

基于 USB 接口 VSAT 基带数据采集系统设计

0 引言

在众多的卫星通信系统中，VSAT（甚小口径卫星终端）卫星通信系统由于具有众多特点，近年来发展很快，特别在专业通信网领域存在广阔的市场，利用此系统进行通信具有灵活性强、可靠性高、成本低、使用方便及小站可直接装在用户端等特点，并且 VSAT 用户数据终端可直与计算机联网，完成数据传递、文件交换、图像传输等通信任务，从而摆脱了远距离通信地面中继站的问题，因此，此系统广泛用于石油、银行、铁路、民航、军事等行业。

在 VSAT 系统中，用户终端通常是由数字卫星接收机和计算机终端组成，数字卫星接收机包括下变频器、解码器、FEC（前向纠错），解复用器、数字基带处理器和频率合成器等部分，接收机输出的数据一般通过 RS-232，RS-422/449 以 1.2kbit/s—64kbit/s 的数据率交给计算机完成应用处理，有些高速卫星数据接收设备解调以后，数据通过额外高速数据卡送入计算机。这种传统的数据卡不仅安装麻烦，且易受机箱内环境的干扰，特别是这种卡通常只能接收单路数据，受计算机插槽数量和地址，中断资源的限制，不可能挂接多个设备，因此，资源浪费尤其严重。

针对这种情况，本文结合卫星接收设备数据采集的实际需要，设计了一种基于 USB（通用串行总线）接口的 VSAT 基带数据采集系统，该系统通过对 USB 接口控制逻辑的合理设计和芯片内部 FIFO 的有效运用，充分应用了 CPLD（复杂可编程逻辑器件）的灵活性，仅采用单片 USB 接口控制芯片就实现了对多路多速率数字信号的实时采集，这与以往采用 USB 集线器来实现对多路多速率数字信号采集的系统相比，具有成本低、实现方便等优点。

该系统最多可同时采集 8 路数字信号，单路最高速率可达 2Mbit/s，系统用于控制的 CPLD 采用 Altera 公司的 EPM7128SLC84；USB 接口控制芯片采用 Cypress 公司 EZ-USB FX2 系列 CY7C68013。

1 系统原理

1.1 USB 简介

USB（通用串行总线）是一种应用在 PC 领域的新型接口技术，支持在主机与各种即插即用外设之间进行数据传输，它由主机预定传输数据的标准协议，在总线上的各种设备分享 USB 总线带宽，它的主要优点是：数据传输速率明显快于一般的串口，支持控制传输、中断传输、块传输的和同步传输 4 种传输方式，以满足不同外设的需要，最多可连接 127 个外设，支持热拔插和即插即用，占用的系统资源少（只占用 1 个 IRP（I/O 请求包））；无

总线竞争等。USB 总线技术的提出就是想利用单一的总线技术来满足多种应用领域的需要，USB1.1 协议支持低速 1.5Mbit/s 和全速 12Mbit/s 两种传输速度。2000 年发布了 USB2.0 协议，它向下兼容 USB1.1 协议，但在 USB1.1 的基础上有了质的提高，其理论传输速度可到 480Mbit/s。

1.2 系统结构

该系统总体框架分采集控制部分、USB 接口控制部分和主机 3 部分，系统框图如图 1 所示。

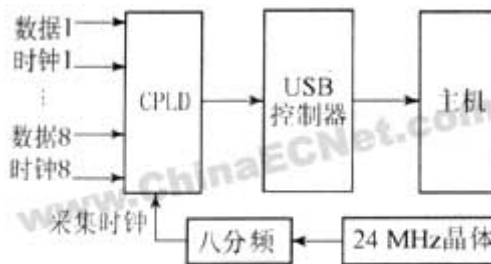


图 1 系统框图

外部输入的 8 路信号中时钟、数据均为 TTL 电平，采集用的统一时钟为 3M 赫兹，是由 USB 控制芯片的外部时钟经八分频得到的。

系统的工作过程如下：每路信号在 CPLD 的控制下，按采集时钟的统一节拍，各路信号的数据和时钟都被采集进 CPLD。在 CPLD 内，先进行各路信号的时钟上升沿检测，如检测到时钟上升沿，则在相应路数的标志位置 1，这表明此时采到的数据有效，若没检测到时钟上升沿，则在相应路数地标志位置 0，表明此时采集的数据无效，这样形成的各路信号数据，再加上各路信号数据是否有效的标志，形成 2 字节的数据送入 USB 控制芯片的 FIFO 中，数据进入 USB 控制芯片后，通过 USB 接口将数据传输到主机，数据在主机中用软件进行处理，按照各路信号数据是否有效的标志位，将有效数据取出，组合成各路数据。

数据送入主机时的格式如图 2 所示，前一字节为各路信号数据是否有效的标志，紧跟着的字节为各路信号的实际数据，采用这种特殊的数据格式，以区分各路信号，便于采用统一的时钟来实现对多路多速率数字信号的实时采集，从而减少了很多硬件电路，降低了成本和复杂性，实现更为方便，这也是本系统的一个创新点。



图 2 数据格式

2 硬件构成

2.1 EZ—USB FX2 (CY7C68013) 芯片

CY7C68013 属于 Cypress 公司的 FX2 系列产品，它提供了对 USB2.0 的完整解决方案，该芯片虽然采用低价的 8051 单片机，但仍然能获得很高的速度，主要包括 USB2.0 收发器、SIE（串行接口引擎）、增强型 8051 处理器、8.5KB 的片上 RAM，4KB FIFO 存储器、I/O 口、数据总线、地址总线和 GPIF（通用可编程接口）。

CY7C68013 与外设有可编程接口 GPIE 和从 FIFO 两种接口方式。可编程接口 GPIF 是主机方式，可以由软件设置读写控制波形，灵活性很大，几乎可以对任何 8bit/16bit 接口的控制器、存储器和总线进行数据的主动读写，使用非常灵活，从 FIFO 方式是从机方式，外部控制器可像对普通 FIFO 一样对 FX2 的多层缓冲 FIFO 进行读写。在本设计中为了方便控制，采用从 FIFO 方式。

2.2 CPLD 芯片 EPM7128SLC84

在本设计中，所用的 CPLD 采用 Altera 公司的 EPM7128SLC84。它属于 MAX7000 系列，是 Altera 公司的基于第 2 代 MAX 结构的 CPLD。它提供多达 5000 个可用门、128 个宏单元，其引脚到引脚延时快达 6ns。可以容纳各种独立的组合逻辑和时序逻辑功能，EPM7128S 的优点是它基于 E2PROM，可以通过 JTAG 口进行在线编辑，设计者可将设计内容从 PC 机上通过下载电缆和 JTAG 口对 EPM7128S 进行任意次修改，它有多达 100 个 I/O 引脚可供编程使用，方便系统扩展存储空间和外设。

本系统使用 EPM7128S 的主要功能是对各路信号的时钟上升沿进行检测，另外还需要完成一些控制功能，完成各路信号数据的合路，形成特定的数据格式，送入 USB 控制芯片的 FIFO 中。

2.3 硬件连接

根据系统对数据传输的速度和实时性的要求，配置 CY7C6013 的工作接口模式为从 FIFO 模式，硬件连接方式如图 3 所示。

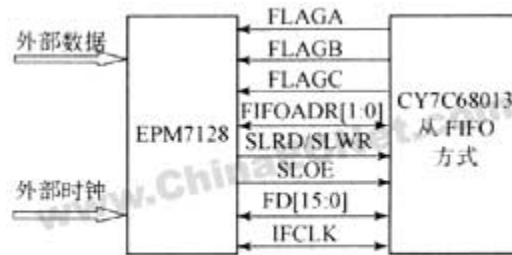


图3 硬件连接框图

CPLD 控制采集外部数据的时钟可同时作为 CY7C68013 的从 FIFO 模式的读写控制时钟，即 CY7C68013 的接口时钟连接到 IFCLK 引脚，FLAGA—FLAGD 为 FIFO 标志引脚，用于映射 FIFO 的当前状态，SLWR/SLRD 是 CY7C68013 从 FIFO 的写使能/读允许信号，CPLD 向 CY7C68013 从 FIFO 提供从 FIFO 输出允许信号 SLOE，仅在数据输出时有效，FD[15: 0] 为 16 位双向数据总线。FIFOADR[1: 0] 为端点 FIFO 选择信号。

3 软件设计

本系统软件设计部分包括固件、驱动程序和应用程序的设计。

3.1 USB 设备固件程序设计

固件程序是指运行在设备 CPU 中的程序，是整个程序设备的核心，可采用汇编语言和 C 语言设计，Cypress 为用户提供了一个固件程序框架，帮助用户完成相当一部分 USB 协议相关的工作，用户只需根据外设功能的具体要求，在相应的函数中填写函数体。Cypress 提供的这一框架程序简化了用户的固件程序设计。

本系统固件开发所使用的编程语言为德国 Keil 公司的 C51 编译器，集成开发环境为 μ Vision2，该固件共包含以下 5 个文件：

1) ez-usb.lib

是 EZ USB 函数库对象程序代码。

2) usbjmtb.obj

定义了 FX2 中 INT2 和 INT4 的中断跳转表。

3) dscr.a51

包含描述符的表数据，包括设备描述符、设置描述符、接口描述符、端点描述符和字符串描述符等，用来定义设备的 ID 号、端点的设置特性，主机

就是通过读取设备的描述符来获取 USB 外设的功能特性，从而给设备加载相应的驱动程序并配置设备。

4) fw.c

是固件运行的主程序文件，负责处理主机发出的各种 USB 设备请求，该程序首先初始化所有的内部状态变量，然后调用 TD_Init 函数进行初始化，并打开中断。之后，固件程序开始列举 USB 设备，直至在端点 0 上接收到 SETUP 令牌包时为止。一旦接收到 SETUP 令牌，其将重复执行下面的任务分配过程：调用用户函数 TD_Poll，以完成用户指定的任务；检测是否有 USB 设备请求（SETUP 令牌）如果有，则执行指令并作出相应的操作，如果没有，则继续向下执行，检测 USB 核是否有 USB 挂起事件，如果有 USB 挂起事件，则调用用户程序 TD_Suspend，交由用户处理；当 TD_Suspend 返回为真时，USB 核检测是否重新开始事件，如果没有 USB 挂起事件，则使处理器处于挂起状态，当检测有重新开始的事件，USB 核调用用户程序 TD_Resume，并继续执行本步骤，当 TD_Suspend 返回为假时，则继续执行本步骤。

5) periph.c

包含各种用户功能函数（如 TD_Init、TD_Poll 等）及其 INT2 中断服务子程序的定义，它实现了本系统的主要功能。

在本系统中，把端点 2 配置为批量传输、IN 方向，封包大小为 512 字节，每微帧包含 2 个封包，数据传输速率为 64Mbit/s 为就可满足实际需要，终端 0 作为默认的控制传输管道来处理 USB 事件。

3.2 驱动程序设计

FX2 的设备驱动程序有两种：一种用来在设备接入时从主机下载固件到 RAM 中，称为固件下载驱动程序（ezloader.sys）；另一种是在设备重新列举后加载的设备驱动程序（ezusb.sys）。应用软件通过该设备驱动程序与 FX2 通信。

固件的程序代码既可以通过外部 E2PROM 下载，也可以通过主机下载在本系统中选择从主机下载，此时外接 E2PROM 中存放的是固件下载的 PID/VID，且第 1 个字节是 0xC0H，系统通过该 PID/VID 找到固件下载驱动程序（ezloader.sys）将相应的固件代码下载到 FX2 的 RAM 中。ezloader.sys 的生成是利用 Cypress 公司提供的固件下载驱动程序源代码和用户编译成功的固件代码。在 Win2000 DDK 中创建的，并根据 E2PROM 提供的 PID/VID 修改相应的 .inf 文件，在将固件下载到 RAM 中后，系统清楚内存中的固件下载驱动程序，并进行重新列举，让 C8051 固件控制 FX2，此时，系统获得由 C8051 固件提供的 PID/VID（与 E2PROM 中的不同），认为有新的 USB 设备接入，并据此加载相应的设备驱动程序。

Ezusb.sys 是本系统运行时所使用的主要驱动程序，它定义了几个 IOCTL 代码，其中：IOCTL_Ezusb_GET_DEVICE_DESCRIPTOR 用于读取 USB 设备描述符；IOCTL_Ezusb_GET_CONFIGURATION_DESCRIPTOR 用于读取 USB 设备的配置信息，IOCTL_Ezusb_BULK_READ 用于主机读取 USB 设备发出的块数据，IOCTL_Ezusb_GET_CURRENT_FRAME_NUMBER，用于读取 USB 传输过程中当前的帧序号。

3.3 用户程序设计

应用程序开发中，可用 VC++ 编制应用程序。所有的用户程序是通过 I/O 控制调用来访问 EZ-USB 的设备驱动程序的，可以把 EZ-USB 设备当成文件来操作，利用 CreateFile 得到 EZ-USB 句柄，通过得到的句柄，用一个 Win32 函数 DeviceIoControl ()，把 IOCTL 和相关的输入输出缓冲区提交给驱动程序，来进行控制传输，用 ReadFile, WriteFile 进行批量传输。当应用软件退出时，用 CloseHandle () 关闭设备。

4 结束语

本文结合实际工程的需要，提出了基于 USB 接口的 VSAT 基带数据采集的设计方案，并给出了硬件系统及软件设计方法，在 VSAT 通信的实际应用中，该系统与后端软件结合，很容易实现对 VSAT 数据的分析与处理，可广泛应用于 VSAT 通信系统中。