

基于 DSP/BIOS 的原始数据采集驱动程序开发

黄 梅^{1,2}, 吴志勇¹, 梁敏华¹, 黄德天^{1,2}

(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)

摘要: 随着 DSP 外围设备种类和数量的增长, 外围设备的驱动编写和使用已经成为一项依赖于具体硬件和操作系统的必需任务。简述了基于 DSP/BIOS 的设备驱动模型, 以 TMS320DM642 为开发平台, 详细介绍了基于 DSP/BIOS 开发原始数据采集驱动程序的流程。完成了基于 DSP/BIOS 视频处理的基本程序框架设计, 并将该驱动应用于 Camera Link 到 SDI 接口转换的项目中, 实现了原始数据的采集、处理与实时显示。实验结果表明该驱动能很好地应用于需要采集原始数据的开发中。

关键词: 设备驱动; 数据采集; DSP/BIOS; TMS320DM642

中图分类号: TP311 文献标识码: A 文章编号: 1002-1841(2011)07-0050-03

Development of Raw Data Capture Driver Based on DSP/BIOS

HUANG Mei^{1,2}, WU Zhi-yong¹, LIANG Min-hua¹, HUANG De-tian^{1,2}

(1. Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, Changchun 130033, China;
2. Graduate School, Chinese Academy of Sciences, Beijing 100039, China)

Abstract: As the increasing of the kinds of peripheral equipment about DSP, the driver of peripheral equipment has to depend on real hardware and operating system. The device driver model based on DSP/BIOS was introduced. The diver development of RAW data capture based on BSP/BIOS was presented particularly. The design of elementary program structure about video processing which based on DSP/BIOS was finished, and it was used in the project of interface changing from Camera Link to SDI and actualized the collection and processing of RAW data and display in real-time. The results of the experiment show that the driver can be well used in the project of raw data capture.

Key words: device driver; data capture; DSP/BIOS; TMS320DM642

0 引言

TMS320DM642(以下简称 DM642)是一款针对多媒体领域的 DSP,能够很好地应用于视频音频的开发。同时, TI 公司推出的实时操作系统 DSP/BIOS 大大简化了 DSP 应用程序的开发和调试。故利用 DSP/BIOS 开发 DM642 是目前普遍采取的一种方法。传统的软件开发从零开始编程,依次完成各种控制和处理,使得系统正常运行。随着新技术的不断涌现,外围器件设备日益增加,从零编写程序带来了不必要的重复工作,并且对软件工程师的硬件掌握要求也很高。该在在课题中设计的 Camera Link 到 SDI 的接口转换平台上,按照 TI 公司提出的标准化驱动模型要求,开发了完整的原始数据采集驱动,提高了软件的可重用性,在软件或硬件改动时可以使相互的影响降为最小,弥补了 TI 提供的驱动开发工具包(DDK)不支持原始数据采集的缺口的缺陷。

1 DSP/BIOS 的设备驱动模型

为了简化开发过程, DDK 定义的驱动模型(如图 1 所示)将一个设备驱动分解为类驱动(class driver)和微型驱动(mini-driver, IOM 驱动)2 层^[1-2]。其中类驱动处在应用程序和微型驱动之间,提供对多线程 I/O 请求的串行化和同步,并且维护设备数据缓冲区,向上提供 API 接口供应用程序调用,向下通

过适配层与微型驱动相连,实现 API 接口函数到微型驱动层的映射。微型驱动为设备驱动的下层,它实现具体外围设备特有的所用控制和初始化操作,通过接收类驱动层发出的调用命令来决定对底层硬件进行怎样的操作。

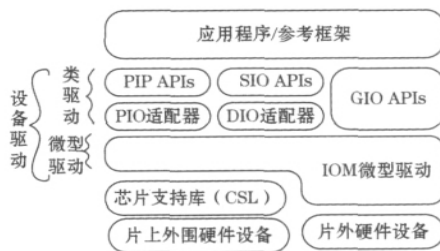


图 1 DSP/BIOS 微型驱动模型

DDK 提供了流输入输出管理模块(SIO)、管道管理模块(PIP)和通用输入输出模块(GIO)3 种类驱动,它们均可以和任何依据分层模型设计的微型驱动一起使用。由于类驱动与硬件无关,每个微型驱动都提供一套标准的微型驱动接口函数供类驱动调用以访问外围设备,以及供 DSP/BIOS 使用来管理设备驱动,所以类驱动不需要开发者做多少修改。而微型驱动采用芯片支持库(CSL)管理外围设备寄存器、内存和中断资源等,由于硬件互不相同,故微型驱动的开发是设备驱动开发的重点。

2 原始数据的采集驱动开发

2.1 视频类驱动

虽然有 SIO、PIP 和 GIO 3 种类驱动模型,但与 SIO 和 PIO 提供的异步阻塞 I/O 模型相比,GIO 提供的同步 I/O 模型是一种更为优越的数据传输模型。GIO 类驱动采用了使用代码量和数据量最小化的设计,能够扩展 API 函数以支持新的应用领域,且具有内建的 IOM 适配层。GIO 的可扩展 API 特性正好可以用在视频开发驱动方面。

在视频驱动程序结构中,为了最大程度地提高视频驱动代码的复用性和通用性,将类驱动又划分为 2 层结构,其上层为 FVID 模型,它是在 GIO 类驱动之上的简单封装,下层则是 GIO 类驱动。FVID 定义的 API 函数如下: FVID_create(应用程序产生一个 FVID 实体对象,即分配并初始化通道对象)、FVID_control(应用程序发送一个控制字到微型驱动)、FVID_alloc(应用程序获取一个缓存区)、FVID_exchange(应用程序与驱动程序交换数据缓冲区)、FVID_free(应用程序归还驱动程序数据缓冲区)、FVID_delete(应用程序删除一个 FVID 实体对象)。

2.2 注册微型驱动

要想在应用程序中正确地调用微型驱动的函数来完成对外部设备的控制,必须在 DSP/BIOS 中注册该微型驱动,方法如下: 在 DSP/BIOS Config 中的 Input/Output -> Device Driver -> User Defined Devices 目录下添加新的设备实例,对其各项属性进行设置。其中就包括指定该设备实例的驱动函数表名称(function table ptr) 一项,例如在个人所开发的原始数据采集中将其命名为 DM642_RawCap_Fxns。在微型驱动中,DM642_RawCap_Fxns 函数表定义如下:

```
IOM_Fxns DM642_RawCap_Fxns
={ mdBindDev ,
( IOM_TmdUnBindDev) IOM_mdNotImpl ,
mdControlChan ,
mdCreateChan ,
mdDeleteChan ,
mdSubmitChan };
```

定义了设备实例的驱动函数表后,在设备实例通道的基础上调用类驱动函数就能调用对应的驱动函数,进而实现对外围设备的操作。FVID、GIO 与微型驱动函数的对应关系如表 1 所示。

表 1 FVID、GIO 与微型驱动函数的对应关系表

FVID	GIO	IOM 驱动函数
——	——	mdBindDev
FVID_create	GIO_create	mdCreateChan
FVID_control	GIO_control	mdControlChan
FVID_alloc	GIO_submit	mdSubmitChan
FVID_exchange	GIO_submit	mdSubmitChan
FVID_free	GIO_submit	mdSubmitChan
FVID_delete	GIO_delete	mdDeleteChan

2.3 驱动函数的实现

2.3.1 mdBindDev 函数

在设备初始化时 DSP/BIOS 会调用 mdBindDev 函数。对于每个配置好的设备实例都要调用一次该函数,并且该函数的调用是在微型驱动的初始化函数调用执行后进行的。在 DM642 核心平台上,该函数的主要内容包括: 在 PCR 寄存器中使能视频端口; 根据传递的参数设定 VPCTL 寄存器的相应位,启动视频端口; 清除 EDMA 通道中断事件对应的中断标志位。

2.3.2 mdCreateChan 与 mdDeleteChan 函数

从应用程序的观点来看,需要在应用程序和设备实例之间创建一个逻辑通信信道,用于交换驱动数据。应用程序可以创建一个或多个这种逻辑通道,这些逻辑通道被微型驱动用通道对象来代表。mdCreateChan 函数负责分配通道对象,并需要根据初始化通道对象的参数域,mdDeleteChan 函数则负责释放指定的通道对象。

在原始数据采集驱动中,mdCreateChan 所完成的主要任务有: 配置 EDMA 传输通道,根据应用程序传递的采集参数设置相应通道的 VCACTL、VCASTRTI、VCASTOP1 寄存器,分配数据采集的帧缓存空间,使能 EDMA 传输完成中断。

mdDeleteChan 则是在 mdCreateChan 函数执行出错的情况下或应用程序调用 FVID_delete 时被调用的,所完成的主要任务有: 禁止视频捕获通道的捕获操作,关闭与该通道相关的所有中断事件,清除相应的中断标志位; 禁止打开的 EDMA 传输通道,释放所分配的 EDMA 通道参数表、传输完成码; 释放所分配的帧缓存空间; 复位视频通道与视频端口。

2.3.3 mdSubmitChan 函数

微型驱动函数 mdSubmitChan 必须处理传递给它的 IOM_Packet 结构体中的命令代码(cmd)。根据该命令代码,函数完成相应的处理。在原始数据的采集驱动中,命令代码有 FVID_ALLOC、FVID_FREE 和 FVID_EXCHANGE。FVID_FREE 对应的任务是释放应用程序返回的 FVID_Frame 对象,FVID_ALLOC 对应的任务是为应用程序分配一个包含有最新采集数据的 FVID_Frame 对象,FVID_EXCHANGE 则是同时完成前两者的所有任务。

2.3.4 mdControlChan 函数

虽然微型驱动要支持的控制操作是依赖于具体设备,但 IOM 还是定义了一些标准的控制代码来供开发者选择实现。mdControlChan 函数不仅在应用程序调用 FVID_control 函数时被调用,在微型驱动内部也会被其他的微型驱动接口函数调用。在 mdControlChan 内部也是通过判断控制代码来调用相应的实现函数。这里仅仅对 VPORT_CMD_START 和 VPORT_CMD_SET_VINTCB 控制代码对应的任务作简要说明。控制代码 VPORT_CMD_START 对应函数所完成的任务是使能视频采集通道,开放采集事件的产生,使能视频端口中断。控制代码 VPORT_CMD_SET_VINTCB 对应函数所完成的任务是根据应用程序传递的参数设置中断回调函数,这就确保一旦有采集中断产生,能够正确调用应用程序所要求的中断函数。

2.3.5 captureEdmaISR 函数

尽管 captureEdmaISR 函数不是微型驱动接口函数,但在

数据的采集中却发挥着举足轻重的作用。captureEdmaISR 函数是被 EDMA 传输完成中断调用的, 执行的任务包括: 将刚刚采集的一帧数据的作为最新采集帧, 更新当前采集帧和下一采集帧的缓存区地址; 如 `chan -> packetIOM` 不为空则调用通道回调函数返回最新采集帧; 根据 EDMA 传输完成码修改 EDMA 传输链中 RAM 参数中的目的地地址。

3 DSP/BIOS 程序框架及驱动测试结果

3.1 DSP/BIOS 程序框架

该次原始采集驱动所应用的 DSP 核心平台主要是实现 Camera Link 到 SDI 的视频信号接口转换。DSP 主要实现的功能是采集 FPGA 传送的原始数据, 缓存到 SDRAM 中, 进行相应的处理, 然后再按照 BT. 656 格式送给解码芯片显示。

软件设计时, 首先需要进行核心硬件初始化, 这些主要都是在 main 函数中完成的, 包括 CSL 初始化、外设使能/禁止、EMIFA 初始化等。在 main 函数中还需要对视频输入输出设备进行初始化, 视频图像处理中所用到的算法模块进行注册, SCOM 单元的建立以及启动各任务模块。然后在 DSP/BIOS 上运行完成相应功能要求的应用程序。

如图 2 所示, DSP/BIOS 上运行完成的应用程序包括将图像采集任务函数 `tskVideoInput()`、图像处理任务函数 `tskVideoProcess()` 和图像显示任务函数 `tskVideoOutput()`, 它们分别实现图像的采集、图像的处理以及图像的显示。建立了这个程序框架后, 仅仅修改处理算法就能实现不同方法的图像处理和实时显示, 而不必去重复编写基本的程序框架函数。

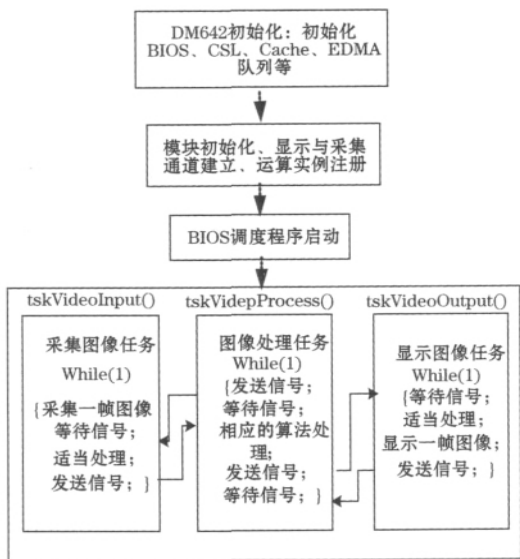


图2 基本程序框架结构

3.2 驱动程序测试结果

与采集驱动有关的主要代码如下:

```

FVID_Handle capChan;
/* 创建采集通道* /
capChan = FVID_create(
    "/VPIACAPTURE/A/1",
    IOM_INPUT
    &status ,

```

```

    &DM642_vCapParamsChan ,
    NULL);

```

/* 设置采集通道的中断处理函数* /

```

FVID_control(
    capChan ,
    VPORT_CMD_SET_VINTCB ,
    &DM642_vCapParamsInt);

```

/* 启动采集* /

```

FVID_control(
    capChan ,
    VPORT_CMD_START ,
    NULL);

```

/* 申请一个采集到最新数据的缓存空间* /

```

FVID_alloc( capChan &capFrameBuf);

```

/* 交换一个数据采集缓存空间* /

```

FVID_exchange( capChan &capFrameBuf);

```

其中 DM642_vCapParamsChan 为采集通路参数, DM642_vCapParamsInt 为视频口全局中断程序接口参数。利用该驱动程序所采集的原始数据图像如图 3 所示。

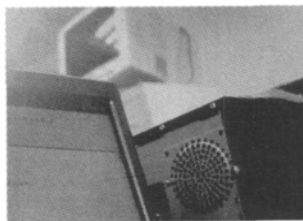


图3 采集到的原始图像

4 结束语

介绍了基于 DSP/BIOS 的外设驱动模型, 并重点针对 DM642 的视频处理系统这一硬件平台, 以开发原始数据的采集驱动为例, 详细介绍了驱动程序的开发步骤和具体内容。在应用程序中修改相应参数值, 就可以很好地实现不同格式的原始数据采集, 大大节省了开发时间, 并可以很好地移植到其他具有相似硬件接口的系统中, 对其他类型接口设备驱动的开发也有参考意义。在文章的最后, 还简单介绍了基于 DSP/BIOS 的图像处理系统的程序框架, 在该框架的基础上, 以后实现各种图像处理的设计时, 就只需重点考虑相应的处理算法。

参考文献:

- [1] Texas Instruments. TMS320 DSP/BIOS User's Guide. Literature Number: SPRU423B 2002.
- [2] TI DSP/BIOS 用户手册与驱动开发. 王军宁, 何迪, 马娟, 等译. 北京: 清华大学出版社 2007: 190-236
- [3] 王跃宗, 刘京会. TMS320DM642 DSP 应用系统设计与开发. 北京: 人民邮电出版社 2009
- [4] 于凤芹, 李江. 利用 EDMA 和 DSP/BIOS 实现高速数据搬移与处理. 仪表技术与传感器 2008(2): 65-67.
- [5] 许延发, 秦庆旺, 倪国强. 基于 DM642 融合系统的 A Trou 小波实时图像融合算法. 光学精密工程 2008, 16(10): 2045-2049.

作者简介: 黄梅(1986—), 硕士研究生, 主要研究方向为实时图像处理。
E-mail: hm2005678@sina.com