

文章编号:1001-9081(2005)09-2216-03

## 多线程环境下 DSP 和主机间高速数据 I/O 方法的研究和实现

周冬梅<sup>1</sup>, 钟晓玲<sup>1</sup>, 王建勤<sup>2</sup>

(1. 成都理工大学信息工程学院, 四川 成都 610059;

2. 德加拉(成都)数码科技有限公司, 四川 成都 610051)

(zhoudm@cdut.edu.cn)

**摘要:**TI TMS320DM642 是款性能优良的多媒体 DSP。文中重点介绍了在多线程环境下 DM642 和计算机间如何通过 PCI 总线进行大批量数据高速输入输出的方法及其实现具体过程。

**关键词:**DM642;EDMA;PCI**中图分类号:**TP391 **文献标识码:**A

## Research and implementation of fast data I/O methods between DSP and host in multithread application

ZHOU Dong-mei<sup>1</sup>, ZHONG Xiao-ling<sup>1</sup>, WANG Jian-qin<sup>2</sup>

(1. College of Information Engineering, Chengdu University of Technology, Chengdu Sichuan 610059, China;

2. Thakral(Chengdu) Digital Technology Development Corporation Ltd, Chengdu Sichuan 610051, China)

**Abstract:** TI TMS320DM642 is a powerful multimedia DSP of TI. Methods transferring large quantities of data between DM642 and PC through PCI data bus in multithread application were introduced, and a fast and stable way was presented in detail.

**Key words:** DM642;EDMA;PCI

## 0 引言

随着 DSP 处理能力的不断升级,单枚 DSP 芯片所能处理的线程个数也在不断的增大,DSP 处理的数据量也越来越大,如何保证各个线程所处理的数据在 DSP 和主机间可靠而高速的交互,即一方面要保证所有线程的大批量数据的及时传输完毕,同时还要保证各个线程所传输的数据都安全可靠地被传输到指定的目的地,是大多基于 DSP 开发的项目必须面对的问题。

本文将结合 TI 的一款最新的功能强大的 DSP——TMS320DM642,介绍通过 PCI 总线实现 DSP 和计算机间大批量数据交互的常用方法及其具体实现,并根据实验结果重点给出一种基于 DSP Master EDMA 和中断的可靠而高速的实现多线程环境下批量数据 I/O 的方法。

## 1 TMS320DM642 简介

TMS320DM642 是 TI 一款专门面向多媒体应用的专用定点 DSP,其时钟高达 600MHz,有 8 个并行运算单元,处理能力高达 4800MIPS。它采用了基于 CACHE 的两层存储架构,内部有 64 个独立的增强 DMA,还具有一批功能强大而可动态配置的外设,为了面向多媒体应用,它集成了 3 个可配置的 Video Port,1 个面向音频应用的 McASP,1 个 10/100Mbps 的 Ethernet MAC 等外设,这些集成的 I/O 外设是目前市场上其他 DSP 芯片所不能比拟的。

此外,DM642 提供了 33/66M 可选的符合 2.2 规范的 32 位 PCI 接口,此 PCI 接口部分引脚和 HPI/EMAC/MDIO 复用,通过拉高 PCI\_EN 引脚,PCI 接口即可生效,其他复用接口即

为不可用。此 PCI 接口支持主/从 (Master/Slave) 工作模式,在主模式下,每次 PCI 传输最大只能到 64k 字节。

## 2 TMS320DM642 和主机间数据传输方法

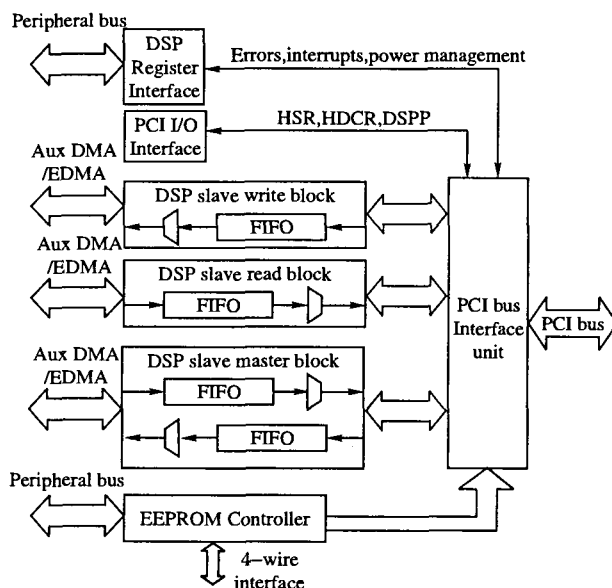


图 1 PCI 功能框图

DM642 有两种接口可和主机通信,在 PCI 卡应用环境下它采用 PCI 接口和主机通信,此时主机指的就是计算机 (PC);而在嵌入式应用环境下,DM642 要通过 HPI 接口和主机通信,此时主机指的是嵌入式微处理器,如 ARM、PowerPC 等处理器。本文所论述的是 PCI 卡应用环境下的 DSP 和主

收稿日期:2005-03-22;修订日期:2005-05-30

**作者简介:**周冬梅(1973-),女,黑龙江逊克人,讲师,硕士,主要研究方向:计算机通信、电子信息技术;钟晓玲(1976-),女,四川富顺人,助教,主要研究方向:电子信息、计算机通信;王建勤(1972-),男,江西南昌人,软件工程师,硕士,主要研究方向:多媒体通信、软件工程。

机(PC)间的数据通讯。

如图 1 所示,DM642 PCI 接口支持如下 4 种 PCI 数据传输操作:

2.1 Slave 写操作

PC 发起写 DSP 的操作,即 PC 在 WDM 驱动程序内直接寻址 DSP 的存储空间并向其中写入数据。

Slave 写操作又可分为 BURST 批量写操作和单字写操作,在基于 DDK 的 WDM 驱动程序内,分别调用 WRITE\_REGISTER\_BUFFER\_ULONG() 和 WRITE\_REGISTER\_ULONG()宏即可实现 BURST 批量数据写操作和单字写操作。

2.2 Slave 读操作

PC 发起读 DSP 的操作,即 PC 在 WDM 驱动程序内直接寻址 DSP 内的存储空间并从中读取数据。

Slave 读操作又分为 BURST 批量传输读操作和单字读操作,在基于 DDK 的 WDM 驱动内,分别调用 READ\_REGISTER\_BUFFER\_ULONG()和 READ\_REGISTER\_ULONG()宏即可实现 BURST 批量数据读操作和单字读操作。

2.3 Master 写操作

DSP 发起,向 PC 内指定存储地址写入数据。DSP 内通过调用 TI CCS 内提供的 csl(chip support library)内函数 PCI\_xfrConfigArgs(),设置操作属性为 1 即可发起 PCI 专用 EDMA 向主机存储空间写入数据。

2.4 Master 读操作

DSP 发起,从 PC 内指定存储地址读取数据。DSP 内通过调用 TI CCS 内提供的 csl(chip support library)内函数 PCI\_xfrConfigArgs(),设置操作属性为 2 即可发起 PCI 专用 EDMA 读取主机存储空间内数据。

2.5 传输实验结果

在一台配置为:P4 1.7G CPU、256M 内存、7200 转的硬盘 PC 机上,插入一张 DM642 的 PCI 卡,以一段 64k 字节的数据为例,分别采用上述 4 种 PCI 操作,记录下 Slave 操作时的计算机所需消耗的时间以及 Master 操作的 DSP 内消耗的时间,列表如表 1 所示:

表 1 Slave/Master 读写操作性能比较

	Slave 写	Slave 读	Master 写	Master 读
时间/ms	46	45	0.3093	0.3095

注:由于计算机时钟精度所限,计算机内消耗的时间统计无法精确到毫秒以下;并且 Master 操作消耗时间是指发起 PCI 传输,直到产生 PCI EDMA 传输完毕的中断之间的时间间隔。

根据 Slave 和 Master 模式读写操作所消耗的时间比较,可以计算出同样大小的一批数据,采用 Master Read/Write 传输要比 Slave Read/Write 传输快 148 倍左右。产生此结果的原因在于 TI TMS320DM642 的 PCI 外设无 DMA 控制器,无法从主机端发起 DMA 数据传输,故此 Slave Read/Write 操作只能依靠 CPU 逐字的传输,而 DSP 内有多达 64 个 EDMA,其中有个 PCI 传输专用的 EDMA,DSP 一旦发起 Master PCI 传输操作,该专用 EDMA 即接管数据传输控制,DSP 即可继续执行后续指令。

3 方法的实现

从表 1 可知,DSP Master 操作的效率要远远高于 Slave 操作,故此 DSP Master 操作是基础,但 DSP Master 操作每次 PCI 传输最多只能 64k 字节,再考虑到多线程环境下 DSP 内可能有多个线程在和主机的多个对应线程在发生数据的交互,如何保证每个线程的数据都安全可靠地被完整地收发,同时又

不过多影响系统的效率? 本文将从如下两个具体传输应用进行问题解决方案的探讨及实现:

3.1 主机到 DSP 的数据传输

要把主机的数据传输到 DSP,即可通过 Slave Write 也可通过 Master Read 操作来完成,鉴于前述实验结果,Slave Write 操作可以不予考虑。

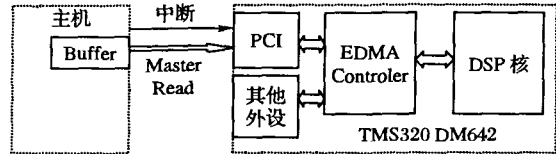


图 2 主机到 DSP 数据传输框图

基于 Master Read 操作,完成主机到 DSP 数据传输的框图如图 2 所示,其具体实现步骤如下:

- 1) 主机把数据缓存区地址和大小设置入 DSP 内自定义的内存映射寄存器,发送中断给 DSP,通知 DSP 数据准备就绪;
- 2) DSP 中断服务程序验明是主机发送数据的中断源后,发送通知给 DSP 接收数据线程,令其执行 PCI Master Read 操作。
- 3) DSP 的接收线程执行图 3 流程,完成数据接收。

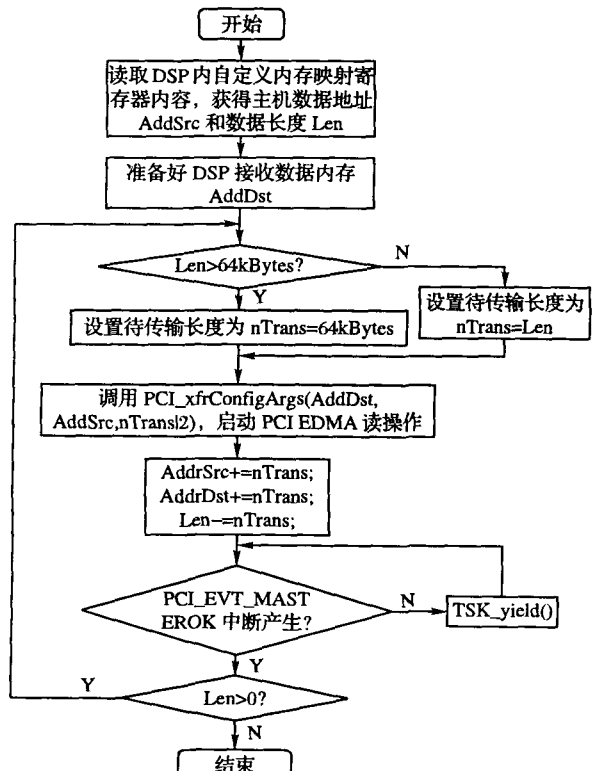


图 3 DSP 接收数据流程图

3.2 DSP 到主机的数据传输

把 DSP 内的数据传输到主机也有两种方法可用,即 Slave Read 和 Master Write,同样理由可以对 Slave Read 不予考虑。

基于 Master Write 操作,实现 DSP 到主机的数据传输的框图如图 4 所示,其具体实现步骤如下:

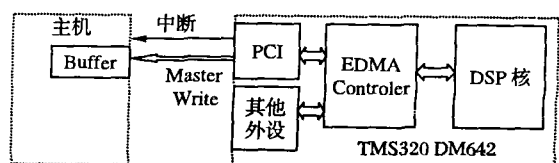


图 4 DSP 到主机的数据传输框图

1) DSP 读取主机预先设置好的主机缓存区地址内存映射寄存器,然后发起 PCI 写操作,其具体实现流程和图 3 流程类似,二者主要差异在于写操作时 DSP 内数据缓存区的地址是源,主机的数据缓存区地址是目的地。由于篇幅所限,此流程本文略过;

2) 所有数据都发送完毕后,通过调用 CCS 内的 csl 库函数 PCI\_dspIntReqSet(),DSP 向主机发送中断,通知主机 DSP 已经把数据完全搬移到主机内存;

3) 主机中断服务程序或中断延迟调用程序验明此中断源是 DSP 上传数据完毕中断,即响应该中断,执行数据的进一步传递或处理。

### 3.3 多线程应用环境下的特殊处理

考虑到多线程应用时所有数据的输入输出都要经过 PCI 总线,在前述过程基础上,还需额外考虑如何避免多线程之间因共享 PCI 总线进行数据交互而导致彼此数据被意外冲刷破坏问题,本文采用如下两个措施解决了此问题:

1) 信号灯保护:把实现 PCI 读写的代码单独写成两个函数,在 DSP 内增加一个信号灯(Semaphore)全局变量,在这两个函数的前后都采用 SEM\_pend()和 SEM\_post()来实现对 PCI 这一共享资源的同步。此措施可有效的避免 DSP 内多线程对 PCI 总线的访问冲突;

2) 竞争的方式来获取 PCI 的控制权:在 DSP 片内存储器内设立一个内存映射寄存器,该寄存器表示 DSP 内 PCI 总线的状态,供主机和 DSP 实时访问。主机某线程在向 DSP 发送数据之前,先查询该寄存器确定 DSP 的 PCI 是否空闲,如空

闲则允许 3.1 过程执行,否则循环等待直到 PCI 完成了其他线程的传输后空闲为止,一旦获得 PCI 使用权后,主机立即置 PCI 状态为忙;在 DSP 内部,数据接收线程一旦接收完毕所有数据后立即置该状态为空闲。此措施则可有效避免主机内多线程间对 PCI 总线的访问冲突以及输入和输出的访问冲突。

综合 3.1 和 3.2 方法和此两点措施,实验表明,多达 6 路线程同时在主机和 DM642 DSP 间进行每秒 25 帧 CIF 图像的输入输出都可以完全正确地得到满足。

## 4 结语

目前,国内有很多公司正在 DM642 上进行机顶盒、数字视频播放器、点播机和网络相机、监控系统网络视频服务器及编解码 PCI 卡等产品开发,为了提高产品的性价比,其中绝大多数产品都要求在每枚芯片上实现多通道(多线程)功能,因此,本文所研究的多线程应用环境下 DSP 和主机间数据 I/O 的方法具有普遍的参考价值。

### 参考文献:

- [1] TI. TMS320C6000 DSP Peripheral Component Interconnect (PCI) Reference Guide[M/CD], 2003.
- [2] TI. TMS320DM642 Video/Imaging Fixed-Point Digital Signal Processor Data Manual[M/CD], 2003.
- [3] TI. TMS320C6000 DSP Enhanced Direct Memory Access (EDMA) Controller Reference Guide[M/CD], 2003.
- [4] BAKER A, LOZANO J. Windows 2000 设备驱动程序设计指南[M]. 北京:机械工业出版社,2001.

(上接第 2213 页)

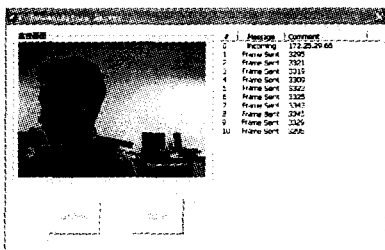


图 6 服务器程序运行效果图

在智能手机上的客户端采用 J2ME/MIDP 实现,实现环境为 JBuilderX + WTK2.2beta, socket 通信部分通过 J2ME 中提供的通用连接框架(GCF)实现<sup>[6]</sup>,监控图像的解码与显示则通过设计一个 MJPEGPlayer 类实现。该类扩展了 J2ME MMAPi 的 Player 接口,并主要用 Image 类和 Timer 类

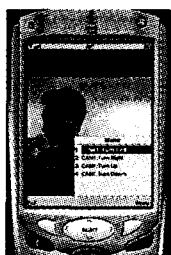


图 7 模拟器上的监控效果图

实现所需的功能。模拟器上的效果如图 7。图 7 中模拟器的配置(网络带宽,计算能力等)已经根据实际环境进行了修改,在 A760 上的效果与此图基本相同。注意图中的菜单部分,通过其中的命令就可以控制监控摄像机的拍摄方向。

## 4 结语

经过反复测试,原型系统在 Motorola A760 手机上监控图像的显示帧数约为 6fps 左右,远端监控设备对控制命令的响应时间小于 3 秒,监控图像与监控现场之间的延迟约为 15 秒

(包括初始连接延迟以及为了防止图像抖动设计的图像缓存造成的延迟),基本能满足一般用户对“家居环境”进行监控的要求,达到了设计目标。

系统在协议设计以及原型实现时,已经为加入其他监控设备留下了接口,下一步准备实现将更多的设备纳入服务器的控制,如加入第二个监控摄像机甚至各种智能家电。这样就能实现用户通过智能手机与整个家居大环境的远距离交互,这也是项目组计划进行深入研究的方向。

### 参考文献:

- [1] OBOE R. Web-interfaced, force-reflecting teleoperation systems[J]. IEEE Transaction on Industrial Electronics, 2001, 48(6): 1257 - 1265.
- [2] BRASCHE G, WALKE B. Concepts, services, and protocols of the new GSM phase 2 + general packet radio service[J]. IEEE Communications Magazine, 1997, 35(8): 94 - 104.
- [3] STABERNACK B, VON COLLN G. An MPEG-4 video codec SOC for mobile multimedia applications[A]. IEEE International Conference on Consumer Electronics[C], 2003. 248 - 249.
- [4] BRADBURY S. Communications protocols and compression techniques for digital CCTV applications[A]. IEE Seminar on CCTV and Road Surveillance[C], 1999. 3/1 - 3/5.
- [5] WALLACE GK. The JPEG still picture compression standard[J]. Communications of the ACM, 1991, 89(10): 1498 - 1517.
- [6] RIGGS R, PEURSEM JV, PATEL M, et al. Programming Wireless Devices with the Java2 Platform[M]. Addison-Wesley, 2002.
- [7] Xirlink, Inc. Xirlink NetworkCamera Software Development Kit v1.00[Z]. 2002.
- [8] 钟玉琢,向哲,沈洪. 流媒体和视频服务器[M]. 北京:清华大学出版社,2003. 6.