

PCI 总线接口芯片 CH365

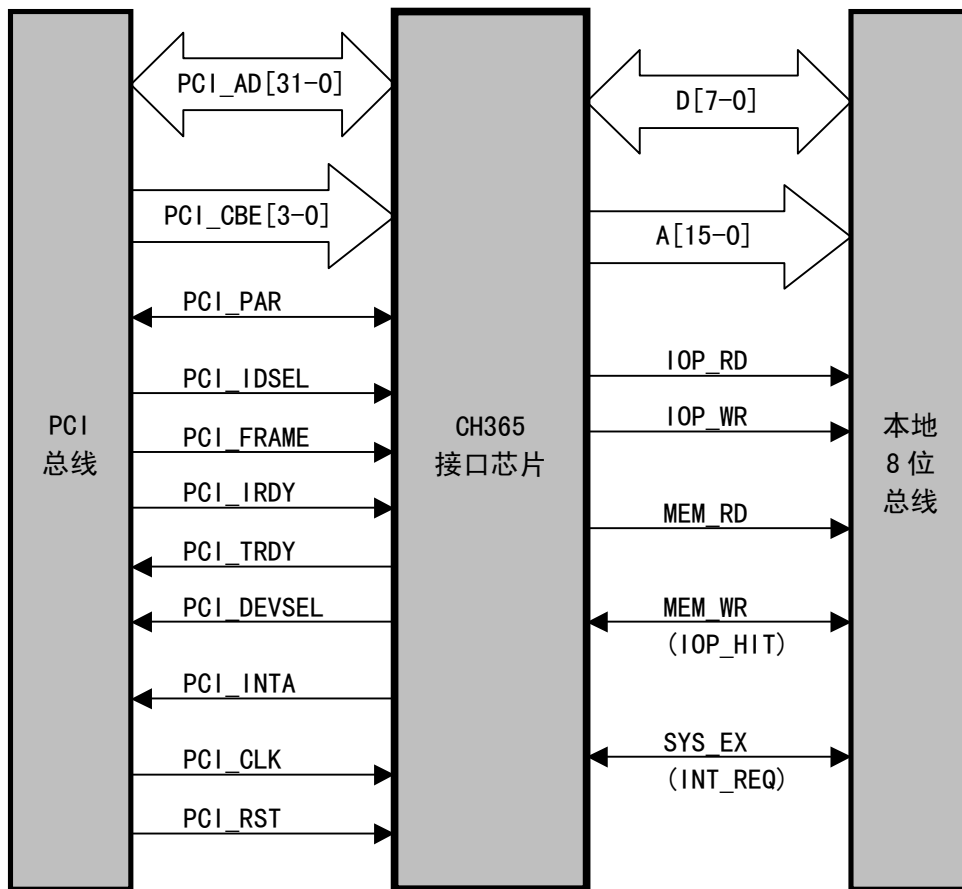
中文手册

版本: 1D

<http://wch.cn>

1、概述

CH365 是一个连接 PCI 总线的通用接口芯片, 支持 I/O 端口映射、存储器映射、扩展 ROM 以及中断。CH365 将 32 位高速 PCI 总线转换为简便易用的类似于 ISA 总线的 8 位主动并行接口, 用于制作低成本的基于 PCI 总线的计算机板卡, 以及将原先基于 ISA 总线的板卡升级到 PCI 总线上。PCI 总线与其它主流总线相比, 速度更快, 实时性更好, 可控性更佳, 所以 CH365 适用于高速实时的 I/O 控制卡、通讯接口卡、数据采集卡、电子盘、扩展 ROM 卡等。下图为其一般应用框图。

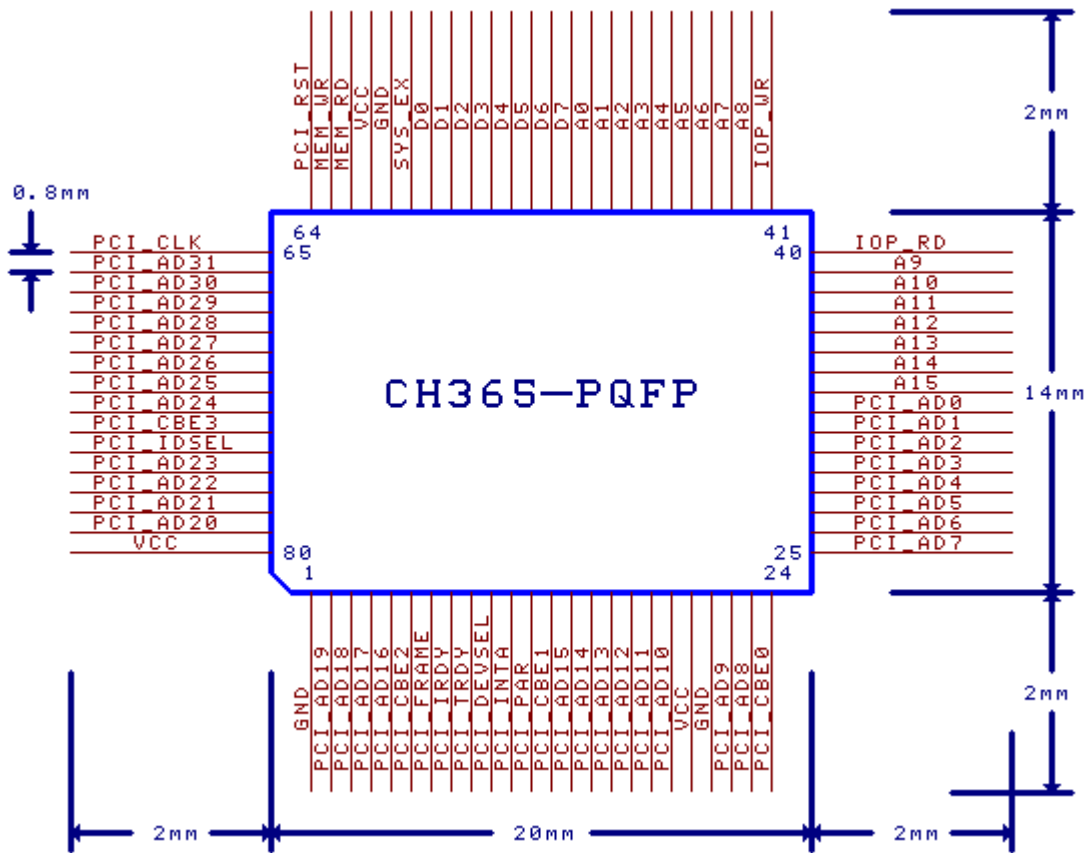


2、特点

- 实现基于 32 位 PCI 总线的从设备接口。
- 转换为主动并行接口: 8 位数据, 16 位地址, I/O 读和写, 存储器读和写。
- 可以设定 PCI 板卡的设备标识 (Vendor ID, Device ID, Class Code 等)。
- 支持以字节、字或双字为单位对 I/O 端口或者存储器进行读写。
- 非突发存取的实测速度可达每秒 7MB, 读写脉冲的宽度从 30ns 到 240ns 可选。
- 自动分配 I/O 基址, 支持长度达 240 字节的 I/O 端口。
- 支持本地硬件定址功能, 自由选择 I/O 地址, 在指定地址实现 I/O 端口。
- 直接升级 ISA 的 I/O 板卡到 PCI 总线, 完全不需要修改原 ISA 卡的相关软件。
- 直接映射支持容量为 32KB 的存储器 SRAM 或者扩展 ROM (Boot ROM)。
- 无需外接元器件扩容支持容量为 64KB 以及 128KB 的存储器或者扩展 ROM。

- 支持扩展 ROM 无硬盘引导, 支持闪存 Flash-Memory 在线升级。
- 可以提供扩展 ROM 应用的子程序库 BRM, 用于 BIOS 环境下用户界面显示及数据处理。
- 支持低电平有效的本地中断请求, 支持中断共享。
- 提供两线串行主机接口, 可以挂接类似 24C0X 的两线串口 EEPROM 器件。
- 内置 4 μ s 至 1ms 的硬件计时单元, 用于软件运行过程中作为延时参考。
- 大引脚间距 (0.8mm) 的 PQFP-80 封装, 与 CH361P 引脚兼容, 直接替代。
- 通用的驱动程序支持 Windows 98/ME/2000/XP, 通过 DLL 提供应用层 API。
- 芯片本身无需驱动程序即可工作, 升级 ISA 板卡可以不需要驱动程序。
- 多项专利技术, 简便易用, 低成本。

3、封装



| 封装形式 | 塑体宽度 | 引脚间距 | | 封装说明 | 订货型号 |
|---------|-------------|-------|---------|---------------|--------|
| PQFP-80 | 20mm x 14mm | 0.8mm | 31.5mil | 标准的 PQFP80 贴片 | CH365P |

4、引脚

4.1. 电源线

| 引脚号 | 引脚名称 | 类型 | 引脚说明 |
|------------|------|----|-------|
| 20, 61, 80 | VCC | 电源 | 正电源端 |
| 1, 21, 60 | GND | 电源 | 公共接地端 |

4.2. PCI 总线信号线

| 引脚号 | 引脚名称 | 类型 | 引脚说明 |
|--|-----------------------|-------------|--------------------|
| 64 | PCI_RST | 输入 | 系统复位信号线，低电平有效 |
| 65 | PCI_CLK | 输入 | 系统时钟信号线，上升沿有效 |
| 2-5 14-19 22-23 25-32 66-73 76-79 | PCI_AD31 ~PCI_ADO | 三态输出 及输入 | 地址和双向数据复用信号线 |
| 6, 13 24, 74 | PCI_CBE3 ~PCI_CBE0 | 输入 | 总线命令和字节使能复用信号线 |
| 12 | PCI_PAR | 三态双向 | 奇偶校验信号线 |
| 75 | PCI_IDSEL | 输入 | 初始化设备选择信号线，高电平有效 |
| 7 | PCI_FRAME | 输入 | 帧周期开始信号线，低电平有效 |
| 8 | PCI_IRDY | 输入 | 发起设备准备好信号线，低电平有效 |
| 9 | PCI_TRDY | 三态输出 | 目标设备准备好信号线，低电平有效 |
| 10 | PCI_DEVSEL | 三态输出 | 目标设备选中信号线，低电平有效 |
| 11 | PCI_INTA | 三态输出 | INTA 中断请求信号线，低电平有效 |

4.3. 本地端信号线

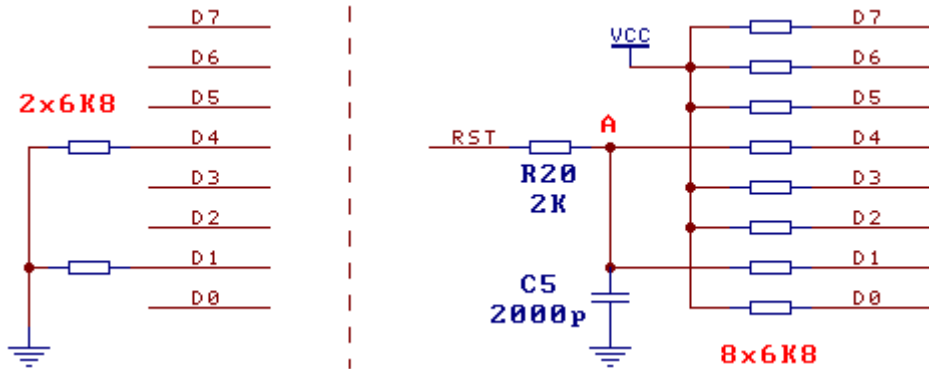
| 引脚号 | 引脚名称 | 类型 | 引脚说明 |
|----------------|---------|-------------|---|
| 51-58 | D7~D0 | 三态输出 及输入 | 8 位双向数据信号线，内置上拉电阻， D7 同时是两线串行接口的 SDA 信号线 |
| 33-39 42-50 | A15~A0 | 输出 | 16 位地址信号线，A15~A10 可以独立控制输出， A15 可以设定为两线串行接口的 SCL 信号线 |
| 40 | IOP_RD | 输出 | I/O 端口的读选通/使能，低电平有效 |
| 41 | IOP_WR | 输出 | I/O 端口的写选通/使能，低电平有效 |
| 62 | MEM_RD | 输出 | 存储器或扩展 ROM 的读选通/使能，低电平有效 |
| 63 | MEM_WR | 输出 | 存储器或扩展 ROM 的写选通/使能，低电平有效 |
| 复用 | IOP_HIT | 输入 | 本地硬件定址请求，低电平有效，内置上拉电阻 |
| 59 | SYS_EX | 输出 | 可以独立控制的输出信号线， 可以设定为两线串行接口的 SCL 信号线 |
| 复用 | INT_REQ | 输入 | 本地中断请求输入，低电平有效，内置上拉电阻 |

5、工作模式设定

为了在不增加引脚的前提下提供更多可用功能，CH365 对部分引脚进行复用，通过“工作模式设定”进行功能选择。“工作模式设定”的具体方法如下：将本地端 8 位数据信号线 D7~D0 采用上拉或者下拉的方式设定为所需的高电平或者低电平，CH365 被复位后根据这些信号线的默认状态设定工作模式以及参数；而这些信号线在作为 8 位数据总线被驱动时，因为一般外部设备的驱动电流不小于 1mA，所以上拉或者下拉不会影响其对数据总线的驱动；另外，CH365 仅在被复位后的 1μs 内一次性设定工作模式以及参数，所以，如果外部设备的驱动能力很小或者是 OC 集电极开路驱动，则可以仅在复位后的短时间内实现下拉，而在其余时间屏蔽下拉或者转换成上拉。

下图为工作模式设定的两种电路，CH365 已经在芯片内部为 8 位数据总线 D7~D0 连接了 40KΩ 的弱上拉电阻，如果在芯片外部连接下拉电阻，则数据线为低电平，否则为高电平。图中左边的电路适用于一般情况，D4 和 D1 被 6.8KΩ 电阻下拉为低电平，故数据线 D7~D0 上的设定值为 11101101（二进制数）；并且由于上拉和下拉的阻值较大，所以不会影响 CH365 和外部设备对数据总线的驱动。当

外部设备的驱动能力小于 1mA 或者是集电极开路驱动时，可以参考图中右边的电路。图中 RST 即 PCI 总线的复位信号，由于电容 C5 的作用，使得 A 点在 RST 复位完成后继续保持几个微秒的低电平，CH365 在此期间完成工作模式设定。图中右边的电路在 D7~D0 上的设定值为 11101101。



下表为设定工作模式和参数所对应的数值（1 即高电平，0 即低电平）。例如，D1=0 则可以定义 PCI 板卡的 ID，D3=0 则启用中断功能，D4=0 则启用本地硬件定址的功能，D3 和 D4 不能都为 0。

| 数据线 | 数据线的设定说明 | 设定值=0 | 设定值=1 |
|-----|---|-------------|-----------------|
| D0 | 设定系统复位后 A15 的默认值 | 复位后 A15 为 0 | 复位后 A15 为 1 |
| D1 | 选择 PCI 厂商和设备标识 ID | 由外部提供 ID | 使用 CH365 默认的 ID |
| D3 | 选择复用引脚 59 的功能 | INT_REQ | SYS_EX |
| D4 | 选择复用引脚 63 的功能 | IOP_HIT | MEM_WR |
| 其它 | 提供给产品制造商使用，数据线的状态可以从 PCI 设备配置空间 41H 中读取，例如：在 D2、D5、D6、D7 引脚上可选地连接下拉电阻，供应用程序识别板卡功能 | | |

6、寄存器

6.1. 基本约定

- 6.1.1. 属性简写：R=完全只读，W=可读可写，S=只读但可以事先设定，....=省略号。
- 6.1.2. 数据的数制：如果以 H 结尾则为十六进制数，否则为二进制数。
- 6.1.3. 数值的通配符以及属性：r=保留（禁止使用），x=任意值，....=省略号。

6.2. 配置空间 0FFH-00H

| 类别 | 地址 | 寄存器名称 | 寄存器属性 | 系统复位后默认值 |
|-----------------------------|--------------------------------|---------------------------------------|---------------------------------------|--------------------------------------|
| 标准 PCI 设备 配置 空间 | 01H-00H | 厂商标识: Vendor ID | SSSS | 4348H |
| | 03H-02H | 设备标识: Device ID | SSSS | 5049H |
| | 05H-04H | 命令寄存器: Command | RRRRRRRRRRRRRRWW | 0000000000000000 |
| | 07H-06H | 状态寄存器: Status | RRRRRRRRRRRRRRRR | 0000010000000000 |
| | 08H | 芯片版本: Revision ID | SS | 10H |
| | 0BH-09H | 设备类代码: Class Code | SSSSSS | 100000H |
| | 0FH-0CH | | SSSSSSSS | 00000000H |
| | 13H-10H | I/O 端口基址: I/O Base Address | RRRRRRRRRRRRRRRR WWWWWWWWRRRRRRRR | 0000000000000000 0000000000000001 |
| | 17H-14H | 存储器基址: Memory Base Address | WWWWWWWWWWWWWWWW WRRRRRRRRRRRRRRRR | 0000000000000000 000000000000x000 |
| | 2FH-18H | | SSSS... SSSS | 0000... 0000H |
| 33H-30H | 扩展 ROM 基址: ROM Base Address | WWWWWWWWWWWWWWWW WRRRRRRRRRRRRRRRW | 0000000000000000 0000000000000000 | |

| | | | | |
|----------------------|---------|--|-------------------------------------|-------------------------------------|
| | 3BH-34H | | SSSSSSSSSSSSSSSS | 000000000000000H |
| | 3FH-3CH | 中断号和中断引脚等： Interrupt Line & Pin | RRRRRRRRRRRRRRRR RRRRRRRRWWWWWWW | 0000000000000000 000000x00000000 |
| 控制 寄存 器和 杂项 | 40H | 芯片控制寄存器 | RRRRRWWW | rrrrr00x |
| | 41H | 8 位总线 D7-D0 输入端口 | RR | xxH |
| | 42H | 芯片状态寄存器 | RRRRRRRR | xxrxrxrx |
| | 43H | 保留 | (禁止使用) | (禁止使用) |
| | 其它 | 4FH-44H 是 43H-40H 的别名地址；0FFH-50H 保留，只读 00H | | |

6.3. I/O 空间 0FFH-00H（寄存器的实际地址为 I/O 端口基址加表中的偏移地址）

| 类别 | 偏移地址 | 寄存器名称和说明 | 寄存器属性 | 复位后的默认值 |
|----------------------------------|----------|----------------|----------|------------|
| 端口 | 0EFH-00H | 标准的本地 I/O 端口 | WW | 连接到 I/O 设备 |
| 存储 器 接 口 | 0F0H | A7-A0 地址设定寄存器 | WWWWWWWW | xxxxxxxx |
| | 0F1H | A15-A8 地址设定寄存器 | WWWWWWWW | 000000xx |
| | 0F2H | 保留 | (禁止使用) | (禁止使用) |
| | 0F3H | 存储器数据存取寄存器 | WW | 连接到存储器 |
| 两线 串 行 接 口 | 0F4H | 数据存取寄存器 | WW | xxH |
| | 0F5H | 控制和状态寄存器 | WRRRRRW | 00000000 |
| | 0F6H | 地址设定寄存器 | WW | 00H |
| | 0F7H | 设备地址和命令寄存器 | WW | 00H |
| 控制 寄 存 器 和 杂 项 | 0F8H | 芯片控制寄存器 | RRRRRWWW | rrrrr00x |
| | 0FAH | 读写速度控制寄存器 | WRRWRWWW | 0rr0r111 |
| | 0FCH | 硬件循环计数寄存器 | RR | xxH |
| | 其它地址 | 保留 | (禁止使用) | (禁止使用) |

6.4. 寄存器的位说明

| 寄存器名称 | 位址 | 属性 | 位的使用说明 | 位值=0 | 位值=1 |
|--|-----|------------|--|-------|--------|
| 芯片控制寄存器 (配置空间 40H) (I/O 空间偏移 0F8H) | 位 0 | W | 设定 A15 引脚的输出值 | 低电平 | 高电平 |
| | 位 1 | W | 设定 SYS_EX 引脚的输出值 | 低电平 | 高电平 |
| | 位 2 | W | 设定 INTA 中断激活状态 | 无中断 | 有中断 |
| 芯片状态寄存器 (配置空间 42H) | 位 0 | S | PCI 设备标识的当前选择 | 外部 ID | 默认 ID |
| | 位 2 | S | 本地硬件定址的功能 | 禁用 | 启用 |
| | 位 4 | S | 内置 Boot-ROM 的功能 | 禁用 | 启用 |
| | 位 6 | S | SYS_EX 输出信号线 | 禁用 | 启用 |
| 两线串口控制和状态寄存 器 (I/O 空间偏移 0F5H) | 位 7 | S | 中断功能的使能状态 | 禁用 | 启用 |
| | 位 0 | W | 两线串行接口的操作状态 | 完成 | 操作 |
| 读写速度控制寄存器 (I/O 空间偏移 0FAH) | 位 7 | W | 选择 SCL 输出信号线 | A15 | SYS_EX |
| | 位 2 | W | 位 2~位 0 设定读写脉冲宽度，步距为 30nS 位 4 为 0，则 000~111 对应 30nS~240nS 位 4 为 1，则 000~111 对应 0nS~210nS | | |
| | 位 1 | W | | | |
| | 位 0 | W | | | |
| | 位 4 | W | 数据和地址输出建立时间 | 15nS | 45nS |
| 位 7 | W | 存储器空间预取控制位 | 禁止 | 使能 | |

7、功能说明

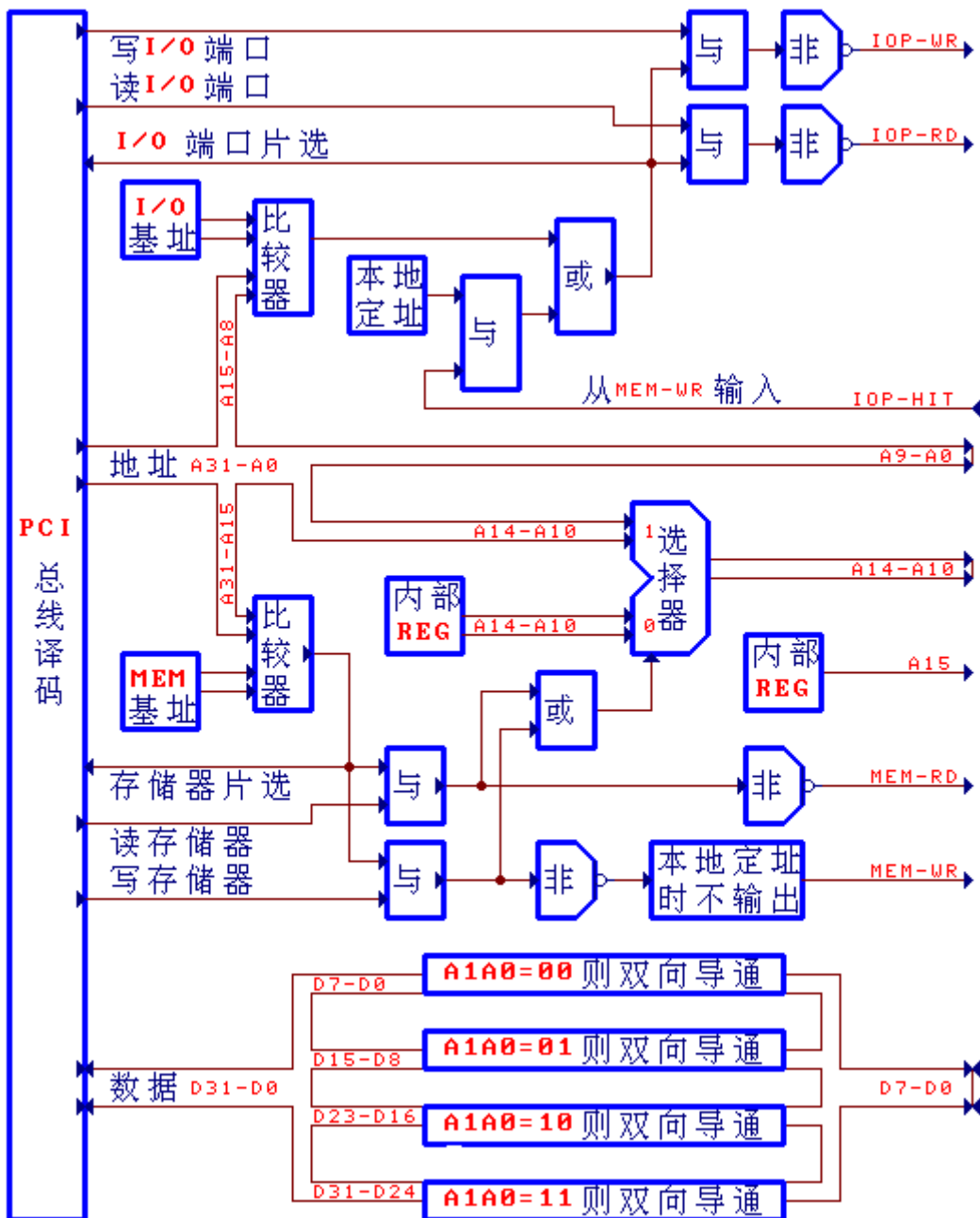
7.1. 空间映射

PC 机中包括三种空间：存储器空间、I/O 空间、配置空间。存储器空间主要包括内存、显存、扩展 ROM、设备缓冲区等，一般用于存放大量数据和进行数据块交换。I/O 空间主要包括设备的控制寄存器和状态寄存器，一般用于控制和查询设备的工作状态以及少量数据的交换。配置空间主要用于向系统提供设备自身的基本信息，并接受系统对设备全局状态的控制和查询。

为了避免地址冲突，PCI 总线要求各个设备所占用的地址能够重定位。重定位是由设备的配置空间的基址寄存器实现的，通常情况下，各个设备的基址寄存器总是被 BIOS 或者操作系统分配为不同的基址，从而将各个设备分别映射到不同的地址范围。在需要时，应用程序也可以自行修改基址。

CH365 的存储器空间占用 32K 字节，偏移地址是 0000H~7FFFH，可以全部提供给外部设备使用，实际地址是存储器基址加上偏移地址。CH365 的 I/O 空间占用 256 字节，去掉 CH365 自用寄存器，还可以提供 240 字节给外部设备使用，偏移地址是 00H~EFH，实际地址是 I/O 基址加上偏移地址。

7.2. 内部结构和信号线



上图是 CH365 内部的主要结构。CH365 对 PCI 总线的各种信号进行译码后，产生内部数据总线 D31~D0、内部地址总线 A31~A0、读 I/O 端口信号、写 I/O 端口信号、读存储器信号、写存储器信号等。图中已经标明了各个信号的传输方向。

结构图右侧的信号是指 CH365 提供给本地端的各个外部引脚。地址线 A15~A0 用于提供相对于基址的偏移地址，数据总线 D7~D0 在读操作时用于输入数据，在写操作时用于输出数据。IOP_RD 用于提供 I/O 读选通脉冲信号，IOP_WR 用于提供 I/O 写选通脉冲信号，MEM_RD 用于提供存储器读选通脉冲信号，MEM_WR 用于提供存储器写选通脉冲信号，上述引脚的读写选通脉冲信号都是低电平有效。CH365 提供的地址线、数据总线、读写选通信号线类似于 ISA 总线的信号线，所以非常适合将 ISA 板卡升级到 PCI 总线上。并且从图中可以看出，CH365 提供的读写选通信号已经在芯片内部被片选控制，CH365 输出的读写选通信号只是在其基址映射范围内有效，所以外部设备不再需要片选译码。

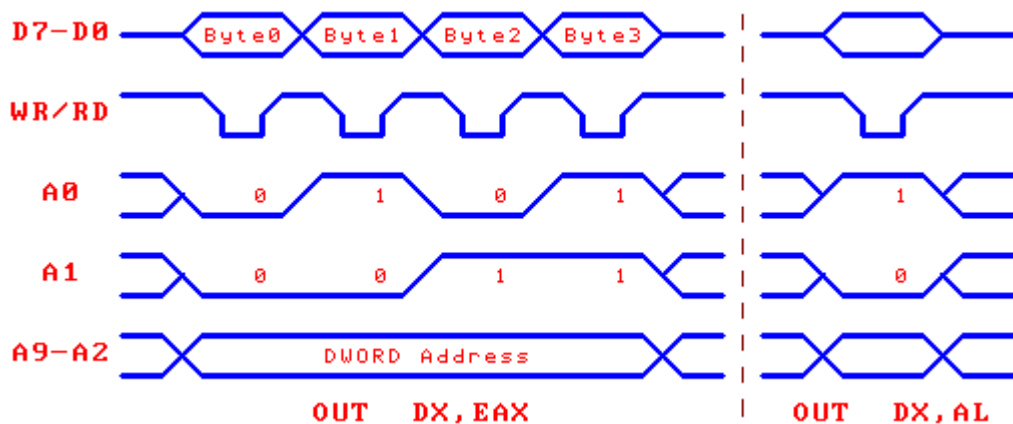
在 I/O 读写操作期间，CH365 的 A7~A0 输出 I/O 端口的偏移地址，提供给外部设备的有效偏移地址范围是 00H~EFH，外部设备可以进一步对 A7~A0 进行译码产生二级片选信号。在 I/O 读写操作期间，CH365 的 A15~A10 保持不变，但可以由内部寄存器事先设定为高电平或者低电平，A9~A8 输出 PCI 总线的地址，如果使用本地硬件定址的功能，则应该对 CH365 的 A9~A0 进行地址译码，通过 IOP_HIT 引脚向 CH365 请求本地定址，以实现与 ISA 总线相兼容的 000H~3FFH 地址范围内的 I/O 端口定址。

在存储器读写操作期间，CH365 的 A14~A0 输出存储器的偏移地址，提供给外部设备的有效偏移地址范围是 0000H~7FFFH。在存储器读写操作期间，CH365 的 A15 保持不变，但可以由内部寄存器事先设定为高电平或者低电平，用于存储器地址线扩展或者页面选择。由于扩展 ROM 属于存储器的一种，所以其操作方式以及操作时序与存储器相同。

7.3. 数据宽度

CH365 支持 PC 机程序以单字节、双字节（字）、四字节（双字）为单位对 I/O 端口或者存储器进行读写。在多字节连续读写操作期间，CH365 每读写完一个字节数据后，就会自动将偏移地址加 1，以指向下一字节的偏移地址。由于 PC 机只要执行一条读写四字节数据的指令，CH365 就会自动将其分解为 4 个 8 位数据的连续读写操作，所以在 PC 机看来，CH365 能够提供 8 位、16 位和 32 位的数据宽度，而且实际情况也是 32 位数据宽度时的工作效率更高，总体数据交换速度更快。

下图左边是汇编指令“OUT DX, EAX”的波形，其中 DX 是 CH365 的 I/O 基址范围内的任意双字节地址，例如 DX=IoBaseAddr+4；右边是“OUT DX, AL”的波形，其中 DX 是 CH365 的 I/O 基址范围内的任意地址，例如 DX=IoBaseAddr+1，对应的 C 语言指令是“outportb(IoBaseAddr+1, Value)”。I/O 读操作和存储器的读写操作与 I/O 写操作的情况相类似。当 PC 机程序执行 32 位数据宽度的指令时，由于 CH365 自动产生连续的 4 次 8 位数据操作，所以数据传输效率比 PC 机程序执行 8 位数据宽度的指令时更高。



7.4. 示例说明

基于 CH365 设计一块类似于打印口的 PCI 板卡。设计约定板卡的 I/O 偏移地址 00H 是数据端口，偏移地址 01H 是状态端口，偏移地址 02H 是控制端口。当插入 PC 机后，该板卡可能被分配一个 I/O

基址 9500H, 则数据端口的实际 I/O 地址是 9500H, 状态端口的 I/O 地址是 9501H, 控制端口的 I/O 地址是 9502H。区分各个端口是对 CH365 的 A7~A0 进行地址译码实现的, 如果不需要其它端口, 也可以只对 A1~A0 进行简化译码。

如果将两块完全相同的上述板卡插入 PC 机, 则第二块板卡也会被系统自动分配一个 I/O 基址, 但一定不会与第一块板卡的 I/O 基址相同。如果第二块板卡的 I/O 基址是 C700H, 则第二块板卡的控制端口的实际 I/O 地址是 C702H, 从而使得两块完全相同的 PCI 板卡分别具有不同的 I/O 端口地址, 避免了 I/O 地址冲突。

板卡设计者和相关的应用程序事先知道各个端口的偏移地址, 但是无法事先知道板卡的 I/O 基址, 所以应用程序在对 PCI 板卡进行 I/O 操作前, 需要通过板卡的配置空间的 I/O 基址寄存器了解当前板卡的 I/O 基址, 再由 I/O 基址加上各个端口的偏移地址计算出各个端口的实际 I/O 地址, 最后根据实际 I/O 地址对各个端口进行 I/O 操作。

存储器方面与 I/O 端口类似, 以 CH365 连接一个 32KB 容量的双口 SRAM 进行高速数据交换为例。如果 CH365 的存储器基址被分配为 E3050000H, 则计算机程序读写物理地址范围 E3050000H~E3057FFFH 就是读写该双口 SRAM。注意, 实际的 PC 机程序通常使用转换后的虚拟地址而不是物理地址; 另外, 如果要在 DOS 下对存储器进行读写, 通常要将存储器基址设置在 DOS 可寻址的 1MB 以下, 例如 000D0000H 或者 000D8000H。

以下是相应的读写过程示例。

- ① 向控制端口写出数据 5AH, 对应 C 语言程序 “outportb(0x9502, 0x5A)”。执行后 CH365 的地址线 A7~A0 输出控制端口的偏移地址 02H(地址 9502 被分解为基址 9500H 和偏移地址 02H, CH365 只输出偏移地址, 不输出基址), CH365 的数据线 D7~D0 输出 5AH, 同时 IOP_WR 输出一个低电平脉冲, 脉冲宽度由 CH365 的读写速度控制寄存器事先设定, 默认是 240nS。
- ② 从数据端口和状态端口读入数据, 对应 C 语言程序 “inport(0x9500)”, 返回结果的低字节是从数据端口读取的数据, 高字节是从状态端口读取的数据。执行后 CH365 的地址线 A7~A0 首先输出数据端口的偏移地址 00H, 同时 IOP_RD 输出第一个低电平脉冲, 外部设备应该将数据输出到数据总线 D7~D0 上; 然后 CH365 的地址线 A7~A0 输出状态端口的偏移地址 01H, 同时 IOP_RD 输出第二个低电平脉冲, 外部设备应该将状态输出到数据总线 D7~D0 上。
- ③ 存储器读写与 I/O 读写类似, 但有两点区别: 第一是 CH365 的地址线 A14~A0 输出 15 位偏移地址, 而 I/O 只有 A7~A0 输出 8 位偏移地址; 第二是用 MEM_RD 引脚输出读控制信号代替 IOP_RD 引脚输出读控制信号, 用 MEM_WR 引脚输出写控制信号代替 IOP_WR 引脚输出写控制信号, 从而能够让外部设备区分出是存储器读写操作, 而不是 I/O 端口的读写操作。

8、参数

8.1. 绝对最大值 (临界或者超过绝对最大值将可能导致芯片工作不正常甚至损坏)

| 名称 | 参数说明 | 最小值 | 最大值 | 单位 |
|-----|------------------------|------|---------|----|
| TA | 工作时的环境温度 | -20 | 70 | °C |
| TS | 储存时的环境温度 | -55 | 125 | °C |
| VCC | 电源电压 (VCC 接电源, GND 接地) | -0.5 | 6.5 | V |
| VIO | 输入或者输出引脚上的电压 | -0.5 | VCC+0.5 | V |

8.2. 电气参数 (测试条件: TA=25°C, VCC=5V, 不包括连接 PCI 总线的引脚)

| 名称 | 参数说明 | 最小值 | 典型值 | 最大值 | 单位 |
|-----|--------------------|------|-----|---------|----|
| VCC | 电源电压 (请参考下面的注意事项) | 3.3 | 5 | 5.5 | V |
| ICC | 工作时的电源电流 | 10 | 20 | 50 | mA |
| VIL | 低电平输入电压 | -0.5 | | 0.8 | V |
| VIH | 高电平输入电压 | 2.0 | | VCC+0.5 | V |
| VOL | 低电平输出电压 (4mA 吸入电流) | | | 0.5 | V |

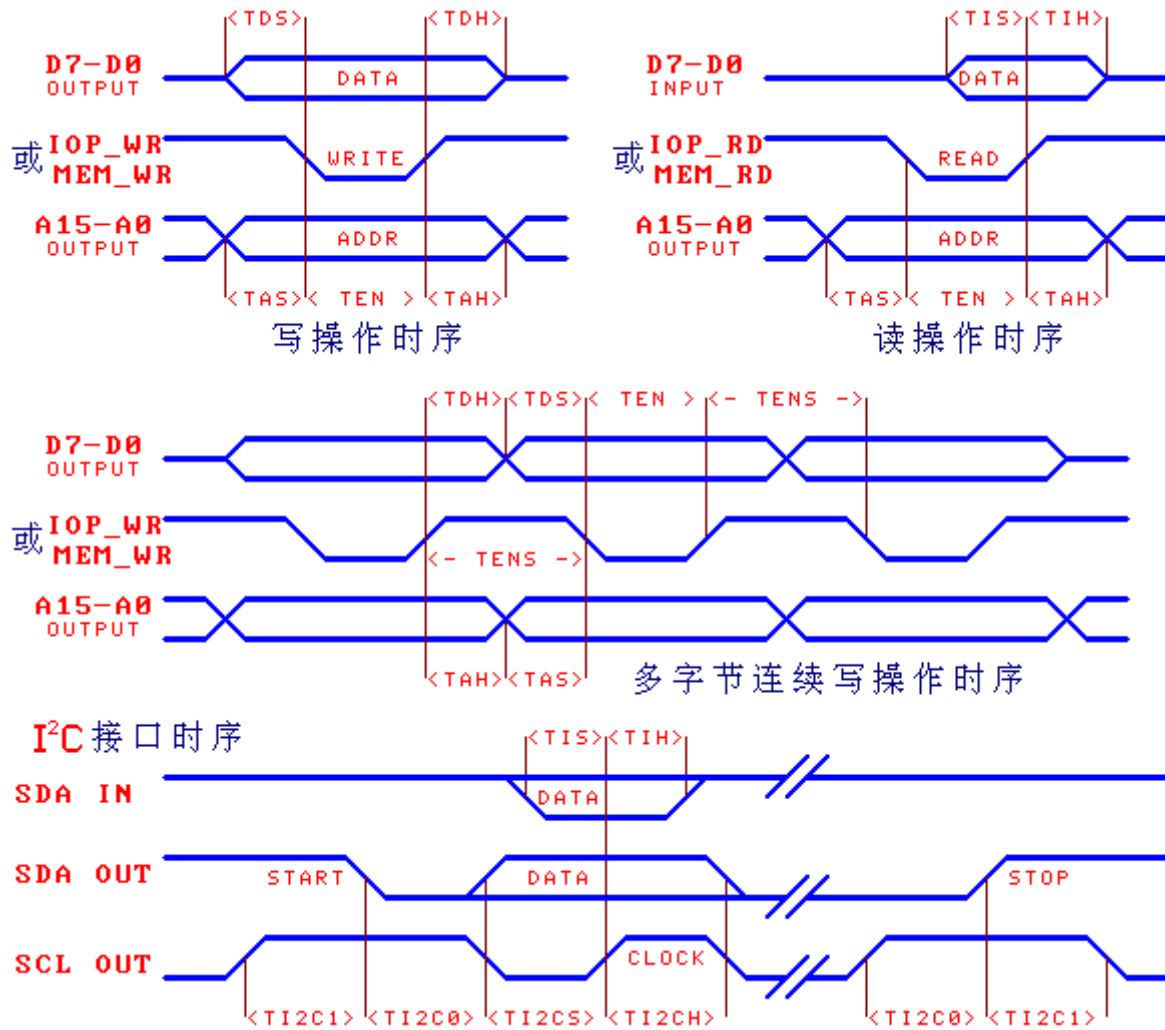
| | | | | | |
|-----|-------------------|-----|----|-----|----|
| VOH | 高电平输出电压（4mA 输出电流） | 4.5 | | | V |
| IIN | 无上拉的输入端的输入电流 | | | 10 | uA |
| IUP | 带上拉的输入端的输入电流 | | 50 | | uA |
| RUP | 上拉电阻的阻值（非线性等同值） | 40 | 60 | 100 | KΩ |

注意事项：CH365 的输入承受电压实际是电源电压加上 0.5V，例如，当 CH365 工作于 3.3V 电源电压时，外部设备提供给 CH365 的输入电压不得超过 3.8V。当 CH365 的电源电压低于 4V 时，PCI 总线的主频不得超过 33MHz，也就是说 PCI 总线不得超频工作。

8.3. 时序参数（测试条件：TA=25℃，VCC=5V，FCLK=33.3MHz，参考附图）

| 名称 | 参数说明 | 最小值 | 典型值 | 最大值 | 单位 |
|-------|---|------------------|----------------|------------|-----|
| FCLK | CLK 输入频率（PCI 总线的主频） | 0 | 33.3 | 40 | MHz |
| TEN | IOP_RD、IOP_WR、MEM_RD、MEM_WR 读选通或写选通的低电平脉冲宽度 | 30 | 可选择 30~240 | 240 默认值 | nS |
| TENS | IOP_RD、IOP_WR、MEM_RD、MEM_WR 多字节连续选通的高电平间隔宽度 | 30 默认值 | 可选择 30 或 60 | 60 | nS |
| TAS | 地址 A15~A0 输出建立时间 | 12 | 可选择 15 或 45 | | nS |
| TAH | 地址 A15~A0 输出保持时间 | 12 | 15 | | nS |
| TDS | 数据 D7~D0 输出建立时间 | 12 | 可选择 15 或 45 | | nS |
| TDH | 数据 D7~D0 输出保持时间 | 12 | 15 | | nS |
| TIS | 数据 D7~D0 输入建立时间 | 15 | | | nS |
| TIH | 数据 D7~D0 输入保持时间 | 0 | | | nS |
| TINT | INT_REQ 中断请求的脉冲宽度 | 80 | 100 | | nS |
| FSCL | SCL 输出频率（两线接口的主频） | FCLK / 128 = 260 | | | KHz |
| TI2C0 | 开始和终止操作时 SDA 低电平时间 | | 3.84 | | uS |
| TI2C1 | 开始和终止操作时 SDA 高电平时间 | | 3.84 | | uS |
| TI2CS | SDA 数据输出建立时间 | | 1.92 | | uS |
| TI2CH | SDA 数据输出保持时间 | | 1.92 | | uS |

- 8.3.1. 读写选通脉冲的宽度 TEN 由读写速度控制寄存器的位 2~位 0 设定，可以以 PCI 总线时钟周期 30nS 为步距在 30nS~240nS 之间进行选择，误差 10%。
- 8.3.2. 在多字节连续读写操作中，读写选通脉冲的间隔 TENS 由读写速度控制寄存器的位 4 设定，设定为 0 则间隔 30nS，设定为 1 则间隔 60nS，误差 10%。在单字节读写操作中，TENS 的最小值是 150nS，一般在 400nS 以上。如果接双口 RAM 或者其它读写速度慢于 25nS 的外围电路，建议设定间隔为 60nS。
- 8.3.3. 地址输出建立时间 TAS 和数据输出建立时间 TDS 是指地址和数据输出有效相对于读写选通脉冲下降沿的提前时间，TAS 和 TDS 由读写速度控制寄存器的位 4 设定，设定为 0 时为 15nS，设定为 1 时为 45nS，误差 10%。
- 8.3.4. 对于读操作，CH365 在读选通脉冲的上升沿从 8 位数据总线 D7~D0 采样数据，所以外部设备应该在读选通脉冲的上升沿之前将有效数据送到数据总线上。数据输入建立时间 TIS 是指要求外部设备送到数据总线 D7~D0 上的数据有效相对于读选通脉冲上升沿的提前时间。
- 8.3.5. 中断请求的脉冲宽度 TINT 是指确保 CH365 接受到中断请求的低电平脉冲的宽度，如果 INT_REQ 处于低电平的时间太短，则 CH365 可能接受不到中断请求。
- 8.3.6. 两线串行接口时序：SDA 信号线是输入输出双向端口，CH365 在 SCL 的下降沿输出数据到 SDA，在 SCL 的上升沿采样 SDA 上的输入数据；在两线串行接口空闲的情况下，SCL 保持低电平，SDA 三态输出禁止，但被芯片内部的上拉电阻保持在高电平。

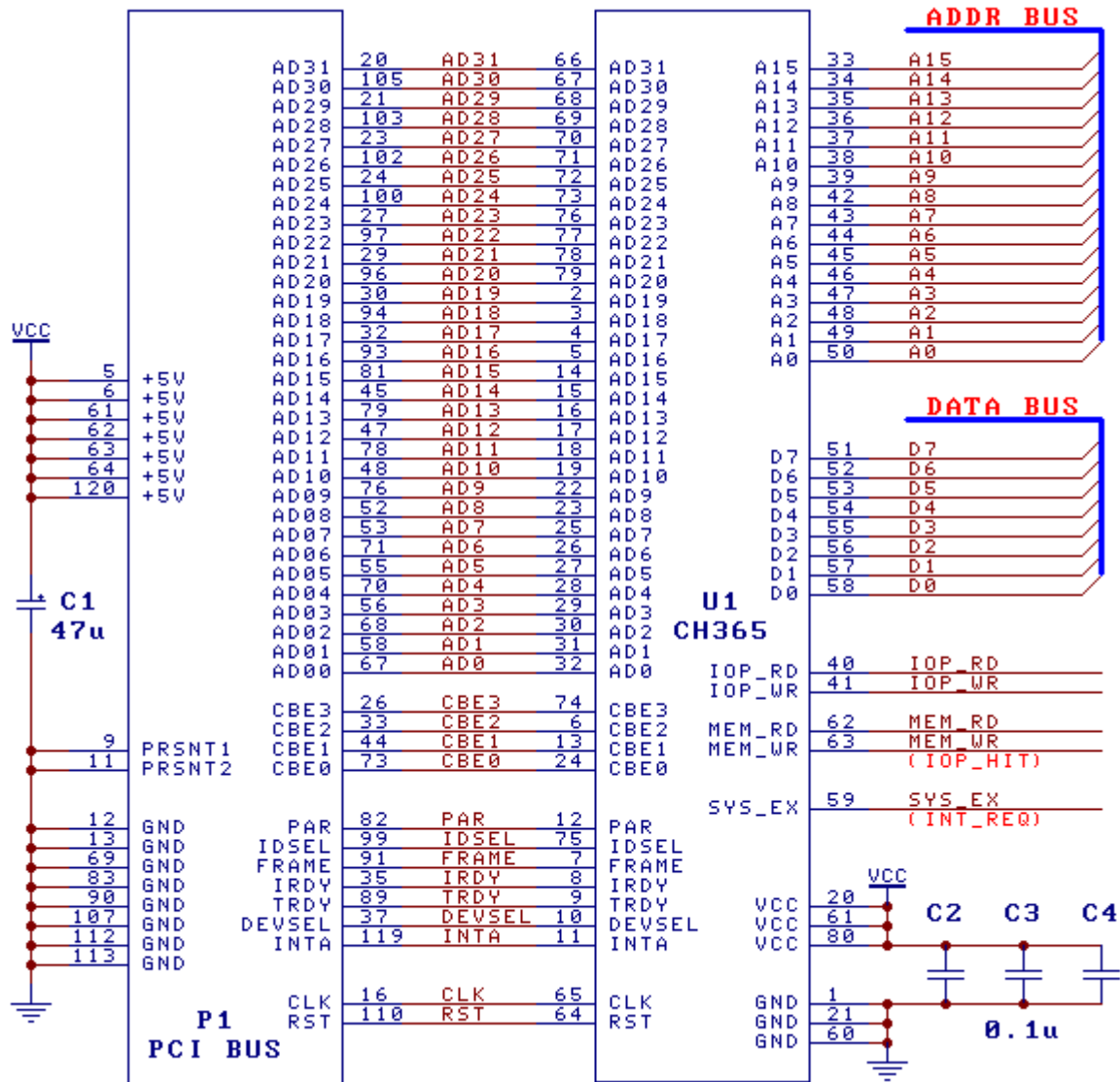


9、应用

9.1. 连接 PCI 总线 (下图)

这是 CH365 与 PCI 总线连接的标准电路，电容 C1~C4 用于电源退耦，C2~C4 是容量为 0.1 μ F 的独石或者高频瓷片电容，数量不应少于 3 个，分别就近并联在 CH365 的三对电源引脚上。连接 PCI 总线的电源线引脚可以自由选择，但数量不宜少于 4 对。

CH365 属于高频数字电路，应该考虑信号阻抗匹配，在设计 PCB 板时需要参考 PCI 总线规范。建议 CH365 的 PCI 信号线的长度都小于 35mm，尽量走弧线或者 45 度线，避免直角或者锐角走线，并且尽量将信号走线布在元件面，而在 PCB 背面保留大面积的接地覆铜。CH365 的 PCI 时钟线 CLK 的长度尽量保持在 50mm~65mm 之间，并且不宜靠近其它信号线，建议在 CLK 两侧及 PCB 背面布置接地线或者覆铜，以减少周边信号线的干扰。



9.2. 连接存储器 (下图左边的电路)

CH365 通过 MEM_RD 和 MEM_WR 与存储器 U2 (型号是 SRAM62256) 连接。在读写存储器 U2 之前, 需要先读取 CH365 配置空间的存储器基址寄存器, 然后由基址加上偏移地址获得 U2 在计算机存储器空间的实际物理地址, 最后通过实际物理地址或者由物理地址转换后的虚拟地址读写存储器 U2 中的内容。

基于 CH365 的通用驱动程序 WDM 和动态链接库 DLL, C 语言程序示例:

```

UCHAR  mByte; // 数据单元, 用于保存从存储器中读出的数据或者准备写入存储器的数据
mPCH365_MEM_REG  mMemBase; // 存储器基址, 实际数据单元的地址等于基址加上偏移地址
CH365OpenDevice( TRUE, FALSE ); // 打开 CH365 设备, 与文件操作一样, 使用前要先打开
CH365GetMemBaseAddr( &mMemBase ); // 获取 CH365 的存储器基址
CH365ReadMemByte( &mMemBase -> mCh365MemPort[0x1234], &mByte );
// 上述操作从存储器的 1234H 地址读取一个字节的数
CH365WriteMemByte( &mMemBase -> mCh365MemPort[0x2E0C], mByte + 0x76 );
// 上述操作将此前读取的数据加上 76H 后写到存储器的 2E0CH 地址
CH365CloseDevice(); // 所有操作结束后, 在应用程序退出前, 应该关闭 CH365 设备
在 DOS 下或者没有操作系统的 PC 机中, 上述操作的汇编程序是:
MOV  AX, 0B10DH ;以双字为单位写 PCI 配置空间
MOV  BX, CH365_PCI_BUS_DEV_ADDR ;CH365 板卡的 PCI 地址, 即总线/设备/功能号

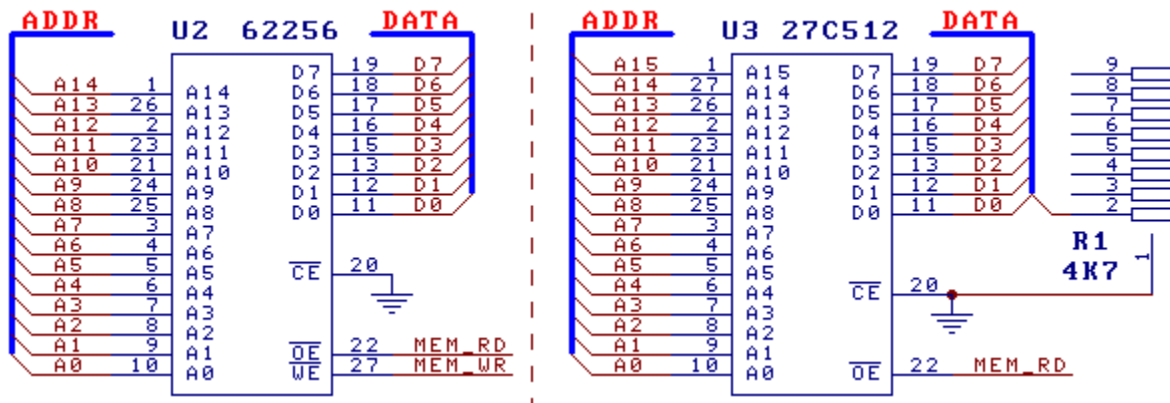
```

```

MOV  ECX, 000D0001H      ;将存储器设定到 D000H 段, 该段一般在 PC 机中空闲
MOV  DI, 0014H           ;存储器基址寄存器的偏移地址 PC_BASE_ADDR1
INT  1AH                 ;设置存储器的基址, 以启用存储器空间
;自动分配的存储器基址一般在 1MB 以上, 由于 DOS 不能寻址, 所以需要修改存储器基址
MOV  AX, 0D000H
MOV  ES, AX              ;将段地址置为 D000H
MOV  AL, ES:[1234H]      ;从存储器的 1234H 地址读取一个字节的数据
ADD  AL, 76H
MOV  ES:[2E0CH], AL      ;将读取的数据加上 76H 后写到存储器的 2E0CH 地址

```

如果将普通的 SRAM 换成双端口 SRAM, 则 CH365 可以通过双端口存储器与外部的单片机或者 DSP 交换数据。如果将 CH365 的读写选通脉冲的宽度设定为 30nS, 并且使用双字为单位进行数据交换, 则数据传输的实测速度可以达到每秒 7M 字节。



9.3. 连接扩展 ROM (上图右边的电路)

CH365 通过 MEM_RD 与 ROM 芯片 U3(型号是 27C512)连接。CH365 支持 EPROM 和闪存 Flash-Memory, 容量为 32KB 或者 64KB, 如果将 SYS_EX 用于 A16 地址线, 则最大容量可达 128KB。一般情况下 CH365 直接支持 32KB 容量的扩展 ROM (即 27C256 芯片的容量), 但可以在扩展 ROM 的程序中通过控制 A15 地址线支持 64KB 容量的 ROM 芯片。可选的下拉电阻 R1 用于 CH365 的工作模式设定, 图中的数据线 D0 被连接了下拉电阻, 所以系统复位后, 地址线 A15 为低电平, 从而选择 U3 的低 32KB (偏移地址为 0000H~7FFFH), 而在需要读取 U3 的高 32KB 时 (偏移地址为 8000H~0FFFFH), 可以通过写芯片控制寄存器的位 0 重新设定 A15 地址线, SYS_EX 用于 A16 地址线时与 A15 的用法类似。另外, PCI 扩展 ROM 的内容通常被 BIOS 复制到 RAM 内存中, 所以需要设置 CH365 的扩展 ROM 基址以重新映射 U3 到存储器空间。CH365 的 A15 引脚不仅能用作地址线, 还可以自由控制, 例如在需要同时连接 SRAM 和 ROM 时, 通过 A15 切换两者的片选。

PC 机中的扩展 ROM 相当于一个电子盘, 如果在其中写入引导程序和应用程序, 即使计算机没有硬盘和操作系统, 扩展 ROM 中的引导程序和应用程序也能够控制计算机实现某些特定的功能。例如, 无硬盘 PC 机用于工业控制, 控制外部设备及作业流程等。

9.4. 连接两线串口的 EEPROM (下图左边的电路)

CH365 的 SCL 信号线可以选用 SYS_EX 或者 A15, 系统复位后的默认选择是 A15, 通常在 A15 用作地址线时选用 SYS_EX, 否则选用 A15。因为 SDA 信号线同时是数据线 D7, 为了防止在 SCL 高电平时由于 SDA 变化而产生不必要的误操作, 通常借助于工作模式设定使 SCL 信号线 (SYS_EX 或 A15) 在系统复位后保持低电平。

CH365 的两线串行接口使用 7 位设备地址, 可以同时连接多个设备。设备地址和命令寄存器的位 7 至位 1 为 7 位设备地址位, 用以选择从设备; 位 0 为命令位, 置 0 代表写操作, 置 1 代表读操作。例如 CH365 连接了两个 24C02 芯片, 下表是从一个芯片读和向另一个芯片写的操作步骤。

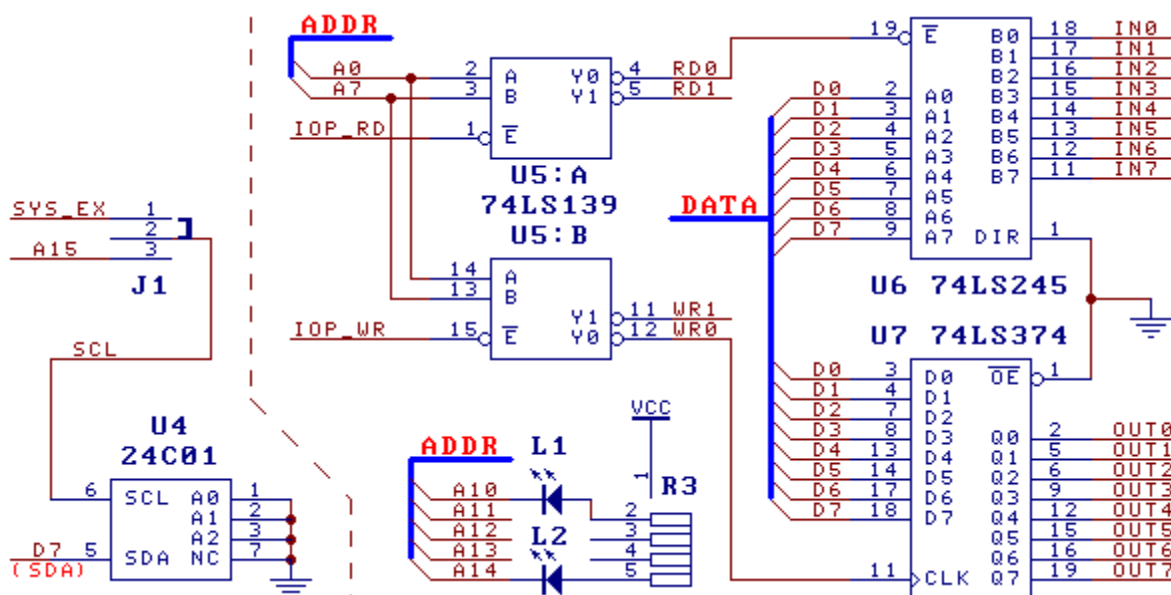
| 操作的步骤 | 从 12H 地址读 | 向 34H 地址写 | 说明 |
|------------------------|----------------------|--------------|----------|
| 查看 24C02 的 A2-A1-A0 引脚 | A2-A1-A0=000 | A2-A1-A0=010 | 实际的设备地址 |
| 设备地址和命令寄存器 | 写入 10100001 | 写入 10100100 | 设备地址和命令 |
| 地址设定寄存器 | 写入 12H | 写入 34H | 设定操作地址 |
| 数据存取寄存器 | 无操作 | 写入 56H | 写入数据 56H |
| 控制和状态寄存器 | 将位 0 置为 1，其它位不变 | | 开始接口操作 |
| 控制和状态寄存器 | 等待位 0 变为 0 或者等待 10mS | | 等待操作完成 |
| 数据存取寄存器 | 读出 78H | 无操作 | 读出数据 78H |

基于 CH365 的通用驱动程序 WDM 和动态链接库 DLL，上述操作的 C 语言程序是：

```

UCHAR  mByte; // 数据单元，用于保存从 24C02 中读出的数据或者准备写入的数据
CH365ReadI2C( 0x50, 0x12, &mByte ); // 从 A2=0=000 的 24C02 的 12H 地址中读取一字节
CH365WriteI2C( 0x52, 0x34, 0x56 ); // 向 A2=0=010 的 24C02 的 34H 地址写入 56H 数据

```



9.5. I/O 端口应用（上图右边的电路）

读选通/使能 IOP_RD 和写选通/使能 IOP_WR 控制 74LS139 的译码使能，74LS139 将地址译码输出 2 路读控制和 2 路写控制，经过 74LS245 输入缓冲和 74LS374 锁存输出，获得 2 组每组 8 位缓冲输入和 2 组每组 8 位锁存输出。例如，CH365 的 I/O 基址被设定为 5A00H，则读取 5A00H 端口就是读取第一组缓冲输入，写入 5A01H 端口就是写入第二组锁存输出。如果 CH365 没有连接扩展 ROM 或者存储器，则空闲的地址线 A14~A10 以及 A15 可以直接作为输出控制线使用。A15~A8 地址设定寄存器在系统复位后 A14~A10 为低电平，A15 在复位期间为低电平，复位后则由工作模式设定，默认为高电平，除非系统复位时数据线 D0 被下拉，由工作模式设定为低电平。

CH365 的输入和输出信号与 TTL 电平和 CMOS 电平兼容，可以连接 ADC/DAC/MCU 等芯片，输出引脚的驱动电流大于 5mA，可以在串接限流电阻后直接驱动 LED 显示。CH365 提供了 8 根地址线 A7~A0 用于 I/O 地址译码，有效偏移地址范围是 0EFH~00H，长度不超过 240 字节，一般情况下，外部电路可以无需片选线或者直接强制片选。

基于 CH365 的通用驱动程序 WDM 和动态链接库 DLL，上述操作的 C 语言程序是：

```

UCHAR  mByte; // 数据单元，用于保存从 I/O 端口中读出的数据或者准备写入 I/O 的数据
mPCH365_I0_REG  mIoBase; // I/O 端口基址，实际数据单元的地址等于基址加上偏移地址
CH365GetIoBaseAddr( &mIoBase ); // 获取 I/O 端口的基址，这是可选操作，不必执行，
// 如果不获取 I/O 基址则可以在 I/O 操作中只指定偏移地址，相当于 I/O 基址为 0，
// 在调用 CH365 的 DLL 后，DLL 会自动将偏移地址加上基址再进行 I/O 操作，
// 存储器与此类似，如果存储器操作中只指定偏移地址，则 DLL 会自动加上存储器基址

```

```

CH365ReadIoByte( & mIoBase -> mCh365IoPort[0x00], &mByte );
// 上述操作从 I/O 端口的 00H 偏移地址读取一个字节的的数据，即读入第一组缓冲输入
CH365WriteIoByte( & mIoBase -> mCh365IoPort[0x01], 0x47 );
// 上述操作将数据 47H 写到 I/O 端口的 01H 偏移地址，即作为第二组锁存输出
CH365SetA15_A8( 0x24 ); // 设置 A13 为高电平，A10 为高电平，其余引脚为低电平
在 DOS 下或者没有操作系统的 PC 机中，上述操作的汇编程序是：
MOV    AX, 0B109H                ;以字为单位读取 PCI 配置空间
MOV    BX, CH365_PCI_BUS_DEV_ADDR ;CH365 板卡的 PCI 地址，即总线/设备/功能号
MOV    DI, 0010H                 ;I/O 端口基址寄存器的偏移地址 PC_BASE_ADDR0
INT    1AH                      ;读取 I/O 端口的基址，由计算机初始化时自动设置
AND    CX, 0FFFEH               ;获得 I/O 端口的基址，最低位为指示位，要屏蔽掉
MOV    BX, CX                   ;该值为 I/O 端口的基址
LEA    DX, [BX].CH365_IO_PORT[0] ;第一组缓冲输入的 I/O 端口地址，I/O 基址加 0
IN     AL, DX                   ;读取 74LS245 缓冲输入的数据
LEA    DX, [BX].CH365_IO_PORT[1] ;第二组锁存输出的 I/O 端口地址，I/O 基址加 1
MOV    AL, 47H
OUT    DX, AL                   ;将数据 47H 写入 74LS374 锁存输出寄存器
LEA    DX, [BX].CH365_MEM_ADDR_H ;A15-A8 地址设定寄存器的 I/O 端口地址，在 CH365 内部
IN     AL, DX                   ;为了保持其余引脚的状态，先读取原 A15-A8 地址
OR     AL, 20H                  ;仅将 A13 置为高电平，而其余引脚保持不变
AND    AL, 0F7H                 ;仅将 A11 置为低电平，而其余引脚保持不变
OUT    DX, AL                   ;将新的 A15-A8 写入地址设定寄存器

```

CH365 的 I/O 端口具有与存储器相同的数据传输速度，只是 PC 机在 I/O 空间的指令少于在存储器空间的指令。一般情况下，外部设备可以将控制和状态等寄存器单元映射到 I/O 空间，而将数据块交换缓冲区映射到存储器空间。

9.6. 连接单片机等

PC 机通过 CH365 与单片机或者 DSP 进行双向数据传输，可以使用四种方式：一是使用双口 SRAM，让 CH365 和单片机能够读写同一块存储器，以大数据块为单位进行双向数据交换；二是使用双向缓冲接口芯片 CH421，分别为 CH365 写单片机和单片机写 CH365 提供 64 字节的缓冲区，以 64 字节的数据块为单位进行双向数据交换；三是使用 8255 提供的异步数据交换，以一个数据字节为单位进行双向数据交换；四是以半字节或者位数据交换为主的低速传输，不需要增加额外的硬件成本，例如使用软件配合的 4 位数据交换接口、CH365 本身提供的两线串行接口、使用软件模拟的 SPI 接口等。

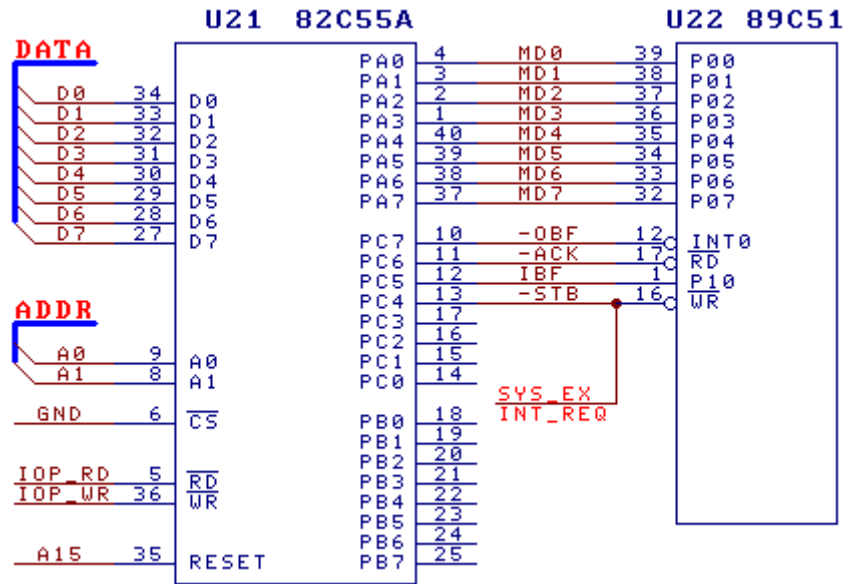
9.6.1. 带握手信号的异步数据交换（下图）

常用的 I/O 扩展芯片 8255，如果 A 口工作于方式 2，则可以在 CH365 和单片机之间提供标准的带握手信号的以字节为单位的异步数据交换。在电路连接上，如果 8255 的 A 口连接 CH365，则由单片机对 8255 进行控制；如果 8255 的 A 口连接单片机，则由 PC 机通过 CH365 对 8255 进行控制。当 8255 的 A 口工作于方式 2 时，可以向 CH365 和单片机提供输入缓冲区空/满，输出缓冲区空/满等握手信号以及中断信号。

图中 U21（82C55A）的 A 口连接单片机 U22（89C51），地址线 A15 用于复位 U21，在计算机应用程序中将 U21 的 A 口设置在方式 2，单片机的中断是低电平触发。

当计算机向单片机写数据后，其写操作使 -OBF 变为低电平，U22 进入中断程序；当单片机在中断程序中读取数据后，其读操作 -ACK 使 -OBF 恢复为高电平；计算机程序查询到 -OBF 的状态，从而可以再次向单片机写数据。

当单片机向计算机写数据后，其写操作 -STB 使 IBF 变为高电平；如果启用了 CH365 的中断功能，则 -STB 向 SYS_EX 输出的低电平脉冲使计算机进入中断服务程序，如果未启用中断功能，则计算机程序使用查询方式检查 IBF 的状态；计算机程序查询到 IBF 的状态并读取单片机的数据，其读操作使 IBF 变为低电平；单片机查询到 IBF 的状态，从而可以再次向计算机写数据。



9.6.2. 使用双向缓冲接口芯片 CH421

CH421 能够提供 CH365 与单片机的快速连接，详细内容请参考 CH421 芯片手册。

9.7. I/O 端口扩展（可以参考上图，将上图中 U22 去掉）

CH365 可以直接连接大多数常用的 I/O 端口扩展电路，图中用 U21 进行 I/O 扩展。由于早期 8255 的读写速度在 300ns 以上，而后期 82C55A 的读写速度可以达到 200ns 以下，所以 U21 应该选用 82C55A 型号。8255 的端口地址被定位于偏移地址 00H~0EFH，实际上只用到 00H~03H 地址，其余都是别名地址。8255 可以将 8 位数据线扩展为 3 个 8 位的数据端口，例如，向偏移地址 03H 写入控制字 10010000B，则将 U21 的 PA 端口设置为 8 位输入，将 PB 端口以及 PC 端口设置为 8 位输出。

基于 CH365 的通用驱动程序 WDM 和动态链接库 DLL，上述操作的 C 语言程序是：

```

UCHAR  mByte; // 数据单元，用于保存从 I/O 端口中读出的数据或者准备写入 I/O 的数据
mPCH365_I0_REG  mIoBase = NULL; // I/O 端口基址，设定为 0 则 DLL 会自动加上基址
CH365ReadIoByte( & mIoBase -> mCh365IoCtrl, &mByte ); // 先读取芯片控制寄存器
CH365WriteIoByte( & mIoBase -> mCh365IoCtrl, mByte | mBitAddr15Out ); // A15=1
CH365WriteIoByte( & mIoBase -> mCh365IoCtrl, mByte & ~ mBitAddr15Out ); // A15=0
// 上述 3 个操作控制 A15 引脚输出高电平脉冲，从而复位 U21 (82C55A)
CH365WriteIoByte( & mIoBase -> mCh365IoPort[0x03], 0x90 ); // 10010000B
// 上述操作设置 8255 的控制字，PA 输入，PB/PC 输出
CH365ReadIoByte( & mIoBase -> mCh365IoPort[0x00], &mByte );
// 上述操作从 8255 的 PA 输入数据
CH365WriteIoByte( & mIoBase -> mCh365IoPort[0x01], 0x8E );
// 上述操作向 8255 的 PB 输出数据 8EH

```

9.8. 其它辅助功能

9.8.1. 硬件循环计数寄存器

CH365 提供了宽度为一个字节的硬件计时单元，以 PCI 总线主频的 128 分频为计时输入，对于 PCI 总线的标准 33.3MHz 主频，硬件循环计数寄存器每隔 3.84μs 增加一个计数，从 00H 计数到 0FFH 再循环到 00H 共需要 983.04μs。通过比较前后两次读取的计数的差值，可以计算出实际延时，用来代替误差较大的计算机软件指令循环。

9.8.2. 8 位总线输入端口

CH365 在 PCI 设备配置空间提供了 8 位总线状态输入端口，读取该端口时 I/O 和存储器的读写选

通控制线都不会输出信号，所以能够读取当前本地数据总线 D7~D0 的静态数值。如果数据总线的 D6 引脚上连接了下拉电阻，则读出的数据的位 6 为 0，否则为 1。一般该功能用于应用程序识别板卡的功能或者扩展 ROM 中的引导程序读取该板卡的外部配置信息，例如，D6 和 D7 下拉代表某种配置方式，仅 D6 下拉又代表另一种配置方式，应用程序或引导程序可以据此作出区别。

9.9. 硬件中断

CH365 支持低电平有效的中断请求，如果数据线 D3 连接下拉电阻，则工作模式设定为启用中断功能，SYS_EX 引脚复用为 INT_REQ，作为本地中断请求输入引脚。

当 INT_REQ 引脚检测到低电平后，CH365 的中断激活状态位（即芯片控制寄存器的位 2）自动设置为 1，CH365 通过 PCI_INTA 引脚向 PCI 总线申请中断。此时即使 INT_REQ 引脚再恢复为高电平，CH365 依然保持在中断激活状态，直到 PC 机的 CH365 中断服务程序将 CH365 的中断激活状态位清除为 0。但是，如果清除中断激活状态位后 INT_REQ 引脚仍然检测到低电平，则 CH365 再次进入中断激活状态。

如果 PC 机程序以软件方式将 CH365 的中断激活状态位设置为 1，也可以使 CH365 进入中断激活状态，从而向 PCI 总线申请中断。这样的软件中断与由外部 INT_REQ 低电平导致的硬件中断具有完全相同的特性，可以用于测试 CH365 的中断功能。

详细说明请参考 CH365 评估板资料中的《CH365 中断功能说明》，标准的中断过程如下：

- ① 外部电路向 INT_REQ 输出低电平有效的中断请求信号，导致 CH365 的中断激活位变为 1。如果外部电路提供的是低电平脉冲，则脉冲宽度必须大于 TINT。
- ② CH365 通过 PCI 总线的 INTA 信号向 PC 机申请中断。
- ③ PC 机进入 CH365 的中断服务程序。
- ④ 如果外部电路提供的中断请求不是脉冲信号，则中断服务程序应该通知外部电路撤消中断请求，也就是将 INT_REQ 恢复为高电平。
- ⑤ 中断服务程序将 CH365 的芯片控制寄存器的中断激活状态位清除为 0，CH365 的 PCI_INTA 引脚恢复为高阻状态，撤消了对 PC 机的中断请求。
- ⑥ 中断服务程序进行必要的中断处理，该步骤也可以在进入中断后先执行。
- ⑦ 中断处理完毕，PC 机退出 CH365 的中断服务程序。

9.10. 本地硬件定址与升级 ISA（适合升级 ISA 总线的 I/O 板卡）

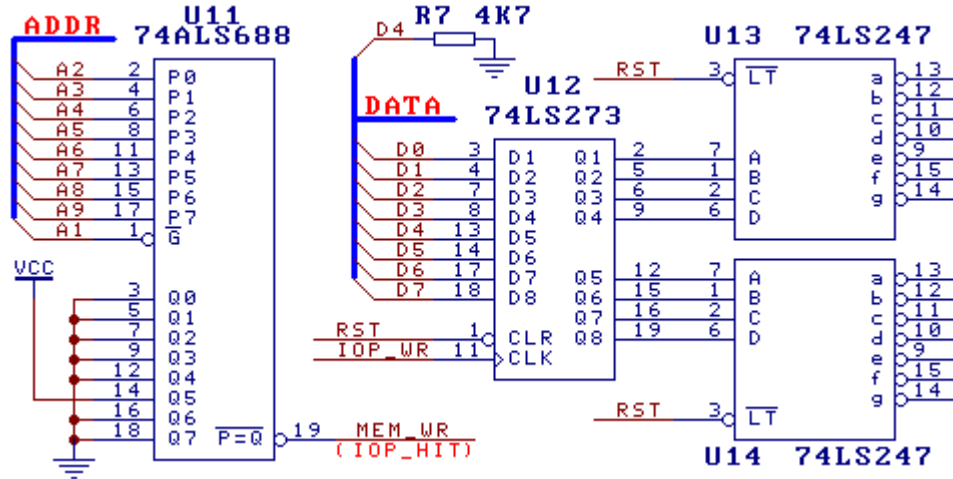
通常情况下，PCI 设备的 I/O 基址是由计算机初始化时自动分配的，一方面保证了一些专用 I/O 地址不会被随意占用，另一方面保证了多个设备的 I/O 地址不会冲突。与 ISA 板卡的 I/O 地址可以灵活使用不同，PCI 总线规范没有提供一种由产品制造商确定板卡 I/O 地址的方法，这样在一些特殊应用中就会有些不方便：第一是在不同类型的计算机上，同一种 PCI 板卡被自动分配的 I/O 地址可能不同；第二是在计算机初始化并且尚未分配 I/O 地址前，板卡无法工作；第三是自动分配的 I/O 地址都在 1000H 地址以上，而不能定位到 3FFH 至 000H 地址范围以及专用的 I/O 地址；第四是将 ISA 板卡直接升级到 PCI 总线时，需要修改应用程序，以便在 I/O 操作前先获取 I/O 端口的基址。

CH365 提供了一种可以由产品制造商选定 PCI 设备 I/O 地址的方法，即本地硬件定址，其原理是将 PCI 设备的部分 I/O 地址译码通过外围的二级译码电路实现。外围的二级译码电路比较简单，与 ISA 总线的 I/O 地址译码类似，CH365 将 PCI 总线 I/O 操作的地址同步提供给外围电路，当外围电路对地址译码匹配后，向 CH365 请求本地硬件定址，再由 CH365 请求 PCI 总线在该地址进行 I/O 操作。

以 PC 机的基于 PCI 总线的故障诊断卡（Debug Card 或 Post Card）为例，PC 机在自检过程中不断地向 0080H 地址的 I/O 端口输出故障代码（Post Code）；但由于 PC 机在完成大部分自检过程后才为 PCI 设备分配 I/O 地址，所以常规的 PCI 板卡无法捕捉到之前的故障代码；即使在分配了 I/O 地址之后，常规的 PCI 板卡也无法定位到专用 I/O 地址 0080H。

下图是使用 CH365 制作的故障诊断卡的部分电路。电阻 R7 用于工作模式设定，数据线 D4 被下拉，设定值是 11101111，CH365 将复用引脚 MEM_WR 设定为本地硬件定址请求输入信号线 IOP_HIT。比较器 U11（可用 74F138 实现），用于对地址 A9~A1 进行匹配比较，当 CH365 提供的地址与预设地址值 0080H~0081H 相同时，U11 输出低电平。CH365 接收到本地硬件定址请求后，输出有效的 IOP_RD 读取数据，或者输出有效的 IOP_WR 写出数据。对于故障诊断卡，只需要接收 I/O 数据，数据触发器

U12 将故障代码锁存，由字符译码电路 U13 和 U14 译码后显示在数码管上。



CH365 内部已经限定本地硬件定址的地址范围是 03FFH 至 0000H，对应于 ISA 板卡的 I/O 地址范围。当 PC 机进行 I/O 操作时，CH365 的 A9~A0 总是同步输出操作地址，外围的二级译码电路只需要对地址 A9~A0 进行匹配比较。CH365 要求外围的二级译码延时不超过 20nS，也就是说，外围电路对地址译码比较后产生本地硬件定址请求的时间不能超过 20nS。上图以等值比较器 74ALS688 说明功能，实际的外围译码电路通常选用可编程器件 16V8 或者译码器 74F138，16V8 建议使用 15nS 速度。

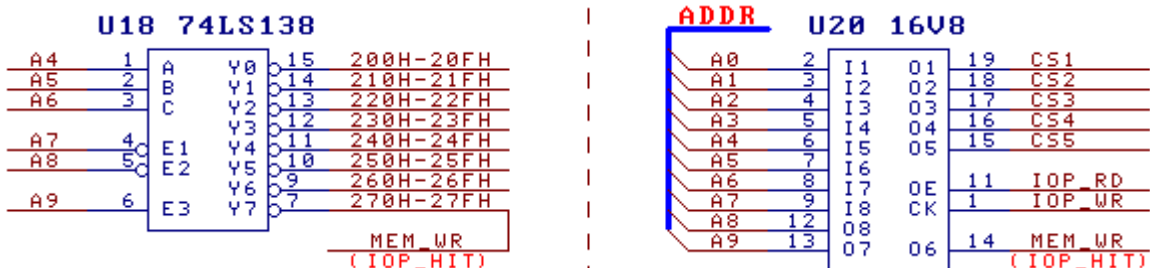
下图左边的电路是用 TTL 逻辑电路实现的二级译码电路，由于 TTL 逻辑电路的速度较低，译码时间可能超过 20nS，所以该图仅用于功能说明和逻辑说明，实际电路建议采用右边的电路，使用简单的可编程器件进行二级译码。

U18 对 CH365 提供的 A9~A4 进行译码，产生 8 个长度为 16 字节的 I/O 地址区域。如果原 ISA 板卡占用 270H~27FH 地址，则将 U18 的 Y7 引脚直接连接到 CH365 的 IOP_HIT 引脚。当 PC 机在 270H~27FH 地址范围进行 I/O 操作时，IOP_HIT 就会被 U18 置为低电平，CH365 在 IOP_RD 和 IOP_WR 引脚输出读写选通信号，外部电路根据读写信号与计算机交换数据，当然外部电路还可以进一步对 A3~A0 进行地址译码。

实际电路建议使用一个可编程器件 U20（型号 GAL16V8）进行二级译码，电路简洁并且使用方便。一般情况下，16V8 在完成二级地址译码后还有剩余的资源（1 脚输入、11 脚输入、15 脚至 19 脚输出），可以用作其它功能，例如，作进一步地址译码，将低位地址与 IOP_RD 和 IOP_WR 组合后进行 I/O 端口选择译码，产生片选信号 CS1~CS5 等。

另外，使用本地硬件定址功能后，CH365 不仅在硬件定址范围响应 I/O 操作，同时也在计算机自动分配的 I/O 地址范围内响应 I/O 操作。例如，硬件定址为 240H~247H，PC 机自动分配的 I/O 地址范围为 C000H~C0FFH，则操作 240H 地址等于操作 C000H 地址，读写 247H 地址等于读写 C007 地址。

基于 CH365 的本地硬件定址功能，产品制造商可以很方便地将原有的 ISA 板卡直接升级到 PCI 总线上，并且仍然可以使用 ISA 板卡的原有 I/O 端口地址，从而不需要对应用程序作任何修改。



9.11. 设定 PCI 设备 ID

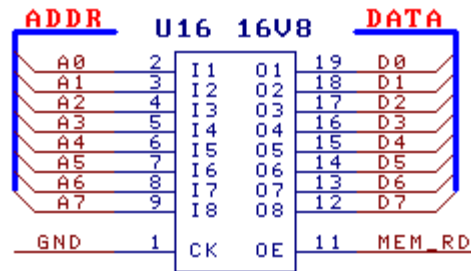
一般情况下，PCI 板卡直接使用 CH365 芯片的默认设备 ID。如果产品制造商需要设定为自己的设备 ID 或者专用的设备 ID，可以在系统复位后将数据线 D1 下拉，对 CH365 进行工作模式设定，使之工作于外部 ID 模式，并由外围电路提供新的设备 ID。

当设定工作模式而使 CH365 使用外部 ID 后，CH365 配置空间中寄存器属性为 S 的所有单元，都

将被直接映射到以 0040H 为起始地址的本地存储器中，也就是说，配置空间 3FH~00H 地址对应本地存储器的 7FH~40H 地址。例如，读取配置空间的 00H 地址就是读取本地存储器的 0040H 地址；读取配置空间的 2CH 地址就是读取本地存储器的 006CH 地址；如果将本地存储器的 0043H~0040H 地址的数据设定为 12345678H，则该 PCI 板卡的厂商标识（Vendor ID）为 5678H，设备标识（Device ID）为 1234H；同理，可以在存储器的 006FH~006CH 地址设定 PCI 板卡的子系统标识（Subsystem ID）。

提供新的设备 ID 的本地存储器包括扩展 ROM 以及其它具有只读属性的存储器。对于支持扩展 ROM 的 PCI 板卡，可以直接将 PCI 设备 ID 设置在 ROM 芯片的 007FH~0040H 地址，而不必再增加任何外围器件。如果 PCI 板卡本身不需要扩展 ROM，但又需要设定 PCI 设备 ID，则有两种方法：一种方法是增加一个额外的 ROM 芯片，仅用于提供新的 PCI 设备 ID，而不作为扩展 ROM；另一种方法是参考下图的电路，使用简单的可编程器件（例如 16V8 等）模拟 ROM 芯片，向 CH365 提供新的 PCI 设备 ID。

下图中 U16 在 MEM_RD 为高电平时，三态输出禁止；在 MEM_RD 低电平有效时，输出与地址 A7~A0 相关的数据。例如，在 MEM_RD 低电平有效时，如果 A7~A0 为 48H 则输出 PCI 板卡的芯片版本号（Revision ID），如果 A7~A0 为 49H 则输出 PCI 板卡的设备主分类代码（Class Code）。



9.12. 通过 I/O 读写存储器

一般情况下，CH365 的本地存储器被映射到计算机的存储器空间，读写本地存储器是在存储器空间进行的，为了支持较大容量的扩展 ROM 或者存储器，以及便于在 DOS 下操作寻址，CH365 还提供了一种将存储器空间转向 I/O 空间的方法，适用于容量较大的扩展 ROM 和数据存取比较有序的存储器。该方法直接支持 64KB 容量的存储器或者扩展 ROM。通过 I/O 读写存储器空间的步骤见下表。

| 寄存器 | 寄存器操作 | 读数据程序 (ASM) | 读数据程序 (C/C++) |
|-----------------------|------------------|--|--|
| A15~A0 地址设定 寄存器 | 写入起始地址 | <pre>mov dx, PORT_ADDR mov ax, START_ADDR out dx, ax</pre> | <pre>outport (PORT_ADDR, START_ADDR);</pre> |
| 存储器 数据存取 寄存器 | 顺序读写数据 地址自动增加 | <pre>mov dx, PORT_DATA mov di, BUFFER_ADDR mov cx, LENGTH rep insb</pre> | <pre>int i; char buf[LENGTH]; for (i=0; i<LENGTH; ++i) buf[i]=inportb(PORT_DATA);</pre> |

程序中用到的常量和变量：

PORT_ADDR 代表 A15~A0 地址设定寄存器的端口地址（即：I/O 基址+0F0H）；

PORT_DATA 代表存储器数据存取寄存器的端口地址（即：I/O 基址+0F3H）；

START_ADDR 代表要存取的数据在扩展 ROM 或者存储器中的起始地址；

LENGTH 代表要存取的数据的长度，也就是字节数；

BUFFER_ADDR 代表存放读出数据的缓冲区的起始地址；buf 为数据缓冲区。