输入输出访问

对IO的访问会送出IO端口号。

对存储器的访问会送出存储单元的地址。

硬件端口：计算机与外部通讯设备进行信息交互的接口，每个接口有固定的地址。

软件端口：计算机与网络通讯设备进行交流的通讯协议的端口。

处理器可以从端口发送数据，也可以接受数据。

汇编语言的IN和OUT来进行读写数据。

默认数据在AL或AX寄存器中，读取的时候读取到AL或AX中。

需要有相应的接口电路和软件实现来实现CPU与设备的交互：

设备按照传送的信息分类，有三种类型的信息：控制、状态、数据

每种信息需要占有一个或多个端口，所以每个IO设备会有若干个端口。

通常对IO设备的寻址，是对该设备的端口的寻址。

比如CPU要打印信息，则需要一些信息：

控制信息：启动打印、暂停打印

状态信息：是否连接、缺纸、缺墨、打印中(busy)

数据信息：打印的内容

IN指令

格式： IN destination, source

IN AL, port#

port#是端口号

样例：

MOV DX, port#

IN AL, DX

OUT

格式： OUT destination, source

OUT port#, AL

port#是端口号

样例：

MOV DX, port#

OUT DX, AL

接口电路解决的问题：

1. 速度匹配问题：CPU速度快，外设速度慢

使用缓冲区

1. 外设的数据总线和控制总线不能直接连接CPU总线上，没有用的时候必须是逻辑断开的，否则外设会一直占有总线。

使能电路，使用缓冲器、锁存器，高阻态

1. 不同外设的数据的格式是不同的

格式转换电路

1. 不同外设电平的规范不一样

电平转换电路

译码器：输入特定的编码才能输出一个有效信号

如输入1001的时候才会输出1，否则只会输出0

独立编址

存储器和I/O端口地址可以相同，根据IO/M信号来区分是IO端口还是存储器地址

存储器映射IO编址

存储器地址和I/O端口地址不可以相同，不需要IO/M信号。

不同点：

1. 不需要IN和OUT指令了，像对内存的访问一样了，用MOV
2. 需要把所有的地址线都引出来，地址译码器的设计会比较复杂
3. 对IO的读和写与对内存的读和写一样了

8255芯片

并行接口芯片

3个输入输出接口，都有8位，称为A口B口C口

C口的8位可以分为两个4位的，可以分别分配成4位输入4位输出。

3个口都有自己的端口地址。

8255内部有控制寄存器，可以规定3个口的工作状态。控制寄存器也有自己的端口地址，所以总共要设置4个端口地址，只要两个位来设置，用A0A1来设置。

剩下的高位地址信号是用来连接一个地址译码器，然后地址译码器的输出作为片选信号。

D0到D7可以直接与数据总线相连接，因为只要8255的片选信号没有选中的时候是高阻态输出，不会影响数据总线的工作。

对控制寄存器的写入：

假设8255的端口为820H~823H，所以控制寄存器的端口是323H

把C口的第7位设置为1

MOV AL, 0FH ;设置的控制字

MOV DX, 323H

OUT DX, AL

单片机：单一芯片的计算机。把控制单元、算术逻辑单元、存储单元、输入输出设备集中在一起。

把CPU和8255芯片接在一起，然后设计一个ROM，就做成一个简单单片机了。

8253芯片：计数器

有3个计数器，一个控制寄存器，每个都有一个端口地址，所以总共有4个端口地址。