

TMS320C672x DSP 引导加载系统的设计与实现

刘 芳, 臧 威

(中国航空工业飞行自动控制研究所 软件工程中心, 陕西 西安 710065)

摘 要: 基于 DSP 的 Flash 引导启动技术已被广泛运用. 传统方式采用 JTAG 仿真器完成应用程序的调试和代码的烧写, 但受航空工程应用环境的制约, 突显操作不便、效率低下的问题. 对此提出了一种全新加载方案, 基于通用性更强的串口传输模式, 通过固化的监控程序和可操作的应用界面, 实现加载过程的跟踪和控制. 工程试验结果表明本方案具备满意的可操作性、高效性和可靠性.

关键词: 数字信号处理器; Flash; 引导加载程序; TMS320C6727

中图分类号: TP316

文献标识码: A

文章编号: 1000-7180(2013)10-0046-04

The Design and Implementation of Bootloader System Based on TMS320C672x DSP

LIU Fang, ZANG Wei

(Software Engineering Center, Flight Automatic Control Research Institute,
Aviation Industry Corporation of China, Xi'an 710065, China)

Abstract: The flash bootloading technique for DSPs has been widely used. In conventional methods, JTAG simulators are applied to the debugging and burning of programs, which, however, proves inefficient and inconvenient in the aviation engineering settings. Based on the serial bus transmission mode, a new scheme is proposed in this paper, realizing the tracing and controlling of the entire loading process through the embedded monitoring program and the user's interface. Engineering tests have lent sufficient evidences to its satisfactory operability, efficiency, and reliability.

Key words: Digital Signal Processor(DSP); Flash; Bootloader; TMS320C6727

1 引言

在 DSP 系列的系统设计中, 引导程序设计是其中的难点之一, 系统启动引导的成功与否决定了应用程序的运行环境是否能正确构建, 即系统启动成功是正确应用运行的前提^[1]. 目前, 大多数数字信号处理器不带内部可擦写存储器, 程序一般在系统上电时, 从外部只读存储器 (Read-Only Memory, ROM) 读入 DSP 程序空间随机存取记忆体 (Random Access Memory, RAM) 中, 然后从指定的地址开始运行程序空间 RAM 中的指令, 该过程称为 DSP 程序的引导加载过程.

TMS320C672x 系列没有自带的 Flash, 一般程

序要烧写到外扩的 Flash 中, 再由 Flash 引导自启动. 由于 Flash 的读取速度比较慢, 一般在上电复位之后, 通过 Bootloader 程序把 Flash 中的应用程序代码引导到 DSP 的内部 RAM 中去执行, 实现系统高速运行. 在 C672X 系列的 DSP 上电复位之后, DSP 会直接由 EDMA 模块采用默认的 ROM 读/写时序从 CE1 空间起始地址上拷贝 1 kB 的代码到 DSP 的内部 RAM 起始地址 0x10000000, 然后 DSP 的 CPU 才开始运行, CPU 从地址 0x10000000 开始执行程序, 这是 DSP 的 Flash 引导自启动的基础, 也是 Bootloader 程序能够被复制进入 DSP 内部 RAM 起始地址运行的关键^[2]. 传统的 Flash 引导加载方法是采用 JTAG 仿真器烧写应用程序, 该方法

在外场试验时极不方便. 本文采用了一种先将 Flash 引导加载的过程烧写到监控程序的方法, 在运行监控程序时就可以完全脱离 JTAG 仿真器, 利用串口传输数据将应用程序加载进 Flash 并自动运行的过程. 同时将 Bootloader 程序控制在 1 kB 内, 从而避免了需要两次编写运行 Bootloader 的麻烦.

2 TMS320C6727 的 Bootloader 引导加载机理

2.1 TMS320C6727 自启动的基本原理

在上电复位之后, TMS320C6727 DSP 会自动去搬移 Flash 起始地址开始的 1 kB 内容. 当 DSP 开始执行程序时, 首先从内部 RAM 的起始地址开始执行程序, 这就是存放 DSP 的启动程序 Bootloader, 而 Bootloader 的功能就是把 Flash 中存放的应用程序代码复制到 DSP 的 RAM 中, 当 Bootloader 程序在 DSP 的 RAM 中执行完成后, 所有的应用程序也就从 Flash 中复制到 DSP 的 RAM 中, 然后 DSP 跳转到 `c_int00()` 复位中断去执行, 开始执行应用程序, 实现了 DSP 的 Flash 引导自启动^[3].

本设计采用的是 Parallel Flash 引导模式^[4]. 在这种引导模式下, 复位结束时, DSP 会自动执行 BOOTLOAD1 程序, 任务是把 Parallel Flash 的前 1 kB 代码搬移到片内执行. 这 1 kB 代码即是 BOOTLOAD2 程序, 用户需要自己编写, 利用这段代码把自己的应用程序再搬移到指定的地方运行. 传统的 Flash 引导加载方法是采用 JTAG 仿真器直接将 Bootloader 程序和应用程序都烧写到 Flash 中, 然后重新上电实现 DSP 的 Flash 引导自启动. 传统方法的引导加载流程如图 1 所示.

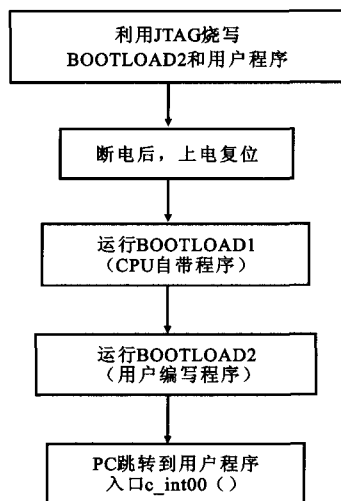


图1 TMS320C6727 的 Flash 引导自启动流程

2.2 TMS320C6727 的 Flash 自启动程序设计的改进方案

传统 TMS320C6727 的 Flash 自启动往往需要利用 JTAG 仿真器将程序烧写到 Flash 中去, 如果更改程序必须拆开机箱连上仿真器才能重新烧写, 这样在外场试验时很不方便. 为了避免在烧写程序的过程中对仿真器的依赖性, 对传统自启动程序的设计方案进行了改进, 利用串口进行数据传输, 先将一个监控程序烧写到 Flash 前半部分中, 然后重新上电自动运行这个监控程序, 在此监控程序的操作过程中将应用程序烧写进 Flash 的后半部分. 同时硬件 FPGA 逻辑中也要设定一个外部离散量开关, 用于切换 Flash 的前后部分. 切换这个开关量到另一端, 再重新上电即可实现对应用程序的自启动. 整个过程除了第一步烧写监控程序时(此时还未去外场试验, 在单板就可以完成, 因此可以使用仿真器)需要仿真器之外, 脱离了 JTAG 仿真器的束缚, 使外场试验时系统软件的升级和维护更加便捷和可操作. 此改进方案的流程图如图 2 所示.

特别需要注意的是, 一旦监控程序被烧写进 Flash, 此程序一般都是经过验证可靠的, 就不能再随意修改, 因此必须对 Flash 的那部分内容进行锁定保护, 以防止非法修改此监控程序. 每次切换到监控程序一端, 自动运行监控程序, 在此完成应用程序的烧写; 切换到应用程序一端, 就可以自动运行应用程序.

3 TMS320C6727 的 Flash 引导加载系统的实现

在 TMS320C6727 的 Flash 引导加载系统实例中, 我们以某系统项目为依托, 通过在其导航计算机板上实施 Flash 自启动程序设计的改进方案, 从而实现了一个全新的 Flash 引导加载系统.

基于 TMS320C6727 的 Flash 引导加载系统的设计实例中, 主要包括两个项目工程: 监控工程(简称 Monitor 端)和应用工程(简称 APP 端). 其系统开发架构如图 3 所示. 无论哪一个工程, 核心在于 CMD 文件的配置以及 Bootloader 程序的编写.

3.1 Bootloader 程序的设计实现

Bootloader 程序的功能就是把烧写在 Flash 中的程序代码和有用信息全部复制到 DSP 内存空间中去执行, 然后指向 `c_int00` 中断去执行程序. 由于本项目实例中 APP 端的应用程序代码较大, 必须被分配到外部 SDRAM 地址空间中, 所以在 Bootload-

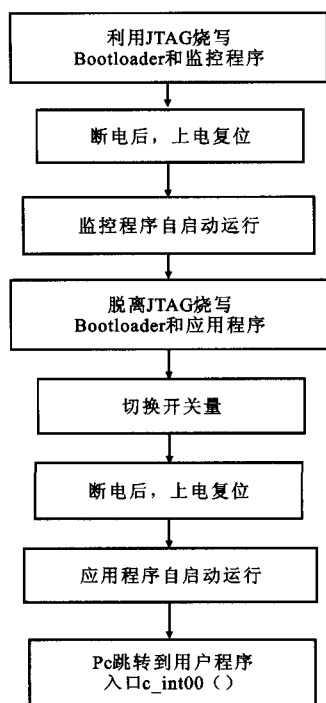


图2 TMS320C6727 的 Flash 自启动程序设计的改进方案

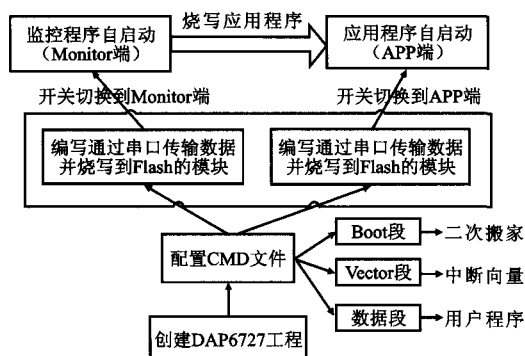


图3 基于 TMS320C6727 的 Flash 引导加载系统开发架构

er 程序中必须初始化 SDRAM 和 PLL. 另外在 Bootloader 程序中必须实现从 Flash 到 RAM 搬移数据的功能, 这样必然涉及到 Flash 的读写操作. 而在本项目实例中配置的是 16MB 的 Flash, 由于 TMS320C6727 的地址总线只有 13 根, 不足以寻址整个 Flash 存储空间, 所以这里将 Flash 分为 64 个页, 通过 PDOUT0 来配置 Flash 的页地址; 而且这 64 个页地址是连续的, 可根据代码的大小进行翻页操作.

在 Bootloader 程序中, 我们实现了以下功能用于完成 Flash 的引导加载:

(1) SDRAM 的初始化: 包括 SDCR、SDRCR、SDTIMR 和 SDSRETR 寄存器的初始化.

(2) PLL 的初始化^[5]: 通过配置 PLL Control/Status Register(PLLCSR)、PLL Multiplier Control

Register(PLLM)、PLL Wrapper Divider x Registers(PLLDIV0, PLLDIV1, PLLDIV2, PLLDIV3) 这几个寄存器就可以得到所需要的 TMS320C6727 内核与 EMIF 时钟^[6].

(3) Flash 的翻页操作: PDOUT0=W, 其中 W 为页码, 取值 0 到 63.

(4) 数据从 Flash 到 RAM 的搬移: 在本项目实例中, 我们总共搬移了 5 个数据段, 分别是: .vec, .text, .cinit, .const 和 .switch. 这 5 个数据段统一放在 copyTable 中, 将 copyTable 的定义放在另一个汇编文件中, 即上面讲过的 CMD 文件的引导表段; 在 Bootloader 程序中将 copyTable 作为外部变量进行引用, 就可以有效的控制 Bootloader 程序的大小在 1 kB 之内, 避免了需要进行两次二次搬家才能将应用程序放在片外 RAM 运行的麻烦.

3.2 TMS320C6727 引导加载系统的流程

整个引导加载系统的流程分为以下几部分:

(1) 外部离散量开关设置为下载程序运行状态, 也就是 Monitor 端, 并且引导模式设置为从 Flash 引导.

(2) 上位机运行终端仿真程序或串口调试工具, 正确设置串口参数.

(3) 连接 JTAG, 烧写 Monitor 端的工程.

(4) 断电后, 脱离 JTAG. 上电复位, 将自动运行 Monitor 端的工程.

(5) 在 Monitor 端工程的运行界面中, 烧写 APP 端的工程.

(6) 外部离散量开关设置为应用程序运行状态, 也就是 APP 端, 并且引导模式设置为从 Flash 引导.

(7) 断电后, 上电复位, 将自动运行 APP 端的工程.

下面以 Monitor 端工程为例, 具体介绍烧写过程.

Monitor 端的工程分为 LOAD 和 FLASH 两个工程. 连接 JTAG, 在 CCS3.3 环境中运行 LOAD 工程. 另外再运行 FLASH 工程, 生成目标码文件 DSP6727_FLASH.out, 运行批处理文件 intel.bat, 生成 BOOT_MON.hex 下载文件.

在运行 LOAD 工程的界面中, 键入 erase/a 命令, 进行 Flash 擦除, 出现 erase ok 说明擦除成功. 然后键入 write/m 命令, 进行 Flash 编程, 首先检测期望编程的 Flash 扇区群是否为空, 为空则准备接收上位机的目标代码, 选择刚才生成的 BOOT_

MON. hex 下载文件进行接收,否则报错退出. 具体过程的运行界面如图 4 所示.

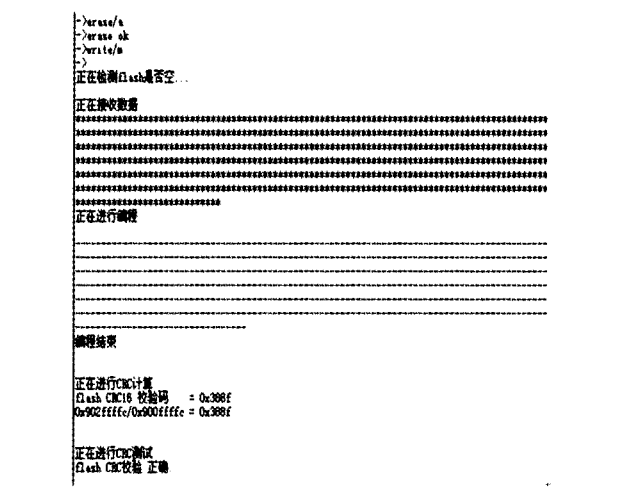


图 4 Monitor 端 LOAD 工程的擦除和编程运行界面

基于以上方法可以看出,LOAD 和 FLASH 工程完成的功能各不相同,LOAD 用于仿真器调试,FLASH 用于生成烧写的目标码文件,因此建议以后每个项目中都可以同时创建好以上两个工程,一个用于加载调试,一个用于固化 Flash 芯片. 应用中可以同时打开,分别激活,加上生成下载目标代码批处理文件,可以简化操作、节省时间. 另外,APP 端工程的烧写与 Monitor 端工程的烧写方法一致,不同在于脱离了 JTAG 来进行烧写.

4 结束语

本文给出了一种采用烧写 Monitor 工程和外部离散量开关切换的方法,实现了在脱离 JTAG 的条件下 APP 端的 Flash 引导自启动. 该方案设计经过试验验证是方便、稳定和可靠的,具有很好的实用性.

参考文献:

[1] 李毅,李连云,张伟宏,等. Bootloader 面向不同结构 Flash 的实现[J]. 计算机工程,2008,34(4):82-83.

[2] 刘远峰,陈志华. 一种新的基于 TMS320C6000 DSP 的 Flash 引导自启动方法[J]. 电视技术,2011,35(21):54-55.

[3] 卞红雨,纪祥春,乔钢. TMS320C6000 系列 DSP 的 CPU 与外设[M]. 北京:清华大学出版社,2007.

[4] Texas Instruments. Using the TMS320C672x bootloader [M]. USA:Texas Instruments,2007.

[5] Texas Instruments. TMS320C672x DSP software programmable phase-locked loop(PLL) controller reference guide[M]. USA:Texas Instruments,2005.

[6] Texas Instruments. TMS320C672x DSP external memory interface (EMIF) user's guide[M]. USA:Texas Instruments,2007.

作者简介:

刘芳女,(1981-),硕士,工程师. 研究方向为嵌入式操作系统及开发环境.

(上接第 45 页)

[3] 杨瑜君. GIS 中最短路径问题的研究与实现[D]. 成都:四川大学,2006.

[4] 刘张雷,史忠科. 一种基于路网变化的动态路径规划策略[J]. 交通运输系统工程与信息,2010,10(3):147-152.

[5] 王士同. 双向启发式图搜索算法 BFFRA * [J]. 电子学报,1990,18(6):34-39.

[6] 郑年波,李清泉,徐敬海,等. 基于转向限制和延误的双向启发式最短路径算法[J]. 武汉大学学报:信息科学版,2006,31(3):256-259.

[7] 任子晖,王坚. 紧急事件的动态交通流模型及双向动态最短路诱导算法[J]. 计算机应用,2008,28(11):2955-2957.

[8] Zhan F B, Noon C E. Shortest path algorithms; an valuation using real road networks[J]. Transportation Science,1998,32(1):65-73.

作者简介:

杨泳男,(1982-),博士研究生. 研究方向为城市智能交通、城市交通规划与管理.