

## 基于 ARM 体系的 PC/104 总线设计\*

占红武<sup>1</sup>, 胥芳<sup>1</sup>, 史伟民<sup>2</sup>

(1. 浙江工业大学 机电工程学院, 浙江 杭州 310032; 2 浙江理工大学 机械与自动控制学院, 浙江 杭州 310018)

**摘要:**在分析了 ARM 处理器系统在指令系统、存储器模型、中断系统、总线时序等方面的特点, 研究了 BM PC/AT 和 PC/104 总线规范的基础上, 提出了一种在 ARM 系统中高兼容性地实现 PC/104 总线的硬件电路方案, 以及与之相应的在 Linux 操作系统中的驱动设计, 并讨论了影响该兼容系统性能的相关因素。解决了在 ARM 系统中无法使用标准 PC/104 模板的难题。采用该方案, 有利于在嵌入式系统的设计中兼顾 ARM 处理器和 PC/AT 体系的优势。

**关键词:** ARM; PC/104; Linux; 设备驱动; 嵌入式应用

**中图分类号:** TP368

**文献标识码:** A

**文章编号:** 1001 - 4551(2008)01 - 0011 - 04

### PC/104 bus design based on ARM system

ZHAN Hong-wu<sup>1</sup>, XU Fang<sup>1</sup>, SHI Wei-min<sup>2</sup>

(1. College of Mechanical & Electrical Engineering, Zhejiang University of Technology, Hangzhou 310032, China;

2. College of Mechanical Engineering & Automatic Control, Zhejiang Sci-Tech University, Hangzhou 310018, China)

**Abstract:** A PC/104 bus compatible hardware circuit scheme and its device driver design in Linux were proposed, which has solved standard PC/104 device apply in ARM system, based on analyzing the instruction set, memory model, interrupt system, and bus timing, etc of ARM processor system, and studying BM PC/AT and IEEE PC/104 bus specification. The influence factors of the compatible PC/104 bus's capability were discussed. The method helps in the design of the embedded system and concurrently fetches ARM processor and advantage of PC/AT system.

**Key words:** advanced RISC machines (ARM); PC/104; Linux; device driver; embedded application

## 0 前言

PC/104<sup>[1,2]</sup>是一种专门为嵌入式应用而定义的工业控制总线规范,以 BM PC/AT 总线规范为基础的个人计算机及其外围设备在实际应用上取得了巨大的成就,IEEE-996 称之为工业标准架构 (ISA) 总线。PC/104 是 ISA 标准的延伸,在 IEEE-P996.1 中,定义为兼容 PC 嵌入式模块标准。PC/104 总线源于实践发展的需要,同时得益于 PC 技术的迅速发展,由于其具有开发环境友好、兼容芯片丰富、标准支持广泛等优势,受到了众多嵌入式产品生产商和系统集成商的欢迎,尽管现在 ISA 设备已不常见,但 PC/104 在嵌入式系统领域中却仍是通行的标准。

ARM 是当前嵌入式系统应用中的热门技术。由于 ARM 处理器的卓越性能、较低的价格以及业内半导体厂商的广泛支持,在嵌入式环境中应用广泛。

ARM 在设计上基于 SoC 思想,对每一个具体应用,优先考虑使用集成周边设备可能性最大的单一芯片,但在板级扩展方面仍然缺乏能被广泛接受的新工业标准。

本研究探讨基于 ARM 体系的 PC/104 总线设计。

## 1 PC/AT 模型和 PC/104 总线

### 1.1 总线信号

PC/104 是一种 8/16 位总线系统,总线时钟为 6 MHz~8 MHz ( $\pm 500$  ppm),周期的占空比为 50%  $\pm$  5%,4 个或 5 个时钟脉冲即可完成一次完整的总线访问<sup>[3,4]</sup>。在嵌入式系统开发中常用以下信号:

#### (1) 地址和数据信号线

BALE 总线地址锁存使能信号线。由平台 CPU 驱动,当 ISA 扩展卡或 DMA 控制器占用总线时,它也被置为逻辑 1。

SA <19: 0>。低 20 根地址信号线,由当前总线的拥用者驱动。

LA <23: 17>。锁存地址信号线,用来访问 16 MB 的存储器地址空间。由当前总线拥有者或 DMA 控制器驱动。

SD <15: 0>。数据信号线。

(2) 总线周期控制信号线

MEMR#。存储器读信号线。

MEMW#。存储器写信号线。

DR# I/O。I/O 读信号线。

DW# I/O。I/O 写信号线。

(3) 中断和 DMA 信号线

RQ<sub>x</sub>中断请求信号线允许扩充卡请求平台 CPU 提供的中断服务。

DRQ<sub>x</sub>DMA 请求信号线由 I/O 资源将其驱动为有效状态,来请求平台 DMA 控制器服务。

DACK<sub>x</sub># DMA 应答信号线由平台 DMA 控制器驱动为有效状态以选中请求 DMA 传送周期的 I/O 设备。

TC 终端计数信号线由平台 DMA 控制器驱动,以表明所有的数据已经被传送。

## 1.2 端口地址和中断系统

PC/AT 系统使用不同的指令访问存储器空间和 I/O 端口空间,访问存储器空间时 PC/104 总线驱动 MEMR#和 MEMW#信号,访问 I/O 端口空间时 PC/104 总线驱动 DR#和 DW#信号。两个空间使用同一套地址线,但是,由于端口空间只有 64 k 有效,进行 I/O 端口访问时地址线上只有 SA <15: 0>是有效的。

PC/AT 系统有专门的端口访问指令用于实现端口访问, Linux 用户程序只需以 root 权限调用 ioperm() 指定需操作的端口地址范围,之后即可自由访问这些端口。由于用于访问端口的一组宏实际上直接对应 CPU 的端口访问指令,存取效率很高。

PC/104 总线上提供了从 RQ2 到 RQ7 共 6 个中断输入信号,在 PC/AT 系统中这些信号由中断控制器 8259 统一管理。外部设备可以通过操作 RQ 线,迫使 CPU 执行预先安排的中断处理例程以保证突发事件的及时响应。由于大部分中断输入资源被系统母板占用, PC/AT 系统中真正可用于扩展模板的硬件中断线很少,实际使用时多采用共享电路方式进行中断线复用,如图 1 所示。

## 1.3 PC/104 总线周期

PC/104 总线周期分为 CPU 驱动和 DMA 控制器驱动两大类,其中 CPU 驱动的包括存储器读/写总线

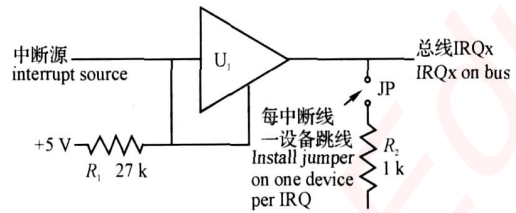


图 1 PC/104 总线中断信号线共享

周期和 I/O 读/写总线周期, DMA 控制器驱动的包括 DMA 存储器读/写总线周期。在嵌入式系统扩展中较常用的是 I/O 读/写总线周期。PC/104 总线的 I/O 读时序,如图 2 所示。其中,典型的  $t_{RD}$  应大于 30 ns,  $t_{AF}$  应大于 50 ns



图 2 PC/104 总线的 I/O 读时序

## 2 EP9315 (ARM 920T) 体系和 PC/104 硬件实现

ARM920T 是一个 32 位哈佛结构处理器,其处理器总线在信号和时序上与 PC/104 均有较大差异。对 SRAM、SDRAM、FLASH 等不同形式的存储器, EP9315 有一个统一的总线访问控制。引脚 CS<sub>n</sub> 用于存储器芯片的选通,可视为地址总线的高位线。EP9315 的 SRAM 控制器支持存储器的 8/16/32 位访问方式。

PC/104 总线使用标准 TTL 逻辑电平,而 EP9315 的处理器总线电平为 3.3 V。双电源总线收发器 74LVC245、74LVC16T245 等可用于实现电平转换。

### 2.1 中断和 DMA 的实现

EP9315 的引脚 NT[3: 0] 是外部中断 3 - 0 的输入。ARM920T 最多允许 64 个 RQ 或 FQ 信号,同时提供 32 个 RQ 硬件优先级和 2 个 FQ 优先级。内部中断可被设置为高电平有效、低电平有效,或者上升/下降沿有效。PC/104 的中断请求信号可以直接用于 ARM920T 的中断申请。考虑信号线的负载后, PC/104 设备中常见的共享中断信号线方式仍可沿用。

ARM920T 的 DMA 模块提供 12 个直接存储器存取通道。其中的 10 个用于周边设备与存储器间的存取访问, 2 个用于存储器到存储器的传送存取。每个 DMA 通道连接到 16 位的 DMA 请求总线,这些请求可能来自音频设备、UART 等,每个 DMA 请求及与之相关的物理地址都被独立地编程。ARM920T 的 DMA 在

应用上与 PC/AT 系统有较大的不同,少数应用中 PC/104 设备如需使用 DMA 资源,则必须完成硬件和驱动的设计。

## 2.2 端口映射和总线时序

PC/AT 系统中,由于端口访问具有兼容性好、效率高、易于实现等特征,在 PC/104 设备中的应用非常普遍。标准 ARM920T 系统可以有 I/O 存储器地址空间,但没有独立的 I/O 端口地址空间及相应的控制线,处理器也不存在专用于 I/O 访问的指令。通过硬件电路可将一段 SRAM 存储器地址映射为 I/O 端口空间。ARM920T 的 SRAM 读时序,如图 3 所示。

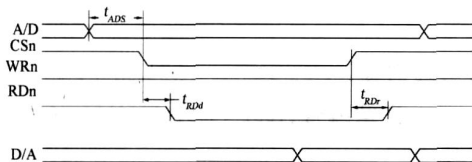


图 3 ARM920T 的 SRAM 读时序

PC/104 的端口地址线只有低 16 位有效 (64 k bytes), 简便起见,实现时直接使用 1 根 CSn 线用于端口映射。为避免处理器总线在访问其他地址空间时意外命中端口空间,CSn 线必须作为高位地址线参与地址译码。图 3 中 CSn 信号在其他地址信号建立后再经  $t_{ADS}$  才建立,而在其他地址信号无效前  $t_{RD}$  先无效。对比图 2 所示时序,新引入 CSna 和 RDna 信号,在外部设备速度允许的前提下,设法保证 PC/104 的标准时序。

设定 CSna 信号的无效时间较容易实现。另典型的  $t_{RD}$  需大于 30 ns,这可通过推迟 RDn 信号的建立时间来满足。以下 VHDL 代码对合适的周期时钟进行计数,以推迟 RDn 信号的建立:

```

... ..
PROCESS (TCLK)
BEGIN
IF (TCLK 'EVENT AND TCLK = '1 ') THEN
IF RDn = '0 ' THEN
    cnt <= 0;
ELSE
    cnt <= cnt + 1;
END IF;
IF cnt = 4 THEN
    RDna <= 0;
END IF;
END PROCESS;
... ..

```

以 16 位或 8 位方式访问 PC/104 的端口空间,在驱动中必须将 EP9315 的 SRAM 控制器设为相应位宽的访问方式。

## 3 Linux 系统下的访问实现

### 3.1 总线设备的 Linux 驱动

Linux 不允许用户程序直接访问存储器物理地址,总线设备如依赖 x86 的 I/O 端口访问模式工作,必须借助内核驱动。此时应将 PC/104 总线视为字符设备,在 /dev 下创建相应的节点<sup>[5,6]</sup>。

设备初始化时必须进行内存资源映射,并确定访问位宽:

```

static void PC104_init(void)
{
    ... ..
    // 将内存资源的物理地址映射到核心虚地址空间
    devPC104.io_base = ioremap(PC104_D_BASE, PC104_D_LEN);
    devPC104.io_width = 8; // 位宽
    PC104_configWidth(devPC104.io_width); // 寄存器
    ... ..
}

```

系统调用 read 实现如下:

```

ssize_t device_read (struct file * filp, char * buf, size_t count,
loff_t * f_pos)
{
    ... ..
    arg->data = readb((devPC104.io_base + arg->addr));
    return 1;
}

```

系统调用 write 实现如下:

```

ssize_t device_write (struct file * filp, const char * buf, size_t
count, loff_t * f_pos)
{
    ... ..
    writeb(arg->data, (devPC104.io_base + arg->addr));
    return 1;
}

```

### 3.2 Boot 阶段的访问

ARM 系统没有 PC/AT 模型中的 BDS 机制,如果在引导过程中需要访问 PC/104 总线设备,只能在 bootbader 中添加代码解决。由于此时 Linux 尚未启动,bootbader 可以直接访问物理地址。

## 4 结束语

利用 ARM 系统成本较低、功耗更小、启动速度快、指令效率高的特点,兼顾 PC/AT 系统和 PC/104 总线设备在产业化基础方面的优势,在 ARM 平台上构建 PC/104 总线,有助于利用现有资源,保护已有投资。

POpen (74, &remoteNode, 68)建立了一个使用 UDP的 SOCKET连接,本地端口是 74,远程端口是 68, &remoteNode为远程主机信息,在远程主机上使用 UDP的通信软件已经完成对 UDP协议的测试。还实现了基于 TCP的应用程序,通过“TcpSocket = TCPListen(8376)”语句建立一个 TCP连接,对端口 8376进行监听,在远程主机上使用 TCP通信软件,根据 ENC28J60的 IP地址和端口 8376,可以实现和以太网通信接口进行 TCP通信<sup>[5,6]</sup>。试验证明以太网通信接口的 UDP、TCP通信稳定可靠。以太网通信接口的软件流程图,如图 3所示。UDP为无连接的服务,只在数据准确性要求不高的场合使用。而 TCP协议面向连接,提供可靠的数据传输,但是实现起来较为复杂,代码量也占了整个协议栈代码量的大部分,由于嵌入式系统的应用一般对数据的准确性和实时性要求都比较高,大都使用基于 TCP的服务。本研究把以太网通信接口应用到电脑绣花机监控系统中就是通过采用基于 TCP的网络连接这一手段,利用端口 8376建立了一个 TCP连接,用于接收远程主机发送的花样文件以及向远程主机发送相关数据等。

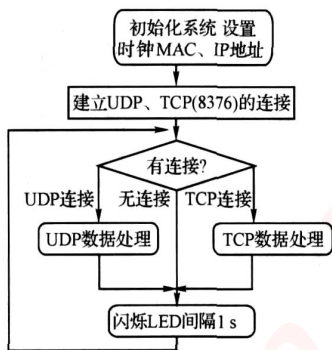


图 3 以太网通信接口软件流程图

(上接第 13页)

由于 ARM 与 PC/AT模型在处理器架构上有本质性的差异,在 ARM平台上构建的 PC/104总线在所有的具体应用中要做到与标准 PC/104规范完全兼容几乎是不可能的,但是在特定的应用需求及确定的操作系统下,这种努力是完全可行并且有效的。对于使用 PC/AT模型 I/O端口资源的 PC/104设备, Linux通过内核驱动以存储器地址实现访问,会造成访问效率的显著下降。另外,在使用遵守 ISA PnP(Plug and Play)规范的设备时,也存在较多的困难。

#### 参考文献 (Reference):

[1] 梁军兵,赵鸿鸣,金建祥.基于 ARM处理器的 PC/104处理器模块的开发[J].工业控制计算机,2004,17(12):

### 3 结束语

符合 IEEE 802.3协议的 ENC28J60不但能提供以太网通信的相应功能,而且体积小,能够简化设计,从而可以设计出目前最小的嵌入式应用以太网解决方案。本研究采用 PIC18F4550作为主控单片机,设计出了基于 ENC28J60的以太网通信接口,实现了以太网通信中的 ICMP、UDP、TCP协议。现在已经把该以太网通信接口应用到电脑绣花机监控系统中,试验证明该通信接口运行稳定、可靠。此外,基于 ENC28J60的嵌入式应用以太网解决方案电路连接简单,功能强大,并且完全符合未来工业以太网控制器的发展趋势,可以应用于通信(VoIP电话适配器)、库存管理、远程诊断、警报系统、保安及遥感、传动器等场合,有着广阔的应用前景。

#### 参考文献 (Reference):

[1] Microchip Technology Inc. ENC28J60 Stand Alone Ethernet Controller with SPI Interface[EB/OL]. [2004-10-20]. <http://www.microchip.com>.  
 [2] Microchip Technology Inc. Microchip TCP/IP stack[EB/OL]. [2002-05-16]. <http://www.microchip.com>.  
 [3] 王廷尧.以太网技术与应用[M].北京:人民邮电出版社,2005.  
 [4] 赵小敏,陈庆章. TCP/IP网络原理与技术[M].北京:高等教育出版社,2006.  
 [5] RAJENDRA Y, PRASHANT P, RAPHAEL F. A reservation-based CSMA protocol for integrated manufacturing networks[J]. IEEE Transactions on Systems, Man, and Cybernetics, 1994, 24(8): 1247-1258.  
 [6] 陈磊,冯冬芹,金建祥,等.以太网在工业应用中的实时特性研究[J].浙江大学学报:工学版,2004,38(6): 29-31. [编辑:李辉]

19-20.

[2] PC/104 Embedded Consortium. PC/104 Specification Version 2.5[M]. San Francisco: PC/104 Embedded Consortium, 2003.  
 [3] 杨素行.微型计算机系统原理及应用[M].北京:清华大学出版社,1995.  
 [4] Institute of Electrical and Electronic Engineers, Inc. IEEE P996, ISA Bus Timing Diagrams[M]. New York: Institute of Electrical and Electronic Engineers, Inc, 1999.  
 [5] 毛德操,胡希明. Linux内核源代码情景分析[M].杭州:浙江大学出版社,2001.  
 [6] 刘洋,李兰英. U-boot的启动流程分析[J].科技资讯,2006(29): 110-111. [编辑:李辉]