

DOI:10.3969/j.issn.1001-4551.2014.03.029

# 基于 TMS320DM642 的 Flash 头文件烧写法的 DSP 二级 BootLoader 开发\*

张 宇<sup>1</sup>, 张勤俭<sup>1\*</sup>, 沈海阔<sup>1</sup>, 刘 伟<sup>2</sup>, 吴如焯<sup>2</sup>

(1. 北京交通大学 机械与电子控制工程学院, 北京 100044; 2. 北京宇航系统工程研究所, 北京 100076)

**摘要:**针对 DM642 掉电后驻留在内部存储器的数据和程序将全部丢失无法实现 DSP 系统脱机工作的问题,提出了一种利用 DSP 的 Boot 机制从外部 Flash 存储器自动加载程序的方法。详细介绍了 Flash 头文件烧写方法原理;针对 Am29LV033C Flash 芯片设计了 Flash 头文件烧写程序,详细说明了两次程序加载方法并给出 Flash 烧写过程;介绍了如何使用 CCS 保存数据,以生成头文件;简要介绍了 DSP 引导过程及二级 Bootloader 程序开发,设计了有项目针对性的二级 Bootloader 程序。最后,通过编写简单的测试程序验证 Flash 烧写方法及二级 Bootloader 引导程序。研究表明,Flash 头文件烧写法简单易懂,适合初学者学习,对不同型号 Flash 芯片操作只需简单修改指令字,可移植性强,易于实现 DSP 的二级 Bootloader 开发。

**关键词:**DM642;Flash;二级 Bootloader;自动加载

中图分类号:TP216+.11

文献标志码:A

文章编号:1001-4551(2014)03-0400-05

## DSP second-level Bootloader developing based on Flash header file burning of TMS320DM642

ZHANG Yu<sup>1</sup>, ZHANG Qin-jian<sup>1</sup>, SHEN Hai-kuo<sup>1</sup>, LIU Wei<sup>2</sup>, WU Ru-xuan<sup>2</sup>

(1. School of Mechanical Electronic and Control Engineering, Beijing Jiaotong University, Beijing 100044, China;

2. Beijing Institute of Aerospace System Engineering, Beijing 100076, China)

**Abstract:** Aiming at the problems of the data and programs in the memory of DM642 always get lost completely once the power is down so DSP system can't work without a computer, one method to load programs automatically from exterior Flash storage by using Boot mechanism of DSP was presented. The theory of header file program ming by Flash was introduced in detail. A burning method of Flash header file for Am29LV033C Flash chip was achieved. The method of twice loading of programs was explained and the process of Flash programming was presented. The way of using CCS to save data so as to generate header files was described. The guiding process of DSP and developing process of the second-level bootloader is briefly introduced here. A second-level Bootloader program specific to the project was designed. A simple testing program was developed to validate the Flash burning method and second-level Bootloader guiding program in the end. The research results indicate that the programming method for Flash header file is very suitable for the beginners for it is easily understood. The method also has excellent adaptability in that only a small part of instructions needs to be modified to fit the operations of different types of Flash chips, as a result the developing of second-level Bootloader becomes easier.

**Key words:** DM642; Flash; second-level Bootloader; auto loading

收稿日期:2013-10-09

基金项目:国家自然科学基金资助项目(50975031);中央高校基本科研业务费专项资金资助项目(M11JB00420)

作者简介:张 宇(1986-),男,河北廊坊人,主要从事信号采集及处理和自动化控制方面的研究. E-mail:11121413@bjtu.edu.cn

通信联系人:张勤俭,男,博士,教授. E-mail:qjzhang@bjtu.edu.cn

## 0 引言

TMS320DM642 是 TI 公司 C6000 系列 DSP 中最新的定点 DSP,采用第二代高性能、先进的超长指令字 *veloci T1.2* 结构的 DSP 核及增强的并行机制,具有极强的处理性能、高度的灵活性和可编程性<sup>[1]</sup>。该芯片内部不带 Flash 或 EEPROM,系统掉电后,驻留在存储器内部的数据和程序将全部丢失。因此,在利用 DM642 做 DSP 系统开发时,若使 DSP 系统能够脱机工作,就必须设计 DSP 的自启动引导装载,这也是 DSP 系统开发非常关键的环节之一。为实现 DM642 的自启动,可通过其 EMIFA 接口外部扩展 Flash 存储器<sup>[2]</sup>,将程序固化在 Flash 存储器中,然后通过 DSP 的片上引导功能,从 DSP 片外加载用户应用程序<sup>[3]</sup>。该方法的关键在于 Flash 存储器中程序的正确烧写和正确的用户引导程序(即二级 Bootloader 程序)开发。

本研究针对 TMS320DM642 DSP 和 AM29LV033C Flash 详细介绍一种 Flash 直接烧写的头文件烧写方法,设计出一种 DSP 自启动方案,通过简单测试程序,验证方案的正确性和可行性。

## 1 Flash 头文件烧写原理

Flash 头文件烧写法是一种需要两次加载程序的方法。首先加载应用程序,加载后在 DSP 内存空间将生成应用程序的反汇编代码。通过 .map 文件,可确定应用程序各段数据所对应的空间,再利用 CCS 的 *data-save* 功能导出适当长度的应用程序,导出的应用程序被保存为 .dat 格式文件。由于 .dat 文件不能直接用于 Flash 烧写,可以把 .dat 文件的所有数据存放到一个数组中,通过 Flash 烧写这个数组是可行的。为了方便起见,可将该数组以头文件的形式加载到 Flash 烧写工程中。然后,将含有该头文件的 Flash 烧写程序加载到 DSP 的内存 RAM 中,通过运行烧写程序即可将头文件中的数组烧写到指定的 Flash 存储器内存中。

该方法需要两次加载程序,所以要将两次加载的程序存放到 DM642 内存的不同地址,以免第二次加载程序时将第一次加载的程序覆盖。TMS320DM642 内部 RAM 地址空间为  $0x0 - 0x3FFFF$ ,共 256 KB,而 Flash 烧写程序仅为 12 KB。因此,对于 DM642 而言,用户应用程序的有效内存可以达到 230 KB,足以满足大多数用户开发。通常应将应用程序加载到内存起始地址,而将烧写程序加载到内存的最后位置。

## 2 头文件法 Flash 烧写程序设计

对 Flash 存储器进行烧写一般有以下几种方法:通过编程器烧写;通过开发商提供的专门烧写软件工具烧写;用户自己编写 Flash 烧写程序直接烧写。

通过编程器烧写程序时,必须在存储芯片焊接到电路板之前将程序烧写到芯片内部,然后再将存储芯片焊接。该方法的缺点是当需要更新程序或本次下载程序出现问题时,必须将芯片取下,非常不便,在取芯片过程时,容易使焊盘脱落,导致板卡无法使用,风险非常大,且高温取片有损芯片及板卡性能。

TI 公司提供了一款专门的 Flash 烧写工具—Flashburn 软件,该软件组合了 Flashburn 和 FBTC 两个程序。其中,Flashburn 程序是可执行文件,提供用户操作界面,能够把 FBTC 程序下载到 DSP 目标板中<sup>[4]</sup>。使用该方法的关键在于编写 FBTC 工程以生成 FBTC.out 文件,对于不同型号的 DSP 芯片和 Flash 存储器,每次都需要修改许多参数和函数,以满足不同目标板的烧写需求,通用性非常差。同时,很多 DSP 开发板的例程只提供了专门用于该板卡的 FBTC.out 文件,并未给出 FBTC 工程源码,因此,当用户想要开发自己的 DSP 系统时,开发 FBTC 工程是非常复杂的。

通过对上述几种方法的分析可知,编程器烧写和 Flashburn 软件烧写,操作复杂,通用性差。而用户自己编写 Flash 烧写程序直接烧写,方法简单,操作灵活,通用性强。针对 AMD 公司的 Am29LV033C Flash 芯片,本研究设计出一种程序代码量非常小的 Flash 头文件烧写法。该方法是一种简便、快捷、通用的 Flash 直接烧写方法,利于初学者使用。

对 Flash 进行烧写操作时,首先需擦除要写入的扇区,然后才可进行烧写。擦除和烧写操作是通过往指定地址写指令字来完成的。在进行 Flash 的擦除和烧写过程中,系统循环检测 Toggle 位是否发生翻转来判定操作是否完成<sup>[5]</sup>。而 Flash 读操作就是简单地从指定的地址读取相应数据即可。相关指令字和 Toggle 位算法参考芯片资料。

Flash 烧写程序的核心代码如下:

```
#include "fangbo3.h" //应用程序生成的.h头文件
void main()
{
    CHIP_Erase();
    FlashBurn ((UINT32 *) FLASHBASE, fangbo3,
1024);
    return;
}
```

```

void FlashBurn ( UINT32 * dest, UINT32 * src, UINT32 length)
// length 为数组元素个数
{
    UINT8 * p = (UINT8 *) dest;
    UINT8 * q = (UINT8 *) src;
    UINT32 LEN = length * 4;
    Write_1Byte ( p, q, LEN);
}

void Write_1Byte( UINT8 * dest, UINT8 * src, UINT32 length)
{
    UINT32 i;
    for(i = 0; i < length; i + +)
    {
        * (unsigned char *) FLASHBASE = 0xAA;
        * (unsigned char *) FLASHBASE = 0x55;
        * (unsigned char *) FLASHBASE = 0xA0;
        * dest = * src;
        Toggle_Check ();
        dest + + ;src + + ;
    }
}

void CHIP_Erease()
{
    * (unsigned char *) FLASHBASE = 0xAA;
    * (unsigned char *) FLASHBASE = 0x55;
    * (unsigned char *) FLASHBASE = 0x80;
    * (unsigned char *) FLASHBASE = 0xAA;
    * (unsigned char *) FLASHBASE = 0x55;
    * (unsigned char *) FLASHBASE = 0x10;
    Toggle_Check ();
}

```

分析程序可知,将.h 头文件中的数组名和数组长度为实参传到 FlashBurn() 函数,即可将存有应用程序信息的数组烧写到 Flash 存储器中。

Flash 烧写及程序加载全过程如图 1 所示。

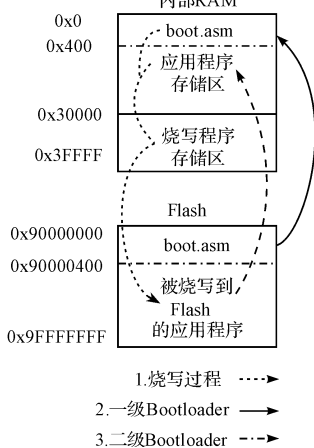


图 1 Flash 烧写及程序加载过程图

图 1 中,0x0 - 0x400 存储区存储汇编语言编写的二级 Bootloader 程序,之后为用户编写的 C 语言的应用程序。为了防止程序覆盖,Flash 烧写程序被存放在 0x30000 ~ 0x3FFFF 存储空间。笔者运行 Flash 烧写程序,将二级 Bootloader 程序和应用程序烧写至 Flash 存储器,如图 1 中箭头 1 所示。系统再次上电时,一级 Bootloader 程序搬运 Flash 起始的 1 KB 程序,此程序恰好为二级 Bootloader 程序,如图 1 中箭头 2 所示。之后 DSP 系统从 0x0 地址运行,即运行二级 Bootloader 程序,二级 Bootloader 程序指引 DSP 将 Flash 中的应用程序搬运至内部 RAM,如图 1 中箭头 3 所示。

### 3 头文件生成方法介绍

通过 CCS 平台中的 data-save 功能,可将目标程序的代码段和数据段所对应的地址区的内容导出,生成 .dat 文件。

CCS 平台的 data-save 功能在 File 菜单下,用户选择 Data 然后点击 Save... 后会弹出一个提示保存 .dat 文件的对话框<sup>[6]</sup>,选择保存路径并填写文件名,点击保存,会再弹出一个对话框,头文件生成对话框如图 2 所示。

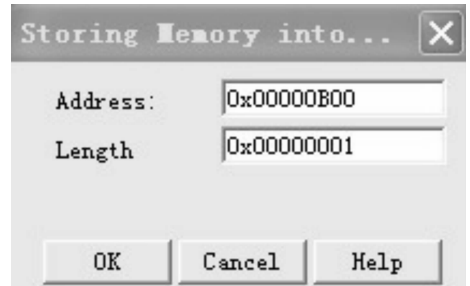


图 2 头文件生成对话框

图 2 中, Address 为要保存数据的起始地址, Length 为要保存数据的长度<sup>[6]</sup>。两个数据的大小需要根据应用工程的 .cmd 文件和 .map 文件确定。通常,应用程序存储在 RAM 内存的起始地址,因此 Address 值一般为 0。通过 .map 文件得到的程序大小是以字节为单位,而 .dat 文件的数据是以双字为单位存储的,因此, Length 的大小为 .map 文件中程序大小的 4 倍。

由于 .dat 文件不能直接用于 Flash 烧写,需要把 .dat 文件的所有数据存放到一个数组中,再将这个数组将以头文件的形式添加到 Flash 烧写程序中。从 .dat 文件到数组.h 文件可以通过下载头文件生成器自动生成。也可以使用 UEditor 软件的列模式手动编辑数组,再保存为 .h 格式的头文件。

## 4 用户引导程序设计

DSP程序分为系统引导程序和用户目标程序两部分。第一部分为系统引导程序,负责系统的初始化、引导加载用户代码及中断处理;第二部分为用户目标程序,通过系统引导程序加载<sup>[7]</sup>。因此,TMS320DM642程序加载过程分为两个步骤。第1步:在系统复位时,EDMA控制器将地址0x90000000开始的1KB程序装入DM642内部RAM的0x0-0x400地址处,并释放CPU,CPU从0x0地址处开始执行程序。通常这1KB程序为引导程序;第2步:执行引导程序,把开发人员的程序从Flash存储器搬运到内部RAM,进入\_c\_int00入口,完成加载过程<sup>[8]</sup>。

其中,\_c\_int00是C语言程序的入口,C语言程序运行前,必须首先初始化C运行环境。因此,用户开发的Bootloader程序必须使用汇编语言编写。具体实现工作包括:

- (1)配置EMIFA接口寄存器和Flash参数;
- (2)初始化栈指针;
- (3)将Flash存储器中1KB以后的程序和数据拷贝到指定的DSP内部RAM物理地址;
- (4)跳转至C程序的入口地址\_c\_int00处<sup>[9]</sup>。

不同型号的DSP和Flash会有不同的寄存器和Flash参数配置,因此本研究仅给出通用的拷贝程序和跳转程序,程序关键代码如下<sup>[10-11]</sup>:

```
_boot_loop:
    ldw    *A4 + +, B0;DSP开始读取Flash中的数据
    nop 5
    stw    B0, *B4 + +
    mvkl   FLASH_SIZE, B5
    mvkh   FLASH_SIZE, B5
    add    1, A1, A1
    cmlt   A1, B5, B1 ;若A1 < B5,则B1 = 1
    nop
```

[B1] B\_boot\_loop ;根据B1判断是否复制完毕,B1为1时执行该条语句,B1为0时不执行

```
    nop 5
    mvkl   .S2_c_int00, B0 ;跳至C程序入口
    mvkh   .S2_c_int00, B0
    nop 5
```

通常将用户引导程序命名为boot.asm的汇编文

件。用户编写的应用程序要想实现自启动,必须包含该引导程序,而且必须存储在Flash的起始1KB地址处,这样在系统复位时,一级Bootloader自动加载的恰好是二级Bootloader程序,然后系统执行二级Bootloader程序将Flash中的应用程序加载到内部RAM,从而实现用户应用程序的全部加载。

## 5 方案验证

为验证本研究提出的方案的正确性,笔者编写了一个简单测试程序。程序关键代码如下:

```
void main()
{
    while(1)
    {
        *(UINT8 *)0x90000000 = 0xFF;
        delay(1); //延时600ns
        *(UINT8 *)0x90000000 = 0x00;
        delay(1);
    }
}
```

该测试程序的功能是向Flash的起始地址处间隔写入0和1,因为没有写Flash烧写的指令字,数据只是出现在数据线上而不会真正写入Flash存储器。笔者将测试程序烧写到Flash后,系统掉电并去掉仿真器,进行手动复位,然后使用示波器测试Flash数据线引脚,得到的验证结果如图3所示。

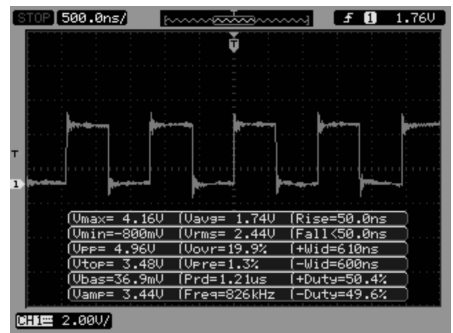


图3 示波器检测图

图3显示结果表明,本研究得到预期的方波信号,验证了该方案的正确性。该测试程序虽然简单,但通用性强,在板卡没有指示灯的情况下仍能直观地验证用户编写的Flash烧写程序和二级Bootloader引导程序是否正确。

(下转第408页)

### 本文引用格式:

张宇,张勤俭,沈海阔,等.基于TMS320DM642的Flash头文件烧写法的DSP二级BootLoader开发[J].机电工程,2014,31(3):400-403,408.

ZHANG Yu, ZHANG Qin-jian, SHEN Hai-kuo, et al. DSP second-level Bootloader developing based on Flash header file burning of TMS320DM642[J]. Journal of Mechanical & Electrical Engineering, 2014,31(3):400-403,408.

《机电工程》杂志: <http://www.meem.com.cn>

(5):204-209.

[3] 窦晓波,胡敏强,吴在军,等. 数字化变电站通信网络性能仿真分析[J]. 电网技术,2008,32(17):98-104.

[4] 童晓阳,廖晨淞,周立龙,等. 基于 IEC61850-9-2 的变电站通信网络仿真[J]. 电力系统自动化,2010,34(2):69-74.

[5] 高 嵩. OPNET Modeler 仿真建模大解密[M]. 北京:电子工业出版社,2010.

[6] 杨光敏,蔡光卉,常 俊,等. 基于 OPNET 的智能配网 AMR 业务建模及仿真[J]. 电力系统通信,2012,33(240):82-87.

[7] Riverbed Technology. OPNET Modeler-OPNET Technologies [EB/OL]. [2012-12-29]. <http://www.opnet.com>.

[8] 鲁宗相,王彩霞,闵 勇,等. 微电网研究综述[J]. 电力系统自动化,2007,31(19):100-107.

[9] LASSETTER R, AKHIL A, MAMAY C, et al. The CETRIS Micro-grid Concept [EB/OL]. [2008-10-08]. <http://certs.tbl.gov/pdf/50829.pdf>.

[10] 王成山,杨占刚,王守相. 微网实验系统结构特征及控制模式分析[J]. 电力系统自动化,2010,34(1):99-105.

[11] 袁新喜,谈志远,陶维青. IEC61850 标准在微电网中的应用探讨[J]. 电测与仪表,2012,49(7):49-53.

[12] ALVAREZ A R, SUBIRACHS A C. Design, management and comissioning of a utility connected microgrid based on IEC 61850[J]. **IEEE ISGT Europe**,2010(1):1-7.

[13] YOO B K, YANG S H. Communication architecture of the IEC61850-based micro grid system[J]. **Journal of Electrical Engineering & Technology**,2011,6(5):605-612.

[14] 邓 卫,裴 玮,齐智平. 基于 IEC61850 标准的微电网信息交互[J]. 电力系统自动化,2012,36(7):1-6.

[15] 韩小涛,聂一雄,尹项根. 基于 OPNET 的变电站二次回路通信系统仿真研究[J]. 电网技术,2005,29(6):67-71.

[16] IEC. IEC61850 Communication networks and systems in substations[S]. IEC,2003.

[编辑:李 辉]

本文引用格式:

韦幸幸,张有兵,谢路耀,等. OPNET 在微网通信系统仿真中的应用[J]. 机电工程,2014,31(3):404-408.

WEI Xing-xing, ZHANG You-bing, XIE Lu-yao, et al. Application of OPNET in microgrid communication system simulation[J]. Journal of Mechanical & Electrical Engineering, 2014,31(3):404-408. 《机电工程》杂志:<http://www.meem.com.cn>

(上接第 403 页)

## 6 结束语

本研究结合实际工程项目,详细介绍了 Flash 头文件烧写法,其中包括 Flash 擦除程序、烧写程序以及头文件的生成方法,该方法简单易懂,且程序代码量非常小,对不同型号的 Flash 芯片只需修改指令字,具有很强的通用性。通过对 TMS320DM642 的程序加载机制和加载方法研究,成功编写了具有项目针对性的二级 Bootloader 程序。最后,通过简单的方波测试程序进行了方案验证。研究结果表明,Flash 头文件烧写法和二级 Bootloader 程序具有正确性和可行性。

### 参考文献 (References):

[1] 许永辉,杨京礼,林连雷. TMS320DM642 DSP 原理与应用实践[M]. 北京:人民邮电出版社,2009.

[2] 韦金辰,李 刚,王臣业. TMS320C6000 系列 DSP 原理与应用系统设计[M]. 北京:机械工业出版社,2012.

[3] Texas Instruments. TMS320C6000 EMIF to External Flash Memory (SPRA568A)[Z]. Texas Instruments,2002.

[4] 王跃宗,刘京会. TMS320DM642 DSP 应用系统设计与开发[M]. 北京:人民邮电出版社,2009.

[5] Adanced Micro Devices. Am29LV033C Datasheet[Z]. Advanced Micro Devices,2003.

[6] 李方慧,王 飞. TMS320C6000 系列 DSPs 原理与应用[M]. 2 版. 北京:电子工业出版社. 2005.

[7] 曾昭健,毛 韬,朱善安,等. 串行引导加载技术在 DSP 网络实验系统中的应用[J]. 机电工程,2007,24(8):17-19.

[8] 韩 飞,胡春海,李 伟. TMS320C6000 系列 DSP 开发应用技巧[M]. 北京:中国电力出版社,2008.

[9] Texas Instruments. TMS320C6000 DSP/BIOS Users Guide (SPRU423E)[Z]. Texas Instruments,2004.

[10] Texas Instruments. Creating a Second-Level Bootloader for FLASH Bootloading on TMS320C6000 Platform With Code Composer Studio (SPRA999A1)[Z]. Texas Instruments, 2006.

[11] 田黎育,何佩琨,朱梦宇. TMS320C6000 系列 DSP 编程工具与指南[M]. 北京:清华大学出版社,2006.

[编辑:李 辉]