第2章 输入输出

- 在微型计算机系统的应用中，CPU除与内存交换信息外，还必然要经常与各种外部设备交换信息。主机与外设进行信息交换过程主要是完成数据输入或输出的传送操作。

- **输入或输出操作：**有选择地启动被微处理器选中的外部设备，以便使其接收来自CPU的数据或向CPU送入数据。

- 数据传送的方向标准通常以微处理器为中心，当数据是由外部设备，如键盘、纸带读入机、光笔等设备向CPU送入时，称为输入传送；而当数据自CPU送到如发光二极管、七段显示器、CRT显示器、点阵打印机、绘图仪等设备时，称为输出传送。

- **输入输出技术：**CPU与外部设备间的连接方法与信息交换手段。

2.1 接口概述

- 2.1.1 接口与端口
- 从广义上讲，接口就是指两个系统或两个部件之间的交接部分，可以是两种硬设备之间的连接电路，也可以是两个软件之间公用的逻辑边界。
- 接口：在微型计算机系统中，CPU与外部设备之间的联系，需要有特定的硬件连接和相应的控制软件。
- 接口技术：完成这一任务的软、硬件的综合称为。对这种硬件、软件的设计。
- 接口（Interface）和端口（Port）是不同的。
- 端口：接口电路中那些完成信息传送，可由程序寻址并进行读写操作的寄存器。
- 原则上讲，若干个端口加上相应的控制逻辑才构成接口。所以，一个接口中往往含有几个端口，CPU可以通过输入指令从端口读出信息，通过输出指令向端口写入信息。CPU寻址的是端口，而不是笼统的外设接口。

2.1.2 接口的功能

- 1．地址译码或设备选择
- 接口必须进行地址译码，从而产生设备选择信号，以使微处理器和指定的外部设备交换信息。
- 2．数据缓冲和锁存
- 在微计算机系统中，数据总线是系统各部分之间公用的双向总线，所有设备分时复用。所以，无论是存储器，还是外部设备，都不能长期占用数据总线，只允许被选中的设备在读/写周期内可用其传送数据。
- 3．信息格式与电平的转换
- 接口应该具有信号传送格式、信号类型、信号电平的转换能力。
- 4．数据传送的协调
- CPU工作是有一定的时序的，CPU与外部设备交换数据时必须采用一定的传送方式进行控制。

2.1.3 接口的一般编程结构

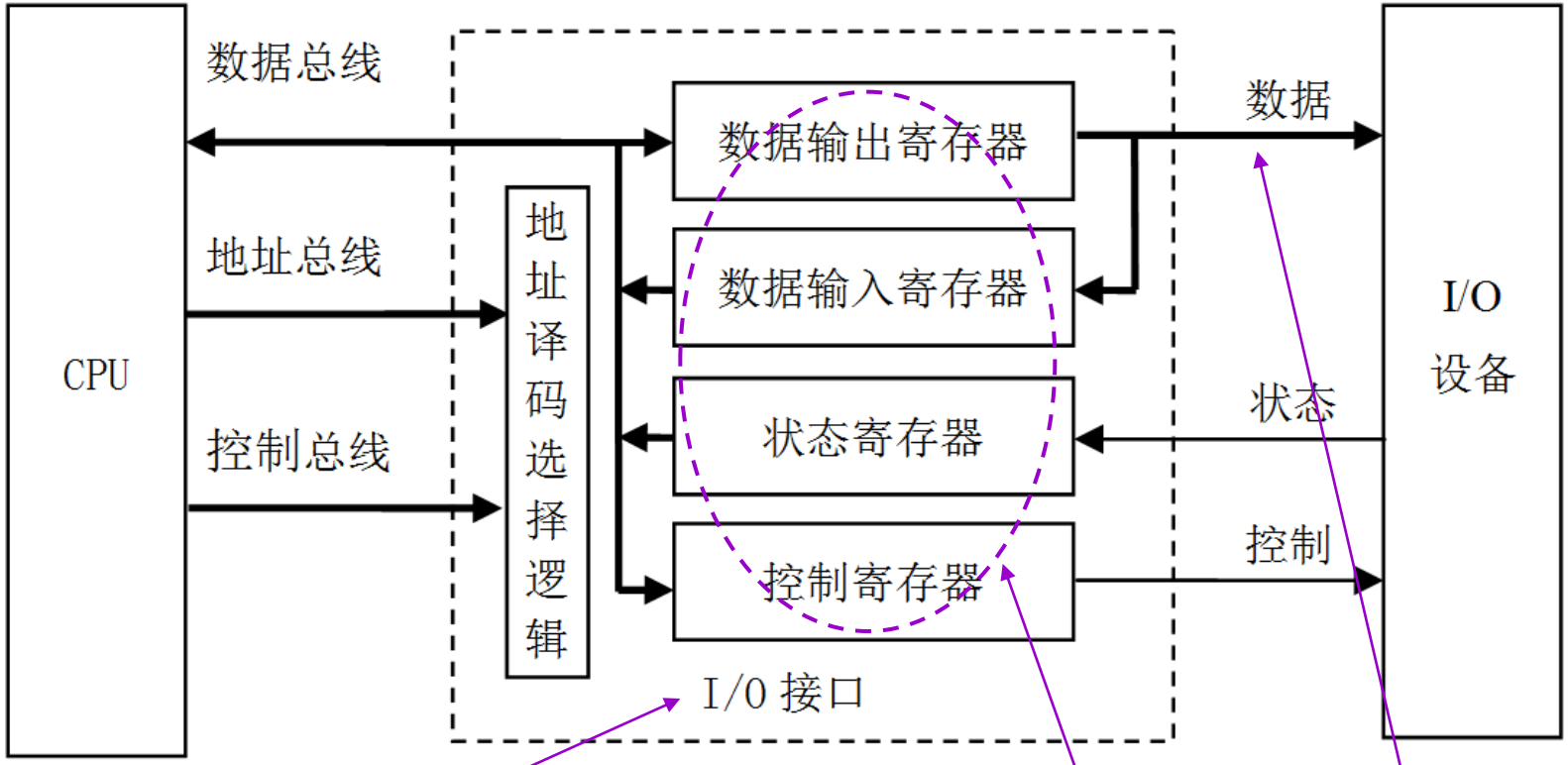


图 2.1.1　I/O 接口的一般编程结构和外部连接示意图

接口组成：寄存器，译码与控制，连接线

从用户角度看，有4个寄存器

数字量，模拟量，脉冲量，开关量

2.1.4 接口的分类

- 1．按应用分类
- 用户交互接口：将来自用户的数据、信息传送给计算机，或将用户所需的数据、信息由计算机传送给外部设备。键盘接口、打印机接口、显示器接口等。
- 辅助操作接口：计算机发挥最基本的处理与控制功能所必须的接口，包括各类总线驱动、总线接收器、数据锁存器、三态缓冲器、时钟电路、CPU与内存的接口等。
- 传感接口：传感接口是输入被监测对象和控制对象变化信息的接口。压力传感器、温度传感器、流速传感器等 。
- 控制接口：微计算机对被监测对象或控制对象输出控制信息的接口。步进电机、电磁阀门、继电器等 。

2.1.4 接口的分类

- 2．按功能分类

（1）按数据传送方式分类：并行接口、串行接口。

（2）按接口通用性分类：通用接口、专用接口。

（3）按接口的可编程性分类：可编程接口、不可编程接口。

（4）按接口输入/输出信号分类：数字接口、模拟接口。

2.2 I/O端口的地址选择

- 微计算机的操作速度很快，可以控制很多外部设备。但是，微计算机采用的是总线结构，只有一组数据线。
- 当CPU发出一个数据信息后，到底哪一个外部设备来接收这个数据呢？不得而知。
- 因此，在微计算机与外部设备交换信息之前，应首先通过地址总线发出地址信息，通过某种编址方式来选中一个外部设备，进而实现信息交换。
- 所谓的“与外部设备”交换信息，确切地应理解为“与外部设备的端口”交换信息。
- 端口的编址（寻址）有两种方式：存储器映像方式、I/O映像方式。
- 常见的端口地址选择方法有三种：门电路组合法、译码器译码法、比较器比较法。

2.2.1 输入输出的寻址方式

- 1．存储器映像方式
- 把一个I/O端口看作是一个存储单元，相当于给每一个I/O端口分配一个存储器地址。
- 优点：

  （1）指令丰富。

  （2）I/O端口空间大。

  （3）寻址的控制逻辑比较简单。
- 缺点：

  （1）I/O端口占用一部分存储器地址空间，使可用的内存空间相对减少。

  （2）对I/O端口的访问和对存储器的访问一样，必须对全部地址线译码，因而地址译码电路比较复杂。

  （3）存储器操作指令的机器码比较长，需要较长的执行时间。

  （4）用存储器指令来处理输入/输出操作，在程序清单中不易区别，给程序的设计、分析、调试带来一定的困难。
- 采用存储器映像寻址方式的计算机有PDP-11小型机、6800系列微型机、6502系列微型机等。

- 2．I/O映像方式
- 在这种寻址方式中，I/O端口空间与存储器空间各自独立，互不干涉，互不影响，故亦称为独立的I/O寻址方式。
- 在指令操作上，对存储单元的一般性传送使用MOV指令，而对I/O端口的传送操作，使用系统专门提供的一组I/O指令，即IN和OUT指令。
- 优点：

  （1）I/O空间与存储器空间各自独立，可分开设计。

  （2）由于采用单独的I/O指令，其助记符与存储器指令明显不同，因而使程序编制清晰，易于理解。

  （3）I/O地址线较少，所以译码电路简单。

  （4）I/O指令格式短，执行时间快。
- 缺点：

  （1）需要专门的I/O指令，且这些指令一般不如存储器访问指令丰富，程序设计灵活性较差。

  （2）参加译码的地址线较少，使外设端口的数目受到限制。

  （3）采用专用的I/O周期和专用的I/O控制线，这不仅使微处理器有限的引脚更加紧张，而且也增加了控制逻辑的复杂性。

2.2.2 用门电路组合法进行端口地址选择

- 门电路组合法：最简单的一种端口地址选择方法，采用与门、或门、非门等作为基本的组合元件。
- 端口都有一个芯片选择信号，简称片选信号，多数是低电平有效。
- 端口地址选择的目的，是当地址线上出现某种信息组合时，在端口地址选择电路的输出端会产生一个有效信号，该信号连到器件的控制端，使器件产生动作，从而完成I/O端口的读/写操作。
- 有效信号有四种状态：高电平、低电平、上跳沿、下跳沿，具体使用哪种状态，视所使用的器件而定。

2.2.3 用译码器译码法进行端口地址选择

- **译码器译码法：**利用译码器芯片对地址进行译码。
- PC机系统板上接口芯片的端口地址译码电路采用译码器译码法。
- 译码器采用74LS138。
- 译码器只直接使用地址线A9-A5，其余的低5位地址线A4-A0未接，留给各接口芯片内部自行译码，以便寻址多个寄存器。

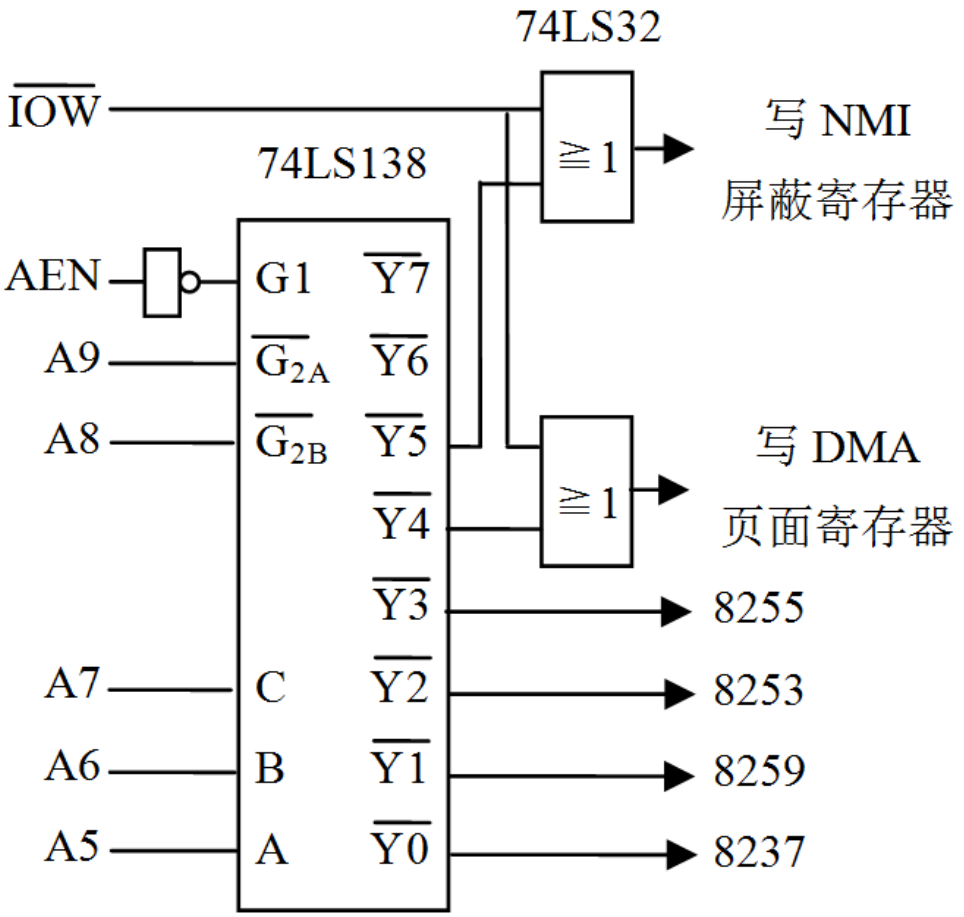


图 2.2.2　PC/XT 系统端口地址译码

2.2.4 用比较器比较法进行端口地址选择

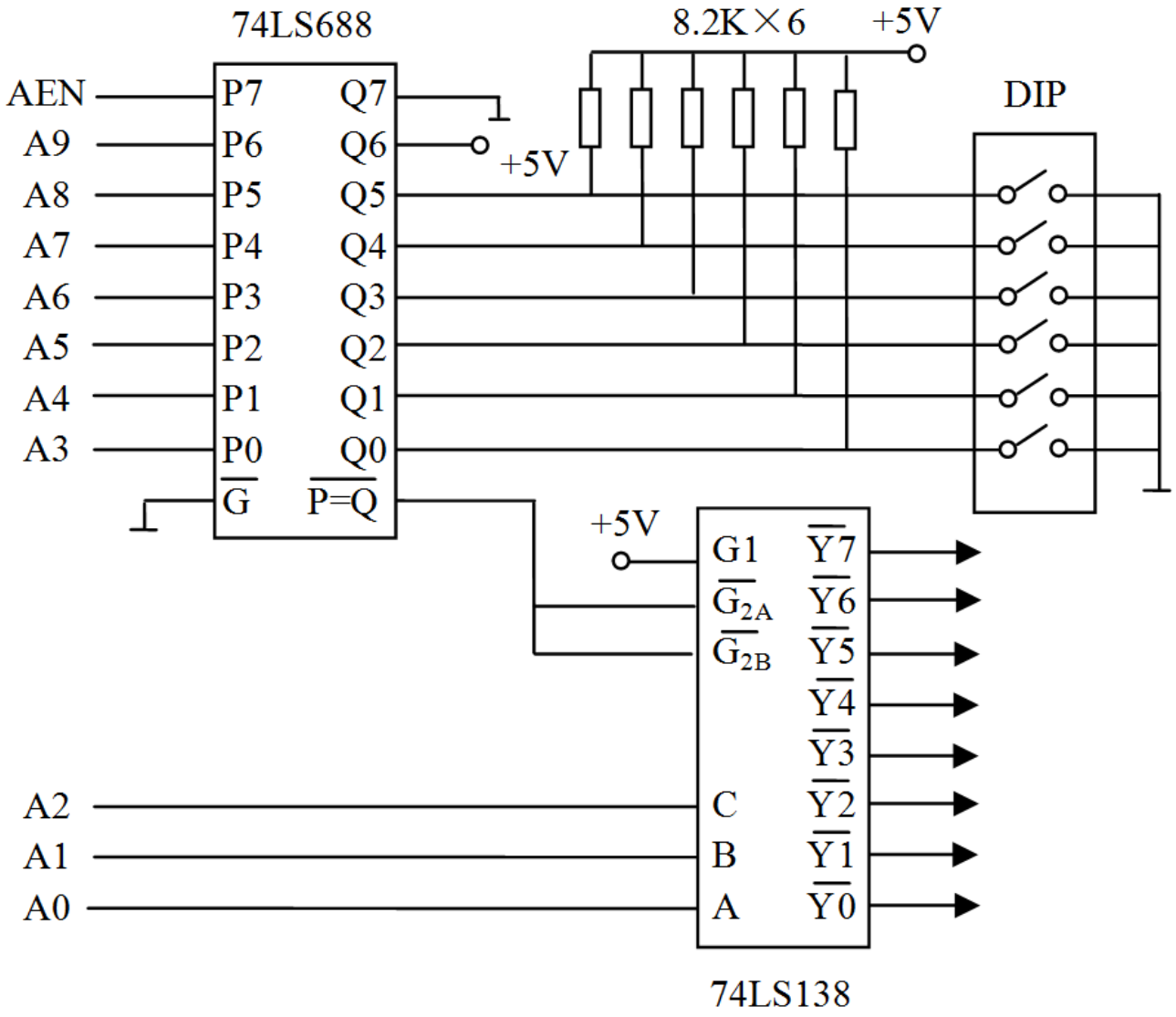


图 2.2.3　采用 74LS688 比较器的端口地址译码

2.3 输入输出控制方式

- 四种基本控制方式：程序查询方式、程序中断方式、DMA方式、I/O处理机方式。
- 前两种主要由程序来实现，后两种主要由附加硬件来实现。目前，微机中多数采用前三种。
- 程序查询方式：CPU通过查询I/O设备的状态，断定哪个设备需要服务，然后转入相应的服务程序。
- 程序中断方式：当I/O设备需要CPU为其服务时，可以发生中断请求信号INTR，CPU接到请求信号后，中断正在执行的程序，转去为该设备服务，服务完毕，返回原来被中断的程序并继续执行。
- 直接存储器存取（DMA）方式：采用这种方式时，在DMA控制器的管理下，I/O设备和存储器直接交换信息，而不需要CPU介入。
- I/O处理机方式：引入I/O处理机，全部的输入/输出操作由I/O处理机独立地承担。

2.3.1 程序查询方式

- 程序查询方式又分为无条件传送方式和查询传送方式。

- 查询：就是询问外部设备的工作状态，通过这一状态来判定外设是否已具备了与CPU交换数据的条件，即外设是否已准备好与CPU交换数据。

- 对输入设备而言，这个状态指示输入设备的数据是否已经准备就绪，CPU是否可以随时来读取这个数据。

- 对输出设备而言，这个状态指示输出设备的数据接收寄存器是否已空，是否可以随时接受CPU送来的数据。

1. 无条件传送方式

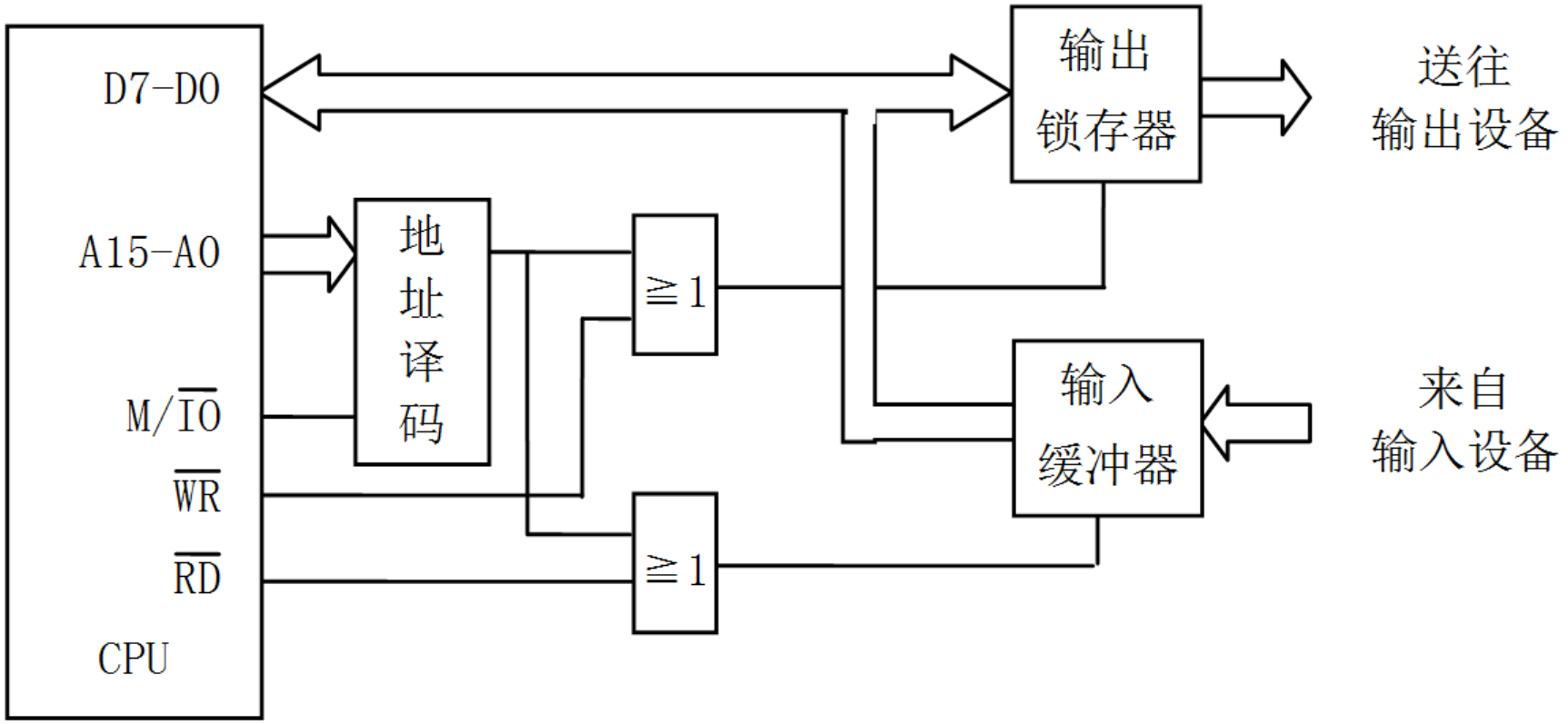


图 2.3.1　无条件传送方式的工作原理

2. 查询输入传送方式

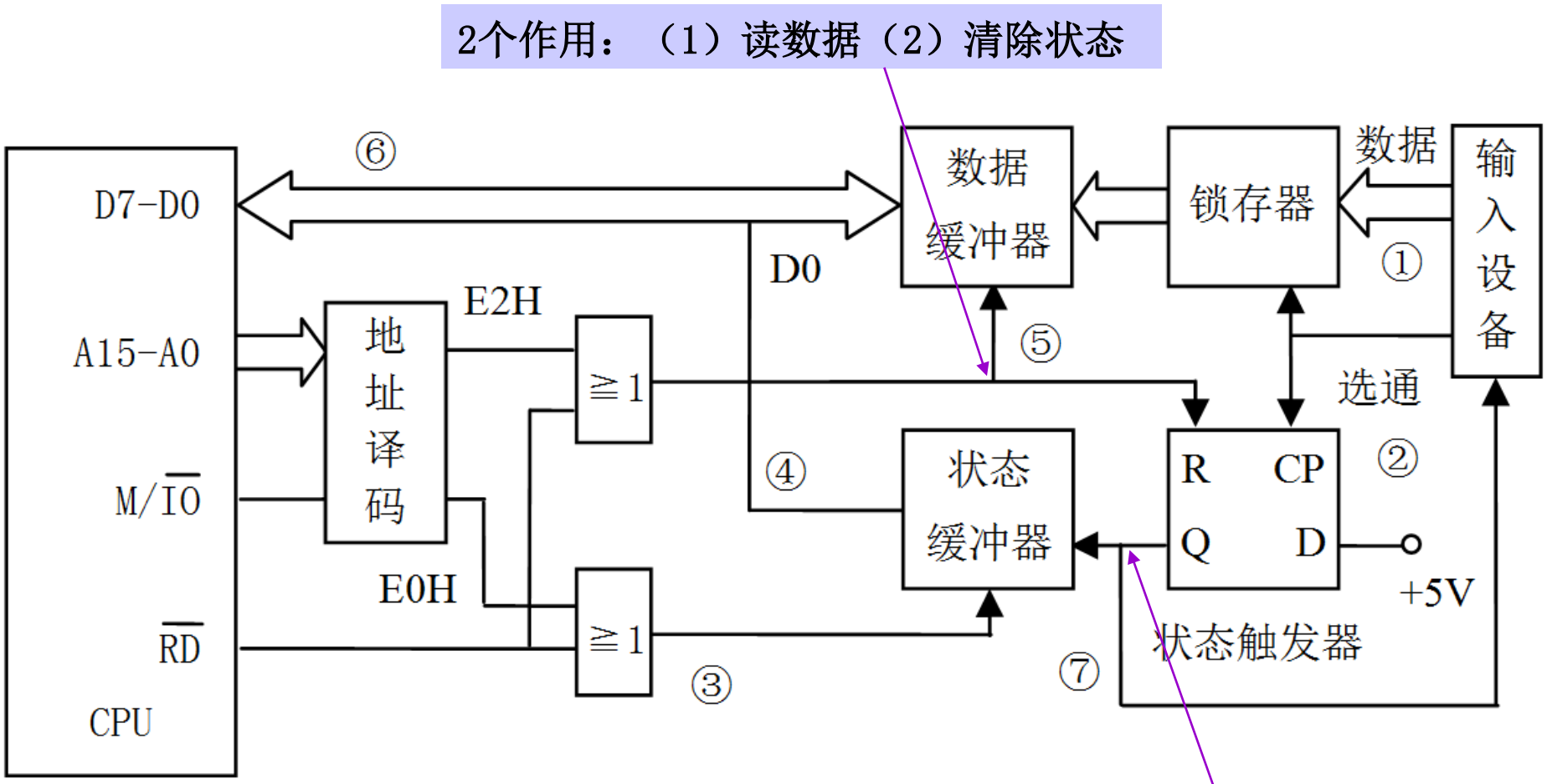


图 2.3.2　查询输入的接口电路

- 查询输入程序段如下：

```
X1:   IN    AL, 0E0H      ; 1取状态字
      TEST  AL, 01H       ; 2测试状态位
      JZ    X1            ; 3 D0=0，未准备好，继续查询
      IN    AL, 0E2H      ; 4取输入数据
      ……                  ; 5数据处理
      JMP   X1            ; 6返回继续查询
```

3. 查询输出传送方式

2个作用：（1）输出数据（2）设置设备“忙”状态

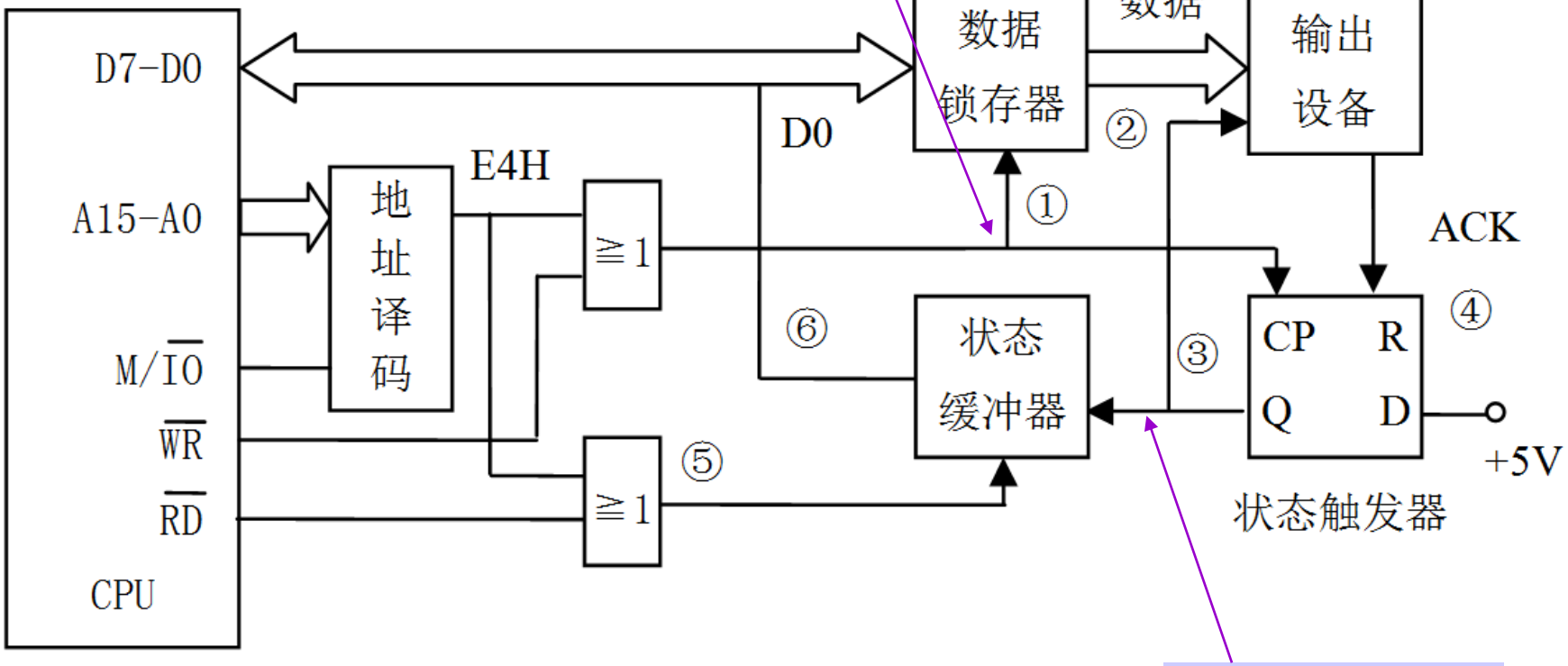


图 2.3.3　查询输出的接口电路

“1”设备忙，不能接收数据

- 查询输出程序段如下：

```
        MOV   AL，41H        ；1 数据41H送AL
        OUT   0E4H，AL       ；2 数据存入锁存器，并使状态触发器为1
X1：    IN    AL，0E4H       ；3 取状态位
        TEST  AL，01H        ；4 测试状态位
        JNZ   X1             ；5 D0=1，设备忙，继续查询
        MOV   AL，42H        ；6 下一个数据42H送AL
        OUT   0E4H，AL       ；7 将下一个数据存入锁存器
        ┇
```

2.3.2 程序中断方式

- 程序查询方式的缺点：CPU和外设只能串行工作，各外设之间也只能串行工作。
- 为了使CPU和外设以及外设和外设之间能并行工作，提高系统的工作效率，充分发挥CPU高速运算的能力，在微机系统中引入了中断技术，利用中断来实现CPU与外设之间的数据传送，这就是程序中断传送方式。
- 在程序中断传送方式中，通常是在主程序中某一时刻安排启动某一台外设的指令，然后CPU继续执行其主程序，当外设完成数据传送的准备后，向CPU发出“中断请求”信号，在CPU可以响应中断的条件下，中断（即暂停）现行主程序的执行，而转去执行“中断服务程序”，在“中断服务程序”中完成一次CPU与外设之间的数据传送，传送完成后仍返回被中断的断点处继续执行主程序。

2.3.3 DMA方式

- 1．DMA传送方式的提出

- 中断方式传输效率仍然不高的因素：仍然是通过CPU执行程序来实现数据传送， CPU要保护断点、转入中断服务程序，每次传送一个字节（或一个字）。

- DMA方式：在外设和内存之间直接传送数据，即直接存储器传输方式。

2．DMA操作的基本方法

- （1）CPU停机方式
- 进行DMA传送时，DMA控制器向CPU发出总线请求信号，迫使CPU在现行的总线周期结束后，使其地址总线、数据总线和部分控制总线处于高阻状态，从而让出对总线的控制权，并给出DMA响应信号。DMA控制器接到该响应信号后，就可以对总线进行数据传送的控制工作，直到DMA操作完成，CPU再恢复对总线的控制权，继续执行被中断的程序。
- （2）周期扩展
- 进行DMA操作时，由DMA控制器发出请求信号给时钟电路，时钟电路把供给CPU的时钟周期加宽，而提供给存储器和DMA控制器的时钟周期不变。用这种方法进行DMA操作，一次只能传送一个字节。
- （3）周期挪用
- 利用CPU不访问内存的那些周期来实现DMA操作，此时DMA操作使用总线不用通知CPU也不会妨碍CPU的工作。

3．DMA控制器的功能

- （1）当外设准备就绪，希望进行DMA操作时，会向DMA控制器发出DMA请求信号，DMA控制器接到此信号后，应能向CPU发总线请求信号。
- （2）CPU接到总线请求信号后，如果允许，则会发出DMA响应信号，从而CPU放弃对总线的控制，这时DMA控制器应能实行对总线的控制。
- （3）DMA控制器得到总线控制权以后，要往地址总线发送地址信号，修改所用的存储器的地址指针。
- （4）在DMA传送期间，DMA控制器应能发存储器或接口的读/写控制信号。
- （5）能统计传送的字节数，并且判断DMA传送是否结束。
- （6）能向CPU发出DMA结束信号，将总线控制权交还给CPU。

4. DMA传送的一般工作过程

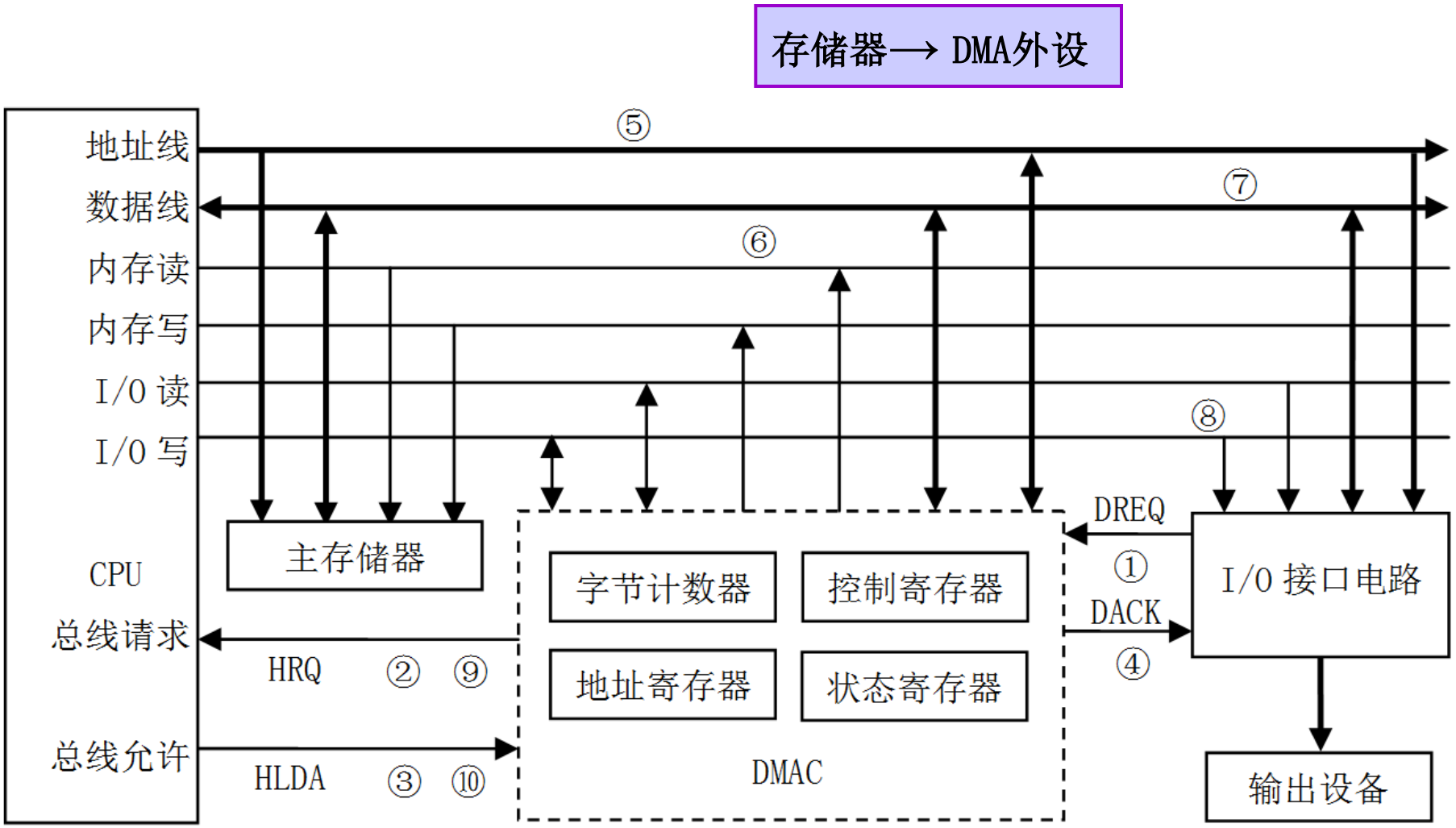


图 2.3.4　存储器向输出设备以 DMA 方式传送数据的示意图

2.3.4 I/O处理机方式

- 为了提高整个系统的工作效率，使CPU完全摆脱管理、控制输入/输出的沉重负担，从20世纪60年代开始又引入了I/O处理机的概念，提出了数据传送的I/O处理机方式。
- 于是，专门用来处理输入输出的I/O处理机IOP应运而生。如Intel 8089就是一种专门配合8086/8088使用的I/O处理器芯片。
- I/O处理机：有自己的指令系统，也能独立地执行程序，能承担原来由CPU处理的全部输入/输出操作。
- 如对外设进行控制、对输入/输出过程进行管理，还可以向CPU报告外设和外设控制器的状态，等等。上述操作都是同CPU程序并行执行的。
- 为了使CPU的操作与输入/输出操作并行进行，必须使外设工作所需要的各种控制命令和定时信号与CPU无关，由I/O处理机独立形成。

2.4 DMA控制器8237A

- 主要的功能有：

1．在一片8237A内有4个独立的DMA通道。

2．每个通道的DMA请求可分别编程允许或禁止。

3．每个通道的DMA请求优先级有两种：固定优先级和循环优先级。固定优先级的顺序是通道0最高，通道3最低。

4．可在外设与存储器，存储器与存储器之间传送数据。

5．四种工作方式：单字节传送方式，数据块传送方式，请求传送方式，级连方式。

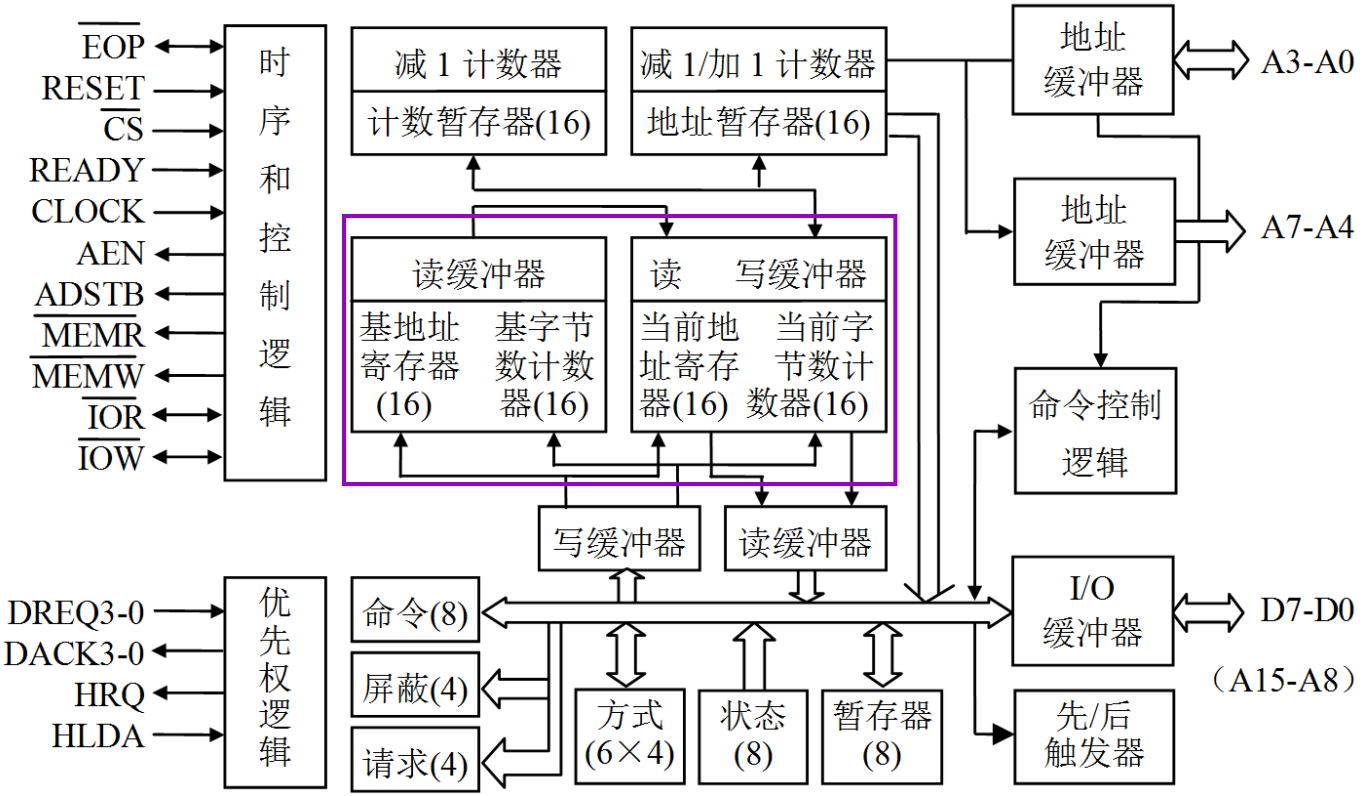
6．可以多片级连，扩展通道数。

7．DMA操作结束有两种方法：一是字节计数器减1由0变为FFFFH，二是外界通过 $\overline{\text{EOP}}$ 输入负脉冲，强制DMA操作结束。

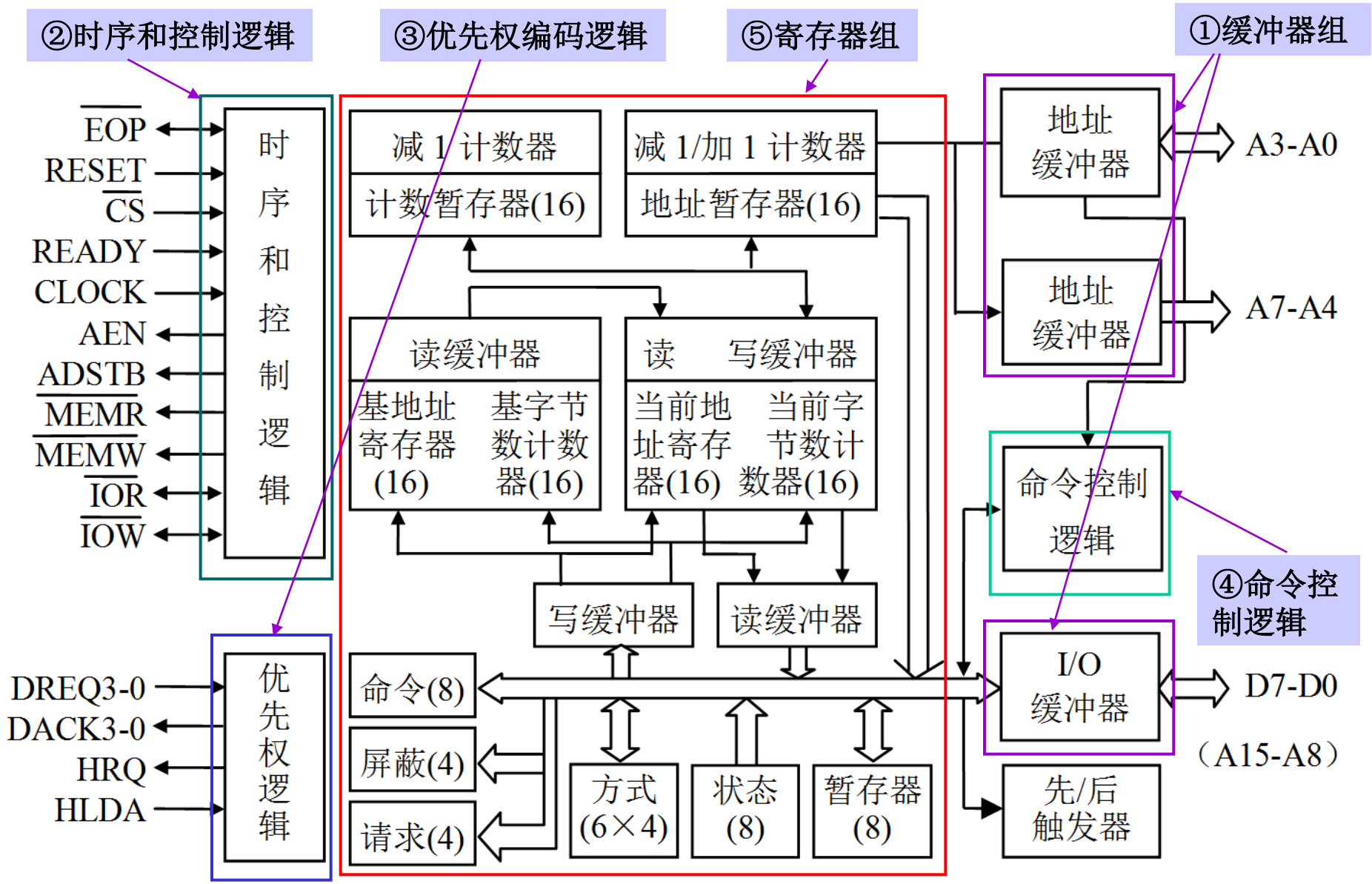
8．DMA操作启动有两种方法：一是外设输入DMA请求信号DREQ，二是通过软件编程从内部启动。

2.4.1 8237A的内部结构及引脚功能

- 8237A内部有4个结构相同的独立通道，图中只画出了一个通道的结构。
- 每个通道都有：基地址寄存器、基字节数计数器、当前地址寄存器，当前字节数计数器，方式字寄存器。



2.4.1 8237A的内部结构及引脚功能

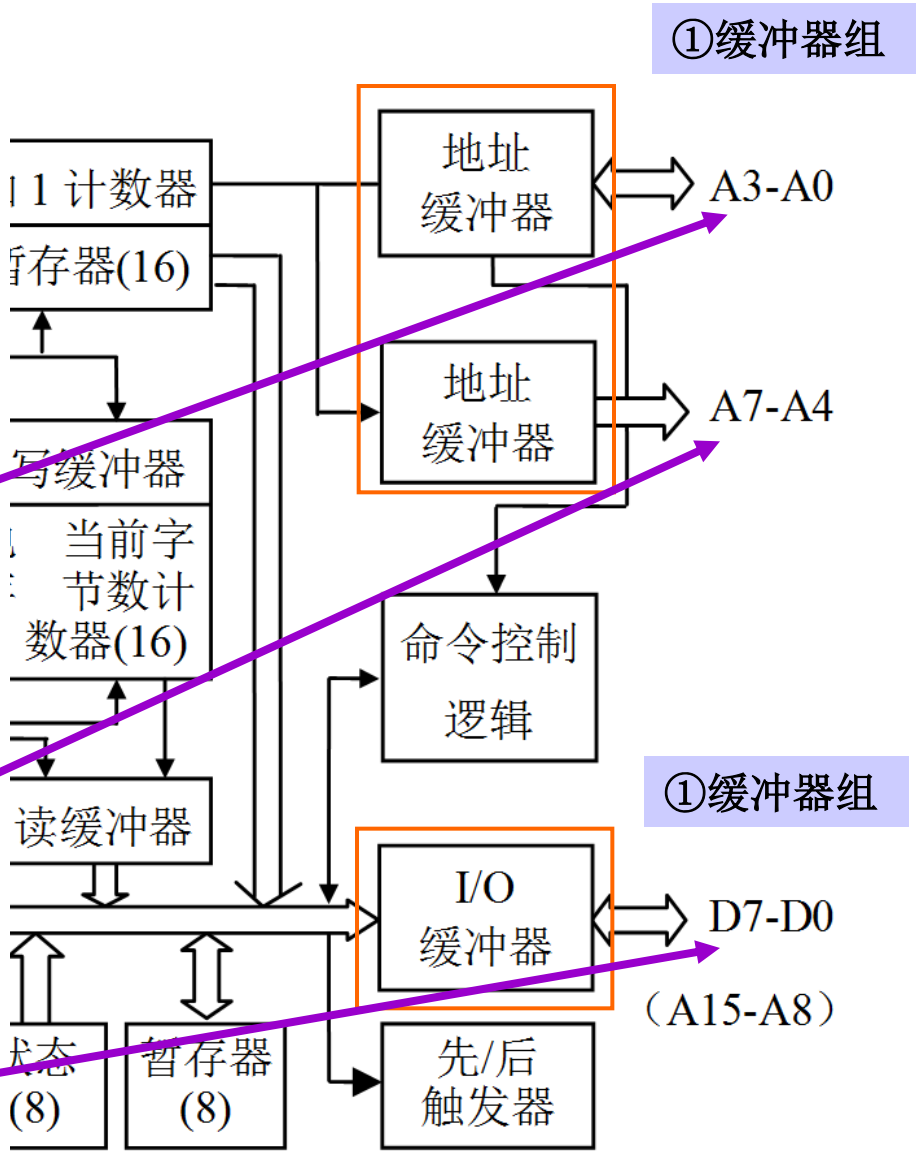


2.4.1 8237A的内部结构及引脚功能

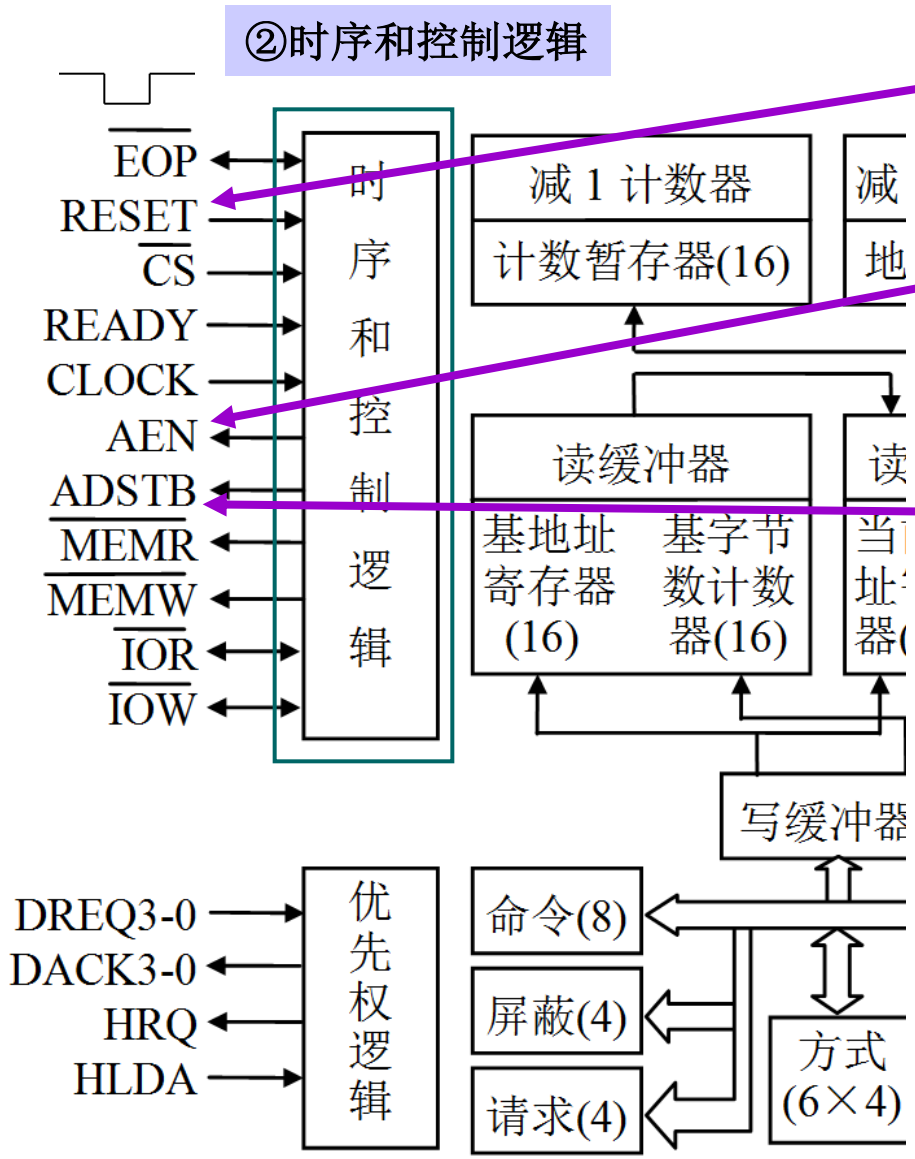
（1）A3-A0：输入/输出，双向。
（2）在空闲周期，它们是地址输入线，CPU用这四根地址线选择8237A内部不同的寄存器。
（3）在DMA操作周期，这四条线是输出，用于提供要访问的存储单元的地址。

A7-A4：输出，仅在DMA操作周期内使用，提供要访问的存储单元的地址。

D7-D0：输入/输出，双向，与系统数据总线相连。



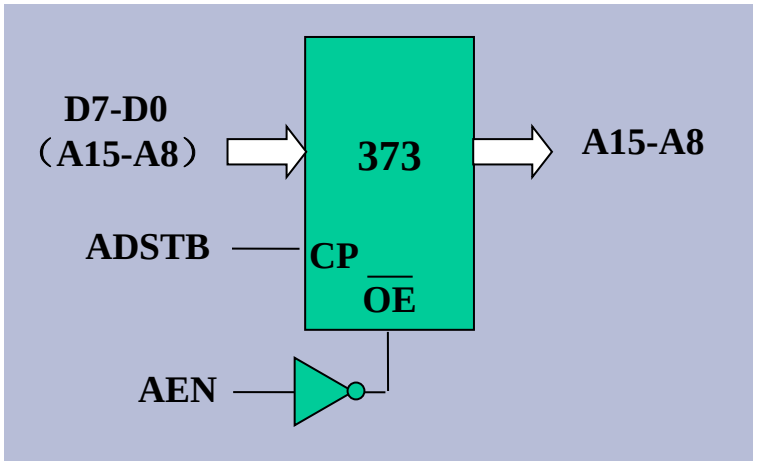
2.4.1 8237A的内部结构及引脚功能



RESET后，屏蔽寄存器为1，其他为0。

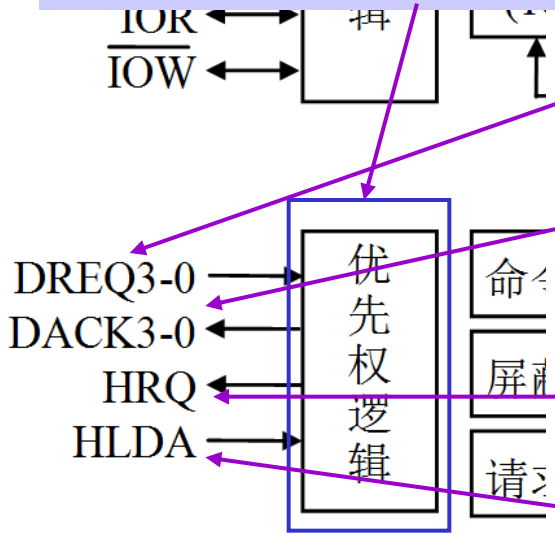
AEN地址允许，DMA期间一直有效。

ADSTB地址选通，可接到373输入脉冲，锁存8237的高8位地址A15-A8



2.4.1 8237A的内部结构及引脚功能

（1）该逻辑对同时提出的DMA请求的多个通道进行优先级排队。
（2）2种优先级编码：固定优先级编码、循环优先级编码。它们均可通过软件编程选定。
（3）固定优先级编码：四个DMA通道的优先级顺序是固定的，即通道0优先级最高，依次降低是通道1，通道2，通道3。
（4）循环优先级编码：最近一次服务的通道被指定为最低优先级。
（5）不论采用哪种优先级编码，经判优某个通道获得服务后，其他通道，不管优先级是高是低，均被禁止，直到已服务的通道结束为止。也就是说，不允许“DMA服务嵌套”。



DREQ3-DREQ0：DMA请求输入信号，DREQ的有效电平可由编程设定。

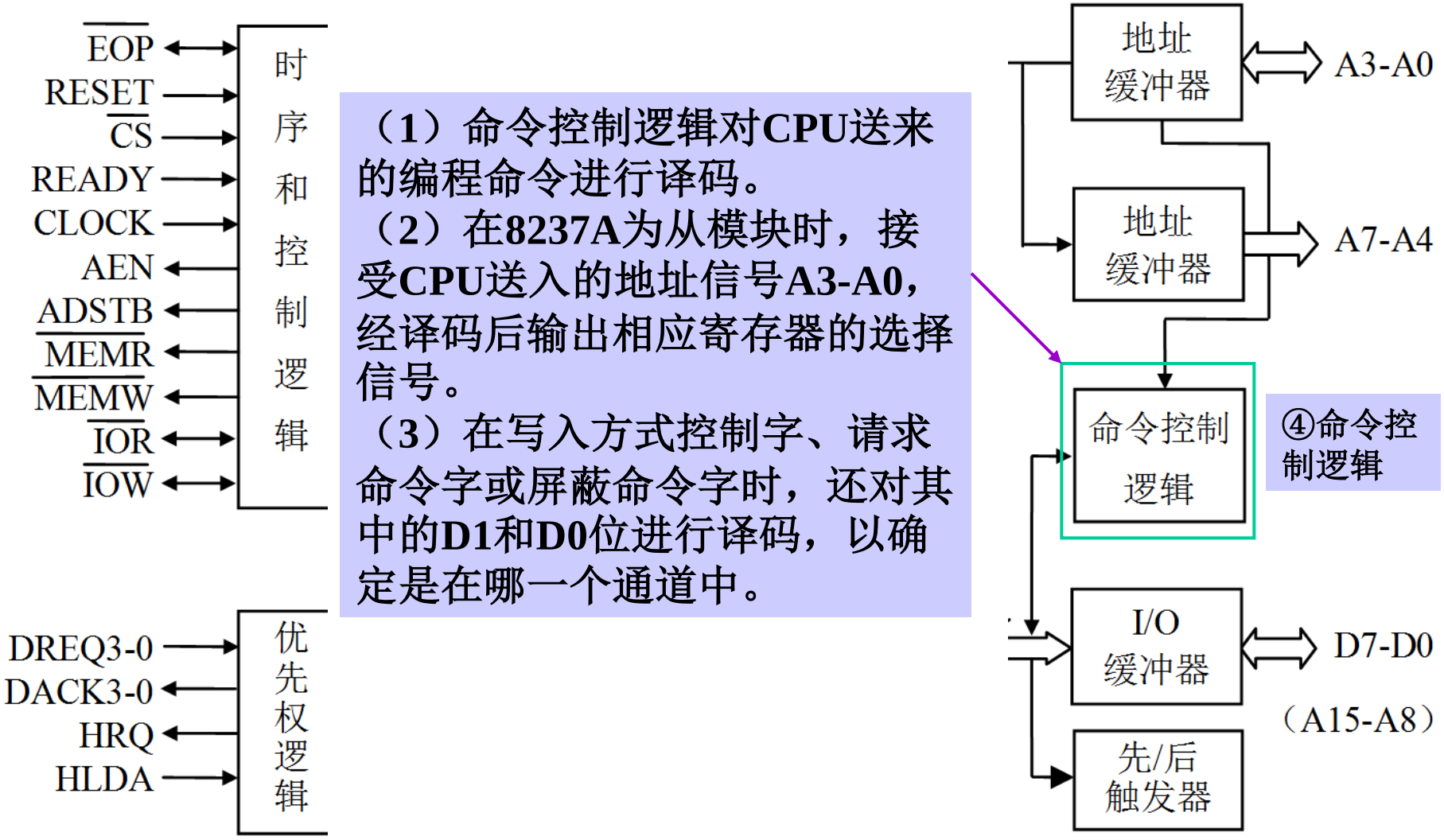
DACK3-DACK0：DMA响应输出信号，DACK的有效电平可由编程设定。

HRQ：总线请求信号，高电平有效。

HLDA：总线保持响应信号，高电平有效。

③优先权编码逻辑

2.4.1 8237A的内部结构及引脚功能

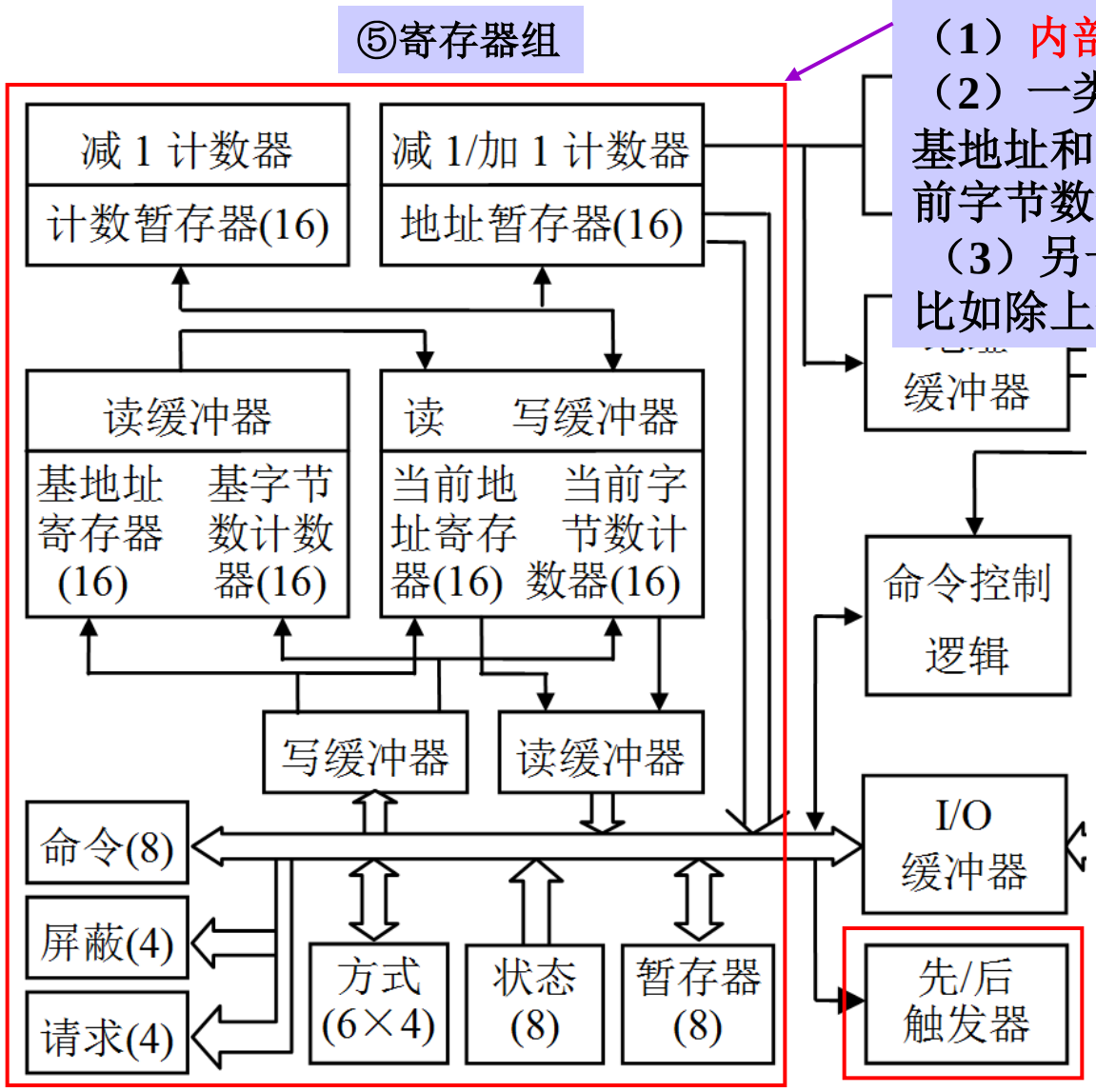


（1）命令控制逻辑对CPU送来的编程命令进行译码。

（2）在8237A为从模块时，接受CPU送入的地址信号A3-A0，经译码后输出相应寄存器的选择信号。

（3）在写入方式控制字、请求命令字或屏蔽命令字时，还对其中的D1和D0位进行译码，以确定是在哪一个通道中。

2.4.1 8237A的内部结构及引脚功能



（1）内部寄存器组分成两大类。

（2）一类是4个通道都有的寄存器，比如基地址和当前地址寄存器，基字节数和当前字节数计数器，方式字寄存器等。

（3）另一类是4个通道共用的一套寄存器，比如除上述以外的其他寄存器。

共有12种寄存器：

1）基地址寄存器	16位×4
2）基字节数计数器	16位×4
3）当前地址寄存器	16位×4
4）当前字节数计数器	16位×4
5）方式字寄存器	6位×4
6）暂存地址寄存器	16位×1
7）暂存字节数计数器	16位×1
8）状态寄存器	8位×1
9）命令寄存器	8位×1
10）暂存寄存器	8位×1
11）屏蔽寄存器	4位×1
12）请求寄存器	4位×1

2.4.2 8237A的工作方式

- 从DMA操作过程的角度，DMA控制器8237A的工作方式：

- （1）主从模态
- （2）传送方式
- （3）传送类型
- （4）优先级编码
- （5）自动初始化方式
- （6）存储器到存储器的传送

1．主从模态

- DMA控制器既可以作为I/O端口接受CPU的读写操作，也可以代替CPU占有总线，控制外设与存储器之间传送数据，它充分体现了DMA控制器的两大特性，即总线的主控性和总线的从属性，按这两大特性，它也就有两种工作模态：主态方式和从态方式。
- （1）主态方式
- 在主态方式时，DMA控制器是总线的控制者，此时，8237A是主模块，它如同CPU一样，掌握总线的控制权，可对涉及的外设端口或存储器单元进行读写操作。
- （2）从态方式
- 在从态方式时，CPU是总线的控制者，而DMA控制器不过是普通的一个外部设备，有若干个端口而已，它的地位同一般的I/O接口芯片是一样的，所以，此时8237A是从模块。

2．传送方式

- 8237A通过编程，可选择4种传送方式：

- 单字节传送方式
- 数据块传送方式
- 请求传送方式
- 级联传送方式

（1）单字节传送方式

- 单字节传送方式时，一次只传送一个字节，然后释放总线。若又有外设DMA请求，8237A再向CPU发下一次总线请求HRQ，获得总线控制权后，再传送下一个字节数据。
- 注意：

1）在DACK有效之前，DREQ应保持有效。

2）即使DREQ在传送过程中一直保持有效，在总线响应后HRQ也将变成无效，并在传送一个字节后DMA控制器释放总线，但由于DREQ一直有效，HRQ很快再次变成有效，在芯片接受到新的HLDA后，下一个字节又开始传输。显然，在两次DMA传送之间至少执行一个完整的机器周期，在此期间，完全可能响应另一个高优先级的DMA请求。

3）每次传送后，当前字节数计数器减1，当前地址寄存器减1或加1，当当前字节数计数器减1由0变成FFFFH时，发出有效信号（产生终止计数TC信号），如果通道编程设为自动初始化方式，则自动地重新装入计数值和地址寄存器。

（2）数据块传送方式

- 数据块传送方式时，响应一次DMA请求，将完成设定的字节数的全部传送。当字节数计数器减1由0变为FFFFH时，产生TC有效信号，使8237A将总线控制权交还给CPU从而结束DMA操作方式，外部有效的信号也可以终结DMA传送。
- 在DACK变成有效之前，DREQ信号必须保持有效。一旦DACK有效，不管DREQ如何，8237A一直不放弃总线控制权。即便是在传送过程中，DREQ变为无效，8237A也不会释放总线，只是暂停数据的传送，等到通道请求信号再次有效后，8237A又继续进行数据传送，直到整块数据全部传完，才会退出DMA操作，将总线控制权交还CPU。
- PC机不能采用这种方式，否则会影响动态存储器刷新和磁盘驱动器的数据传送，它们都不允许另一个DMA传送长期占用总线。

（3）请求传送方式

- 请求传送方式又称查询方式，类似数据块传送，但每传送一个字节后，检测DREQ状态，若无效则停止，若有效则继续DMA传送。在下述情况之一发生时，将停止传送：

  1）DREQ变为无效。

  2）字节数计数器减1由0变为FFFFH，产生TC信号。

  3）外界输入有效信号。

- 当DREQ无效时，8237A停止传送，内部的当前地址寄存器和当前字节数计数器还保留当时的数值，一旦外设准备好要传送的新数据，可以再次使DREQ变为有效，就可以使传送继续下去。当DREQ无效时，8237A停止传送，此时释放总线，当DREQ重新有效时，将重新开始一次DMA请求过程。

（4）级连方式

- 这种方式允许连接一个以上的芯片来扩展DMA通道的个数。
- 连接方法：将扩展的DMA芯片的HRQ和HLDA分别连到主片的某个通道的DREQ和DACK。
- 主片的连接通道起两个作用：

（1）优先级连接的作用，即将从片的4个DMA通道纳入到主片的优先级管理机制。

（2）向CPU输出HRQ和传递HLDA。

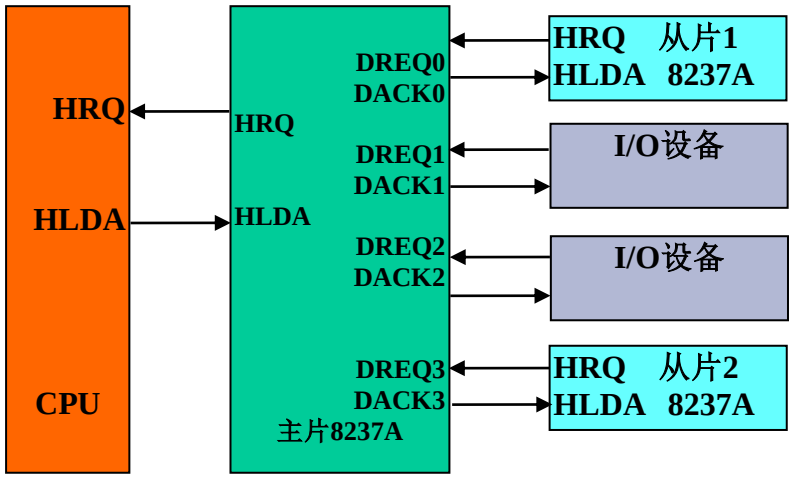


图2.4.2 8237A级联方式示意图

3．传送类型

- 三种传送类型：
- （1）DMA读
- 8237A输出有效的$\overline{\text{MEMR}}$和$\overline{\text{IOW}}$信号，把存储器的数据读到I/O设备。
- （2）DMA写
- 8237A输出有效的$\overline{\text{IOR}}$和$\overline{\text{MEMW}}$信号，把I/O设备的数据写到存储器。
- （3）DMA校验
- 这是一种伪传输，实际上是校验8237A芯片内部的读写功能，也就是对读传输功能或写传输功能进行检验。
- 在这种传送类型中，8237A芯片的操作如同DMA读和DMA写一样，产生地址信号以及对响应等，但对存储器和I/O设备的控制线均处于无效状态，禁止实际传送。DMA校验传输功能是器件测试时才使用的，一般的使用者对这项功能并不感兴趣。

4. 优先级编码

- 8237A有2种优先级编码：固定优先级、循环优先级。

- 固定优先级中4个通道的优先级顺序是固定的，DREQ0最高，DREQ3最低。
- 循环优先级中4个通道的优先级顺序是可变的，但其变化仍有一定的规律。当某一个通道申请DMA请求并被响应服务后，它就被指定为最低优先级，它的下一级就成为最高优先级。
- 值得注意的一点是，无论在任何情况下，DMA请求禁止嵌套服务。当一个通道的DMA请求被响应并服务后，其它3个通道的DMA请求将都被禁止，无论它们的优先级是高还是低。优先权排队只在DMA响应之前有效，DMA响应之后则无效。

5. 自动初始化方式

- 通过对方式字寄存器的编程，可设置某个通道为自动初始化方式。
- 自动初始化方式：当该通道完成一个数据传送并产生信号时（可能是由内部的TC产生，也可能是外部产生），用基地址寄存器和基字节数计数器的内容，使相应的当前地址寄存器和当前字节数计数器恢复初值。
- 当前地址寄存器和当前字节数计数器的最初值，是由CPU在初始化编程时写入的（这个最初值同时也写入到基地址寄存器和基字节数计数器），但在DMA传送过程中，当前地址寄存器和当前字节数计数器的内容被不断修改，而基地址寄存器和基字节数计数器的内容维持不变（除非重新编程）。在自动初始化以后，通道就作好了进行另一次DMA传送的准备。

6. 存储器到存储器的传送

- 利用这种方式，可以使数据块从一个存储空间传送到另一个存储空间，将程序的影响和传输时间减到最小。

- 占用8237A的2个通道： #0、#1
- 传送方向：#0地址寄存器（源）→ #1地址寄存器（目的）
- 计数方法：采用#1字节数计数器
- 启动方法：设置一个#0的软件DREQ启动

2.4.3 8237A的工作时序

- 8237A的工作时序有3种：正常时序、压缩时序、扩展写时序
- 对应各种工作时序有两类工作周期：空闲周期和操作周期（有效周期） 。
- 全部工作周期分为7种时钟状态（时钟周期）：

（1）空闲状态SI：没有DMA请求时，8237A就处于空闲周期 。进行两种检测：

1）有无CS信号，以确定有无CPU对8237A的操作要求。

2）有无DREQ信号，以确定是否有I/O设备送来有效的DMA请求。

（2）起始状态S0：过渡过程。

（3）传送状态S1：准备高8位地址。

（4）传送状态S2：为存储器提供16位地址。

（5）传送状态S3：输出读控制信号。

（6）传送状态S4：输出写控制信号。

（7）等待状态SW：协调外设与存储器之间的传送速度 。

1. 工作周期

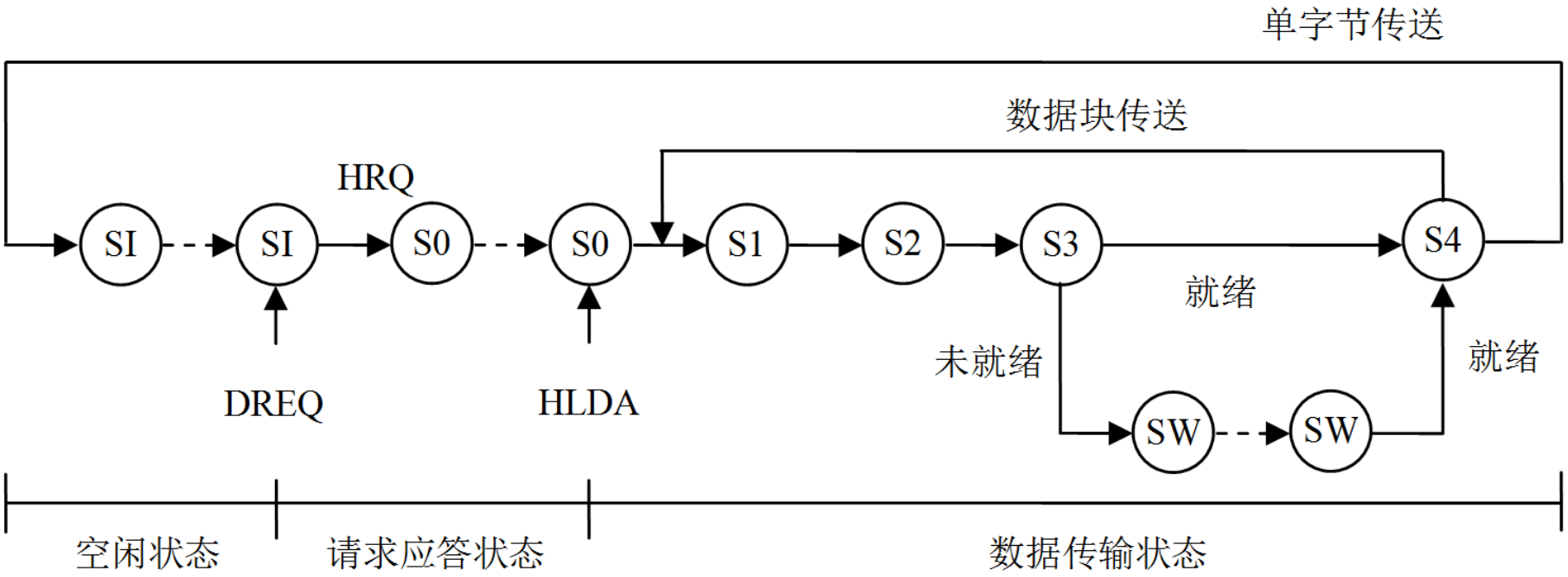


图 2.4.3　8237A 工作时序 7 种状态

2. 正常时序

- 8237A DMA控制器可选择正常时序、压缩时序和扩展写时序等操作时序。

- 不同操作时序的实质是在控制读、写脉冲发出的时间与时钟信号CLK的对应关系。

- 正常时序：传送一个字节数据包含4个时钟脉冲周期，即S1-S4状态。产生的读写脉冲信号与这4个状态有确定的对应关系。若是数据块传送中不改变高8位地址，则省去S1，只占用S2、S3、S4三个时钟周期。

3．压缩时序

- 压缩时序方式所占用的脉冲数将减少。
- 压缩时序：把读命令的宽度压缩到等于写命令的宽度，省掉了S3，即由S4完成读和写的操作。
- 所以，在压缩时序方式下传送一个字节数据需要占用3个时钟周期，即S1、S2、S4，而在大多数情况下高8位地址并不改变，于是省掉了S1，因此，在数据块传送中大多数情况占用2个时钟周期，即S2和S4。
- 此时用S2状态修改低8位地址值，用S4状态完成读和写的操作，也就是把正常时序中S3和S4二个状态的功能压缩在一个状态中完成。
- 由于压缩时序传送类型只用2个状态完成一个数据字节的传送，因此它具有更高的数据传送速率。

4. 扩展写时序

- 在正常时序操作下，可选择扩展写方式，即写命令提前到读命令，从S3状态开始（一般情况下，读为S3、S4状态，写为S4一个状态）。

- 也就是说，写命令同读命令一样，扩展为2个时钟周期。

2.4.4 8237A的编程

- 8237A依靠它的可编程特性实现它的各种工作方式的选择和设定。
- 8237A在HLDA信号处于无效的任何时间里，即使HRQ有效，也可以接受CPU对它的编程。CPU对8237A的编程初始化工作是通过8237A的端口进行的。

- 8237A的端口是用A3-A0低4位地址线编址的，共有16个端口地址，其具体编址情况如表2.4.1所示。
- 假设我们以DMA代表16个端口地址的首地址，那么写通道2基字节数计数器的端口地址可表示为DMA＋05H，写方式寄存器的端口地址可表示为DMA＋0BH。

8237A内部端口地址分配

表 2.4.1　8237A 内部端口地址分配

	A3	A2	A1	A0	寄存器说明
通道0	0	0	0	0	通道 0　写：基地址寄存器和当前地址寄存器 通道 0　读：当前地址寄存器
	0	0	0	1	通道 0　写：基字节数计数器和当前字节数计数器 通道 0　读：当前字节数计数器
通道1	0	0	1	0	通道 1　写：基地址寄存器和当前地址寄存器 通道 1　读：当前地址寄存器
	0	0	1	1	通道 1　写：基字节数计数器和当前字节数计数器 通道 1　读：当前字节数计数器
通道2	0	1	0	0	通道 2　写：基地址寄存器和当前地址寄存器 通道 2　读：当前地址寄存器
	0	1	0	1	通道 2　写：基字节数计数器和当前字节数计数器 通道 2　读：当前字节数计数器
通道3	0	1	1	0	通道 3　写：基地址寄存器和当前地址寄存器 通道 3　读：当前地址寄存器
	0	1	1	1	通道 3　写：基字节数计数器和当前字节数计数器 通道 3　读：当前字节数计数器
	1	0	0	0	写：命令寄存器；　读：状态寄存器
	1	0	0	1	写：请求寄存器；　读：非法
	1	0	1	0	写：一位屏蔽字寄存器；　读：非法
	1	0	1	1	写：方式寄存器；　读：非法
	1	1	0	0	写：清先/后触发器软件命令；　读：非法
	1	1	0	1	写：8237A 复位软件命令；　读：暂存寄存器
	1	1	1	0	写：清屏蔽寄存器软件命令；　读：非法
	1	1	1	1	写：四位屏蔽字寄存器；　读：非法

（1）初始化编程命令字总共有8个。
（2）控制命令有5个：方式字、命令字、请求字、屏蔽字、状态字。
（3）软件命令有3个。

1. 控制命令字

- **方式字：**写入端口地址0BH，选择传送方式和传送类型，设置自动初始化方式和地址增量方向。
- **命令字：**写入端口地址08H，选择DREQ、DACK有效极性，读写时序，优先级编码方式等。
- **请求字：**写入端口地址09H，发生软件DMA请求。
- **屏蔽字：**写入端口地址0AH或0FH，允许或禁止通道的DMA请求。
- **状态字：**从端口地址08H读出，反映通道DMA请求状态和是否有TC信号。

方式字

方式字

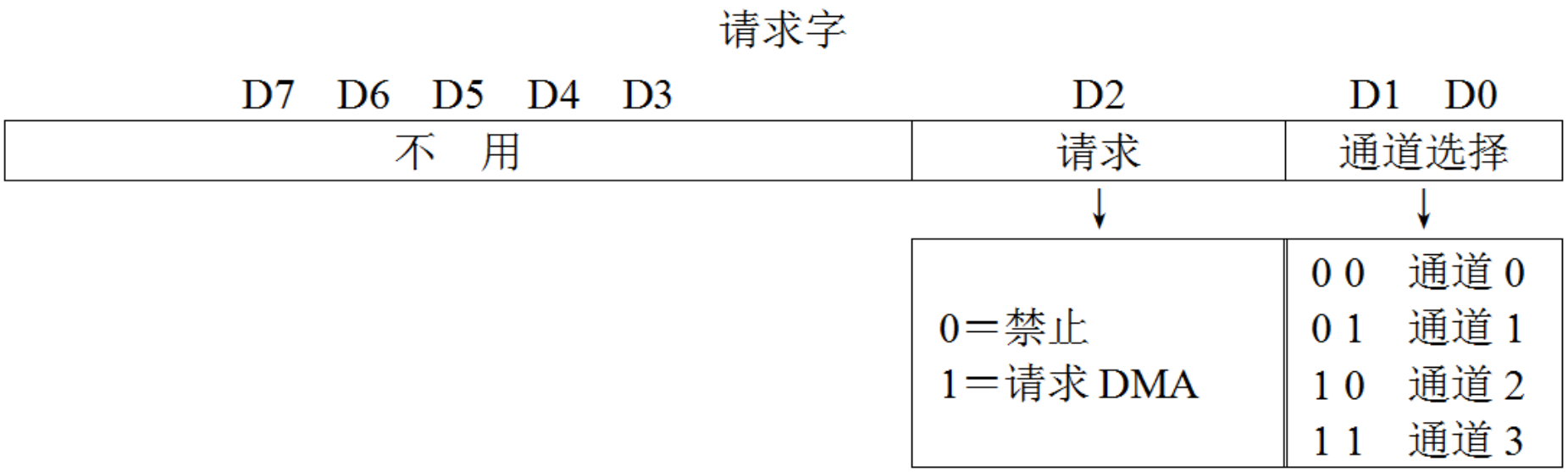
D7 D6	D5	D4	D3 D2	D1 D0
传送方式	地址增减	自动初始化	传送类型	通道选择
↓	↓	↓	↓	↓
0 0 请求方式 0 1 单字节方式 1 0 数据块方式 1 1 级联方式	0＝递增 1＝递减	0＝禁止 1＝允许	0 0 DMA 校验 0 1 DMA 写 1 0 DMA 读 1 1 无效	0 0 通道 0 0 1 通道 1 1 0 通道 2 1 1 通道 3

命令字

命令字

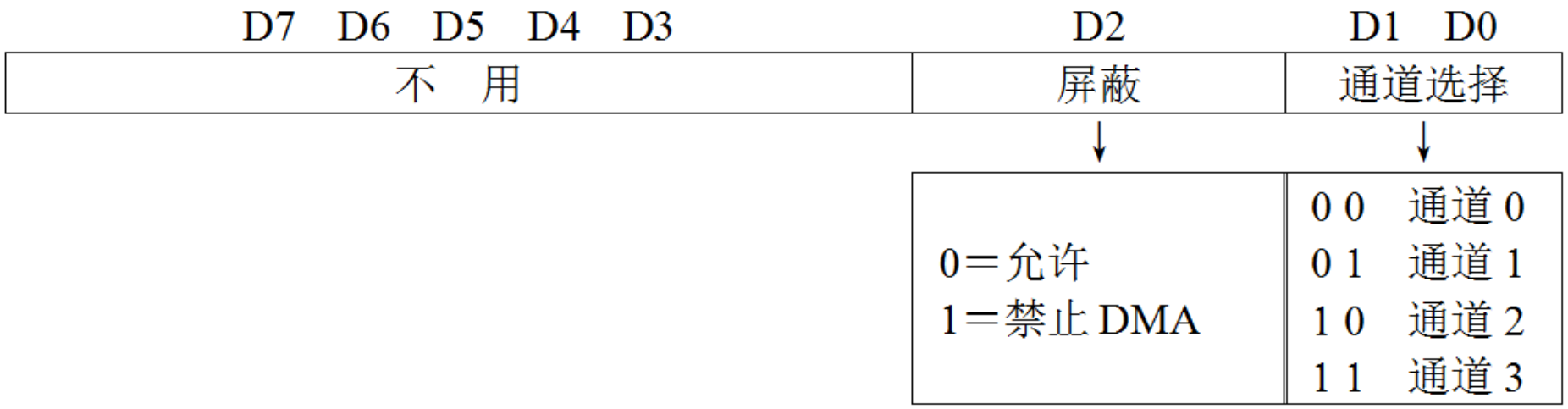
D7	D6	D5	D4	D3	D2	D1	D0
DACK 极性	DREQ 极性	写入 选择	优先级 编码	读写 时序	工作 允许	通道 0 保持	存储器 间传送
↓	↓	↓	↓	↓	↓	↓	↓
0＝低电平有效 1＝高电平有效	0＝高电平有效 1＝低电平有效	0＝滞后写入(先读后写) 1＝扩展写入(同时读写)	0＝固定 1＝循环	0＝正常 1＝压缩	0＝允许 1＝禁止	0＝禁止 1＝允许	0＝禁止 1＝允许

请求字

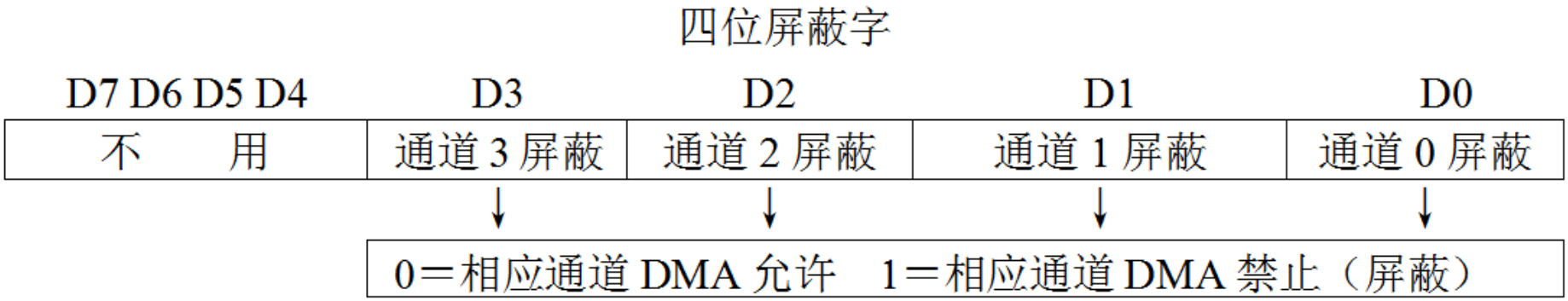

请求字
D7 D6 D5 D4 D3
D2
D1 D0
不 用
请求
通道选择
0＝禁止
1＝请求 DMA
0 0 通道 0
0 1 通道 1
1 0 通道 2
1 1 通道 3

一位屏蔽字

一位屏蔽字


D7 D6 D5 D4 D3
D2
D1 D0
不 用
屏蔽
通道选择
0＝允许
1＝禁止 DMA
0 0 通道 0
0 1 通道 1
1 0 通道 2
1 1 通道 3

四位屏蔽字


四位屏蔽字
D7 D6 D5 D4
D3
D2
D1
D0
不　　用
通道 3 屏蔽
通道 2 屏蔽
通道 1 屏蔽
通道 0 屏蔽
0＝相应通道 DMA 允许　1＝相应通道 DMA 禁止（屏蔽）

状态字

状态字

D7	D6	D5	D4	D3	D2	D1	D0
通道 3	通道 2	通道 1	通道 0	通道 3	通道 2	通道 1	通道 0
↓	↓	↓	↓	↓	↓	↓	↓
0＝无 DMA 请求 1＝有尚未处理的 DMA 请求				0＝未结束 1＝已接收到终止结束信号			

2．清除命令

- 软件命令：清除命令有3条。这三条命令与数值无关，不需通过数据总线，即执行输出指令时，AL的内容可随便设置，只要对特定的端口地址执行一次写操作，依靠这个地址和控制信号，命令就生效。这就是所谓的“软件命令”。
- （1）清先/后触发器软件命令（写入端口地址0CH）
- 即清除字节指针命令，是专为16位寄存器的读/写而设置的。因为数据线是8位，所以16位数据要分两次读/写。而且要使用同一个端口地址。为区分两个高低字节，8237A设置了先/后触发器作为字节指针，为0时对应低字节，为1时对应高字节。每次读/写操作，字节指针自动翻转一次。系统复位后，先/后触发器被清0。
- 清先/后触发器软件命令使字节指针（先/后触发器）被清0。
- （2）复位软件命令（写入端口地址0DH）
- 此命令与硬件的RESET信号功能相同。
- （3）清屏蔽寄存器软件命令（写入端口地址0EH）
- 其功能是将4个通道的屏蔽位清除，允许它们接受DMA请求。

3．8237A的编程步骤

- （1）CPU发复位软件命令；
- （2）写入基地址及当前地址值；
- （3）写入基字节数和当前字节数初值；
- （4）写入方式字；
- （5）写入屏蔽字；
- （6）写入命令字；
- （7）写入请求字，可用软件DMA请求启动通道，也可在（1）－（6）完成以后，等待外部DREQ请求信号。

例2.4.1 8237A数据块传送

- 设在某8088系统中，用8237A通道1将内存1000H单元开始的24K字节数据转存到软盘之中（暂不考虑20位地址的问题，可认为1000H就是基地址的初值）。采用数据块方式传送，地址增量方式，只传送一遍，设DREQ和DACK低电平有效，当A15-A4=0000 0000 0111时选中8237A，要求设计8237A通道1的初始化程序。

- 1．端口地址
- A3-A0由8237A芯片内部译码，编码范围是从0000到1111，再与A15-A4组合，则端口地址范围是0070H-007FH。
- 2．传送字节数
- 24K字节对应16进制数为6000H，但写入通道字节数计数器的值应为6000H-1=5FFFH，因为TC的产生不是在计数器由1到0的跳变处，而是在计数器由0到FFFFH的跳变处。所以写入的计数初值应比实际字节数少一个。
- 3．方式字
- 按题目要求，方式字的组合为：1000 1001B
- 4．一位屏蔽字
- 按题目要求，一位屏蔽字的组合为：0000 0001B
- 5．命令字
- 按题目要求，命令字的组合为：0100 0000B

初始化程序

```
START:  MOV     DX，007DH       ；发复位软件命令
        OUT     DX，AL
        MOV     DX，0072H
        MOV     AL，00H
        OUT     DX，AL          ；送基地址和当前地扯低8位
        MOV     AL，10H
        OUT     DX，AL          ；送基地址和当前地址高8位
        MOV     DX，0073H
        MOV     AL，0FFH        ；送基计数值和当前计数值低8位
        OUT     DX，AL
        MOV     AL，5FH         ；送基计数值和当前计数值高8位
        OUT     DX，AL
        MOV     DX，007BH
        MOV     AL，89H         ；写入方式控制字，DMA读传送
        OUT     DX，AL
        MOV     DX，007AH
        MOV     AL，01H         ；写入屏蔽字
        OUT     DX，AL
        MOV     DX，0078H
        MOV     AL，40H         ；写入命令控制字
        OUT     DX，AL
```

2.4.5 8237A在PC机中的应用

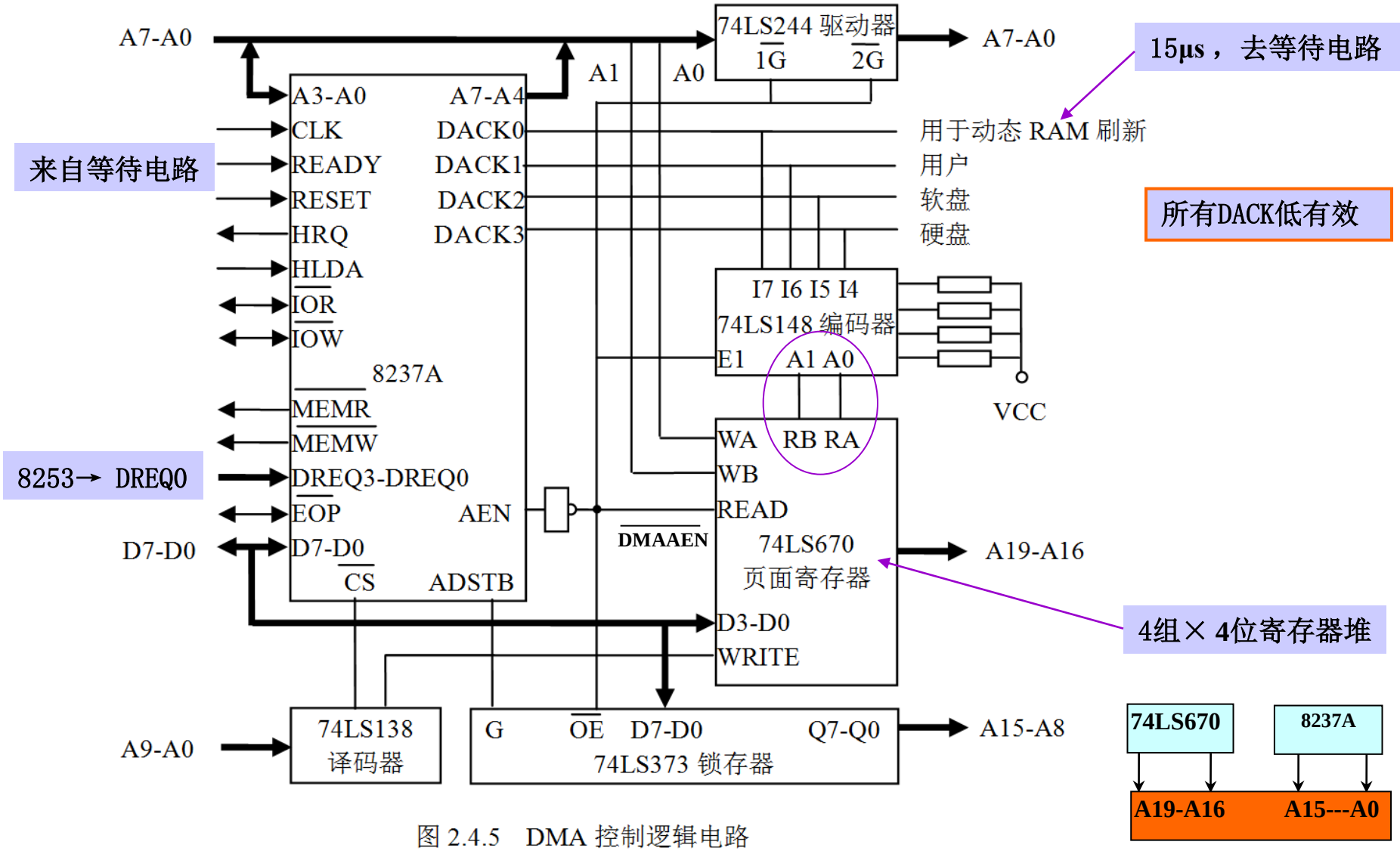


图 2.4.5　DMA 控制逻辑电路

74LS148的引脚和逻辑关系

输入控制，低有效

A1-RB，A0-RA，A2不用

优先标志

输出允许

输入									输出				
E1	I0	I1	I2	I3	I4	I5	I6	I7	A2	A1	A0	S	E0
1	×	×	×	×	×	×	×	×	1	1	1	1	1
0	1	1	1	1	1	1	1	1	1	1	1	1	0
0	×	×	×	×	×	×	×	0	0	0	0	0	1
0	×	×	×	×	×	×	0	1	0	0	1	0	1
0	×	×	×	×	×	0	1	1	0	1	0	0	1
0	×	×	×	×	0	1	1	1	0	1	1	0	1
0	×	×	×	0	1	1	1	1	1	0	0	0	1
0	×	×	0	1	1	1	1	1	1	0	1	0	1
0	×	0	1	1	1	1	1	1	1	1	0	0	1
0	0	1	1	1	1	1	1	1	1	1	1	0	1

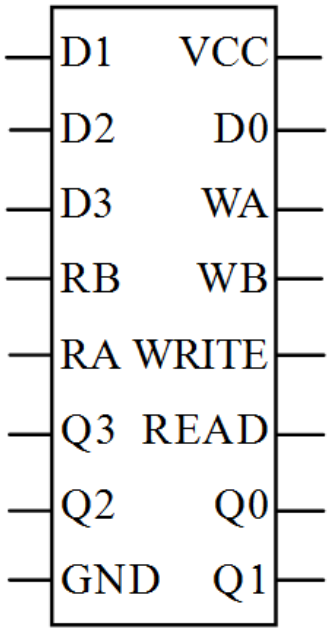
DACK0→ I7
DACK1→ I6
DACK2→ I5
DACK3→ I4

可连接下一级编码器的E1，对更多输入端编码

图 2.4.6　74LS148 的引脚和逻辑关系

优先级I7高，I0小

74LS670的引脚和逻辑关系



写控制

每个寄存器4位，保存高4位地址

WRITE	WB	WA	功能
0	0	0	写入 0 组寄存器
0	0	1	写入 1 组寄存器
0	1	0	写入 2 组寄存器
0	1	1	写入 3 组寄存器

对应的DMA通道

#0

#1

#2

#3

READ	RB	RA	功能
0	0	0	读出 0 组寄存器
0	0	1	读出 1 组寄存器
0	1	0	读出 2 组寄存器
0	1	1	读出 3 组寄存器

读控制

读、写时，对应寄存器的编号

图 2.4.7　74LS670 的引脚和逻辑关系

（1）初始化程序段

```
        MOV     AL，04
        OUT     08H，AL          ；输出控制命令，关闭8237A，使它不工作
        OUT     0DH，AL          ；发总清命令
        MOV     DX，00H          ；通道0地址寄存器对应的端口号
        MOV     CX，0004         ；4个通道
X1：    MOV     AL，0FFH
        OUT     DX，AL           ；写入地址低位，先/后触发器在总清时已清除
        OUT     DX，AL           ；写入地址高位，这样16位地址为0FFFFH
        ADD     DX，2            ；指向下一个通道的地址寄存器
        LOOP    X1               ；使4个通道的地址寄存器中均为0FFFFH
        MOV     AL，58H          ；方式字
        OUT     0BH，AL          ；对通道0进行方式选择，单字节传输，DMA读
                                 ；地址加l变化，设置自动初始化功能
        MOV     AL，41H
        OUT     0BH，AL          ；对通道1设置方式，单字节传输，DMA校验，
                                 ；地址加1变化，无自动初始化功能
        MOV     AL，42H
        OUT     0BH，AL          ；对通道2设置方式，同通道1
```

```
MOV     AL，43H
OUT     0BH，AL          ；对通道3设置方式，同通道1
MOV     AL，0            ；命令字
OUT     08H，AL          ；对8237A设控制命令，DACK为低电平有效，
                         ；DREQ为高电平有效，固定优先级，启动工作
OUT     0AH，AL          ；使通道0去除屏蔽
INC     AL
OUT     0AH，AL          ；使通道1去除屏蔽
INC     AL
OUT     0AH，AL          ；使通道2去除屏蔽
INC     AL
OUT     0AH，AL          ；使通道3去除屏蔽
```

（2）测试程序段

```
        MOV     DX，2           ；2是通道1的地址寄存器端口
        MOV     CX，0003        ；测试3个通道
X1：    IN      AL，DX          ；读地址的低位字节
        MOV     AH，AL
        IN      AL，DX          ；读地址的高位字节
        CMP     AX，0FFFFH      ；比较读取的值和写入的值是否相等
        JNZ     X2              ；如不等，则转X2
        ADD     DX，2           ；指向下一个通道
        LOOP    X1              ；测下一个通道
        ┆                       ；后续测试
X2：    HLT                     ；如出错，则停机等待
```

测试的过程：写入-读出-比较-判断

第2章 结 束