

P 模式。作者根据 I E E E 1 2 8 4 上规定的时序图进行了时序设计，而计算机上提供的 E P P 版本是 E P P 1 . 7 或 E P P 1 . 9 ，实际使用中它们没有不匹配的地方。

2 用 E P P 扩展 I S A 接口的总体设计

计算机主板上一般有 2 ~ 3 个 I S A 插槽，可以扩充一些 I S A 插卡。大多数基于 I S A 标准的微机数据采集和控制只用到了 I S A 接口的数据线、地址线、A E N、A L E、# I O R、# I O W 等信号，有的也用到了中断和 D M A 的信号。只要了解这些信号之间的时序关系，我们完全可以自己用逻辑器件“制造”出 I S A 接口，文献 [2] 介绍了用单片机扩展总线的技术，包括 I S A、S T D 总线等。作者曾经用并口的 S P P 模式和 8 0 C 1 9 6 单片机扩展出了计算机的 I S A 接口，但是 I O 读写速度只能达

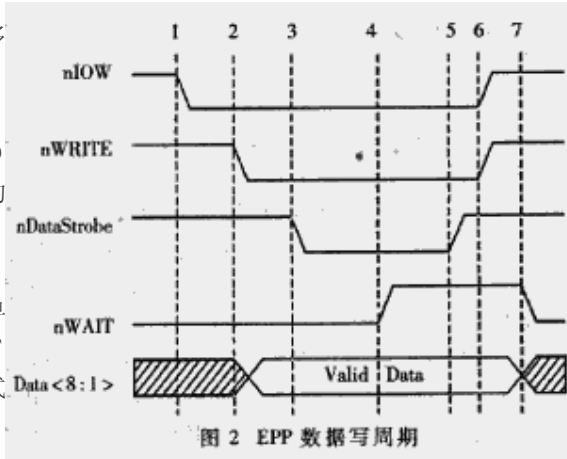


图 2 EPP 数据写周期

到 2 0 K B / s 左右，总体的效果不太理想。E C P 和 E P P 都能进行高速双向数据通讯，但是 E C P 的实现远比 E P P 复杂的多，其性能和 E P P 却大致相当，所以最终的方案采用了 E P P 模式。由于单片机是一个单任务的串行控制器，如果它只是用来扩展 I S A 接口，那么可以达到比较高的速度；否则，I S A 总线的速度会大大降低，最后变得失去使用价值，所以最终的外设芯片不能采用单片机，只能采用 D S P 或者大规模的可编程逻辑器件。作者选用了后一种方案。

用 E P P 扩展 I S A 口的硬件核心是一片可编程逻辑器件，如 C P L D ，它一方面负责与计算机通过 E P P 协议进行双向数据通讯，另一方面负责产生 I S A 接口时序，系统体系相当简洁而高效，电路原理如图 1 所示。E P P 的数据线 D 0 ~ D 7 和信号线 n W r i t e、n D s t r b、n A s t r b、n I n i t、n W a i t、I n t r 直接与 C P L D 的双向 I / O 线相连。另外，E P P 没有定义标准并口的第 1 2、1 3、1 5 三个引脚，这些引脚用户可以灵活使用。电路图上并口的第 1 3 脚和 C P L D 连了起来，可以提供其他的功能。C P L D 提供了 I S A 接口的 D 0 ~ D 7，A 0 ~ A 1 5，A L E，A E N，# I O R，# I O W，I R Q、I O R D Y 等信号。扩展的 I S A 接口提供了 1 6 根地址线，可以寻址 6 4 K 的 I O 空间，这比计算机所提供的 I O 空间 (1 K) 大了许多倍。用户可以专门设计具有 6 4 K 寻址能力的数据采集和控制板，也可以用只有 1 K 寻址能力的数据采集和控制板，在这种情况下，地址线的高 6 位被忽略了，但这并不影响系统的正常使用。

3 硬件操作方法

E P P 协议定义的并行口提供了四种传送周期：数据写周期、数据读周期、地址写周期和地址读周期。数据周期一般用于计算机和外设间的数据传送，地址周期一般用于传送地址、通道、命令和控制等信息。实际上，数据周期和地址周期并没有那么严格的界限，可以把地址周期看做另一种数据周期，二者并没有太大的区别。图 2

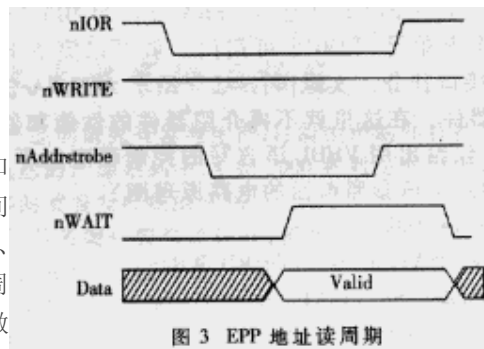


图 3 EPP 地址读周期

是EPP数据写周期的时序图,图中的nIOW信号实际上在进行EPP数据写时并不会产生,只不过是表示所有的操作都发生在一个IO周期内,只有这样,才能使EPP获得ISA总线的数据操作速度。图2中的nDataStrobe信号如果换为nAddStrobe信号,就是EPP地址写周期。图3是EPP地址读周期,也是发生在一个IO周期内。

EPP定义了一个计算机用于控制外设初始化的信号:nInit,如果用户不希望控制外设的初始化,则可以不处理这个信号;用户也可以挪用nInit信号做其他的用途,在本设计方案的地址周期中,如果nInit为高,则表示该地址对应ISA接口的高8位地址,否则,对应低8位地址。用这种办法解决了用8位的EPP地址扩展16位的ISA地址的难题。实践证明,这是一个方便实用的解决方法。

用EPP扩展ISA接口,最根本的任务是把EPP的数据读写周期快速地转化为ISA的IO读写周期。在ISA的时序中[3],时钟的频率是4.77MHz,典型的一个ISA周期要用4个时钟周期,即大概1μs的时间。在EPP的读周期中,EPP首先发出读命令,然后等待ISA的数据响应,如果ISA仍然以4个时钟来进行IO读操作,那么EPP很有可能会由于超时而发生时序错误,数据也必然会错。解决这个问题可以用提高ISA接口的时钟频率的办法,如提高到8MHz甚至是16MHz,但是这样的话ISA卡可能会来不及响应而发生数据错误,所以这种方法不可取;另一种方法是改造ISA接口的时序,使得既能满足ISA卡的时序要求,又不至于造成EPP的超时错误。仔细分析ISA的IO读时序,CPU在T1发出地址信号并发出ALE信号,在T2发出读命令,在T3采样READY信号,以决定是否产生等待周期,如果不需要等待,则在T4读取数据,完成整个读周期。可以发现,对于本系统,T1周期是可以省略的,因为CPLD可以在EPP的地址周期内设定要寻址的IO地址,而没有必要在ISA周期内再发送地址,这样ISA的IO读周期就从4个时钟减少到3个时钟;如果可以保证ISA卡设备可以在一个时钟内送出有效的数据,则T3也可以省略,这样一个ISA读周期实际上只占用了两个时钟,不会造成EPP的超时错误。对EPP数据写周期,因为CPLD可以先把数据写到缓冲中,首先保证EPP时序,再把数据从缓冲写到ISA设备中去,所以不会造成超时错误。ISA的读写时序经过这样的简化处理后,可以满足ISA设备和EPP两方面的时序要求。

4 CPLD的编程

系统选用的CPLD是ALTERA公司的MEP7064,它有64个宏单元,1250个可用门,就可以完成EPP和ISA的接口任务。文献[4]详细介绍了ALTERA公司的CPLD器件,在这里就不再介绍器件的性能和使用方法了。仅给出用VHDL语言写的控制程序如下(部分信号的意义请参照前面的电路原理图)。

```
Process (clk)

Type ISAType is (Idle, RD, WR);

Variable ISA: ISAType;

Variable IOR: STD_LOGIC;

Variable IOW: STD_LOGIC;
```

```
Variable EPPBuf: STD_LOGIC_VECTOR (7 downto 0);

Begin

Case ISA is

When Idle =>

If IOR= '1' then

# IORD <= '0';

IOR = '0';

ISA := RD;

Else if IOW= '1' then

# IOWR <= '0';

ISADat aBuf<=EPPBuf;

IOW:= '0';

ISA:=WR;

End if;

When RD =>

EPPBuf:=ISADat aBus;

# IORD<= '1';

ISA:=Idle;

When WR =>

# IOWR <= '1';

ISA:=Idle;

End case;

End;
```

End Process;

5 计算机对EPP / ISA的操作

计算机通过EPP协议用CPLD扩展出ISA接口,现有的ISA卡就可以通过ISA接口、CPLD和EPP协议间接地连到了计算机上。对于ISA卡的使用者而言,无论从硬件的角度还是从软件的角度来看,都好像是这块ISA卡直接插在计算机的ISA槽内,其速度也完全能够达到应用的要求。EPP协议的用户编程接口是协议定义的几个IO口地址。以并口基地址为378H为例,378H是SPP数据口,379H是SPP状态口,37AH是SPP控制口,37BH是EPP地址口,37CH是EPP数据口。对ISA卡的操作顺序是:首先通过设置nInit为高和写EPP地址口来设置ISA卡的高8位地址,再通过设置nInit为低和写EPP地址口来设置ISA卡的低8位地址,就可以通过EPP数据口对ISA卡进行读写操作了。作者本人做出的系统对单一的地址进行操作时,写操作的速度可以达到1000~1200KB/s,最高可以达到1310KB/s,读操作的速度可达800~1100KB/s,完全能满足数据采集和控制的要求。如果是对多个地址进行操作,由于设置地址要占用一个或者两个EPP地址周期,所以数据传输速度会有所损失。

6 与其它通讯方案的比较

(1) RS232串口:最通用的一种连接方法。但是它支持数据传输速率最大为10~20KB/s,对于一般的数据采集和控制系统而言显得有些慢。

(2) SPP:数据通讯速度比串口快,可以达到150KB/s。但是SPP用做数据输入时很麻烦,用多次IO才能完成一次完整的数据读取,速度要牺牲很多,况且外设的设计并不比EPP简单。所以,如果选择了并口方案,就不能选择SPP模式,除非用户仅仅是做数据输出并且对速度没有很高的要求。

(3) ECP:与EPP相比ECP最大的优势是它支持DMA操作,如果系统工作时有大批量的数据要传输,用ECP模式可以大大减轻计算机CPU的负担,提高系统的整体性能。但是获得ECP的高性能的代价是必须重新设计比EPP复杂得多的接口软件(指CPLD的控制软件),同时计算机软件方面还必须要编写硬件驱动程序,这对于一般的计算机应用系统开发者而言还是一个不小的困难。

(4) PCMCIA(the Personal Computer Memory Card Industry Association):发展了信用卡大小的外设与PC机连接的标准。最初,PCMCIA标准仅对于内存卡,现在已扩展到I/O设备。数据能以最大5MB/s的速率传输。但PCMCIA卡不支持DMA,这就增加了数据采集和CPU处理之间的时间。因而,目前的PCMCIA I/O卡需大容量缓存。PCMCIA设备的另一缺点是尺寸太小(最大尺寸86mm×54mm×10mm),不能用于控制数据采集系统中的一些模拟电路部分。

(5) USB(Universal Serial Bus):支持12Mbps的数据传输速度,支持127个外围设备,支持PNP(Plug and Play),支持热插拔,并且总线本身可以提供用户系统电源。信号传输采用差分方式,可以抑制比较强的共模干扰。USB具有很大的发展前途。在未来的计算机上,可能不再提供RS232串口,可能不再提供并口,

但绝对不可能没有U S B接口。U S B的使用不象并口那样简洁,它必须要有专用的接口芯片的支持才能用在系统中。不少芯片商已经提供了U S B的产品,如U S B H U B、U S B接口、U S B单片机等等。当然用户自己也可以把U S B协议写到P L D芯片中去,使接口和系统融为一体。

综上所述,采用E P P扩展计算机的I S A接口是一种新颖的计算机外设设计方案,它具有非常高的性能价格比,能够达到绝大多数基于I S A接口的数据采集和控制系统的通讯速度要求。这种方案大大扩展了笔记本电脑对于I S A接口设备的适应能力,省去了用户对扩展箱的需求。仅仅改写该系统的C P L D程序和计算机的控制程序就可以提供新的功能,如做成双I S A接口系统,或者改造成S T D、S T E总线系统等等,而系统的硬件不需做任何的改动。E P P的确是一种有前景的实用接口技术,值得微机外设设计者和使用者采用。

本文内容来自互联网,著作权归原作者所有。由电子零件城(<http://www.epcity.com/>)整理并制作成PDF文件,仅供个人学习之用,不得用于任何商业目的,否则后果自负。如果您认为本PDF文件侵犯了您的任何权利,请来信epcity@epcity.com通知,本站立即删除。

搜集整理: 电子零件城-笨笨兔 (QQ: 154502842) 2004-04-10