

文章编号 1674-2915(2009)05-0432-06

基于 TMS320C6416 的光电实时跟踪 自引导系统的实现

程灵燕^{1,2}, 苏宛新¹, 程飞燕³

(1. 中国科学院 长春光学精密机械与物理研究所, 吉林 长春 130033;

2. 中国科学院 研究生院, 北京 100039; 3. 兰州大学 信息科学与工程学院, 甘肃 兰州 730000)

摘要: 为了实现光电跟踪系统在脱机运行时实时跟踪运动目标, 对基于 TMS320C6416 的光电实时跟踪自引导系统进行了研究。采用了 ROM Bootloader 加载方式, 实现了用户代码上电自动加载。由于在默认情况下, 系统只能拷贝 1 kB 数据到内部 L2 RAM, 而要达到 TMS320C6416 的高速运算, 必须将大量代码 (>1 kB) 从外部 FLASH 拷贝到内部 L2 RAM 中运行, 因此引入了二次引导过程。文中介绍了自引导系统的组成及实现原理, 分别在 Non-BIOS 系统和 BIOS 系统下编写了二次引导程序并在实际应用中得到了验证。实验表明, 基于 TMS320C6416 的光电实时跟踪自引导系统, 满足了系统的实时性要求, 在实际系统中运行稳定可靠。

关键词: 光电实时跟踪系统; TMS320C6416; 二次引导

中图分类号: TP391 **文献标识码:** A

Implementation of bootloader in photoelectrical real-time tracking system based on TMS320C6416

CHENG Ling-yan^{1,2}, SU Wan-xin¹, CHENG Fei-yan³(1. *Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, Changchun 130033, China;*2. *Graduate University of Chinese Academy of Sciences, Beijing 100039, China;*3. *School of Information Science and Engineering, Lanzhou University, Lanzhou 730000, China*)

Abstract: To implement tracking a motion target in real time by a photoelectrical tracking system on off-line, a bootloader system was developed for the photoelectrical tracking system based on TMS320C6416 DSP. By using a ROM bootloader way, the custom code could be loaded automatically on the system. However, this system only could copy the 1 kB data into the L2RAM in a default mode, which did not meet the requirements of the TMS320C6416 DSP for high speed running on off-line. Therefore, this paper induced a secondary bootloader way to copy the rest custom codes in the external ROM/FLASH into the high speed DSP. The detailed programmes of secondary bootloader were written in the Non-BIOS and the BIOS system, respectively, and then they were verified in applications. The experiments show that the bootloader based on TMS320C6416 can

well run in the practical system and can meet the requirements for stability, reliability and real time.

Key words: photoelectrical real-time tracking aystem; TMS320C6416; secondary bootloader

1 引言

光电实时跟踪系统因为能够可靠捕获并稳定跟踪运动目标和实现对目标的实时测量和直观监视,且价格低廉而在许多领域得到了广泛的应用。实现运动目标的实时跟踪,不仅要运动目标进行快速识别与定位,而且要求光电跟踪系统脱机运行,这些都对系统处理器的性能提出了极高的要求。

TMS320C6416 是 TI 公司推出的高速定点 DSP,时钟频率最高可达 1 GHz,其最主要的特点是采用了先进的甚长指令结构(VLIW),每个时钟周期可以执行 8 条指令,所有指令都可以条件执行。同时,TMS320C6416 完成 1 024 点定点 FFT 的时间只要 10 μ s,比传统 DSP 要快 1~2 个数量级,这一高速处理能力具有不可替代的优势,可以很好地满足系统的实时性要求。

为了提高光电实时跟踪系统工作性能,本文选择 TMS320C6416 作系统主处理芯片,并对其自引导系统进行了研究。由于传统直接访问外部非易失存储设备的程序运行方式访问时间最快也要几十纳秒^[1],不能满足光电跟踪系统实时跟踪运动目标的要求,本文采用了 TMS320C6416 的 ROM bootloader 的程序运行方式来实现用户代码上的电自引导,即将用户程序存储在外部非易失存储设备里,CPU 复位后首先把用户程序拷贝到内部 RAM 或外部高速 RAM、DRAM 中,然后用户程序在 RAM 中运行。

2 光电实时跟踪自引导系统的组成及实现原理

2.1 光电实时自引导系统的组成

文中光电实时跟踪系统的主处理器是 TMS320C6416,它提供了 3 种引导方式。根据系统上电复位前相应管脚的设置,DSP 决定系统进

人对应的引导方式。为了实现系统脱机自动加载,采用 ROM bootloader 方式,将 BEA[19:18] 设置为 1 0 (TMS320C6416 EMIFB 8 位 ROM bootloader)。本系统选用 AMD 公司的 AM29LV33C 作为存储用户代码的 FLASH 存储器。该模块的硬件设计电路框图如图 1 所示。

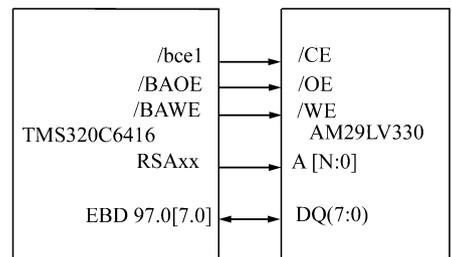


图 1 EMIFB_FLASH 模块硬件电路框图

Fig. 1 Block diagram of hardware circuit for EMIFB_FLASH module

2.2 自引导系统的实现原理及软件流程

自引导系统的实现需要 EDMA 控制器和 CPU 协同工作,其步骤如下:

1) 在 DSP 复位时,采用默认的 ROM 访问时序,通过 EDMA 控制器自动把 EMIFB CE1 空间 ($0 \times 64000000 \sim 0 \times 64000400$) 的 1 kB 数据拷贝到地址 0 处的存储器 ($0 \times 00000000 \sim 0 \times 00000400$) 中。

2) 拷贝结束后,CPU 从地址 0 处开始运行程序。复位后,CPU 根据加载配置初始化相应的 EMIF 控制寄存器。

需要注意的是 TMS320C6416 只支持 8 位的 ROM BOOT 方式,ROM 里数据存储的 endian 格式必须与系统的配置一样,并且只有 EMIFB CE1 空间 ($0 \times 6400 0000 \sim 0 \times 67FF FFFF$) 可以作为 ROM BOOT 时数据存储的空间。

由于系统的用户程序肯定 > 1 kB,要达到 DSP 的高速运算,又不能将大量用户代码放在外部 ROM /FLASH 中运行,因此必须在 1 kB 代码数据中编写二次引导程序,将其他的代码拷贝到

RAM 中运行。所以 1 kB 的程序要做成一个加载器,也就是手工写的 Bootloader,利用这 1 kB 程序把其余的用户程序从 FLASH 搬入 RAM。加载器搬运其他用户程序又是一次加载,称为二次引导

(Secondary Bootloader)。二次引导过程如图 2 所示。

3 二次引导的实现

3.1 CCS 下存储器的配置

CCS 是一个集编程、编译、连接、实时调试、跟踪及实时分析应用程序于一体的完整的 DSP 集成开发环境,可以加速并提高创建和测试实时嵌入式信号处理系统的开发过程。CCS 为 C6000 系列 DSP 提供 DSP/BIOS 功能。DSP/BIOS 是一个尺寸可裁剪的实时多任务操作系统内核,是为需要实时线程调度与同步、主机与目标 DSP 间通信或实时监测的应用而设计的。下面分别在 Non-BIOS 系统、BIOS 系统下展开二次引导程序的讨论。

对于二次引导程序,需要定义一个存放其代码数据的代码段 bootload,其运行和加载地址分别为 BOOTRAM 和 BOOTFLASH。表 1 为系统的存储器配置表。

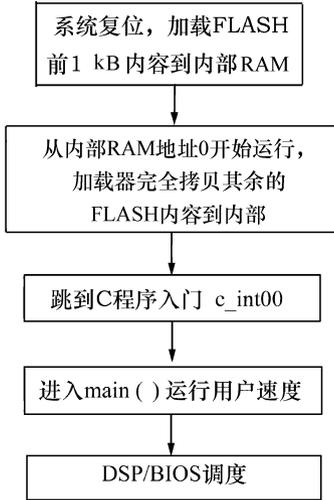


图 2 二次引导过程图

Fig. 2 Processing of secondary bootloader

表 1 存储器段配置表

Tab. 1 Memory configuration

段名	基地址	段长度	备注
BOOTRAM	0 × 00000000	0 × 00000400	FLASH 存储器分为这两部分
BOOTFLASH	0 × 64000000	0 × 00000400	
RESTFLASH	0 × 64000400	FLASHSIZE-1	内部 RAM 分为这两部分
IRAM	0 × 00000400	0 × 000ffc00	

在 Non-BIOS 系统下,分配存储器,.cmd 连接命令文件中需增加以下代码:

```

MEMORY
{
    BOOTRAM: origin = 0 × 00000000, length =
0 × 00000400
    BOOTFLASH: origin = 0 × 64000000, length
= 0 × 00000400
    RESTFLASH: origin = 0 × 64000400, length
= FLASHSIZE-1
}
SECTIONS
{

```

```

    .bootload: load = BOOTFLASH, run =
BOOTRAM
}

```

在 BIOS 系统下分配存储器。图 3 是在 DSP/BIOS 中的存储器空间配置图。

这里需要注意的是必须取消“create a heap in this memory”选项,属性为 code/data。原来的存储区地址若与新增的存储区冲突,要做相应的改动,比如 IRAM 的基本地址就要改为 0 × 00000400。

根据 CPU 访问段的频率决定段的运行地址。只调用一次或很少调用的代码段和数据段的运行、加载地址都分配到 FLASH 空间;需多次调用

的代码段和数据段的运行地址分配到 RAM,加载地址分配到 FLASH 空间;非初始化段只分配运行

地址 RAM。图 4 是 DSP/BIOS 对运行地址和加载地址的分配。



图 3 DSP/BIOS 中的存储器空间配置图

Fig. 3 Memory space configuration in DSP/BIOS

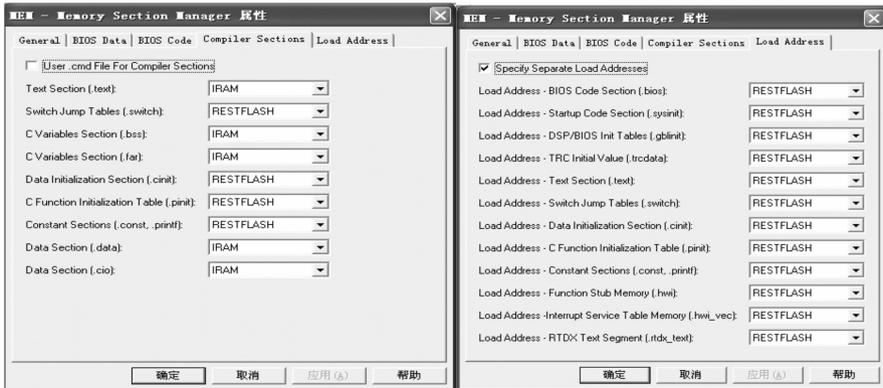


图 4 DSP/BIOS 对运行和加载地址的分配图

Fig. 4 Assignment of running and loading addresses by DSP/BIOS

3.2 编写二次引导程序

系统在开始启动时,还没有初始化 C 语言环境,所以二次引导程序用汇编语言编写。下面是编写二次引导程序的步骤:

1) 配置 EMIF 等配置寄存器。本文中 TMS320C6416 的 EMIFB CE1 空间配置为 8 位异步方式,因为 ROM 加载只支持这种方式。

2) 根据各段的加载地址和运行地址,将各个段中的程序和数据从 FLASH 拷贝到 RAM 相应的位置。这些信息以字节数、运行地址、加载地址的顺序定义在复制表(copytable)中。

3) 跳到 C 程序的入口函数 c_int00()。注意

C 程序的入口点并不是 main() 而是 c_int00,这是因为在调用 main()之前,系统必须先建立 C 语言运行环境。下面是 Non-BIOS 系统下的主要代码:

```

.sect" .bootload" /* 将代码分配到 .Bootload
*/
.ref _c_int00 /* 读取复制表地址然后拷贝
表中的所有段 */
MVKL copytable, A3
MVKH copytable, A3
copysectiontop:
LDW * A3 + +, B0/ * 段长度 */
LDW * A3 + +, A4/ * 段运行地址 (RAM

```

```

地址) */
    LDW * A3 + + , B4 / * 段加载地址 (FLASH
地址) */
    NOP    2
    [! B0] B copydone / * 判断是否拷贝完所有
段 */
    NOP 5
copyloop: / * 段数据拷贝 */
    LDB * B4 + + , B5
    SUB B0 , 1 , B0
    NOP 4
    STB B5 , * A4 + +
    [B0] B copyloop / * 判断该段是否拷贝完毕
*/
    [-B0] B cpysectiontop
    NOP 5
copydone: / * 所有段拷贝完毕跳转至 c_
int00 */
    MVKL . S2 _c_int00 , B0
    MVKL . S2 _c_int00 , B0
    B . S2 B0
    NOP 5
    . global textsize , textrun , textload / * 复制表,
定义了各段的字节数、运行和加载地址 */
copytable:
; ; . text
. word textsize
. word textrun
. word textload
. word 0
. word 0
. word 0

```

代码段的加载地址和段字节数不是固定的。定义变量 `textsize`、`textrun`、`textload` 来表示代码段字节数、代码段运行地址、代码段加载地址。在程序编译和连接之后, CCS 软件除了生成可执行的 (.out) 目标文件之外, 还会生成一个 (.map) 文件。在这个文件中, 包括决定各段在存储器中位置的段连接信息, 如段的大小、段的加载地址和运行地址, 二次引导程序正是根据这些信息来完成程序的拷贝工作。在引导程序文件的结尾处, 加

上一个复制表“copytable”程序段, 把上述段信息按顺序写在 copytable 中。引导程序顺序地提取 copytable 中的信息, 进行拷贝工作。但是, 如果用这种方法取得段信息, 在每次编译程序后, 都要察看 .map 文件, 然后手动地去修改 copytable 中的内容, 这显然是非常麻烦的, 如果不注意还可能出现错误。为了简化这个步骤, 使用 `LOAD_START`、`RUN_START` 和 `SIZE` 这 3 个指令。这样, 在编译后, `LOAD_START`、`RUN_START` 和 `SIZE` 指令会自动提取段的相应信息并存到符号 `textload`、`textrun`、`textload` 中去。注意的是 `textsize`、`textrun`、`textload` 等符号需要在 .cmd 文件中定义。代码如下:

```

. text: LOAD = FLASH REST , RUN = IRAM
LOAD_START( textload ) ,
RUN_START( textrun ) ,
SIZE( textsize )

```

在 BIOS 系统下, 大部分的代码与上述代码相同。对于复制表, 在 CCS3.00 及更高的版本下, 使用 `DSP/BIOS` 和 `table` 命令, 连接器可以更加简单、灵活的产生和管理复制表。由于 CCS 自动生成复制表, 用户不必手动编写。

```

. text load = RESTFLASH , run = IRAM , table
(BINIT)
. data load = RESTFLASH , run = IRAM , table
(BINIT)
. binit load = RESTFLASH / * 复制表在 . binit
段中 */

```

4 结 论

本文采用了 ROM bootloader 的方式来实现系统的自引导。由于在默认情况下系统只能拷贝 1 kB 数据到 L2 RAM, 而要达到 DSP 的高速运算, 不能将大量代码放在外部 FLASH 中运行, 因此本文编写了二次引导程序, 将其他的代码拷贝到 L2 RAM 中运行。本文分别在 Non-BIOS 系统和 BIOS 系统下编写了二次引导程序, 最终实现了光电实时跟踪系统的自引导。实践证明, 基于 TMS320C6416 的光电实时跟踪系统能够在脱机运行时实现上电自动加载, 将存储在 FLASH

AM29LV33C 中的用户代码自动拷贝到 TMS320C6416 的 L2 RAM,很好地实现了运动目标的实时跟踪。该系统的可靠性高,已在实际设计开发过程中得到了验证。

参考文献:

- [1] 季昱,林俊超,余本喜. DSP 嵌入式应用系统开发典型案例[M]. 北京:中国电力出版社,2005. 9.
JI Y, LIN J CH, YU B X. *Typical Cases of DSP Embedded Application Development* [M]. Beijing: China Electric Power Press, 2005. 9 (in Chinese)
- [2] Texas Instruments Incorporated. TMS320C64x/C64x + DSPCPU and Instruction Set Reference Guide, [EB/OL]. [2008-10]. http://focus.ti.com/lit/ug/spru732_h/spru732_h.pdf.
- [3] TMS320C6000 Assembly Language Tools v 6. 1 User's Guide, [EB/OL]. [2008-05]. <http://focus.ti.com/lit/ug/spru186p/spru186.pdf>.
- [4] Code Composer Studio Development Tools v3. 3 Getting Started Guide. [EB/OL]. [2006-10]. <http://www.ece.osu.edu/~potter.ECE609/manuals/spru509h.pdf>.
- [5] TMS320C6000 Tools: Vector Table and Boot ROM Creation. [EB/OL]. [2004-04]. http://www.eetindia.co.in/articles/2004jun/a/2004jun16_dsp_mem_an.pdf?sources=download.
- [6] TMS320C6000 Boot Mode and Emulation Reset. [EB/OL]. [2003-11]. <http://focus.tij.co.jp/jp/lit/an/spra978/spra978.pdf>.

作者简介:程灵燕(1983—),女,内蒙古人,硕士研究生,主要从事 DSP、FPGA 在实时视频信号的采集和处理中的应用。

E-mail:chengly1983@163.com

苏宛新(1962—),男,研究员,主要从事嵌入式系统开发与设计(DSP, FPGA, 高速电路设计)、SOPC 片上系统设计和开发、高速大面阵图像传感器的成像处理、光学测量设计、光学瞄准系统、定向定位系统方面的研究。

E-mail:swx123456@mail.jl.cn