

接口芯片和端口

1. 端口的读写只有**in,out** , 一个读一个写 (CPU的视角来看)
2. 访问端口
CPU 通过地址线 将 **60h**的信息发出
通过控制线发出读命令
端口所在的芯片将**60h**端口的数据通过数据总线将数据送入CPU

外中断

1. 可屏蔽中断

1. IF=0 CPU 可以不响应可屏蔽中断
2. IF=1 执行完当前指令后执行中断

2. 不可屏蔽终端

1. 不可屏蔽中断的中断类型码固定为**2**

PC机键盘的处理过程

1. 键盘输入

- 扫描码 (通码) 送入 **60h**
- 断码 = 通码 + 80h

2. 引发9号中断

- 有数据送入 **60h**
- 芯片向 **CPU** 发出 中断类型码为 **9**的可屏蔽中断
- 如果 **IF=1** CPU 执行中断例程

3. 执行9号中断例程

int9 的基本键盘处理流程

- 读出 **60h** 的扫描码
 - 如果是扫描码 将扫描码和字符码送入内存中的键盘缓冲区
 - 如果是控制键的扫描码则将其转变为状态字节存入内存中的状态字节的单元
- 对键盘系统进行相关控制

编写中断例程

新的中断例程大概为

1. 从60h端口读出键盘的输入
 - a. `in al 60h`
2. 调用BIOS 的int9 的中断例程
 - a. `pushf`
 - b. `call dword ptr cs:[200h] ;cs==0`
3. 功能的实现