

《微型计算机原理与接口技术》

第5版

第6章

I/O接口和并行接口

芯片8255A



本章主要内容：

§6.1 I/O接口

§6.2 8255A的工作原理

§6.3 8255A的应用举例



§6.1 I/O接口

6.1.1 I/O接口的功能

6.1.2 I/O端口及其寻址方式

6.1.3 CPU与外设间的数据传送方式

6.1.4 PC机的I/O地址分配



6.1.1 I/O接口的功能

1. 采用I/O接口的必要性

- 计算机与外设间交换数据、状态和控制命令的过程统称为通信(Communication)。
- CPU与外设交换信息的过程，和它与存储器交换数据那样，也是在控制信号的作用下通过数据总线来完成的。
- 存储器芯片的存取速度与CPU的时钟频率在同一数量级，存储器本身又具有数据缓冲能力，所以CPU与存储器可以很方便地交换数据，但与外设交换数据的过程要复杂得多。



计算机与外设间的信息交换存在的问题

1) 速度不匹配

CPU速度比外设的速度高很多，且不同外设速度差异甚大。

2) 信号电平不匹配

CPU都用TTL电平，而外设大多是复杂的机电设备，往往不能为TTL电平所驱动，有自己的电源系统和信号电平。

3) 信号格式不匹配

CPU传送的通常是8位、16位或32位并行数据，而外设使用的信息格式各不相同。有模拟量、数字量或开关量；有电流量、电压量；有些采用串行方式，有些用并行方式。

4) 时序不匹配

外设都有各自的定时和控制逻辑，与CPU的时序不一致。

- 因此，输入输出设备不能直接与CPU的系统总线相连，必须在CPU与外设之间设置专门的接口(Interface)电路来解决这些问题。



2.接口的功能

1) 设置数据缓冲解决速度不匹配问题

事先把要传送的数据准备好，在需要的时刻完成传送。经常使用锁存器和缓冲器，并配以适当的联络信号来实现这种功能。

2) 设置电平转换电路解决电平不一致问题

如计算机和外设间进行串行通信时，可采用MAX232和MAX233等芯片来实现电平转换，在第9章讨论。

3) 设置信息转换逻辑满足各自格式要求

将外设传送的模拟量，经A/D转换成数字量，送到计算机去处理。计算机送出的数字信号经D/A转换成模拟信号，驱动某些外设工作。



2. 接口的功能

4) 设置**时序控制**电路同步CPU和外设的工作

接口电路接收CPU送来的命令或控制信号、定时信号，实施对外设的控制与管理，外设的工作状态和应答信号也通过接口及时返回CPU，以**握手联络(handshaking)**信号来保证主机和外部I/O操作实现同步。

5) 提供**地址译码**电路

计算机中存在多个外设，每个外设需要与CPU交换几种信息，因此接口电路中常含若干**端口**，其I/O地址由接口电路中的地址译码电路提供。

6) 提供I/O控制、读/写控制及中断控制等逻辑。



- ◆ 可见，I/O接口电路是外设与计算机之间传送信息的交接部件，它在两者间起协调作用，每个外设都要通过接口电路才能与主机相连。
- ◆ 随着VLSI技术的发展，出现了许多通用可编程接口芯片，可利用它们方便地构成接口电路。
- ▶ 本章先介绍接口的基本知识，接着介绍可编程并行接口芯片8255A。后面几章将重点讨论其它几种常用可编程I/O接口芯片的工作原理、编程方法以及这些芯片如何与CPU和外设相连等问题。



§6.1 I/O接口

6.1.1 I/O接口的功能

6.1.2 I/O端口及其寻址方式

6.1.3 CPU与外设间的数据传送方式

6.1.4 PC机的I/O地址分配



6.1.2 I/O端口及其寻址方式

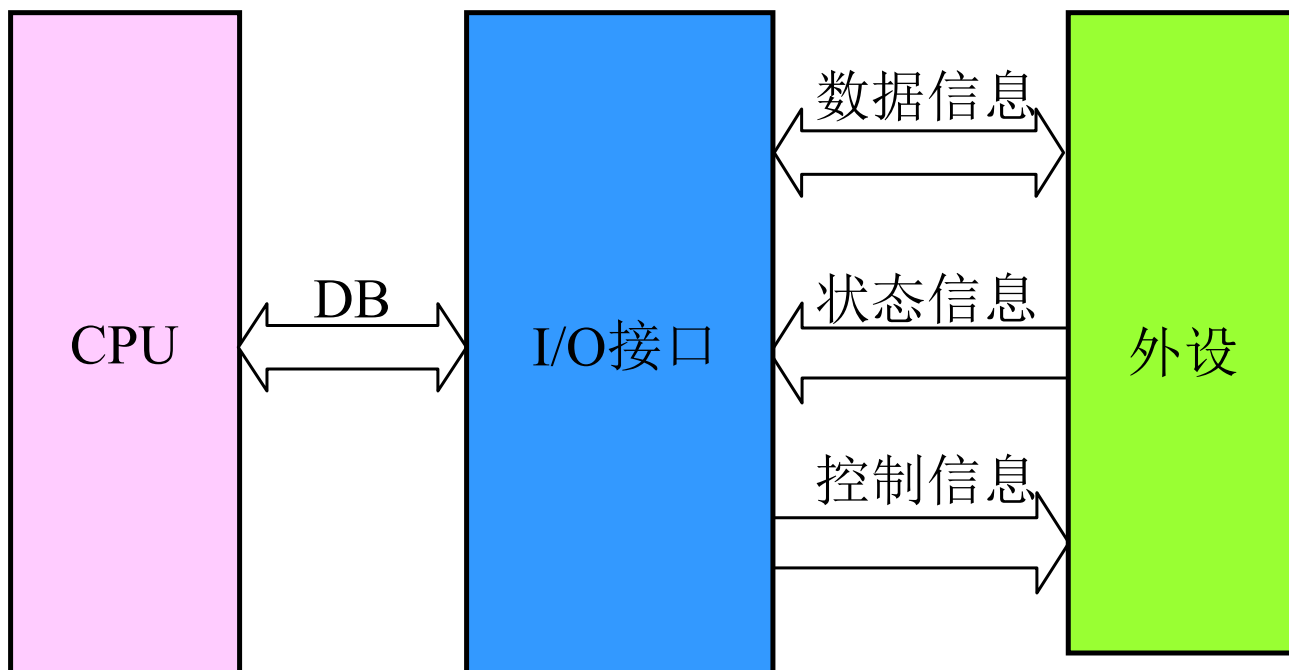
1. I/O端口

- CPU与外设通信时，主要传送数据信息、状态信息和控制信息。
- 在接口电路中，这些信息分别进入不同的寄存器，通常将这些寄存器和它们的控制逻辑统称为I/O端口(Port)，CPU可对端口中的信息直接进行读写。
- 在一般接口电路中都要设置以下几种端口：

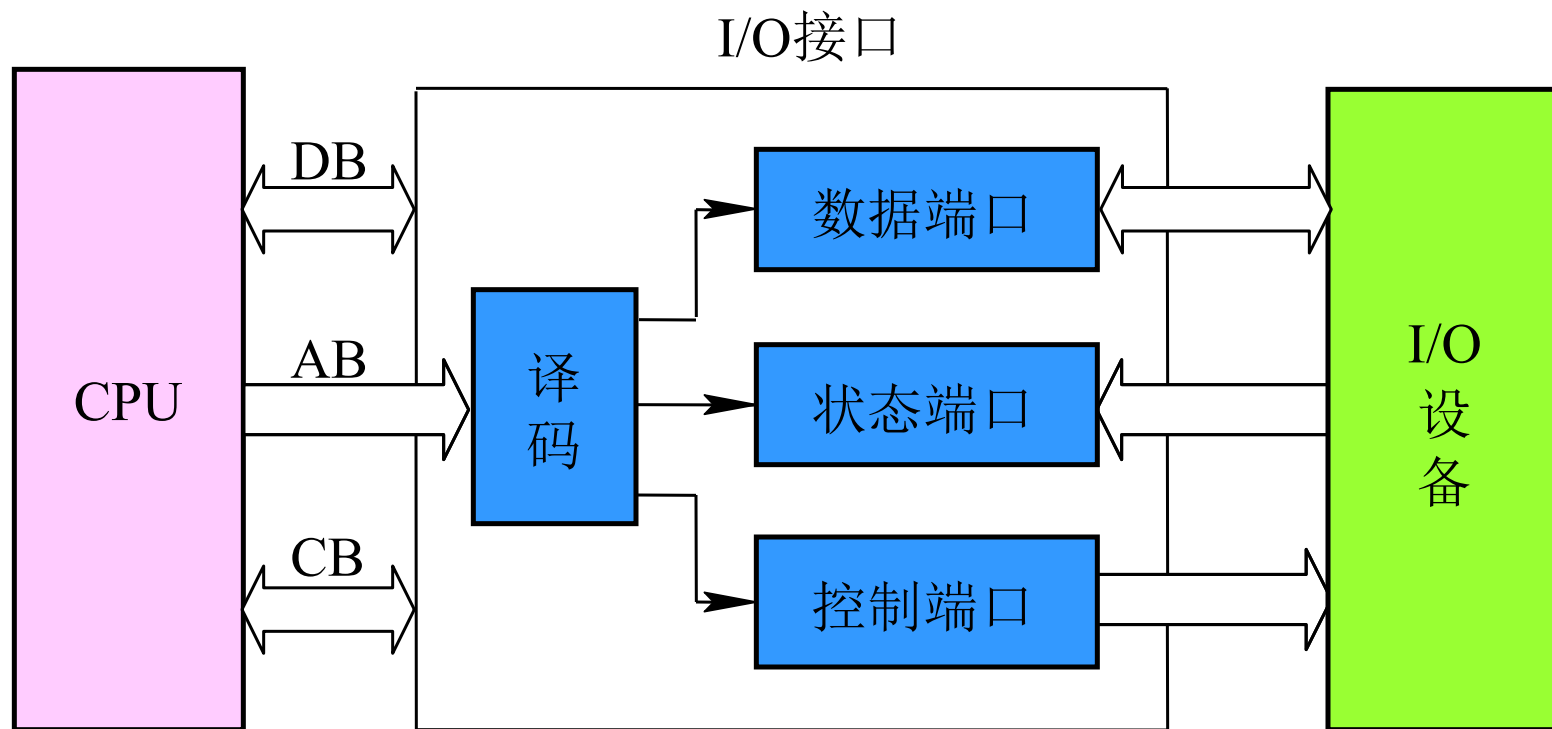


2. CPU与外设交换的信息

数据信息、状态信息和控制信息



3. I/O接口的基本结构



I/O端口是接口中存放不同信息的寄存器。接口和端口是两个不同的概念，若干个端口加上相应的控制电路才构成接口。

1. I/O端口

1) 数据端口

- 数据端口(Data Port)或数据口，用来存放外设送往CPU的数据，或者CPU要输出到外设去的数据，长度一般为1~2字节。数据口主要起数据缓冲作用。

2) 状态端口

- 状态端口(Status Port)指示外设的当前状态。每个外设具有几个状态位，它们可由CPU读取，以测试或检查外设的状态，决定程序的流程。



1. I/O端口

状态口中常用的状态位：

□ 准备就绪位(Ready)

- 如果是输入端口，该位为1表明端口的数据寄存器已准备好数据，等待CPU来读取；当数据被取走后，该位清0。
- 若是输出端口，该位为1表明端口中的输出数据寄存器已空，可以接收CPU的下一个数据了；当新数据到达后，这位便清0。



1. I/O端口

❑ 忙碌位(Busy)

表明输出设备是否能接受数据。

- 若该位为1，表示外设正在进行I/O传送操作，暂时不允许CPU送新的数据过来。
- 本次数据传送完毕，该位清0，表示外设已处于**空闲状态**，又允许CPU将下一个数据送到输出口。

❑ 错误位(Error)

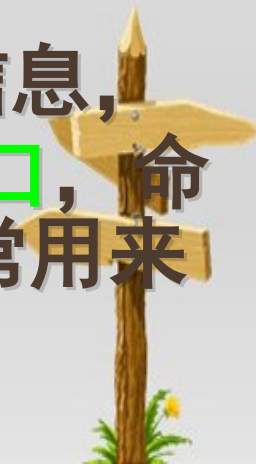
- 如果在数据传送过程中产生了某种错误，可将错误状态位置1，以便CPU进行相应的处理。
- 系统中可以设置若干**错误状态位**，表明不同性质的错误，如**奇偶校验错**、**溢出错**等。



1. I/O端口

3) 命令端口

- 命令端口(Command Port)也称为控制端口(Control Port)，用来存放CPU向接口发出的各种命令和控制字，控制接口或设备的动作。
- 常见的命令信息位有启动位、停止位、允许中断位等。
- ▶ 通常，CPU与外设交换的数据以字节为单位，因此一个外设的数据端口含有8位。
- ▶ 状态口和命令口可以只包含1位或几位信息，所以不同外设的状态口允许共用一个端口，命令口也可共用。D触发器和三态缓冲器常用来构成这两种端口。



4. I/O端口的编址

接口中每个端口都有一个地址编号，称为端口地址。

微机系统中，I/O端口的编址方式有两种，即：

统一编址 独立编址

I/O端口的统一编址方式

I/O端口和存储单元统一编排地址号，由I/O端口地址和存储单元地址共同构成一个统一的地址空间。

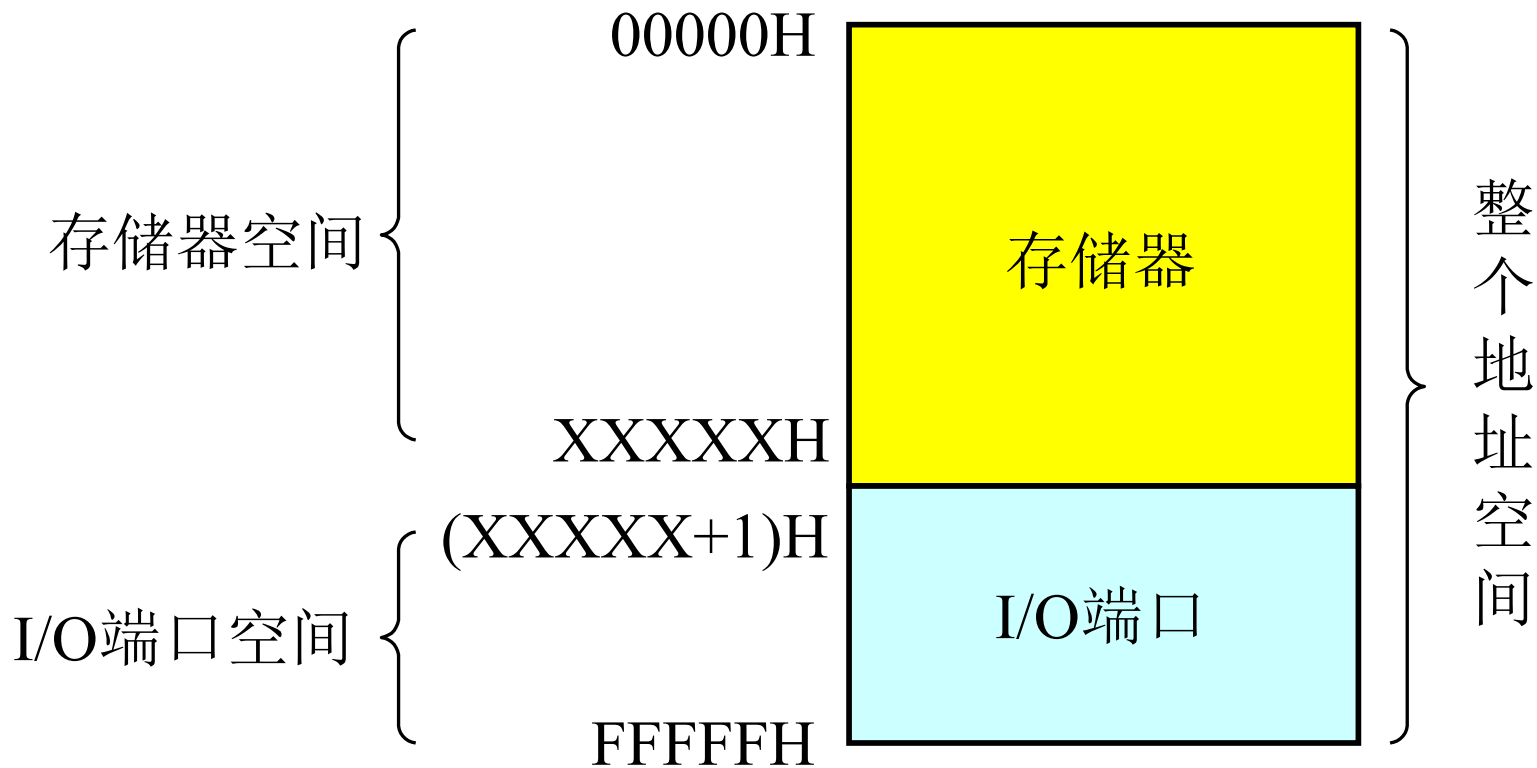


图7.4 I/O端口与内存单元统一编址

统一编址方式的特点

优点： 1、简化了指令系统的设计；

2、I/O端口和存储单元可用同样的寻址方式，给编程者提供了很大的方便。

缺点： 减少了内存的地址可用范围。

I/O端口的独立编址方式

建立了两个地址空间，一个为内存地址空间，一个为I/O地址空间。内存地址空间和I/O地址空间是相对独立的。

80x86 CPU组成的微机系统都采用独立编址方式。

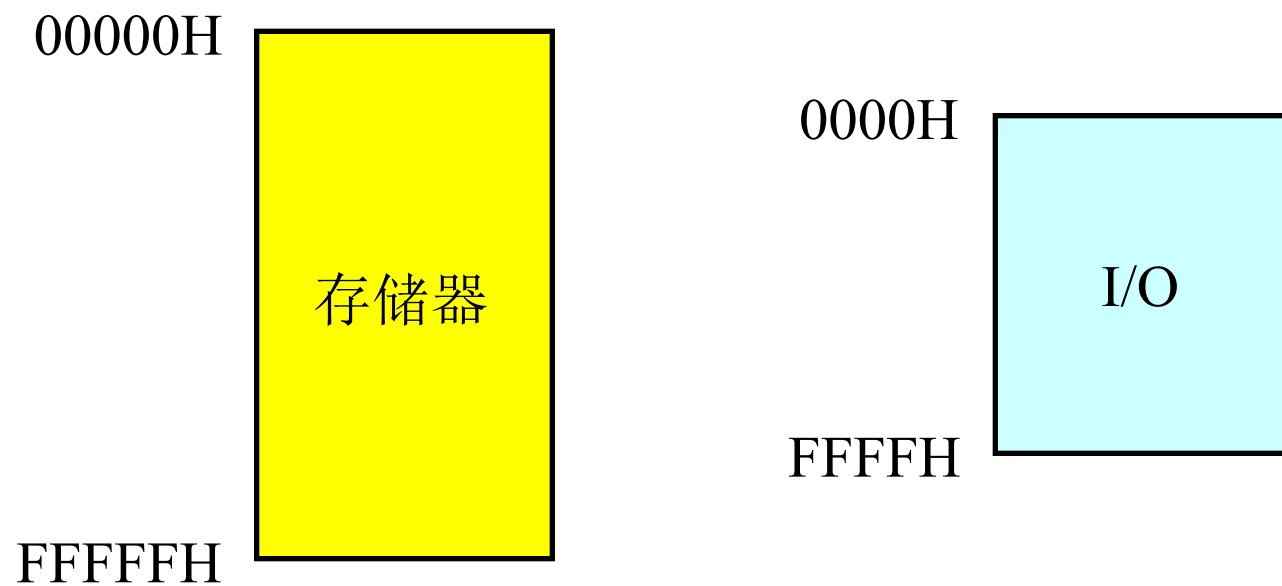


图7.5 I/O端口与内存单元独立编址

优点： 1、存储器地址空间不受I/O端口地址空间的影响
2、专用的输入/输出指令与访问存储器指令有明显区别，便于理解和检查。

缺点： 1、专用I/O指令增加了指令系统复杂性，且I/O指令类型少，程序设计灵活性较差；
2、要求CPU提供专门的控制信号以区分对存储器和I/O端口的操作，增加了控制逻辑的复杂性。

§6.1 I/O接口

6.1.1 I/O接口的功能

6.1.2 I/O端口及其寻址方式

6.1.3 CPU与外设间的数据传送方式

6.1.4 PC机的I/O地址分配



6.1.3 CPU与外设间的数据传送方式

1. 程序控制方式（软件）

数据传送在程序控制下完成，又可分为无条件传送和条件传送两种方式。

2. 中断方式（软件）

外设发中断请求，CPU响应后完成数据传送。

3. DMA方式（硬件）

DMA控制器临时接管CPU的地址、数据和控制总线，实现批量数据的传送。



1. 程序控制方式

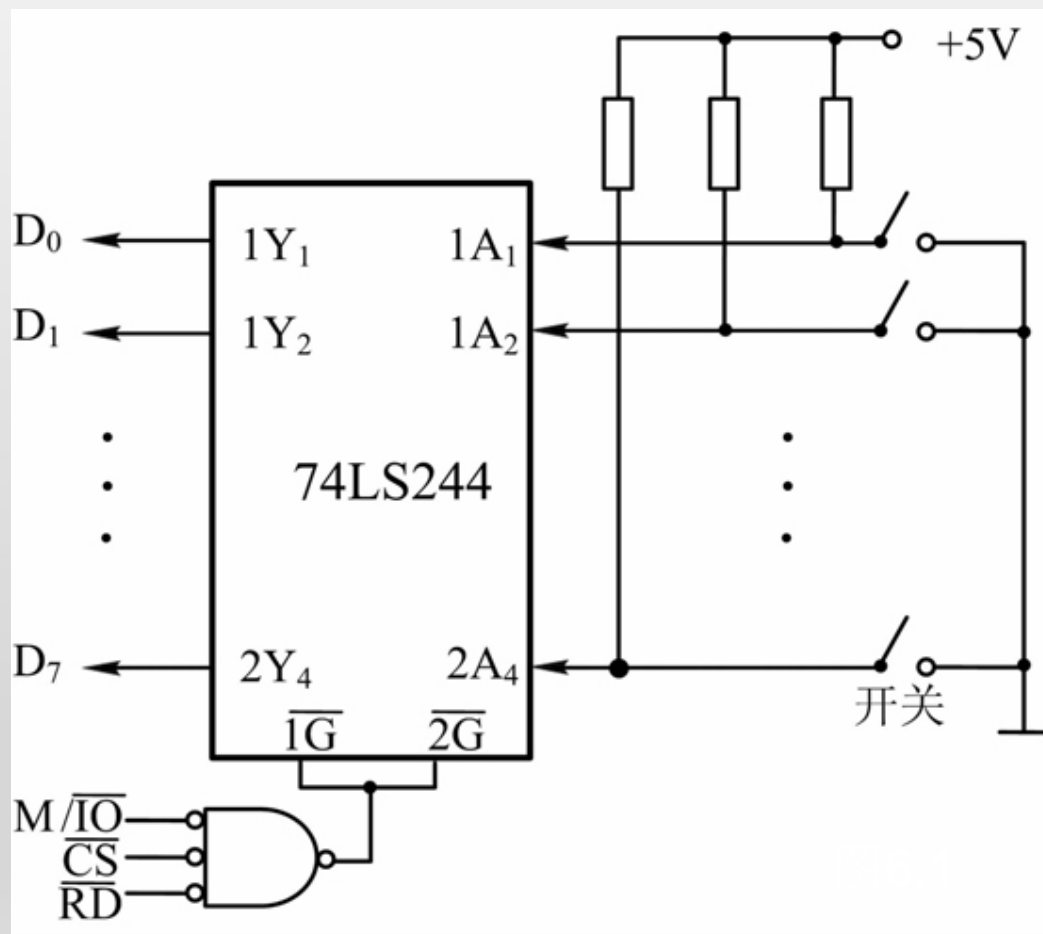
1) 无条件传送方式

- 也称**同步传送方式**，主要用于对简单外设进行操作，所需的硬件和软件都较少。
- 对于这类外设，任何时刻在输入时均已准备好数据，输出时已处于接收数据状态，程序可不必检查外设的状态。
- 在需要进行输入或输出操作时，可以直接执行输入输出指令，I/O指令执行后，立即进行数据传送操作。

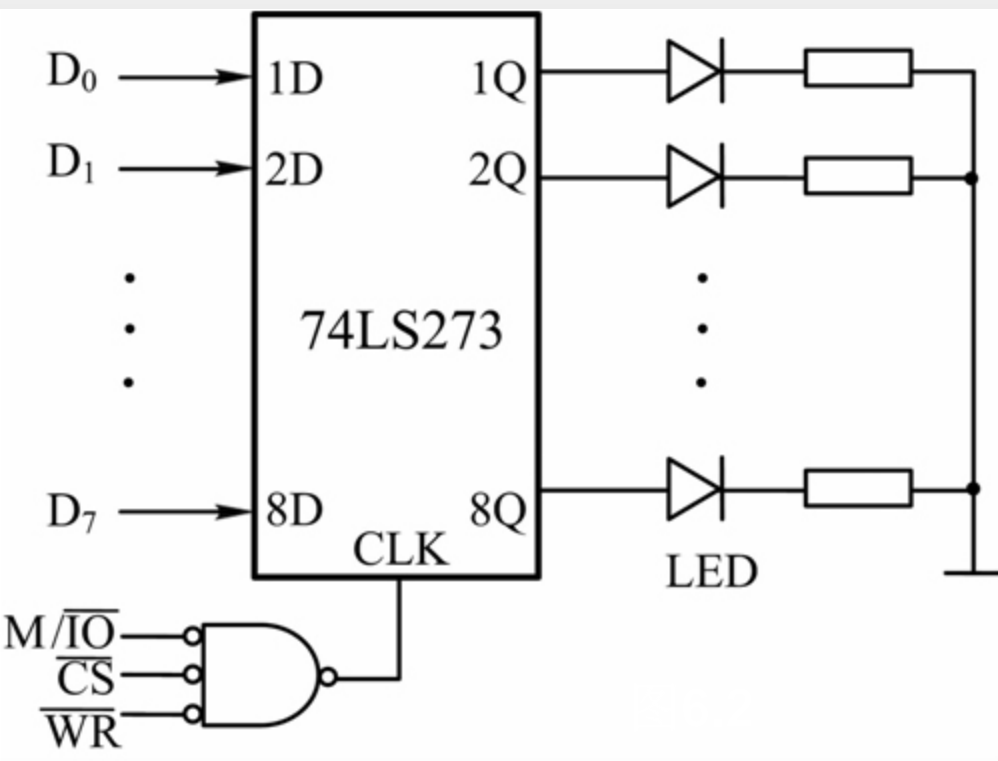


1) 无条件传送方式

- 最简单的输入端口：按键开关的状态输入入口。
- 开关经1个8位三态缓冲器接到数据总线，用IN指令将开关状态读入缓冲器。1-断开，0-闭合。
- 可随时读取开关状态。



1) 无条件传送方式



- 最简单的输出口：
LED的点亮和熄灭控制。
- 1个8位锁存器接8个
LED显示器，用OUT
指令将命令字送到
LED，1-点亮，0-熄灭。
- 可随时改变**LED**状态。



1. 程序控制方式

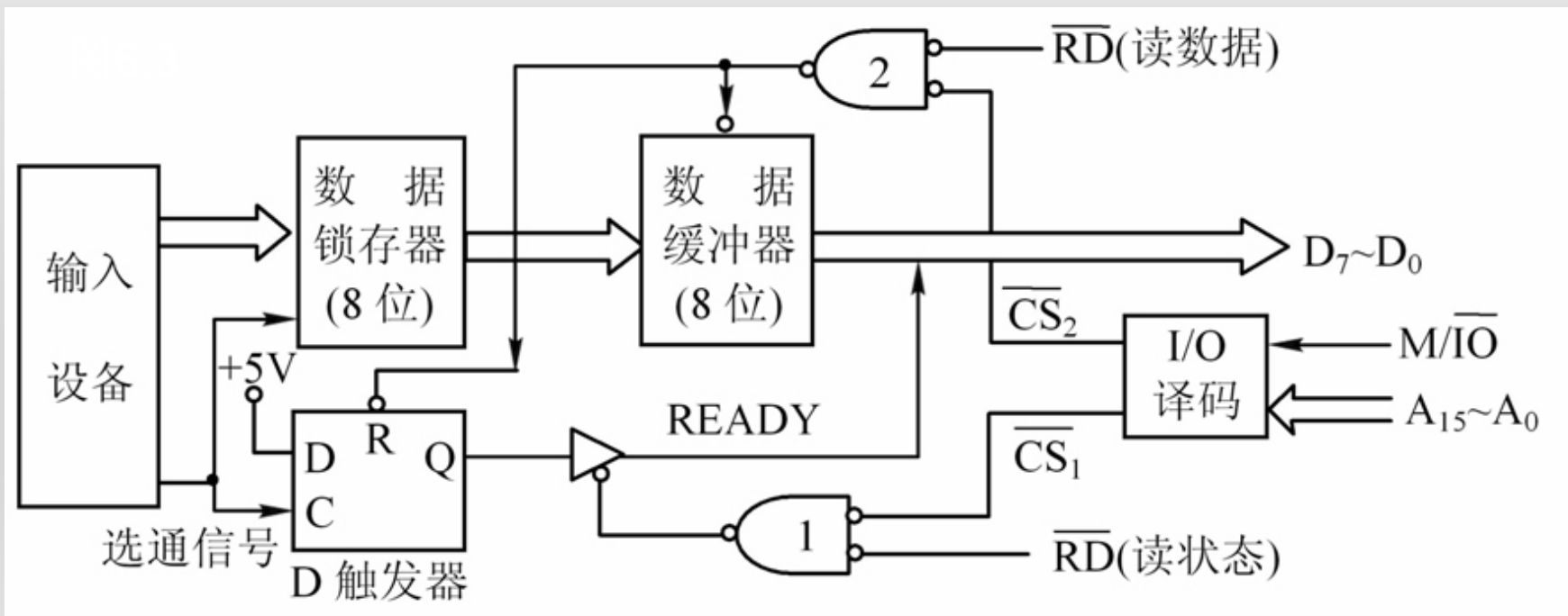
2) 条件传送

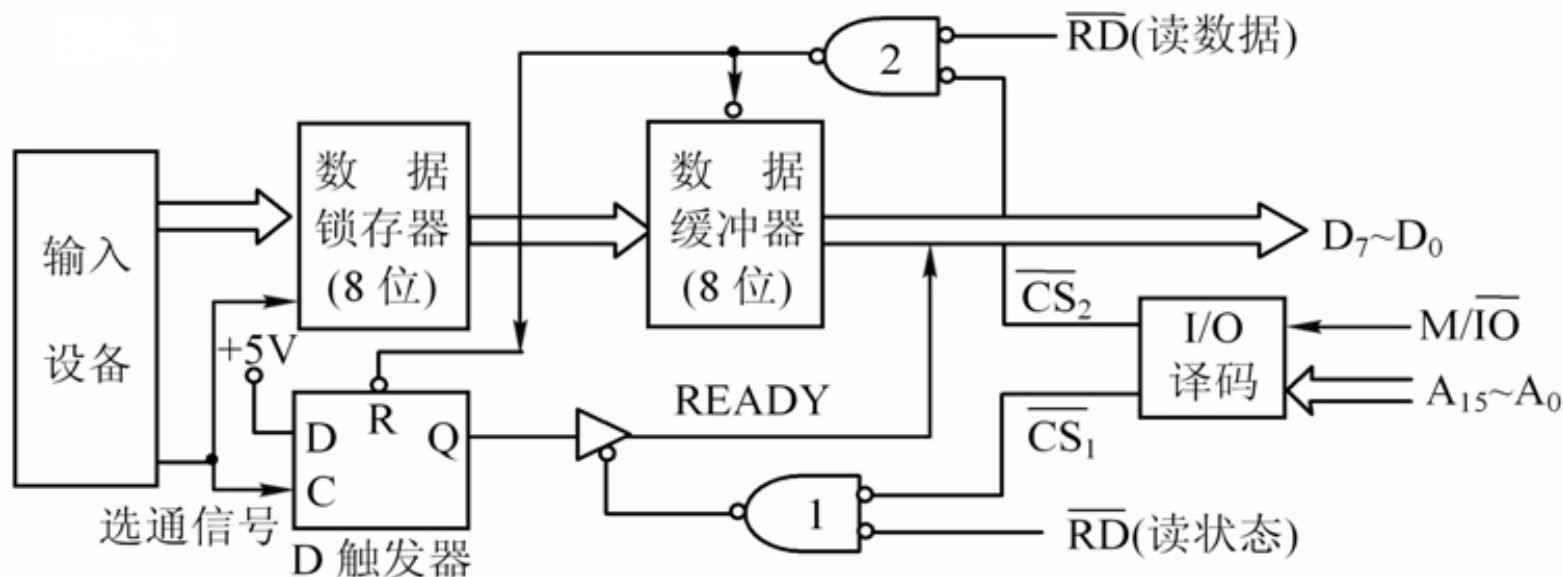
- 也称为**查询式传送方式**。CPU与外设交换数据一般很难满足无条件传送的条件。在开始传送前应先确认外设已处于**准备好传送数据的状态**，才能进行传送，即采用查询式传送方式。
- 这时CPU要先执行一条IN指令，从外设的状态口读取它的当前状态。如果外设**未准备好数据**或处于**忙碌状态**，则要反复执行读状态指令，不断检测外设状态，直至外设准备好了或不忙时为止。



● 查询式输入接口电路

- 包括输入数据口和状态口。状态口($\overline{CS1}$ 有效)由1个D触发器和1个三态门构成, 输入数据口($\overline{CS2}$ 有效)由1个8位锁存器和1个8位缓冲器构成。
- 执行IN指令选中状态口时, 读取状态信息READY;
执行IN指令选中数据口时, 从缓冲器读取数据信息。

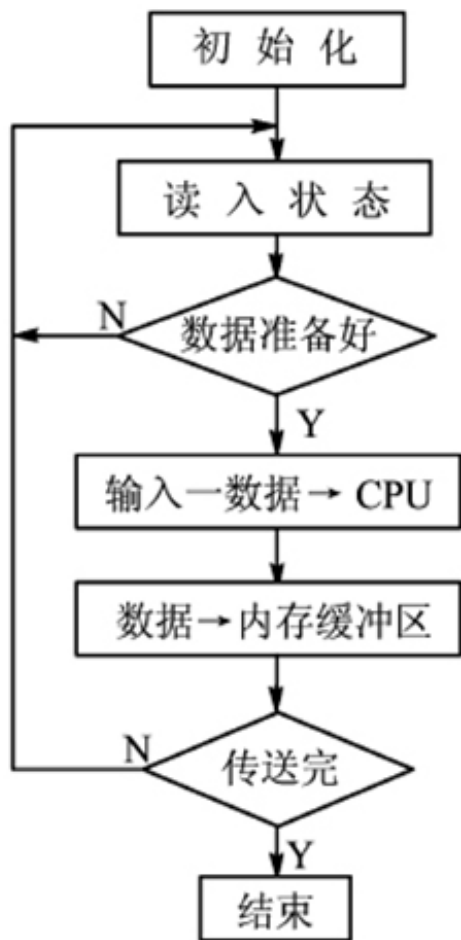




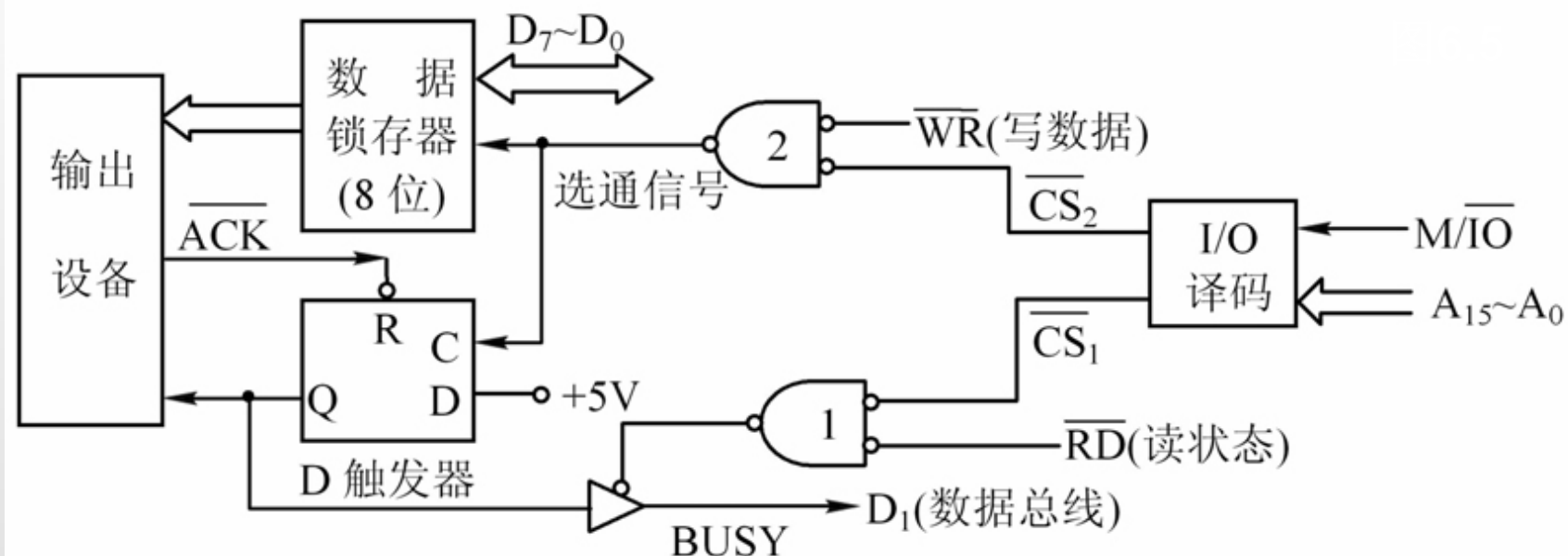
工作过程:

- ① 输入设备准备好数据后，向接口电路发选通信号，将数据打入锁存器，并使D触发器的Q端置1。
- ② 执行IN指令读状态口，**CS1**和**RD**有效。Q端高电平经缓冲器送到数据线，作为READY信号读入累加器。
- ③ CPU查READY=1，执行IN指令读数据口，**CS2**和**RD**有效使数据缓冲器开启，锁存数据经缓冲器送上数据总线并进入累加器，同时将D触发器清0，一次数据传送完毕。
- ④ 接着开始下个数据的传送，直至全部数据传送完毕。

● 查询式输入数据的程序段：



MOV BX, 0 ; 初始化地址指针
MOV CX, COUNT_1 ; 传送字节数
READ_S1:
IN AL, PORT_S1 ; 读入状态位
TEST AL, 01H ; 数据准备好?
JZ READ_S1 ; 否, 循环检测
IN AL, PORT_IN ; 准备好, 读数据
MOV [BX], AL ; 存入内存缓冲区
INC BX ; 修改地址指针
LOOP READ_S1 ; 未完, 继续
... ; 已传送完



● 查询式输出电路和工作过程

- ① CPU准备输出数据时先执行IN指令，它使状态口三态门开启，从数据总线D₁位读入BUSY状态。BUSY=1，外设忙；BUSY=0，CPU可向外设输出数据。
- ② 外设不忙时，执行OUT指令选中数据口。**CS2**和**WR**有效，门2输出低电平选通信号，锁存器选通，将数据送向外设。同时，其后沿还使D触发器翻转，置Q为高电平，即把状态口的BUSY位置成1，表示忙碌。
- ③ 输出设备取走数据后，送回一个应答信号，将D触发器清0，即置BUSY=0，允许CPU送出下个数据。

2) 条件传送

- 查询式输出数据的程序段：

```
MOV    CX, COUNT_2    ; 传送的字节数
READ_S2:
IN      AL, PORT_S2    ; 读入状态位
TEST    AL, 02H        ; 忙否?
JNZ     READ_S2        ; 忙，循环检测
MOV     AL, 待输出数据 ; 不忙
OUT     PORT_OUT, AL   ; 输出数据
LOOP    READ_S2        ; 未传送完，循环
...
; 已传送完
```



2. 中断方式

- 查询式数据传送，CPU会化许多时间去等待数据或外设准备就绪，效率很低。

例如：若一个操作员每秒可从键盘输入5个字符，平均每个字符占 $200000\mu\text{s}$ ，而计算机只要用 $10\mu\text{s}$ 就能从键盘读入1个字符，这样就有 $999950\mu\text{s}$ 花在检测键盘状态和等待上，即99.99%的时间因等待而浪费了。



2. 中断方式

- 采用中断方式后，CPU平时可以执行主程序，只有当输入设备将数据准备好了，或者输出端口的数据缓冲器已空时，才向CPU发中断请求。CPU响应中断后，暂停执行当前的程序，转去执行管理外设的中断服务程序。
- 在中断服务程序中，用输入或输出指令在CPU和外设之间进行一次数据交换。等输入或输出操作完成之后，CPU又回去执行原来的程序。
- ▶ 有关中断的概念，将在第8章中详细介绍。



3. DMA方式

1) DMA方式的提出

- 当CPU与高速I/O设备交换数据，或者与外设进行成组数据交换时，中断方式仍显得太慢。
- DMA(Direct Memory Access)传送方式，也就是直接存储器存取方式，可以在不受CPU干预的情况下，实现存储器与外设间的高速交换数据。



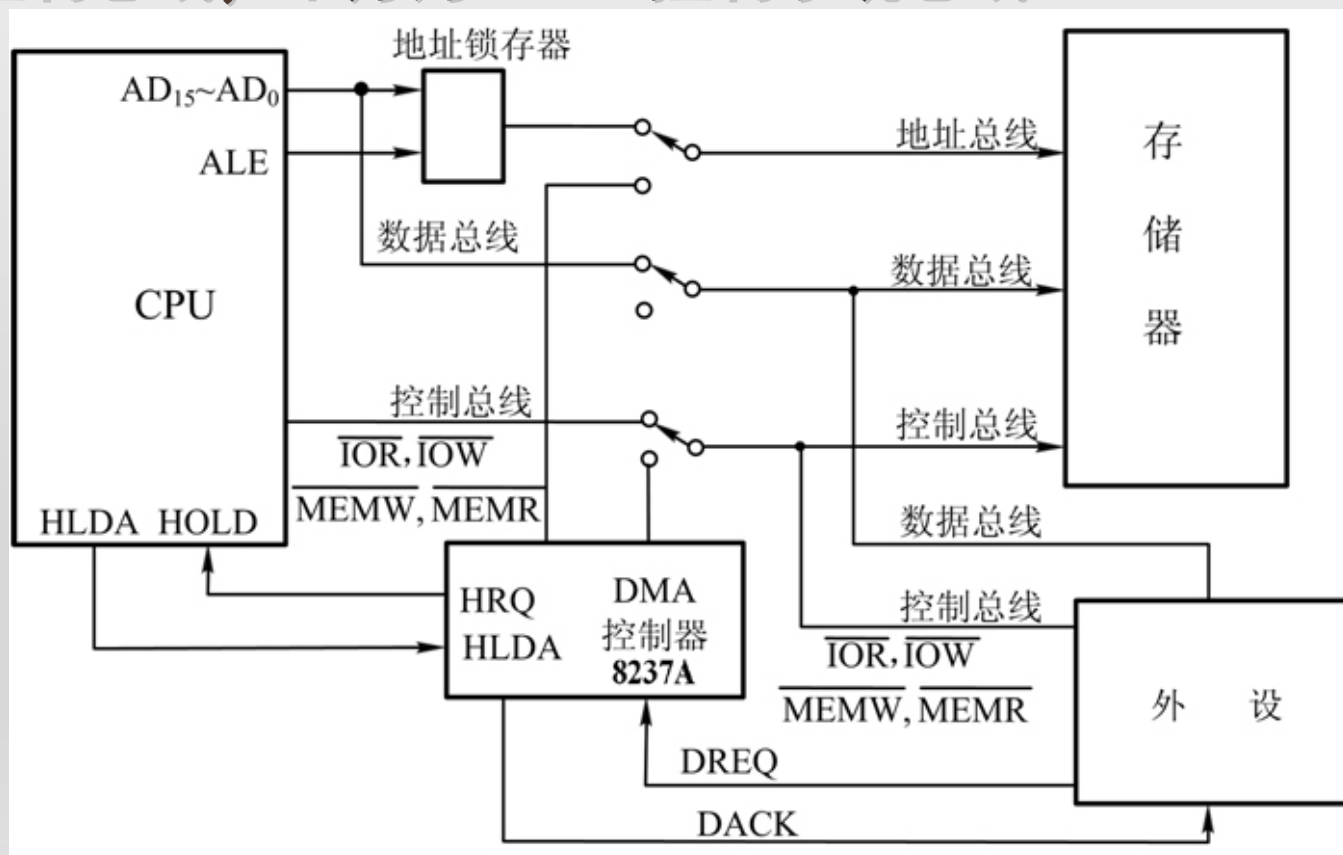
3. DMA方式

- DMA方式也要利用系统的数据总线、地址总线和控制总线来传送数据。
- 平时总线由CPU管理，当外设需用DMA方式传送数据时，可请求CPU让出总线控制权，用专用硬件接口电路（DMA控制器）来取代CPU，临时接管总线，控制外设和存储器之间直接进行成批、高速的数据传送，不用CPU干预。
- DMA控制器功能：能给出访问内存所需的地址信息，并能自动修改地址指针；能设定和修改传送的字节数，还能向存储器和外设发出相应的读/写控制信号；DMA传送结束后，能释放总线，把总线控制权交还给CPU。
- ▶ 8237A是一种典型的DMA控制器，其工作原理和使用方法等，将在第11章专门介绍。



2) DMA控制器的连线和操作

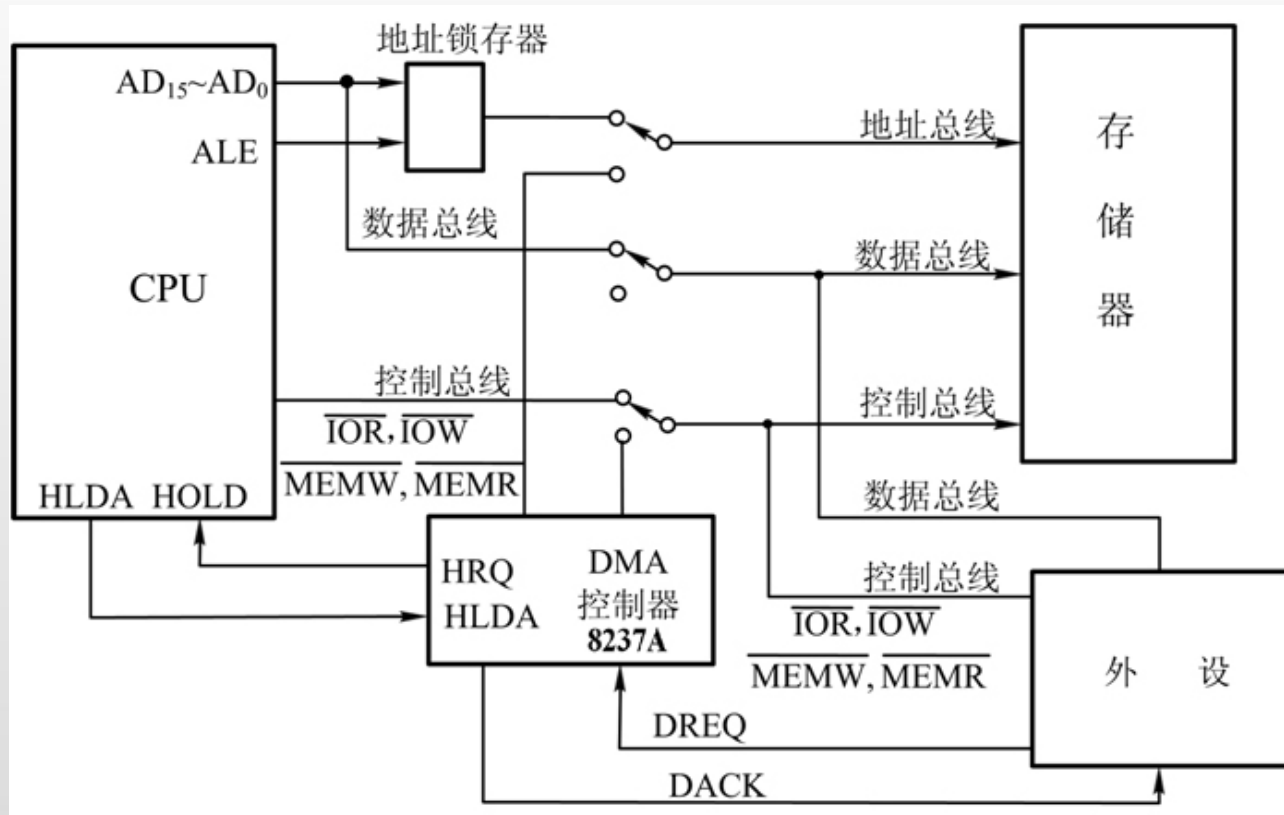
- 如图6.7，设外设为磁盘驱动器，在DMA控制器8237A控制下，可实现它与内存间的DMA数据传输。CPU与8237A分时使用地址、数据和控制总线。图中3个开关打在上方表示CPU控制总线，下方为8237A控制系统总线。



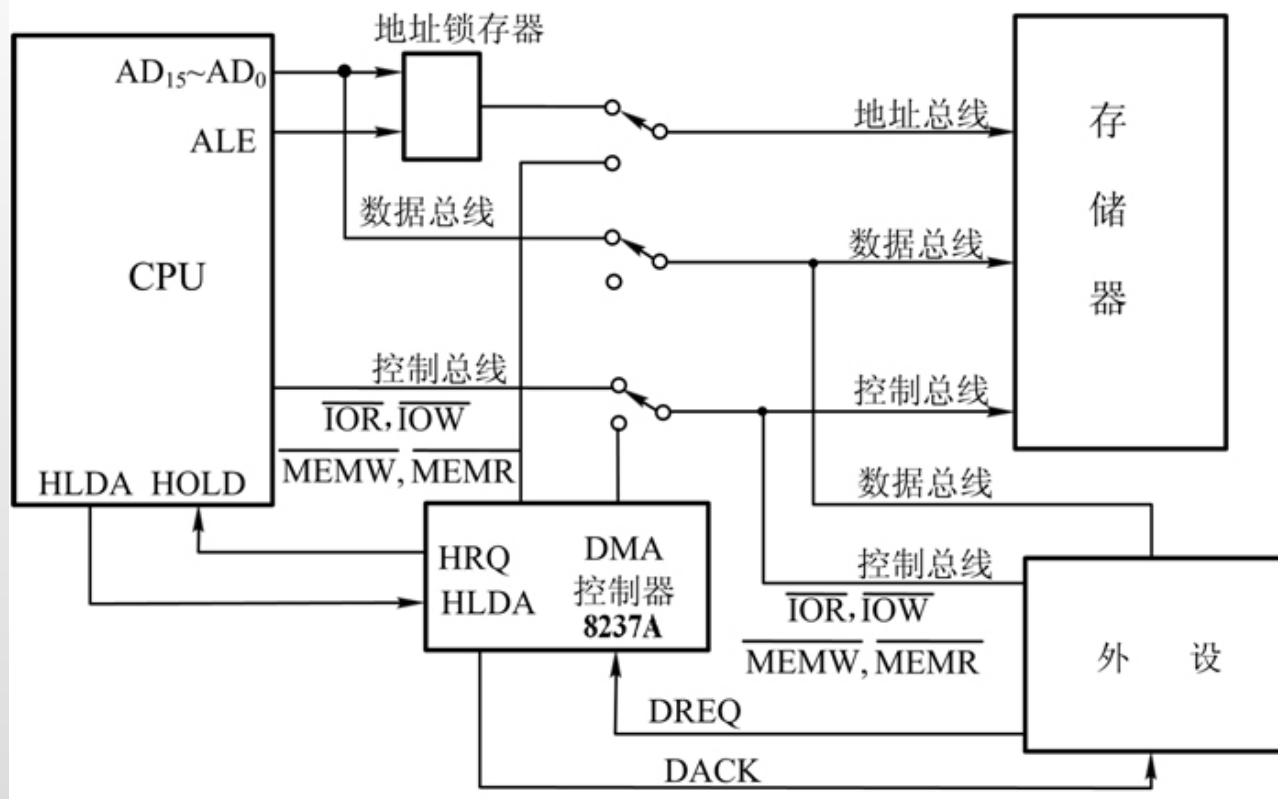
用DMA方式读磁盘的过程

- (1) 系统启动时开关打向上方，总线由CPU控制。DMA传输前，对DMA控制器初始化编程，设定要传送的字节数和内存块起始地址，选定DMA通道和所用的传送方式等。
- (2) CPU向磁盘控制器发读盘命令，由磁盘控制器找到要读取的数据位置，并开始读出数据。





(3) 磁盘准备好1个字节后，就向8237A发DMA请求信号DREQ。如通道没被屏蔽，8237A就送保持请求信号HRQ到CPU的HOLD脚。CPU完成当前的总线操作后，就中止程序的运行，将总线浮空，并发回保持响应信号HLDA给8237A；8237A送出控制信号，使3个开关打向下方，让总线和8237A相连，而与CPU脱开，接管了总线。



(4) 8237A获得总线控制权后，便通过地址总线向存储器发送要写入的第一个数据的地址信号。随后又向磁盘发DMA确认信号DACK，通知磁盘准备好要输出的字节。

(5) 8237A使控制总线上的I/O读信号和存储器写信号有效。使磁盘能向数据总线输出数据，使所寻址的存储单元能接受从数据总线上写入的数据。

(6) 完成1字节传送后，DMA控制器自动修改地址，指向下个字节，并将字计数器减1，之后再重复上述传送过程。这批数据传送完毕，DMA过程结束。

(7) 传送结束后，DMA控制器撤销保持信号HRQ，并释放总线。3个开关又拨回上方，总线与CPU相连，CPU恢复对它的控制权，继续执行后续程序。

► 每个时钟周期结束后，CPU都检测HOLD脚，看是否有DMA请求。若有，便中止正在执行的程序，进入DMA周期。



§6.1 I/O接口

6.1.1 I/O接口的功能

6.1.2 I/O端口及其寻址方式

6.1.3 CPU与外设间的数据传送方式

6.1.4 PC机的I/O地址分配



6.1.4 PC机的I/O地址分配

1. PC/XT机的I/O端口分配

- PC机中，中断控制、DMA控制、动态RAM刷新、系统配置识别、键盘代码读取及扬声器发声等都是由可编程I/O接口芯片控制的。
- 接口芯片包括：中断控制器8259A、DMA控制器8237A-5、并行接口芯片8255A-5、计数器/定时器8253-5等，都要使用I/O端口地址。
- 除键盘、显示器、打印机、磁盘驱动器等常规外设，在系统板上还有8个I/O扩展槽，可在槽中插入I/O适配器，即扩展卡，提供数据采集卡、通信卡等其它外设接口，它们也需用I/O端口地址。系统统一安排这些端口的地址。

- PC/XT机系统中，使用低10位地址($A_9 \sim A_0$)寻址I/O端口，I/O地址空间占1K。当 $A_9=0$ 时，寻址系统板上的512个端口；当 $A_9=1$ 时，寻址I/O通道上的512个端口。
- 系统板和I/O通道上的I/O端口地址分配见表6.1。表中，系统板地址含两部分，前面是译码电路生成的地址，而括号()中是I/O接口芯片实际使用的地址。



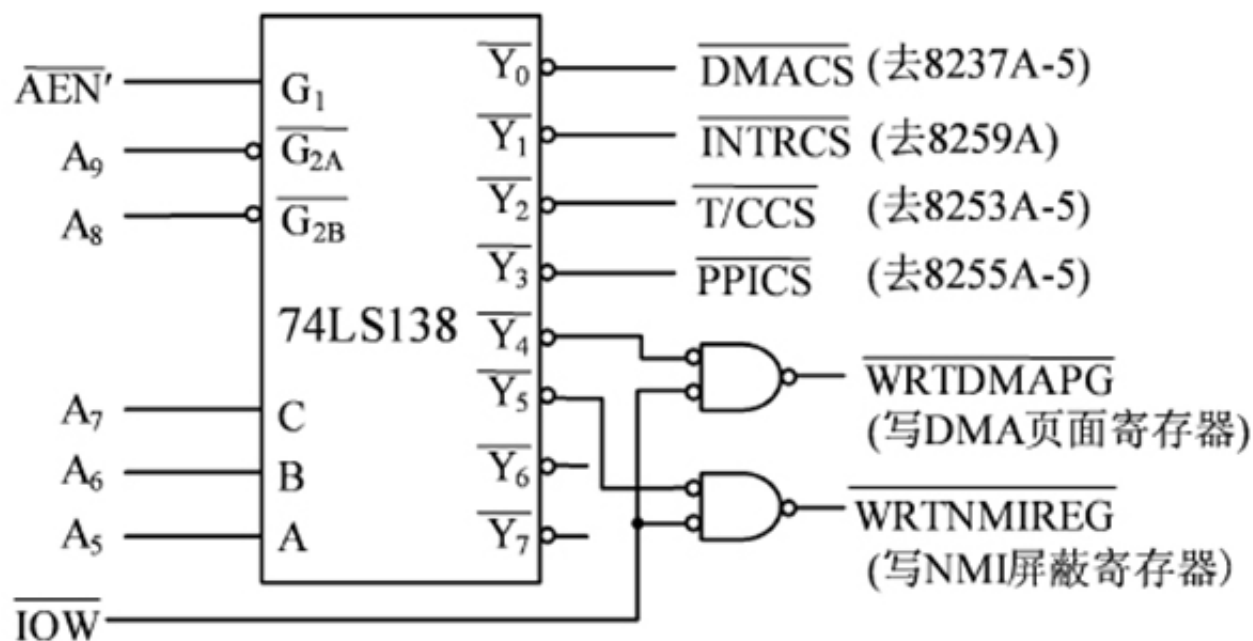
表 6.1 PC/XT 机的 I/O 端口分配表

分类	地址范围(H)	I/O 设备(端口)
系 统 板	000~01F(00~0F) 020~03F(20~21) 040~05F(40~43) 060~07F(60~63) 080~09F(80~83) 0A0~0BF(A0) 0C0~0DF 0E0~0FF	8237A-5 DMA 控制器 8259A 中断控制器 8253-5 计数器/定时器 8255A-5 并行接口 DMA 页寄存器 NMI 屏蔽寄存器 保留 保留
I/O 通 道	200~20F 2F8~2FF 300~31F 320~32F 378~37F 380~38F 3B0~3BF 3F0~3F7 3F8~3FF	游戏 I/O 异步通信 2(COM 2) 实验卡(原型卡) 硬磁盘适配器 并行打印机接口 同步通信控制器 单显/打印机适配器 软磁盘适配器 异步通信 1(COM 1)



● 系统板上I/O端口译码电路

- 各接口芯片的片选信号由74LS138译码电路产生，在CPU控制系统总线时， $\overline{AEN}'=1$ ，这时若 $A_9A_8=00$ ，则译码器选通，对输入 $A_7A_6A_5$ 译码，在 $\overline{Y}_0 \sim \overline{Y}_7$ 中产生一个低电平输出信号，接到相应接口芯片的 **CS** 端或控制端。



- 电路在I/O读写命令控制下工作。I/O地址的低4位 $A_3 \sim A_0$ 用作控制芯片内部寄存器的选择信号，这样每个译码输出端都包含 $2^4=16$ 个端口地址。
- 有些接口芯片内部有16个寄存器，例如8237A DMA控制器，使用00~0FH共16个端口地址。
- 多数接口芯片内部没有16个寄存器，如8259A只有2个寄存器，8253-5和8255A各有4个寄存器，较高位地址可以不用，仅用 A_1 和 A_0 等低位地址选择端口。
- 8259A占用的端口地址范围为020H~03FH，但实际只用到最低2个端口地址20H和21H，而NMI屏蔽寄存器仅用了1个I/O地址A0H。



表 6.2 PC/AT 及兼容机的 I/O 端口地址分配表

分类	地址范围(H)	I/O 设备(端口)
系 统 板	000~01F	DMA 控制器 1,8237A-5
	020~03F	中断控制器 1,8259A(主片)
	040~05F	定时器,8254-2
	060~06F	键盘接口处理器,8042
	070~07F	实时时钟,NMI 屏蔽寄存器
	080~09F	DMA 页寄存器,74LS612
	0A0~0BF	中断控制器 2,8259A(从片)
	0C0~0DF	DMA 控制器 2,8237A-5
	0F0	清除协处理器忙信号
	0F1	复位协处理器
	0F8~0FF	协处理器
I/O 通 道	1F0~1F8	硬磁盘
	200~207	游戏 I/O 口
	278~27F	并行口 2(LPT2)
	2F8~2FF	串行口 2(COM2)
	300~31F	实验卡(原型卡)
	360~36F	保留
	378~37F	并行打印机口 1(LPT1)
	380~38F	SDLC,双同步通信口 2
	3A0~3AF	双同步通信口 1
	3B0~3BF	单色显示器/打印机适配器
	3C0~3CF	保留
	3D0~3DF	彩色/图形监视器适配器
	3F0~3F7	软磁盘控制器
	3F8~3FF	串行口 1(COM1)

2. PC/AT机端口地址

●以80286为CPU的PC/AT机中，也只使用低10位地址进行I/O端口地址的译码。

●地址范围000~3FFH, 但使用两片8237A-5 DMA 控制器, 两片8259A中断控制器, 定时器使用8254-2。

●PC/AT及其兼容机的I/O端口地址分配如表6.2。

