
上海交通大学
网络学院

微机原理与应用

The Principle & Application of Microcomputer

王春香　副教授

wangcx@sjtu.edu.cn

第六章 输入输出与中断

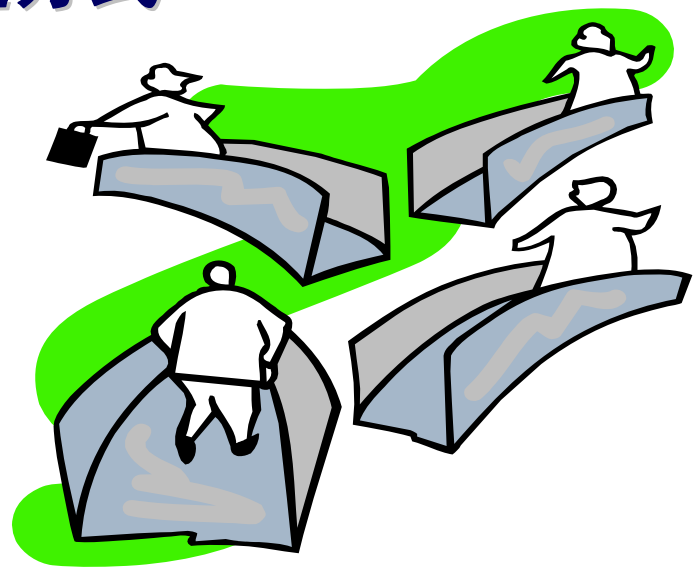
主要内容

6.1 输入输出接口概述

6.2 CPU与外设之间数据传送方式

6.3 中断技术

6.4 8086/8088中断系统和中断处理



第六章 输入输出与中断

学习要求

- 着重理解接口基本结构的特点。
- 掌握CPU与外设之间数据的传送方式与控制方式。
- 正确理解中断源、向量中断、中断优先权等基本概念。
- 重点掌握8086/8088中断系统及其用户定义的内部中断处理方法。
- 正确理解和灵活运用中断向量表。

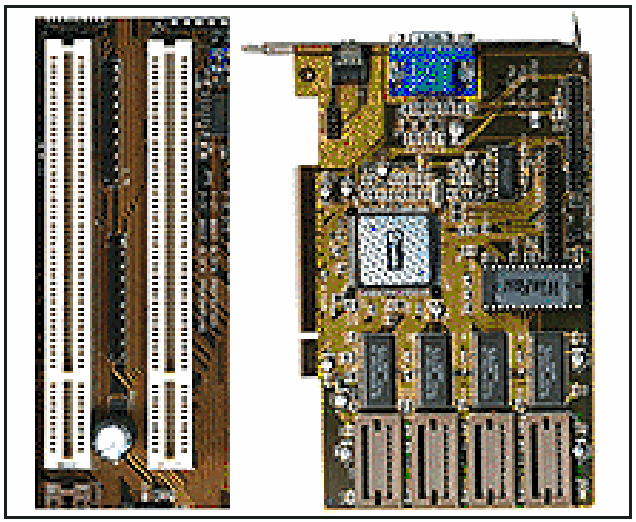
6.1 输入输出接口概述

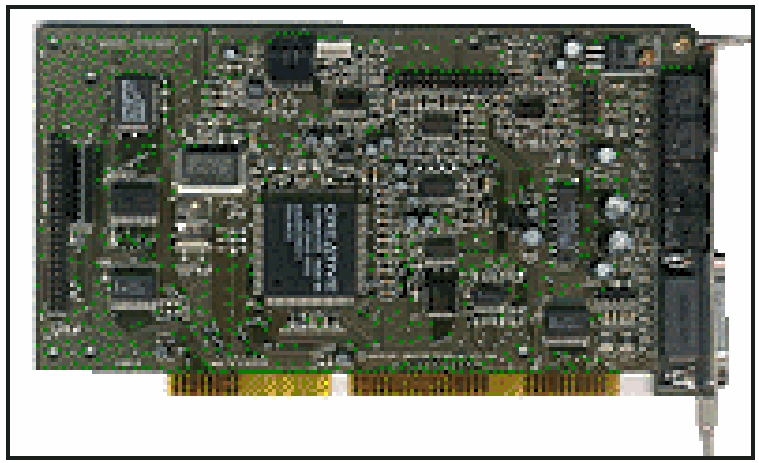
6.1.1 CPU与外设的连接

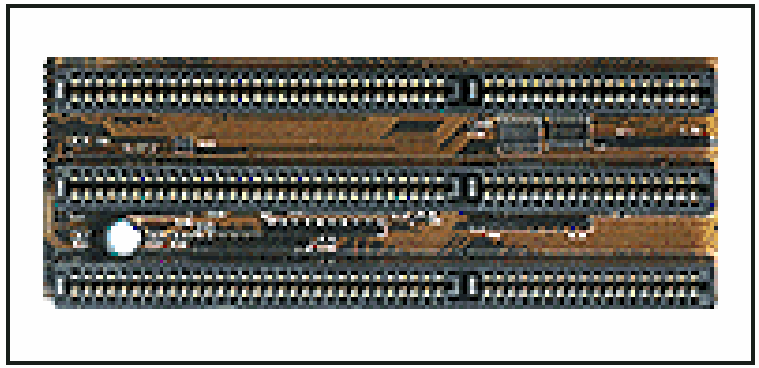
- 外设种类繁多：机械式、电动式、电子式、电磁式
- 信号类型复杂：数字量、模拟量、开关量
- 处理信息速率相差甚远：如手动键盘输入和磁盘输入
- 外设数据传递方式：并行，串行

接口电路(芯片)：CPU与外部设备之间实现信息交换的连接电路(硬件)，简称接口。

6.1 输入输出接口概述







6.1 输入输出接口概述

❋ 接口用途小结

① 进行地址译码或设备选择，以便使CPU能与某一指定外部设备通讯；

② 状态信息的应答，以协调数据传送之前的准备工作；

③ 进行中断管理，提供中断信号；

④ 进行数据格式转换，如正负逻辑转换，串行与并行数据转换等；

⑤ 进行电平转换，如TTL电平与MOS电平间的转换；

⑥ 协调速度，如采用锁存、缓冲、驱动等；

⑦ 时序控制，提供实时时钟信号。

6.1 输入输出接口概述

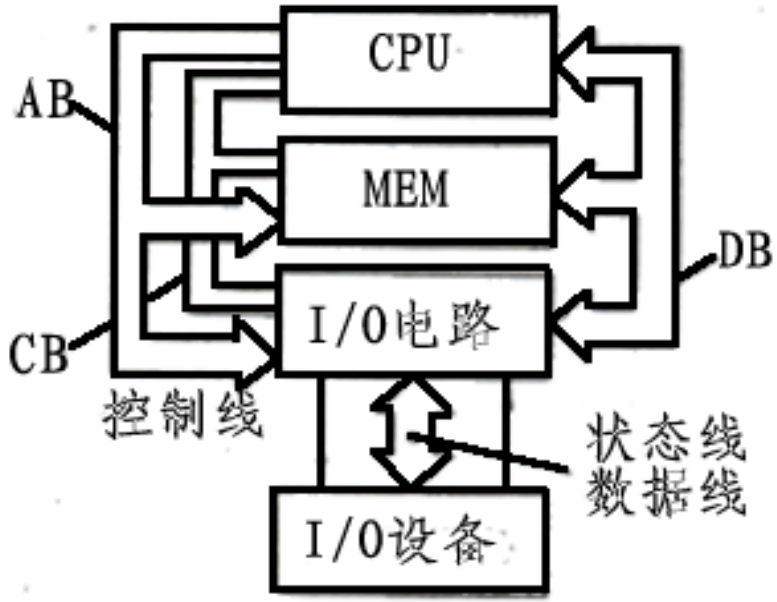
6.1.1 CPU与外设的连接

CPU对外设的输入输出操作类似于存储器的读写操作，即I/O读写，但外设与存储器有诸多不同。

	存储器	I/O设备
不同点	品种有限	品种繁多
	功能单一	功能多样
	传送一个字节	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

6.1 输入输出接口概述

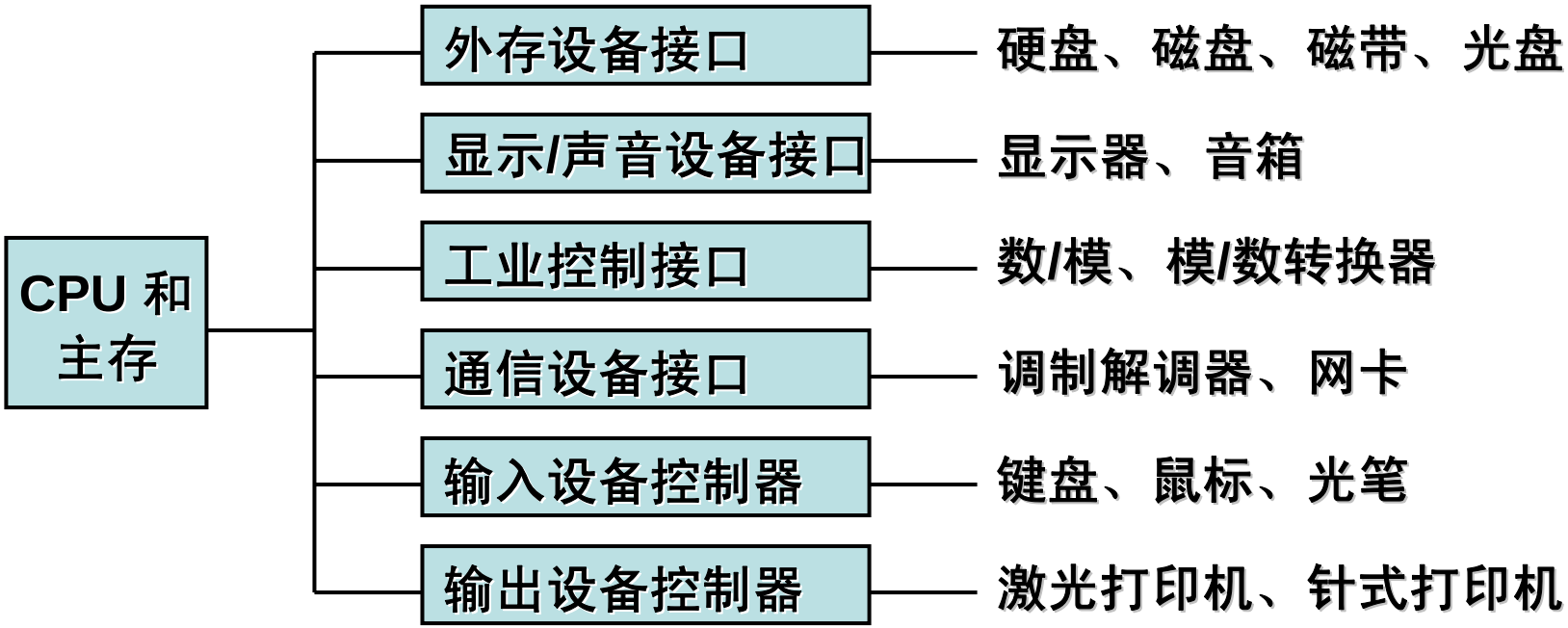
6.1.1 CPU与外设的连接



外设与计算机的连接不能像存储器那样直接挂到总线(DB、AB、CB)上，必须通过各自的专用接口电路 (接口芯片) 与主机连接。

6.1 输入输出接口概述

6.1.1 CPU与外设的连接



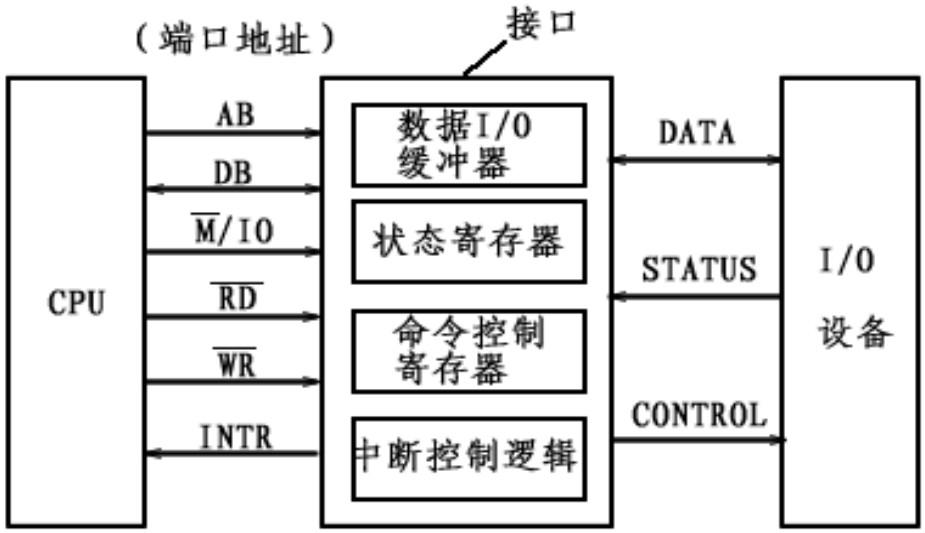
计算机 I/O 系统结构图

6.1 输入输出接口概述

6.1.2 接口电路的基本结构

接口电路基本结构同它传送的信息种类有关。

信息可分为3类：数据信息，状态信息，控制信息。



6.1 输入输出接口概述

6.1.2 接口电路的基本结构

① 3种性质不同信息，经不同端口分别传送。每个端口都有自己的端口地址，用不同的端口地址来区分不同的信息。

② 用输入输出指令来寻址外设时，外设状态作为一种输入数据，而CPU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位或低16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。

6.1 输入输出接口概述

6.1.2 接口电路的基本结构

1. 数据信息

① **数字量：**由键盘、磁盘机、磁带机、卡片机等读入的信息，或主机送给打印机、磁盘机、磁带机、显示器及绘图仪的信息。

通常为8位二进制数或ASCII代码。

6.1 输入输出接口概述

6.1.2 接口电路的基本结构

1. 数据信息

② **模拟量：**计算机用于检测、数据采集或控制时，现场信息是连续变化的物理量 (如温度、压力、位移等)，经传感器把非电量转换成电量，经放大得到模拟电流或电压。

计算机不能直接接收和处理模拟量，须经A/D (模/数) 转换，才能输入计算机。

计算机输出的数字量也须经D/A (数/模) 转换后才能去控制执行机构。

6.1 输入输出接口概述

6.1.2 接口电路的基本结构

1. 数据信息

③ **开关量：**两个状态，如开关的闭合/断开，电机的运转/停止，阀门的打开/关闭等。

用一位“0”或“1”二进制数表示。

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。

6.1 输入输出接口概述

6.1.2 接口电路的基本结构

2. 状态信息

外设当前所处工作状态信息，CPU与外设间可靠交换数据条件。

输入时：告知CPU有关输入设备数据是否准备好 (Ready=1?)

输出时：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。

6.1 输入输出接口概述

6.1.2 接口电路的基本结构

3. 控制信息

用于控制外设的启动或停止。

6.2 CPU与外设之间数据传送方式

- 直接程序控制方式
  - ✓ 无条件程控传送方式

    仅传输数据信息，不传输控制、状态信息
  - ✓ 有条件程控传送方式（查询方式）

    要传输数据、控制、状态信息
- 中断控制方式
- 直接存储器存取（DMA）控制方式

6.2 CPU与外设之间数据传送方式

6.2.1 程序传送

CPU与外设间的数据交换在程序控制下进行。

即IN或OUT指令控制。

- ✓ 无条件程序控制传送方式
- ✓ 有条件程序控制传送方式（查询方式）

6.2 CPU与外设之间数据传送方式

6.2.1 程序传送

1. 无条件传送（又称同步传送）

最简单的输入/输出控制方式，用于控制CPU与低速接口之间的信息交换。

例如，开关、继电器、7段显示器、机械式传感器等简单外设。

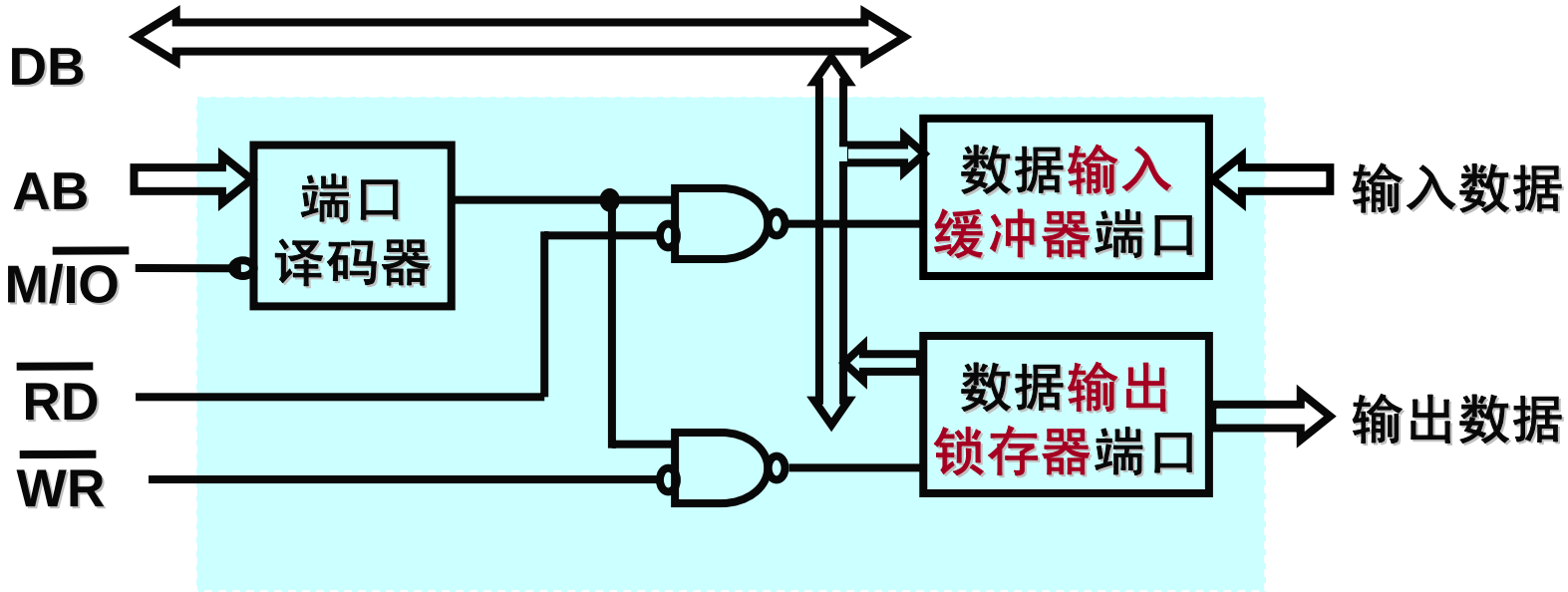
这类信号变化缓慢，当需要采集这些数据时，外设已将数据准备就绪了。无需检查端口的状态，就可立即采集数据。

对少量数据传送来说，它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。

6.2 CPU与外设之间数据传送方式

6.2.1 程序传送

1. 无条件传送（续1）



注：输入接口为缓冲器，输出接口为锁存器

6.2 CPU与外设之间数据传送方式

6.2.1 程序传送

1. 无条件传送（续2）

输入数据时，由于数据保持时间相对于CPU的处理时间长得多，故输入端可直接用输入缓冲器与CPU的数据总线相连。

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，锁存允许端$\overline{CE}$=1(为无效电平)时，数据总线上的新数据不能进入锁存器。

只有当确知外设已取走CPU上次送入锁存器的数据，方能在$\overline{CE}$=0 (为有效电平) 时将新数据再送入锁存器保留。

6.2 CPU与外设之间数据传送方式

6.2.1 程序传送

1. 无条件传送（续3）

输入时，假定来自外设的数据已输入至三态缓冲器；

当CPU执行IN指令时，所指定的端口地址经地址总线的低16位或低8位送至地址译码器，CPU进入了输入周期；

选中的地址信号和M/$\overline{IO}$（以及$\overline{RD}$）相“与”后，去选通输入三态缓冲器，把外设的数据与数据总线连通并读入CPU。

当CPU执行IN指令时，外设的数据是已准备好的，否则就读错。

6.2 CPU与外设之间数据传送方式

6.2.1 程序传送

1. 无条件传送（续4）

输出时，假定CPU的输出信息经数据总线已送到输出锁存器的输入端；

CPU执行OUT指令时，端口地址由地址总线的低 8 位地址送至地址译码器，CPU进入了输出周期；

所选中地址信号和M/$\overline{IO}$（以及$\overline{WR}$ 信号）相“与”后，去选通锁存器，把输出信息送至锁存器保留，由它把信息通过外设输出。

CPU执行OUT指令时，必须确信所选外设的锁存器是空的。

6.2 CPU与外设之间数据传送方式

6.2.1 程序传送

1. 无条件传送（实例）

16位数据采集系统，被采集数据是8个模拟量，由继电器绕组P0、P1… P7分别控制触点K0、K1…K7逐个接通。

每次采样用一个4位 (每位为一个十进制数) 数字电压表测量，把被采样的模拟量转换成16位BCD代码(即对应4位十进制数的4个BCD码)，高8位和低8位通过两个不同的端口 (其地址分别为10H和11H)输入。

CPU通过端口20H 输出控制信号,以控制某个继电器的吸合,实现采集不同通道的模拟量。

6.2 CPU与外设之间数据传送方式

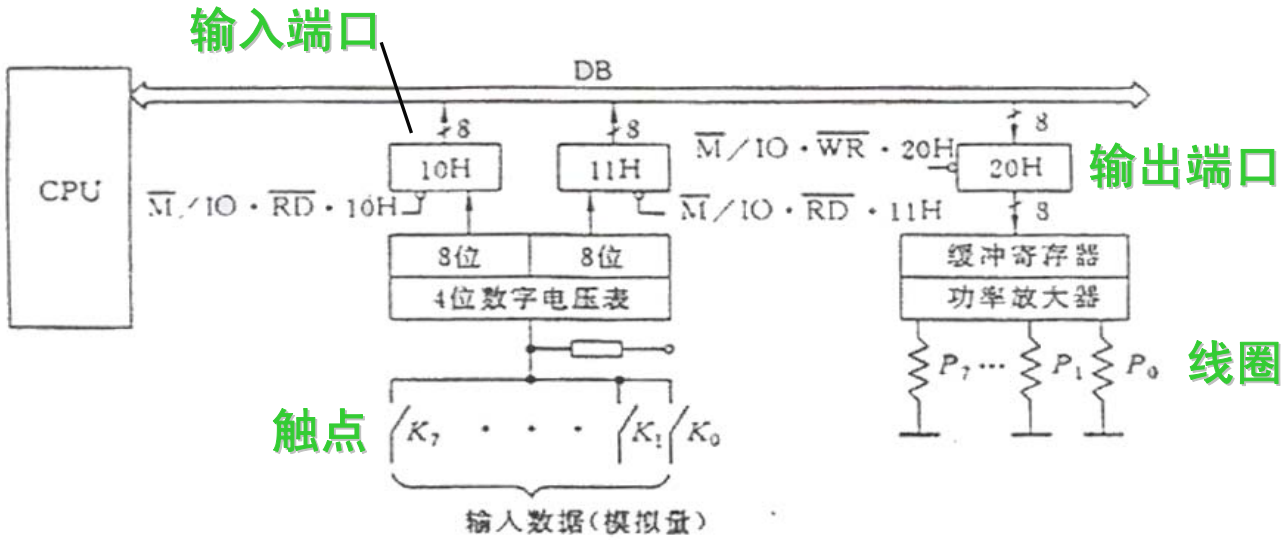
采集过程要求：

① 先断开所有继电器线圈及触点，不采集数据。

② 延迟一段时间后，K0闭合，采集第1个通道模拟量，保持一段时间，使数字电压表能将模拟电压转换为16位BCD码。

③ 分别将高8位与低8位BCD码存入内存，完成第1个模拟量输入与转存。

④ 利用移位与循环实现8个模拟量的依次采集、输入与转存。



6.2 CPU与外设之间数据传送方式

```
START:  MOV  DX, 0100H        ; 01H→DH，置吸合第1个继电器代码
                              ; 00H→DL，置断开所有继电器代码
        LEA  BX, DSTOR        ; 置输入数据缓冲器的地址指针
        XOR  AL, AL           ; 清AL及进位位CF
AGAIN:  MOV  AL, DL
        OUT  20H, AL          ;断开所有继电器线圈
        CALL NEAR DELAY1      ;模拟继电器触点释放时间
        MOV  AL, DH
        OUT  20H, AL          ;先使P0吸合
        CALL NEAR DELAY2      ;模拟触点闭合及数字电压表转换时间
        IN   AX, 10H          ;输入
        MOV  [BX], AX         ;存入内存
        INC  BX
        INC  BX
        RCL  DH, 1            ;DH左移(大循环)1位，为下一个触点吸合作准备
        JNC  AGAIN            ;8位都输入完了吗？没有，则循环
DONE:                         ;输入已完，则执行别的程序段。
```

6.2 CPU与外设之间数据传送方式

6.2.1 程序传送

2. 程序查询传送 (条件传送—异步传送)

也是一种程序传送，与无条件同步传送不同，是有条件的异步传送。

条件是：在执行输入(IN指令)或输出(OUT指令)前，先查询接口中状态寄存器的状态。

输入时，由状态信息指示要输入数据是否已“准备就绪”；

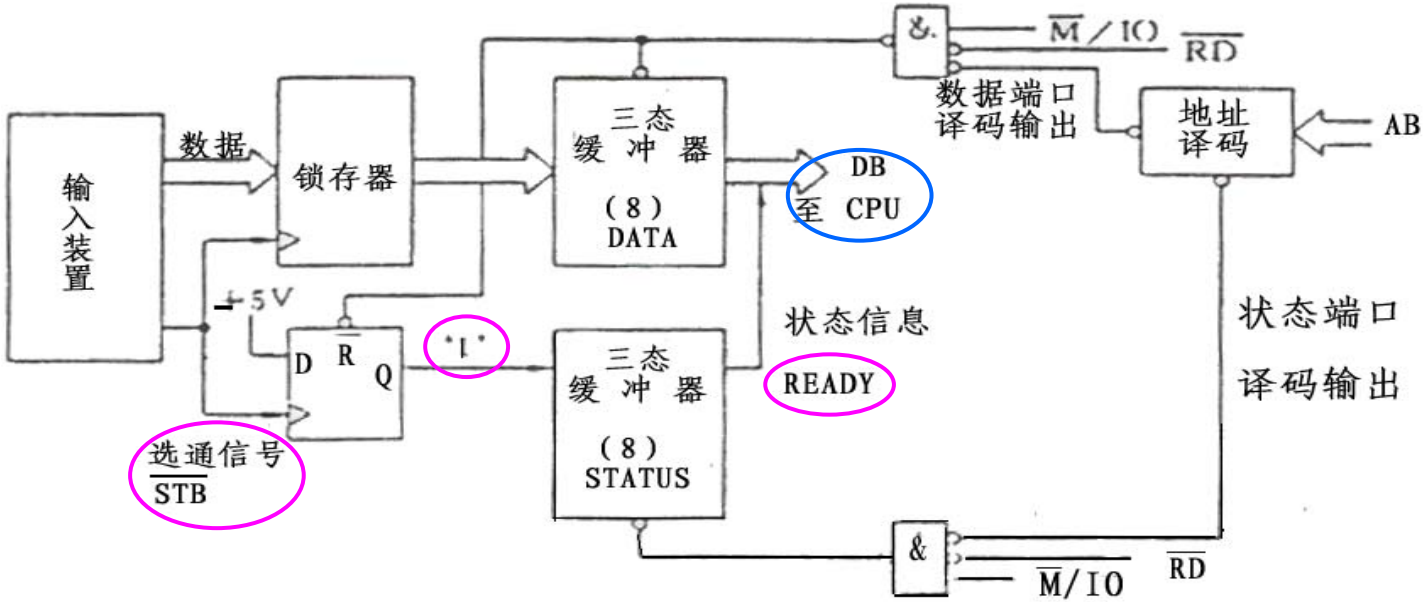
输出时，由它指示输出设备是否“空闲”，由此条件来决定执行输入或输出。

6.2 CPU与外设之间数据传送方式

6.2.1 程序传送

2. 程序查询传送

① 程序查询输入



数据与状态必须有不同的端口分别输入至CPU数据总线。

读入数据命令使状态信息清0，为下次输入新数据做准备。

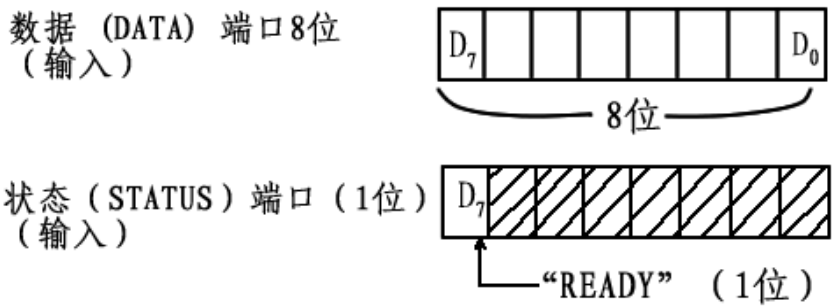
6.2 CPU与外设之间数据传送方式

6.2.1 程序传送

2. 程序查询传送

① 程序查询输入的数据和状态信息

读入的数据是 8 位，而读入的状态信息往往是1位，因此，不同的外设其状态信息可使用同一个端口，只要使用不同的位就可以。

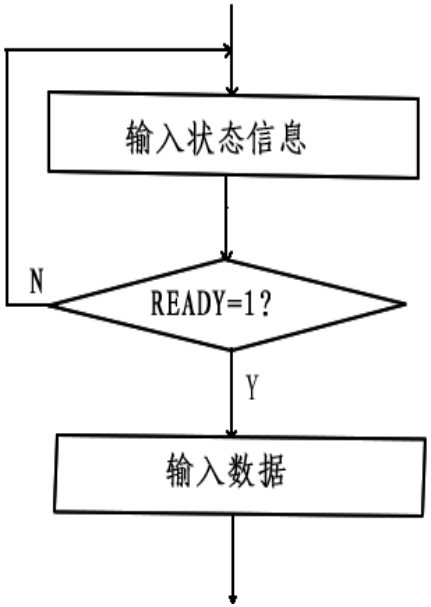


6.2 CPU与外设之间数据传送方式

6.2.1 程序传送

2. 程序查询传送

① 查询输入方式的程序流程图



6.2 CPU与外设之间数据传送方式

6.2.1 程序传送

2. 程序查询传送

① 程序查询输入部分的程序

```
POLL:   IN  AL, STATU_SPORT   ; 读状态端口的信息
        TEST AL, 80 H         ; 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
```

6.2 CPU与外设之间数据传送方式

6.2.1 程序传送

2. 程序查询传送

② 程序查询输出

输出时CPU也必须了解外设状态，看外设是否有“空闲”，若有“ 空闲 ”，则CPU执行输出指令；否则就等待再查。

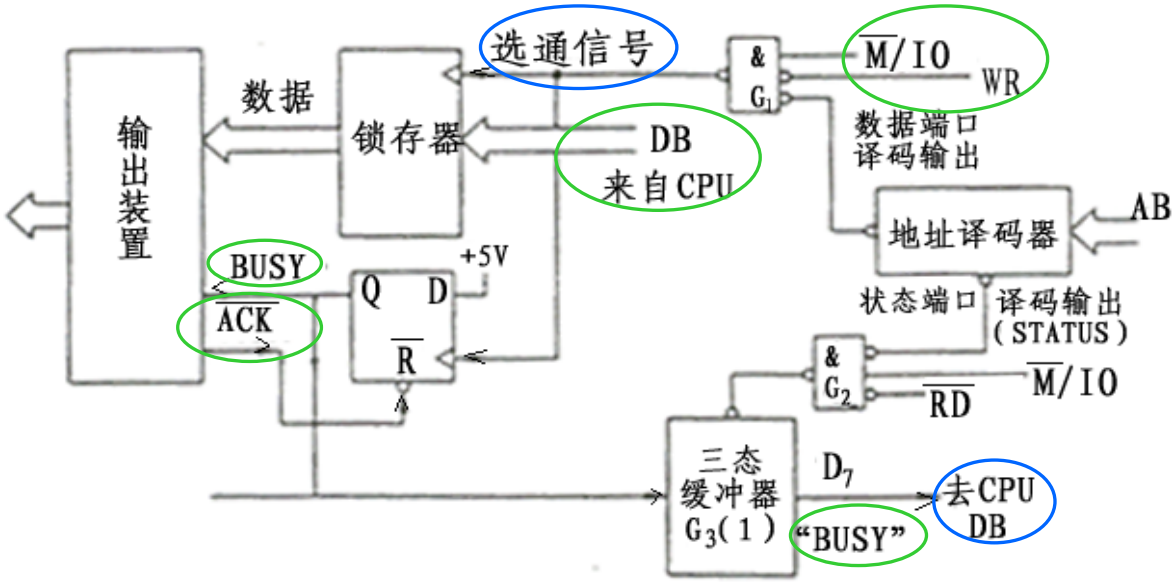
因此，接口电路中也必须要有状态信息的端口。

6.2 CPU与外设之间数据传送方式

6.2.1 程序传送

2. 程序查询传送

② 程序查询输出

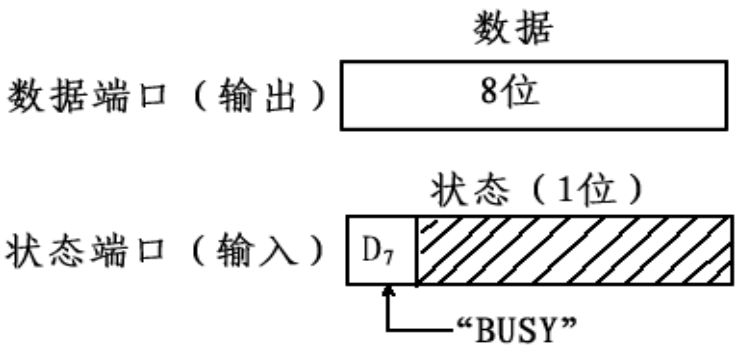


6.2 CPU与外设之间数据传送方式

6.2.1 程序传送

2. 程序查询传送

② 程序查询输出的端口信息

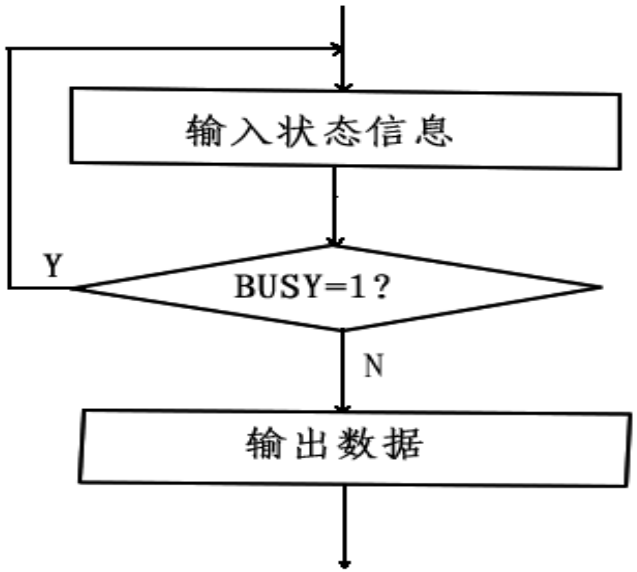


6.2 CPU与外设之间数据传送方式

6.2.1 程序传送

2. 程序查询传送

② 查询输出方式的程序流程图



6.2 CPU与外设之间数据传送方式

6.2.1 程序传送

2. 程序查询传送

② 程序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, 由内存读取数据
      OUT   DATA_PORT, AL  ;输出到DATA地址端口单元
```

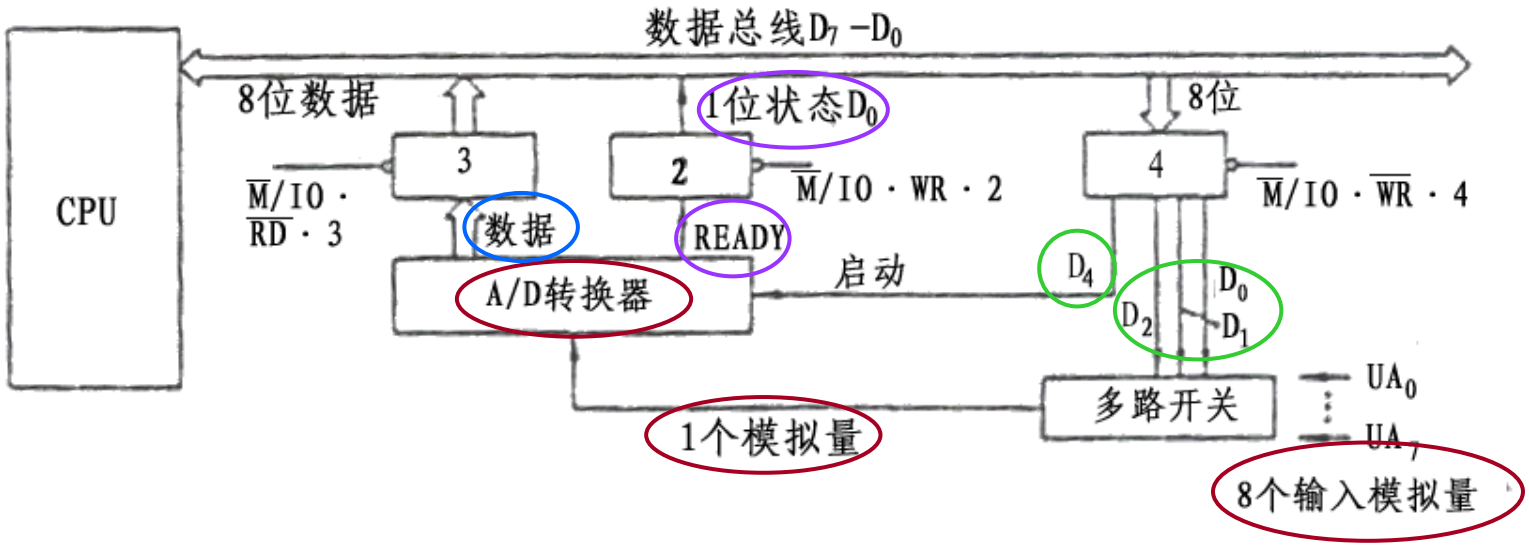
STATUS_PORT和DATA_PORT为状态和数据端口的符号地址；
STORE为待输出数据的内存单元的符号地址。

6.2 CPU与外设之间数据传送方式

6.2.1 程序传送

3. 采用查询方式的数据采集系统

8个模拟量输入数据采集系统，用查询方式与CPU传送信息。



该数据采集系统，用到了3个端口，它们有各自的地址。

6.2 CPU与外设之间数据传送方式

6.2.1 程序传送

3. 采用查询方式的数据采集系统（续1）

采集过程要求：

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

6.2 CPU与外设之间数据传送方式

```
STARE: MOV DL, 0F8H          ; 设置启动A/D转换信号,
                             ; 低3位选通多路开关通道
       MOV AX, SEG  DSTOR    ; 设置输入数据的内存单元地址指针
       MOV ES, AX
       LEA  DI, DSTOR
AGAIN: MOV AL, DL
       AND AL, 0EFH          ; 使D4= 0
       OUT 04, 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
       OUT 04, AL            ; 选输入通道并启动A/D转换
POLL:  IN  AL, 02            ; 输入状态信息
       SHR  AL, 1            ; 查AL的D0
       JNC  POLL             ; 判READY=1?若D0=0,未准备好,则循环再查
       IN  AL, 03            ; 若已准备就绪,则经端口 3 将采样数据输入至A
       STOSB                 ; 输入数据转送内存单元
       IN   CDL              ; 输入模拟量通道增1
       JNE  AGAIN            ; 8个模拟量未输入完则循环
                             ; 输入已完,执行别的程序
```

6.2 CPU与外设之间数据传送方式

6.2.1 程序传送

4. 程序查询输入/输出传送方式的执行过程

① CPU从接口状态端口中读入外设状态信息“READY”或“BUSY”。

② 根据读入的状态信息进行判断。

程序查询输入时，若READY=0，则外设数据未准备好，CPU继续等待查询，直至READY=1，执行下一步操作;

程序查询输出时，若BUSY=1，则外设正在“忙”，CPU继续等待询，直至BUSY=0时，执行下一步操作。

③ 执行输入/输出指令，进行I/O传送。完成数据的输入/输出，同时将外设的状态信息复位，一个8位的数据传送结束。

6.2 CPU与外设之间数据传送方式

6.2.1 程序传送

5. 小结

当计算机工作任务较轻或CPU不太忙时，可以应用程序查询输入/输出传送方式，它能较好地协调外设与CPU之间定时的差别；程序和接口电路比较简单。

主要缺点是：CPU必须作程序等待循环，不断测试外设的状态，直至外设为交换数据准备就绪时为止。这种循环等待方式很花费时间，大大降低了CPU的运行效率。

6.2 CPU与外设之间数据传送方式

6.2.2 中断传送

程序查询方式降低了CPU运行效率，在实时控制系统中，往往有数十乃至数百个外设，由于工作速度不同，要求CPU服务随机的，有些要求很急迫。

若用查询方式除浪费大量等待查询时间外，还很难使每一个外设都能工作在最佳工作状态。

为提高CPU执行有效程序的工作效率和提高系统中多台外设的工作效率，可以让外设处于能主动申请中断的工作方式，当有多个外设及速度不匹配时，尤为重要。

6.2 CPU与外设之间数据传送方式

6.2.2 中断传送

中断是外设或其他中断源中止CPU当前正在执行程序，而转向该外设服务的程序，一旦服务结束，又返回原程序继续工作。

外设处理数据期间，CPU不必浪费大量时间去查询其状态，只外设处理完毕主动向CPU提出请求。

CPU在每一条指令执行的结尾阶段，均查询是否有中断请求信号，若有，则暂停执行现行程序，转去为申请中断的某个外设务。

6.2 CPU与外设之间数据传送方式

6.2.2 中断传送

优点：大大提高了CPU工作效率。

缺点：需由CPU通过程序来传送数据，并在处理中断时，还要“保护现场”和“恢复现场”，要占用一定时间。

对于高速外设以及成组交换数据的场合，显得太慢。

6.2 CPU与外设之间数据传送方式

6.2.3 直接存储器存取 (DMA)传送

- DMA (Direct Memory Access) 方式

由专门硬件电路执行I/O交换，外设接口直接与内存进行高速数据传送，而不必经过CPU。

不必进行保护现场等额外操作，便可实现对存储器直接存取。

这种专门的硬件电路就是DMA控制器，简称为DMAC。

优点：速度快，数据传送的速率只受存储器访问的限制。
CPU不参与操作，省去取指令，指令译码、存取数等。

缺点：硬件电路较复杂。

6.3 中断技术

➢ 中断的产生

- 生活中的中断
  - ✓ 上课时手机响了
  - ✓ 印度洋发生海啸
- 社会中的中断
  - ✓ 日本袭击珍珠港
  - ✓ 美国发生 911
- 计算机中的中断
  - ✓ 鼠标点击新的图标
  - ✓ QQ 有新的消息

突发性

6.3 中断技术

➢ 中断系统

- **中断请求（突发事件）**
  - ✓ 上课时手机响了
  - ✓ 日本袭击珍珠港
  - ✓ QQ 有新的消息
- **中断处理（响应事件）**
  - ✓ 关手机或接手机
  - ✓ 美国用原子弹炸广岛
  - ✓ 与 Q 友聊天

无请求不响应　有请求才响应

6.3 中断技术

6.3.1 中断概述

1. 中断常用术语

① 中断

CPU在执行正常程序时，为处理一些紧急发生的情况，暂时中止当前程序，转而对该紧急事件进行处理（中断服务程序），并在处理完后返回正常程序的过程。

6.3 中断技术

6.3.1 中断概述

② 中断源

引起中断的事件或原因，或发出中断申请的来源。

a) 外部设备：中、慢速外设，如键盘、打印机、A/D转换器等。

高速外设，如磁盘或磁带，向CPU提出总线请求，进行DMA传送。

b) 实时时钟：在自动控制中，常遇到定时检测与控制，这时可采用外部时钟电路。

c) 故障源：计算机内设有故障自动检测装置，如发生运算出错、存储器读出出错、外部设备故障、电源掉电等意外事件时，这些装置都能使CPU中断，进行相应的中断处理。

d) 为调试程序设置的中断源：CPU执行了特殊指令(自陷指令）或由硬件电路引起的中断，如断点设置、单步调试等。

6.3 中断技术

6.3.1 中断概述

③ 中断向量

中断服务子程序的入口地址，即中断服务子程序的第一条指令的地址在存储器中的存放位置。

④ 中断向量表

中断向量构成的表格，位于存储器的最低地址单元。

⑤ 断点

执行的现行程序被中断时的下一条指令的地址，又称断点地址。

⑥ 现场

中服程序中应保护和恢复的相关信息。

6.3 中断技术

6.3.1 中断概述

⑦ 中断优先级

系统中多个中断源同时提出中断请求时，需按中断的轻重缓急给每个中断源指定一个优先级别。

⑧ 中断嵌套

中断服务程序运行中响应更高级别中断请求。

6.3 中断技术

6.3.1 中断概述

⑨ 中断类型号

处理器对各类中断的中断源进行的统一编号n，n的取值范围是0~255。

⑩ 中断分类

- 外部中断（硬件中断）
  - 可屏蔽中断 (INTR)
  - 不可屏蔽中断 (NMI)
- 内部中断（软件中断）（*执行指令所引起的*）

6.3 中断技术

6.3.1 中断概述

- 中断系统概念

为实现中断而设置的各种硬件与软件，包括中断控制逻辑及相应管理中断的指令。

- 中断技术的概念

把实现中断所需要的软硬件技术称为中断技术。

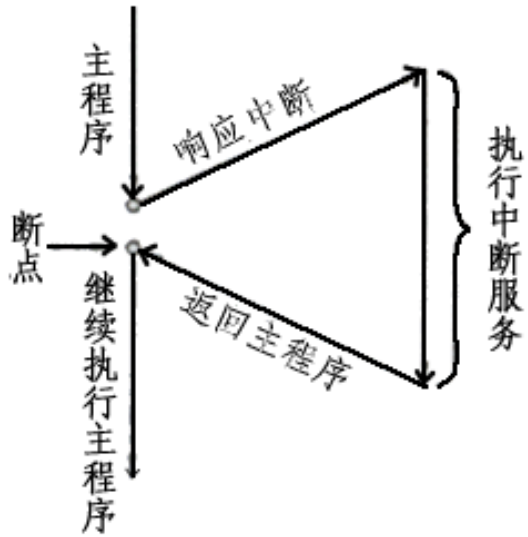
6.3 中断技术

6.3.1 中断概述

3. 中断系统功能

① 能响应中断、处理中断及返回

中断源发出中断请求，CPU决定是否响应，若响应，则保护断点和现场，转入相应中断服务程序，中断服务结束后，恢复现场和断点，继续执行原程序。



6.3 中断技术

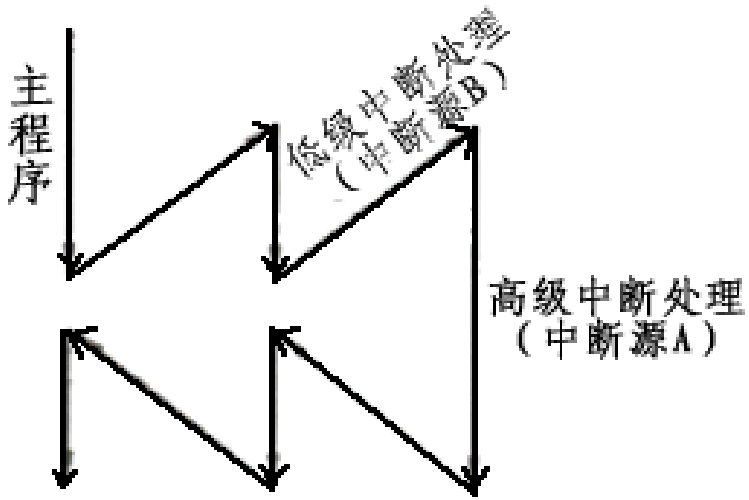
② 能实现优先权排队

- ✓ 按各中断请求的重要程度排列CPU响应的次序称为中断优先级。
- ✓ 即同时有多个中断请求到来时，CPU会首先响应和处理优先级别最高的中断请求。
- ✓ 中断优先级的实现可以用软件或硬件设置。

6.3 中断技术

③ 高级中断源能中断低级的中断处理

两重中断(或两级嵌套)，还可进行多重中断 (或多级嵌套)。



6.3 中断技术

6.3.1 中断概述

4. 中断的应用

中断除能解决快速CPU与中、慢速外设速度不匹配的矛盾，以提高主机的工作效率之外，在实现分时操作、实时处理、故障处理、多机连接以及人机联系等方面均有广泛的应用。

6.3 中断技术

6.3.2 单个中断源的中断过程

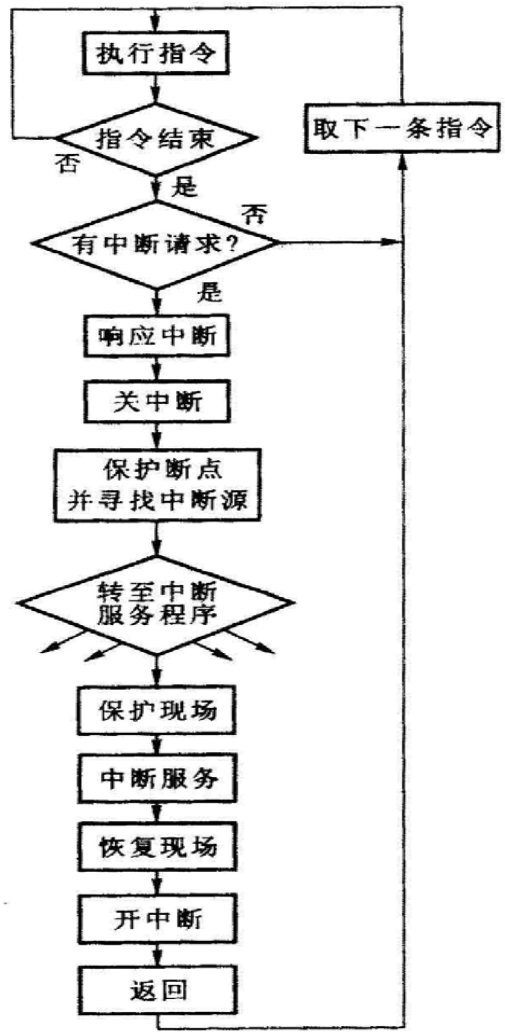
包括：

中断请求

中断响应

中断处理

中断返回等环节。



6.3 中断技术

6.3.2 单个中断源的中断过程

1. 中断源向CPU发中断请求信号的条件

① 设置中断请求触发器

每个中断源，要向CPU发中断请求信号，首先应能由它的接口电路提出中断请求，该请求能保持着，直至CPU接受并响应该中断请求后才能清除。

在每个中断源的接口电路中设置一个中断请求触发器，由它产生中断请求。

6.3 中断技术

6.3.2 单个中断源的中断过程

1. 中断源向CPU发中断请求信号的条件（续）

② 设置中断屏蔽触发器

中断请求能否允许以中断请求信号(如INTR)发向CPU，应能受CPU的控制，以增加处理中断的灵活性，为此，在接口电路中，增设一个**中断屏蔽触发器**。

若有多个中断源，则可将多个外设的中断屏蔽触发器组成一个端口，用输出指令来控制它们的状态。

6.3 中断技术

6.3.2 单个中断源的中断过程

2. CPU响应中断的条件

① CPU开放中断

CPU采样到 INTR 信号后是否响应它，由CPU内设置的中断允许触发器 (如IF) 的状态决定。

IF的状态可由专门设置的开中断与关中断指令来改变。

此外，当CPU复位或响应中断后，也能使CPU关中断。

6.3 中断技术

6.3.2 单个中断源的中断过程

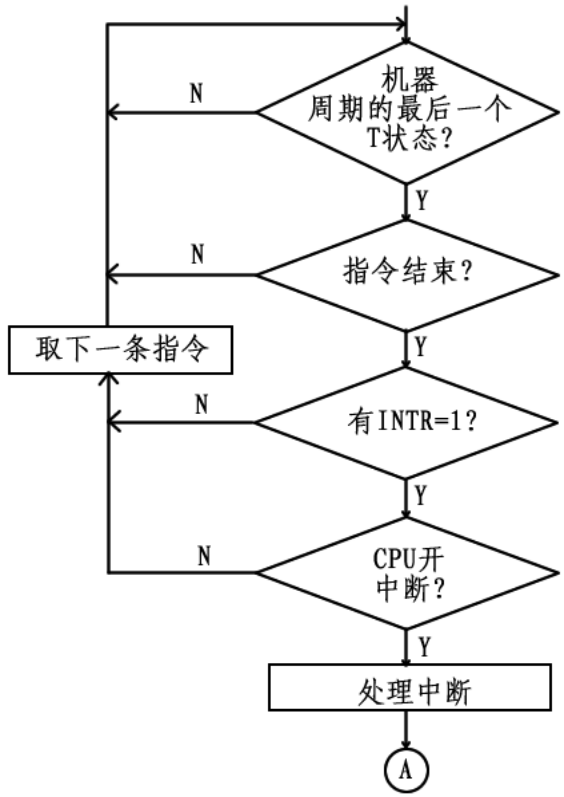
2. CPU响应中断的条件（续）

② CPU在现行指令结束后响应中断

CPU开中断时，若有中断请求信号发至CPU，它也并不立即响应。

只有当现行指令运行到最后一个机器周期的最后一个T状态时，CPU才采样INTR信号；

若有此信号，CPU进入中断响应周期。



6.3 中断技术

6.3.2 单个中断源的中断过程

3. CPU响应中断及处理过程

① 关中断

② 保留断点

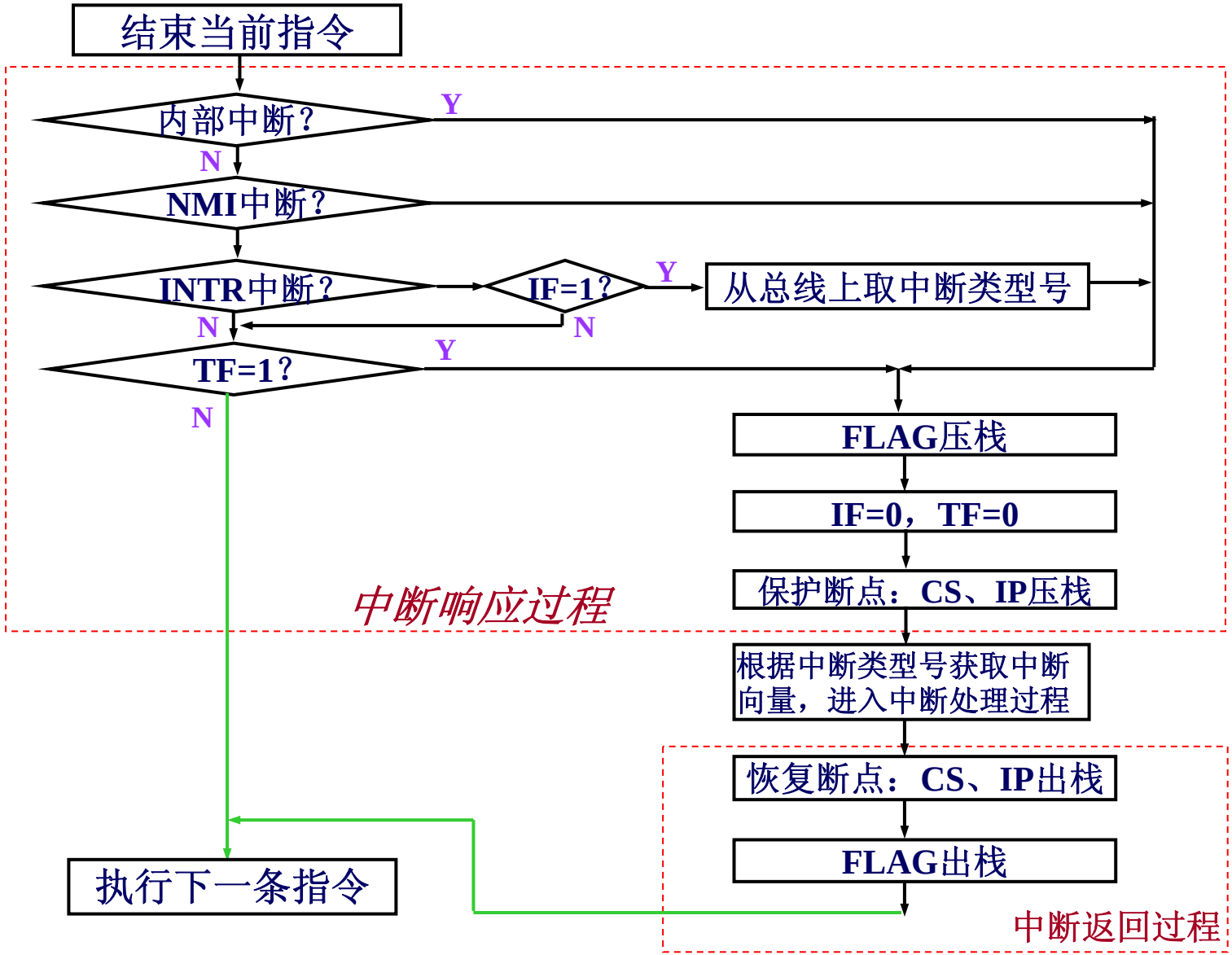
③ 保护现场

④ 给出中断入口(地址)，转入相应的中断服务程序

⑤ 恢复现场

⑥ 开中断与返回

6.3 中断技术


结束当前指令
内部中断？
Y
N
NMI中断？
INTR中断？
IF=1？
Y
从总线上取中断类型号
N
N
TF=1？
Y
N
FLAG压栈
IF=0，TF=0
保护断点：CS、IP压栈
中断响应过程
根据中断类型号获取中断向量，进入中断处理过程
恢复断点：CS、IP出栈
FLAG出栈
中断返回过程
执行下一条指令

6.3 中断技术

6.3.3 向量中断

向量中断(Vectored Interrupt)，是指通过中断向量来找中断入口地址进而转向中断服务程序的一种方法；

中断向量则是用来提供中断入口地址的一个地址指针。

8086/8088CPU的中断系统就是采用这种向量中断。

6.3 中断技术

6.3.4 中断优先权

实际系统中，具有多个中断源，而CPU的可屏蔽中断请求线往往只有一条。

要求CPU按多个中断源优先权由高至低依次来响应中断申请。

同时，当CPU正在处理中断时，要能响应更高级的中断申请，而屏蔽掉同级或低级的中断申请。

CPU 可通过软件查询技术或硬件排队电路两种方法来实现按中断优先权对多个中断源的管理，也有专门用于协助CPU 按中断优先权处理多个中断源的中断控制芯片，如8259A芯片。

6.3 中断技术

6.3.4 中断优先权

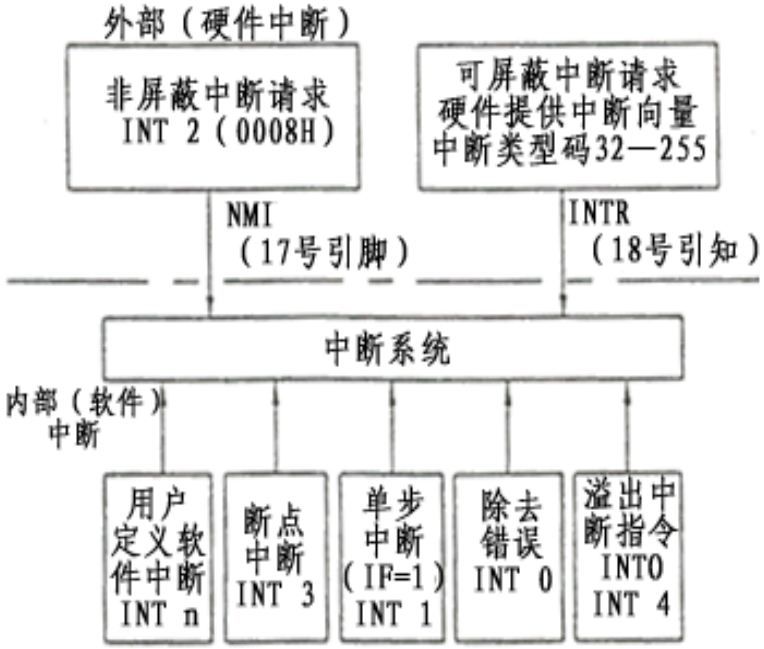
表6.2　8086/8088的中断

优先级	中断名	中断类型	说明
高 ↑	除法	类型0	商大于被除数（软件中断）
	INT n	类型n	内部检查用中断（软件中断）
	INTO	类型4	溢出用（软件中断）
	NMI	类型2	非屏蔽中断（硬件中断）
	INTR	由外设送入	可屏蔽中断（硬件中断）
低	单步	类型1	调试用（软件中断）

6.4 8086/8088中断系统和中断处理

6.4.1 8086/8088中断系统

- ✓ 简要、灵活、多用
- ✓ 采用中断向量结构
- ✓ 每个中断给定一个中断类型号，供CPU识别
- ✓ 可处理256种类型的中断
- ✓ 中断可来自外部，即由硬件产生
- ✓ 中断可来自内部，即由软件（中断指令）产生
- ✓ 满足某些特定条件(陷阱)后引发CPU中断



6.4 8086/8088中断系统和中断处理

6.4.1 8086/8088中断系统

1. 外部中断

8086/8088 CPU有2条引脚供外部中断源请求中断：

一条是高电平有效的可屏蔽中断INTR；

另一条是正跳变有效的非屏蔽中断NMI。

6.4 8086/8088中断系统和中断处理

6.4.1 8086/8088中断系统

1. 外部中断

①可屏蔽中断

CPU的INTR引脚上出现的请求信号须保持到当前指令的结束。

每条指令最后一个时钟周期CPU对INTR引脚采样，如采样到有中断请求信号产生，是否响应取决于标志寄存器中IF状态。

若IF=0，CPU处于关中断状态，不响应INTR；

若IF=1，CPU处于开中断状态，响应INTR，并通过INTA引脚发回响应信号，启动中断过程。

6.4 8086/8088中断系统和中断处理

6.4.1 8086/8088中断系统

1. 外部中断

① 可屏蔽中断（续）