

基于 RTX 的实时处理系统设计

张前福^{1,2}, 王华闯¹, 许俊平¹, 杨建军³

(1 中科院光电技术研究所, 四川 成都 610209; 2 中科院研究生院, 北京 100039;

3 中国人民解放军 93617 部队, 北京 101400)

摘 要: 在图像视频实时处理领域, 针对传统的专用处理系统开发难度大、开发周期长的缺点, 提出了一种 RTX (Real-Time Extension For Windows) 加 Windows 的通用系统架构. 采用了 PCI 作为传输总线, 使用 DMA 加中断的数据传输模式, 保证了数据的实时传输性能; 处理算法工作在 RTX 下, 设计了 RTX 和 Windows 的进程间通信功能, 保证了系统处理的实时性能同时还减轻了开发的难度, 缩短了系统的开发周期. 最后通过实验进一步证明了把视频图像的实时处理任务移植到该系统下是可行的.

关键词: Windows 实时扩展; PCI 实时驱动; DMA; 进程间通信

中图分类号: TP393

文献标识码: A

文章编号: 1000-7180(2012)08-0058-04

Real-Time Processing Program Design Based on RTX

ZHANG Qian-fu^{1,2}, WANG Hua-chuang¹, XU Jun-ping¹, YANG Jian-jun³

(1 Institute of Optics and Electronics, Chinese Academy of Sciences, Chengdu 610209, China;

2 Graduate School of the Chinese Academy of Sciences, Beijing 100039, China;

3 The Unit 93617 of PLA, Beijing 101400, China)

Abstract: This paper designed a new system for real-time video processing by using RTX and Windows, solving the difficult development and long development cycle problems which exist in the routine embedded special processing platform. To achieve the real-time transfer performance, we use the PCI-Bus working in the DMA and interrupt mode; meantime, algorithms perform in the RTX environment which supplies the real-time processing ability. Since the RTX works together with Windows system, the proposed framework can greatly augment the power for human-machine interaction. Furthermore, experimental results show that it is feasible to transfer the real-time video processing tasks to this system from the routine embedded system.

Key words: real-time extension for Windows; PCI real-time driver; DMA; inter-process communication

1 引言

随着人们对图像视频领域的处理效果的要求越来越高, 关于视频实时处理方面的算法也越来越复杂. 特别是在跟踪和检测识别领域, 大部分的算法都涉及到连续帧的检测问题, 需要预先对图像进行多帧的缓存, 传统的做法是采用 FPGA 加 DSP 的专用处理架构. 这样做虽然能够达到实时处理的目的, 但是作为一个专用的系统架构, 其开发周期比较长, 不

仅仅需要专业的人士进行开发维护, 同时其适用面也比较窄, 不利于相似项目的开发移植, 一旦更换应用场景, 整个平台都会面临着重新开发的问题.

随着 Windows 的普及, 人们开始积极寻求在 Windows 平台上进行实时任务的处理. Windows 有着丰富的库资源, 在图像视频处理显示领域将有着巨大的优势.

但是 Windows 本身不是一个实时操作系统, 而是一个通用操作系统, 其线程和中断的优先级太

少,决定了它任务完成的不确定性,并且它存在着优先级倒置的情况,特别是在中断处理过程中,大部分的中断处理工作都会放在 DPC(延迟过程调用例程)中进行,但是 DPC 的执行是按照先后的队列形式执行的,所以会使得优先级高的处理程序反而后执行。

实时操作系统有 DOS 和 VxWorks, DOS 操作系统人机交互界面太差,对于设计界面应用程序很难; VxWorks 图形处理能力太弱,软件开发周期长,并且价格比较昂贵。为了利用 Windows 强大的图形处理能力和普及力度,本文提出了一种基于 RTX 环境的数据传输处理系统,采用装有 RTX 和 Windows 的计算机实现视频图像的实时处理系统,满足了视频图像实时处理的需求。

2 RTX 架构及其实时性能分析

2.1 RTX 架构

RTX(Windows 实时扩展)是美国 IntervalZero 公司开发的基于 Windows 的实时解决方案,是目前 Windows 平台的唯一纯软件的硬实时扩展子系统。RTX 系统的架构^[1]如图 1 所示。RTX 不对 Windows 系统进行任何封装或修改,其通过在 HAL(Hardware Abstraction Layer)增加实时扩展来实现基于优先级的抢占式的实时任务的管理和调度。RTX 提供了对微软 Visual Studio 系列开发平台的全面支持,支持 Windows 下的大部分 API 函数,使得开发人员可以最大程度的减少开发周期和成本,同时支持对 I/O、内存等硬件资源的直接访问,方便

了驱动程序等的开发^[2]。RTX 系统框架如图 1 所示。

2.2 RTX 实时性能分析

对于一个实时操作系统,人们最关心的是它的线程调度能力、中断处理能力以及定时器的性能。

在线程调度方面,RTX 提供了自己的线程调度器,拥有 128 个线程优先级,任务调度器总是运行最高优先级的任务,直到有更高优先级的线程就绪或者自己的时间片用完而同等优先级的线程就绪才会中断当前任务,并且通过对硬件抽象层的实时扩展,使得当 RTX 下有任务执行时能够屏蔽 Windows 的任务和中断。RTX 下任务调度器的任务切换时间很短,是 Windows 的几十甚至几百分之一^[3]。表 1 所示为 RTX 和 Windows 的线程切换时间对比。

表 1 RTX 和 Windows 线程切换时间

系统类型	最小用时/ μs	最大用时/ μs	平均用时/ μs
RTX	0.5	1.8	0.7
Windows	0.7	117	2.0

通过硬件抽象扩展层,RTX 可以对 PIC(可编程中断控制器)或者 APIC(高级可编程中断控制器)进行重新编程,从而达到中断隔离的目的,即 RTX 的中断可以中断 Windows,当 RTX 任务运行的时候屏蔽掉 Windows 中断。Windows 下的定时器的最小周期为 1 ms,但由于时间片切换等的影响,常常为数毫秒以上,无法满足实时系统的要求; RTX 提供了更为精确的时钟和定时机制,其定时分辨率达到 100 ns,最小定时周期为 100 μs ^[4]。

在图像视频处理领域中多数算法都涉及到图像像素的遍历情况,为了体现 RTX 的实时性能,本文对 1024 * 1024 的图像遍历 100 次,来测试其执行的确定性能。使用的主机参数为:CPU 为 Celeron(R),主频为 2.66 GHz,内存为 1.5 GB。测试的结果如表 2 所示。

表 2 遍历时间测试结果

系统类型	有跳转/ μs	无跳转/ μs
RTX	371	280
Windows	1 937	375

其中有跳转指的是 for 循环最内层中有 if/else 的判断,无跳转指的是 for 循环的最内层只是简单的加运算。为了分析其执行的确定性,对源代码进行

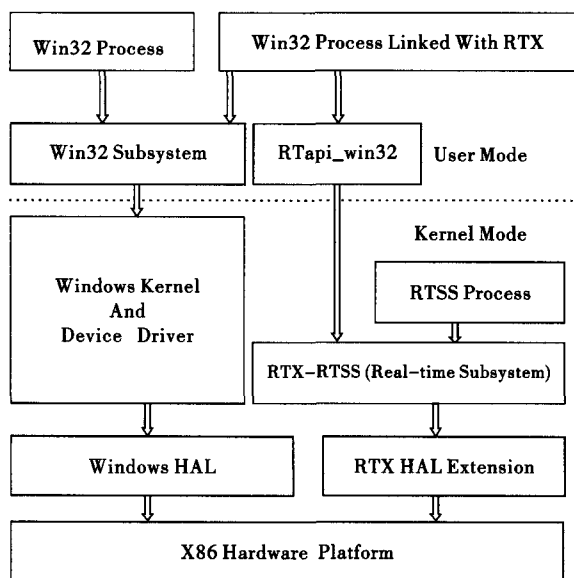


图 1 RTX 系统架构

了反汇编,然后对 CPU 的指令分析,发现 RTX 下对源代码的执行是确定性的,而 Windows 下的执行时不确定性的.由此可以看出 RTX 在实时处理方面的性能是非常优异的.

3 基于 RTX 的实时处理程序构架

3.1 系统总体数据流程

系统设计要求能够向用户展示实验结果,而外部的数据采集在板卡上进行,数据的实时处理在 RTX 下完成.所以从数据的最开始采集到通过 GUI 展示要经过三个层次的通信:采集卡和外设的通信、采集卡和 RTX 系统的通信、RTX 系统和 Windows 系统之间的通信.具体的数据流程如图 2 所示.

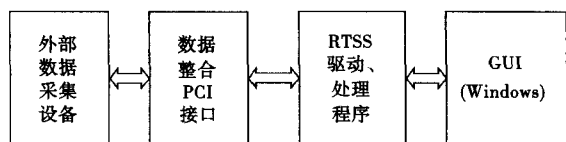


图 2 系统数据流程

其中,外设和采集卡之间的通信主要是完成数据整合,即让数据流按照 PCI 的协议传输;RTX 和 PCI 接口芯片的通信完成数据流从采集卡到主机的传输,主要是驱动程序的设计;而 RTX 和 Windows 之间的通信主要是用户控制界面和 RTX 驱动程序之间的通信,主要涉及到数据的交换和同步的设计.从上述分析来看,本次设计的主要有两部分:一是 RTX 下驱动程序的设计,将外部的数据高速实时的导入到 PC 的 RTX 环境下,从而保证数据的实时处理;二是完成 RTX 和 Windows 进程之间的通信,以确保系统强大的人机交互能力.

3.2 RTX 下驱动程序设计

RTX 下驱动程序的设计和 Windows 下不同.在设计即插即用驱动程序的时候,在 Windows 下必须按照 WDM 的模型进行设计,涉及到 IO 管理器等的操作显得很复杂^[5],在 RTX 下没有设备管理器,RTX 系统在安装的同时已经为自己支持的设备建立起了即插即用功能,所以在进行相关设备驱动开发的时候,必须先把相应的设备移植到 RTX 下.对于 PCI 设备而言,相应的驱动程序流程如图 3 所示.

对于扫描总线、映射寄存器、以及注册中断、申请相应资源等的操作,RTX 都提供了相应的 API 函数供使用,进一步方便了驱动程序的设计.由于我们采用 PCI 总线主要是用来进行高速数据的传输,

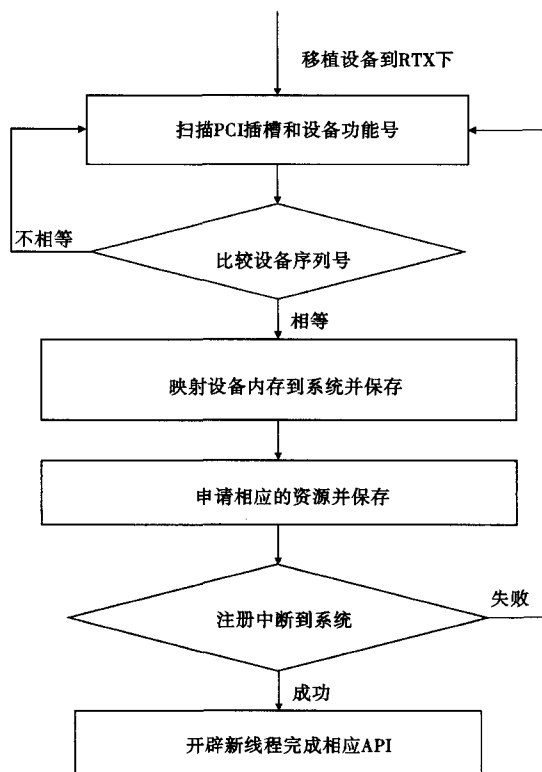


图 3 RTX 下 PCI 驱动设计流程

所以采用 DMA 加中断的方式来进行大量块数据的传输.这里采用的是 DMA 从操作的模式,我们对 DMA 的相关寄存器进行相应的设置即可.关于 DMA 的操作流程如下:

(1) 设置方式寄存器:DMAMODE,传输模式等的设置;

(2) 设置 PCI 总线侧的地址:DMAPAD,主机物理内存开始读写地址;

(3) 设置 LOCAL 地址寄存器:DMALAD,局部端开始读写地址;

(4) 设置传输字节数:DMASIZE,本次 DMA 传输的字节数;

(5) 设置 DMA 描述符寄存器:DMADPR,如果是采用 SGL 传输模式的话,本寄存器的内容是第一个描述符的内存地址,否则本寄存器只用作指示传输方向;

(6) 设置命令寄存器:DMACSR,通过本寄存器使能 DMA、开始 DMA、取消 DMA,同时对第三位的读写还能清除 DMA 中断;

(7) 等待 DMA 完成事件通知,然后进行相应的处理.

3.3 RTX 和 Windows 通信设计

在 Windows 下应用程序和驱动程序之间的数

据传递依靠的是 MDL(内存描述符列表),二者之间可以直接进行数据的传递交互。但是 RTSS 和 WIN32 是两个独立的子系统,直接的数据交换是做不到的,所以二者的通信也是一个难点。一般的进程间通信方式有文件映射、共享内存和信号量、共享变量(通过动态链接库)等,考虑到传输数据的速度和安全性能,本文采用的是共享内存和事件的方式。通过共享内存达到二者之间数据快速交换,使用共享事件做到两个子系统下进程的同步^[6]。RTX 下的共享事件和共享内存实现的机制和 Windows 很类似,关键是共享内存和事件的创建和打开必须使用 RTX 提供的 API 函数,否则的话可能造成通信失败或者传输不正常的现象。

4 系统测试及其结果分析

本次测试采用的主机配置为:CPU 为 Celeron (R),主频为 2.66 GHz,内存为 1.5 GB。数据采集卡配置为:FPGA 芯片采用的是 Xilinx 公司的 Vertex2pro20,PCI 接口芯片采用的 PLX Technology 公司的 Plx9054,晶振采用的是 40 MHz 的有源晶振。对于接口芯片我们配置的模式是突发 32 位读写的方式,数据传输采用 J 模式下的 DMA 方式。FPGA 模拟数据源,产生锯齿波数据,通过产生的中断信号来通知上位机读取数据。上位机一次读取数据为 1 MB。通过在 RTX 下统计在开始发起 DMA 读数据时的时间(时钟精度为 100 ns)与 DMA 传输完成时的时间差来计算软件上的数据传输速度,并把接收到的数据传输到 Windows 下显示出来,测试界面如图 4 所示。实验结果表明测试的数据的传输速度为 98 MB/s 左右。而图像上显示的完整性也证明了传输数据的连贯性。

为了进一步的证明采集到的数据的正确性,采用 Xilinx 开发套件自带的观察工具 Chipscope 来观测数据线上的数据,通过对比读取数据并把读取的数据回传来比较,看其是否一致来判断传输的正确性。其观测结果如图 5 和图 6 所示。

通过观测发现,读取和发送的第一个数据都是 0X006C00,证明数据传输的正确性。由图还可以看出,读取数据和写入数据的时序是基本相同的,唯一不同的是 WR_N 信号量,当读取数据的时候是低电平,当写入数据的时候是高电平。其中信号 TCLK 是对原始时钟的二分频,用以参考数据的传输速度。通过观测采集到的波形我们发现,每一次请求总线的操作共有 20 个时钟周期,其中有 13 个用于传输

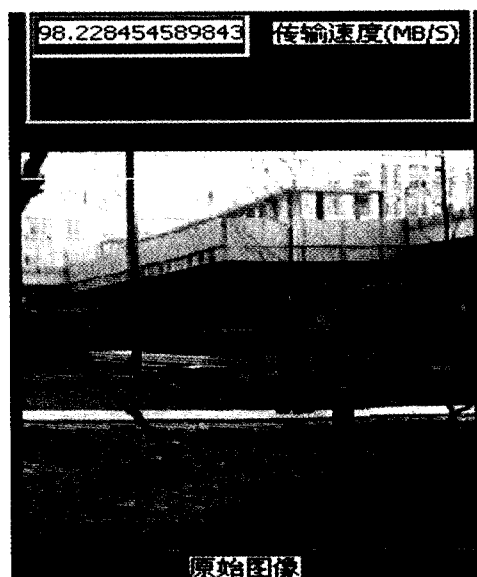


图 4 RTX 和 Windows 通信效果

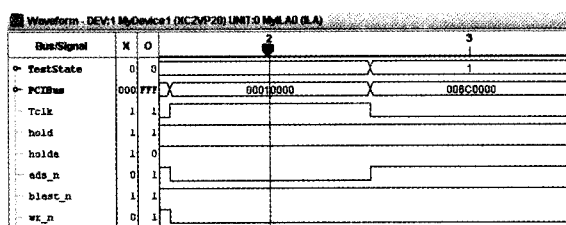


图 5 ChipScope 观测读取数据

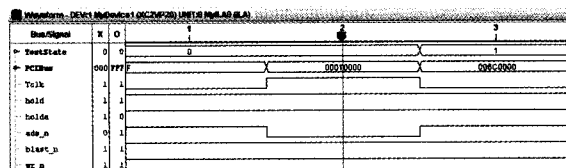


图 6 ChipScope 观测写入数据

数据,1 个用于传输地址,其余 6 个作为空闲应答开销,一次数据传输有 32 位,这样可以得到数据传输速度为 104 MB/s。由于计算机系统任务的开销,与 PCI2.2 协议规定的 133 MB/s 的理想速度相比还有一定距离,但是这样的速度已经足以满足大容量数据流的传输,能够满足数据的实时处理的需求。

5 结束语

本文提出了一种 RTX 加 Windows 的通用系统架构,该系统采用了 PCI 作为总线,采用 DMA 加中断的工作模式,数据传输速度可以达到 104MB/S,保证了系统的获取数据的实时能力;利用 RTX 对 Windows 的实时性能扩展,完全可以将一般的实时

(下转第 65 页)

4 结束语

移动网格中有大量的闲散资源,这些资源性能异构且动态变化,其中移动资源还具有间歇连接性.可分任务类型在这种动态资源环境中的调度还少有研究,本文针对这种情况提出了动态循环算法DCS,用环结构来组织任务,采用轮询的方法动态调度任务,不需要预测任务在资源上的执行时间.通过性能分析和仿真实验(因版面限制省略)证明.调度算法体现了重调度和复制任务的思想,能够很好的适应移动设备的间歇连接性.

参考文献:

- [1] 刘晏兵,尚明生,肖云鹏. 网格高性能调度及资源管理技术[M]. 北京:科学出版社,2010.
- [2] Litke A, Skoutas D, Varvarigou T. Mobile grid computing: changes and challenges of resource management in a mobile grid environment[C]//workshop: "Access to Knowledge through Grid in a Mobile World", PAKM 2004 Conference. Vienna,2004.
- [3] Katsaros K, Polyzos G C. Evaluation of scheduling policies in a Mobile Grid architecture[C]//International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS 2008).

UK: Edinburgh, 2008:390-397.

- [4] 张园园,余镇危. 基于覆盖网络的应用层选播服务的设计与研究[J]. 微电子学与计算机,2011,28(12):72-74.
- [5] 孙文乾,吴启明. 无线网络资源调度优化策略[J]. 微电子学与计算机,2011,28(7):179-182.
- [6] 徐志立,王红霞. 粒子群优化算法在网格 workflow 调度中的应用[J]. 微电子学与计算机,2011,28(11):65-67.
- [7] Katsaros K, Polyzos G C. Optimizing operation of a hierarchical campus-wide mobile grid[C]//2007 IEEE 18th International Symposium on Personal Indoor and Mobile Radio Communications, conference (PIMRC2007). Greece: Athens, 2007:1-5.
- [8] Fujimoto N, Hagihara K. Near-optimal dynamic task scheduling of independent coarse-grained tasks onto a computational grid[C]//Proc. of ICPP 2003. Kaohsiung, Taiwan, China, 2003.

作者简介:

鞠宏军 男,(1973—),硕士,副教授.研究方向为计算机网络及应用.

高永清 男,(1974—),博士研究生,讲师.研究方向为 DSP 与通信、计算机网络.

(上接第 61 页)

处理任务移植到个人计算机上来进行.该架构适用能力强,当应用场景更换时,仅仅需要更换前端采集设备即可,缩短了开发周期;其处理阶段的开发都是基于 Windows 平台,降低了开发的难度.

参考文献:

- [1] 俞锋. 雷达一次视频 RTX 实时传输和显示的软件实现[J]. 电子工程师,2008,34(4):4-6.
- [2] 黄键,庞亚华,薛顺虎. RTX 环境下 PCI 设备实时驱动的开发[J]. 计算机工程,2010,36(20):211-213.
- [3] 李晓辉,严卫生,范欣. 基于 RTX 的水下航行器实时仿

真研究[J]. 计算机测量与控制,2010,18(4):850-852.

- [4] 杨红松,杜承烈,尤涛. 基于 RT-TCP/IP 的虚拟试验通信平台研究[J]. 测控技术,2009,28(3):60-63.
- [5] 张凡,史彩成. Windows 驱动开发详解[M]. 2 版. 北京:电子工业出版社,2002.
- [6] 蒋志文. 基于 RTX 的建模与实时仿真软件 YH-RT-SIM 的设计[J]. 计算机应用,2010,30(6):1635-1637.

作者简介:

张前福 男,(1985—),硕士研究生.研究方向为实时信号处理.