基于 C 总线多 ID系统的高速主机接口设计

张 骋 蔡惠智 何国建
中国科学院声学研究所 北京 1 0 0 0 8 0

摘 要 : C 总线是一种成熟全面的计算机总线 通过它主机可以方便地为 D 加载程序 进行调试和监控工作 另外还能在主机和 D 系统间完成高速数据传输工作。本文结合一个已经成功应用的设计 阐述了基于双状态机 + C结构的主机接口设计 并给出了逻辑框图。利用 P 公司的 P C 接口芯片 5, I6 提供了高达 9 0 的主机访问 s D 和 S IS D 的速度。M

关键词 : C 总线 C IF P G A D S P S D R A M C a c h e

H i g h - s p e e d h o s t i n t e r f a c e

Z H A N , C A h e H u i Z h i J i a n
(I n s t i t u t e o f A c o u s , C h i n e A c a d e

A b s t r a c t : C P C I i s a p o w e r f u l c o m p u t e r b u s , w h i t h e h o s t . I n t h e o t h e r h a n d , i t c a n b e a h i g h s p e e c a t e d d e s i g n , t h i s p a p e r s h o w s a n e w s t r u c t u r e h W i t h t h e P C I i n t e r f a c e c h i p , P C I 9 6 5 6 , t h e F P G A s

K e y : C o P r C d F s P ; u S S P D R A a M h e

在现代通信、雷达和声纳系统中 随着实时处理要求的不断提高 对数字信号处理系统也提出了更高的要求。板载多片高性能的 D 芯片P 配合大容量的 S D 可以很好地满足上述要求 并且已经成为了数字信号处理系统发展的趋势。采用 C 总线集成系统 可以方便主机进行调试 控制和管理 D 系统P 系统中的主机接口可以使主机通过 C 总线访问板上的 D 和 S IS D 芯片A 这是多 D 系统设计的关键点之一。

不同于以往简单地使用一个 C 进行粘合逻辑设计 本文提出了一种基于双状态机 + C 预存预取的主机接口设计结构。在主机接口中设立了一个 C a降c h e 低了 C 总线与板上 D 和 S IS D 芯片的耦合度 并且设计了两个独立状态机分别进行控制。这显著提高了主机访问 D 和 S IS D 的速度 为 D 系统的应用提供了更广阔的平台。本文详细阐述了如何完成 C 总 C I 线和 D、SSPD 芯片间的数据传输 分析了设计难点 并给出了逻辑框图。

接上页)

[章晓卿 刘中元 非接触式 R 读写器系统的研究 [J] . 国外电子元器件 , 2 0, (0 2 6) 5 - 1 6 .
[3] K l a 德 著 陈大才 编译 王桌人 审校 射频 e r 识别技术 [北京] 电子工业出版社 , 2 0 0 1 .
[4] L A 著 邵贝贝 译 μS C / J G 源代码公开的实时嵌入式操作系统 [北京] 中国电力出版社 , 2 0 0 1 .
[赛奎春 宋坤 . S Q 数据库开发实例解析 r [北京] : . 机械工业出版社 , 2 0 0 6 .
[清宏计算机工作室 . D 编程技术 网络与数据库篇) [M] . 北京 机械工业出版社 , 2 0 0 1 .
[马海莲 . D 组件应用实例 7[北京] 电子工业出版社 , 2 0 0 3 .
[彭勇 王建芬 . D e 网络与通信开发应用 [北京] : . 中国水力出版社 , 2 0 0 0 .
[杨世裟 赵辉编 . A S P + 动态网站开发从基础 v e r 到实践 [北京] 电子工业出版社 , 2 0 0 6 .
[袁国昌] 景鹏森 . 动态网页设计教程 [北京] 科学出版社 , 2 0 0 5 .

系统设计方案

图 是系统设计框图 系统采用 P 公司的 P C I 9 6 接口芯片 它可以很方便地将时序相对复杂的 P 协议 I 转化为相对简单的局部端访问协议。在基本不损失性能的同时 简化了逻辑设计要求 使开发者可以更为关注后端数据接口问题。

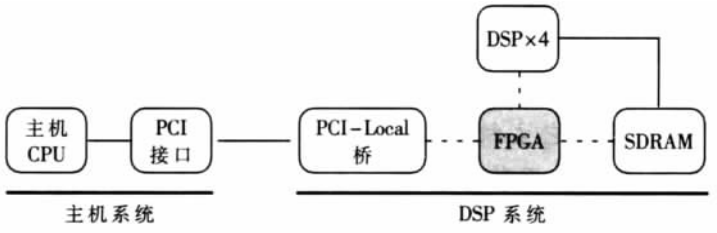

图 1 多 DSP 系统设计框图

F I 采用 AX 公司的 nXx 2 V 它有近 0l 万门 0 的逻辑资源和 7 2 的 0 K B o 可以灵活搭建控制逻辑和 C 缓存。h e

D 采用 P A 公司的 A D S P 共有 T 片 S 工作 0 1 频率是 6 0 总共可以提供 1 4 . 4 的运算能力 [O]P S

S D 采用 A HM 公司的 xH Y 5 7 V 容量共 1 有 1 2 可以基本满足数据存储的要求。

2 F 的 P 接口设计

2 . 1 在系统中的作用

F I 主要实现如下功能接口 :(1 接口 D 提供一个 P C I 总线到 o d 共享总线的界面 完成两套总线之间的逻辑仲裁及读写控制信号等 ;(2) 接口 Q R 提供一个 P C I 总线到 cSaDl 总线的界面 ;(3) F L 接口 S;(链路口 ;(5) 管理模块。i 图 s 给出了 F I 的各种接口与系统其他部分的关系图。本文将重点讨论主机和 D、SSPD 间的访问。

2 设计思想

P 局部端的时钟是 6 6 而 HD 共享总线为 1 0 0 时钟的不匹配会给逻辑设计提出很多时序方面的问题。另外 ,P 局部端数据总线是 3 位 2 而 S D 数据总线是 6 位 4 如何匹配数据宽度也是一个问题。而且 P 局部端和 D、SSPD P A M 在控制时序上也有很大差别。

图 中显示的是共享总线结构 , D、SSPD 和 R A F M 都挂在 D 的外 P 部总线上。 D 之间的通讯可以使用 D 总线 P ,各个 D 访问 P S D 时 R 也要选择 D 总线 P ,而且当主机访问 D 通讯时 ,也会不可避免地使用 D 总线。因此不难得出这样的结论 :D 总线 P 将可能成为系统的瓶颈所在。所以在设计主机接口时 必须提高总线的使用效率 减少申请 D 总线的次数 每次申请使用 D 总线时都要尽可能多地传输数据。

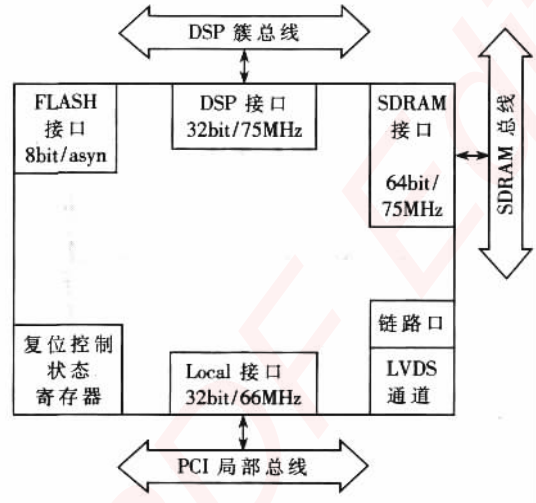

图 2 FPGA 的逻辑功能框图

在微处理器设计中 ,C a 被用来缓存数据、解决高速 S C 访问低速存储器时的瓶颈问题。为了解决上述两大问题 ,在 F I 的设计中也采用了类似 C a 的结 h e 构来隔离不同总线间的传输。在 C a 的两边有两个状态机来控制 C a 的读写和总线数据的访问。使用 C a 后 c , b 共享总线和 P 局部端总线将被去耦合 , 这样可以使两级总线的数据吞吐量都尽量达到自己的峰值速度。 F P 内部有丰富的存储资源 ,大块的 B l o 可以方便地搭建成 C a 而且 h Ce 越大 越 e 能提高主机访问 D 和 S D 的效率 M 减小占用 D S P 总线的时间 ,从而可以缩短 D 间通过 D 总线互访时的等待时间。

图 是 D S P 接口框图 R 都是基于双状态机加上 C a 结构 h [。e 两个状态机同时监测 C a 当前 h e 空、满或是数据个数等状态 以决定其动作 另外状态机间还有命令通道 局部端状态机用它向 D S P / S D 端状态机发出命令。由于这部分跨越了两个不同频率的时钟域 ,因此必须加上同步电路以防止寄存器不定态的产生。

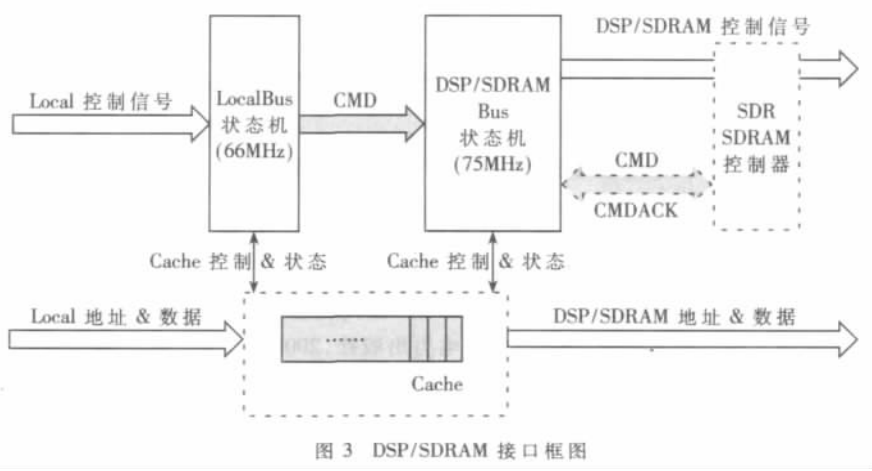

图 3 DSP/SDRAM 接口框图

SDRAM接口与 DSP接口不同处是它还有一个标准的 SDRAM控制器 负责将自定义的 SDRAM读写命令翻译成 SDRAM控制信号线 RAS# CAS#和 WE#的组合。将 SDRAM控制器独立出来可以使得设计更加模块化，避免 SDRAM端状态机过于庞大[3]。

2.3 DSP/SDRAM接口的实现

主机访问 DSP时，必须遵守 DSP的流水线协议 其中重要的是读写时的流水深度 读操作时流水深度始终为四个周期 写操作时流水深度始终为一个周期。主机执行来自或去往 DSP的突发操作时 支持超过四字的连续突发操作。当主机发出突发首地址 只要 BRST信号有效，DSP就在内部对地址累加。首次传送的起始地址和最后一次传送的结束地址必须四字对齐。这里只支持 DSP端 字突发。

SDRAM的访问协议很常见 限于篇幅 不再赘述。

每次主机开始访问 DSP/SDRAM时，PCI局部端使用 LHOLD和 ADS来启动一次传输，LWR=表明是读过程。局部端状态机向 DSP/SDRAM端状态机发出 start命令。开始时 Cache为空 而且 DSP/SDRAM提供数据要超过一段时间，所以在读操作的开始阶段要无效 ready_n使局部端等待。 DSP/SDRAM端状态机接收到局部端状态机的 start信号后 开始从 DSP/SDRAM读出数据 填入 Cache中。

当预定义的数目被满足后，局部端状态机使 ready_有效以允许当前的读操作。 DSP/SDRAM端将会根据用户设定是否突发读写 DSP/SDRAM的方式，一直连续地读 DSP/SDRAM数据 写入 Cache中。除非接收到 stop命令才回到 IDLE状态 或是在 Cache快要填满进入等待状态 放弃 DSP共享总线 这样可以使 片 DSP之间的数据传输尽可能少地被干扰。每次主机读 DSP/SDRAM时 都会直接从这个 Cache中读出数据。如果 Cache中的数据少于预定义的数目，则 ready_将被无效 以使当前的数据传送等待。一旦 LHOLD信号无效 当前这段 PCI传送结束 或是局部端地址不连续了 局部端有一个地址寄存器 它标志 Cache中下一个数据的地址，如果它和当前局部端地址不同表示预取的 Cache已经无效)局部端状态机就向 DSP/SDRAM端状态机发出 stop命令。 DSP/SDRAM端状态机清空 Cache，准备下一次访问。读 SDRAM的同时开启一个定时器，一定的时间间隔内要在读命令中插入刷新命令，防止数据丢失。

主机写 DSP/SDRAM的操作过程因为有 Cache的存在显得很简单[4]。因为局部端和 DSP/SDRAM端之间有 Cache完全隔离，所以局部端状态机只要判断 Cache中还有足够的空余位置就开始往 Cache中分别写入地址和数据 两者是一一对应的 。)局部端状态机在写的过程中，根据 PCI9656的 blast信号来判断单次还是突发以及突发是否结束 如果 Cache中空余位置少于 个则进入等待状态。 DSP/SDRAM端状态机一旦看到 Cache不为空 便从 Cache中读出地址和数据 整合了一段数据后开始申请 DSP共享总线 按照协议规定的时序要求将数据写到 DSP/SDRAM中。对 DSP的写操作就像写 SRAM一样简单方便 写 SDRAM稍微复杂一些 除了要像读一样插入刷新命令外，每次写 SDRAM到了页末时必须及时发出预充电命令 防止地址错误地回转到页首 另外每次写完 SDRAM后同样发出预充电命令 关闭本页 防止在同一个 Bank内打开两页。

2.4 DSP/SDRAM接口性能

33MHz 32位的 PCI总线理论极限速度是 132MB/s，实际速度要有一些折扣。由于 FPGA访问 DSP和 SDRAM理论带宽分别有 300MB/s和 600MB/s，因此 DSP/SDRAM接口的瓶颈在 PCI端。在研华 MIC3353主板上 主机无其他任务 重复访问 DSP内部一段 64K数据的测试环境下，接口的 DMA读速度有 90MB/s, DMA写有 38MB/s。与此同时，Bittware的同类型板卡 Tiger-6U-cPCI的 DMA读速度是 86MB/s, DMA写速度最高可以达到 40MB/s两者的 DSP接口访问速度基本相同，但是在 Bittware的设计里 主机要访问 SDRAM必须要借助 DSP的 SDRAM控制器 占用 DSP的一个 FLYBY通道 会影响 DSP的正常运转。本系统提供了一个主机直接访问 SDRAM的接口。

本文首先提出了一个通用 DSP系统的设计方案，主要给出了 FPGA在系统中的位置和作用。然后简要介绍了 FPGA的各个功能模块，着重针对 DSP和 SDRAM接口进行了讨论 针对数据宽度和时钟速率不匹配的特点 提出双状态机 +Cache的设计结构 给出相应的 FPGA设计框图和设计思路。比较国际上知名的其他板卡 本系统的 DSP接口的访问速度已经达到了较高水平，一个高访问速率的主机接口的建立 可以使得系统运行中的主机控制 DSP过程尽可能少地影响 DSP系统的运行。而一个高访问速率的 SDRAM接口的建立，也为主机和处理板间大容量数据交换提供了可能，这一点在进行数据存储和雷达信号处理中尤其有用。该设计已经被应用于某大型信号处理系统 取得了良好的效果。

参考文献

[1]刘书明 罗勇江 .ADSP TS20X系列 DSP原理与应用设计 北京 电子工业出版社，2007.

[2] GOLSON S.State machine design techniques for Verilog and VHDL.Synopsys Journal of High-Level Design, 1994, (9): 1-48.

[3] Altera.SDR SDRAM Controller White Paper V1.1.2002, 5.

[4]杨宗凯 数字专用集成电路的设计与验证 北京 电子工业出版社，2004.

嵌入式资源免费下载

总线协议：

1. 基于 PCIe 驱动程序的数据传输卡 DMA 传输
2. 基于 PCIe 总线协议的设备驱动开发
3. CANopen 协议介绍
4. 基于 PXI 总线 RS422 数据通信卡 WDM 驱动程序设计
5. FPGA 实现 PCIe 总线 DMA 设计
6. PCI Express 协议实现与验证
7. VPX 总线技术及其实现
8. 基于 Xilinx FPGA 的 PCIE 接口实现
9. 基于 PCI 总线的 GPS 授时卡设计
10. 基于 CPCI 标准的 6U 信号处理平台的设计
11. USB30 电路保护
12. USB30 协议分析与框架设计
13. USB 30 中的 CRC 校验原理及实现
14. 基于 CPLD 的 UART 设计
15. IPMI 在 VPX 系统中的应用与设计
16. 基于 CPCI 总线的 PMC 载板设计
17. 基于 VPX 总线的工件台运动控制系统研究与开发
18. PCI Express 流控机制的研究与实现
19. UART16C554 的设计
20. 基于 VPX 的高性能计算机设计
21. 基于 CAN 总线技术的嵌入式网关设计
22. Visual C 串行通讯控件使用方法与技巧的研究
23. IEEE1588 精密时钟同步关键技术研究
24. GPS 信号发生器射频模块的一种实现方案
25. 基于 CPCI 接口的视频采集卡的设计
26. 基于 VPX 的 3U 信号处理平台的设计
27. 基于 PCI Express 总线 1394b 网络传输系统 WDM 驱动设计
28. AT89C52 单片机与 ARINC429 航空总线接口设计

VxWorks：

1. 基于 VxWorks 的多任务程序设计
2. 基于 VxWorks 的数据采集存储装置设计
3. Flash 文件系统分析及其在 VxWorks 中的实现
4. VxWorks 多任务编程中的异常研究
5. VxWorks 应用技巧两例
6. 一种基于 VxWorks 的飞行仿真实时管理系统
7. 在 VxWorks 系统中使用 TrueType 字库
8. 基于 FreeType 的 VxWorks 中文显示方案
9. 基于 Tilcon 的 VxWorks 简单动画开发
10. 基于 Tilcon 的某武器显控系统界面设计
11. 基于 Tilcon 的综合导航信息处理装置界面设计
12. VxWorks 的内存配置和管理
13. 基于 VxWorks 系统的 PCI 配置与应用
14. 基于 MPC8270 的 VxWorks BSP 的移植
15. Bootrom 功能改进经验谈
16. 基于 VxWorks 嵌入式系统的中文平台研究与实现
17. VxBus 的 A429 接口驱动
18. 基于 VxBus 和 MPC8569E 千兆网驱动开发和实现
19. 一种基于 vxBus 的 PPC 与 FPGA 高速互联的驱动设计方法
20. 基于 VxBus 的设备驱动开发
21. 基于 VxBus 的驱动程序架构分析

Linux：

1. Linux 程序设计第三版及源代码
2. NAND FLASH 文件系统的设计与实现
3. 多通道串行通信设备的 Linux 驱动程序实现
4. Zsh 开发指南-数组
5. 常用 GDB 命令中文速览
6. 嵌入式 C 进阶之道
7. Linux 串口编程实例
8. 基于 Yocto Project 的嵌入式应用设计
9. Android 应用的反编译
10. 基于 Android 行为的加密应用系统研究
11. 嵌入式 Linux 系统移植步步通
12. 嵌入式 CC++语言精华文章集锦
13. 基于 Linux 的高性能服务器端的设计与研究
14. S3C6410 移植 Android 内核
15. Android 开发指南中文版

16. 图解 Linux 操作系统架构设计与实现原理（第二版）
17. 如何在 Ubuntu 和 Linux Mint 下轻松升级 Linux 内核
18. Android 简单 mp3 播放器源码
19. 嵌入式 Linux 系统实时性的研究
20. Android 嵌入式系统架构及内核浅析
21. 基于嵌入式 Linux 操作系统内核实时性的改进方法研究
22. Linux TCP IP 协议详解
23. Linux 桌面环境下内存去重技术的研究与实现

Windows CE：

1. Windows CE.NET 下 YAFFS 文件系统 NAND Flash 驱动程序设计
2. Windows CE 的 CAN 总线驱动程序设计
3. 基于 Windows CE.NET 的 ADC 驱动程序实现与应用的研究
4. 基于 Windows CE.NET 平台的串行通信实现
5. 基于 Windows CE.NET 下的 GPRS 模块的研究与开发
6. win2k 下 NTFS 分区用 ntldr 加载进 dos 源代码
7. Windows 下的 USB 设备驱动程序开发
8. WinCE 的大容量程控数据传输解决方案设计
9. WinCE6.0 安装开发详解
10. DOS 下仿 Windows 的自带计算器程序 C 源码
11. G726 局域网语音通话程序和源代码
12. WinCE 主板加载第三方驱动程序的方法
13. WinCE 下的注册表编辑程序和源代码
14. WinCE 串口通信源代码
15. WINCE 的 SD 卡程序[可实现读写的源码]
16. 基于 WinCE 的 BootLoader 研究

PowerPC：

1. Freescale MPC8536 开发板原理图
2. 基于 MPC8548E 的固件设计
3. 基于 MPC8548E 的嵌入式数据处理系统设计
4. 基于 PowerPC 嵌入式网络通信平台的实现
5. PowerPC 在车辆显控系统中的应用
6. 基于 PowerPC 的单板计算机的设计

7. 用 PowerPC860 实现 FPGA 配置
8. 基于 MPC8247 嵌入式电力交换系统的设计与实现
9. 基于设备树的 MPC8247 嵌入式 Linux 系统开发
10. 基于 MPC8313E 嵌入式系统 UBoot 的移植

ARM：

1. 基于 DiskOnChip 2000 的驱动程序设计及应用
2. 基于 ARM 体系的 PC-104 总线设计
3. 基于 ARM 的嵌入式系统中断处理机制研究
4. 设计 ARM 的中断处理
5. 基于 ARM 的数据采集系统并行总线的驱动设计
6. S3C2410 下的 TFT LCD 驱动源码
7. STM32 SD 卡移植 FATFS 文件系统源码
8. STM32 ADC 多通道源码
9. ARM Linux 在 EP7312 上的移植
10. ARM 经典 300 问
11. 基于 S5PV210 的频谱监测设备嵌入式系统设计与实现
12. Uboot 中 start.S 源码的指令级的详尽解析
13. 基于 ARM9 的嵌入式 Zigbee 网关设计与实现
14. 基于 S3C6410 处理器的嵌入式 Linux 系统移植

Hardware：

1. DSP 电源的典型设计
2. 高频脉冲电源设计
3. 电源的综合保护设计
4. 任意波形电源的设计
5. 高速 PCB 信号完整性分析及应用
6. DM642 高速图像采集系统的电磁干扰设计
7. 使用 COMExpress Nano 工控板实现 IP 调度设备
8. 基于 COM Express 架构的数据记录仪的设计与实现
9. 基于 COM Express 的信号系统逻辑运算单元设计
10. 基于 COM Express 的回波预处理模块设计
11. 基于 X86 平台的简单多任务内核的分析与实现
12. 基于 UEFI Shell 的 PreOS Application 的开发与研究