

第五章 输入输出及其接口

本章学习要求：

了解输入输出接口的基本结构，掌握计算机与外设之间进行数据传送的3种基本输入输出方法及对输入输出接口的要求。

5.1 为什么要用接口电路

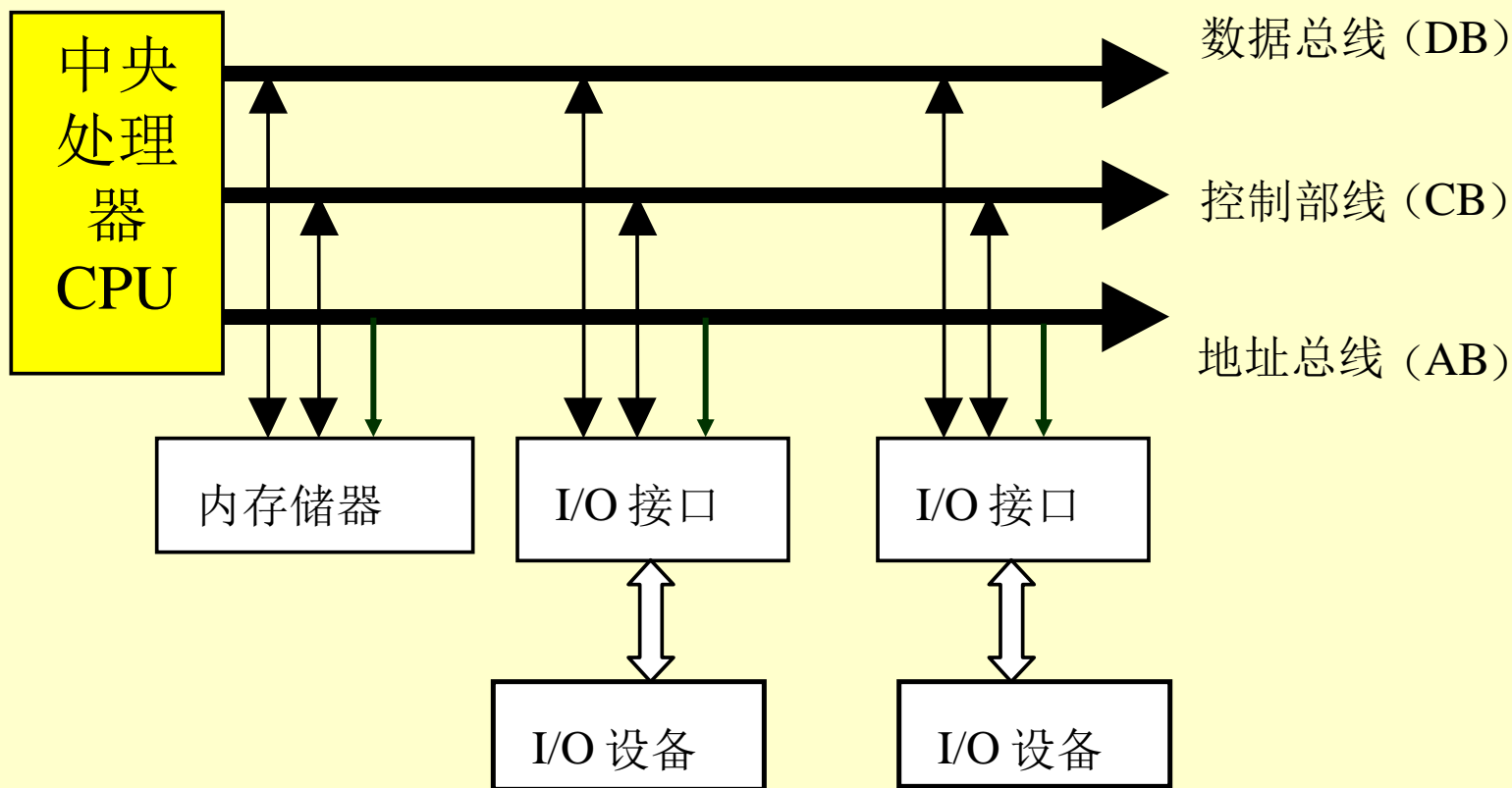
一、CPU与外设间的连接

- 接口概念

计算机通过外围设备同外部世界通信或交换数据称为“输入 / 输出”。

把外围设备同微型计算机连接起来实现数据传送的控制电路称为“外设接口电路”，简称“**外设接口**”。(介于主机和外设之间的一种缓冲电路称为I/O接口电路)。

CPU 与外设的连接不能向 CPU 同存储器的连接那样，直接挂在总线上，而必须通过**输入/输出接口**电路来连接，才能实现对外设的有效控制和管理。



二、为什么要用接口电路

原因: 存储器与外设的不同

	MEM	I/O设备
不同点	品种有限	品种繁多
	功能单一	功能多样
	传送方式单一	传送规律不同
	与CPU速度匹配	与CPU速度不匹配
	易于控制	难于控制
结论	可与CPU直接连接	需经过I/O电路与CPU连接

种类:

接口电路种类很多:

- 总线控制器（8288芯片）
- 时钟发生器（8284A芯片）
- 总线裁决器（8289芯片）
- 通用接口电路（如8251、8255、8253、8237、8259、74LS244/245、74LS273、74LS374/373等）
- 专用接口（如电平转换接口、A/D和D/A转换器等）

接口芯片有两类:一是可编程芯片，二是简单芯片。

二、为什么要用接口电路

■ 接口按功能分为两类

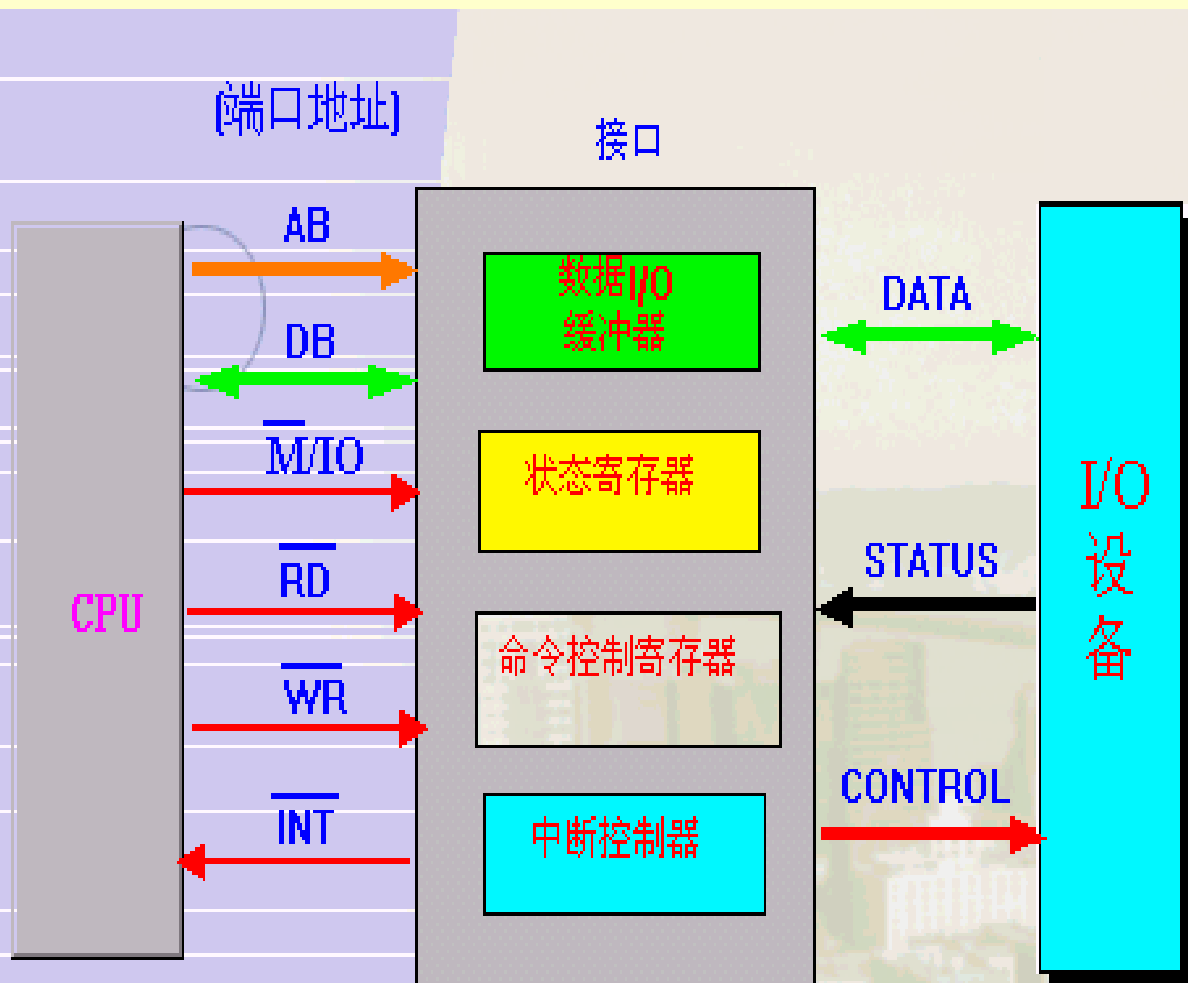
- 使CPU正常工作所需要的辅助电路
- 输入/输出接口

■ 外部设备为什么一定要通过接口和主机相连？

- 时序上的原因
- 外部设备功能多种多样的原因
- 外部设备的信息既有数字式，又有模拟式的原因
- 多个外设共享总线的原因
- 速度的原因

I/O接口电路作用：起隔离、锁存、记忆、变换作用

5.2 CPU和输入/输出设备之间的接口信息



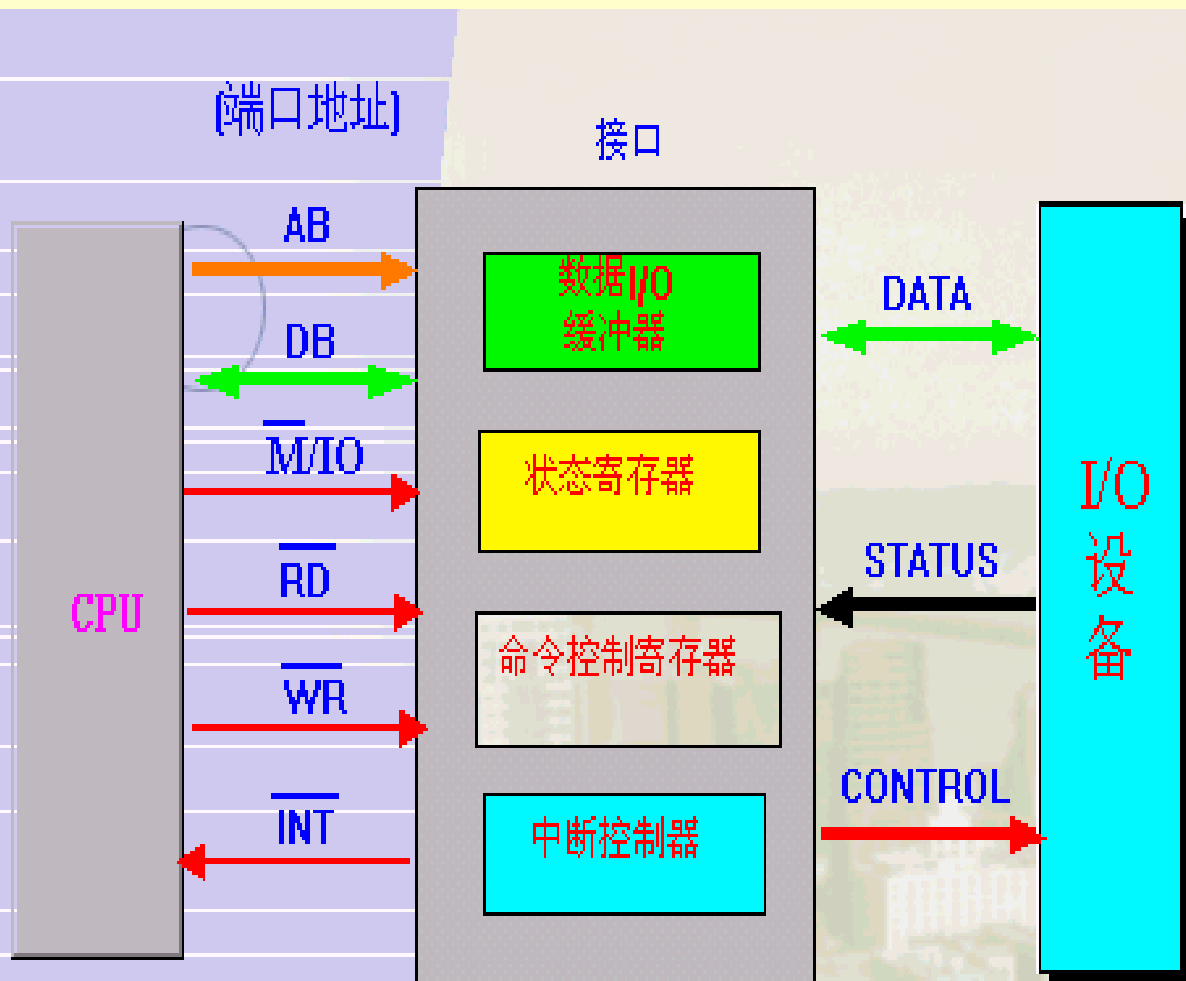
接口电路基本结构及其连接

接口电路的基本结构同它传送信息种类有关。接口与外设之间的信号可分为三类：

- ① 数据信息
- ② 状态信息
- ③ 控制信息

.... 10100100101

5.2 CPU和输入/输出设备之间的信号



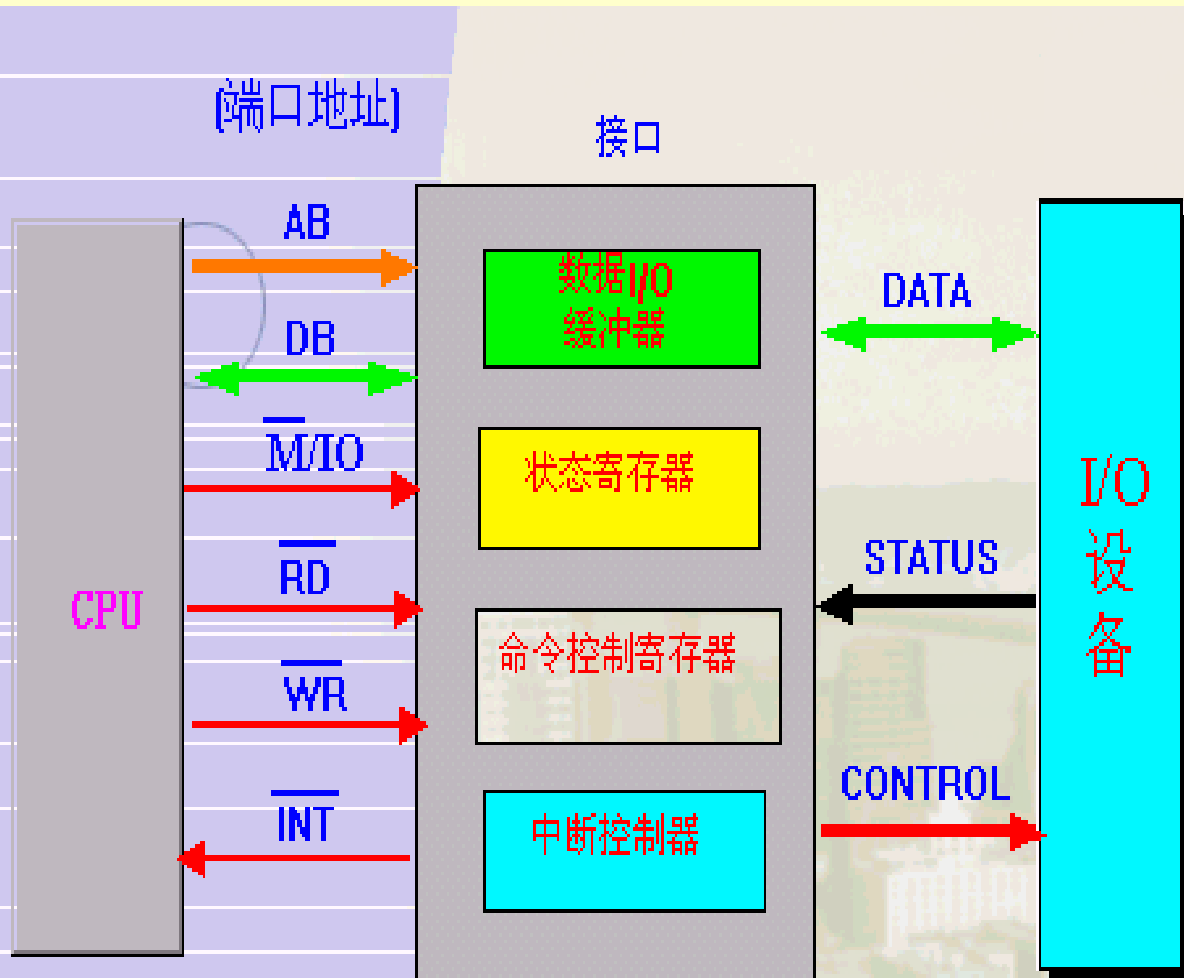
接口电路基本结构及其连接

数据信息

这是最基本的一种信息。它包括：

1. 数字量：通常为8位二进制数或ASCII代码。
2. 模拟量：当计算机用于检测、数据采集或控制时，大量的现场信息是连续变化的物理量（如温度、压力、流量、位移、速度等），经传感器把非电量转换成电量并经放大即得到模拟电流或电压，这些模拟量，计算机不能直接接收和处理，必须经过A/D（模/数）转换，才能输入计算机；而计算机输出的数字量也必须经D/A（数/模）转换后才能去控制执行机构。
3. 开关量：是一些“0”或“1”两个状态的量，用一位“0”或“1”二进制数表示。一台字长为8位的微机一次输入或输出可控制8个这类物理量。数据是通过数据通道传送的。

5.2 CPU和输入/输出设备之间的信号



接口电路基本结构及其连接

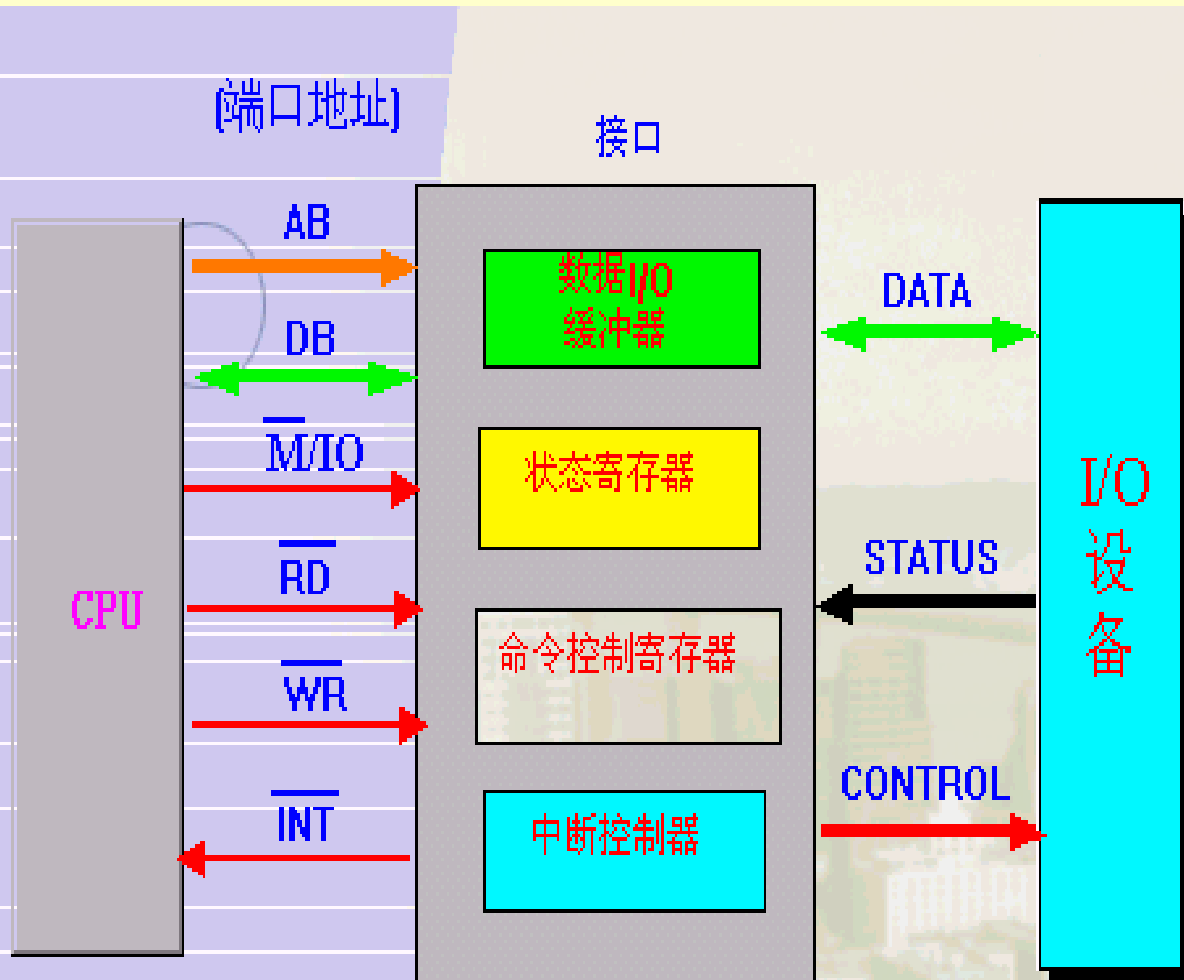
状态信息

状态信息是反映外设当前所处工作状态的信息，以作为CPU与外设间可靠交换数据的条件。

当输入时，它告知CPU：有关输入设备的数据是否准备好 (Ready=1?)；输出时，它告知CPU：输出设备是否空闲 (Busy=0?)。

CPU是通过接口电路来掌握输入输出设备的状态，以决定可否输入或输出数据。

5.2 CPU和输入/输出设备之间的信号



接口电路基本结构及其连接

控制信息

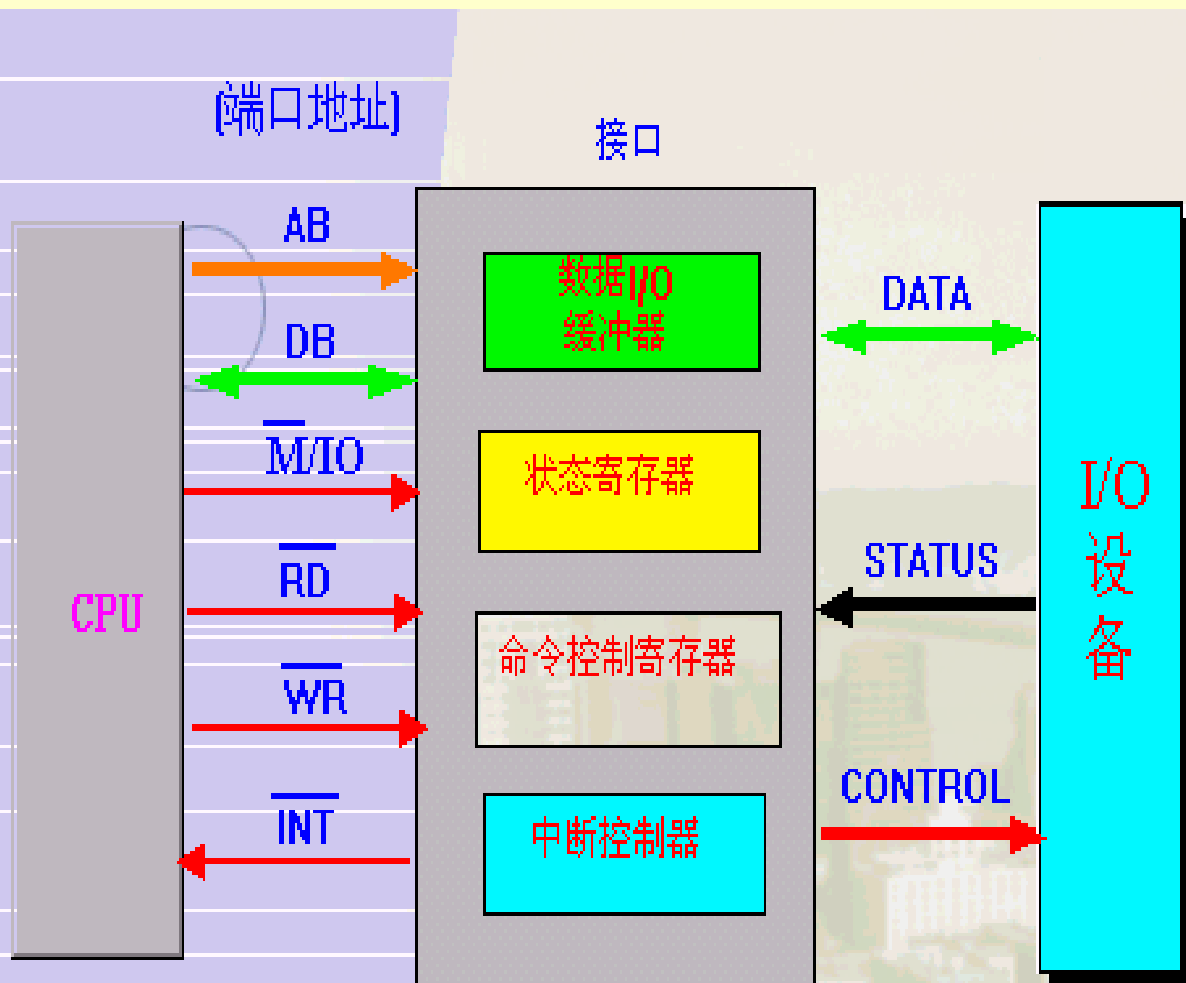
它用于控制外设的启动或停止。接口电路根据传送不同信息的需要，其基本结构安排有一些特点。

1. 三种信息（数据、状态、控制）的性质不同，应通过不同的端口分别传送。如数据输入/输出寄存器（缓冲器）、状态寄存器与命令控制寄存器各占一个端口，每个端口都有自己的端口地址，故能用不同的端口地址来区分不同性质的信息。

2. 在用输入输出指令来寻址外设（实际寻址端口）的CPU中，外设的状态作为一种输入数据，而CPU的控制命令，是作为一种输出数据，从而可通过数据总线来分别传送。

3. 端口地址由CPU地址总线的低8位地址信息来确定，CPU根据I/O指令提供的端口地址来寻址端口，然后同外设交换信息。

5.3 接口部件的I/O端口

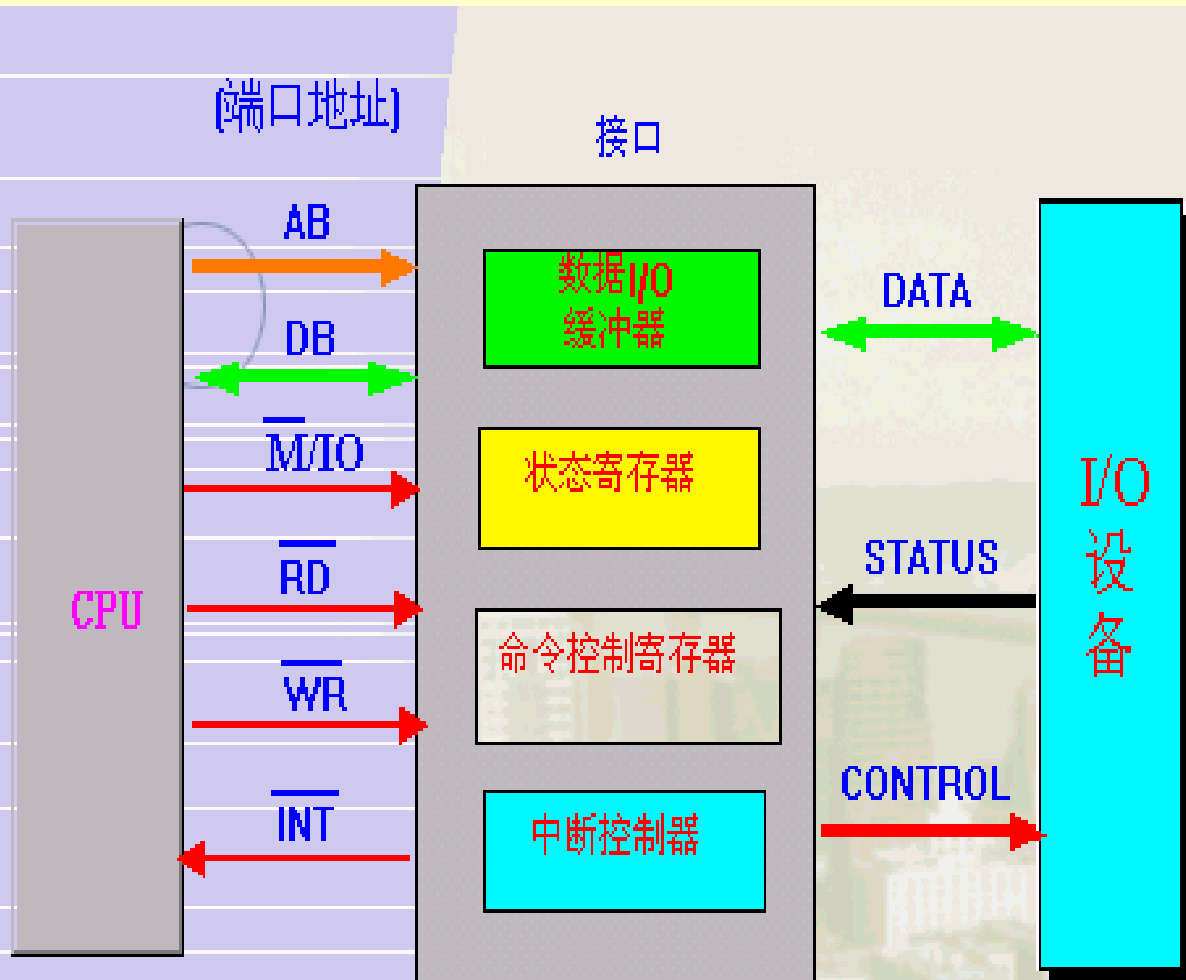


接口电路基本结构及其连接

每个接口部件都包含一组寄存器，CPU和外设进行数据传输时各类信息在接口中进入不同的寄存器，一般称这些寄存器为I/O端口，每个端口有一个端口地址，8086可寻址外设的端口地址为64K。

有了端口地址，CPU对外设的输入/输出操作归结为对接口芯片各端口的读/写操作。

5.3 接口部件的I/O端口



应该指出:

不管是输入还是输出, 所用到的地址总是对**端口**而言的, 而不是对**接口部件**而言的。

接口电路基本结构及其连接

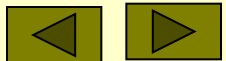
.... 10100100101

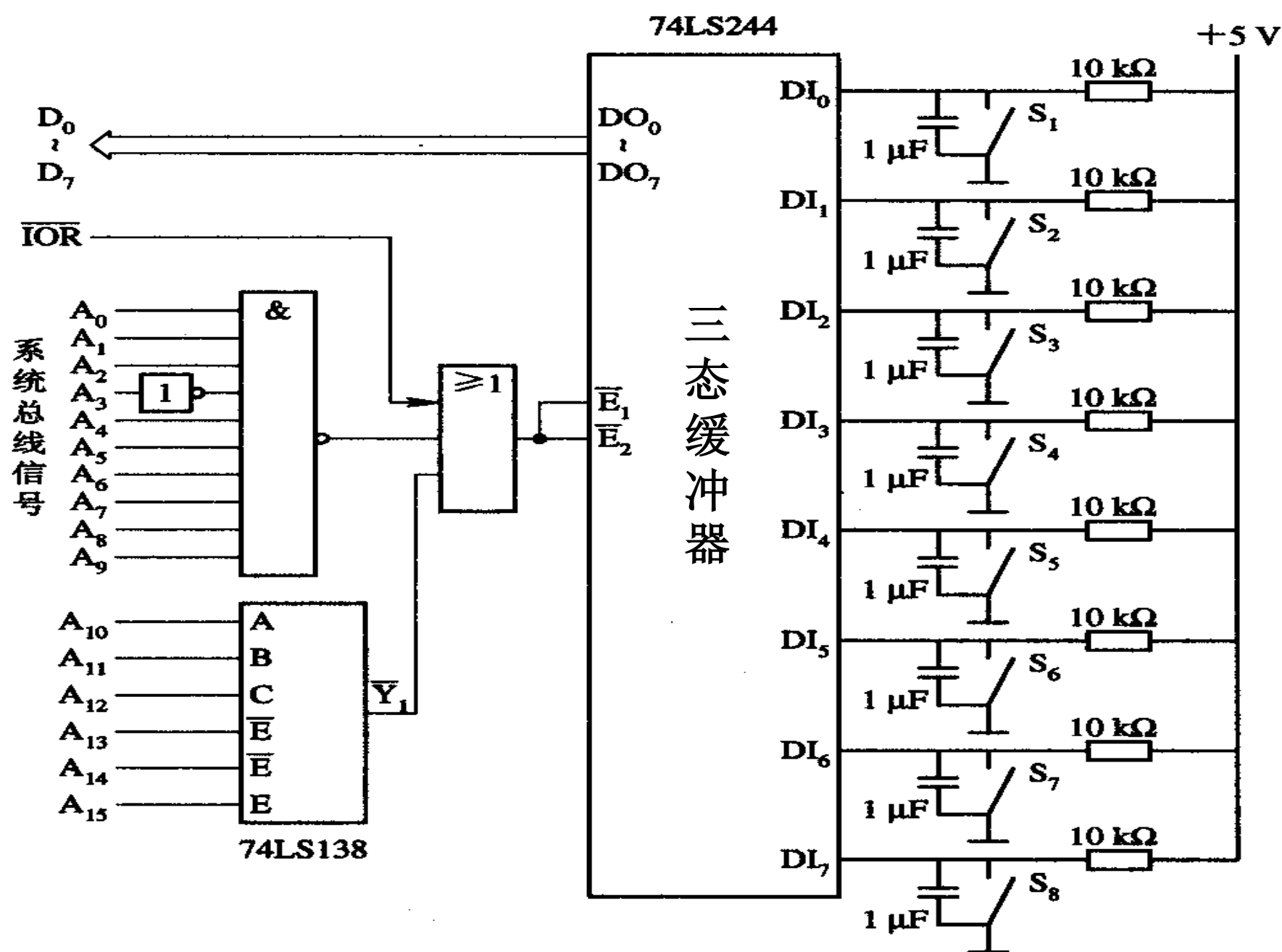
为什么单独的三态门只能做输入接口，而不能做输出接口；单独的锁存器的作用正好与其相反？

选择接口时应注意：

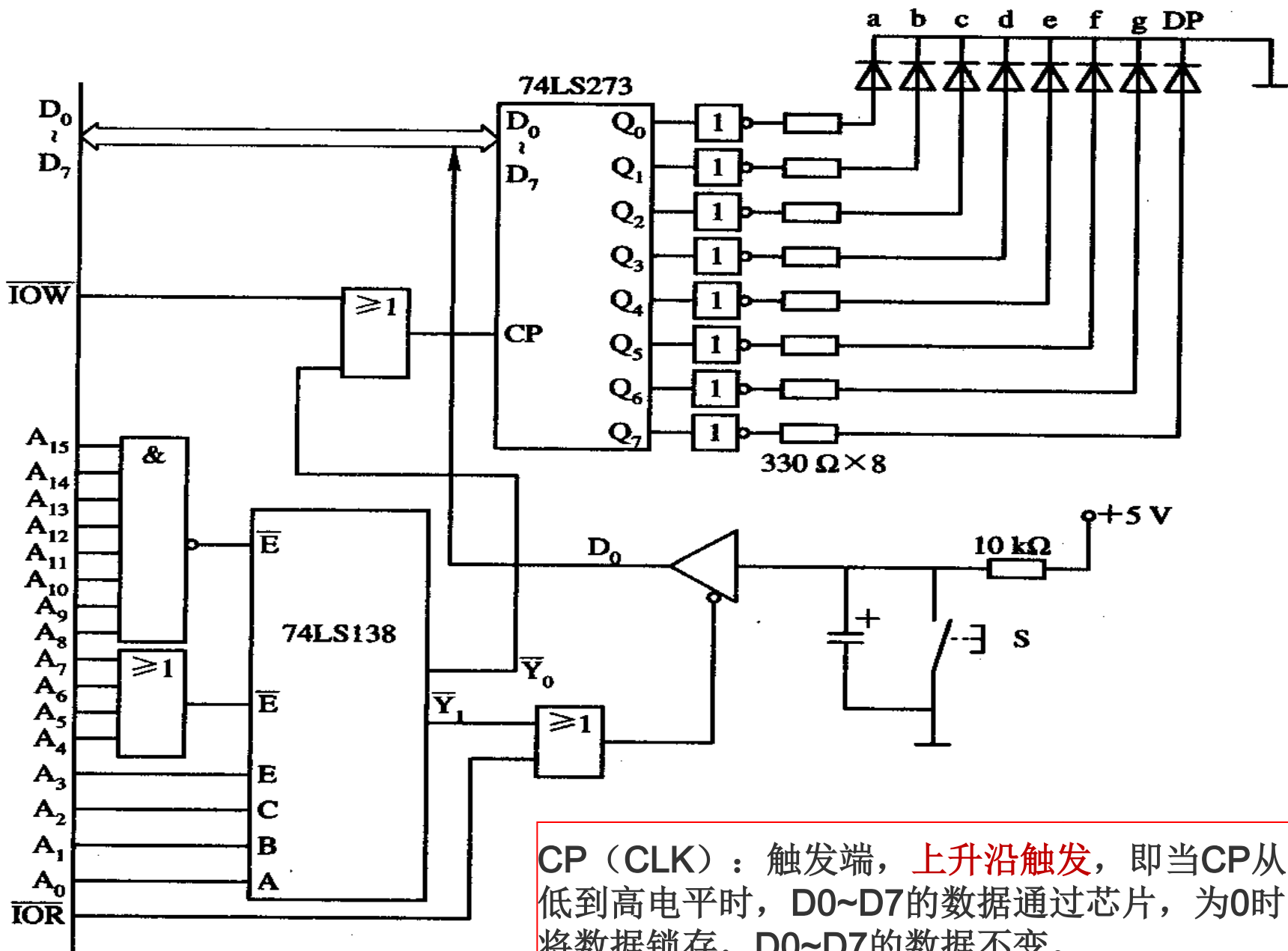
输入端口必须采用具有三态门控制的芯片：输入数据时，因简单外设输入数据的保持时间相对于CPU的接收速度来说较长，故输入数据时通常不用加锁存器来锁存，而直接使用三态缓冲器与CPU数据总线相连，起隔离作用，可避免总线竞争。如74LS244等。

输出端口必须使用锁存器：输出数据时，一般都需要锁存器将要输出的数据保持一段时间，其长短和外设的动作相适应。锁存时，在锁存器允许端CE（为无效电平）时，数据总线上的新数据不能进入锁存器。只有当确知外设已经取走CPU上次送入的数据，方能在CE=0电平时将新数据再送入锁存器保留。如：74LS373/74LS374（带三态门），74LS273等。





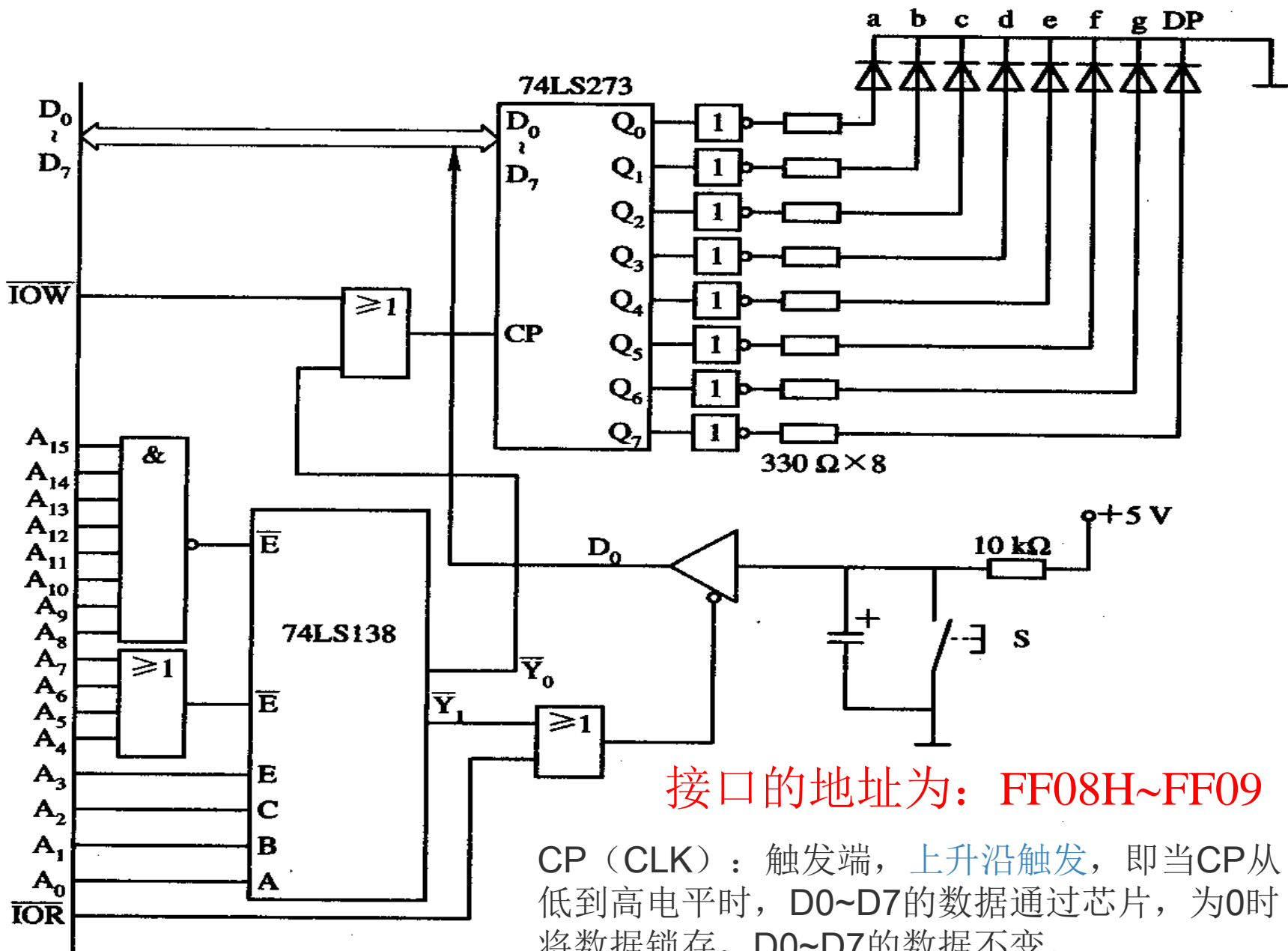
系统总线



CP (CLK)：触发端，**上升沿触发**，即当CP从低到高电平时， $D_0 \sim D_7$ 的数据通过芯片，为0时将数据锁存， $D_0 \sim D_7$ 的数据不变。

图b 开关及数码管的接口电路

系统总线



接口的地址为：FF08H~FF09

CP (CLK)：触发端，上升沿触发，即当CP从低到高电平时， $D_0 \sim D_7$ 的数据通过芯片，为0时将数据锁存， $D_0 \sim D_7$ 的数据不变。

图b 开关及数码管的接口电路

5.4 CPU和外设之间的数据传送方式

CPU与外设之间的数据传送方式，概括起来有如下三种：

- ❖ 程序传送方式
- ❖ 程序中断方式
- ❖ DMA方式

5.4.1 程序传送方式

程序传送方式是指CPU与外设间的数据交换在程序控制（即IN或OUT指令控制）下进行。

又分为：
 { 无条件传送方式
 { 条件传送方式。

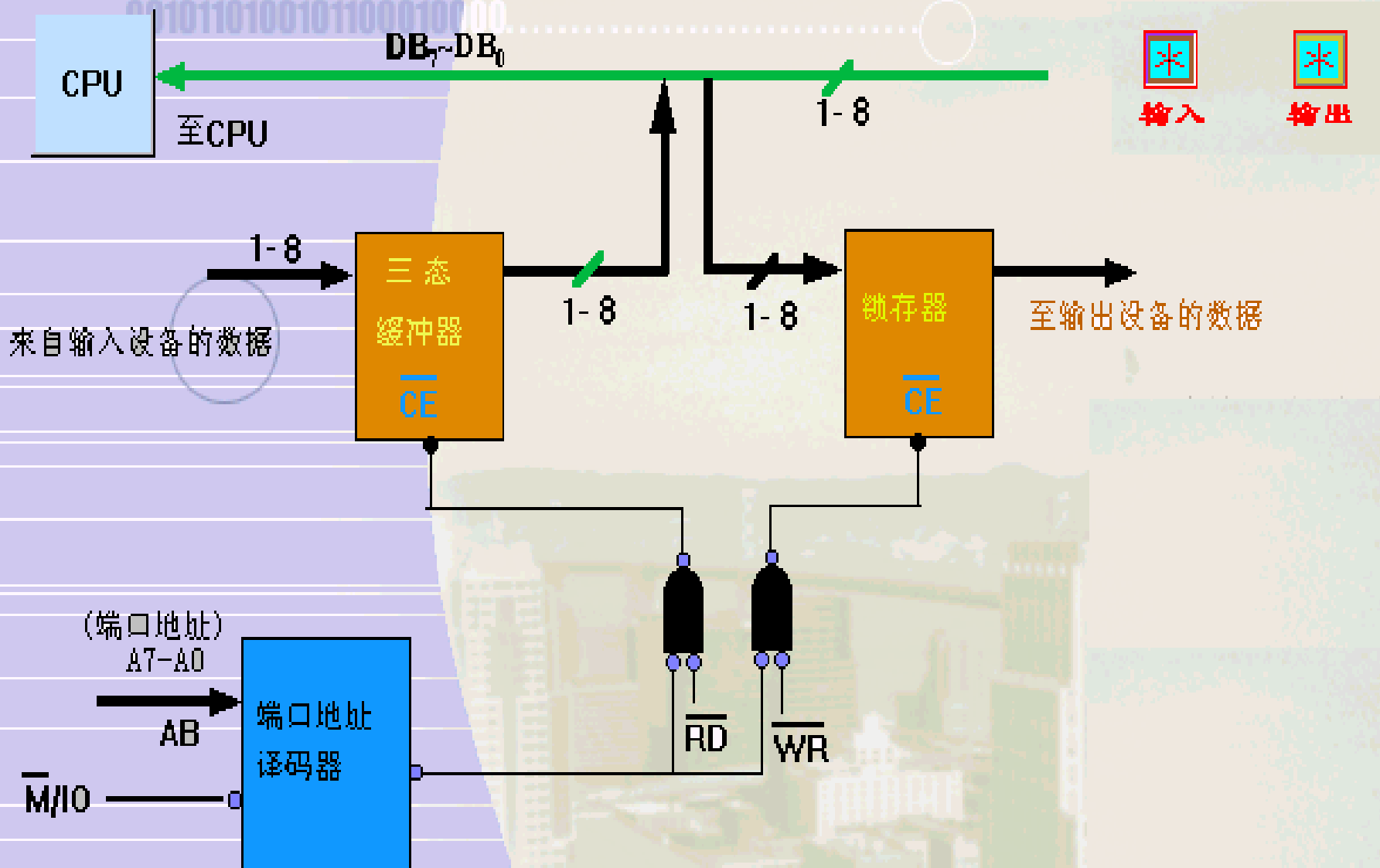
一、无条件传送方式

无条件传送（又称同步传送）是一种不需要查询外设的状态而可直接进行传送信息的一种传送方式。

这种传送方式只对固定的外设（如开关、继电器、七段显示器、机械式传感器等简单外设）在固定时间用IN 或OUT 指令来进行信息的输入或输出，其实质是用程序来定时同步传送数据。对少量数据传送来说，是最省时间的一种传送方法，适用于各类巡回检测和过程控制。

01101000010100110100001010001011010000101001011010010100

0010110100101100010000



无条件程序传送的输入输出方式

二、条件传送方式

条件传送也称为**查询方式**传送。用条件传送方式时，CPU通过执行程序不断**读取并测试外设的状态**，如果外设处于**准备好**状态（对输入设备）或**空闲**状态（输出设备）则CPU执行**输入**（IN指令）或**输出**（OUT指令）指令与外设交换信息。

可见，对于条件传送，一个数据的传送过程由3个环节组成：即

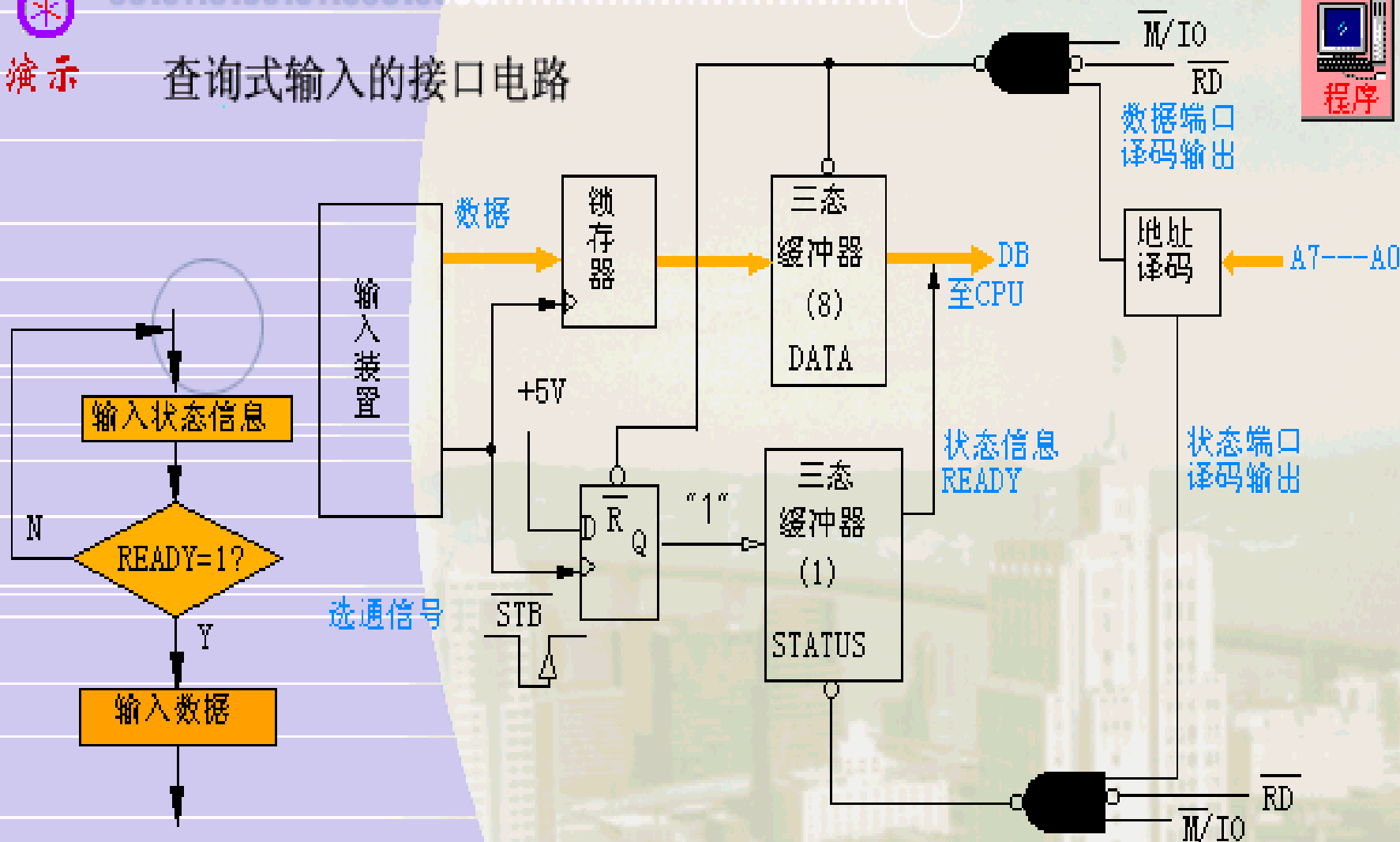
- ① CPU从**接口**中读取**状态字**。
- ② CPU检测状态字的对应位是否满足“**就绪**”条件，如不满足转①
- ③ 如表明外设**已处于“就绪”**状态，则**传送数据**。

1. 查询方式输入



演示

查询式输入的接口电路



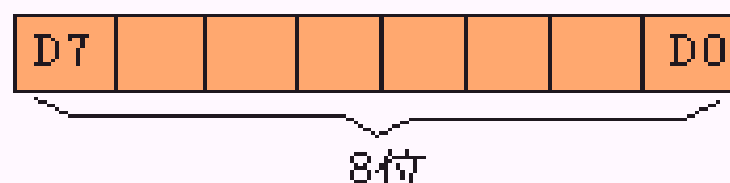
1. 查询方式输入

查询输入方式的程序流程图

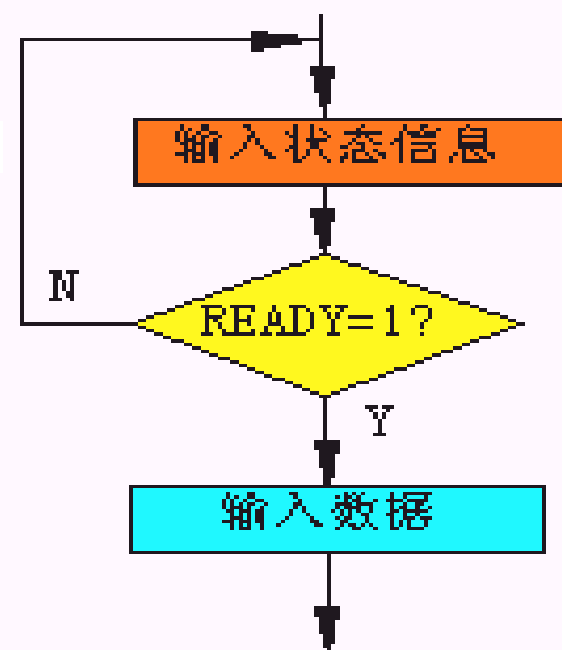
这种查询输入方式的程序流程图如图右所示。

读入的数据是8位，而读入的状态信息往往是一位的，所以，不同的外设其状态信息可以使用同一个端口，但只要使用不同的位就行。如下图所示

数据(Data)端口(8位)
(输入)



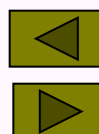
状态(Status)端口(1位)
(输入)



查询输入部分的程序：

```
POLL:  IN  AL , STATUS_PORT ;读状态端口的信息
        TEST AL , 80H       ;设“准备就绪”(READY)信息在D7位
        JE  POLL            ;未“准备就绪”，则循环再查
        IN  AL , DATA_PORT ;已“准备就绪”(READY=1)，则读入数据
```

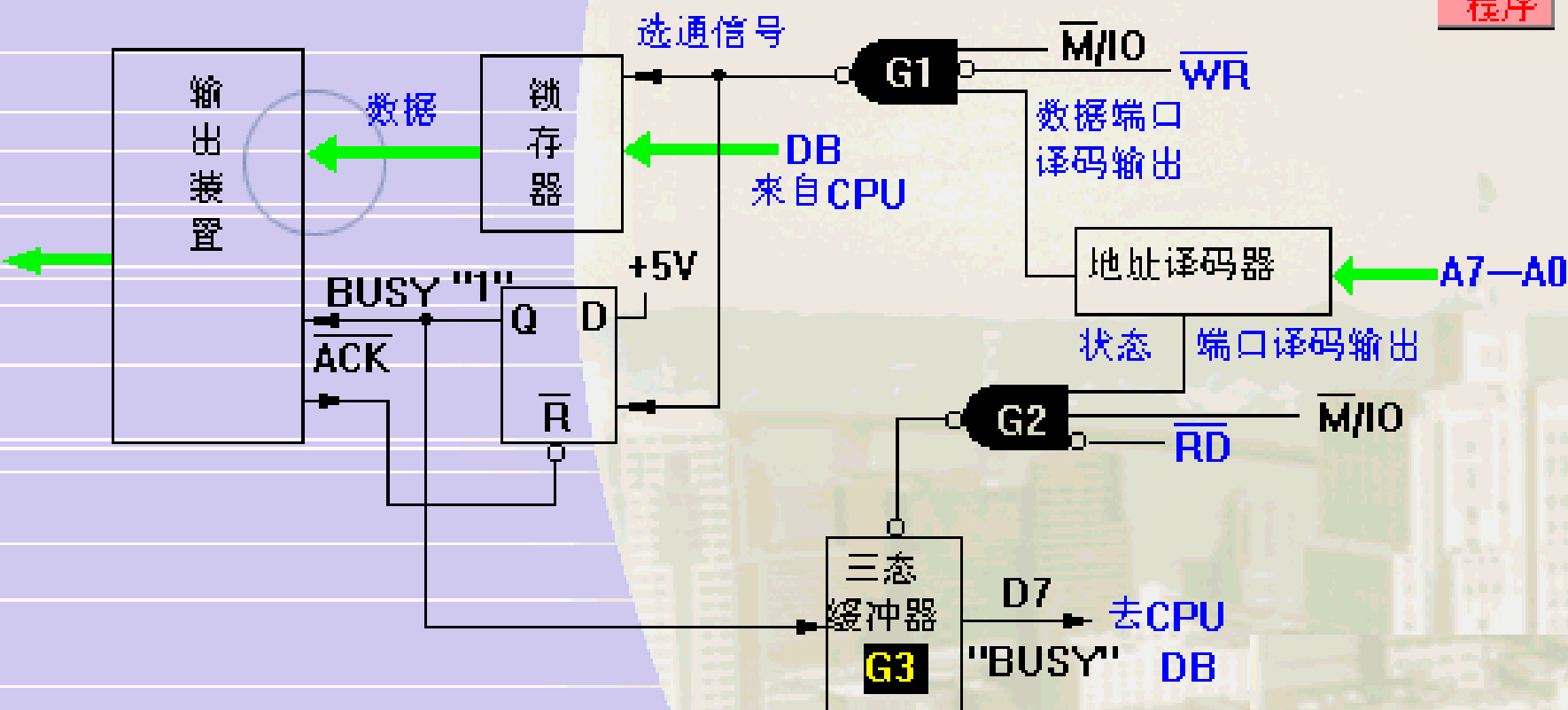
这种CPU与外设的状态信息的交换方式，称为应答式，状态信息称为“联络”(hand-shake)信息。



2. 查询方式输出



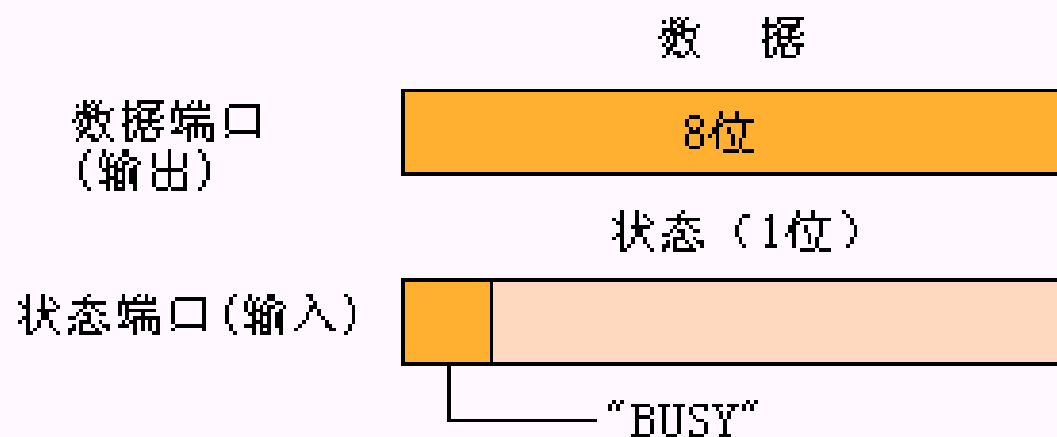
演示



查询式输出的接口电路

2. 查询方式输出

查询式输出的端口信息

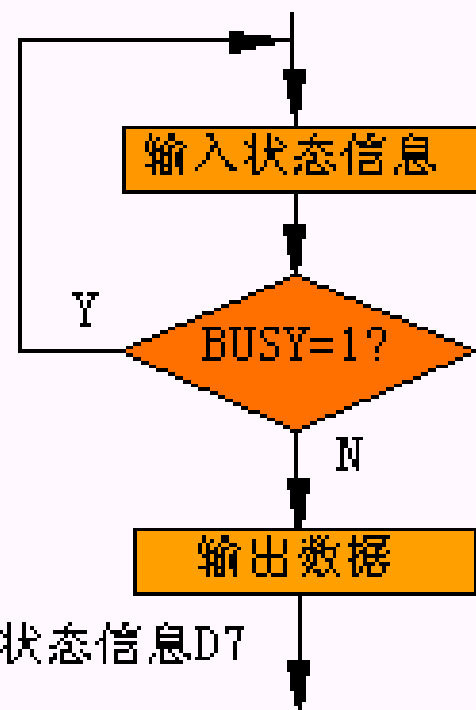


查询输出部分的程序:

```
POLL:  IN  AL , STATUS_PORT    ;查状态端口中的状态信息D7
      TEST AL , 80H
      JNE POLL                ;D7=1即忙线=1, 则循环再查
      MOV  AL , STORE          ;否则, 外设空闲, 则由内存读取数据
      OUT  DATA_PORT , AL     ;输出到DATA地址端口单元
```

其中, STATUS和DATA分别为状态端口和数据端口的符号地址; STORE为要输出的数据的内存单元的符号地址。

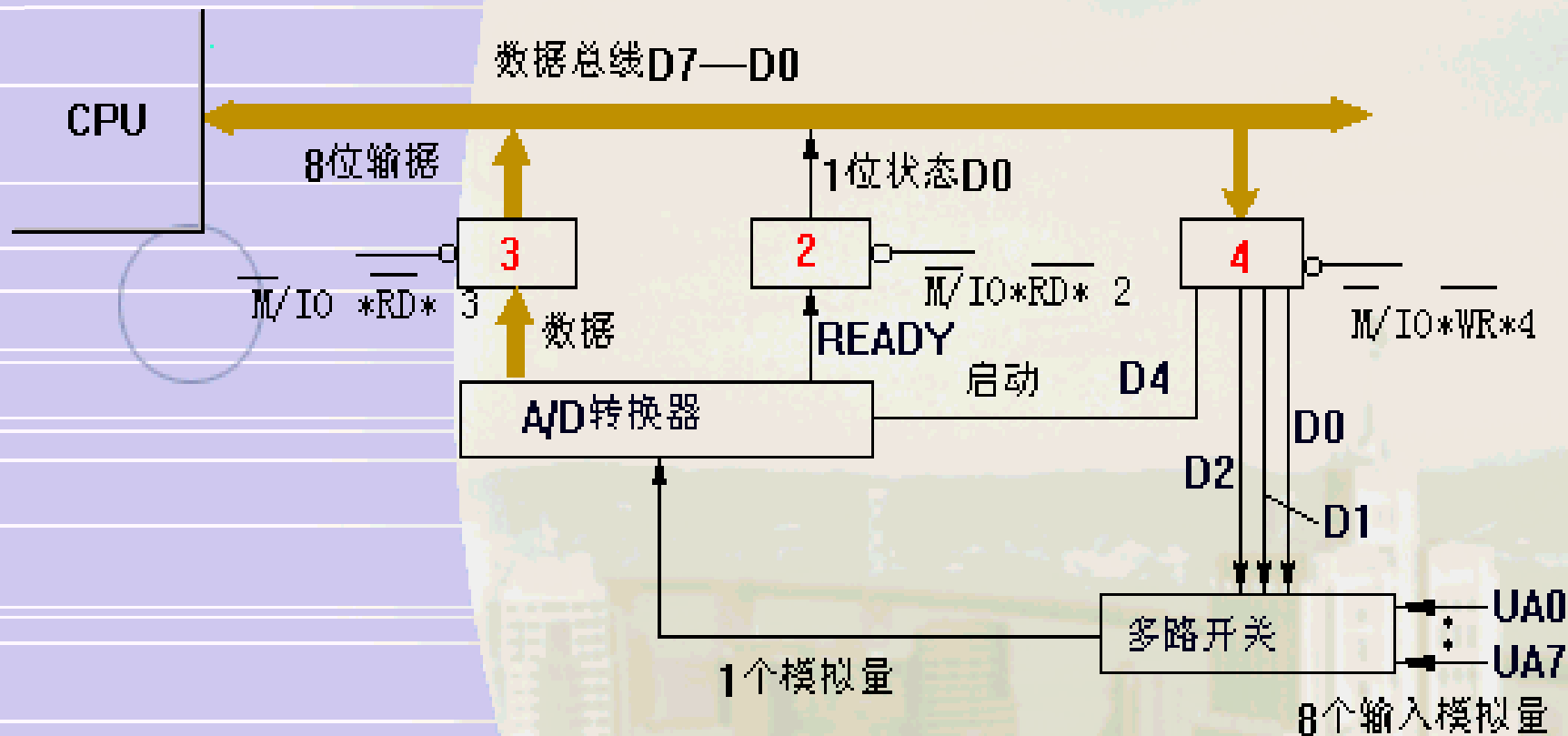
查询输出方式的程序流程图



3. 查询方式应用举例

[例1]

查询式数据采集系统

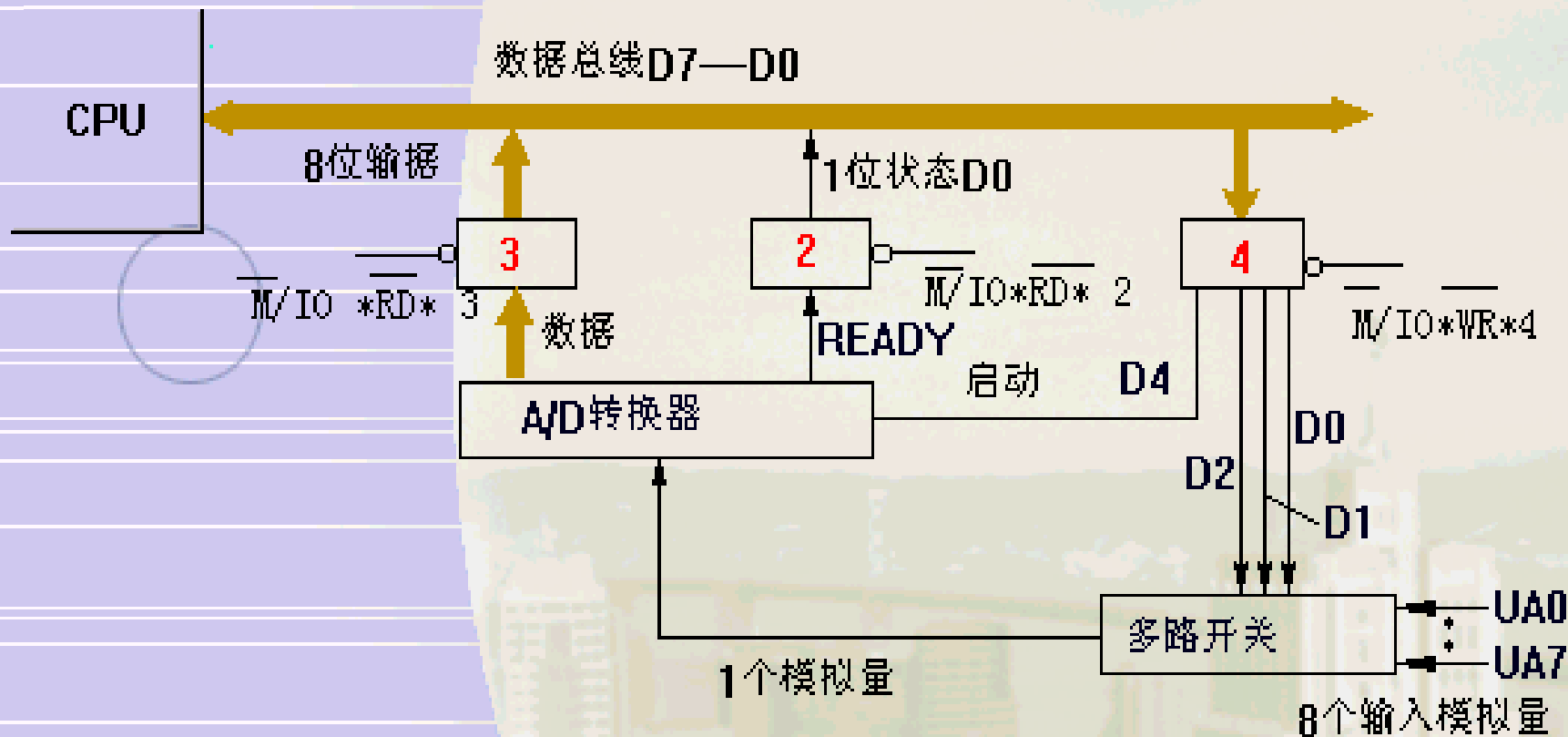


8个输入模拟量，经过多路开关——它由端口4输出的3位二进制码(D2、D1、D0)控制(000—相应于UA0输入、...、111—相应于UA7输入)，每次送出一个模拟量至A/D转换器；同时A/D

3. 查询方式应用举例

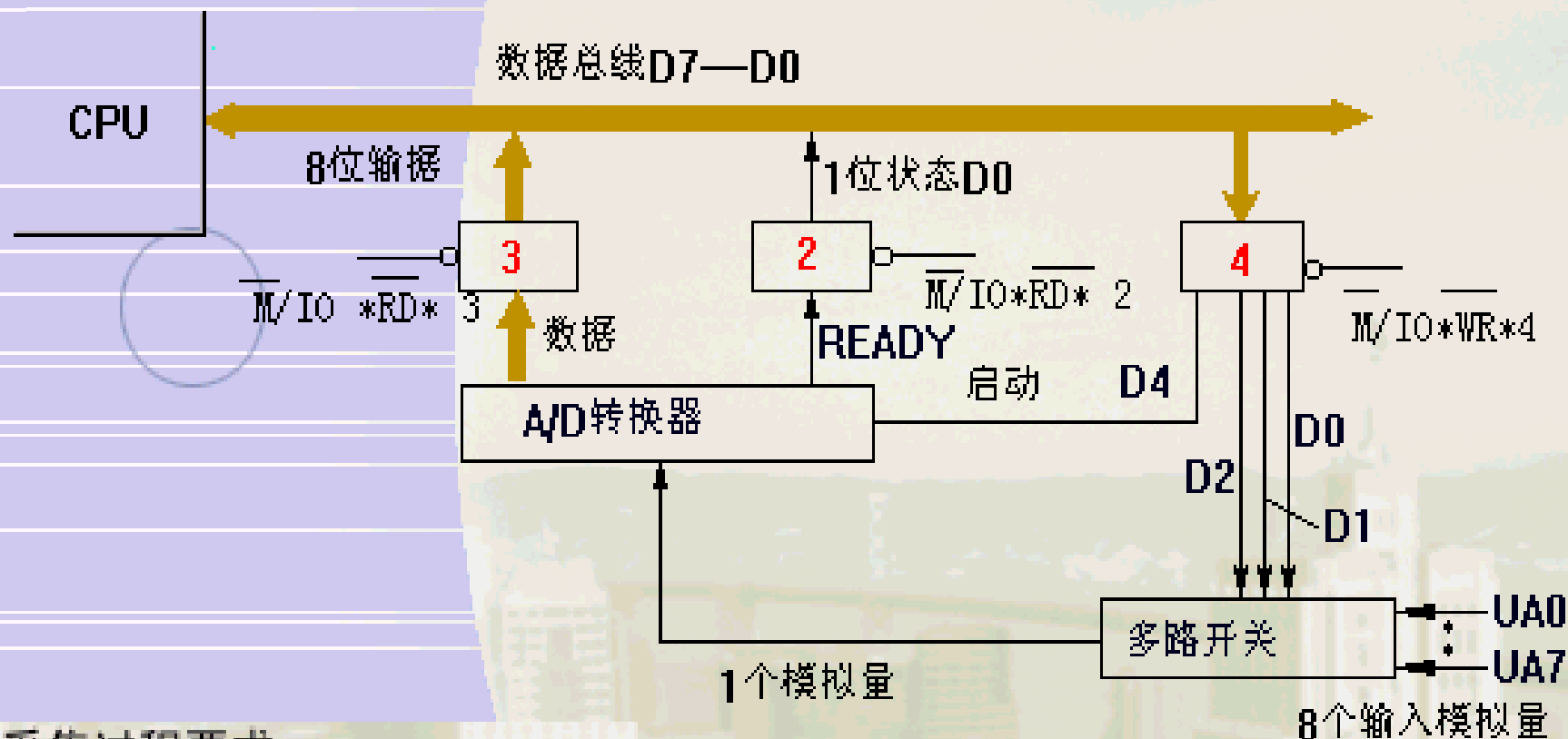
00101101001011000100100.....

查询式数据采集系统



转换器由端口4输出的D4位控制启动和停止。A/D转换器的READY信号由端口2的D0输至CPU数据总线；经A/D转换后的数据由端口3输入至数据总线。

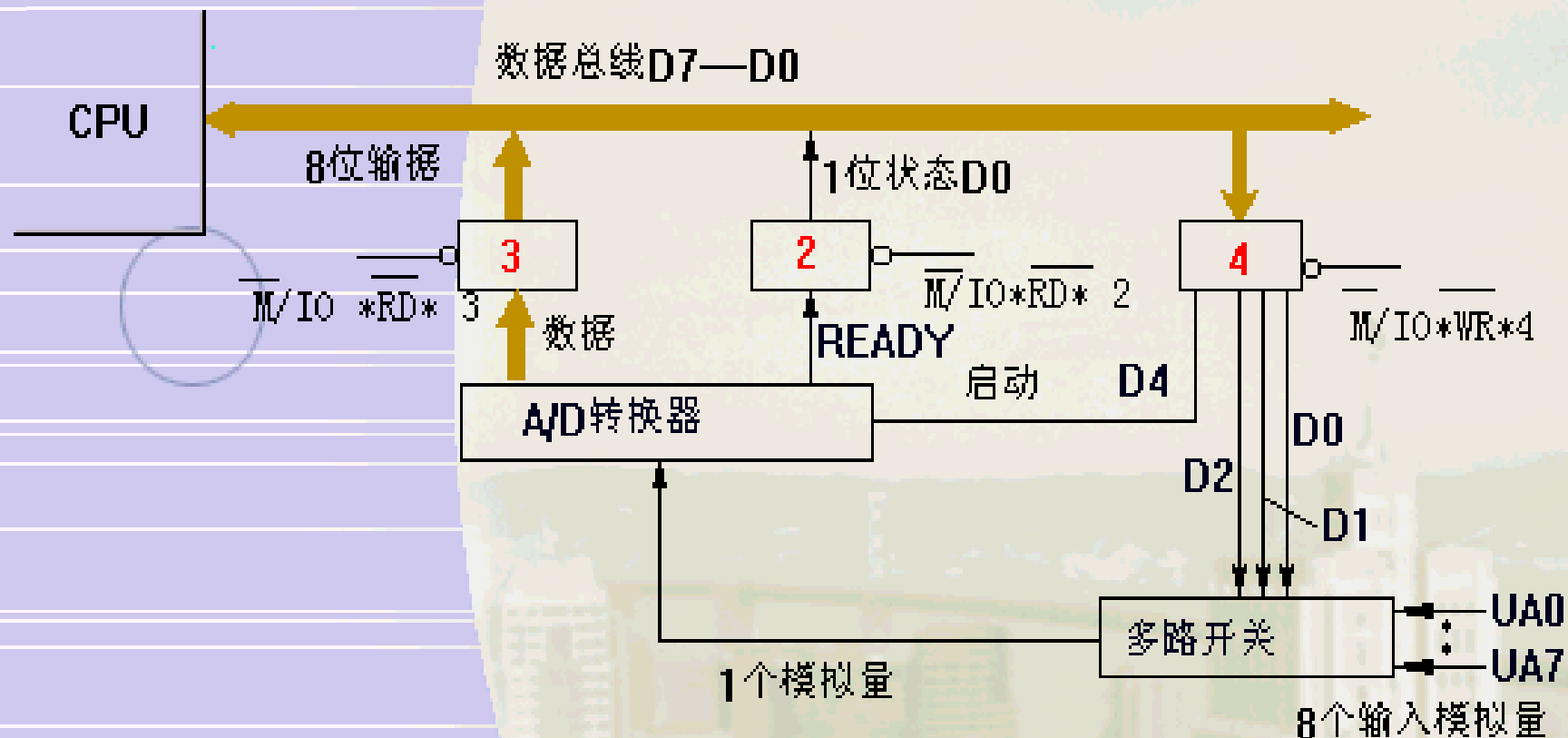
查询式数据采集系统



采集过程要求:

- (1) 初始化。
- (2) 先停止A / D转换。
- (3) 启动A / D转换，查输入状态信息READY。
- (4) 当输入数据已转换完 (READY = 1，即准备就绪)，则经由端口3输入至CPU的累加器A中，并转送内存。
- (5) 设置下一个内存单元与下一个输入通道，循环8次。

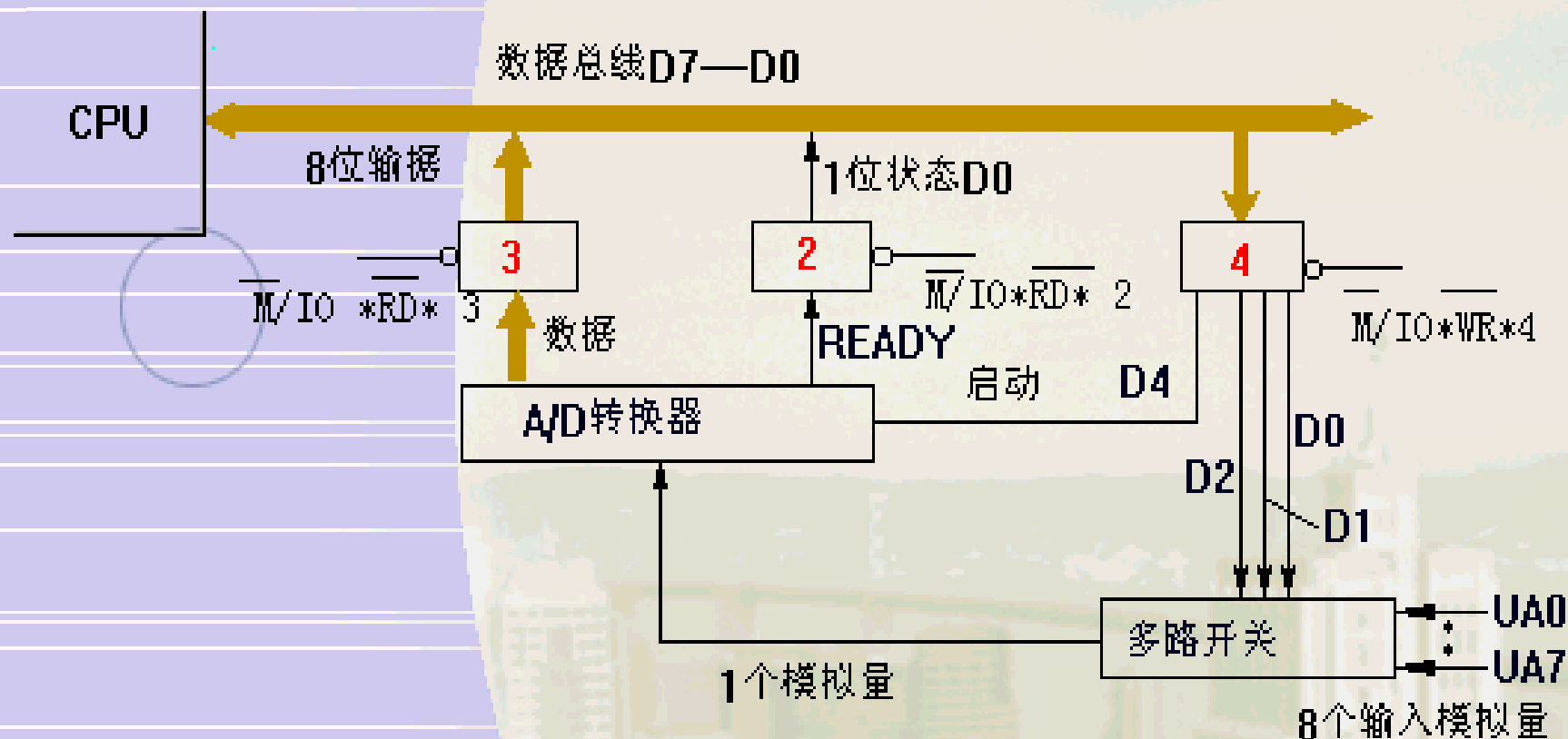
查询式数据采集系统



```

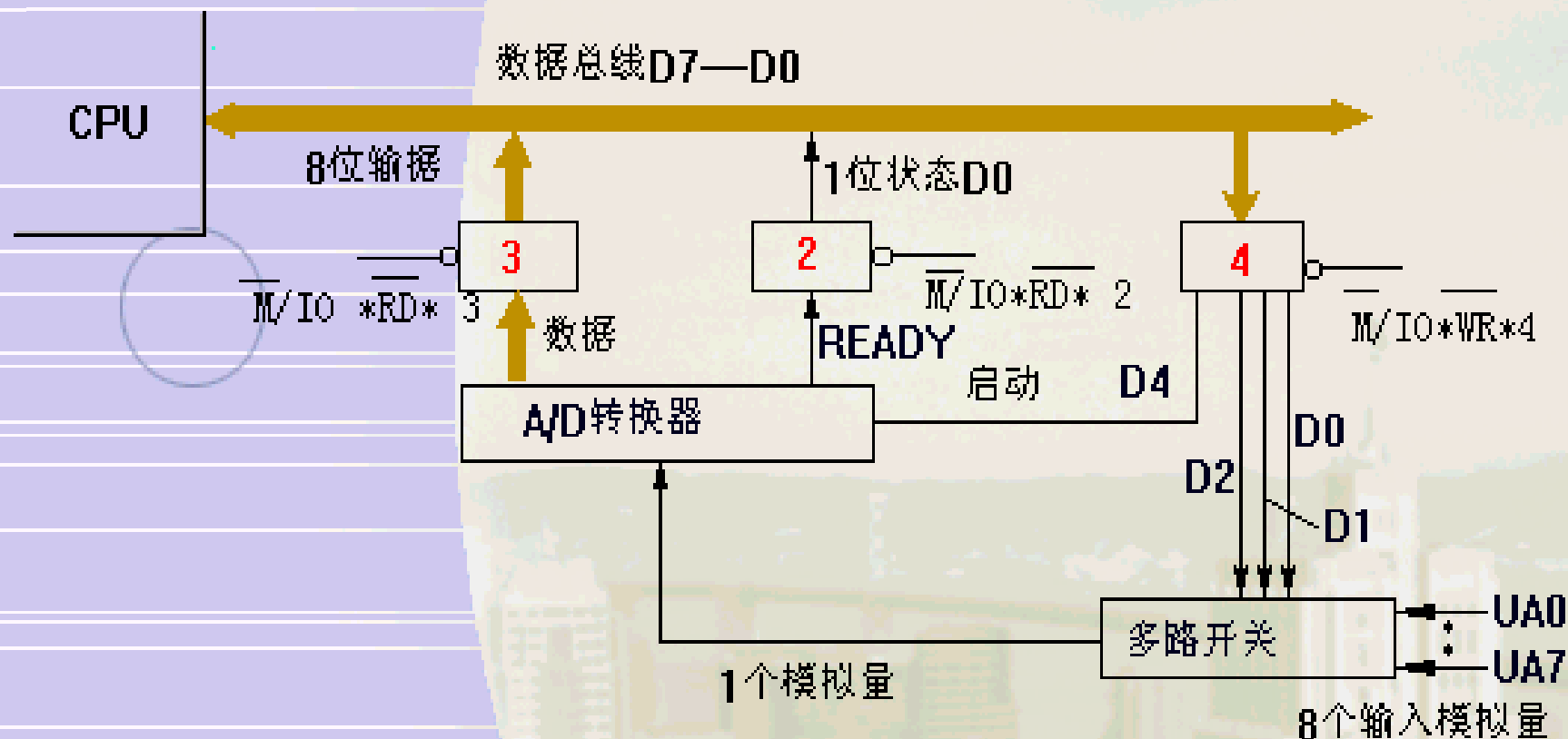
START:  MOV DL, 0F8H      : 设置启动A / D转换的信号,
                                : 且低3位选通多路开关通道
        MOV DI, DSTOR    : 设置输入数据的内存单元地址指针
AGATN:  MOV AL, DL
        AND AL, 0EFH     : 使 D4 = 0
        OUT 04H, AL      : 停止A / D转换
        CALL DELAY       : 等待停止A / D转换操作的完成
        MOV AL, DL
  
```

查询式数据采集系统



OUT 04H, AL	: 选输入通道并启动A / D转换
POLL: IN AL, 02H	: 输入状态信息
SHR AL, 1	: 查A的D0
JNC POLL	: 检验READY=1? 若CF=0, 即D0=0
	: 未准备好, 则循环再查。

查询式数据采集系统



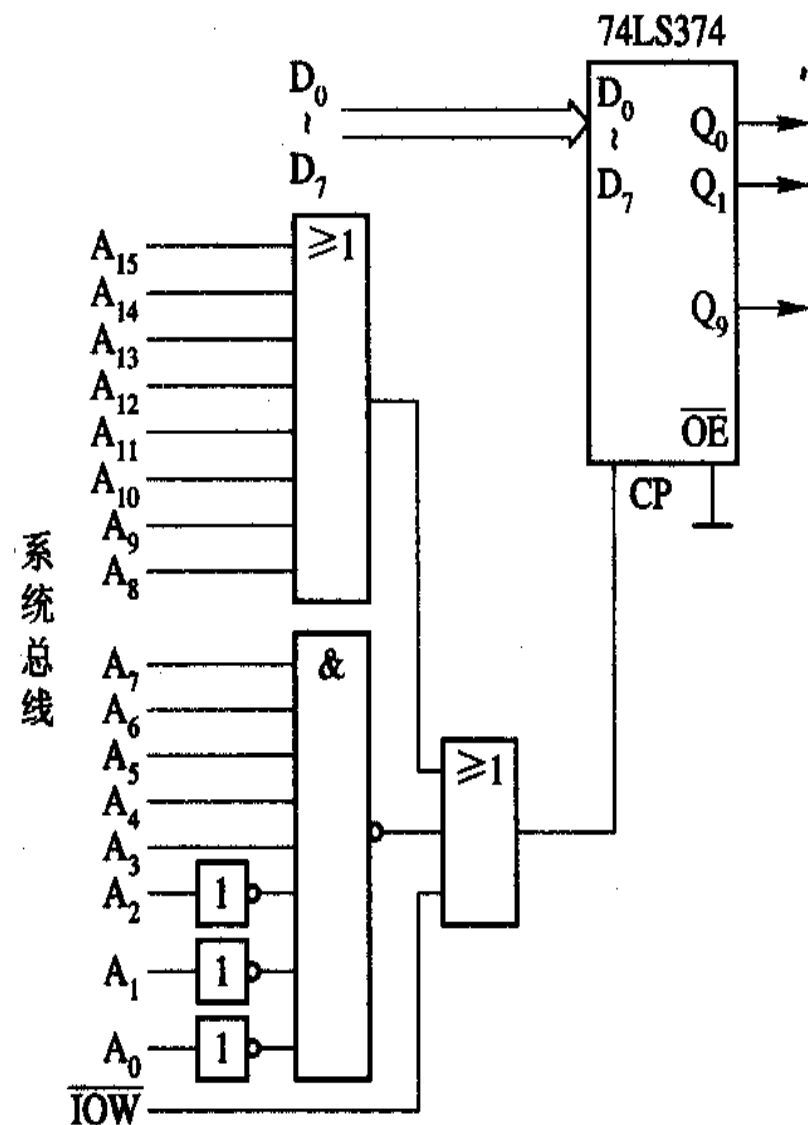
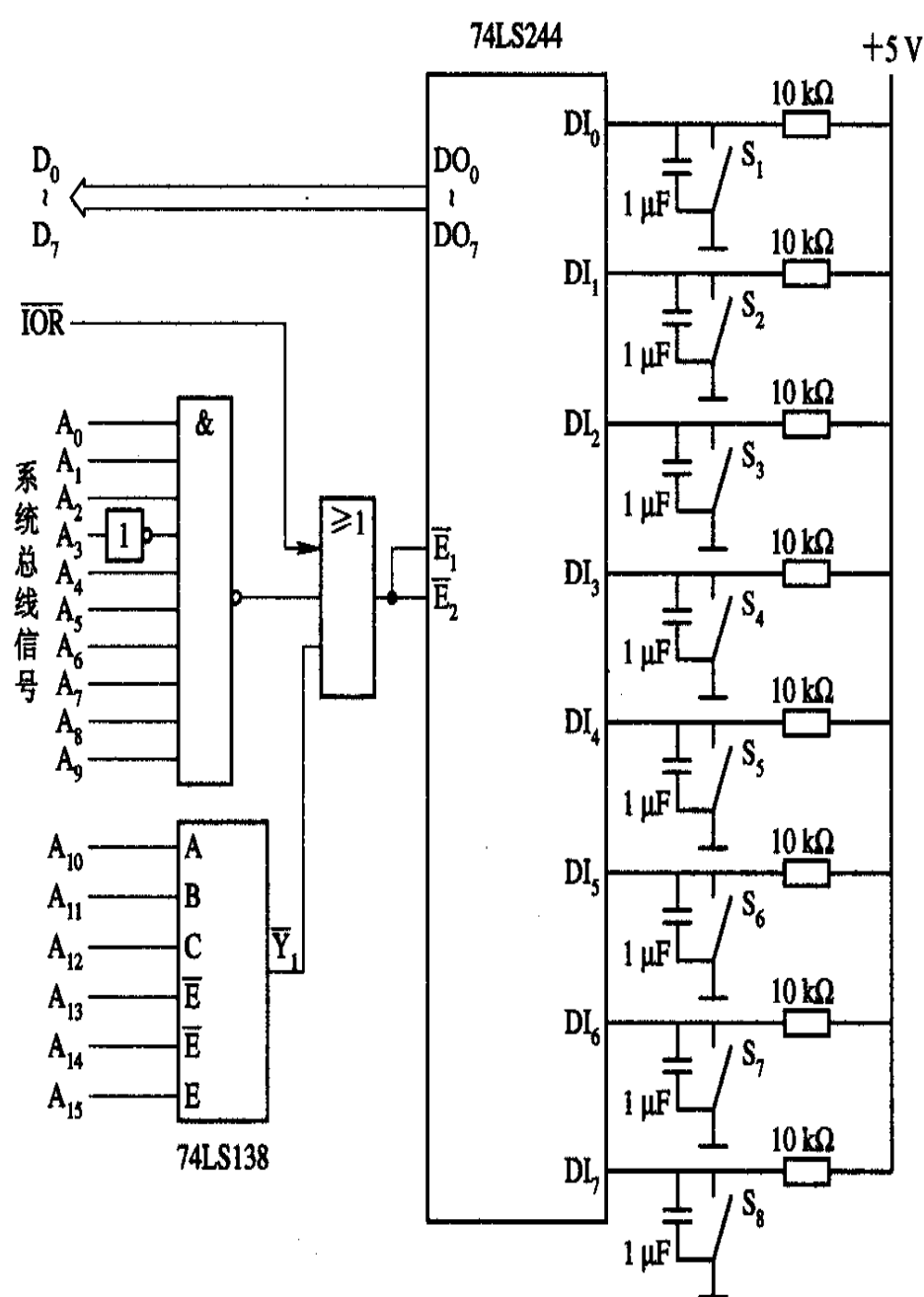
IN AL, 03H: 若已准备就绪, 则经端口 3 将采样数据输入至AL
 STOSB: 输入数据转送内存单元
 INC DL: 输入模拟量通道增 1
 JNE AGAIN: 8 个模拟量未输入完则循环

[例2] 利用具有三态输出的锁存器（74LS374）作为输出接口，接口地址为00F8H，若前图a输入接口的bit3、bit4和bit7**同时为1**时，将DATA为首地址的10个内存数据连续由输出接口输出；若不满足条件则等待，试编程序。

注：输入接口用前面的图a，其接口地址为87F7H。

根据此例要求，**即主要是查询接口地址为87F7H的输入接口的状态，确定是否输出10个数据**

根据给定的接口地址，其输出接口硬件连接图如下所示



程序段如下：

```
TDATAP: MOV DX, SEG_DATA
        MOV DS, DX
        MOV SI, OFFSET DATA
        MOV CX, 10
L1:     MOV DX, 87F7H
        IN AL, DX
        AND AL, 98H
        CMP AL, 98H
        JNE L1
        MOV DX, 00F8H
L2:     MOV AL, [SI]
        OUT DX, AL
        INC SI
        LOOP L2
        HLT
```

；判断是否同时为1

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
1	0	0	1	1	0	0	0

98H

4. 系统与多个外设连接时的数据传送

当系统与多个外设连接并利用查询方式进行数据的输入/输出时，可以有两种方法：

(1) 利用轮流查询的方式来检测接口的状态位。

这种查询方式，可以通过程序的优先级来决定设备的优先级。根据这样的思想，可根据各外设的轻重缓急，给外设安排一个优先级链。

4. 系统与多个外设连接时的数据传送

(2) 利用循环查询的方式来检测接口的状态位。

这种查询方式使几个外设处于完全等同的地位，即没有优先级。

例如：见教材P₁₈₇程序所示

5.4.2 中断传送方式

1. 为何要采用中断方式

可使CPU与外设并行工作，消除等待时间，提高CPU的工作效率和提高系统中多台外设的工作效率。

2. 中断的工作原理

对于中断传输过程，为了具有实时性能，一般采用外部中断。

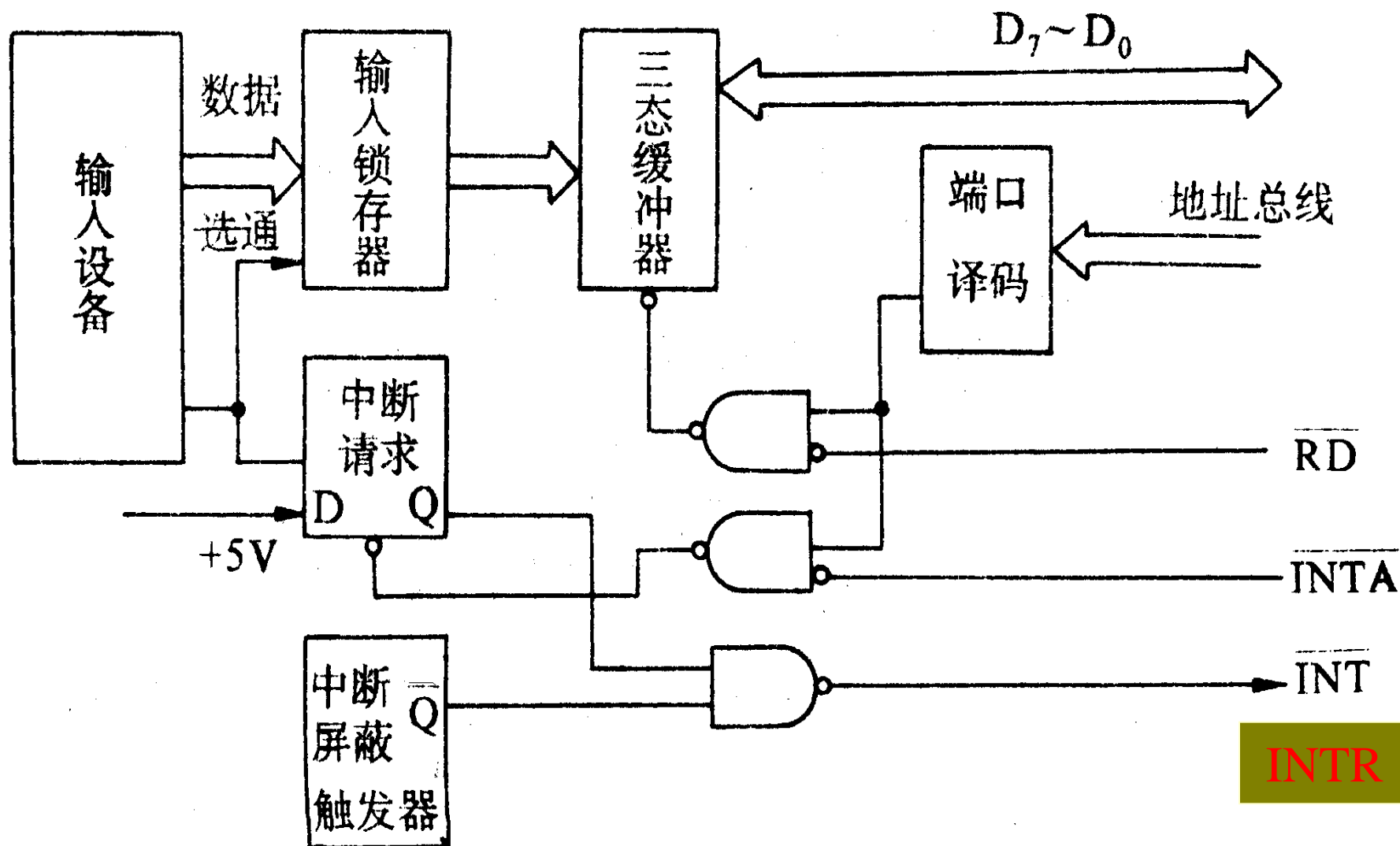
有两个引脚：NMI：非屏蔽中断

INTR：可屏蔽中断

中断响应过程 (INTR)

CS: IP
↓ 中断服务
IN AL, PORT
OUT PORT, AL
IRET (SP+6)

- 由INTA连续送两个负脉冲，从数据总线上读取中断向量码
- 保护断点：
 - 将PSW入栈
 - 清除IF、TF标志
 - 将CS、IP入栈
- 得到中断服务程序入口地址：
 - IP ← (n×4)
 - CS ← (n×4+2)
- 保护现场：用一系列PUSH命令将中断服务程序中用到的一些寄存器值保护起来。



中断方式输入的接口电路

3. 中断优先级问题的解决

微型计算机系统中对中断优先级采用以下三种方法来解决：即

- 软件查询方式 P_{164} 图6-10和程序
- 简单硬件方式——菊花链法
- 专用硬件方式——中断控制器

5.4.3 DMA传送方式（即直接存储器传输方式）

1. DMA传送的提出

利用中断传送方式，虽可提高CPU的工作效率，但对于高速外设以及成组交换数据的场合，显得太慢，不能满足要求。**原因：**

- 仍需CPU通过**程序**来实现数据传送，并在处理中断时，还要“**保护断点、标志**”、“**保护现场**”和“**恢复现场**”。
- 是按**字或字节**来进行传输的。

为了解决这个问题，实现按**数据块**传输，就需要改变传输方式，这就是直接存储器传输方式，即**DMA方式**。

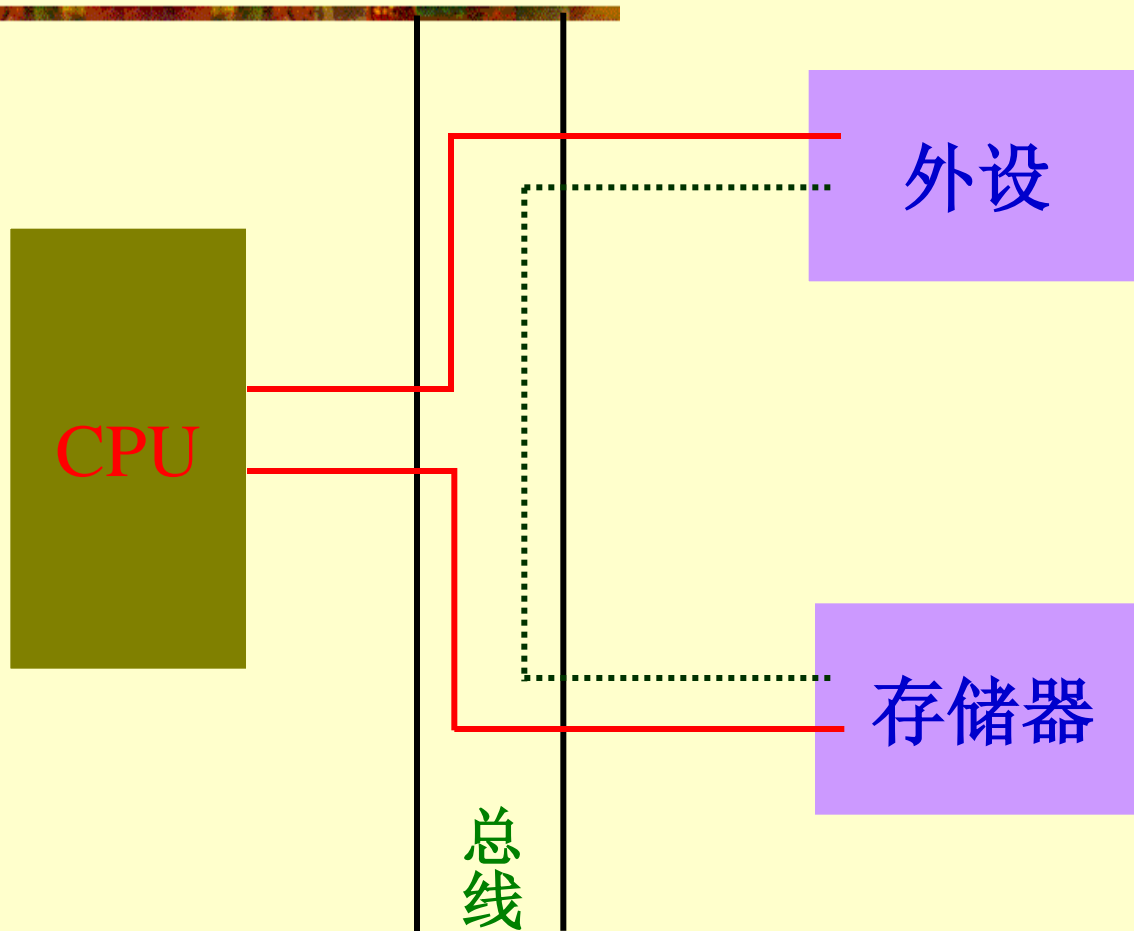
5.4.3 DMA传送方式（即直接存储器传输方式）

2. DMA传送的含义

DMA (Direct Memory Access) 方式或称为数据通道方式是一种由专门的硬件电路执行 I/O 交换的传送方式，它使外设接口可直接与内存进行高速的数据传送，而不必经过CPU，这样就不必进行保护现场之类的额外操作，可实现对存储器的直接存取。这种专门的硬件电路就是DMA控制器，简称为DMAC。

——：执行程序指令的数据传送路径

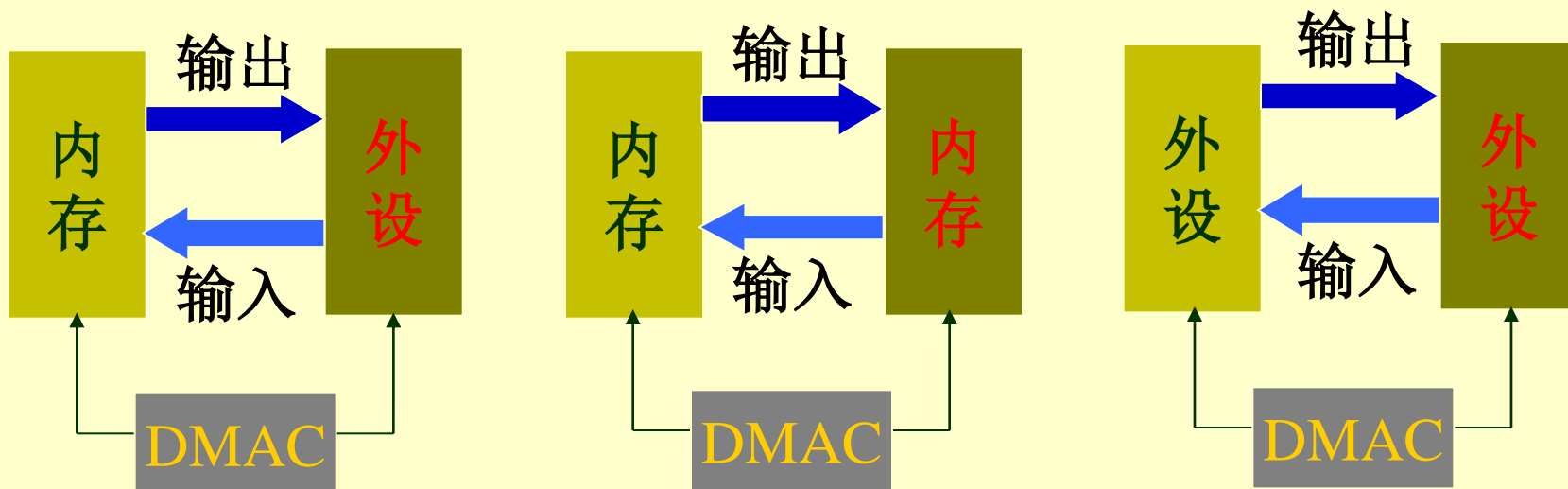
.....：DMA方式的数据传送路径



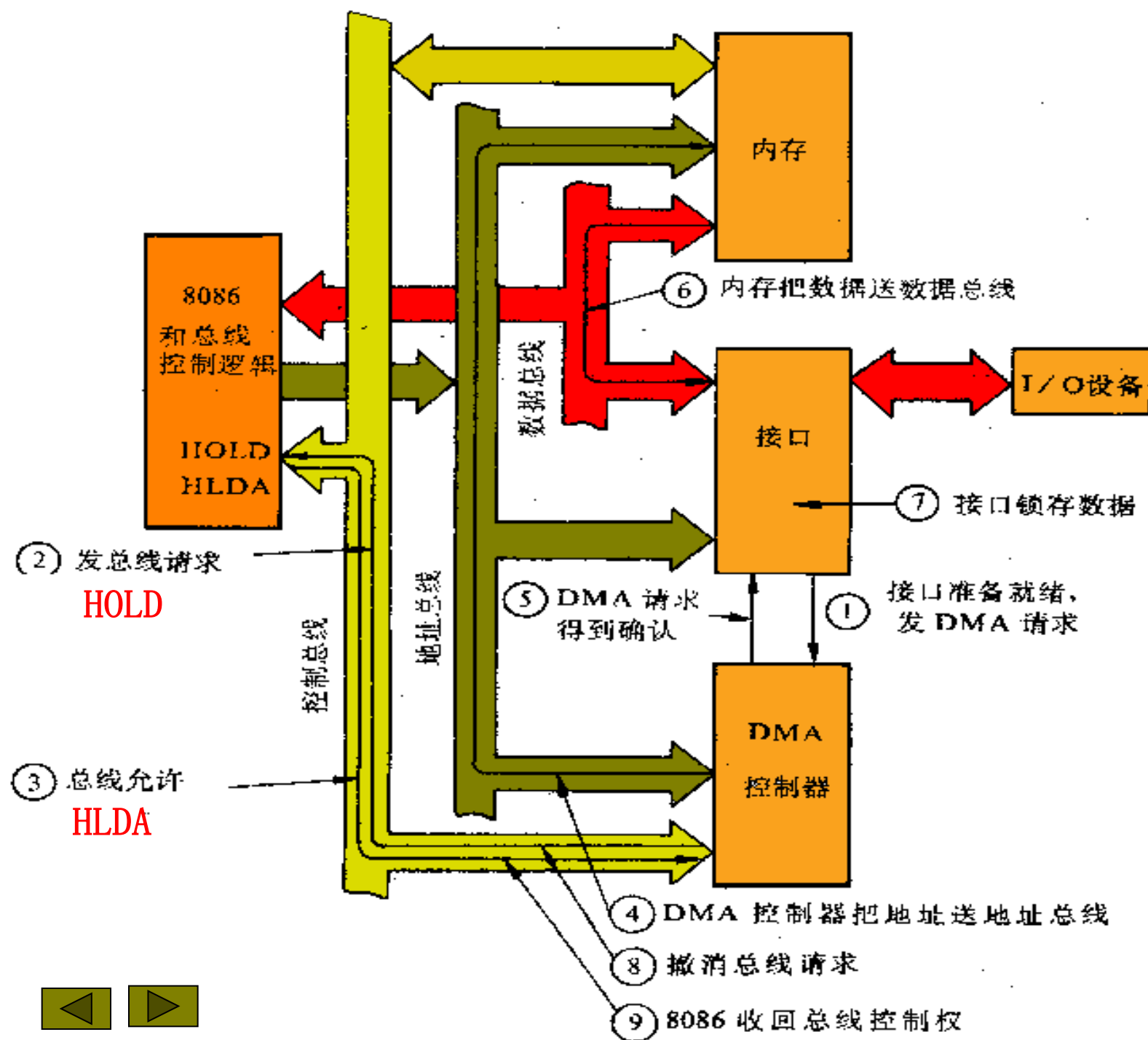
图为两种不同的传送路径

3. 几种不同形式的DMA传送

在DMAC的控制下，可实现如下三种形式的高速数据传送。



4. DMA 传送的原理



4. DMA控制器的工作特点

- ❖ 是一个接口电路，因为它也有I/O端口地址。
- ❖ 能够控制系统总线，可以提供一系列控制信号，像CPU一样操纵外设和存储器之间的数据传输，所以DMA控制器又不同于一般的接口电路。
- ❖ DMA控制器控制的数据传输不同于其它方式的传输；它在传输数据时**不用指令**，而是通过硬件逻辑电路用固定的顺序发地址和用读/写信号来实现高速数据传输，CPU完全不参与，数据也不经过CPU，而是直接在外设和内存之间传输。

总结：

1. 三种基本输入输出方式

即程序直接控制方式、程序中断控制方式和存储器直接存取（DMA）方式。它们**传送信息的速度依次越来越快**，其**传送效率也越来越高**，但其实现和管理的复杂性也越来越高。

程序直接控制和中断控制传送方式下的信息传送是通过IN和OUT指令实现的，

DMA方式则在存储器与外设之间架起直接访问的通路，因此与CPU的IN、OUT指令无关，其存储速度是芯片的速度。



2. 系统与接口部件之间的通信联络

在查询方式下，是通过程序来检测接口中状态寄存器中的“准备好”（READY）位，来确定当前是否可以进行数据传输；

在中断方式下，当接口中已经有数据要送往CPU或准备好接收数据时，接口会向CPU发一个外部中断请求，CPU在得到中断请求后，如果响应中断，便通过运行中断处理程序来实现输入/输出；

在DMA方式下，外设要求传输数据时，接口会向DMA控制器发请求信号，DMA控制器转而往CPU发送一个总线请求信号，以得到总线的控制权，如果得到DMA允许，那么，就可以在没有CPU参与的情况下实现DMA传输。