个选项执行闪存（使用闪存 API）的串行编程。 更多细节，请参阅《*TMS320F281x 引导 ROM 串行闪存编程*》（文献编号：SPRAAQ2） [9]。 这是一个在现场重编程中非常有用的文档。

6.3 预防措施

有几个预防措施应该引起您的注意，这些措施可以防止闪存永久锁定或者将闪存置于一个未知的状态：

- 在执行当中不要停止擦除过程，因为这样会永久地锁住 CSM 并且还会导致闪存单元的删除。 擦除过程首先将所有单元编辑为零，然后在所有位被擦除前，从扇区的位中移除电荷。 如果在 CSM 单元被设定为零之后且电荷被移除之前执行停止，它将被永久锁定。 CSM 位置中的所有零是一个实现持久安全的芯片特性。 由于不允许擦除过程执行其后置条件步骤也可引起删除的发生，这就确保了位不会保留在已删除状态。

注: 如果闪存处于删除模式，与 Code Composer Studio 片载闪存编程器和 SD 闪存一起提供的删除恢复功能可被用于尝试使闪存脱离删除模式。

- 确保保存 CSM 密码的地址单元 0x3F7FF8-0x3F8000 未被用于 Code Composer Studio 项目连接器命令文件内的代码分配。 如果代码被载入到 CSM 密码单元，密码将被项目代码写覆盖并且密码变成未知，从而使得闪存被永久锁定。
- 当使用闪存 API 时，虽然从闪存运行要求算法从内部 RAM 运行，但是您不能修改闪存存储器。
- 闪存编程电压引脚 V_{DD3VFL}应该被一直连接至 3.3V 电源轨上，这一点很重要。 需要这个电压来读取/编辑闪存。 请注意，在操作期间，闪存擦除和写入操作会汲取额外的电流。 要获得汲取自 V_{DD}和 V_{DDIO}电源轨的电流的典型值，请见器件专用数据表中的闪存参数表。 为了实现可靠运行，系统电源必须能够支持超过这个电流要求的电流。

7 结论

这个应用报告显示出开发周期分为几个阶段并为适应这些阶段提供了闪存编程解决方案。 这些选项包括：JTAG，串行、嵌入式、和生产解决方案。 有一个推荐的文档来帮助开发以及提供一些指南和需要注意的隐患。 借助于所有这些信息，您将能够很好地处理 TMS320F28xxx DSC 的闪存存储器。

8 参考

1. F281x 闪存工具：http://focus.ti.com/dsp/docs/dspplatformscontento.tsp?sectionId=2&familyId=1406&tabId=2026
2. F280x 闪存工具：http://focus.ti.com/dsp/docs/dspplatformscontento.tsp?sectionId=2&familyId=510&tabId=517
3. 《从 *TMS320F28xxx DSP* 上的内部闪存存储器中运行一个应用》（文献编号：SPRA958）
4. SD 闪存：http://emulators.spectrumdigital.com/utilities/sdflash/
5. Signum：http://www.signum.com/Signum.htm?p=c2000.htm
6. 《在 *TMS320F28xxx DSC* 上将编译器部分从闪存复制到 *RAM*》（文献编号：SPRAAU8）
7. *下载：TMS320F2823x 闪存 API* (SPRC665)
8. *下载：TMS320F2833x 闪存 API (v2.00)* (SPRC539)
9. 《*TMS320F281x 引导 ROM 串行闪存编程*》（文献编号：SPRAAQ2）
10. 《*TMS320x281x DSP 系统控制和中断参考指南*》（文献编号：SPRU078）
11. 《*TMS320x280x，2801x，2804x DSP 系统控制和中断参考指南*》（文献编号：SPRU712）
12. 《*TMS320x2833x，2823x 系统控制和中断参考指南*》（文献编号：SPRUFB0）

重要声明

德州仪器(TI) 及其下属子公司有权在不事先通知的情况下, 随时对所提供的产品和服务进行更正、修改、增强、改进或其它更改，并有权随时中止提供任何产品和服务。客户在下订单前应获取最新的相关信息，并验证这些信息是否完整且是最新的。所有产品的销售都遵循在订单确认时所提供的TI 销售条款与条件。

TI 保证其所销售的硬件产品的性能符合TI 标准保修的适用规范。仅在TI 保证的范围内，且TI 认为有必要时才会使用测试或其它质量控制技术。除非政府做出了硬性规定，否则没有必要对每种产品的所有参数进行测试。

TI 对应用帮助或客户产品设计不承担任何义务。客户应对其使用TI 组件的产品和应用自行负责。为尽量减小与客户产品和应用相关的风险，客户应提供充分的设计与操作安全措施。

TI 不对任何TI 专利权、版权、屏蔽作品权或其它与使用了TI 产品或服务的组合设备、机器、流程相关的TI 知识产权中授予的直接或隐含权限作出任何保证或解释。TI 所发布的与第三方产品或服务有关的信息，不能构成从TI 获得使用这些产品或服务的许可、授权、或认可。使用此类信息可能需要获得第三方的专利权或其它知识产权方面的许可，或是TI 的专利权或其它知识产权方面的许可。

对于TI 的产品手册或数据表，仅在没有对内容进行任何篡改且带有相关授权、条件、限制和声明的情况下才允许进行复制。在复制信息的过程中对内容的篡改属于非法的、欺诈性商业行为。TI 对此类篡改过的文件不承担任何责任。

在转售TI 产品或服务时，如果存在对产品或服务参数的虚假陈述，则会失去相关TI 产品或服务的明示或暗示授权，且这是非法的、欺诈性商业行为。TI 对此类虚假陈述不承担任何责任。

TI 产品未获得用于关键的安全应用中的授权，例如生命支持应用（在该类应用中一旦TI 产品故障将预计造成重大的人员伤亡），除非各方官员已经达成了专门管控此类使用的协议。购买者的购买行为即表示，他们具备有关其应用安全以及规章衍生所需的所有专业技术和知识，并且认可和同意，尽管任何应用相关信息或支持仍可能由TI 提供，但他们将独力负责满足在关键安全应用中使用其产 品及TI 产品所需的所有法律、法规和安全相关要求。此外，购买者必须全额赔偿因在此类关键安全应用中使用TI 产品而对TI 及其 代表造成的损失。

TI 产品并非设计或专门用于军事/航空应用，以及环境方面的产品，除非TI 特别注明该产品属于“军用”或“增强型塑料”产品。只 有TI 指定的军用产品才满足军用规格。购买者认可并同意，对TI 未指定军用的产品进行军事方面的应用，风险由购买者单独承担，并且独力负责在此类相关使用中满足所有法律和法规要求。

TI 产品并非设计或专门用于汽车应用以及环境方面的产品，除非TI 特别注明该产品符合ISO/TS 16949 要求。购买者认可并同意，如果他们在汽车应用中使用任何未被指定的产品，TI 对未能满足应用所需要求不承担任何责任。

可访问以下URL 地址以获取有关其它TI 产品和应用解决方案的信息：

	产品		应用
数字音频	www.ti.com.cn/audio	通信与电信	www.ti.com.cn/telecom
放大器和线性器件	www.ti.com.cn/amplifiers	计算机及周边	www.ti.com.cn/computer
数据转换器	www.ti.com.cn/dataconverters	消费电子	www.ti.com/consumer-apps
DLP® 产品	www.dlp.com	能源	www.ti.com/energy
DSP - 数字信号处理器	www.ti.com.cn/dsp	工业应用	www.ti.com.cn/industrial
时钟和计时器	www.ti.com.cn/clockandtimers	医疗电子	www.ti.com.cn/medical
接口	www.ti.com.cn/interface	安防应用	www.ti.com.cn/security
逻辑	www.ti.com.cn/logic	汽车电子	www.ti.com.cn/automotive
电源管理	www.ti.com.cn/power	视频和影像	www.ti.com.cn/video
微控制器 (MCU)	www.ti.com.cn/microcontrollers		
RFID 系统	www.ti.com.cn/rfidsys		
OMAP 机动性处理器	www.ti.com/omap		
无线连通性	www.ti.com.cn/wirelessconnectivity		
	德州仪器在线技术支持社区	www.deyisupport.com	

邮寄地址： 上海市浦东新区世纪大道 1568 号，中建大厦 32 楼 邮政编码： 200122