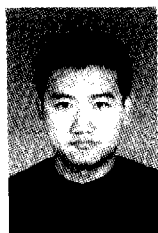


## TMS320F2812 芯片的 FLASH 烧写技术

孙 轶, 许少尉

(中国航空计算技术研究所, 陕西 西安 710068)



**摘 要:**介绍了数字信号处理器 TMS320F2812 芯片内部闪存(FLASH)的两种技术: CCS 插件烧写技术和 Flash281x\_API 库烧写技术。提出了一种利用算法、摆脱 JTAG 口, 使用 RS422 传输方式进行 FLASH 程序烧写的方法。由于 RS422 具有抑制共模干扰的能力, 通讯距离往往能够达到几千米, 所以这种方法非常适用于机载、弹载设备的外场加载和远程控制, 具有较大的实用价值。

**关键词:** F2812; RS422; FLASH programming; F281x\_API

**中图分类号:** TP342.3

**文献标识码:** A

**文章编号:** 1671-654X(2006)05-0046-04

## 引言

TMS320F2812 芯片是 32 位的控制专用数字信号处理器, 是 TMS320C2000™平台中的重要一员, 具有精度高、速度快、集成度高等特点, 作为目前控制领域最高性能的处理器之一, 它内含 128K × 16bit 闪存, 运算速度为 150MIPS。该处理器的大容量 FLASH 存储空间, 适用于低功耗、高性能的控制系统, 足以满足一般算法需求, 但是用户代码固化在 FLASH 中的运行速度要慢于直接装载到高速 RAM 存储器中。本文首先研究了两种常用的基于 JTAG 接口的 FLASH 烧写技术: CCS 插件烧写技术和 Flash281x\_API 库烧写技术, 然后提出了一种已在工程中实现, 仅通过一根串口电缆与 PC 机通讯即可实现的 FLASH 烧写方法。

## 1 TMS320F2812 片内 FLASH 结构介绍

TMS320F2812 芯片上有 128K × 16bit 嵌入式 FLASH 存储器和 1K × 16bit 一次可编程 EPROM (OTP) 存储器。FLASH 存储器由四个 8K × 16bit 扇区和 6 个 16K × 16bit 扇区组成, 各个扇区的地址空间见表 1, 通过这些扇区名或地址空间指定 Flash281x\_API 库中函数的操作参数, 可实现对 FLASH 具体扇区的擦除、编程或校验操作。

## 2 CCS2.0 生成目标代码

我们把调试完成后的程序生成用于烧写目的的二进制代码称为目标代码, 目标代码具备以下要素:

- 1) 在 .cmd 文件中声明 FLASH 映射空间, 见表 2;
- 2) 在 .cmd 文件中 “codestart” 指定程序启动地址,

“ .text ” 指定目标代码烧入 FLASH 的扇区位置。以 FLASH 直接启动为例, 通过

“ codestart : > BEGIN PAGE = 0 ” 和 PAGE0 段对 “ BEGIN : origin = 0x3F7FF6, length = 0x000002 ” 的声明配合完成启动地址的指定; 通过

“ .text : > FLASH PAGE = 0 ” 完成目标代码烧入 Sector A 的指定;

表 1 FLASH 扇区地址空间

| 地 址                     | 程序和数据空间              |
|-------------------------|----------------------|
| 0x3D 8000H ~ 0x3D 9FFFH | Sector J, 8K x 16    |
| 0x3D A000H ~ 0x3D BFFFH | Sector I, 8K x 16    |
| 0x3D C000H ~ 0x3D FFFFH | Sector H, 16K x 16   |
| 0x3E 0000H ~ 0x3E 3FFFH | Sector G, 16K x 16   |
| 0x3E 4000H ~ 0x3E 7FFFH | Sector F, 16K x 16   |
| 0x3E 8000H ~ 0x3E BFFFH | Sector E, 16K x 16   |
| 0x3E C000H ~ 0x3E FFFFH | Sector D, 16K x 16   |
| 0x3F 0000H ~ 0x3F 3FFFH | Sector C, 16K x 16   |
| 0x3F 4000H ~ 0x3F 5FFFH | Sector B, 8K x 16    |
| 0x3F 6000H ~ 0x3F 7FFFH | Sector A, 8K x 16    |
| 0x3F 7F80H ~ 0x3F 7FF5H | 当使用 CSM 后, 全部写为 0    |
| 0x3F 7FF6H ~ 0x3F 7FF7H | 放 Jump-to-Flash 入口地址 |
| 0x3F 7FF8H ~ 0x3F 7FFFH | FLASH 密码 (128-位)     |

3) 定义跳转指令 LB \_c\_int00 用于运行 c\_int00 函数, 可以单独编写一个 .asm 文件加入用户工程中, 内容如下:

```
.ref _c_int00
.sect "codestart"
LB _c_int00
.end
```

表 2 cmd 文件内重要内容

| 内 容  | 意 义  |
|--|--|
| OTP:origin = 0x3D7800H, length = 0x000800H     | 指定 FLASH 内资源<br>(OTP, PASSWORD 地址;<br>扇区 A ~ J 地址) |
| FLASHJ:origin = 0x3D8000H, length = 0x002000 H |  |
| FLASHI:origin = 0x3DA000H, length = 0x002000H  |  |
| FLASHH:origin = 0x3DC000H, length = 0x004000H  |  |
| FLASHG:origin = 0x3E0000H, length = 0x004000H  |  |
| FLASHF:origin = 0x3E4000H, length = 0x004000H  |  |
| FLASH E:origin = 0x3E8000H, length = 0x004000H |  |
| FLASHD:origin = 0x3EC000H, length = 0x004000H  |  |
| FLASHC:origin = 0x3F0000H, length = 0x004000H  |  |
| FLASHB:origin = 0x3F4000H, length = 0x002000H  |  |
| FLASHA:origin = 0x3F6000H, length = 0x001FF6H  |  |
| PASSWDS:origin = 0x3F7FF8H, length = 0x000008H | "Jump to Flash" 引导<br>重新复位后启动地址                    |
| BEGIN:origin = 0x3F7FF6H, length = 0x000002H   |  |
| RESET:origin = 0x3FFFC0H, length = 0x000002H   | 分配存储程序代码的扇区<br>(0x3D 8000H ~ 0x3D 9FFFH)           |
| .cinit: > FLASHJ PAGE = 0                      |  |
| .text: > FLASHJ PAGE = 0                       | 程序起始地址   |
| codestart: > BEGIN PAGE = 0                    |  |
| .const: > FLASHB PAGE = 1                      | 用于常量初始化的扇区<br>(0x3F 4000H ~ 0x3F 5FFFH)            |
| .econst: > FLASHB PAGE = 1                     |  |
| .reset: > RESET, PAGE = 0                      | 指定_c_int00 起始地址                                    |

4) c\_int00 函数调用的就是调试完成后的 C 程序代码。

综上所述,通过 CCS 2.0 平台,在用户工程中替换 .cmd 文件和增加一个 .asm 文件,即可生成可用的、后缀名为 .out 的目标代码。

### 3 两种常用烧写技术

TMS320F2812 芯片的 FLASH 烧写一般使用 CCS 2.0 集成编程环境,通过 JTAG 接口仿真系统将最终用户程序写入 FLASH。下面介绍基于 JTAG 接口方式下的两种烧写技术。

#### 3.1 使用 CCS 插件

TI 官方网站上可以免费下载 TMS320 DSP 系列相应 FLASH 烧写插件,插件烧写具有操作简单、兼容性好等特点。

插件安装成功后,在 CCS 编译器的 Tools 下拉菜单中会自动生成一个选项“F28xx On-Chip Flash Programmer”,只有当 CCS 编译器通过 JTAG 接口发现可用 DSP 资源后,点击该选项,才能进入烧写界面,操作界面和对应功能单元见图 1。

按照以下步骤可以完成对 TMS320F2812 片上 FLASH 的烧写操作:

##### a. 配置频率

要与用户实际硬件电路和初始化 DSP 设置值一致,OSCCCLK (MHz) 指定输入时钟频率值,PLLCLK value 指定 PLLCR 寄存器值,根据这两个值,插件自动计算系统时钟频率 SYSCCLKOUT 和配置烧写算法。

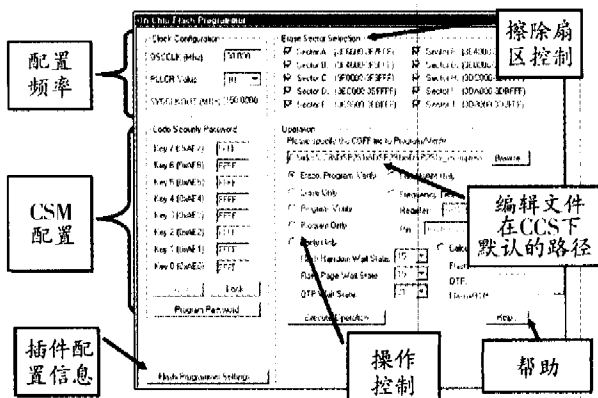


图 1 插件烧写界面和功能单元示意图

##### b. CSM 配置

操作 FLASH 前必须先点击 Unlock 按钮,将 128 位 CSM 全部写“1”完成解锁操作,否则,插件界面上所有操作菜单全部为虚。CSM 八组 Key 值可以更改,如果全部写“0”,那么 FLASH 将被锁死,所以必须谨慎使用。

##### c. 插件配置信息

插件配置最重要的功能是指定插件与 CCS 编译器的接口文件,这样才能保证插件正常工作,所以务必指定正确路径下的正确 API 接口文件。

#### d. 擦除扇区控制

TMS320F2812 片上 FLASH 分成 10 个相对独立的扇区,通过该插件上的复选框,可以任意选择需要擦除的扇区。

#### e. 编程文件在 CCS 下默认的路径

指出被烧写的 .out 文件所在路径,默认路径是当前工程中的 .out 文件存放路径。

需要注意:插件版本必须与 DSP 版本型号对应,否则无法完成烧写。关于插件版本与 DSP 版本型号对应关系不再赘述。

### 3.2 使用 Flash281x\_API 库

FLASH281x\_API 库烧写技术利用了 TI 公司提供的一种编程算法,该算法具有定义良好和操作简单等特点,算法中函数定义如下:

#### a. 指定擦除扇区

Uint16 Flash2812\_Erase(SectorMask, &FStatus)

1) SectorMask:指定哪个扇区将被擦除;

2) &FStatus:执行擦除操作后返回的状态值,能够判断操作是否成功。

#### b. 烧写程序到 FLASH

Uint16 Flash2812\_Program(&FlashAddr, &BuffAddr, Length, &FStatus)

1) &FlashAddr:程序从 FLASH 内哪个地址开始烧写;

2) &BuffAddr:即将准备烧写的程序当前存放空间的首地址;

3) Length:有多少个 16 位字需要烧写;

4) &FStatus:执行烧写操作后返回的状态值,能够判断操作是否成功。

#### c. 校验烧写到 FLASH 中的程序

Uint16 Flash2812\_Verify(&FlashAddr, &BuffAddr, Length, &FStatus)

1) &FlashAddr:从 FLASH 内哪个地址开始比较;

2) &BuffAddr:被比较文件的存储首地址;

3) Length:有多少个 16 位字需要比较;

4) &FStatus:执行烧写操作后返回的状态值,能够判断操作是否成功。

FLASH API 与用户工程建立关联后,目标代码通过调用 API 函数,实现对 FLASH 擦除、烧写和校验等操作,需要注意一点 CSM 中值不能全部写 0,否则 FLASH 将永久锁定,无法解锁。

## 4 PC 机串口烧写 FLASH 技术

前面提及的两种 FLASH 烧写技术必须利用 JTAG 接口,JTAG 接口易于操作,方便调试,但是受到空间和传输距离的限制,试想,如果一台装入复杂封闭环境的用户 DSP 系统的程序需要更新和升级,那么使用 JTAG 接口实现是相当困难的,接下来介绍的这种烧写方法能够解决这个难题。

### 4.1 技术介绍

通过串口烧写或加载程序的技术由来已久,并已在很多 DSP 系统中得到应用。虽然不同的 DSP 系统,可能有不同的实现方法,但是实现思路基本一致,即用“程序”烧写“程序”,TMS320F2812 系统也一样,前一个“程序”是指已固化程序,用于实现串口烧写的时机判断以及烧写的具体实施等,该程序中使用了 FLASH281x\_API 库编程算法(详见 3.2 章节);后一个“程序”才是用户的应用目标代码。

TMS320F2812 系统组成上,硬件方面使用 RS422 协议芯片、带有 RS232 串口的 PC 机以及 RS232/RS422 转换器。软件方面分为驻留在 PC 机的顶层软件和驻留在目标机的底层软件,顶层软件用于实现用户程序的解析等工作,底层软件用于实现将串口发来的数据烧写至 FLASH。

首先将底层软件通过 JTAG 口烧写至 FLASH(烧写方法详见 3.1 章节),之后就可以彻底甩开 JTAG 口。TMS320F2812 系统每次上电复位,先运行底层软件,软件程序判断是否需要重新烧写 FLASH,如果需要,则将串口发来的目标代码烧写至 TMS320F2812 片内 FLASH 指定扇区,如果不需要,则继续执行原有用户目标代码。

### 4.2 技术原理和关键点

该技术原理以及核心内容是把已解析的用户程序,用已固化的底层软件烧写至 FLASH。已固化的底层软件必须存储于 FLASH 内指定扇区,这个扇区不允许擦除和更改,完全独立于用户应用程序的存储区域。FLASH 内已固化底层软件 and 用户应用程序的区域划分可以通过 .cmd 文件解决。

图 2 是该技术的软件流程图,标号 1、2、3 是该技术的 3 个关键点。

关键点 1:使用 TMS320F2812 芯片默认引导方式“Jump to Flash”,即上电复位后,直接运行底层软件,用底层软件判断是否需要烧写 FLASH;

关键点 2:用户程序已被顶层软件解析,解析后的用户程序包括以下信息:烧入 FLASH 的首地址,程序长度和对应 16 进制数据;

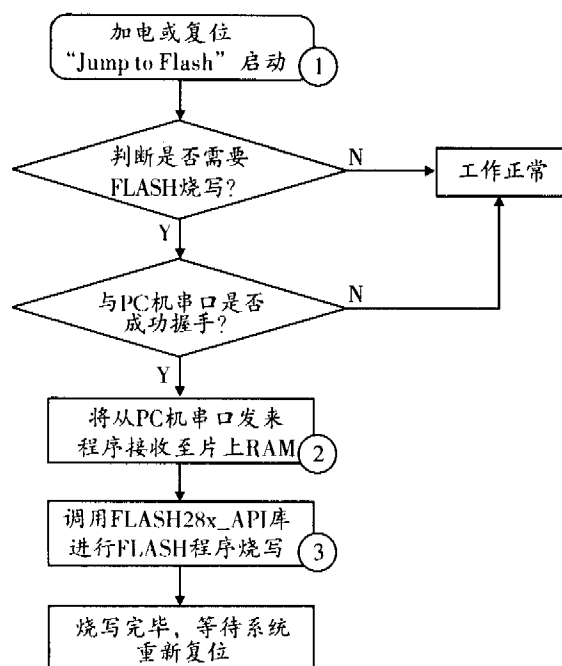


图2 软件流程图

关键点3:由于TMS320F2812片上FLASH不支持在其中一个扇区运行程序去擦除或烧写其它扇区这种做法,所以,完成接收数据和烧写FLASH工作的这部分程序必须搬移至片内RAM或片外RAM上运行。

#### 4.3 技术难点

PC机串口烧写技术的难点笔者归纳两点:

第一:用户程序的解析。因为常用烧写方法通过CCS集成编译环境来解析.out文件,解析过程对于我们来讲是不透明的,一旦脱离CCS,就必须重新构造解析算法。其实,通过分析用户程序的.map文件和读取FLASH内部由.map文件中规定的程序、数据、变量、向

量等有效段数据形成的.dat文件,就可以构造出简单可行的解析算法。

第二:烧写程序搬移至RAM。由于TMS320F2812片上FLASH扇区操作的特殊要求,用于烧写FLASH功能的程序段必须运行于FLASH之外,所以可以将该段程序以数组形式存放,烧写程序时先写入RAM,然后在RAM中运行,进而实现正确烧写FLASH。

## 5 结论

本文提及的几种烧写技术,均已得到工程验证。三种技术各有优缺点,实际项目开发中应根据阶段或环境选择合适的烧写方法。一般来讲,基于JTAG口的烧写技术应用于研发调试阶段,可以有效提高项目进度;串口烧写FLASH技术应用于定型阶段,可以有效降低维护成本,特别是对于机载、弹载设备的外场加载和远程控制,该技术非常实用。

#### 参考文献:

- [1] 苏奎峰,吕强,耿庆锋,等. TMS320F2812 原理与开发[M]. 北京:电子工业出版社,2005.
- [2] TI. TMS320F281x Data Sheet (Rev. 1) [EB/OL]. www.ti.com.cn,2005.
- [3] TI. TMS320F28x Boot ROM Reference Guide (Rev. A) [EB/OL]. www.ti.com.cn,2005.
- [4] TI. TMS320F28x Assembly Language Tools User's Guide [EB/OL]. www.ti.com.cn,2005.
- [5] TI. Flash281x\_API\_V100 [EB/OL]. www.ti.com.cn,2005.
- [6] TI. TMS320F281x Flash Programming API Revision Change [EB/OL]. www.ti.com.cn,2005.

## TMS320F2812 On\_Chip FLASH Programming Technique

SUN Yi, XU Shao-wei

(Aeronautics Computing Technique Research Institute, Xi'an 710068, China)

**Abstract:** In this paper, two methods for programming TMS320F2812 on-chip FLASH, which are based on CCS Plug-in and F281x Flash API respectively, are discussed. And, a new programming method through RS422 interface instead of JTAG is proposed. Because of the good performance of RS422 on constraining the common mode interference, long distance communication can be realized, even several kilometers far. So that this method is very suitable for the out-field load and remote control of on-board or on-missile devices, and has great value in practice.

**Key words:** F2812; RS422; FLASH programming; F281x\_API