

基于 DSP 的 USB 设备固件程序开发

USB equipment firmware procedure development based on DSP

(成都理工大学) 詹鹏 郭勇 蔡顺燕 高嵩
ZHAN Peng GUO Yong CAI Shun-yan GAO Song

摘要: 本文介绍了一种采用 TMS320VC5509A 自带的 USB 模块来实现 USB 数据传输的方案, 对该 DSP 内部的 USB 模块的构造及其传输原理做了分析, 详细的介绍了如何利用 CCS 内部集成的片级支持库(CSL)来实现 USB 设备固件程序的设计, 并给出了相关的部分代码。该方案大大降低了系统硬件设计的复杂度和调试的难度, 提高了系统的集成度和稳定性, 适用于便携式设备中。

关键词: TMS320VC5509A; CSL; DSP; USB; 固件程序

中图分类号: TP311、TN919、TP334 **文献标识码:** B

Abstract: This article introduced a method to realize USB data transmission using the TMS320VC5509A's own USB module, analyzed the DSP's interior USB module's structure and its transmission principle, how to realize the USB firmware procedure design using chip support library which is integrated in CCS has been introduced in detail, and the related partial codes has been given. This method greatly reduced the complexity of the system's hardware design and the debugging difficulty, it enhanced the system integration rate and the stability, using in the portable equipment is suitable.

Key Words: TMS320VC5509A; CSL; DSP; USB; Firmware

1 引言

通用串行总线(USB)是一种高传输速率的串行接口总线, 由于它具有即插即用、易于扩展、传输速率高、连接简单、使用方便的特点, 被广泛的应用于各种计算机外设、数据采集、数字设备以及工业控制等领域中。选取合适的 USB 控制芯片是 USB 数据传输系统设计的重要环节, 目前常采用 USB 控制芯片有两类, 一类是不带 MCU 的控制芯片, 如 PDIUSB12; 而另一类是带 MCU 的, 如 EZ-USB 系列等。这两类专用的控制芯片都需要通过芯片所提供的接口来与系统处理器进行通信, 而本文介绍的 USB 接口方案没有采用专用的 USB 控制芯片, 而是采用 DSP 芯片 TMS320VC5509A 内部集成的 USB 模块来完成 USB 通信, 使系统硬件设计和调试难度都大大降低, 同时也提高了系统的集成度和稳定性。

2 USB 接口设计

2.1 TMS320VC5509A 简介

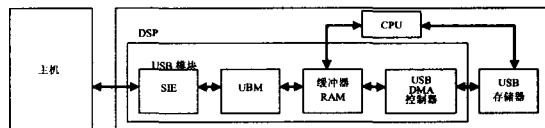
TMS320VC5509A 是 TI 推出的高性能、低功耗的定点 DSP 处理器, CPU 频率最高可以运行在 200MHz (内核电压工作在 1.6v), 每个时钟周期可执行一到两条指令, 有两个算术逻辑单元(ALU), 两个硬件乘法器, 是一款具有较高性价比、高集成度、低功耗的 DSP 芯片, 适用于便携式设备中。TMS320VC5509A 集成了 128K 16Bits RAM, 32K 16Bits 的 ROM, 且带有 EMIF 接口, 可实现与多种存储器之间的无缝连接。片内还有丰富的外设: 2 个 20 位的定时器; 3 个多通道缓冲串口(McBSP); USB 全速接口(12Mbps); I2C 接口; 实时时钟等。

詹鹏: 在读研究生

基金项目: 四川省基础研究项目(05JY029-087-2)

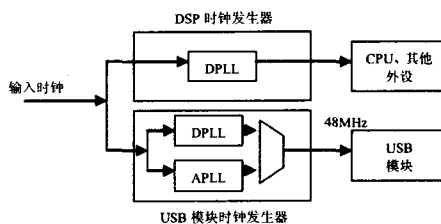
2.2 TMS320VC5509A 的 USB 模块介绍

TMS320VC5509A 自带的 USB 模块支持 USB1.1 协议全速标准, 该 USB 模块有两个控制端点(仅用于控制传输)和 14 个通用端点, 通用端点支持中断、批量和同步传输。DSP 的 USB 模块由串行接口引擎(SIE)、USB 缓冲管理单元(UBM)、缓冲器 RAM、USB 的 DMA 控制器和缓冲器 RAM 仲裁器组成。DSP 存储器与 USB 主机之间的数据传输模型如图(1)所示: 在 IN 事务中, CPU 或 USB 的 DMA 控制器将准备发送到主机的数据放到缓冲器 RAM 中, 等待 USB 缓冲管理单元将数据取出后送到串行接口引擎(SIE), 由 SIE 负责数据的发送; 在 OUT 事务中, SIE 接收到由主机发来的数据后送到 UBM, 再送入到缓冲器 RAM 中, 等待 CPU 或 USB 的 DMA 控制器将数据取出后送到 DSP 的存储器中。



图(1) DSP 存储器与 USB 主机之间的数据传输

2.3 USB 模块的时钟发生器:



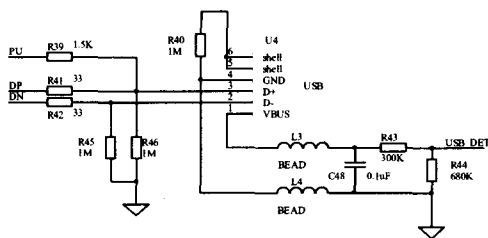
图(2)时钟发生器

USB 模块有专门的时钟发生器, 独立于 CPU 的时钟发生器, 如图(2)所示, 由 DSP 时钟发生器输出的时钟送入到 CPU 和

其他外设(不包括 USB 模块),而 USB 模块的时钟由单独的时钟发生器为它提供。USB 模块的时钟可选择用模拟锁相环(APLL)或数字锁相环(DPLL)来产生。模拟锁相环与数字锁相环相比有它独特的优势,TI 推荐使用模拟锁相环来产生 USB 模块的时钟。提供给 USB 模块的时钟必须设置为 48MHz。

2.4 USB 接口硬件设计

如图(3)所示,DP、DN 和 PU 端口接到 5509A 的 USB 模块端口引脚上,DSP 的 DP、DN 端口以差分方式传送数据,PU 端口通过 1.5k 的上拉电阻接到 USB 总线的 D+端,DSP 可以通过编程来控制 PU 端口,从而控制设备与 USB 总线的连接与断开。主机 USB 接口提供的电源经电阻分压后送入 DSP,供 DSP 检测是否与主机的 USB 接口连接上。



图(3) USB 接口硬件连接图

3 USB 设备固件程序设计与实现

USB 驱动程序主要包括两个部分:设备固件程序和主机端的设备驱动程序,设备固件程序是指运行在设备端 CPU 中的程序,它主要完成 USB 协议的处理及设备与主机的数据交换,设备固件程序设计是重点也是难点,本文主要介绍设备固件程序的编写。

3.1 DSP 的 CSL 编程

在 DSP 系统软件设计中,一般会大量涉及到对片上外设的操作,而这往往要消耗过多的时间和精力。TI 为 5000 系列和 6000 系列的 DSP 提供了各自的片级支持库 CSL (chip support library),CSL 库函数主要用于配置和控制 DSP 片上外设,使片上外设更容易使用,缩短开发时间,增加可移植性。用 CSL 来配置和管理 DSP 的 USB 模块可以使 USB 固件程序的开发变得更加方便和快捷。CSL 已集成到 CCS 中,我们可利用 CCS 提供的图形用户接口(GUI)来对 USB 模块进行相应的初始化配置,或者直接调用 CSL 库函数来配置。

3.2 USB 设备枚举过程

当主机检测到有设备连接到 USB 总线时,主机要向 USB 设备发出一系列的设备请求,获取 USB 设备的一些属性,如设备支持的最大传输速率、设备接口特性、设备端点个数以及每个端点支持的传输方式等。接着主机为 USB 设备分配一个唯一的设备地址,然后 USB 设备才可以正常使用,这个过程叫做枚举。USB 设备的枚举过程分以下几步:

- 1.USB 设备加电,并连接上 USB 总线。
- 2.主机检测到 USB 设备,总线复位,集线器发送复位信号并维持至少 10 毫秒。
- 3.复位完成,USB 设备处于缺省状态,此时设备将以缺省地址 0 响应主机请求
- 4.主机发出请求,从缺省地址 0 读取 USB 设备的设备描述符。
- 5.主机为该 USB 设备分配一个新的设备地址。
- 6.主机从新的设备地址再次读取 USB 设备的设备描述符。
- 7.主机读取设备的配置,包括配置描述符、该配置的所有接口描述符、接口的所有端点描述符以及字符串描述符。

8.主机加载设备驱动程序,USB 枚举过程结束,USB 设备可以正常使用了。

3.3 USB 设备固件程序设计

USB 设备固件程序主要包括两个部分:USB 模块的初始化和 USB 中断服务程序,下面分别对这两个部分的实现做个简单的介绍。

3.3.1 USB 模块的初始化

USB 模块初始化流程为:首先关闭中断,然后设置 API 函数指针向量,设置 USB 模块时钟,初始化 USB 设备的端点,调用函数 USB_init()初始化 USB 模块,用函数 IRQ_plug()初始化中断向量表,再打开中断,最后调用函数 USB_connectDev()使 USB 的 D+端通过 1.5k 电阻上拉,从而使设备接入 USB 总线,随后开始 USB 设备的枚举过程。下面对 USB 模块初始化过程中涉及到的部分 CSL 库函数做个简单介绍:

USB 模块时钟设置函数 USB_initPLL(),该函数有三个参数,分别是 USB 模块的输入时钟、USB 模块的输出时钟(必须设置为 48MHz)、输入时钟的分频数(该参数在 USB 模块的寄存器中用 2 bit 来设置,即分频数只能设定为 1,2,3 或 4,所以在硬件设计时需考虑好 DSP 的外部输入时钟频率,使 USB 模块的时钟能够设置为 48MHz)。

端点初始化函数 USB_initEndptObj()有七个参数,该函数用于对端点的端点号、端点的传输方式(控制传输,中断传输,批量传输和同步传输)、端点能够接收的包的最大值、引发该端点产生中断的中断事件、产生该端点中断后去执行的函数等属性进行相应的设置。

初始化中断向量表函数 IRQ_plug()有两个参数,第一个参数为中断事件 ID(DSP 中各种类型的中断在 CSL 的头文件中都定义了不同的 ID 值),第二个参数为中断函数地址(产生与事件 ID 对应的中断时转而执行的中断函数的地址)。USB 模块初始化的部分代码如下:

```

USB_EpObj usbEpObjOut0, usbEpObjIn0.....; //创建 USB 端点
USB_EpHandle myUsbConfig[] = {&usbEpObjOut0, &usbE-
pObjIn0,.....,NULL};
void USB_Init()
{ .....
CSL_init(); //CSL 初始化
INT_DisableGlobal(); //关中断
USB_setAPIVectorAddress(); //初始化 USB 模块 API 函
数向量指针
USB_initPLL(12,48,0); //设置 USB 模块时钟,必须设置
为 48MHz
event_mask = USB_EVENT_RESET | .....; //引发端点中断
的事件
////端点初始化
USB_initEndptObj(USB0, &usbEpObjOut0, USB_OUT_EP0,
USB_CTRL,0x40, event_mask, USB_ctl);
USB_initEndptObj(USB0, &usbEpObjIn0, USB_IN_EP0,
USB_CTRL,0x40, event_mask, USB_ctl);
..... //其它端点初始化程序
USB_init(USB0, myUsbConfig, 0x40); //USB 模块初始化
IRQ_plug(usbId,&USB_isr); //初始化中断向量表

```

技
术
创
新

```

IRQ_globalEnable();    //开中断
USB_connectDev(USB0); //设备连接到 USB 总线上
.....
}

```

3.3.2 USB 中断服务程序

当产生 USB 中断时,程序会执行相应的 USB 中断程序,在 USB 中断程序中我们可以调用函数 USB_evDispatch()来处理中断事务,该函数会清除相应的中断标志位,并且发布 USB 中断事件,从而去执行相应的端点中断函数。若是控制端点 0 中断,则执行控制端点 0 的中断函数,完成相应的复位、挂起、SETUP(设备请求包)等操作。若是其他通用端点中断则按照该端点初始化时定义的传输方式来完成数据传输。USB 中断函数及端点中断函数的部分程序如下:

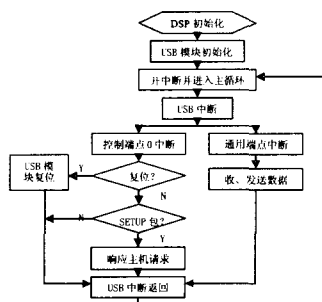
```

interrupt void USB_isr()    //USB 中断
{
    USB_evDispatch();
    //控制端点 0 中断处理函数
    void USB_ctl (USB_DevNum DevNum,    USB_EpHandle
hEp0In,USB_EpHandle hEp0Out)
    {
        .....
        if(USB_ctl_events & USB_EVENT_RESET) {.....} //复
位处理
        if(USB_ctl_events & USB_EVENT_SUSPEND) {.....} //
挂起处理
        if ((USB_ctl_events & USB_EVENT_SETUP) ==
USB_EVENT_SETUP) //收到 SETUP 包
        {
            if (USB_getSetupPacket (DevNum,&USB_Setup) ==
USB_TRUE)
            {.....} //处理 SETUP 包,完成相应的 USB 设备枚举操
作
        }
        .....
    }
    ..... //其它端点中断函数
}

```

3.4 USB 设备固件程序流程图

如图(4)所示,DSP 及其 USB 模块的初始化完成后进入主循环,等待 USB 中断。若是控制端点中断则进入控制端点中断服务程序,完成设备枚举的相关操作;若是通用端点中断,则按照通用端点定义的传输方式来完成数据的传输。然后中断返回,进入主循环。



图(4)USB 设备固件程序流程图

4 结论

本文介绍的采用 TMS320VC5509A 自带的 USB 接口来完

成 USB 数据传输的方案,使系统硬件设计的复杂度和系统调试的难度都大大的降低,同时也提高了系统的集成度和稳定性,减小了系统体积和功耗,适合于在便携式产品中应用。

创新点:未采用传统的 MCU 加专用的 USB 控制芯片的方案,而是采用 DSP 芯片内部集成的 USB 模块来完成 USB 通信,使系统硬件设计和调试难度都大大降低,同时也提高了系统的集成度和稳定性。

参考文献

[1]TMS320VC5509A Fixed-Point Digital Signal Processor Texas Instruments Incorporated June 2006.

[2]滕岩峰,周雪峰 带 USB 接口的 100MHZ 高速数据采集系统的设计与实现[J]微计算机信息 2006.22.7-2.

[3]Using the USB APLL on the TMS320VC5509A Texas Instruments Incorporated February 2004.

[4]陈启美,丁传锁编著 计算机 USB 接口技术 南京大学出版社 2003.

[5]TMS320C55x CSL USB Programmer's Reference Guide Texas Instruments Incorporated October 2001.

[6]TMS320C55x Chip Support Library API Reference Guide Texas Instruments Incorporated September 2004.

作者简介:詹鹏(1982-),男,在读研究生,成都理工大学通信与信息系统专业研究生。

Biography:ZHAN Peng (1982-), Male, Postgraduate Student, Major in communication and information system of Chengdu University of Technology.

(610059 四川成都 成都理工大学)詹鹏 郭勇 蔡顺燕 高嵩 (Chengdu University Of Technology Si Chuan Cheng Du 610059)Zhan Peng Guo Yong Cai Shun-yan Gao Song
通讯地址:(610059 成都 成都理工大学研究生院榕树园 4 单元 4-3) 詹鹏

(收稿日期:2008.03.13)(修稿日期:2008.4.25)

(上接第 122 页)

[3]李驹光,聂雪,江泽明,王兆卫. ARM 应用系统开发详解[M]. 北京:清华大学出版社,2003 年 12 月。

[4]李维是,郭磊. 液晶显示技术[M]. 北京:电子工业出版社,2000.

作者简介:闫明,1981 出生,男,汉族,山东青岛,硕士研究生,中国海洋大学信号与信息处理专业,主要研究方向为:嵌入式、机器人;李欣,1949 出生,男,汉族,中国海洋大学信息学院电子系教授,主要研究方向:接入网、嵌入式系统。

Biography:Yan Ming (1981-), male, Han, borned in Qingdao city Shandong province, graduate student, studying in Ocean University of China. Research direction: Embedded System, Robot;Li xin(1949-), male, Han, professor, working in Ocean University of China. Research direction: Access Network, Embedded System..

(266100 山东青岛 中国海洋大学信息科学与工程学院) 闫明 李欣 吴伟

(Electrical Engineering Department, Ocean University of China, Qingdao, 266100) YanMing LiXin WuWei

通讯地址:(266100 山东 山东青岛松岭路 238 号中国海洋大学信息学院 2005 级研究生电子班)闫明

(收稿日期:2008.03.13)(修稿日期:2008.4.25)