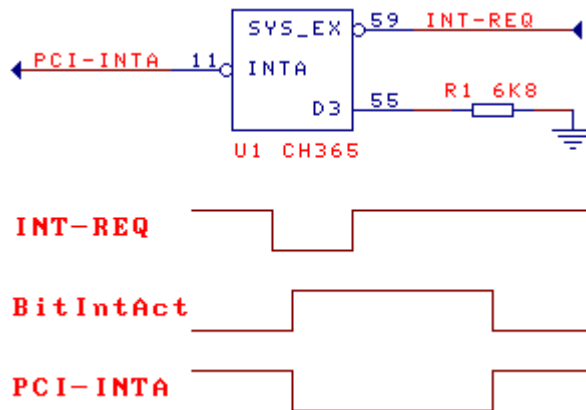
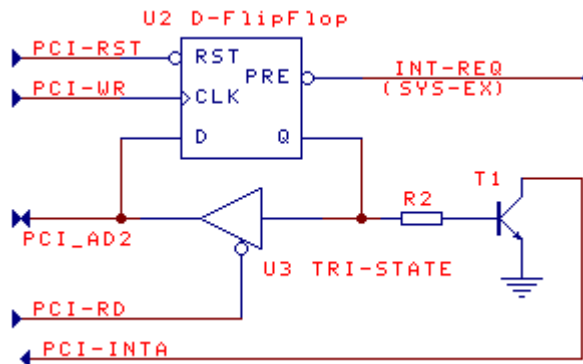


1、概述

CH365 芯片具有软件和硬件中断功能，与中断功能有关的两个引脚分别是 PCI_INTA 和 SYS_EX。如果 CH365 的数据线 D3 连接了下拉电阻，则工作模式设定为启用中断功能，SYS_EX 引脚自动复用为 INT_REQ，作为本地中断请求输入引脚，低电平有效，外部电路需要请求计算机中断时只要向该引脚提供低电平脉冲。PCI_INTA 总是作为 PCI 中断三态输出引脚，不用中断功能时可以悬空不接，需要中断功能时可以与 PCI 总线的 INTA 相连接，低电平有效，中断激活时输出低电平，中断未激活时输出高阻。



2、工作原理



上图是 CH365 芯片内部中断功能的简化原理图。

U2 是 D 触发器，用于提供 CH365 的中断激活状态位 BitIntAct（即芯片控制寄存器的位 2，在 CH365 的 I/O 空间的偏移地址 0F8H）。U3 是三态输出，用于在 PCI 读操作时输出中断激活状态。PCI 总线的复位信号 PCI_RST 连接到 U2 的复位端 RST，本地中断请求输入信号 INT_REQ 连接到 U2 的置位端 PRE。计算机程序可以通过读操作 PCI-RD 或者写操作 PCI-WR 查询或者修改 U2 的状态，也就是中断激活状态位。

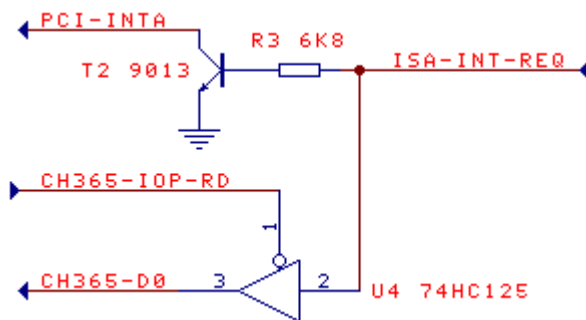
当 INT_REQ 引脚检测到低电平后，CH365 的中断激活状态位 BitIntAct 被设置为 1，CH365 通过 PCI_INTA 引脚向 PCI 总线申请中断。此时即使 INT_REQ 引脚再恢复为高电平，CH365 依然保持在中断激活状态，直到计算机的 CH365 中断服务程序通过写芯片控制寄存器将 CH365 的中断激活状态位清除为 0。但是，如果清除中断激活状态位后 INT_REQ 引脚仍然检测到低电平，则 CH365 再次进入中断激活状态。

如果计算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。

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 信号向计算机申请中断。
- ③ 计算机进入 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，撤消了对计算机的中断请求。
- 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，计算机退出 CH365 的中断服务程序。

3、升级 ISA 中断



如果用 CH365 升级已有的 ISA 板卡，则在中断功能方面存在下述情况：

- ① ISA 板卡的中断请求信号是高电平有效，而 CH365 的中断请求以及 PCI 总线的中断请求都是低电平有效。
- ② ISA 板卡的中断通常是独占使用的，而 PCI 板卡的中断通常是共享的。

参考上图，用外部电路部分代替 CH365 芯片本身的中断功能电路，具体包括：

- ① 将 CH365 的 PCI_INTA 引脚悬空，不连接 PCI 总线 INTA。
- ② 参照上图，用电阻 R3（6.8K Ω ）和三极管 T2（9013）组合成反相器，输入端连接原 ISA 板卡的中断请求 ISA_INT_REQ，输出端代替 CH365 的 PCI_INTA 引脚连接到 PCI 总线的 INTA 中断引脚。
- ③ 如果需要在 PCI 中断的共享，还需要增加中断状态查询电路，图中由 U4（三态输出电路 74HC125）向 CH365 的中断服务程序提供中断激活状态位。建议不使用上述外部电路，而使用 CH365 自身的中断电路，并在 INT_REQ 前加反相器。

如果不需要支持中断共享，则可以直接升级 ISA 板卡的中断。当原 ISA 中断请求信号 ISA_INT_REQ 高电平有效后，T2 将其反相后驱动 PCI 总线中断 PCI_INTA，计算机进入 CH365 的中断服务程序，完成中断事务处理后，应该通知外部电路撤消 ISA 中断请求信号，也就是将 ISA_INT_REQ 恢复为低电平，T2 截止。

如果需要在 PCI 中断的共享，则在计算机进入 CH365 的中断服务程序后，首先应该查询是不是本卡的中断请求，是则继续处理，否则直接退出中断，以便让其它卡的中断服务程序进行处理。以上图为例，可以执行 I/O 读操作并检查数据位 0 的状态，为 1 则是本卡的中断请求，否则是与本卡共享同一个中断的其它卡的中断请求。图中 U4 的输出使能信号直接是 IOP_RD，而在实际应用中，U4 的输出使能信号可能是 A7-A0 部分地址译码后与 IOP_RD 的逻辑组合，ISA 中断状态可以输出到 D7-D0 中的任一位。

PCI 总线各个板卡的中断号（或称为中断级 IRQ_L）通常是自动分配的，对于 DOS 或

者 Windows 98/ME/2000 操作系统，中断号可以是 0 至 15 之间的数值，实际上只有可能是 3、4、5、7、9、10、11、12、15，对于运行在较新计算机芯片组上的 Windows XP 操作系统，中断号还有可能是 16 至 23 之间的数值。

如果从原 ISA 卡直接升级，有可能 PCI 总线自动分配的中断号不一定与原 ISA 卡的中断号相同，而为了避免修改原 ISA 板卡的中断服务程序，可以强制设定 PCI 板卡的中断号。有些主板的 BIOS 允许手工设定 PCI 板卡中断号，一般是在 CMOS 中 ISA PNP/PCI 等类似的子菜单中。如果 CMOS 中没有提供中断设定功能，则可以使用程序进行设定。

3.1. 强制设定中断号（只在 BIOS 或者 DOS 下有效，不支持 Windows 系统）

C 语言的源程序示例

```
union REGS  mReg;
struct SREGS mSegReg;
mSegReg.ds = 0xf000;
mReg.x.bx = dosCH365PciAddr;
/* dosCH365PciAddr 是 CH365 板卡的地址，即总线号/设备号/功能号 */
/* 该地址可以通过搜索 CH365 板卡获得，具体可以参考 CH365DOS.C */
mReg.x.ax = 0xb10f; /* 设定硬件中断 IRQ */
mReg.h.cl = 0x0a; /* INTA */
mReg.h.ch = New_IRQ_Number; /* 新中断号，数值可以是 0 至 15 */
int86x ( 0x1a, &mReg, &mReg, &mSegReg ); /* 调用 PCI 的 BIOS */
```

汇编语言的源程序示例

```
push  ds
mov   ax, 0f000h
mov   ds, ax
mov   bx, dosCH365PciAddr
mov   ax, 0b10fh
mov   cl, 0ah
mov   ch, New_IRQ_Number ;新中断号
int   1ah
pop   ds
```

3.2. 获得自动分配的中断号，或者回读手工设定的中断号

C 语言的源程序示例

```
union REGS  mReg;
mReg.x.bx = dosCH365PciAddr;
mReg.x.ax = 0xb108; /* 读取配置空间一个字节 */
mReg.x.di = 0x003c; /* mPcInterLine, 中断号或者中断线寄存器 */
int86 ( 0x1a, &mReg, &mReg ); /* 调用 PCI 的 BIOS */
返回 mReg.h.cl 是自动分配的中断号，是 0 至 15 之间的一个数。
```

汇编语言的源程序示例

```
mov   bx, dosCH365PciAddr
mov   ax, 0b108h
mov   di, 003ch          ;mPcInterLine, 中断号或者中断线寄存器
int   1ah
返回 cl 寄存器是自动分配的中断号
```

4、中断程序编写

以下涉及的 C 语言源程序可以参考 CH365DOS.C 以及 CH365CKD.C。

4.1. 关于中断屏蔽

PC 机的 8259 具有中断屏蔽寄存器，当屏蔽位为 0 时允许中断，为 1 时禁止中断。由于 PCI 板卡的中断通常是自动分配的，所以只能在获取自动分配的中断号后才能确定中断屏蔽位。以下是 C 语言源程序。

```
/* dosIntLine 是自动分配的中断号 */  
unsigned char    mIntMask, mIntPort;  
mIntPort = dosIntLine < 8 ? 0x21 : 0xa1; /* 中断屏蔽设置端口 */  
mIntMask = inportb( mIntPort ); /* 保存原中断屏蔽字节 */  
outportb( mIntPort, mIntMask & ~ ( 0x01 << ( dosIntLine & 0x07 ) ) );  
/* 根据中断号移位并取反，允许指定中断号的中断 */
```

4.2. 关于中断结束

PC 机的 8259 支持中断嵌套，要求在一个中断服务完成后送出中断结束 EOI 命令，也就是硬件中断完成后必须设置结束标志。由于 PCI 板卡的中断通常是自动分配的，所以中断服务程序在发出 EOI 命令时需要考虑级联嵌套关系。以下是 C 语言源程序。

```
/* dosIntLine 是自动分配的中断号 */  
if ( dosIntLine >= 8 ) { /* 中断号 8 至 15，被连接到从 8259 上 */  
    outportb( 0xa0, 0x20 ); /* 中断号 8 至 15，向从 8259 发出 EOI */  
    if ( inportb( 0xa0 ) == 0 ) outportb( 0x20, 0x20 );  
    /* 如果从 8259 没有中断嵌套，则向主 8259 发出 EOI */  
}  
else outportb( 0x20, 0x20 ); /* 中断号 0 至 7，向主 8259 发出 EOI */
```