

# DSP 在线升级与资源优化再配置

李 鹏

(济宁学院物理系, 济宁 273155)

**摘要:** 基于 TI TMS320C5410, 提出数字信号处理器(DSP)内的一种动态程序加载技术, 研究并实现 DSP 的在线升级与资源优化再配置。利用该技术解决 DSP 中原有 boot-loader 程序的局限性, 使 DSP 在启动时可选择性地加载应用程序, 实现 DSP 内资源的优化配置和程序运行过程中的动态程序加载与释放, 对 DSP 内的资源进行优化再配置。通过开发设计在线程序写入, 实现 DSP 内程序的在线升级与在线调试。

**关键词:** 数字信号处理器; 动态加载; 优化再配置; 在线升级

## Online Update and Resource Optimal Reconfiguration in DSP

LI Peng

(Department of Physics, Jining University, Jining 273155)

**【Abstract】** This paper proposes a technology for dynamic program loading based on TI TMS320C5410 DSP. By using this technology, the limitation in boot-loader program of Digital Signal Processor(DSP) is solved and any bytes long program can be loaded selectively, optimal configuration of resources in DSP are implemented. Dynamic program loading and release in the running process is realized, the optimal reconfiguration of resources in DSP is done, and online update and online debugging of program in DSP are developed.

**【Key words】** Digital Signal Processor(DSP); dynamic loading; optimal reconfiguration; online update

### 1 概述

数字信号处理器(Digital Signal Processor, DSP)是专门用于数字信号处理的微处理器, 它是随着数字信号处理技术的发展而出现的。目前, DSP已广泛应用于通信与信息系统、信号与信息处理、自动控制、雷达、军事、航空航天、医疗等方面。本文基于TI TMS320C5410 DSP<sup>[1]</sup>, 研究并实现了DSP的在线升级与资源优化再配置。利用动态程序加载技术, 解决了DSP中Boot-loader程序<sup>[2]</sup>无法多次调用、加载程序长度有限制等局限, 使DSP在启动时可有选择性地加载应用程序, 实现DSP内资源的优化配置。

### 2 DSP 与扩展程序存储器

TMS320C5410 DSP芯片是TI司为了实现低功耗、高性能而专门设计的定点DSP芯片, 其片内具有程序ROM空间 16 K 字、在每个机器周期内可以被访问 2 次的双通道RAM(DRAM)空间 8 K 字、在每个机器周期内可以被访问 1 次的单通道RAM(SRAM)空间 56 K 字<sup>[3]</sup>。其内部的ROM空间非常有限, 因此在本系统的应用中, 将使用 8 bit Flash 芯片 SST39VF040 作为外围扩展存储器, 其具有 512 KB 存储空间, 可以保证在不加电的情况下, 存储DSP执行指令和运行过程中必需的数据信息。

本系统采用内存映射的方式访问外部扩展器件, 令 MP/M C=1, DSP处于微处理器模式下, 使DSP片内以 8000h 开始的地址用于内存映射的区域。在系统应用中, 可通过片选访问Flash芯片内的所有存储空间。

### 3 DSP 内动态程序加载技术的实现与应用

#### 3.1 动态程序加载模块的设计与实现

在本系统中, 动态程序加载技术将被实现并封装成一个相对独立的应用模块——动态程序加载功能模块。该功能模

块需要DSP片内ROM中固化的加载引导程序(Boot-loader), 在系统加电启动后加载入片内RAM, 因此针对DSP及Flash芯片的硬件特性, 在本系统中采用 8 bit 加载引导模式<sup>[2,4]</sup>。

动态程序加载模块的基本功能为: (1) 读取 Flash 芯片中指定地址内存的程序代码及数据信息, 将其加载到 DSP 片内 RAM 中; (2) 将接收到的程序代码及数据信息写入 DSP 片内指定的 RAM 中。

根据应用的需要, 本功能模块的入口参数如下: (1) 待读取的程序或数据的源地址; (2) 读取长度; (3) 待写入的目的地址。程序代码如下:

```
loadwordstart:
    stlm    b, brc
    rptb   loadwordend-1
    ld     *ar1+, 8, b    ; read MSB of data
    mvdk  *ar1+, ar3     ; read LSB of data
    andm  #0ffh, @ar3   ; Mask off upper 8-bits
    nop nop
    or    @ar3, b       ; Concatenate
    rpt   #10           ; insert at least 10 cycles
    nop                ; between read and write
    writa bl            ; write object data to destination
    add   #1, a         ; increment destination address
loadwordend
    nop
```

**基金项目:** 国家“863”计划基金资助项目“PKI/PMI 高速加密和验证设备研制”(2003AA141120)

**作者简介:** 李 鹏(1980 -), 男, 助教、硕士, 主研方向: 嵌入式系统, 网络安全

**收稿日期:** 2007-09-15 **E-mail:** penyleecn@hotmail.com

其中,累加器 A 中存放待写入的目的地址;累加器 B 中存放读取长度;寄存器 AR1 中存放读取的源地址。因为采用了 8 bit 扩展存储芯片,所以程序会连续读取 2 个数据空间的内容,并将读取的前一内容作为数据的高 8 bit、后一内容作为数据的低 8 bit,合成一个 16 bit 数据,放置到对应的 RAM 区中。

### 3.2 加载引导时 DSP 内资源的优化配置

当 DSP 芯片上电复位后,其片内固化的 Bootloader 程序将首先被执行,用以将指定程序加载到 DSP 的片内 RAM 中<sup>[2]</sup>,但因其具有不易修改、只能在芯片加电复位时被调用一次且载入程序的长度有限等特性,所以其应用具有很多限制,不仅不能完整地加载所需要的程序和数据,而且不能实现有选择性地对 DSP 内的资源优化配置,因此,本系统中的动态程序加载模块将作为扩展的加载引导程序,首先由 Bootloader 程序加载入片内 RAM,然后动态程序加载模块接替 Bootloader 程序来加载其他代码和数据。其实现流程如图 1 所示。

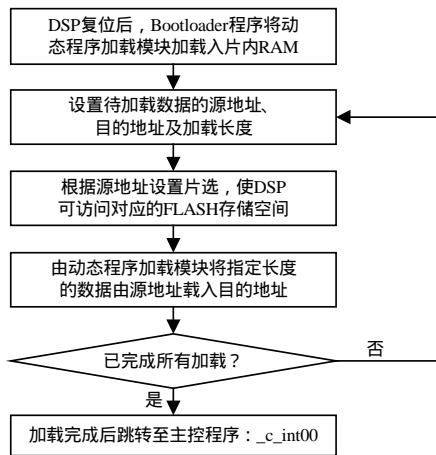


图 1 加载引导流程

为保证 DSP 内程序系统的正确加载,动态程序加载模块应被放置于 Boot 表的前端<sup>[5]</sup>。实现方法如下:在 CMD 文件中,对动态程序加载模块所在的代码段(.sect ".bloader")做如下配置:“.bloader: paddr = boot”,而所有被设置为“boot”状态的代码段,在生成 Boot 表的过程中,将被置于 Boot 表的开始位置。

### 3.3 运行中 DSP 内资源的优化再配置

TMS320C5410 DSP 芯片拥有的 RAM 空间非常有限,仅 64K 字。该空间虽然可以满足一般应用,但不适合结构复杂的应用。一般采用外扩 RAM 芯片的方式实现 RAM 空间的扩展。但在实际应用中,该方法存在很多弊端:(1)增加 RAM 芯片将会直接增加产品的成本;(2)增加芯片的数量会导致整个产品稳定性的降低;(3)增加芯片的同时产品的功耗也会增加;(4)对于片外扩展 RAM 的存取,将会降低 DSP 程序的执行效率。

针对以上问题,本系统在实现过程中采用了另外一种设计方法:通过对应用程序模块的动态加载与释放,解决片内 RAM 空间限制,实现 DSP 内资源的优化再配置。对于 DSP 内的应用系统来说,其所有的功能模块并不会被同时调用,甚至某些功能模块很少被调用,即使调用,对效率的要求也不高。因此,本系统采用动态加载的方式调用此类功能模块,使其共用片内 RAM 空间,对 DSP 内的 RAM 空间进行优化再配置。其实现流程如图 2 所示。

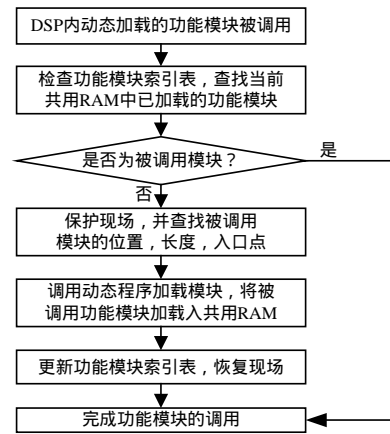


图 2 程序动态加载流程

### 3.4 DSP 的在线升级与在线调试

基于 DSP 的产品一般都采用专用芯片作为程序和数据存储器,本系统即采用 SST39VF040 作为存储器。对于芯片内数据的写入,生产阶段可采用专用设备写入,但对于成型并投入使用的产品来说,不可能轻易将芯片取下。因此,针对 DSP 产品来说,实现在线升级功能至关重要。

在本系统中,通过动态程序加载模块与 Flash 写入模块<sup>[6]</sup>的配合,实现了 DSP 产品的在线升级功能。上层应用程序将升级数据传入 DSP 内,由动态程序加载模块解析其写入地址与长度,然后通过 Flash 写入模块的调用,将数据写入扩展的 Flash 存储器中,最后通过动态程序加载模块将升级后的内容重新加载到 DSP 内的 RAM 空间,实现无需 DSP 复位的在线升级。其应用流程如图 3 所示。

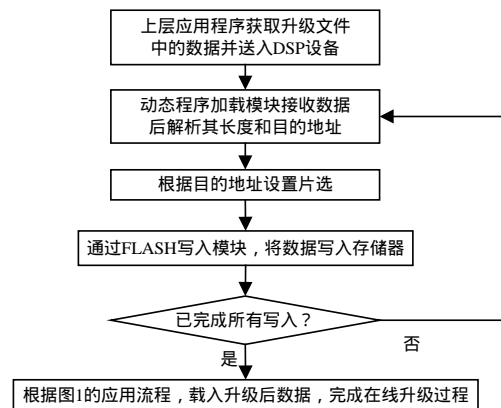


图 3 系统在线升级流程

在线调试功能的实现过程是:由动态程序加载模块接收上层应用程序送入的数据,解析并直接加载到 DSP 片内 RAM,待所有数据接收完成后,跳转到主控程序 \_c\_init00 位置运行。通过在线调试功能,可有效地提高开发效率,缩短开发时间。

## 4 结束语

综上所述,利用动态程序加载技术研究并实现的 DSP 资源优化再配置,不但充分发挥了 DSP 数值运算能力,而且解决了其片内资源利用问题,并提高了以 DSP 为核心的嵌入式产品的稳定性、降低了产品的成本。在实际应用系统中,以该技术研究并实现的 DSP 在线升级与在线调试,提高了以 DSP 为核心的嵌入式产品的易用性,缩短了产品开发周期。目前,该技术已应用于以 DSP 为核心的高速密码设备中,具有较好的推广与应用价值。(下转第 234 页)