

第6章 微机和外设的数据传输

■ 为什么要用接口电路

● 什么叫I/O接口

CPU和外部设备之间的数据传输叫输入/输出（I/O）。为了达到数据传送的目的，把CPU与外部设备连接起来的硬件电路称为I/O接口电路。

● 为什么要用接口电路

- ✓ 速度差异
- ✓ 时序不匹配
- ✓ 信息格式不匹配
- ✓ 信息类型不匹配

■ CPU和输入/输出设备之间的信号

1. 数据信息

数字量：二进制表示的数据。如：01111101

模拟量：随时间连续变化的量。如：0----5V、0----10mA

开关量：二种状态变化的量。如：开关的开与关。

2. 状态信息

反映外设当前所处的状态。如：打印机是否有纸，是否正在打印。

3. 控制信息

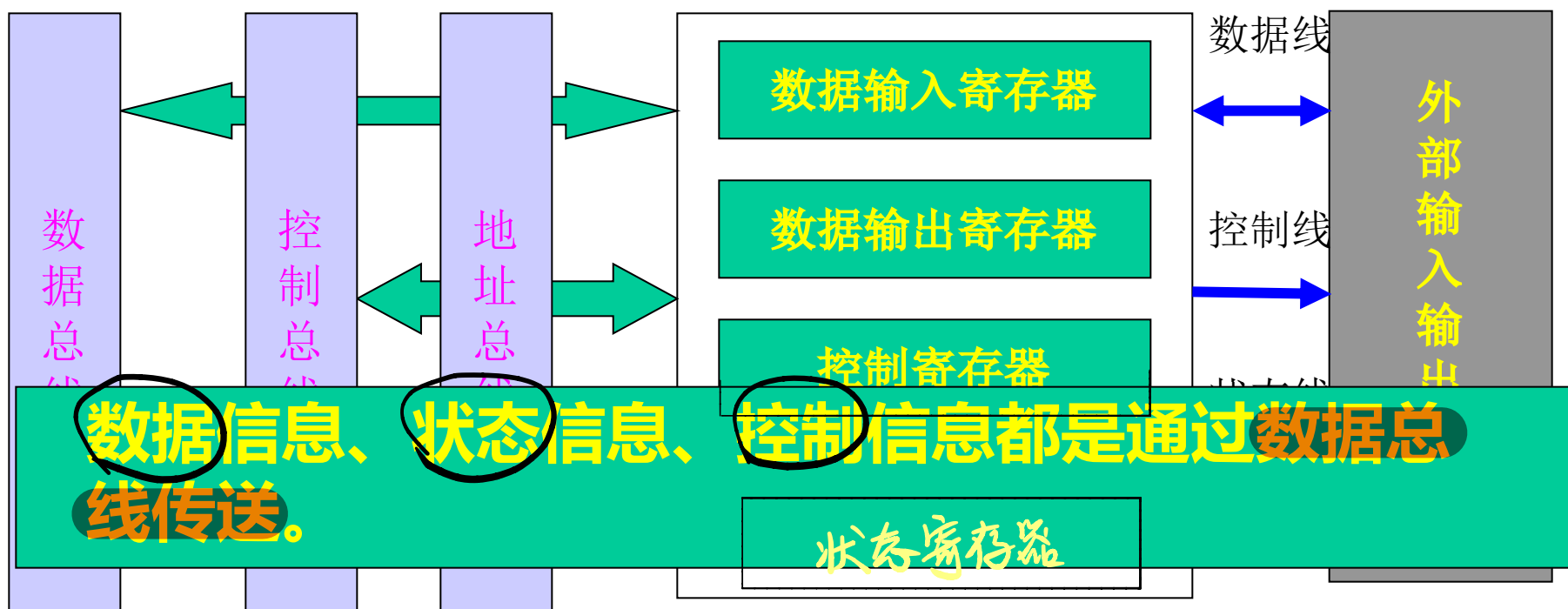
用于控制外设工作。如：使打印机走纸、换页。

■ 接口部件的I/O端口和寻址方式

● 端口

每个接口部件都包含一组寄存器，这些寄存器一般称端口。每个端口都对应有端口地址。

- 数据端口 用于存放接口部件的数据信息。
- 状态端口 用于存放反映接口部件的状态信息。
- 命令端口 用于存放CPU送给接口部件的命令信息。



● 端口寻址

1) 存储器对应的I/O端口

统一

外设作为存储器的一个单元来对待，故每一个外设中的寄存器都占用一个存储器的地址，对I/O的存取操作与对存储器的操作一样。

- ✓ CPU对I/O的操作可使用全部的存储器操作指令；
- ✓ 内存和外设的地址分布图是同一个；
- ✓ 不需要专门的I/O指令；
- ✓ 外设占用了内存地址，使内存容量减少；
- ✓ 需要较长的地址，指令长度加长，从而增长了指令的执行时间。

2) 独立编址的I/O 端口

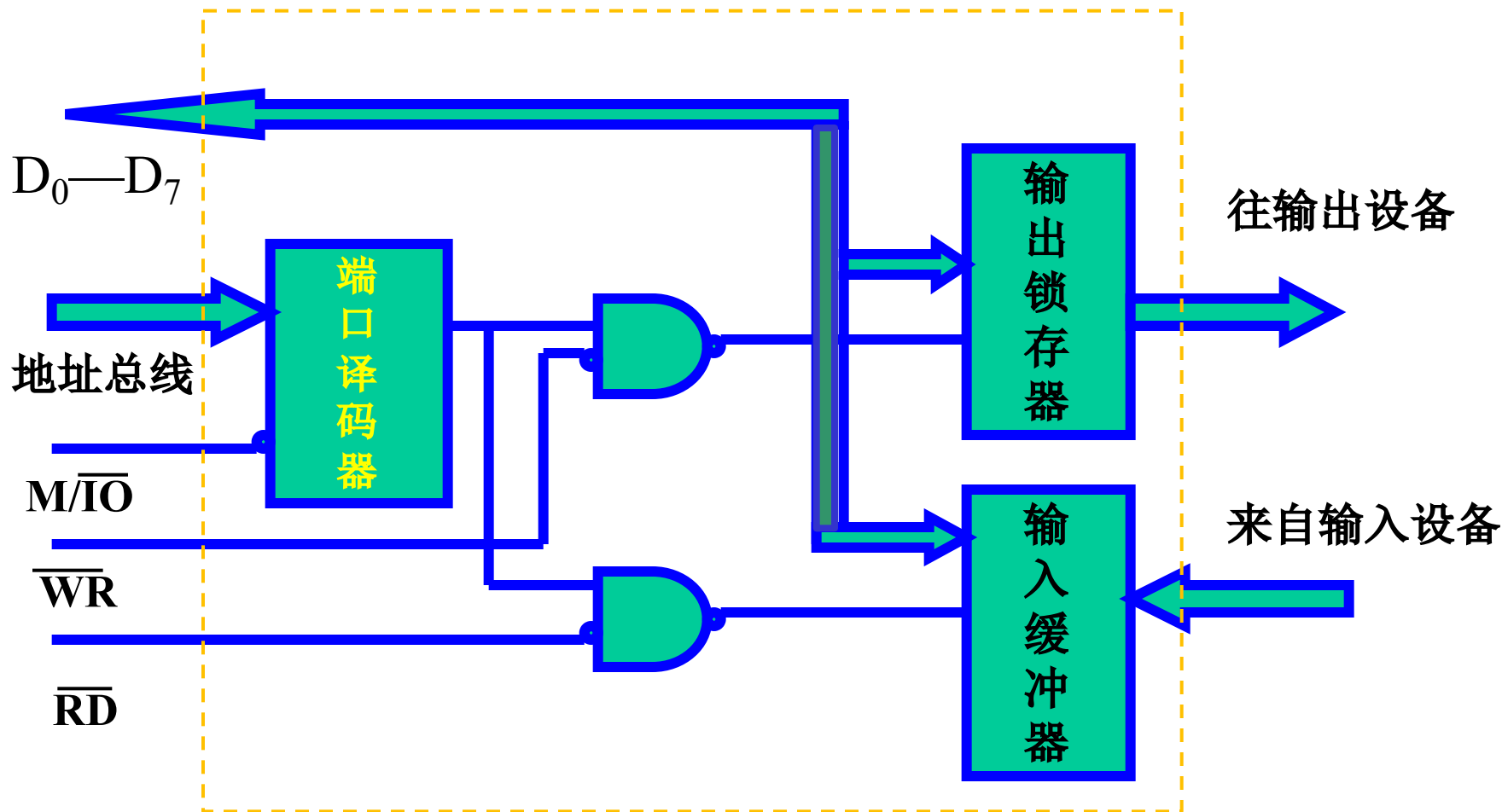
CPU有专门的I/O指令。

- ✓ CPU对I/O的操作使用专门的I/O操作指令；
- ✓ 内存和外设分别有自己的地址分布图；
- ✓ 因有专门的I/O指令，所以指令长度短执行时间快。

■ CPU和I/O间的数据传送方式

1 无条件传送方式

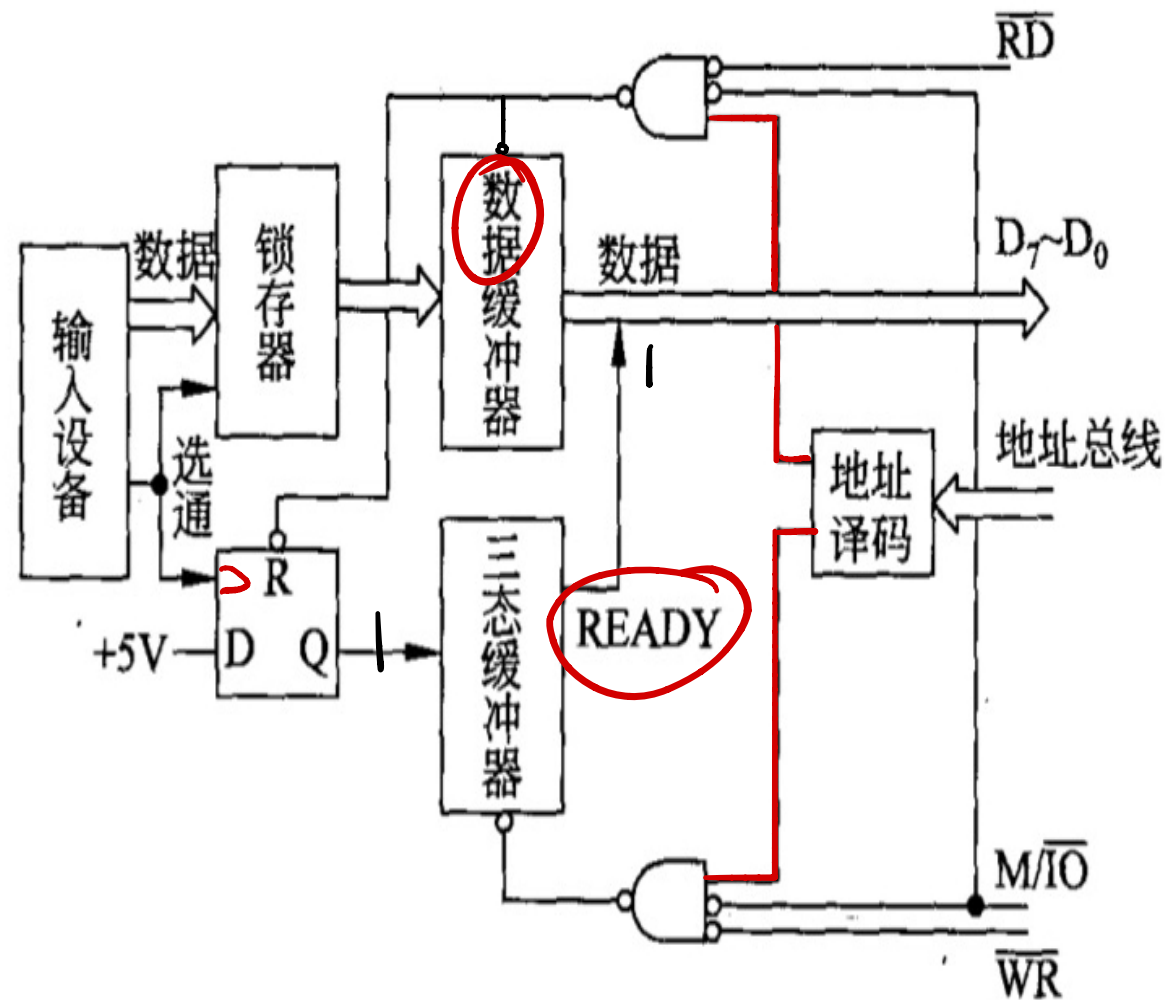
特点：不进行任何查询，发送端只管送，接收端只管收；
要求发送端和接收端严格同步。



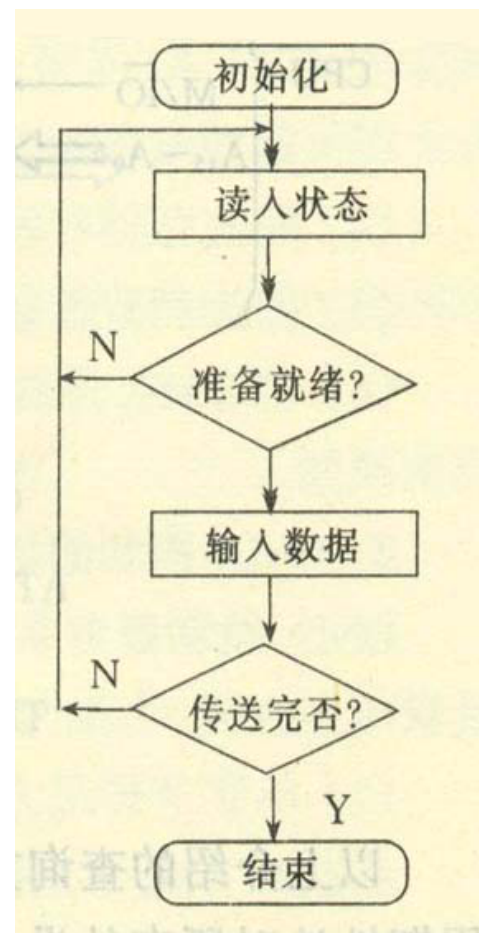
2 条件传送方式（查询传送方式）

特点：每传送（或接收）一个数据前都要进行一次状态查询，只有在条件满足时才进行数据传送。

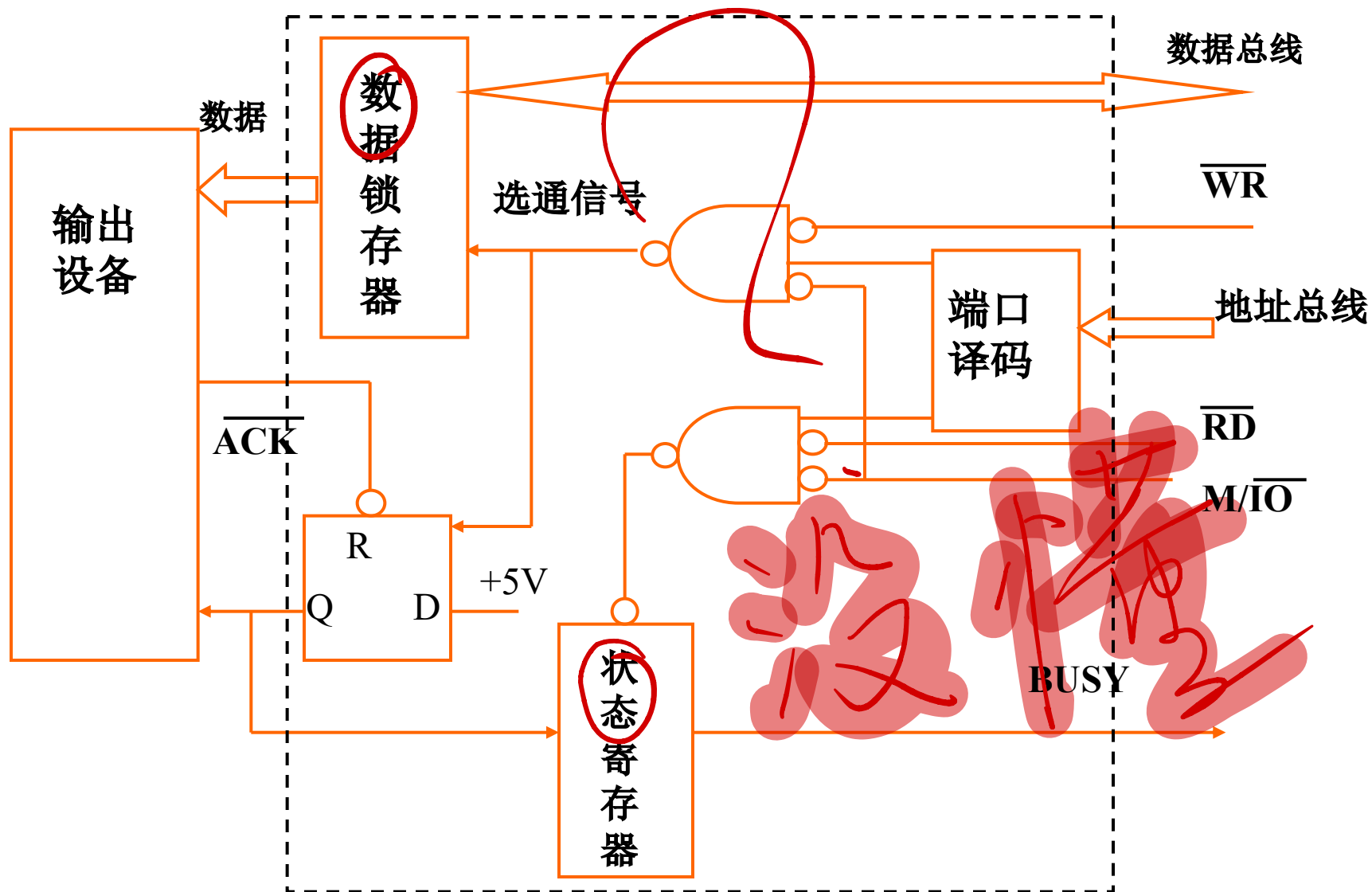
1) 查询式输入



CPU



2) 查询式输出



条件传送方式例1:

假设从键盘往缓冲区输入1个字符行，遇到回车符（ODH）或字符行超过80个字符时，输入便结束，并自动加上1个换行符（OAH）。如在输入的81个字符中未见到回车符，则在终端上输出信息“**BUFFER OVERFLOW**”

数据输入端口:52H；状态端口:56H；输入状态信息连数据D1；数据输出端口:54H；输出状态信息连数据D0。利用最高位偶校验，校验出错输出错误信息；校验正确，清除校验位，传送数据。

告诉寄存器地址

```
DATA_SEG SEGMENT
    MESSAGE DB 'BUFFER OVERFLOW', 0DH, 0AH
    ...
DATA_SEG ENDS

COM_SEG SEGMENT
    BUFFER DB 82 DUP(?)           ; 接收缓冲区
    COUNT DW ?                   ; 计数器
COM_SEG ENDS
```

```
...
CODE SEGMENT
ASSUME DS: DATA_SEG, ES: COM_SEG, CS: CODE
STAT: MOV AX, DATA_SEG
      MOV DS, AX
      MOV AX, COM_SEG
      MOV ES, AX
      MOV DI, OFFSET BUFFER      ; 计数器指向缓冲区首址
      MOV COUNT, DI
      MOV CX, 81                 ; 字符行长度
      CLD                       ; 清方向标志
```

串指令


```

NEXT_IN:  IN      AL, 56H      ; 读状态端口
          TEST    AL, 02H 0000 0010 ; 测试D1位
          JZ      NEXT_IN     ; 未准备好, 等待, 再测
          IN      AL, 52H     ; 准备好, 输入字符
          OR      AL, 0       ; 偶校验
          JPE     NO_ERROR
          JMP     ERROR

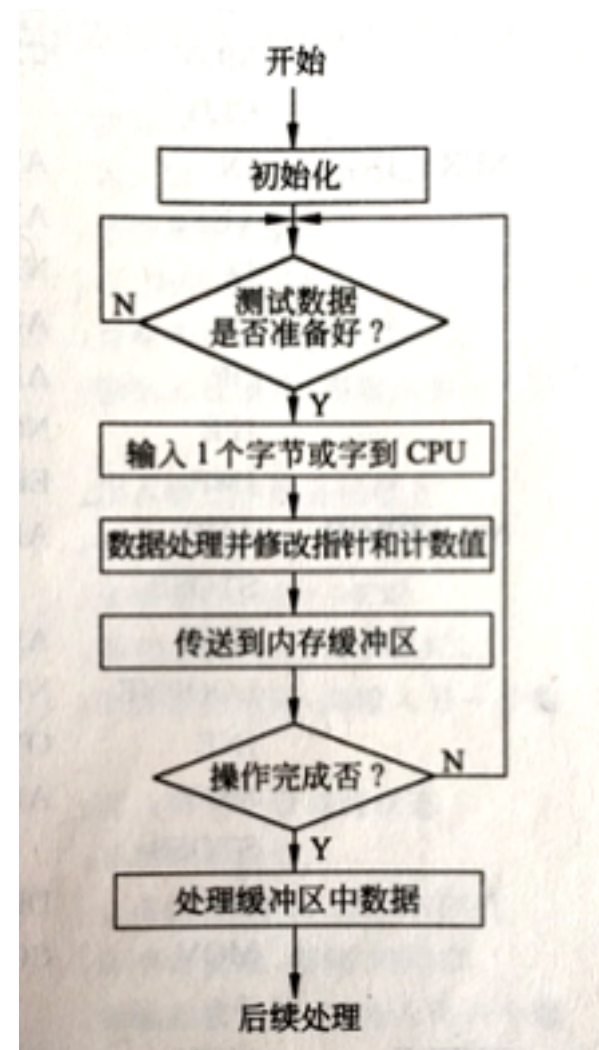
NO_ERROR: AND    AL, 7FH      ; 清校验位
          STOSB             ; 字符送缓冲区
          CMP     AL, 0DH     ; 是否为回车符
          LOOPNE  NEXT_IN    ; 不是回车, 再输入
          JNE     OVERF      ; 不是回车且溢出, 转
          MOV     AL, 0AH     ; 加换行符
          STOSB
          SUB     DI, COUNT   ; 计算输入字符数
          MOV     COUNT, DI

...
OVERF:    MOV     SI, OFFSET MESSAGE
          MOV     CX, 17

NEXT_OUT: IN      AL, 56H
          TEST    AL, 01H 1测试D0位
          JZ      NEXT_OUT
          LODSB
          OUT     54H, AL
          LOOP    NEXT_OUT

ERROR:    ...

```



多设备实现条件传送方式例1:

● 多设备轮流查询（有优先级）

```
TREE_IN: MOV    FLAG, 0
INPUT:   IN      AL, STAT1 状态口
        TEST    AL, 20H 0010 0000
        JZ      DEV2
        CALL    PROC1子程序
        CMP     FLAG, 1
        JNZ     INPUT
DEV2:    IN      AL, STAT2
        TEST    AL, 10H 0001 0000
        JZ      DEV3
        CALL    PROC2子程序
        CMP     FLAG, 1
        JNZ     INPUT
DEV3:    IN      AL, STAT3
        TEST    AL, 08H 0000 1000
        JZ      NO_INPUT
        CALL    PROC3子程序
NO_INPUT: CMP     FLAG, 1
        JNZ     INPUT
...
```

；清除标志
；读入第一个设备的状态
；是否准备就绪
；否，则转DEV2
；准备就绪，调用PROC
；如标志被清除，则输入另一个数

- FLAG标志的设置可使得设备具有优先级别；
- STAT1设备优先级最高，STAT2次之，STAT3最低；

多设备实现条件传送方式例2:

- 多设备轮流查询（无优先级，循环查询）

TREE_IN:	MOV	FLAG, 0	; 清除标志
INPUT:	IN	AL, STAT1	; 读入第一个设备的状态
	TEST	AL, 20H	; 是否准备就绪
	JZ	DEV2	; 否, 则转DEV2
	CALL	PROC1	; 准备就绪, 调用PROC
DEV2:	IN	AL, STAT2	
	TEST	AL, 10H	
	JZ	DEV3	
	CALL	PROC2	
DEV3:	IN	AL, STAT3	
	TEST	AL, 08H	
	JZ	NO_INPUT	
	CALL	PROC3	
NO_INPUT:	CMP	FLAG, 1	
	JNZ	INPUT	

■ 程序控制传送方式的特点

优点：

- 接口简单

缺点：

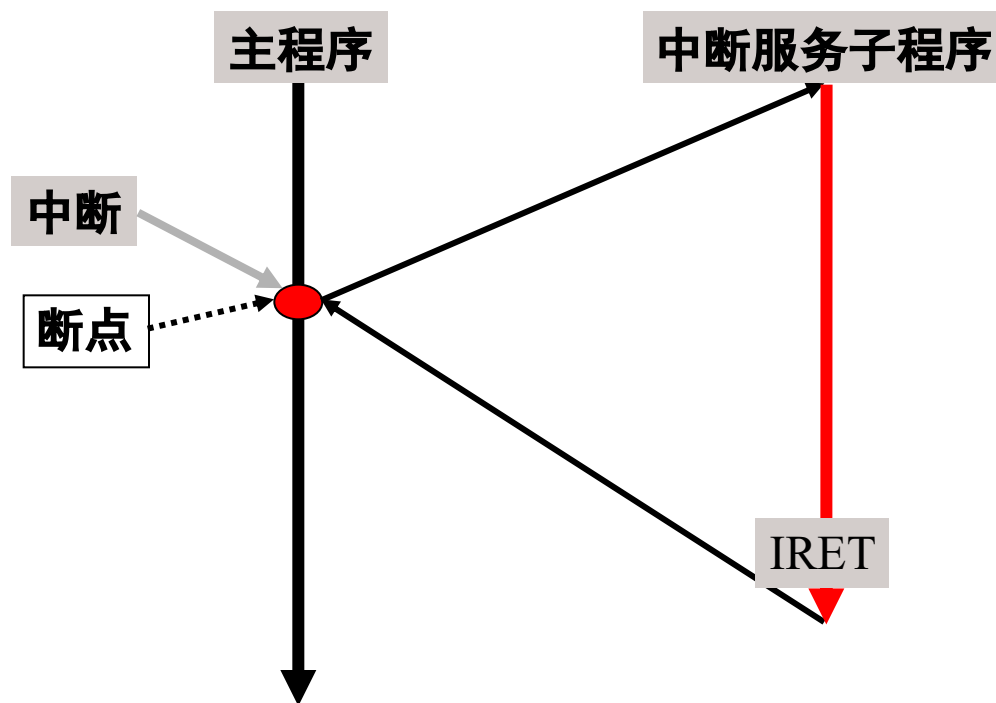
- CPU需要花大量的时间去读状态字，效率低
- 实时性差

- 适用于外设不多，实时性要求不高的系统

3 中断方式

- 中断传送方式的原理

当外围设备准备好向CPU传送的数据或者外设已准备就绪接收CPU的数据，就向CPU发中断请求，或者计算机系统有异常事故要求CPU处理，CPU暂停原程序的执行，启动中断服务程序，转去中断处理，处理完毕后，又返回原来的断点，继续执行原来的程序



特点：实时性好，CPU利用率高。

● 8086CPU的中断方式

1) 中断操作和中断系统

(1) 8086的中断分类

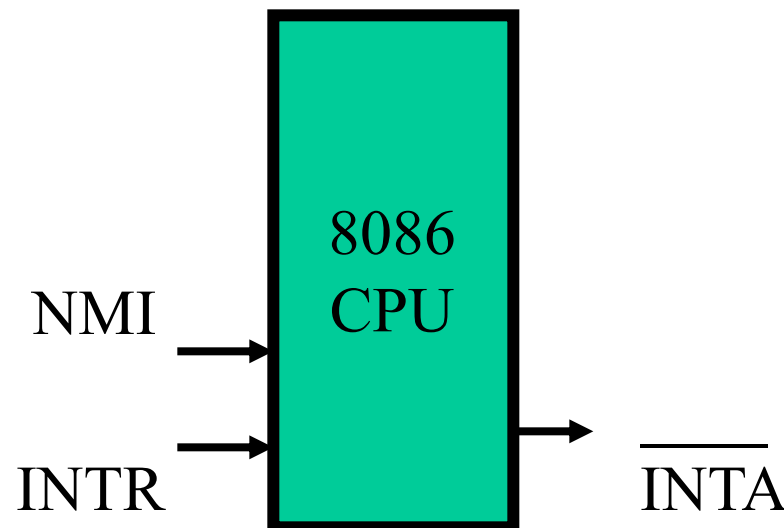
A 硬件中断(外部中断)

•NMI 不可屏蔽中断
上升沿触发;
类型码=2

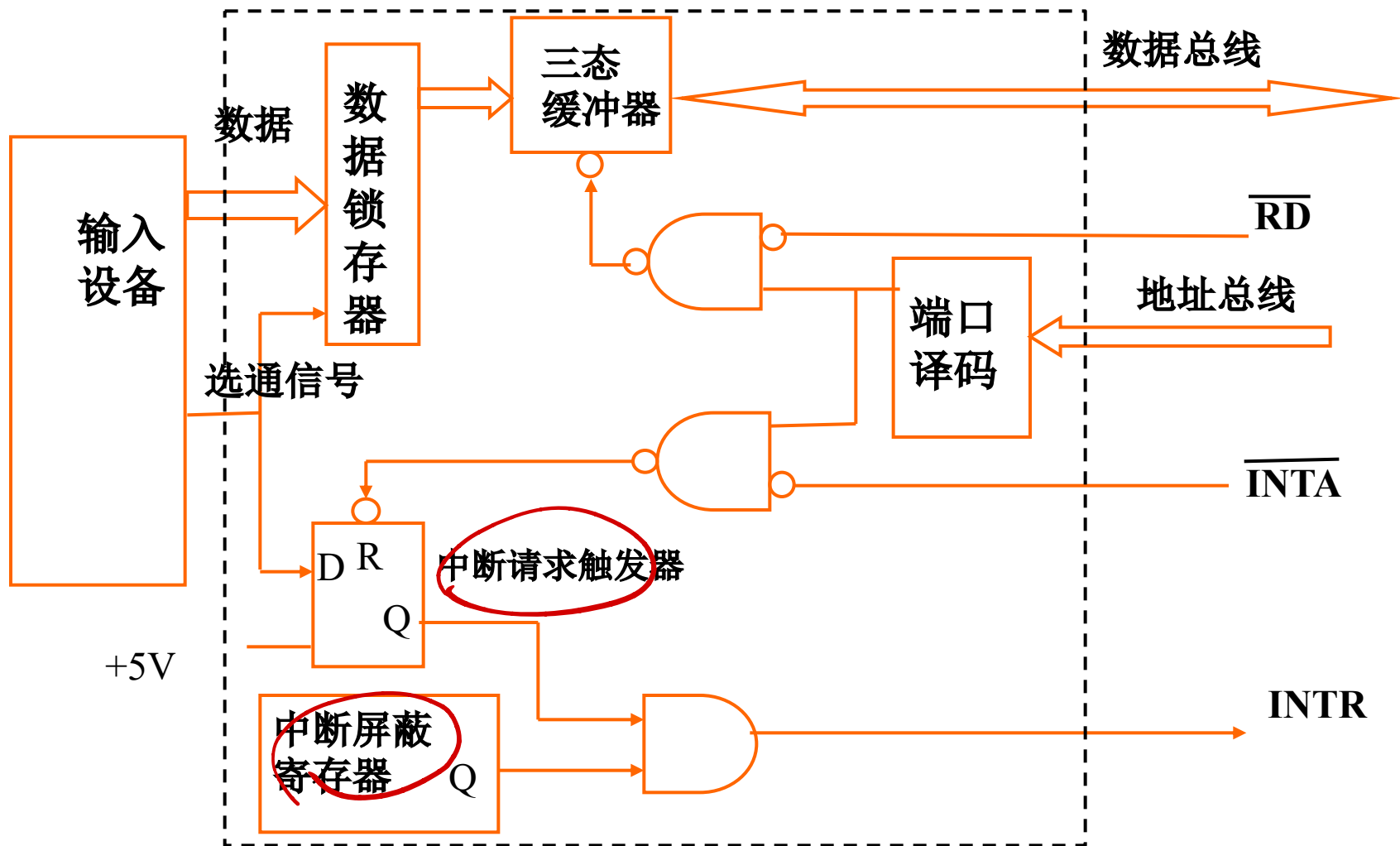
•INTR 可屏蔽中断
高电平有效;
IF=1时才响应中断;
类型码由申请中断的外设给CPU

B 软件中断(内部中断)

以INT指令或由CPU的某些运算错误产生;
类型码由INT指令直接给出或事先设定;
进入中断时,不需要执行中断响应总线周期.



中断接口



■ 中断传送方式的特点

- 中断传送是一种比查询传送效率更高的程序传送方式
- 数据传送的中断服务程序需预先设计好
- 中断请求由外设随机向CPU提出
- CPU对请求的检测是有规律的：一般是在每一个指令的最后一个时钟周期，采样中断请求输入引脚
- 但是
 - 每传送一次数据，就要中断一次
 - 不适用大数据量传送

4 DMA方式

- DMA方式的提出

- 希望克服程序控制传送的不足：

外设->CPU->存储器

外设<- CPU<-存储器

- 直接存储器存取DMA：

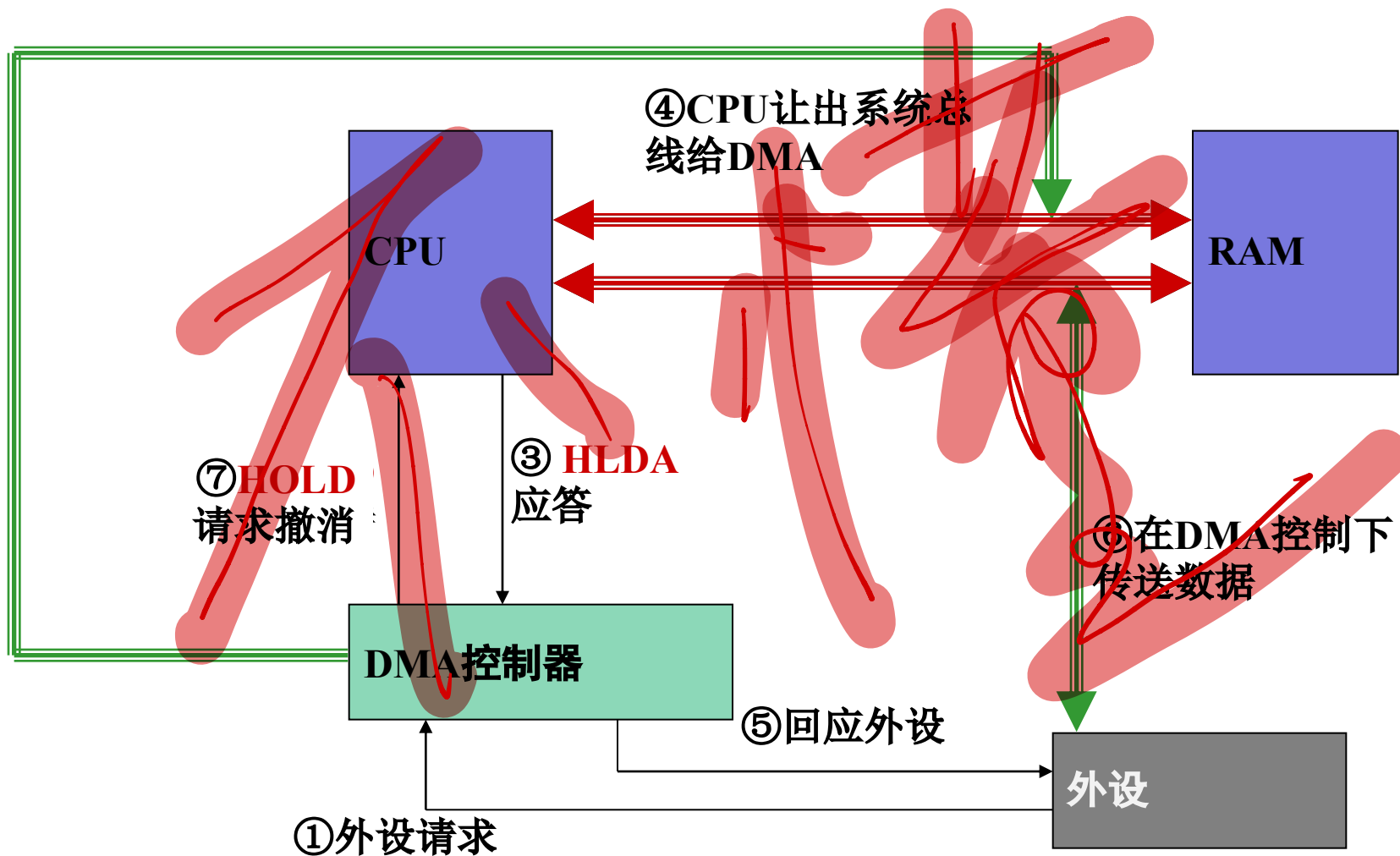
外设->存储器

外设<-存储器

- CPU释放总线，I/O操作完全由**DMA控制器**管理

● DMA传送原理

整个数据传送过程无程序参与,由DMA控制传送



● DMA控制器结构



■ DMA传送方式的特点

- 数据交换不通过CPU
- 专门的接口电路直接控制I/O设备与存储器交换数据
- 适用于高速度大批量数据传送，如磁盘存取、图像处理等
- 3种形式
 - 存储器与I/O设备
 - 存储器与存储器
 - I/O设备与I/O设备