

HPI 在主从式系统数据通信中应用

朱世东, 张道信, 吴小培

(安徽大学 计算智能与信号处理教育部重点实验室, 安徽 合肥 230039)

摘要: 数据通信是主从式系统设计的一个关键技术, HPI 是 TI DSP 的一个重要外设, 文中旨在介绍 TMS320C54x 中主机接口 HPI 在数据通信中的应用。介绍了数据通信的概念、HPI 的结构及其工作原理。最后以 TMS320C5402DSP 为例, 重点给出了一种 DSP 与 51 单片机通信的具体硬件连接方案和软件实现方法, 并且给出了部分程序代码。从这种数据通信方案中, 可以发现 HPI 所具有的一系列优点, 对于复杂系统和更加灵活新颖的数据传输/共享设计具有一定的借鉴意义。

关键词: 主从式系统; 数据通信; DSP; HPI; 89C51

中图分类号: TP391; TN919

文献标识码: A

文章编号: 1673-629X(2006)12-0212-03

Application of HPI in Data Communication of Master - Slave System

ZHU Shi-dong, ZHANG Dao-xin, WU Xiao-pei

(Education Ministry Key Lab. of Intelligent Computing and Signal Processing, Anhui University, Hefei 230039, China)

Abstract: Data communication is a pivotal technology in the design of master-slave system. HPI (host port interface) is a kind of important peripheral of TI DSP. The aim of this paper is to introduce the HPI of TMS320C54x and its application in data communication. The conception of data communication is introduced; the structure and principle of HPI are introduced in detail. In the end, taking TMS320C5402DSP as example, introduces the hardware connections and software applications when communicating with C51, and also part of the procedure code was presented. From this scheme of data communication, may find out a series of advantages of HPI. For the design of complex system and flexible and novel scheme of data communication / share, there is a certain extent meaning of using for reference.

Key words: master-slave system; data communication; DSP; host port interface; 89C51

0 引言

DSP 作为一种先进的可编程数字信号处理器, 在近几年来, 随着性价比的提高和 DSP 技术的普及, 在很多领域得到越来越广泛的应用。但在很多应用场合, 由于系统除了要完成数字信号处理功能外, 还有许多诸如与外部系统的通信、控制、人机接口等控制功能, 如果仅用一个 DSP 来实现, 往往不容易完成包括数字信号处理在内的所有功能, 这充分暴露了其控制能力不足的缺点, 而传统的单片机具有很强的控制功能, 在这种情况下, 单片机加上一个 DSP 构成主从式系统, 每个处理器完成系统的一部分工作, 是一个比较理想的方案。在这种主从式系统中, 一般单片机作为主机, 而 DSP 则作为从机, 主机可以控制从机的复位、运行和挂起, 从机在主机的控制下完成所分担的一部分工作。

收稿日期: 2006-03-04

基金项目: 国家自然科学基金项目(60271024); 安徽省人才资助基金项目(2004Z028)

作者简介: 朱世东(1975-), 男, 安徽芜湖人, 硕士研究生, 研究方向为现代信号处理; 吴小培, 教授, 博士生导师, 研究方向为现代信号处理。

1 DSP 主从式系统中的数据通信

在设计主从式 DSP 系统时, 主机将从外部获得的数据交由从机处理, 而从机则将处理后得到的有关结果传递给主机, 由主机将处理结果发送至其他系统, 显然, 主机与 DSP 的数据通信是必须考虑的重要环节, 不同的通信方式应用于不同的场合和目的, 一般实现双机数据通信的方式主要有: 串行通信(利用处理器本身提供的串行口或在芯片的基础上用软件或者硬件开发一个串行口实现双机通信, 这种方式相对简单, 适合于通信量不是很大的场合); 并行通信(利用处理器的 I/O 功能, 在主机和 DSP 之间增加缓冲器或者锁存器实现双机通信。与串行通信方式相比, 效率稍高些); 共享存储器方式(这种方式还可以分为两种: 一种是主机和从机不能同时访问共享存储器, 一般利用 DSP 提供的 DMA 功能; 另一种是主机和从机可以同时访问共享存储器, 一般采用双端口存储器来实现)。在实时系统等数据量很大的应用中, 要充分利用并行通信方式的优越性。同时为了满足实时性要求, 必须保证足够的数据传输速率^[1]。

2 主机接口(HPI, Host Port Interface)

HPI^[2,3]是 TI 公司在新一代、高性能 DSP 芯片上配

置的,主要是为了满足 DSP 与其它的微处理器接口而专门设计的。它与主机之间能够实现并行、高速的数据传输。DSP 芯片中的 HPI 分为 HPI8 和 HPI16,分别针对具有 8 位和 16 位数据线的总线,每一种又分为标准型和增强型,两者的主要区别在于:标准型只能访问固定的地址空间而增强型可以访问整个 DSP 的片内存储器。这里以 5402 为例,介绍一下 HPI 的结构和原理。

HPI8 是一个 8 位增强型并行口,用来与主设备或主处理器接口。外部主机是 HPI 的主控者,它可以通过 HPI 直接访问 DSP 的存储空间,包括存储器映像寄存器。当 C54xx 与主机进行信息交换时,HPI 是主机的一个外围设备。如图 1 所示为 C54xxHPI,主要由以下 5 个部分组成:

- * HPI 存储器(DARAM)。HPI RAM 主要用于 C54x 与主机之间传送数据,也可以用作通用的双寻址数据 RAM 或程序 RAM。

- * HPI 地址寄存器(HPIA)。它只能由主机对其直接访问。寄存器中存放当前寻址 HPI 存储单元的地址。

- * HPI 数据锁存器(HPID)。它也只能由主机对它直接访问。如果当前进行的是读操作,则 HPID 中存放的是要从 HPI 存储器中读出的数据;如果当前进行的是写操作,则 HPID 存放的是要写到 HPI 存储器的数据。

- * HPI 控制寄存器(HPIC)。C54x 和主机都能对它直接访问,它映像到 C54x 数据存储器的地址为 002CH。

- * HPI 控制逻辑。用于处理 HPI 与主机之间的接口信号。

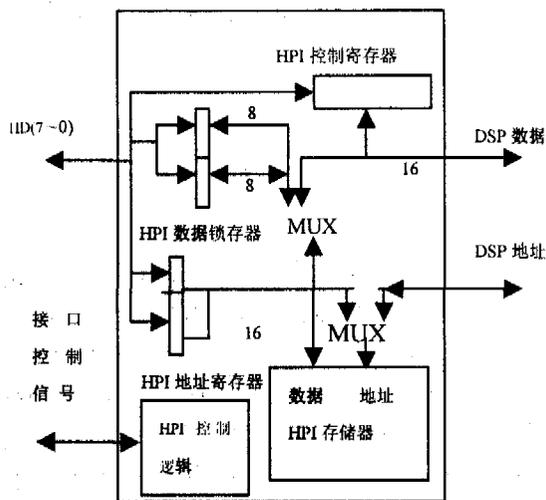


图 1 主机接口 HPI8 组成框图

HPI 有以下两种工作方式:

- 1) 共用寻址方式(SAM)。这是常用的操作方式。在这种工作方式下,主机和 C54x 都能寻址 HPI 存储器。如果 C54x 与主机的周期发生冲突,则主机具有寻址优先权,DSP 则等待一个周期。

- 2) 仅主机寻址方式(HOM)。在这种操作方式下,仅仅只能让主机寻址 HPI 存储器,C54x 则处于复位状态或者处在所有内部和外部时钟都停止工作的空转状态。DSP 处于最小功耗配置。

HPI 口数据传送由两部分组成:内部传送和外部传送。内部传送在 HPI 口寄存器和片内 RAM 之间进行。外部传送在主机和 HPI 口寄存器之间进行。在 SAM 方式下,HPI 支持的传输速率是 $(F_d \times n) / 5$,其中 F_d 为 DSP 的主频, n 为每进行一次外部寻址的周期数;在 HOM 方式下,主机能每 50ns 寻址一个字节,速度更快。

3 TMS320C5402DSP 利用 HPI 和 89C51 单片机实现数据通信

3.1 硬件电路设计

本通信电路采用单片机 89C51 作为主机与 C5402 实现数据通信,系统接口电路如图 2 所示。当 HAS 接主机的 ALE 端时,HD0 ~ HD7,HCNTL0,HCNTL1,HBIL,HR/W 接数据/地址复用总线。主机控制信号线 HCNTL0,HCNTL1,用于选择 3 个寄存器 HPIA,HPID 和 HPIC。读写信号线 HR/W 用于指示 HPI 传输方向。HDS1,HDS2,HCS 用于数据选通。HRDY 和 HINT 组成和主机通信的握手信号。需要注意的是,DSP 与单片机的电平是不兼容的,信号进入 DSP 之前必须进行电平转换,否则容易损坏 DSP。本接口使用 TI 的芯片 74LVC16245A,其中 G 为使能端,低电平有效;DIR 为方向控制端,高电平 A→B,低电平 B→A^[1,4]。

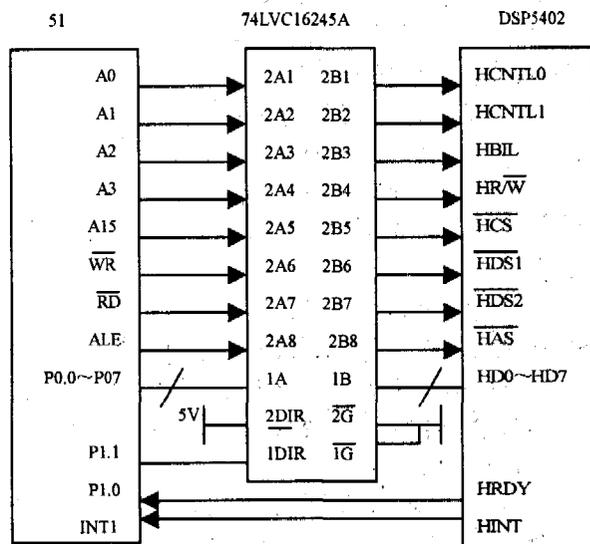


图 2 51 单片机和 5402DSP 的硬件接口原理图

在采用 5402 的 HPI 引导方式时,还得在硬件电路上直接将 DSP 的 HINT 脚与 INT2 脚相连,因为 HPI 模式是复位后首先检查的模式,引导加载程序检查标志寄存器中 INT2 位是否为 1,如果是则进行 HPI 加载。复位发生后,引导加载程序首先将数据存储器 007FH 单元初始化为 0,代码传完后,在该单元写入代码存放的位置,以便 DSP 跳去执行,但是如果当芯片复位之后,引脚 MP/MC=0,位于 DSP 片内 ROM 的引导加载程序将开始执行。

3.2 通信软件设计

89C51 作为 HPI 接口通信的主动方,可以采用端口方式直接访问 5402 片内 RAM。对 HPI 口进行数据读写需

要 3 个步骤:设置控制寄存器 (HPIC), 写地址寄存器 (HPIA), 读写数据寄存器 (HPID)。在寄存器读写过程中, 单片机通过端口发送控制信号, 检测状态信号, 完成对 HPI 口访问时序的模拟。一般采用共用寻址方式, 整个通信过程以各部分状态的变化来推进^[1,4,5]。

在编程时要特别注意下列问题:

* DSP 的采样选通信号由 HDS1, HDS2 和 HCS 的组合逻辑来组成。另外, 还要通过 HPI 的控制寄存器 (HPIC) 中的 BOB 位来指定第 1 个字节作为高 8 位还是低 8 位, 所以主机在访问 HPI 时, 应首先对 HPIC 进行初始化, 并注意对 BOB 位的设置。

* 在外部传送中, 由于 DSP 的数据是 16 位, 而单片机的是 8 位, 所以单片机要分两次将数据传给 DSP, 即将 16 位的数据分成两个字节来传输。这时, 可以通过控制 HPI 口的 HBIL 信号来指定此次传输的是第 1 个还是第 2 个字节。主机将 HBIL 拉低, 通知 DSP 此次传输的是前 8 位, 将 HBIL 抬高, 则通知 DSP 此次传输的是后 8 位。

* 主机对地址寄存器 (HPIA) 的写操作会初始化一次内部传输。当主机通过两次对 HPIA 的写操作后, HPIA 就得到了主机的访问地址。这时内部的 DMA 控制器就会根据这个地址将相应单元的内容读到 HPI 内部的数据锁存器中, 再对 HPID 进行两次读操作就可以将数据读出。如果将 HPIA 设置成自动递增模式, 就会在数据传输的同时完成 HPIA 加 1, 于是又启动了一次内部传输。这样有利于数据的连续转移。

* 注意设置 HPIC 中的 XHPIA 位。XHPIA = 1 时, 表示对 DSP 的 7 位扩展地址进行操作; XHPIA = 0 时, 表示对 DSP 的低 16 位地址进行操作。由于 DSP 复位后, XHPIA 的状态是不确定的, 所以必须首先对 XHPIA 进行设置。

* 主机和 DSP 可以互相中断。主机通过向 HPIC 中的位 DSPINT 写入 1 来中断 DSP。该位总是被读为 0。

下面的例子程序, 能够实现单片机从 DSP 的指定地址读取数据, 再将数据写到 DSP 的其他存储单元。

1) DSP 端程序:

将要发送给单片机的数据放在指定的存储单元, 在设置好初始化状态后, 向单片机发中断通知数据已经准备好。并等待主机发中断, 将数据存入目标位置。DSP 在中断中对接受的数据进行处理。

2) 单片机端程序:

作为主控部分, 要完成初始化后, 等待 DSP 发中断通知数据已经准备好, 单片机在检测到中断后, 还要判断中断是否有效, 再从 HPI 口读写数据。主程序片断如下:

```
void main(void)
```

```
{
    CS1_Init(); // 单片机初始化
    HPI_Init(); // HPI 初始化
    EA = 1; // 开放所有中断
```

```
while(1)
{
    If (Flag_Hint == 1)
    {
        unsigned int i;
        int l_data[16];
        GetData[0] = 0x80; // DSP 存储数据地址
        GetData[1] = 0x00;
        HPIA_Write(); // 主机设置的子程序
        for (i=0; i<0x8; i++)
        {
            HPID_Read(); // 单片机从 DSP 读数据
            l_data[2*i+1] = GetData[1]; // 保存接受数据
            l_data[2*i] = GetData[0];
        }
        GetData[0] = 0x87; // 单片机写入数据地址, 以地址连续方式写入时
        GetData[1] = 0x00; // 地址先增 1
        HPIA_Write(); //
        for (i=0; i<0x8; i++)
        {
            GetData[1] = l_data[2*i+1];
            GetData[0] = l_data[2*i];
            HPID_Write(); // 单片机向 DSP 发送数据
        }
        GetData[0] = 0x0c;
        GetData[1] = 0x0c;
        HPIC_Write(); // 单片机配置寄存器 HPIC
        Flag_Hint = 0;
    }
}
```

4 结束语

文中介绍了一种典型的单片机和 DSP 的数据通信方案, 并在本实验室通过试验电路的测试, 在实践中发现, HPI 提供了方便而灵活的接口, 而且外围电路简单, 几乎不需要附加任何逻辑电路。在 HPI 通信方式下, DSP 的片内存储器对外界完全透明, 由主机通过访问 HPI 的地址和数据寄存器来完成对 DSP 片内存储器的读写。特别是在嵌入式 DSP 产品开发调试阶段, 使在线修改 DSP 存储器的数据成为可能, DSP 在通过 HPI 口和主机通信的过程中完全没有硬件和软件开销, 由 DSP 自身的硬件协调冲突, 从而不会打断 DSP 正常程序的运行。因此, 这种数据通信方案, 对于复杂系统和更加灵活新颖的数据传输/共享设计具有一定的借鉴意义。

参考文献:

- [1] 张雄伟. DSP 芯片的原理与开发应用[M]. 第 3 版. 北京: 电子工业出版社, 2004. (下转第 236 页)

表 4 实验结果

	Apriori 算法	FP - Growth 算法
生成的规则数	24	19
检测到的入侵事件	35	46
生成规则运行时间(s)	6.03	4.60

从上表可以看出 FP - Growth 算法在生成规则少于 Apriori 算法的情况下检测出了更多的入侵事件,有效地降低了规则的冗余度和系统的漏警率;同时生成规则所需要的时间也比 Apriori 算法要少。在处理实际入侵检测所产生的海量数据库和各种各样的入侵事件时,FP - Growth 算法的速度优势将更加明显地体现出来。从总体上来看,FP - Growth 算法的性能要优于 Apriori 算法,能够更好地为入侵检测系统服务。

4 结束语

文中研究了在入侵检测中应用 FP - Growth 算法。详细讨论了 FP - Growth 的实现方法,并对它进行了改进,实验证明改进的 FP - Growth 算法比传统的关联算法在入侵检测中的应用效果更好。但是我们注意到 FP - Growth 算法也没有检测出所有的入侵事件,因此进一步提高检测的精确度成了今后工作中应重点研究的内容。

(上接第 121 页)

了彼此间的关联关系。

下面是系统的整个工作流程,假设用系统管理员的角色登陆 login.jsp,系统将请求提交给 loginservlet,loginservlet 根据用户提交的信息调用业务逻辑层相关的 bean,并进行相应的帐户和密码验证,若通过则由 loginservlet 转发到 main.jsp,否则转发到 login.jsp。验证成功后管理员要添加人员,系统即调应 addperson bean,当管理员添加完信息提交后,系统将通过数据访问层在数据库的 person 表中添加相应的人员信息。

4 结语

由上文可以看出,整个系统使用了 MVC 设计模式^[5],系统的主要优点是:

(1)在开发的过程中,只要定义好相应的接口规则,开发人员即可以专注于自己模块的开发,从而提高了系统的开发效率。

(2)Web 层与其它层相分离,可以根据当前的情况更新页面内容,增加系统需求的新功能,而其它的代码却无

(上接第 214 页)

子工业出版社,2003。

[2] TMS320C54X DSP CPU and Peripherals[M]. US: Texas Instruments. 1999.

[3] 赵红怡. DSP 技术与应用实例[M]. 北京:电子工业出版社,

参考文献:

- [1] HAN JIAWEI, PEI JIAN, YIN YIWEN, et al. Mining Frequent Patterns without Candidate Generatin [C]//In: Proc Conf on the Management of data(SIGMOD'00, Dallas, TX). New York, NY, USA: ACM Press, 2000.
- [2] Agrawal R, Imieliński T, Swami A. Mining Association Rules between Sets of Items in Large Databases[C]//Proc Conf on Management of Data. New York, NY, USA: ACM Press, 1993:207-216.
- [3] 王新宇,杜孝平,谢昆青. FP - Growth 算法的实现方法研究[J]. 计算机工程与应用, 2004(9):174-176.
- [4] 吕 锋,陈华胜. 关联算法的改进及其在审计数据挖掘中的应用[J]. 武汉理工大学学报:信息与管理工程版, 2004, 26(5):5-9.
- [5] 赵艳铎,宋斌恒. 基于逆向 FP - 树的频繁模式挖掘算法 [J]. 计算机应用, 2005, 25(6):1385-1387.
- [6] 朱秋萍,毛平平,罗 俊. 基于关联规则的入侵检测系统 [J]. 计算机工程与应用, 2004(26):160-162.
- [7] Bonchi F, Goethals B. FP - Bonsai: the ART of Growing and Pruning Small FP - trees: Proc 8th Pacific - asia Conference on Knowledge Discovery and Data Mining, PAKDD' 04, Sydney, Australia [C]. Heidelberg, Germany: Springer - Verlag, 2004:155-160.

需太大的改变,从而保证了软件的可扩展性。

(3)业务逻辑层与其它层的分离,最大程度地提高了代码的重用性,只要对公共的组件进行优化,系统的整体性能即可得到明显的提高。

通过实践证明,该系统在 J2EE 框架中采用了 MVC 模式开发,极大地提高了系统的健壮性、可维护性和可扩展性。从而为许多基于 B/S 的系统提供了参考。

参考文献:

- [1] Bergsten H. JSP 设计[M]. 第 3 版. 北京:中国电力出版社, 2004.
- [2] 孙卫琴. 精通 Struts: 基于 MVC 的 JavaWeb 设计与开发 [M]. 北京:电子工业出版社, 2005.
- [3] Roman E. 精通 EJB[M]. 第 2 版. 北京:电子工业出版社, 2002.
- [4] 易可可,陈志刚. 基于 MVC 模式的 Web OA 系统设计与研究[J]. 计算机工程与应用, 2005(4):112-115.
- [5] 杨开英,刘 树. MVC 设计模式在 J2EE 平台上的研究与实现[J]. 微机发展, 2004, 14(11):114-116.

2003.

[4] 汤华庚. TI54xxDSP 与 51 单片机的接口技术[J]. 单片机与嵌入式系统应用, 2004(1):27-30.

[5] 谢劲励. TMS320VC54X 接口技术应用研究[J]. 工业控制计算机, 2004, 17(3):17-19.