

基于 X86 32 位 CPU 的开放式实验平台设计

姚放吾^{1,2}, 王嘉平¹

(1. 南京邮电大学 计算机学院, 江苏 南京 210003;
2. 南京邮电大学 计算机技术研究所, 江苏 南京 210003)

摘要:开发开放式实验平台,不仅能实现与现有教学内容相匹配,而且能让学生深入理解和掌握 X86 32 位 CPU 的工作原理和接口技术,对提高教学质量具有非常重要的意义。文中设计的开放式平台,基于 486 系列 CPU,采用 Verilog 语言实现总线时序和控制逻辑,具有 8 位/16 位/32 位操作特性以及可扩展功能。软件平台包括 PC 宿主机集成操作环境软件和开放平台目标机固件程序。集成环境调试软件支持汇编语言实模式程序和保护模式程序,也支持 C 语言程序的编辑、编译和符号化调试。

关键词:32 位微处理器;开放式;平台;微机实验

中图分类号:TP368.2

文献标识码:A

文章编号:1673-629X(2012)06-0123-04

Open Experiment Platform Design Based on X86 32-bit CPU

YAO Fang-wu^{1,2}, WANG Jia-ping¹

(1. College of Computer, Nanjing University of Posts & Telecommunications,
Nanjing 210003, China;
2. Institute of Computer Technology, Nanjing University of Posts and
Telecommunications, Nanjing 210003, China)

Abstract: The development of open experimental platform can not only realize the match with the existing teaching content, but also make students better understand and master the Working principle and interface technology of 32-bit X86 CPU. What's more, it has the extremely vital significance to improve the quality of teaching. Open platform based on 486 series CPU is designed, on which the bus timing and control logic are achieved using the Verilog language, with 8-bit/16-bit/32-bit operating features and extensibility. Software platform includes the integration operational environment software on PC host machine and the target machine firmware program on open platform. Debugging software in the integrated environment supports real-mode assembly language program, a protected mode program and a C language program editor with the compiler and symbolic debugging.

Key words: 32 bit CPU; open; platform; microcomputer experiment

0 引言

近年来,国内大专院校各个专业讲授《微机原理》课程的内容已从原先的 8088 16 位 CPU 过渡到 32 位 CPU 结构,包括 386、486^[1]、Pentium 系列等。但是实验设备与环境还停留在早期的 16 位 CPU 阶段,无法满足新的教学实验的要求。为了和现有教学内容相匹配,使学生深入理解和掌握 X86 32 位 CPU 的工作原理和接口技术,开发相应的具有开放特性的实验平台

对提高教学质量将起到非常重要的作用。

开放式微机实验平台具有独立的 X86CPU,数据总线宽度 32 位,允许单次总线周期实现 8 位/16 位/32 位存储器和 I/O 操作;具有开放性和可扩展性;具有控制底层 I/O 接口^[2]的特性;Windows 平台集成操作软件支持汇编语言实模式、保护模式编程^[3,4]与符号化调试,支持 C 语言编程、I/O 操作与符号化调试。

1 硬件平台设计

硬件平台总体框架如图 1 所示。32 位 CPU 采用 486DX^[5]系列产品,FLASH 和 SRAM 分别存放系统固件和用户程序;总线缓冲和数据交换电路用于实现与不同宽度设备的数据连接;USB 接口和串口用于和 PC 机连接通信,在上位机集成操作软件的控制下,完成实验程序的下载及远程调试^[6]。

收稿日期:2011-10-31;修回日期:2012-02-02

基金项目:江苏高校优势学科建设工程资助项目(yx002001);江苏省企校合作项目(2010 外 42)

作者简介:姚放吾(1953-),男,教授,硕士生导师,研究方向为并行计算机及其体系结构、嵌入式技术和计算机在通信中的应用;王嘉平(1987-),女,硕士研究生,研究方向为计算机在通信中的应用。

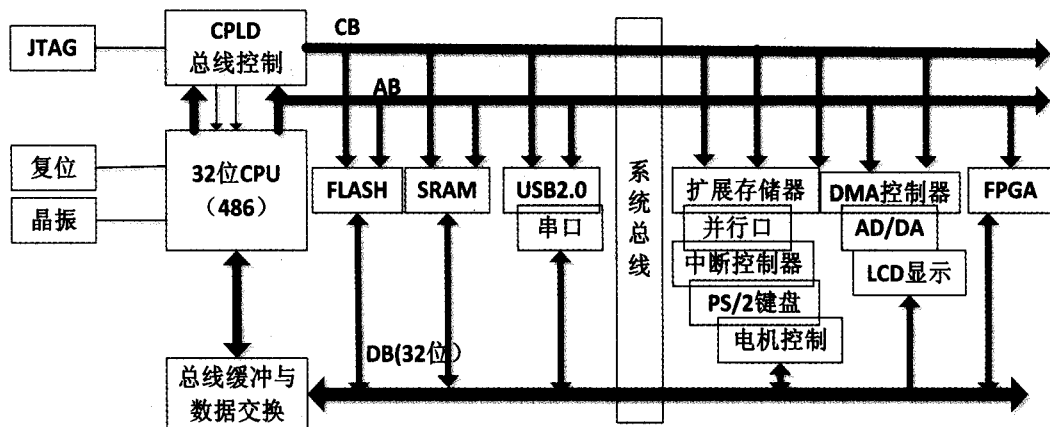


图1 硬件平台总体框图

为了使系统具有通用性、开放性,除了常用接口单元以外,设计了系统总线用于扩展功能,应用FPGA技术实现接口单元的高层次设计,以适应不同层次的教学需要。

1.1 CPU总线控制器设计

CPU总线控制器是整个系统的核心,其作用如同计算机主板上的芯片组。主要功能是对不同类型的总线周期生成相应的控制时序。总线周期包括复位操作、存储器读写和I/O读写、等待状态插入、中断响应、停机、总线请求和应答等。由于486CPU^[7]具有动态改变数据总线访问宽度的特性,即能够动态以8位、16位和32位宽度进行总线操作,因此存储器和I/O总线周期又细分为8位、16位和32位的读写总线周期。此外,为使32位CPU数据总线能够与不同位宽的存储器和I/O接口连接,设计了数据交换电路控制逻辑。整个总线控制器采用Verilog语言编写+Altera CPLD实现。图2是总线控制器产生的总线周期时序示意图。

系统复位之后,总线控制逻辑检测每个新的总线周期的产生,当ADS=0时表示一个新的总线周期开始,然后时序电路以此同步开始工作,依据CPU产生的总线状态信号使用状态机分别产生各个总线周期的时序,包括开始、结束以及相应的控制信号。这期间,如果需要插入等待状态,则产生READY=0的信号通知CPU插入等待状态,在不需要插入时将该信号置1以结束当前总线周期。

因为486CPU数据总线允许以三种不同总线宽度访问外围部件,于是需要根据访问要求适时产生8位和16位数据访问的总线控制信号BS8和BS16(BS8或BS16置0),没有BS8和BS16信号则表示32位访问周期。这样设计的主要目的是为了读写8位、16位外围设备

数据时软件应用具有更好的兼容性,否则相邻两个单元需要采用4的倍数地址才能访问。因为32位CPU访问外部时,每个单元地址对应一个数据字节,如果是8位数据宽度访问,地址A1A0=00--11四种状态对应的单元数据分别出现在数据总线D7-D0、D15-D8、D23-D16和D31-D24,此时如果不使用BS8信号,每次8位数据访问实际使用的都是32位总线周期,为了使数据都出现在D7-D0端以便正确访问外围部件,就要求编程时相邻两个单元的地址要加4。而使用BS8信号表示它是一个8位总线周期,虽然数据依次出现在对应的四组8位总线中,但是通过总线数据交换电路,可将其它三组8位数据交换至D7-D0中,这样就使软件能够很好的兼容8位指令操作。

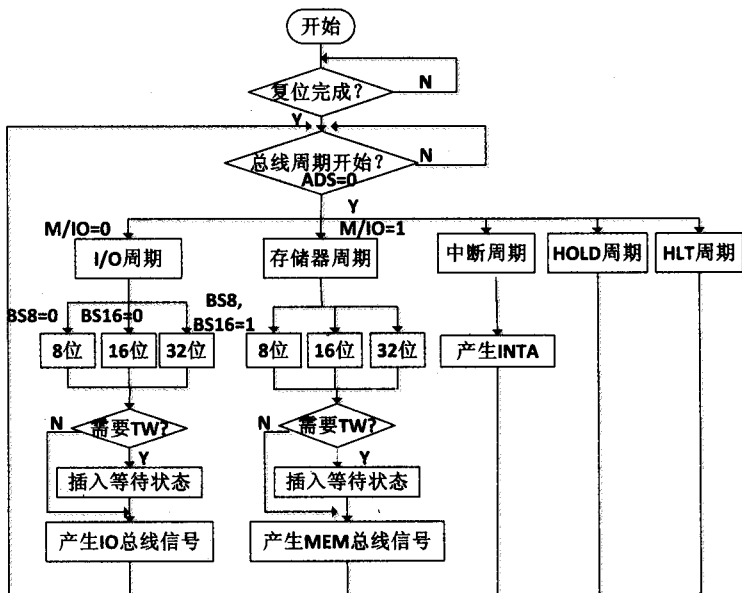


图2 总线周期时序流程

1.2 8/16/32位数据交换控制电路设计

486CPU的数据总线为32位,当它与32位存储器和I/O连接时,将32位数据总线与外围部件的D31-D0直接相连,即可实现单次总线周期传送32位的操作。除此之外,CPU允许动态改变数据总线为8位或

16 位传送宽度,但是与 8 位或 16 位部件连接,为了保证软件兼容,需要考虑字节数据和字数据的交换功能。假设一个 32 位指令操作如 MOV EAX, 0FFFFFFFH, 如果选择 8 位对齐总线操作,CPU 将产生 4 个 8 位总线周期,把 32 位数据分四次在四组字节数据总线上分时传送,如果选择 16 位对齐总线操作,CPU 将产生 2 个 16 位总线周期,将 32 位数据分两次在 D15-D0 和 D31-D16 分时传送。为了保证每次 8 位操作其数据都出现在外部总线 ED7-ED0,每次 16 位操作其数据都出现在外部总线 ED15-ED0,需要使用数据交换功能实现此目的。

486CPU 有四个字节允许信号 BE0, BE1, BE2, BE3,当其有效时,表示有效数据出现在对应的一组 8 位数据总线中,如 BE0 对应数据出现在 D7-D0 端 (BE0=0),第二个周期数据出现在 D15-D8 端 (BE1=0),第三个周期数据出现在 D23-D16 端 (BE2=0),第四个周期数据出现在 D31-D24 端 (BE3=0)。通过字节交换电路,将第二到第四个周期出现的数据依次交换至 D7-D0。16 位操作 (BS16=0) 时,第一个周期 BE1=BE0=0,数据出现在 D15-D0 端,第二个周期 BE3=BE2=0,数据出现在 D31-D24 端,通过字交换电路,将第二个周期出现的数据交换至 D15-D0 端。数据交换电路见图 3。

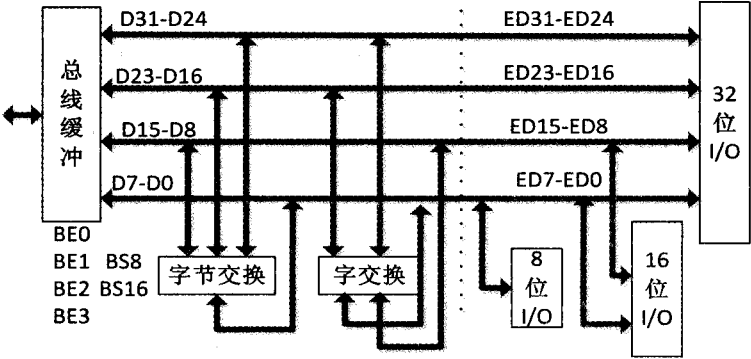


图 3 数据交换控制电路框图

1.3 通信接口设计

通信接口包括 USB2.0 和 UART,硬件平台通过该接口与 PC 机连接,用于程序的下载和远程调试。USB2.0 采用 CH372 设计, UART 采用兼容 16550 的接口芯片设计。通信接口采用中断方式与 486CPU 交换数据,但 USB 和串口每次产生中断的条件有所不同,USB 接口采用块传输方式,每传输 64 个字节产生一次中断,而串口每传输一个字符中断一次。因此,系统固件在中断处理程序中对两种类型的数据要分别处理。

1.4 DMA 和中断电路设计

DMA 采用 8237,DMA 通过三态缓冲电路与系统总线连接,使用 DMA AEN 信号控制 CPU 总线的占用和浮空。因为外部扩展存储器是由 4 个 8 位存储体构成的 32 位结构,允许 8 位、16 位、32 位读写操作,而 8237 是 8 位操作,为了实现对外部 8 位存储体的访问操作,设计了存储器选体逻辑,以保证 DMA 操作期间对连接 D7-D0 端存储器的正确读写。

中断控制器使用 8259。8259 与 CPU 连接时,要处理好两个信号 INT 和 INTA。INT 是 8259 向 CPU 申请中断的请求信号,它与 CPU 的 INTR 端连接,INTA 是中断响应信号,当 8259 连续接收到两个 INTA 信号,将会把中断类型码送至数据总线 D7-D0,使 CPU 最终转入相应的中断服务程序。486 CPU 采样到中断请求之后,如果满足中断响应条件,将产生中断响应周期信号,但是它并不产生 INTA 信号,因此设计了相关逻辑,一旦检测到中断响应周期,使用状态机连续产生两个 INTA。

2 软件平台设计

软件平台包括 PC 宿主机集成操作环境软件和开放平台目标机固件程序。宿主机和目标机程序相互配合,使用户在 Windows 98/XP/7 环境下实现汇编程序和 C 程序的编译链接、下载、运行及远程调试。调试软件^[8,9]基于窗口和源码级显示原理设计,无论是汇编语言和 C 语言,都可在显示窗口中直接观察到以变量名命名的显示结果,摒弃了早期 DEBUG 调试软件 16 进制的显示方式,进一步方便了用户使用。

2.1 固件设计

固件采用汇编语言编写,主要模块包括:面向 CPU 底层的调试功能;用于和 PC 通信的 USB 及串口通信功能;命令解释与处理功能;实模式和保护模式切换及现场恢复功能等。其模块层次结构如图 4 所示。

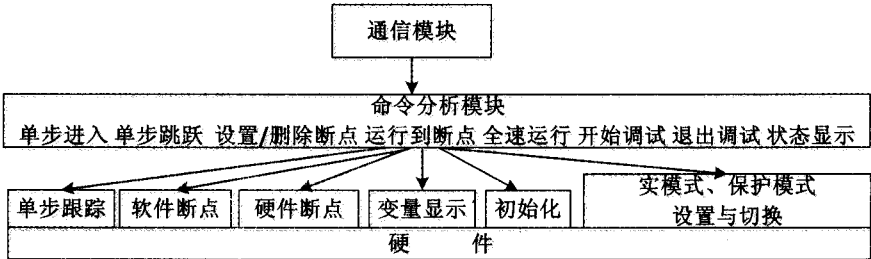


图 4 固件模块层次结构

调试功能分为单步调试和断点调试^[10,11],单步调试使用 CPU 标志寄存器 EFLAG 中的陷阱标志 T 实

现。当需要单步执行一条用户指令时,由宿主机将对应用户指令的 IP 地址下发,然后由固件将 T 标志置 1 后转去执行用户指令,CPU 执行完用户指令即转入类型 1 中断,在 1 型中断处理程序中将 CPU 现场内容上传给 PC,PC 集成软件做出相应格式化处理后显示。断点调试使用软件断点和硬件断点两种方法实现。软件断点通过在需要设置断点的指令位置用单字节断点指令 0CCH 置换原有指令内容实现,程序运行一旦遇到断点指令即转入类型 3 的断点中断处理程序,将 CPU 现场内容上传给 PC,经格式化处理后显示,当断点取消或继续运行时则恢复原有指令代码内容。硬件断点无需插入 0CCH 指令,只要将断点的 32 位线性地址写入 CPU 内部 4 个断点寄存器之一,当程序运行到指令地址与设置的断点地址相同时,产生类型 1 中断,进入 1 型中断后的处理过程与软件断点处理类似。硬件中断因为无需插入断点指令,因此可以在 ROM 空间和数据单元地址设置断点,使用灵活方便,但是由于断点寄存器数量的限制,所以往往将两种断点机制综合使用。

2.2 集成环境软件设计

集成环境操作软件运行于 PC Windows 平台,采用 VC6 开发。其主要功能包括:支持多文档关键字着色的文件编辑功能;使用中介程序和管道技术对工具软件^[12] MASM/LINK/BCC/TLINK 的输出结果进行重定向,将输出信息在用户窗口中显示;分析用户程序 EXE 文件的存储映像,获取代码段重定位信息,提取代码段和数据段内容;建立语句行与实际指令地址之间的对应关系,实现程序调试的各个功能,并且跟踪显示汇编指令或 C 语句单步运行的轨迹;建立变量与存储单元的对应关系,实现变量的显示与修改;通过 USB 驱动程序和串口控件实现与目标机的通信等。整个软件设计成主程序+动态链接库(DLL)的结构,上面所述功能分布在这几个模块当中。

其框图如图 5 所示。

3 结束语

开放式平台的建立,将 32 位主流微机技术和教学基础理论紧密结合,对于提高教学质量将起到很好的

作用。目前系统已提交给合作单位使用,达到了较好的效果。随着计算机技术的发展和进一步普及应用,对开放式平台也提出了更高的要求,包括新型 32 位/64 位 CPU 的应用,以及系统接口技术和应用技术等,面对新技术的挑战,需要我们做更进一步的深入研究。

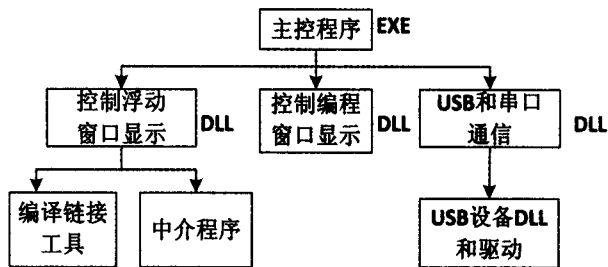


图 5 集成调试环境软件框图

参考文献:

- [1] Embedded Intel486 Processor Family Developer's Manual [EB/OL]. 1997. <http://www.intel.com>.
- [2] 仇玉章. 32 位微型计算机原理与接口技术[M]. 北京: 清华大学出版社, 2000.
- [3] 姚放吾, 丁 隰. 嵌入式远程调试器保护模式下调试功能的实现[J]. 计算机技术与发展, 2011, 21(4): 242-245.
- [4] 杨季文. 80x86 汇编语言程序设计教程[M]. 北京: 清华大学出版社, 1998.
- [5] Embedded IntelDX2 Processor [EB/OL]. 1997. <http://www.intel.com>.
- [6] 李红卫. 嵌入式远程调试工具的研究与实现[J]. 微计算机信息, 2009, 25(1-2): 87-89.
- [7] Embedded Intel486 Processor Hardware Reference Manual [EB/OL]. 1997. <http://www.intel.com>.
- [8] 王亚磊, 姚放吾, 罗 威, 等. 基于 386EX 的嵌入式软件调试器的设计与实现[J]. 计算机技术与发展, 2007, 17(s): 264-266.
- [9] 包 磊, 姚放吾. 基于远程设备的汇编源码调试器的设计与实现[J]. 计算机技术与发展, 2009, 19(11): 155-158.
- [10] 周洪波. 嵌入式系统软件开发环境中调试器的设计[J]. 微计算机信息, 2006, 22(5-2): 60-61.
- [11] Rosenberg J. How Debuggers Work [M]. New York: Wiley Computer Publishing, 1996.
- [12] 姚放吾, 金 鑫. 基于 Monitor 的 C 语言源代码调试器的设计与实现[J]. 计算机技术与发展, 2011, 21(6): 55-58.