文章编号: 1000-8349(2006)02-0160-07

研究论文

用 PCI 中断实现射电望远镜跟踪的方法

王锦清, 范庆元
(中国科学院 上海天文台, 上海 200030)

摘要: 讨论运用 PCI 9054 (美国 PLX 公司生产的接口芯片) 作为接口芯片的 PCI (Peripheral Component Interconnect) 板卡的软硬件设计, 以实现天线跟踪的两个时间同步中断。利用标准秒信号中断作为系统时钟同步信号, 并同步产生时间间隔为 20 ms (或 40、50 ms , 可选) 的中断信号, 来处理天线跟踪指令输出。中断信号通过 PCI 中断口 INTA# 接入计算机, 在驱动中识别不同的中断信号, 并在应用程序响应中断处理后, 命令 ACU (Antenna Control Unit) 机, 实现射电天文望远镜的同步跟踪。其控制过程分 3 部分阐述: 硬件设计、驱动程序设计、安装及应用; 着重讨论了前两者的设计方法及思路。

关 键 词: 天文观测设备与技术; PCI 中断; 同步; 驱动程序; PCI 9054 ; 秒信号
中图分类号: P111.44 **文献标识码**: A

1 引 言

目前国内射电天文望远镜 (天线) 的同步跟踪方式各不相同, 最为普遍的是在编程中以独占轮询方式读取分频钟作为时间信息, 用于执行天线跟踪。在这样的跟踪方式中, 控制过程的不等间隔性将在位置环控制中引入不可避免的跟踪误差。当然, 也有以软件中断方式产生等间隔命令来执行天线跟踪输出的, 它的中断方式因为能更合理地保证相对实时性能而优于轮询方式。目前秒信号的接入方式基本上都以串口、并口或者 ISA (Industry Standard Architecture) 中断线接入为主, 天线的控制平台只能低于 Windows98 , 有的还在 DOS 状态下, 它们都以独占方式访问硬件, 没有驱动程序的过渡, 从而加大了系统运行的危险性。随着计算机技术的不断发展, 天线控制平台也迫切需要升级换代。

射电天文望远镜观测需对一个目标进行一定时间的跟踪定位, 其时间同步取自氢钟秒信号。为了保证定位精度, 通常每间隔几十毫秒就调整一次天线的位置。调整天线的时间间隔越

收稿日期: 2005-04-29 ; **修回日期**: 2005-09-28

短，定位精度越高。考虑传动机械的响应速度等因素，在保证精度的前提下，我们以 20 ms (或 40、50 ms ，可选) 的等间隔中断控制天线跟踪。

PCI (Peripheral Component Interconnect) 总线是对过去总线的继承和发展，目前还没有一种总线可以从根本上取代它的地位。由于本身的优势， PCI 在市场中占有主流地位。它的扩展是独立于微处理器的，所以在利用未来 CPU 以及多 CPU 时，可以同样用 PCI 扩展卡。

我们的目标是开发一块 PCI 板卡，利用其中断方式，实现天线跟踪控制的时间同步，以提高天线跟踪控制系统的位置环路精度。当然使用 PCI 接口还有其他多方面的考虑，例如：板卡还将完成 A/D 采样功能以及外部串、并行数据的输入功能。鉴于 PCI 板卡与普通板卡相比并无特别之处，以下只介绍如何利用它来实现射电望远镜的同步跟踪。

2　硬件设计

我们采用 PCI 9054 (美国 PLX 公司生产的接口芯片) 作为 PCI 接口芯片，它桥接于 PCI 总线和本地总线之间，这样只要对 PCI 9054 本地总线进行操作就可以了， PCI 9054 在 PCI 端自动“解译”本地端总线的时序，从而简化了硬件接口的设计。

另外，我们用 Xinlinx 公司的 CPLD9572 芯片实现逻辑和时序，由 1 MHz 晶振分频产生数十毫秒 (20 、 40 、 50 ms 任选) 的均匀时间间隔。然而，由于在 Windows2000 下多线程的定时时间颗粒为 10 ms，它能引起操作系统本身的非实时性，天线转动时机械响应速度也存在误差，即使硬件板卡上的时间精度能达到纳秒级，在应用层仍无济于事。在晶振工作频率选用 1 MHz 的条件下，外部秒信号引起的硬件中断精度可达到 1 μm 。

硬件的总体框图如图 1 所示。

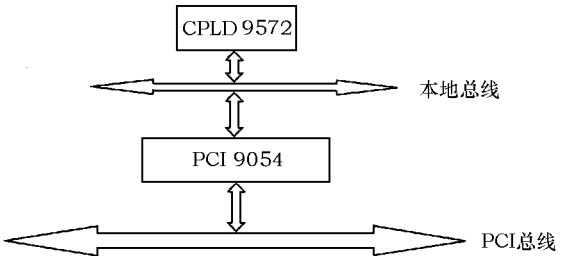

图 1　硬件的总体框图

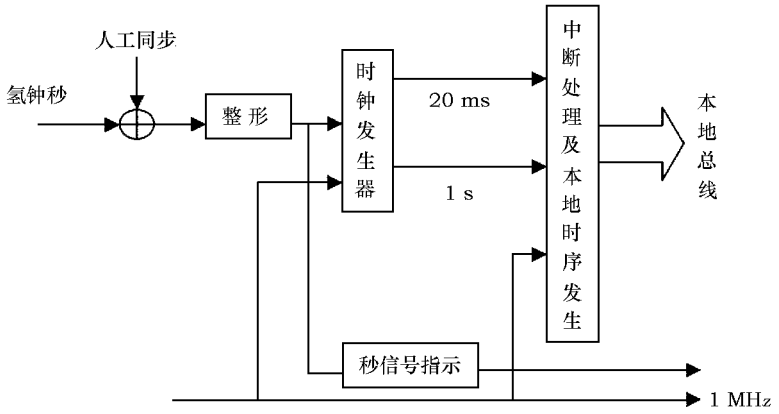

图 2　CPLD 9572 功能框图

至于 CPLD 9572 的内部逻辑可用 VHDL (Very High Speed Integrated Circuit Hardware Description Language) 描述 [1] ，其功能概述如图 2 所示。其中，整形模块主要是对长距离传输的氢钟秒脉冲进行整形，以提高时钟比对的可靠性。氢钟秒信号和人工同步秒信号进行逻辑 “或” 操作，在没有标准秒信号场合实现人工同步。当外部秒脉冲存在时，时钟发生器模块同步产生 20 ms (40 、 50 ms 可选) 的毫秒信号。秒信号指示模块用 200 ms 闪烁表明标准秒信号的存在，否则，同步秒信号不存在，随机产生 1 s 和 20 ms 信号。 “中断过程处理及本地时序发生” 模块实现 PC 机的硬件实时中断，即对桥接芯片 PCL 9054 的本地端总线进行操作。

3 驱动程序设计

目前开发 WDM (Windows Driver Module) 驱动程序的常用工具主要有以下几种：微软的 DDK (Device Driver Kit) 、 Numega 公司的 DriverWorks 、 Jungo 公司的 WinDriver ，后两者是把 DDK 做了一些封装，开发速度比较快，因为它们都有向导完成总体程序框架，该框架已有基本的即插即用、电源管理等功能，只要定义部分代码就可以使用。由于硬件本身的工作比较单一，需要与操作系统接触的层面不是很宽，而且效率优先，所以我们还是采用了 DDK 。它的结构清晰，直接面对底层硬件，用 SoftIce 调试也比较方便。

PCI 总线支持硬件资源动态自动配置，以支持即插即用。在 PCI 设备插入 PCI 插槽或上电后， PCI 总线配置机构自动根据 PCI 设备的要求实现配置。 PCI 总线支持内存读写、 I/O 端口读写、中断机制和 DMA (Direct Memory Access) 功能。由于这些硬件特点， PCI 设备的 WDM 驱动程序的设计比较复杂。

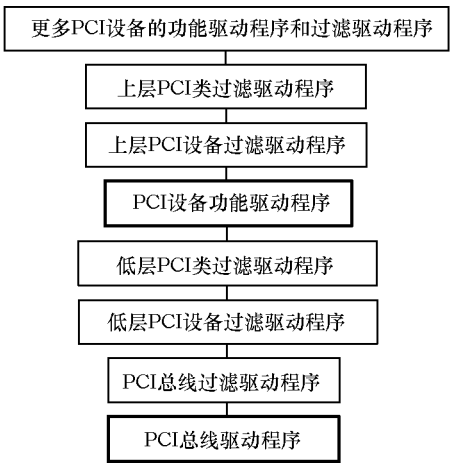

图 3 PCI 总线的分层驱动程序结构

在 WDM 中，我们采用了分层的驱动程序体系结构：总线驱动程序或类驱动程序在最底层直接与设备联系；设备功能驱动程序在上层通过与低层驱动程序连接，实现设备的功能；类过滤驱动程序或设备过滤驱动程序在中间层，用于数据的过滤或转换。具体结构如图 3 所示 [2] 。

实际开发中， PCI 设备的 WDM 驱动程序一般只要开发一个设备功能驱动程序即可。设备功能驱动程序直接与 PCI 总线驱动程序联系，进行硬件操作，以实现 PCI 设备的功能。 PCI 总线驱动程序由操作系统实现 [3] ，过滤驱动程序一般在特殊情况下需要编写。这里我们只讨论 PCI 设备功能驱动程序的设计。

PCI 设备驱动程序在框架上与其他类型的设备驱动程序基本相同，它包括初始化、创建设备、卸载和删除设备、即插即用处理、分发例程处理、电源管理、 WMI (Windows Management Instrumentation) 等部分 [4] ，限于篇幅，在此只讨论本设计中 PCI 设备的特别之处，主要由以下四方面构成：

(1) 获得设备资源

PCI 设备的硬件资源由 PCI 配置机构动态分配。 PCI 设备实现 PCI 配置寄存器，并提出需要分配的硬件资源，由 PCI 配置机构分配资源。驱动程序需要取得这些资源，才能操作硬件。因此， PCI 设备的硬件资源分配与管理是驱动程序中很重要的部分。硬件资源主要包括映射内存空间、 I/O 空间、中断 [2] 。在 WDM 体系中， PCI 总线驱动程序实现了对 PCI 设备资源的枚举，设备驱动程序通过向 PCI 总线驱动程序传递设备配置 IRP_MJ_PNP (一种消息请求包) [5] ，经总线驱动程序处理后，设备驱动程序得到 PCI 设备的资源信息。系统的 PNP (即插即用) 管理器在取得设备资源后，自动向驱动程序发出 IRP_MN_START_DEVICE (一种消息请求包)[5] 的 IRP (I/O 请求包) ，在该 IRP 栈中包含了设备的资源信息。每个支持 PNP 功能的驱动程序，都应实现 IRP_MN_START_DEVICE 处理。该处理应先交给低层驱动程序处理，然后再根据 IRP 栈内内容进行资源分配 [4] 。调用相应函数进行资源分配时一定需要有错误返回处理，如果不符合要求，立即退出，否则在其他例程中会发生错误，使系统崩溃。同时，在退出系统前，一定要释放已分配的资源。

(2) 内存处理

Windows 工作在保护模式下，它与实模式的区别在于 CPU 寻址方式不同，可以实现虚拟内存。 Windows 系统把其内存分为分页和非分页内存。分页内存一般用于应用程序，系统提供分页和分段，使用户应用程序使用的内存可以在程序空闲时由系统将其从物理内存调配到硬盘中，以节省物理内存资源，当程序重新运行时，再由系统将其调配到物理内存，这样，系统可以得到比物理内存大得多的内存量，允许更多的应用程序保持运行。而非分页内存为系统常驻内存，不可以从物理内存调配到硬盘上。在 WDM 驱动程序中，对于硬件的内存映射一般需要用非分页内存，因为一些运行在 DISPATCH_LEVEL 或更高中断级的例程禁止使用分页内存。 DISPATCH_LEVEI 是一种设备的中断优先级别，驱动程序共使用 3 种中断优先级别，从低到高依次为 PASSIVE_LEVEL 、 DISPATCH_LEVEL 和 DIRQL[2] 。

PCI 设备的驱动程序获得的设备内存是一段映射物理内存，它无法直接使用，需要将其映射成系统可以访问的非分页内存。函数 MmMapIoSpace 可完成该功能。

(3) 中断处理

PCI 总线的每个槽共享一个中断 INTA# ，因此在中断处理时必须十分小心，若处理不当，就会导致系统崩溃。驱动程序首先要在 IRP_MN_START_DEVICE 中获得中断资源，然后将其连接到中断处理函数中，使其当有中断请求时，能进入中断服务例程。连接中断的函数为 IoConnectInterrupt 。在中断服务例程中，根据硬件信息首先判断该中断是否是该设备发出的申请。系统在接收到中断后，顺序调用各个注册了该中断资源的驱动程序的中断处理例程，如果有返回 TRUE 的例程，就代表该中断已处理，不用再调用其他例程；如果中断没有处理则返回 FALSE ，继续调用其他例程。如果最终返回错误，则系统被扰乱，造成崩溃。

在中断服务例程中，相应的处理最好简洁快速，因为中断例程运行的级别很高，当有中断请求时，不但应用程序的执行被打断，而且在硬件中断级以下的所有程序的运行均被打断。 WDM 提供了 DPC (Deferred Procedure Call) 例程，该例程是在中断例程中耗时但不需要立即处理的任务延时处理。比如，驱动程序接受应用程序写 PCI 设备数据，当它写完后，硬件产生的中断标志执行完毕，需要结束该 IRP ，这时就可以将结束 IRP 这个耗时的任务交给 DPC 完成。

(4) DoorBell 寄存器中断功能的实现

DoorBell 寄存器即门铃寄存器，它是 PCI 9054 芯片内部的一种寄存器。在中断服务例程中，先读取中断状态寄存器，再判断 PCI 9054 的中断允许位是否置位，如果置位 (即由本地端总线写了 DoorBell 寄存器，这可通过上述的 CPLD 9572 实现)，就读 DoorBell 寄存器，判断是不是和本地端总线写入的相同，如果相同，就在设备扩展结构中保存该值 (如果不同，说明是 PCI 9054 的其他中断)，然后用对该 DoorBell 寄存器写入相同的值， DoorBell 寄存器由于被本地端总线和 PCI 端总线相同的值写了两次 (第二次是由驱动程序通过 PCI 总线写的)，就自动复 “0”(这是 DoorBell 寄存器的基本特征)。然后用 KeInsertQueueDpc 函数调用所定义的 DPC 例程， DPC 例程用事件通知应用程序发生中断。

对于驱动程序中其他的细节问题，这里就不一一展开了。

4 安装及应用程序

在安装 PCI 板卡时，可以用 Windows2000 (或 Windows XP) 自带的硬件安装向导加载驱动程序，在板卡插上后系统会提醒有新硬件存在并要求提供驱动程序，只要把上述驱动程序的路径告诉操作系统，即可实现安装。用 Visual C++ 建立应用程序，在初始化阶段就建立与 PCI 板卡的连接 [6]，开设监听线程或消息函数，以接收驱动程序发来的中断信号，并在该线程或消息函数中计算新的时间值，这样，控制程序的时间系统就由外部硬件来决定。图 4 是

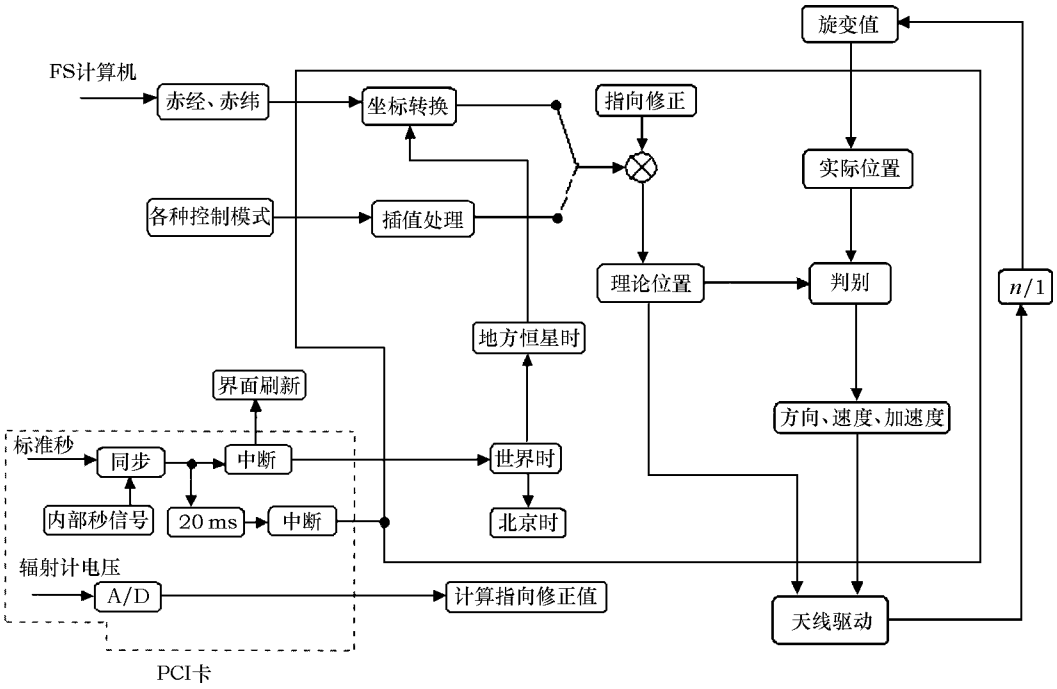

图 4 射电望远镜控制框图

射电望远镜的控制框图 (左下虚线中为 PCI 卡功能框图， 20 ms 中断线与右边框图相连表示每 20 ms 右边框图中所有程序都执行一次) 。

该控制的主要功能如下：

(1) 每 20 ms 计算各种时间 (这 20 ms 由前面所述的硬件产生) ；

(2) 通过网络接受 FS (Fields System ，国际上通用的天文观测软件) 计算机 (另一台控制用的计算机) 的控制，或读取本地计算机上的各种轨道文件，并作不同的插值处理；

(3) 每 20 ms 读取射电望远镜当前状态参数，并与理论值 (这个值对应于当前时间) 相比，然后折算到射电望远镜的转向、方位、俯仰，再叠加指向修正等参数；

(4) 一方面，与监管程序通信，把当前射电望远镜的各种参数通过网络送到远程的另一台监管计算机。另一方面，与 GPS 通信 (图 4 中没有给出) ， GPS 可以提供起始时间；

(5) 对整个界面进行有序维护，对天线的方位和俯仰进行了模型仿真，确保天线的安全运行。

当前控制程序的界面如图 5 所示。

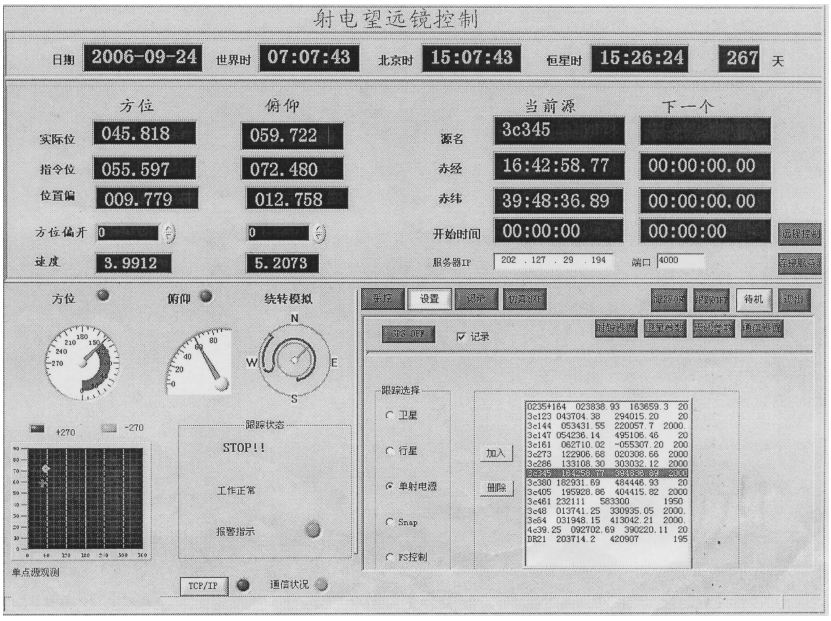

图 5　射电望远镜控制程序界面

目前，在上海和乌鲁木齐两台站已使用上述程序及硬件来观测，并都以氢钟秒作同步信号。直观的控制界面使观测的自动化程度大大提高。与原来 DOS 和 Windows98 下的控制程序相比，新的控制程序可以说是取其精华，去其糟粕，既继承了老程序中很多优秀的方面，又大胆革新，改善了控制性能及其直观性。

参考文献:

[1] 姜立东. VHDL 语言设计及应用, 北京: 北京邮电大学出版社, 2001
[2] Walter O. Programming the Windows Model Driver, Seattle: Microsoft Press, 1999
[3] 张惠娟, 周利华, 翟鸿鸣. Windows 环境下的设备驱动程序设计, 西安: 西安电子科技大学出版社, 2003
[4] Chris C. Windows WDM 设备驱动程序开发指南, 孙义, 马莉波, 国雪飞等译, 北京: 机械工业出版社, 2000
[5] Windows 2000 DDK Documents, Seattle: Microsoft Press, 1999
[6] 官章全, 刘加明. Visual C++6.0 库类大全, 北京: 电子工业出版社, 1999

The Application of PCI Interrupt in the Control of Radio Telescope

WANG Jin-qing, FAN Qing-yuan

(*Shanghai Astronomical Observatory, Chinese Academy of Sciences, Shanghai 200030, China*)

Abstract: This paper describes hardware and software designs for a PCI board based on PCI bridge chip PCI 9054. The board is used to generate two clocks interrupt to realize the synchronous tracking of the radio telescope. The standard second signal generates the 20 ms (40 ms or 50 ms) clock interval interrupt which is used to send the tracking commands. The two clock interrupts input to the computer via PCI interrupt interface INTA# can be distinguished from the different interrupt codes in driver programme. The application programme will send commands to the antenna interface to implement the tracking of the radio telescope after it responses the interrupts. The control process will be described through three aspects. First is the hardware design. Second is the driver programme design, and the third is the application programme design. The first and second aspects are described in details.

Key words: astronomical facilities and technique; PCI interrupt; synchronize; driver programme; PCI 9054 chip; second signal