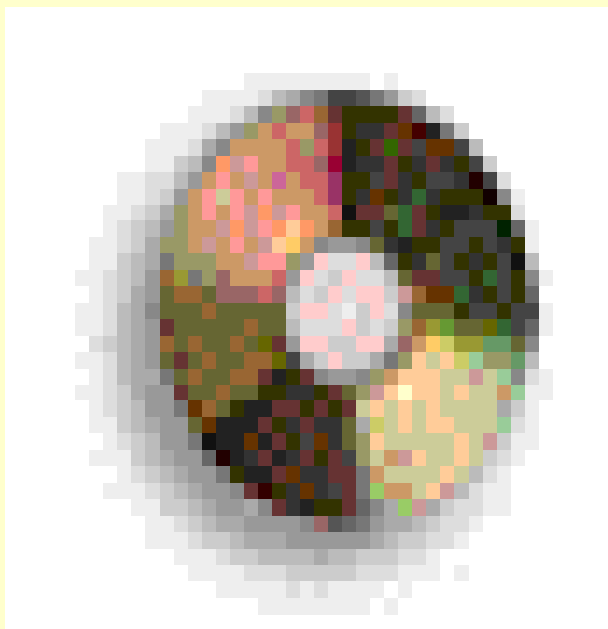


第六章 输入和输出



一、概述

二、输入输出的寻址方式

三、CPU与外设数据传送的方式

1. 程序直接控制方式

① 无条件传送方式

② 查询传送方式

2. 中断传送方式

3. 直接数据通道传送 (DMA)

四、各传送方式的比较

一、概述

计算机在使用中，不可避免地要与外部设备打交道，输入和输出一些信息数据和信息。

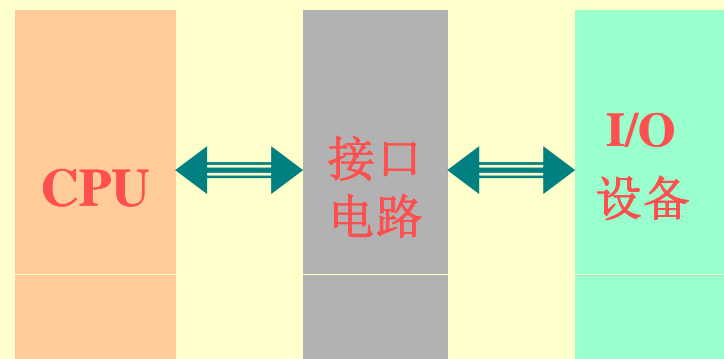
输入——通过键盘、纸带读入机、卡片输入机、扫描仪、
A/D（模/数）转换等；

输出——显示器、打印机、D/A（数/模）转换等。

另外，在微机中，软、硬盘也作为输入输出设备。

- 由于 ① 输入的信息的数据形式不同
（数字量、模拟量、开关量等）；
② 输入信息的速度不同
（键盘输入、磁盘输入）；

所以，在CPU与外设进行数据传输时，
需要通过接口电路来实现。



CPU与外设之间的接口信息

CPU与一个外设交换信息，通常需要以下一些信号：

1. 数据（Data）

数字量——键盘等输入的数据；

模拟量——如温度、压力、流量等；

（需先经过A/D转换，将其变为数字量，由CPU处理后，再经过D/A转换，变为模拟量，进行控制），这部分工作有接口电路来完成。

开关量——如开关的合与断等。

2. 状态信息（Status）

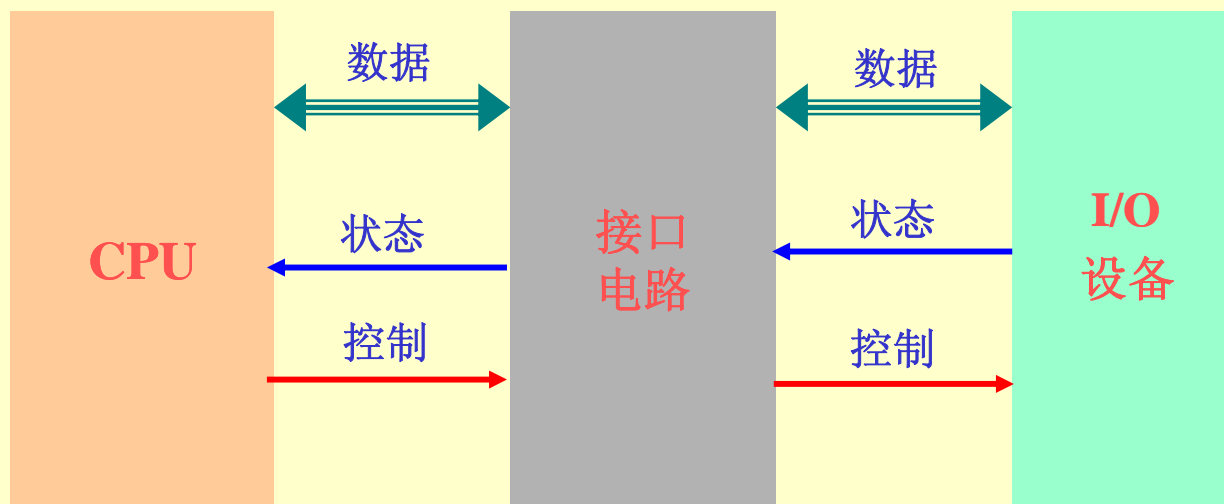
输入时，输入设备的信息是否准备好；

输出时，输出设备是否有空等。

3. 控制信息（Control）

控制输入输出设备的启动和停止等。





为了将数据、状态、控制信息区分开，在接口电路中，设有不同的端口如数据端口、状态端口、控制端口，以接收、存放、输出不同的信息。



端口——用于存放信息的8位或16位锁存器，缓冲器等。
在PC机中通常为8位

二、输入输出的寻址方式

CPU 寻址外设有两种方式：

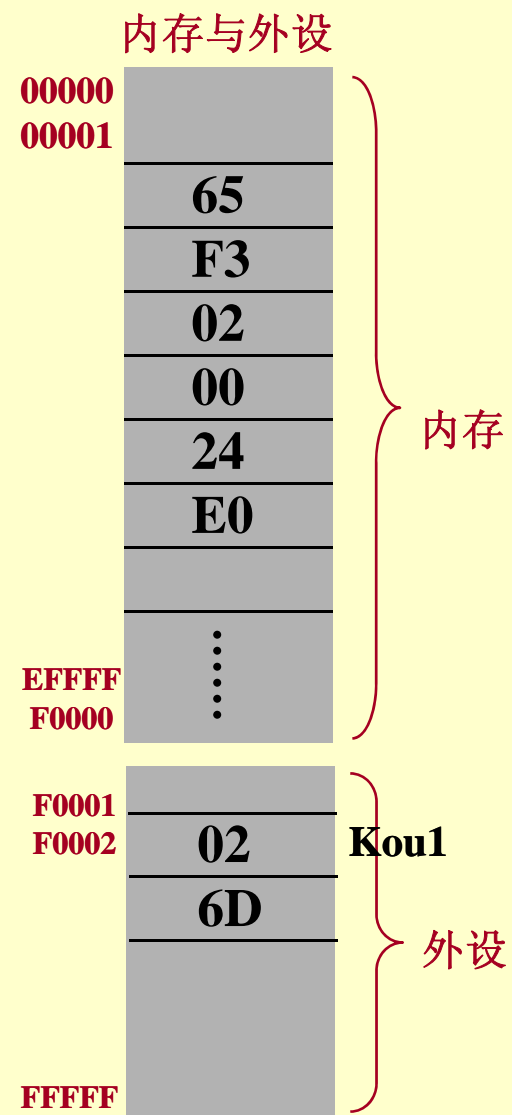
1. 将存储器与外设端口统一编址

将外设接口电路的一个端口作为存储器的一个单元。

优点： 不需要专门的输入输出指令，可用全部的存储器操作指令。（指令多且灵活）

如： `mov kou1,bx`

缺点： 外设占用内存单元，相对减少了内存容量。



2. 外设端口单独编址

优点：不占用内存

缺点：CPU需设专门的I/O指令。

I/O 指令：

- 若端口地址在 **0~FFH** 范围内，则用：

IN AL, 端口地址 (输入)

OUT 端口地址, AL (输出)

如： **IN AL, PORT1**

OUT PORT2, AL

可寻址 $2^8=256$ 个端口

- 若端口地址在 **0100~FFFFH** 范围内，则用：

MOV DX, 端口地址

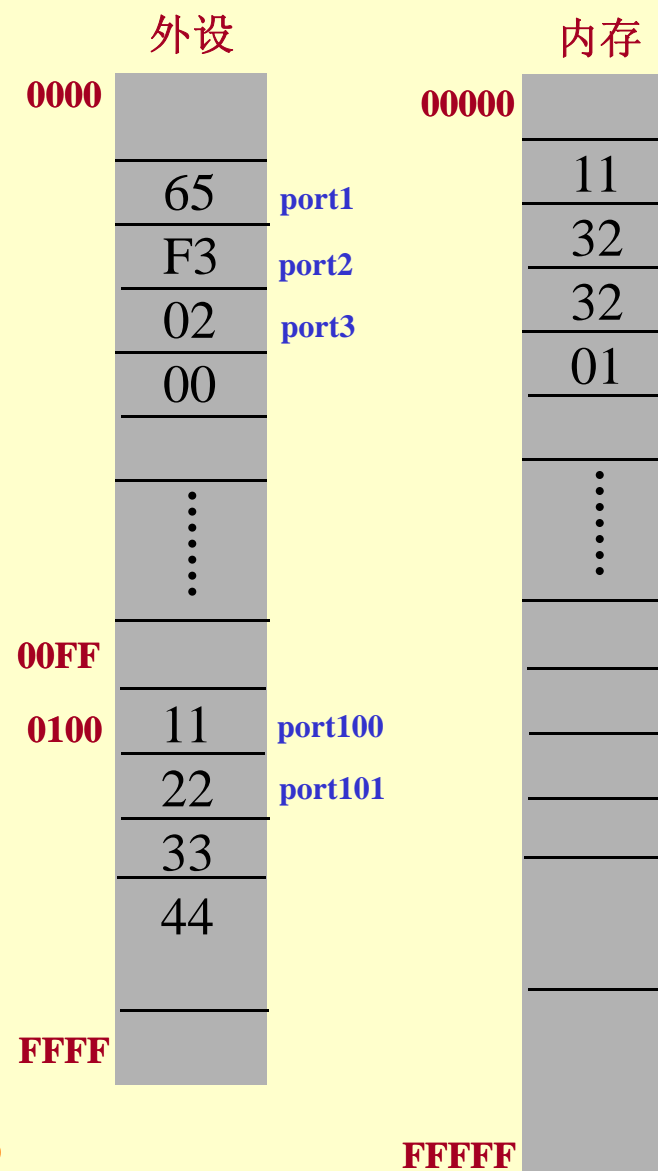
IN AL, DX

OUT DX, AL

如： **MOV DX, PORT100**

IN AL, DX

可寻址 $2^{16}=64K$ 个端口



附:

1. IBM PC/XT I/O 端口地址分配图

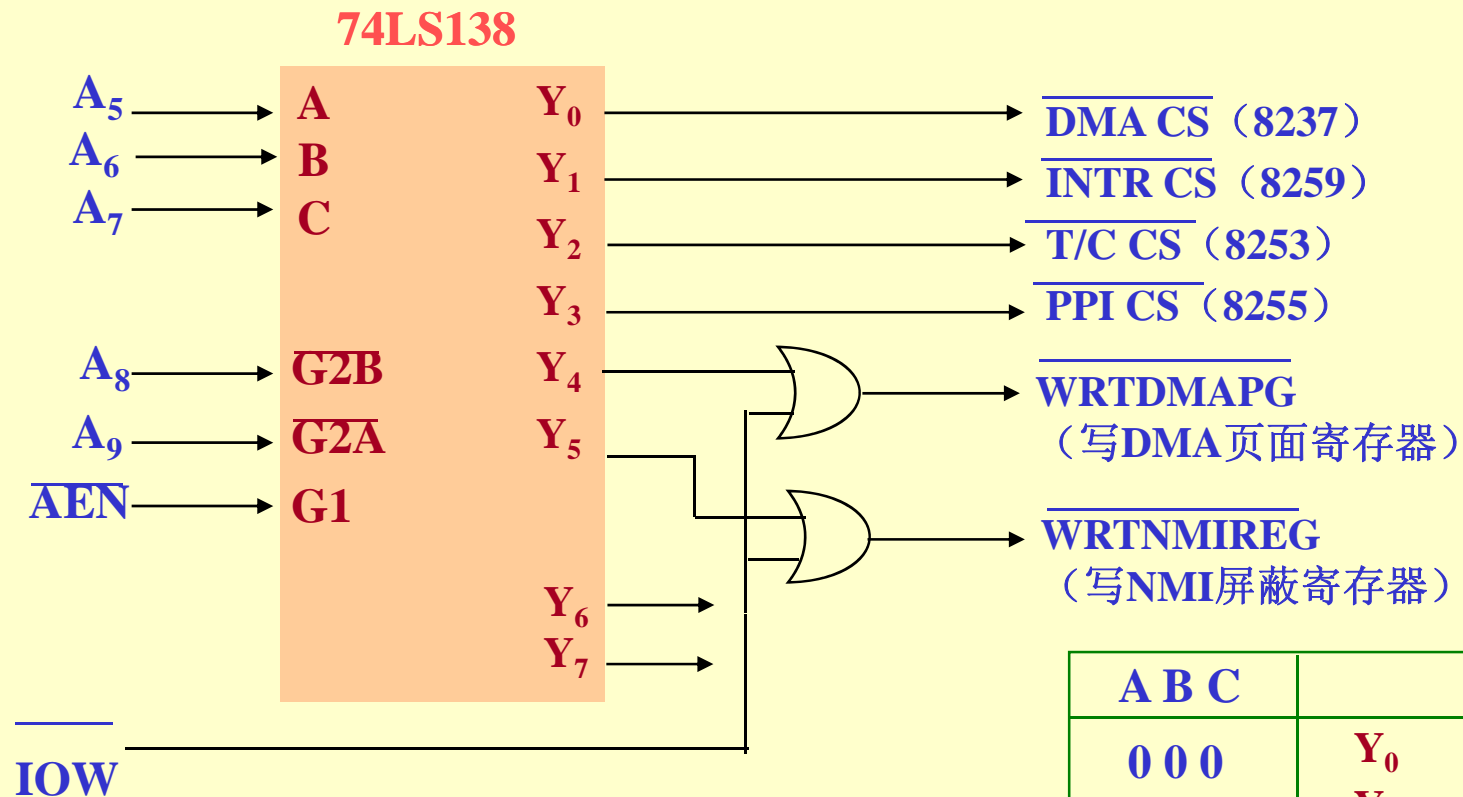
0000	32字节	0000~000F	8237A—5 DMA控制器
001F			
0020	32字节	0020~0021	8259A 中断控制器
003F			
0040	32字节	0040~0043	8253A—5 定时/计数器
005F			
0060	32字节	0060~0063	8255A—5 并行接口芯片
007F			
0080	32字节	0080~0083	DMA 页面寄存器
009F			
00A0	32字节	00A0~00BF	NMI 屏蔽寄存器
00BF			
0080	320字节		
01FF			



0200		0200~020F	游戏控制口
		0210~0217	扩展部件
		0218~02F7	未用
		02F8~02FF	异步通信卡（第二个）
		0300~031F	实验卡
		0320~032F	硬盘适配器
		0330~0377	未用
		0378~037F	并行打印机
		0380~038F	SPLC通信
		0390~03AF	未用
		03B0~03BF	单色显示器 / 打印机
		03C0~03CF	未用
		03D0~03DF	彩色/ 图形显示卡
		03E0~03EF	未用
		03F0~03F7	软盘适配器
03FF		03F8~03FF	异步通信卡（第一个）



2. IBM PC/XT I/O 端口地址译码方法



当 $\begin{cases} G1=1 \\ \overline{G2A}=\overline{G2B}=0 \end{cases}$ 时，允许对A、B、C进行译码。

译中的对应输入端将变为低电平，其它7个保持高电平。

A	B	C	
0	0	0	Y_0
0	0	1	Y_1
0	1	0	Y_2
0	1	1	Y_3
1	0	0	Y_4
1	0	1	Y_5
1	1	0	Y_6
1	1	1	Y_7

如: ① OUT 43H, AL

A ₉	A ₈	A ₇	A ₆	A ₅	A ₄	A ₃	A ₂	A ₁	A ₀
0	0	0	1	0	0	0	0	1	1

$$\begin{cases} A_9 = A_8 = 0 \\ AEN = 1 \end{cases}$$

A₇ A₆ A₅ = 0 1 0 —— 译中 Y₂

② OUT 20H, AL

A ₉	A ₈	A ₇	A ₆	A ₅	A ₄	A ₃	A ₂	A ₁	A ₀
0	0	0	0	1	0	0	0	0	0

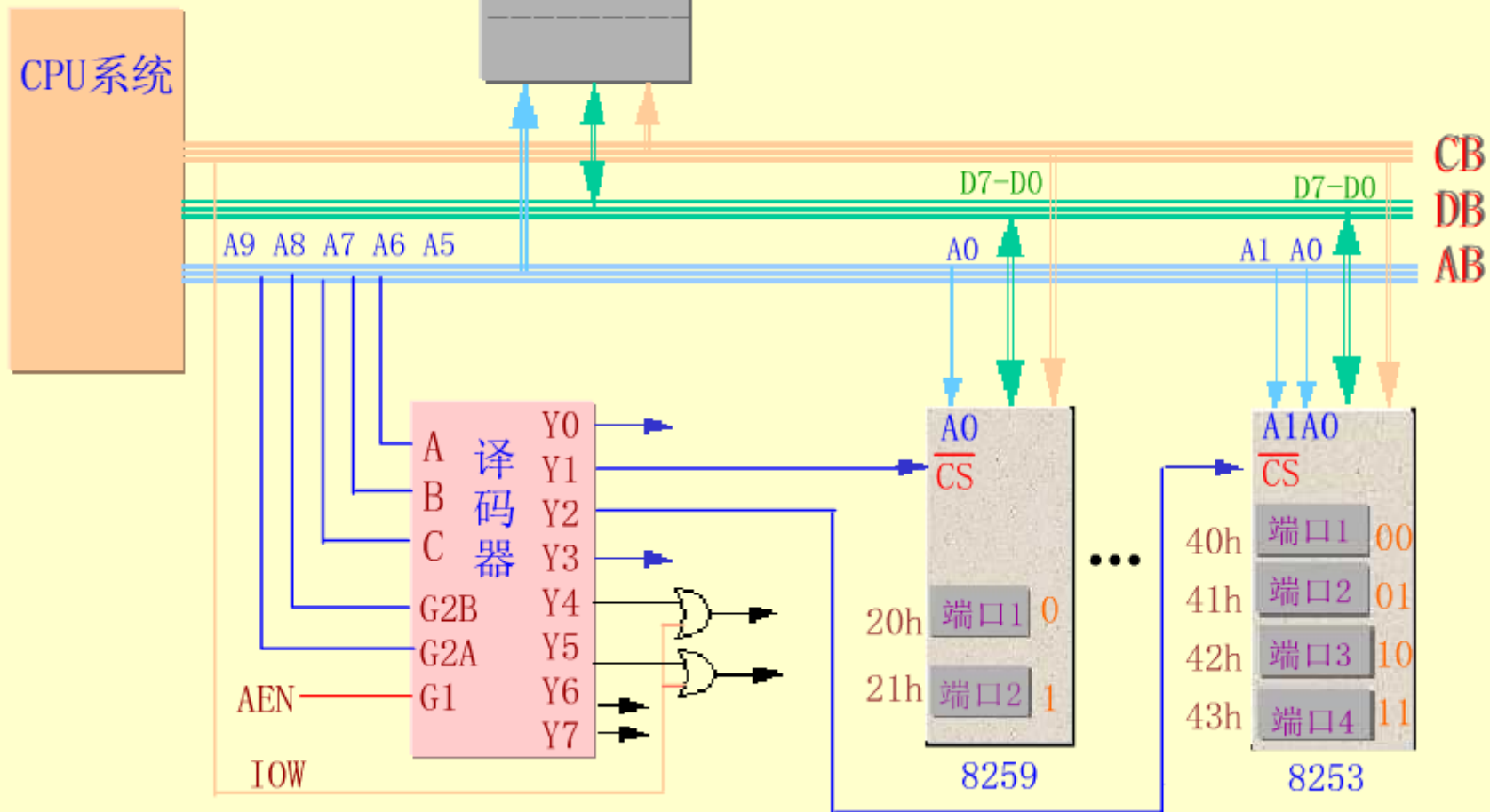
$$\begin{cases} A_9 = A_8 = 0 \\ AEN = 1 \end{cases}$$

A₇ A₆ A₅ = 0 0 1 —— 译中 Y₁



1011E MOV AL, 90
10120 OUT 43H, AL

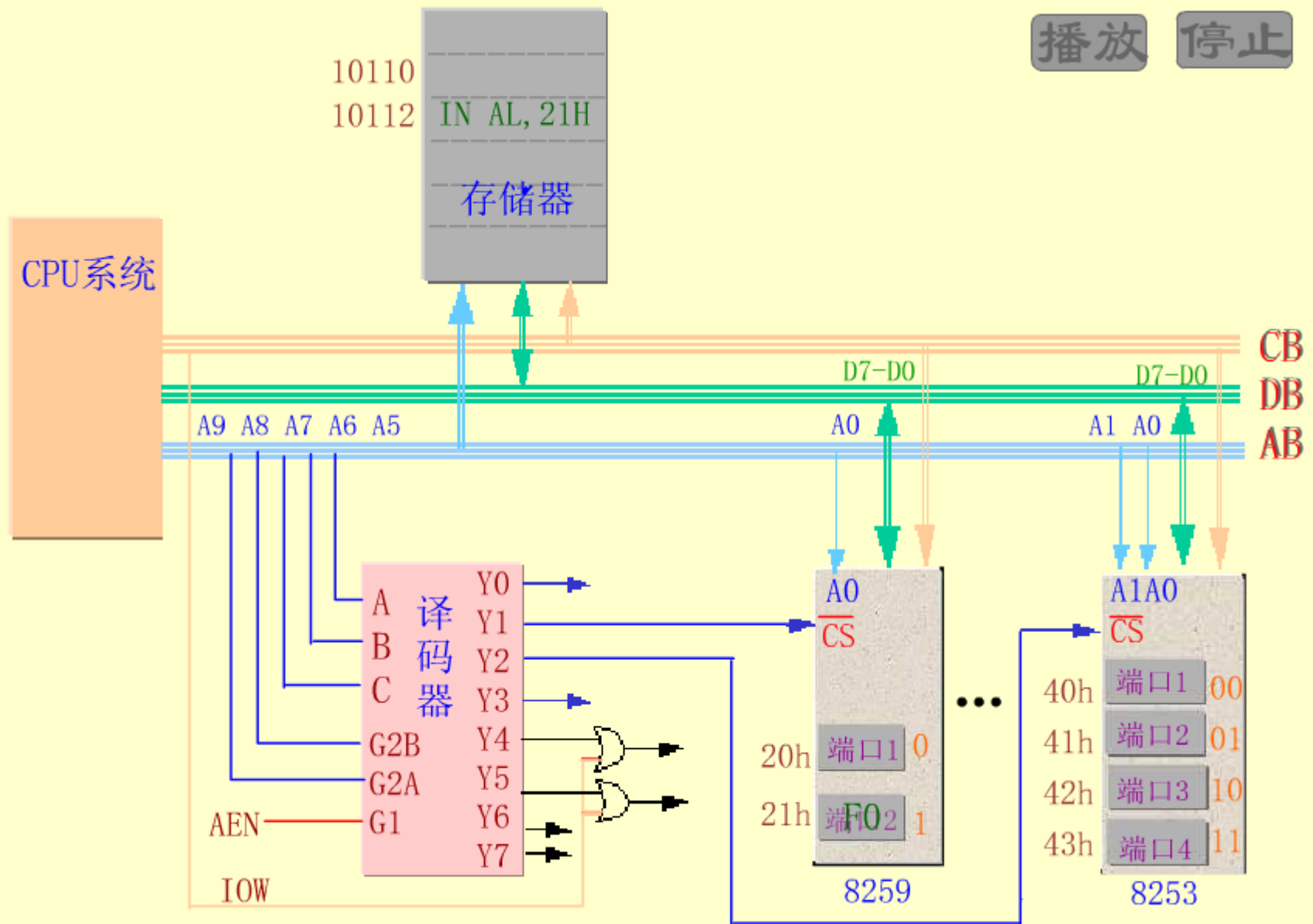
播放 停止



OUT 43H, AL



播放 停止



IN AL, 21H



三、CPU与外设数据传送的方式

CPU与外设数据传送的方式有三种：

1. 程序直接控制方式

- ① 无条件传送方式
- ② 查询传送方式

2. 中断传送方式

3. 直接数据通道传送（DMA）

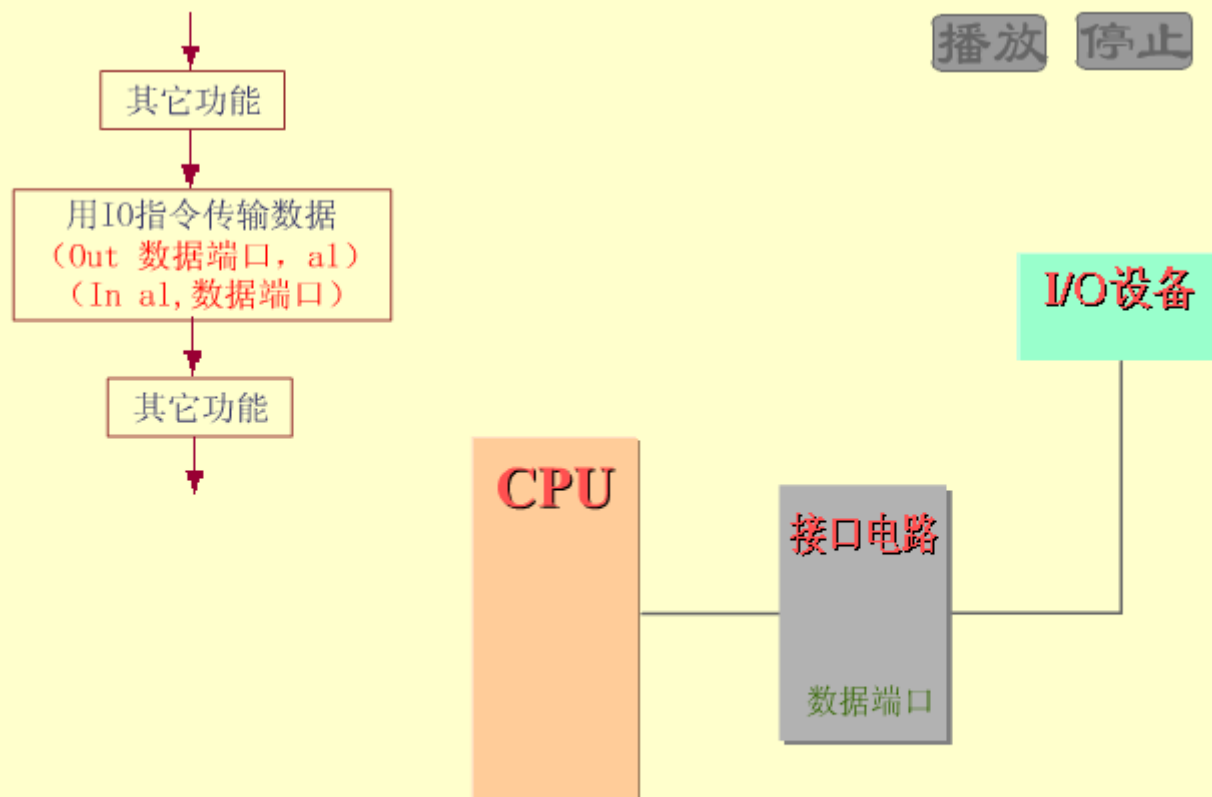


1. 程序直接控制方式

① 无条件传送方式

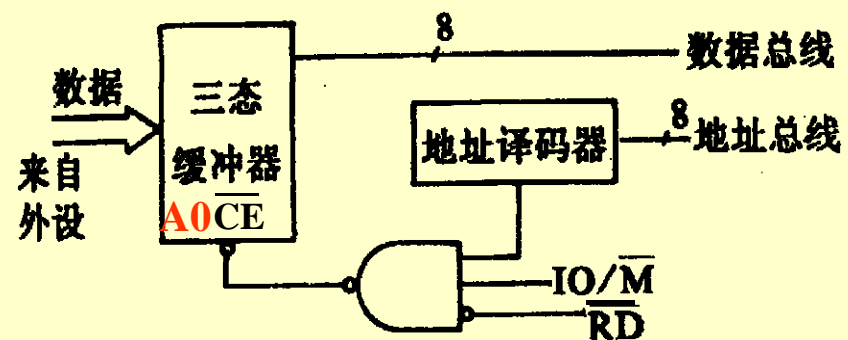
无条件传送方式又称同步方式，即CPU的动作必须与外设同步，否则，传送数据出错。

传送流程演示



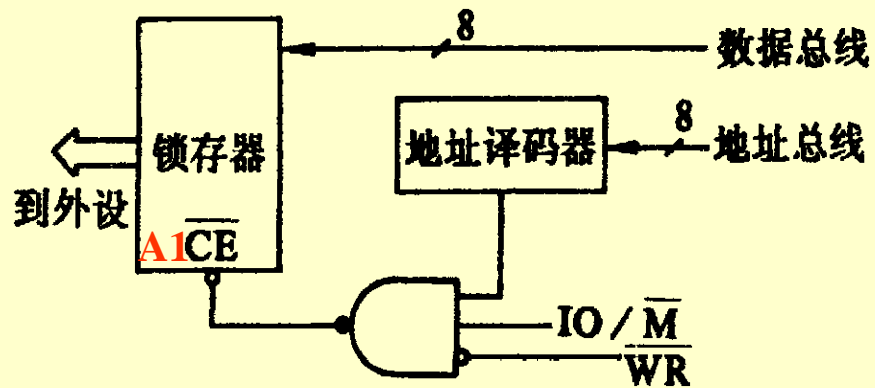
输入方式

IN AL, 0A0H



输出方式

OUT 0A1H, AL



例：采用同步传送的数据采集系统

Data segment

Dsiok dw 100 dup(?)

Data ends

Code segment

assume cs:code,ds:data

Start: mov ax,data

mov ds,ax

mov dx, 0100

lea bx, dsiok

xor al, al

Again: mov al, dl

out [20h], al

call near delay1

mov al, dh

out [20h], al

call near delay2

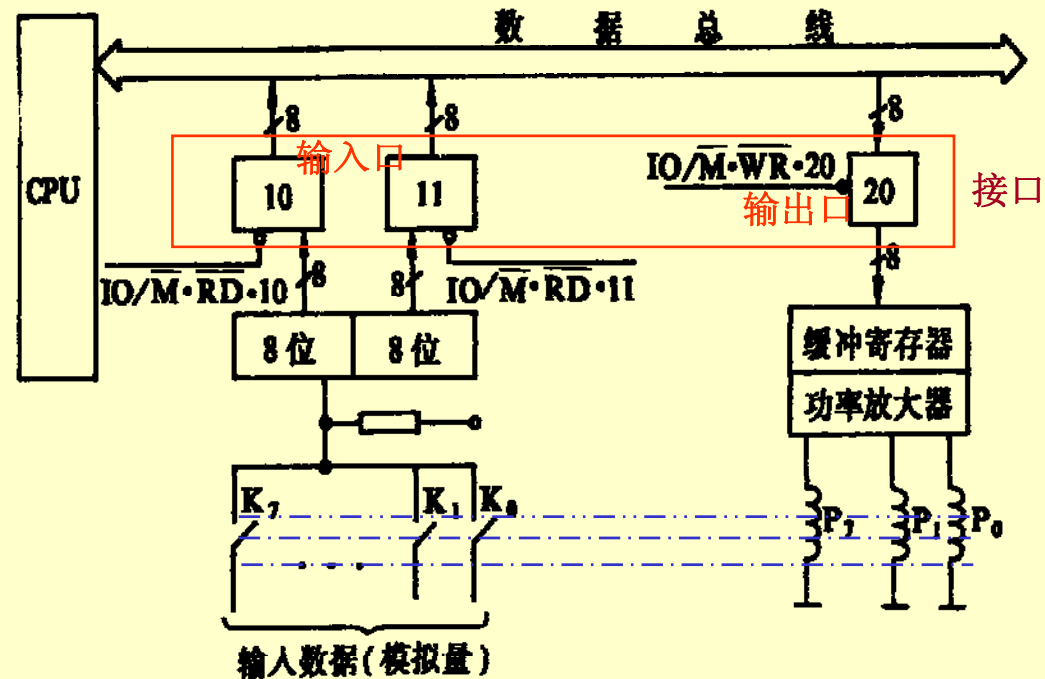
in ax, [10h]

mov [bx], ax

inc bx

rcl dh, 1

jnc again



```

mov ah, 4ch
int 21h
Delay1 proc near
push ax
push dx
mov ax, 3ffh
Loop2: mov dx, 0ffff h
Loop1: dec dx
jnz loop1

```

```

dec ax
jnz loop2
pop dx
Pop ax
ret
Delay1 endp
Code ends
end start

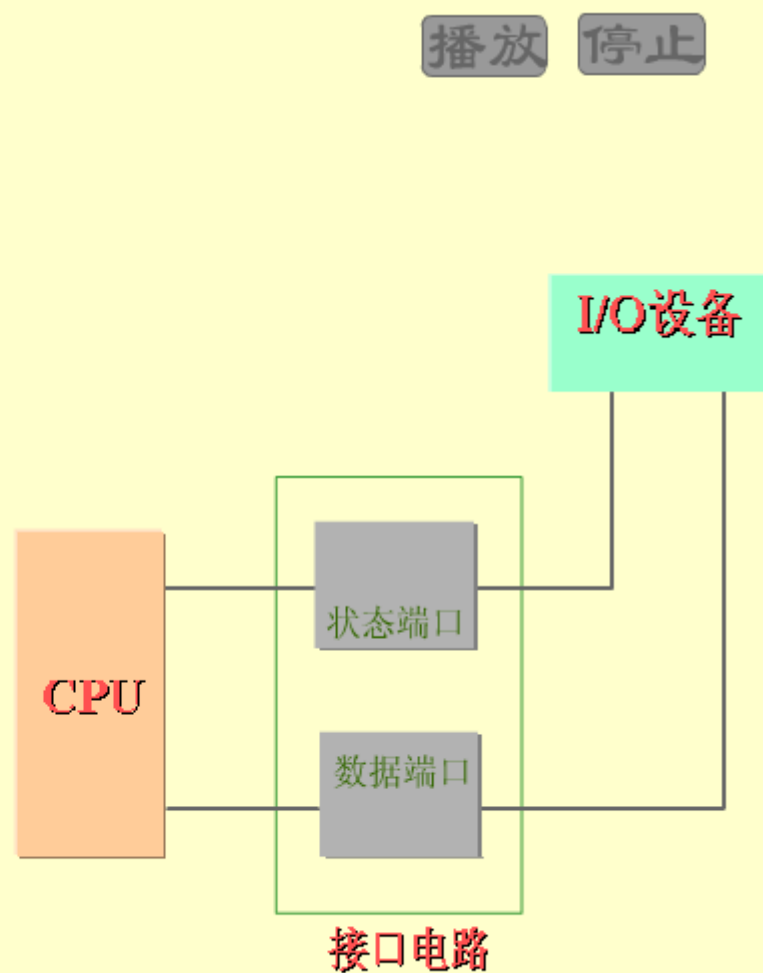
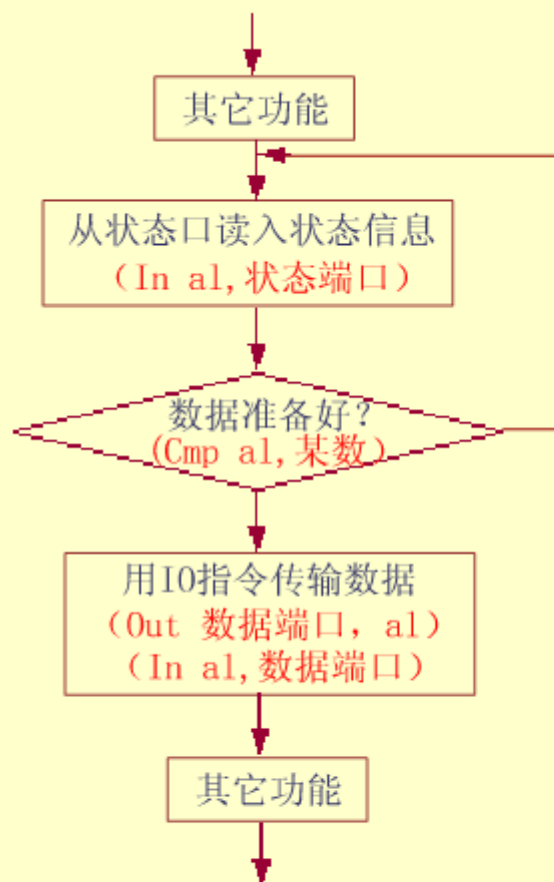
```



② 查询传送方式

在传送前，查询一下外设的状态，当外设准备好了以后才传送，否则，等待。

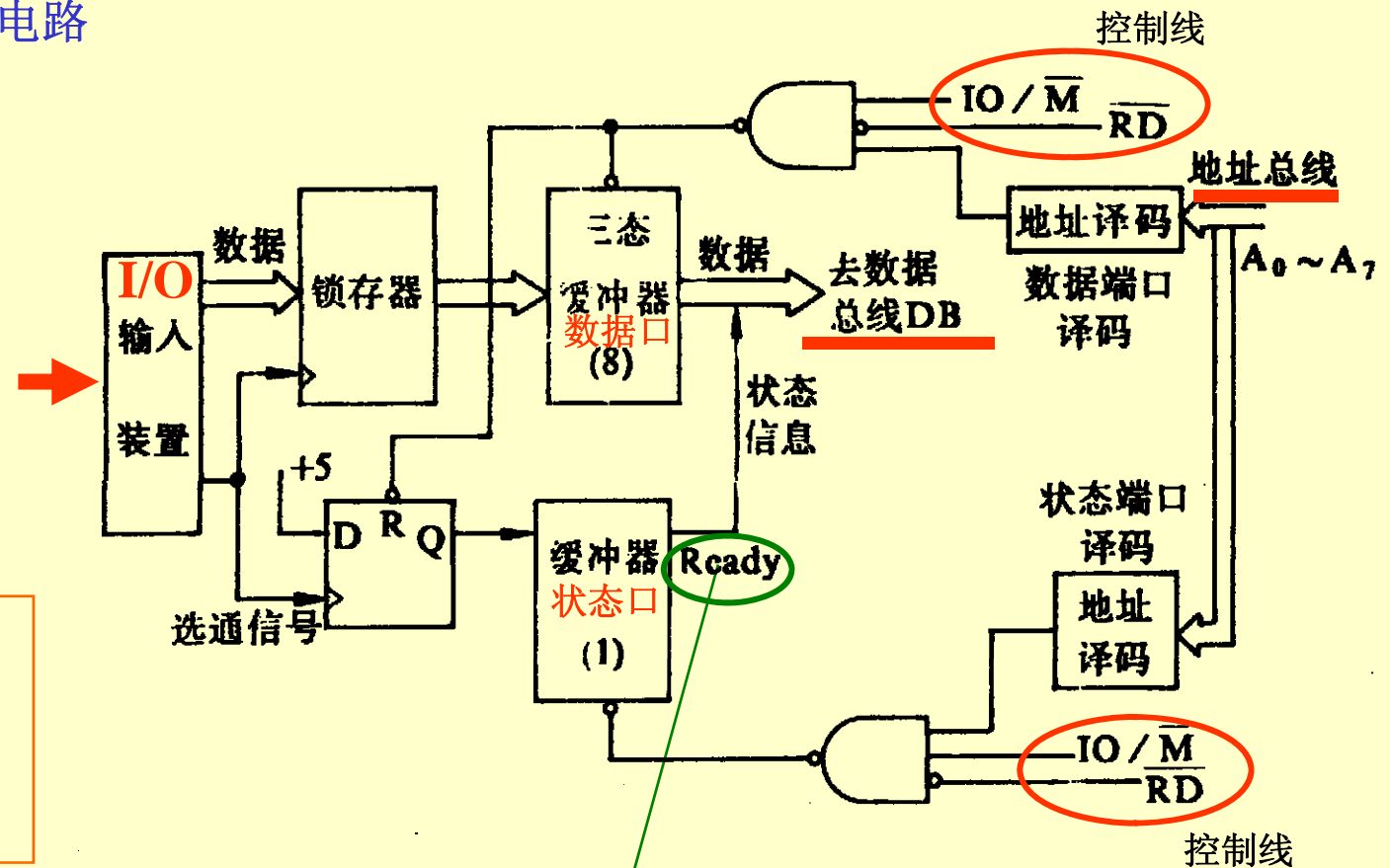
传送流程演示



播放 停止



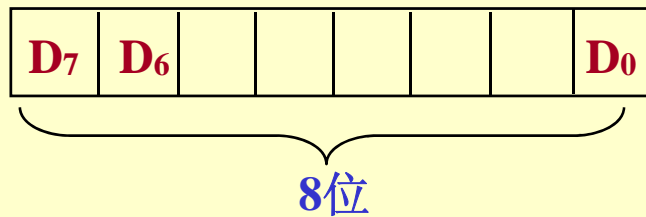
- 查询式输入接口电路



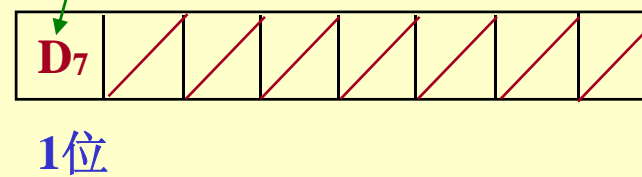
```

Poll: In al, S_port
      Test al, 80h
      Jz poll
      In al, D_port
    
```

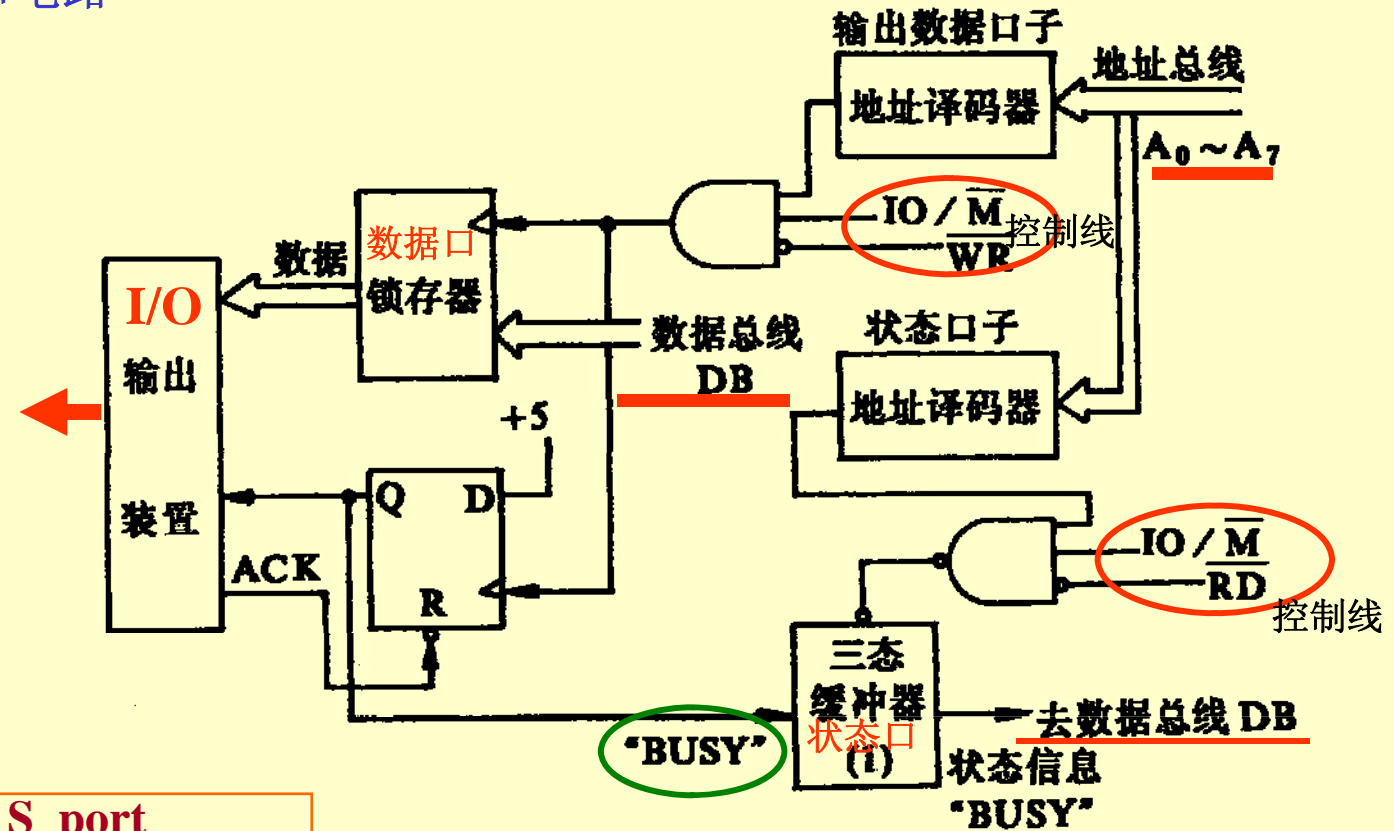
数据端口



状态端口



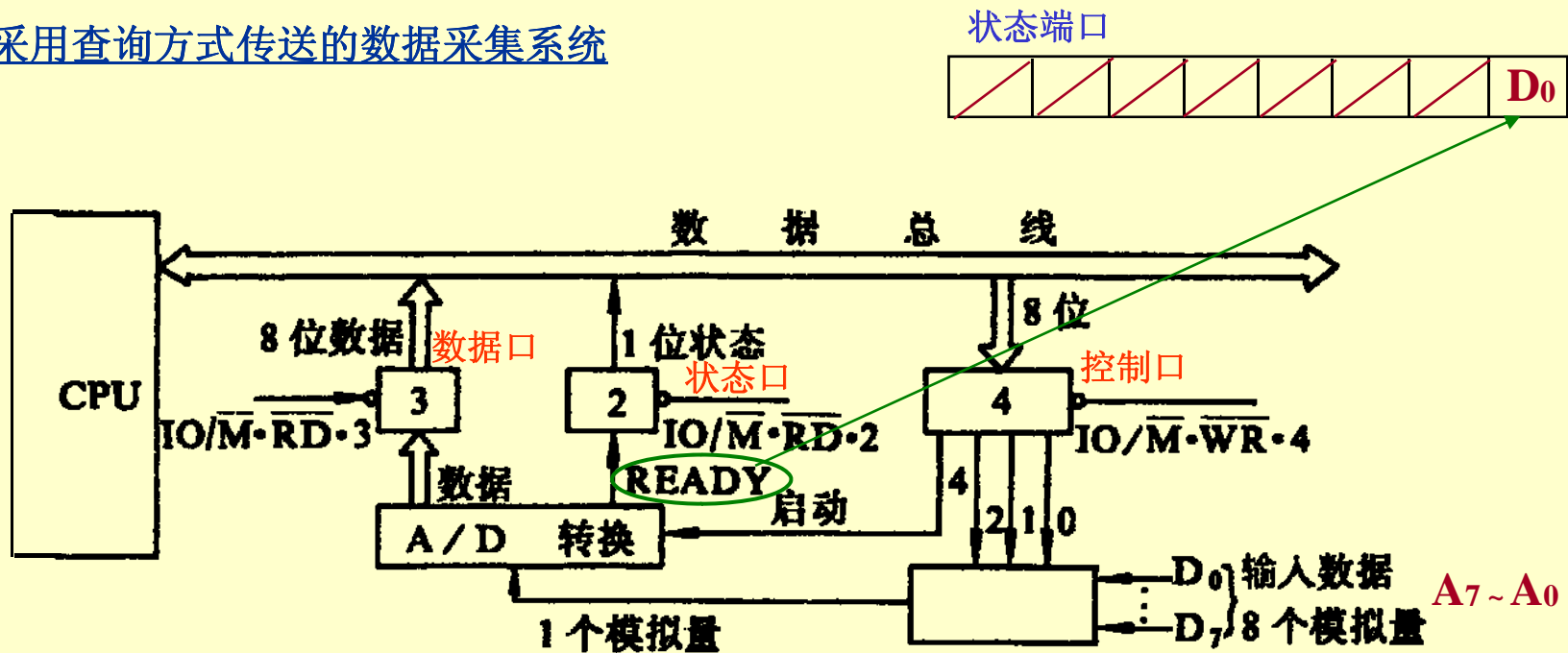
- 查询式输出接口电路



```

Poll: In  al, S_port
      Test al, 80h
      Jnz  poll
      mov  al, store
      Out  D_port, al
    
```

例：采用查询方式传送的数据采集系统



```

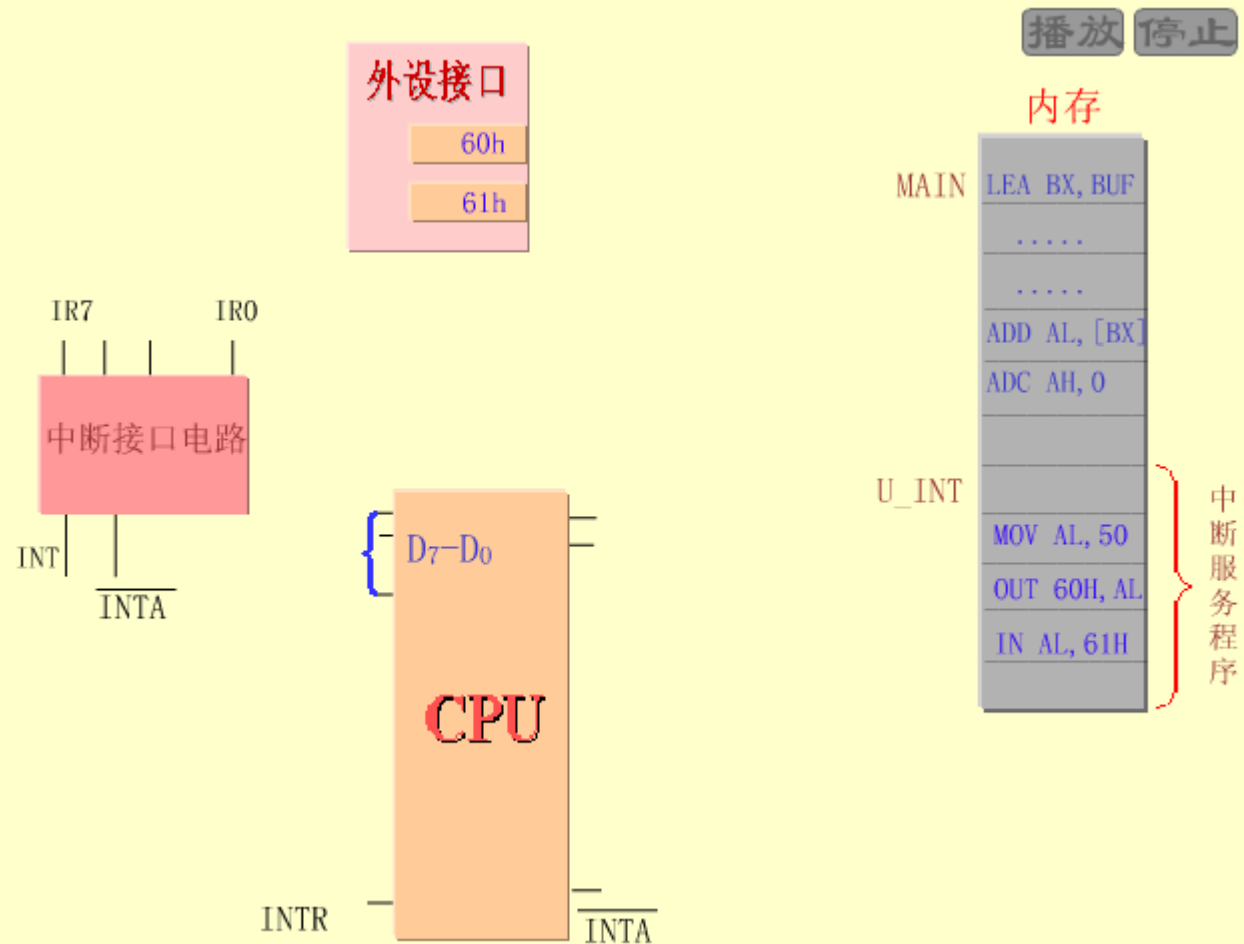
Start: mov dl, 0f8h    ; 1111,1000 启动A/D转换
      lea di, detor    ; 输入数据存放地址
Again: mov al, dl      ;
      and al, 0efh     ; ^ 1110,1111, AL=1110,1000
      out [4], al      ; 停止A/D转换, 并选择模拟量A0
      call delay       ;
      mov al, dl       ; AL=1111,1000
      out [4], al      ; 启动A/D转换
    
```

```

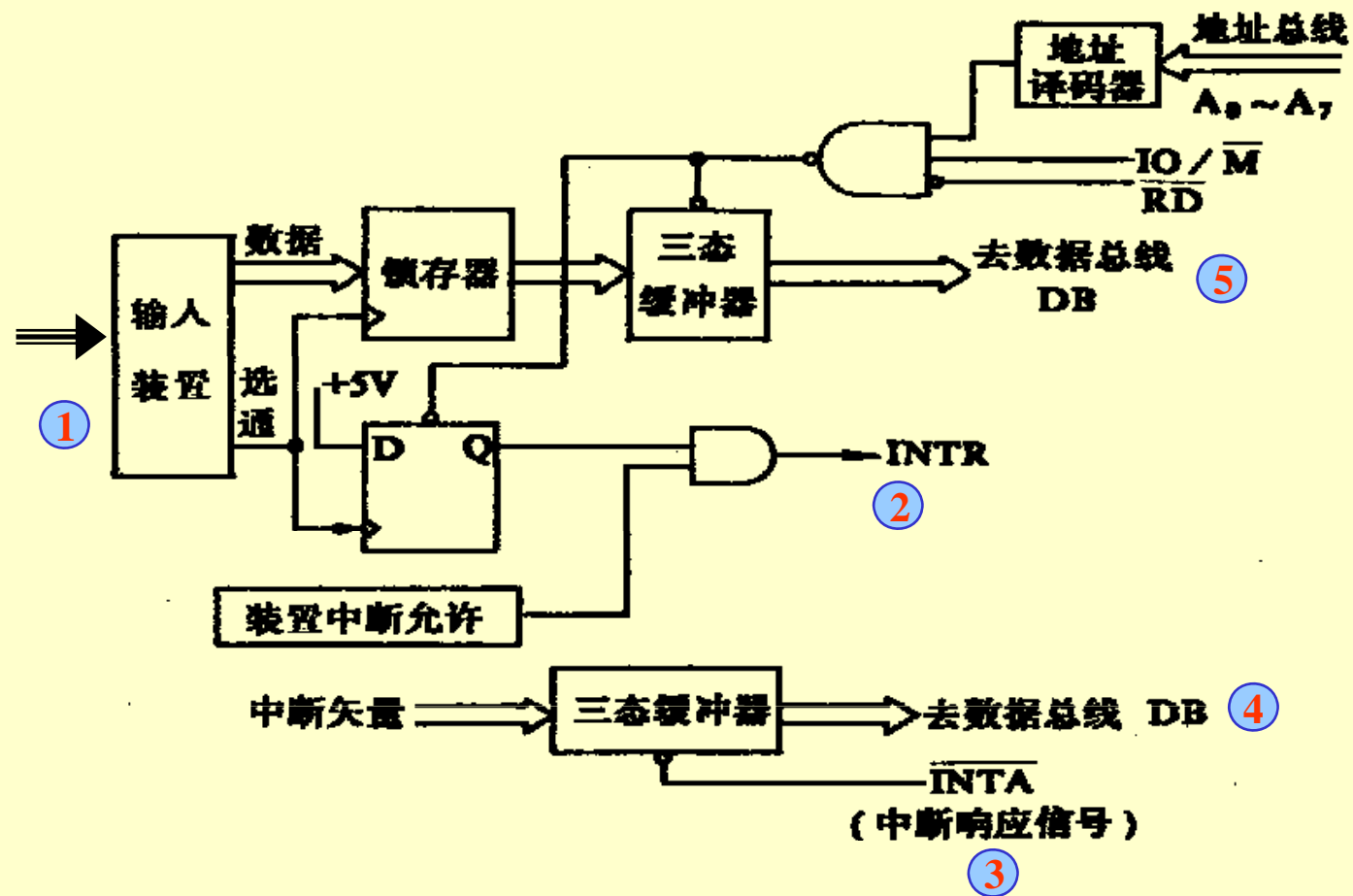
poll: in al, [2]      ; 输入状态信息
      shr al, 1       ;
      jnc poll        ; 若未Ready,等待
      in al, [3]      ; 否则, 输入数据
      stosb           ; 将数据存入内存
      inc dl          ; 取下一个模拟量
      jne again       ;
    
```

2. 中断传送方式

传送流程演示

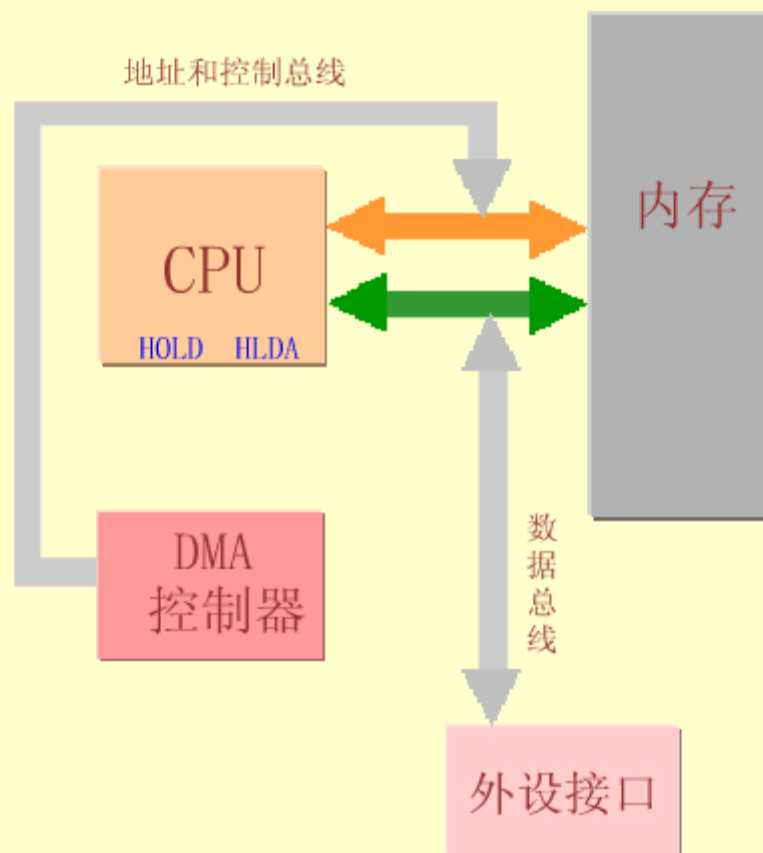


中断传送方式的接口电路



3. 直接数据通道传送 (DMA)

传送流程演示

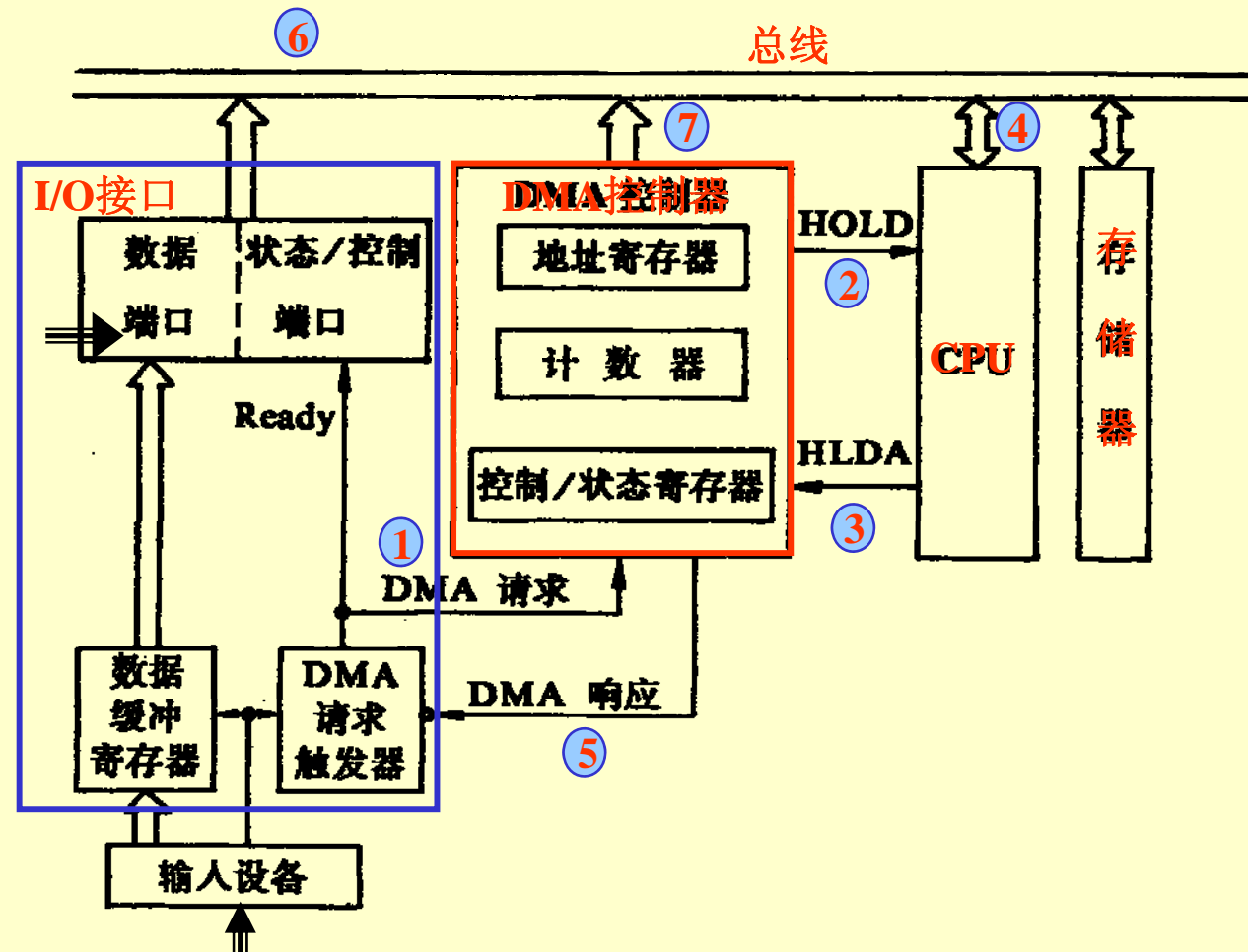
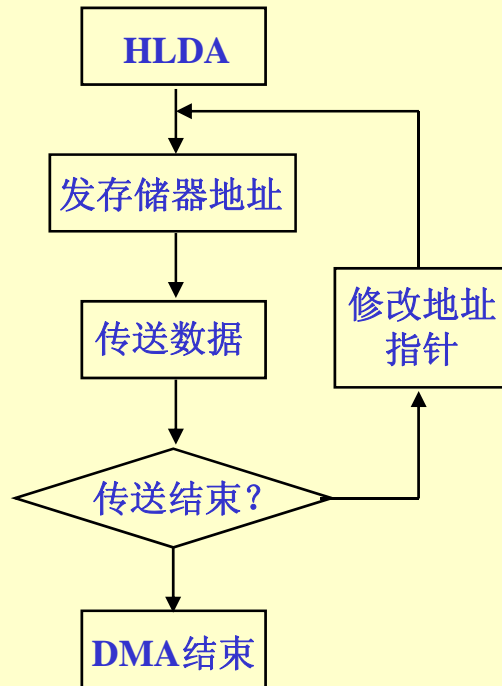


说明:

播放 停止



DMA传送方式的接口电路



四、几种传送方式的比较：

1. 无条件传送： 只能用于外部设备与CPU 的动作同步时，否则出错。这种方式已 较少使用。

2. 查询传送：

- 接口简单，
- 但在传送过程中，若外设数据没有准备好，则CPU一直在查询、等待，而不能做其他事情。CPU的效率低下。

3. 中断传送：

- 只有当外设数据准备好时(向CPU发出请求)， CPU才进行数据传送（在中断服务程序中），其余时间CPU可以做其他事情。CPU效率大大提高。
- 但是，每传送一次数据，CPU都要执行一次中断服务程序，在中断服务程序中，除执行 **IN** 和 **OUT** 指令外，还要进行下列工作：
保护断点、保护标志寄存器、保护某些通用寄存、恢复等一些工作，
95%的时间是额外开销，从而传送效率并不高。

4. DMA传送： 在DMAC的控制下，外设直接和存储器（也可外设与外设，存储器与存储器之间）进行数据传送，而不必经过CPU，传送速度基本取决于外设与存储器的速度，从而传送效率大大提高。

