TMS320VC5509 在线烧写 Flash 并自举启动方法研究

陈若珠，胡金平，李战明

（兰州理工大学 电气工程与信息工程学院，甘肃 兰州 730050）

摘 要：为了解决 TMS320VC55X 系列 DSP 系统程序代码的保存问题，设计了一种利用 JTAG 接口，在线烧写 Flash 并实现自举启动的方法。这种在线编程的方法利用并行外部存储器加载(EMIF)接口将 TMS320VC5509 和 Flash 芯片相连接，通过搬移程序将应用程序的已初始化段按照 C55X 系列 DSP 引导表格式烧写进外部扩展的 Flash 存储器中，从而实现自举启动。该方法为 DSP 系统的软件维护和升级带来了方便，具有实际的应用价值。

关键词：TMS320VC5509；自举启动；在线编程；Flash

中图分类号：TN911.72 **文献标识码**：A **文章编号**：1674-7720(2011)02-0009-03

Research of online programming external Flash and bootloading based on TMS320VC5509

Chen Ruozhu, Hu Jinping, Li Zhanming

(College of Electrical and Information Engineering, Lanzhou University of Technology, Lanzhou 730050, China)

Abstract: In order to preserve the application code of the C55X DSP, a method of online programming external flash and bootloading using JTAG based on TMS320VC5509 chip is designed. This method connects the DSP and Flash through the EMIF, moves the initialized section of the application code to external flash based on the format of the C55X DSP Bootloader using the flitting program. This method is convenient for user to maintain and upgrade the DSP spftware, and could be extensively applied to other DSP systems.

Key words: TMS320VC5509; bootloading; online programming; Flash

随着数字信号处理技术的快速发展，DSP 被广泛应用于各种数字信号处理系统中。最终开发的系统若要脱离仿真器运行，必须将程序代码存储在非易失性存储器中。Flash 是一种可在线进行电擦写而掉电后信息又不丢失的存储器，它具有功耗低、容量大、擦写速度快等特点。如何将程序烧写进 Flash，并在上电时加载到 DSP 内部的 RAM 中，是 Flash 在 DSP 系统应用中的两个基本问题[1]。本文基于 TI 公司的 TMS320VC5509A 和 AMD 公司的 AM29LV800 开发系统，详细阐述了在线烧写 Flash 并实现自举启动的方法。

1 硬件电路设计

图 1 为 TMS320VC5509A 与 AM29LV800 的连接示意图[2]，Flash 扩展在 CE1 空间，起始地址为 200000。由于 TMS320VC5509A 只有 14 根地址线 A0~A13，又因为 Flash 作为数据存储空间使用时的地址编码采用字寻址方式，则 DSP 的 A0 信号无效，所以 AM29LV800 芯片的低 13 位地址线 A0~A12 连接 TMS320VC5509A 的地址线 A1~A13，高 6 位地址线 A13~A18 由缓冲串口来扩展。AM29LV800 是低功耗 Flash，工作在 2.7 V~3.6 V 电压下，一般存储数据可以保存 100 年以上，可以重复编程次数高达 10 万次。A18~A0 为外部地址管脚，DQ0~DQ15 为 16 条数据线，$\overline{CE}$ 为片选控制管脚，$\overline{OE}$ 为输出控制管脚，$\overline{WE}$ 为写入控制管脚。

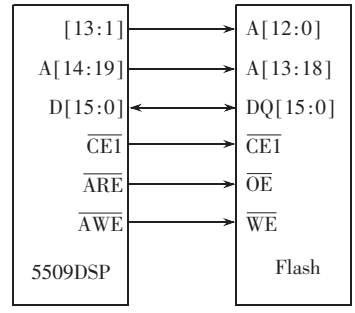

图 1 TMS320VC5509 与 AM29LV800 连接图

2 自启动过程分析及启动表结构

DSP 系统的 bootloader 是指在系统上电时将一段存

储在外部非易失性存储器中的程序搬移到 DSP 片内或片外扩展的高速 RAM 中并执行的代码。Bootloader 程序永久性地存储在 DSP 以 FF8000H 开始的 ROM 中，DSP 系统在复位后 PC=FF8000H，即从 Bootloader 程序首地址开始执行。

TMS320VC5509 DSP 的 Bootloader 有多种加载方式[3]，如表 1 所示，设置 DSP 的 GPIO0-GPIO3，DSP 在复位时读取这 4 个引脚上的状态以确定所使用的启动模式。本文使用 16-bit EMIF 加载方式，虽然连线复杂，需要考虑并行非易失存储器 Flash 与 EMIF 接口的匹配关系，但是它的优点很多：不需要外部时钟驱动，非易失存储器种类多样，容量较大，除了存储下载表之外，还可存储系统需要保存的关键数据，以便在掉电时保存信息。

表 1 TMS320VC5509A 的加载方式

BOOT[3:0]/GPIO[3:0]	引导方式
0001	Mcbsp0 的串行 EEPROM 引导
0010	USB 接口引导
0101	EHPI(多元引导)方式
0110	EHPI(非多元引导)方式
1000	来自外部 16-bit 异步内存引导
1001	Mcbsp0 的串行 EEPROM 引导
1011	并行 EMIF 引导方式
1110	Mcbsp0 的同步串行引导
1111	Mcbsp0 的同步串行引导

在这些加载模式下，下载程序之前先要生成一张载入表，即引导表。引导表的结构如图 2 所示，引导表携带的信息有代码段和数据段信息，向 DSP 下载程序的入口点地址、寄存器配置信息和可编程延时信息。

字节地址+0	字节地址+1	字节地址+2	字节地址+3
入口点地址(32 bit)			
需配置寄存器数(32 bit)			
寄存器地址		寄存器值	
延迟指针		延迟计数	
段字节数(32 bit)			
段起始地址(32 bit)			
数据	数据	数据	数据
数据	数据	数据	数据
32 bit 0(引导表结束)			

图 2 引导表结构

读引导表可知以下信息：程序入口地址是引导表加载结束后用户程序开始执行的地址，也就是用户程序生成的 map 文件中显示的入口地址；需配置寄存器数表明后面有多少个需要配置的寄存器；当延时标志为 0xFFFF 时，执行延时，延时长度决定了在寄存器配置后延时多少个 CPU 周期才进行下一个动作；段字节数、段起始地址和数据表示用户程序中定义的各个段的内容；引导表以 32 个 0 为结束标志。

生成引导表的方法：通过在 DOS 环境下使用 hex55.exe 转换工具。在转换操作之前，先把用户程序生成的 .out 文件、包含转换选项的 CMD 文件 hex5509.cmd 和转换工具 hex55.exe 放在同一个文件夹里，在 DOS 方式下先将路径修改为文件所在的位置，然后在此路径下运行命令 hex55 hex5509.cmd，即可生成想要的.hex 文件。

在转换时，提供引导表的相关配置信息的 CMD 文件这里被命名为 hex5509.cmd，文中用到的 hex5509.cmd 的内容为：

```
-boot                    /* 创建一个引导表 */
-v5510:2                 /* 选择合适的 DSP 引导表格式 */
LED.out                  /* 输入文件 */
-o FLASH.hex             /* 输出文件 */
-a                       /* 输出格式为 straight ASCII*/
-parallel16              /*16 位并行异步加载模式 */
-memwidth 16
-romwidth 16
-e 0x000004d4            /* 程序的入口地址 */
-map LED.map             /* 输出 map 文件 */
-delay 0xffff            /* 延时 */
```

3 Flash 烧写

Flash 的读操作与传统 EPROM 读操作相同。由于芯片使用软件保护模式进行操作，用户编程时，只要向指定的地址写入指定的序列，就可以启动 Flash 芯片内部的写状态机，完成指定的操作。表 2 为 Flash 的操作命令说明(对芯片的擦除和编程都是按照字进行的)，表中所有的数据都是十六进制数。

Flash 的正确操作顺序：先复位，再擦除，最后编程。按照表 2 提供的操作命令时序来实现对 AM29LV800 的擦除和编程，PA 为编程地址，PD 为编程数据。Flash 扩展在 CE1 空间，起始地址是 200000，所以操作时所有地址必须加上 200000。例如烧写工程中擦除部分命令为：

表 2 AM29LV800B 的操作命令说明

命令序列 步骤		复位	擦除	编程
第 1 步	地址	XXX	555	555
	数据	F0	AA	AA
第 2 步	地址		2AA	2AA
	数据		55	55
第 3 步	地址		555	555
	数据		80	A0
第 4 步	地址		555	PA
	数据		AA	PD
第 5 步	地址		2AA	
	数据		55	
第 6 步	地址		555	
	数据		10	

```
deminaddr = (int *)CESECT1;
addbias = 0x0555;
*(deminaddr+addbias) = 0x00aa;
addbias = 0x02aa;
*(deminaddr+addbias) = 0x0055;
addbias = 0x0555;
*(deminaddr+addbias) = 0x0080;
addbias = 0x0555;
*(deminaddr+addbias) = 0x00aa;
addbias = 0x02aa;
*(deminaddr+addbias) = 0x0055;
addbias = 0x0555;
*(deminaddr+addbias) = 0x0010;
delay(100)
```

芯片擦除需要占用 6 个总线周期，而芯片编程需要 4 个总线周期，依照表 3 的数据，在每个总线周期对相应地址写入命令字就可以了。用户一般都是对芯片进行写操作，写操作只能使‘1’变‘0’，而擦除只能使‘0’变为‘1’。图 3 为擦除和编程命令波形图，清楚地显示了擦除和编程操作过程。

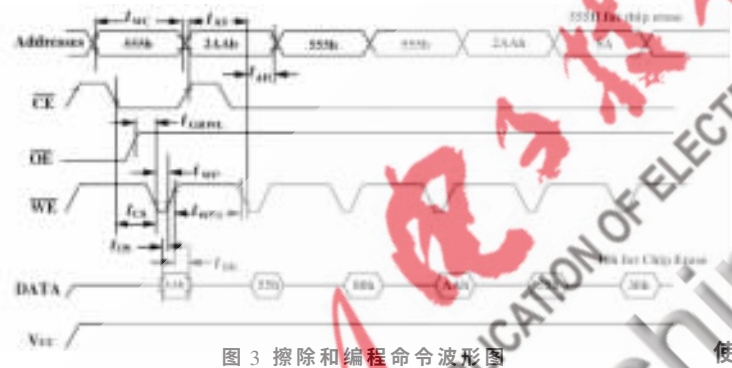

图 3 擦除和编程命令波形图

判断编程或擦除的结束是当把编程或擦除的命令字按照其时序写入 Flash 时，在写编程命令时序或擦除命令时序的最后一个 WE 上升沿到来之后，AM29LV800 会自动运行一个嵌入在 Flash 内部的算法来判断编程或擦除操作是否结束。采用触发位校验的方法，检测数据切换位 DQ6(Toggle Bit)的状态，连续读数据会使 DQ6 的值在‘0’和‘1’之间来回切换，当编程或擦除结束时，DQ6 就停止值的切换[4]。因此，可以通过连续两次读 DQ6 的值来判断编程或擦除是否结束，当两次读得的值相同时，说明编程或擦除结束，否则没有。触发位检测算法流程图如图 4 所示。

4 程序的烧写实现

本系统在 CCS 仿真环境下对 Flash 进行在线编程。先建立一个 Flash 的烧写工程，并在工程中将要烧写进 Flash 的引导表文件通过 CCS 的 LOAD DATA 功能直接加载进 DSP 的内存，根据加载的首地址和数据长度，在仿真环境下烧写进 Flash 中。值得注意的是，程序加载的

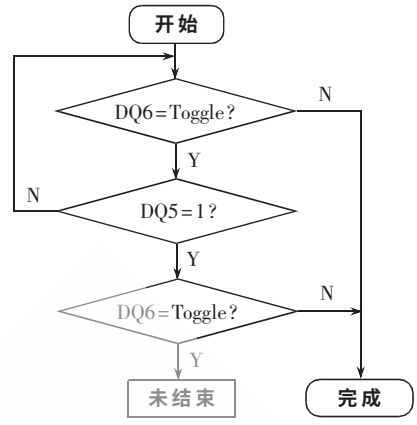

图 4 触发位检测算法流程图

内存空间不能与 Flash 的烧写程序重叠，否则烧写失败。烧写完成以后，关掉电源，拔掉仿真器电缆，让仿真器和计算机脱开；重新打开电源，实验板上指示灯闪烁，表明烧写进 Flash 程序正在运行，自启动成功。

需要补充的是，经过 hex55.exe 文件转化后的 hex 文件的引导表文件不能直接导入 CCS 中，CCS 只支持将特别规定的 DAT 格式文件通过 LOAD DATA 导入内存，所以在导入之前必须先将引导表转化成 DAT 格式文件，这个工作可以由 VC 编写一个简单的 C 语言转化程序实现。

本文阐述了一种针对 TMS320VC5509A DSP 简单有效的 Flash 烧写方法，并提出了程序自举引导的实现方法。可以有效地解决程序代码存储问题和 DSP 脱机自举问题，不仅提高了调试效率，也增加了系统的灵活性。本文讨论的引导方法包括硬件设计及相关程序，已经在笔者的实际开发语音项目中使用并成功运行。

参考文献

[1] 薛金辉，王忠勇. TMS320VC5501 在线烧写 Flash 并自举启动方法研究[J]. 通讯技术，2009，42(09)：193－195.

[2] Texas Instruments. TMS320VC5509 DSP external memory interface (EMIF) reference guide[S]. Literature Number: SPRU670，2003(10).

[3] Texas Instruments. Using the TMS320VC5509A bootloader [S]. SPRA375F，2008(9).

[4] Advanced Micro Devices. am29lv800b [R].AMD Technical specification，publication #21490，2000(8).

(收稿日期：2010－09－13)

作者简介：

陈若珠，女，1963 年生，高级工程师，主要研究方向：语音识别、嵌入式。

胡金平，男，1985 年生，研究生，主要研究方向：语音识别、嵌入式研究。