

基于 Linux 平台的天气雷达高速数据采集系统设计

王 江, 何建新, 李学华
(成都信息工程学院, 四川 成都 610225)

摘 要: 分析了基于 PCI 总线的天气雷达高速数据采集系统工作原理和实现方案, 实现了基于 Linux 平台的天气雷达高速数据采集系统的驱动、数据采集及处理的设计。该设计已成功应用于天气雷达中。

关键词: PCI 总线; 天气雷达; 数据采集; Linux 平台

中图分类号: TN957.52

文献标识码: A

Design of high speed weather radar data acquisition system based on Linux platform

WANG Jiang, HE Jian Xin, LI Xue Hua
(Chengdu College of Information Engineering, Chengdu 610225, China)

Abstract: This paper analyses working principle and realizing methods of high speed weather radar data acquisition system based on PCI bus, completes driver design, data acquisition and processing based on Linux platform. The design has been successfully used in weather radar.

Key words: PCI bus; weather radar; data acquisition; Linux platform

Linux 是开放源代码、网络化的操作系统, 具有稳定、高效、内核可自由配置等特点。采用 Linux 操作系统作为开发平台与采用 VxWorks 和 Windows 作为开发平台相比不仅有免费的优势, 而且对于发展核心技术, 提高信息安全有着极其重要的意义, 是自主研发的一个方向。随着我国天气雷达的快速发展, 天气雷达在民用和军事领域的应用越来越广泛。目前, 天气雷达高速数据采集、传输等功能大都基于 Windows 平台进行开发实现, 实现由 Windows 平台向 Linux 平台的转换, 对于发展中国自主知识产权气象软件核心技术, 提高信息安全有着极其重要的意义。本文基于 Linux 操作系统(2.6.23 内核), 实现了天气雷达高速数据采集及处理, 对天气雷达系统由 Windows 平台向 Linux 平台移植具有参考价值。

1 数据采集系统分析

1.1 工作原理

天气雷达高速数据采集系统由 I/Q 数据采集、FIFO 缓冲、FPGA 控制模块及 PCI 总线芯片 PLX9054 组成, 系统结构框图如图 1 所示。

其核心部分为 FPGA 控制模块, 主要实现 4 个功能: (1) 根据雷达量程和距离分辨率调整 AD 采样时钟的频率; (2) 控制 AD 两路采样数据写入 FIFO 缓冲以及 DMA 传输时从 FIFO 中连续读出数据; (3) 与 PCI 总线控制芯

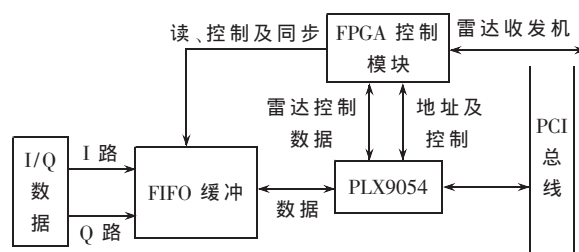


图 1 天气雷达高速数据采集系统结构框图

片 PLX9054 进行控制交互, 实现局部总线的读写控制、地址译码和中断触发; (4) 与 PLX9054 进行数据交互, 使用 FPGA 内部 RAM 保存控制天气雷达发射机和接收机工作的命令及数据, 完成对雷达工作的控制。

I/Q 数据采集输出数据速度为 2.4 Mb/s(I、Q 数据各为 24 bit), 通过 FPGA 控制, 合并写入到 FIFO 缓冲。FIFO 缓冲为 32 bit、深度为 16 KB 的数据缓存, 有效位为 24 bit; 与计算机总线的接口采用 32 bit 的 PCI 总线接口芯片 PLX9054, PC 机通过它完成与数据采集系统的数据交互。

1.2 PC 机与采集系统数据访问

1.2.1 局部地址分配

PC 机与天气雷达高速数据采集系统的数据访问通过 PCI 总线芯片 PLX9054 实现, 访问目标为 FIFO 缓冲和 FPGA 片内 RAM。对 FIFO 缓冲和 FPGA 片内 RAM 分

配不同局部地址,通过对局部地址总线进行译码确定需要访问的目标。FIFO 数据采用 DMA 传输方式连续读出,每次传输长度最大为 8 KB,因此,设置 FIFO 空间 32 bit 局部地址范围为 0x20000000~0x2000FFFF,映射空间为 PCI 地址空间 bar2。

FPGA 片内 RAM 位宽为 32 bit,采用局部地址范围为 0x40000000~0x400FFFFF。映射空间为 PCI 地址空间 bar3。

1.2.2 数据访问方式

FIFO 缓冲为可编程 FIFO,由 FPGA 根据片内存储的雷达参数进行控制。当 FIFO 中数据达到设置的大小时,触发 PCIPLX9054 引发 PCI 中断,通过该中断向驱动程序申请发起 DMA 传输。驱动程序响应该中断后设置 PCIPLX9054 的 DMA 寄存器,发起方向从局部总线到 PCI 总线的 DMA 传输,完成 FIFO 数据向 PC 机的传输。

FPGA 片内 RAM 通过 PCI 地址映射,使用 IO 操作方式进行数据读写访问。

2 Linux 程序实现

天气雷达高速数据采集系统基于 Linux2.6 内核平台程序,由基于 PCI 总线的设备驱动程序、数据采集及处理程序组成。

2.1 设备驱动程序实现

设备驱动程序的功能是在 Linux2.6 系统平台下,通过对 PCI 总线控制芯片 PLX9054 的控制,实现 PC 机与采集板的数据交互,实现雷达采集数据的实时接收及雷达工作状态的控制。从功能结构上,设备驱动主要由设备装载及初始化、提供给用户层的接口函数、中断及 DMA 数据传输三部分组成。

2.1.1 设备装载及卸载

设备装载和卸载即是 Windows 系统中设备驱动安装和卸载。在 Linux2.6 中,装载和卸载主要通过调用系统提供的 PCI 总线驱动注册函数 pci_register_driver 和注销函数 pci_unregister_driver 来实现,其注册的 driver 数据结构如下:

```
static struct pci_driver plx9054_pci_driver = {
    .name = "pci9054",
    .id_table = plx9054_pci_tbl,
    .probe = plx9054_probe,
    .remove = plx9054_remove,
};
```

当系统引导时,高速数据采集系统板上电后,板上 PCI 总线固件对 PLX9054 设备进行缺省配置,为设备的每个地址区域分配好资源。装载函数在驱动中用 module_init 声明,进行设备的初始化工作,调用 PCI 总线驱动注册函数,根据 id_table 定义的信息对 PCI 设备 PLX9054 进行探测识别,完成 PLX9054 设备驱动的 PCI 总线注册,然后调用驱动注册的 probe 函数,完成 PLX9054 设备使能、PCI 地址空间映射、PLX9054 设备数据结构初始化、设备注册(注册为字符设备)等操作。卸载函数用

module_exit 声明,是装载的逆过程,通过调用 PCI 总线驱动卸载函数,继而调用注册的 remove 函数、注销设备及 PCI 地址空间映射、释放分配给设备的数据结构空间等操作,完成系统资源释放。

2.1.2 设备接口函数

设备接口函数为用户提供与设备进行交互的不同功能接口,主要完成用户与设备的读、写及设备控制访问等功能,其数据结构设计为:

```
static const struct file_operations plx9054_fops = {
    .owner = THIS_MODULE,
    .open = plx9054_open, /* 打开设备 */
    .release = plx9054_close, /* 释放设备 */
    .ioctl = plx9054_ioctl, /* IO 操作 */
    .read = plx9054_read, /* 读设备 */
};
```

这个数据结构在设备装载过程中,作为设备注册的参数完成与设备挂接、使用户在系统调用中能访问到上述关联的设备接口函数。

设备打开函数 plx9054_open 完成以下操作:设备私有数据结构内存空间分配及初始化,DMA 连续物理内存空间申请,为天气雷达最大距离探测采集数据的大小;调用 pci_map_single 将分配的 DMA 内存空间映射为用于 DMA 传输地址;DMA 数据缓冲区分配(容量为 2 MB),缓冲区保存每次 DMA 传输中获取的数据;调用 request_irq 对中断函数 p9054_interrupt 进行中断号申请;对天气雷达工作参数进行缺省配置,启动雷达工作。设备释放函数 PLX9054_close 完成的操作与设备打开函数与 PLX9054_open 相反。IO 操作函数 PLX9054_ioctl 通过 sys_ioctl 系统调用,完成用户与设备的信息查询、雷达工作参数配置与查询,如驱动软件版本号、雷达当前工作各参数配置和查询,主要用于对 FPGA 中 RAM 内存空间的雷达工作参数数据访问。由于这部分数据对访问速度要求不高,使用 IO 操作能很好地满足设计要求。读设备函数 PLX9054_read 将驱动 DMA 数据缓冲区中的数据实时读取到用户空间缓冲区内,完成采集数据从设备到用户的读取操作。

2.1.3 中断处理及 DMA 传输

中断处理及 DMA 传输是驱动的核心部分,它的设计质量直接影响到雷达采集数据能否正确、实时地从采集系统板传送到设备驱动缓冲区,并实时被用户获取。中断处理工作流程如图 2 所示。

由于 DMA 数据缓冲区远大于每次 DMA 数据传输大小,这对于雷达数据采集中因用户读取数据延迟而导致的缓冲区溢出有较好的抑制作用,而相对于现有计算机平台,2 MB 内存容量已足够。DMA 数据缓冲区设计为循环缓冲区,使用读、写指针维护,在中断处理中更新其写指针,在 PLX9054_read 中更新其读指针,采用溢出则覆盖的机制,既方便管理,节省资源,同时又可对溢出进

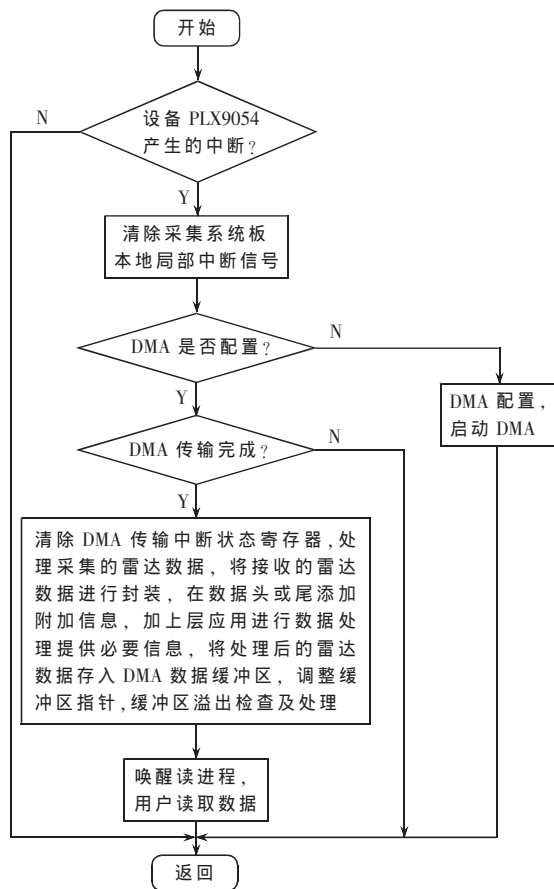


图2 中断处理流程

行计数,很方便进行实时性能的测试。

2.2 数据采集处理程序实现

数据采集处理程序是雷达高速数据采集系统的应用处理程序,完成雷达高速数据采集及对采集数据的数字信号处理、输出天气雷达生成一次产品。其主要流程如图3所示。

在采集 I/Q 数据时, I/Q 两路数据采用交叉采集方

式存入 FIFO 缓冲中,每路数据有效位为 23,最高位为符号位。因此,在采集数据处理中需对读取的数据进行 I/Q 数据分离,即将采集数据恢复为原始 I、Q 两路数据,同时将 24 bit 数据转换为 32 bit 数据方式存储,以满足后面数字信号处理要求。

在软件设计中,不同数字信号处理功能采用不同数字信号处理函数实现,在采集数据获取前进行选择配置,数字信号处理功能增加可通过增加相关的数字信号处理函数来满足,以方便功能的完善和移植。通过

对 I/Q 数据分离后获得的数据进行相关数字信号处理即可获得天气雷达中的一次产品,如功率和反射率等。

本文基于 Linux2.6 平台,设计实现了天气雷达高速数据采集系统的驱动和数据采集及对采集数据的数字信号处理,并在现有天气雷达上完成了调试验证。目前,本设计已成功地应用在现有的天气雷达中。

参考文献

- [1] 何建新.现代天气雷达[M].成都:电子科技大学出版社, 2004.
- [2] CORBET J, RUBINI A. Linux 设备驱动程序[M]. 3 版. 北京:中国电力出版社, 2006.
- [3] PLX Technology, Inc. PCI9054 data book version 1.0[R]. 1998. (收稿日期:2008-10-28)

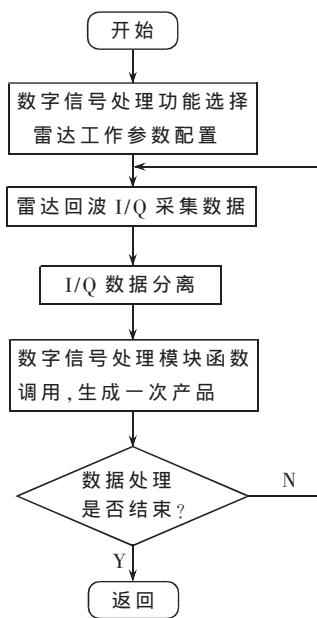


图3 采集数据处理流程

(上接第 122 页)

表4 防伪读写器作为发送方 PDU 格式

SCA	PDU 类型	MR	DA	PID	DCS	VP	UDL	UD
-----	--------	----	----	-----	-----	----	-----	----

表5 防伪读写器作为接收方 PDU 格式

SCA	PDU 类型	OA	PID	DCS	SCTS	UDL	UD
-----	--------	----	-----	-----	------	-----	----

消费者选择输入手机号时,防伪读写器在用户信息(UD)中将同时发送商品 ID 号和消费者手机号,如果没有输入,则只发送商品 ID 号。

本文给出了基于 RFID 和 GSM 通信的防伪读写器的设计与实现,以酒类防伪和家电类防伪为模型,对系统进行了实际测试,运行良好,能够实时准确地反馈验证结果给消费者,操作简单,显示直观。该系统能够满足大部分贵重商品的防伪需求,特别是通过后台服务器验证的防

《电子技术应用》2009 年第 5 期

伪机制,具有技术难度高难以仿造的特点。该防伪机制为电子防伪提供了新的途径,具有广阔的应用前景。

参考文献

- [1] 谭民,刘禹,曾隽芳,等.RFID 技术系统工程及应用指南[M].北京:机械工业出版社,2007.
- [2] 李龙,杨莉.射频 IC 卡在酒瓶防伪装置中的应用[J]. 自动化技术与应用,2007,26(8):113-114.
- [3] 3Logics.TRH031M 13.56MHz multi-protocol reader IC datasheet[Z]. 2005.
- [4] Wavcom. WISMO quik Q2400 series Q2403A product specification[G]. 2002.
- [5] ISO/IEC 15693[S]. Identification cards—contactless integrated circuit(s) cards—vicinity cards. ISO/IEC, 2000. (收稿日期:2008-12-07)