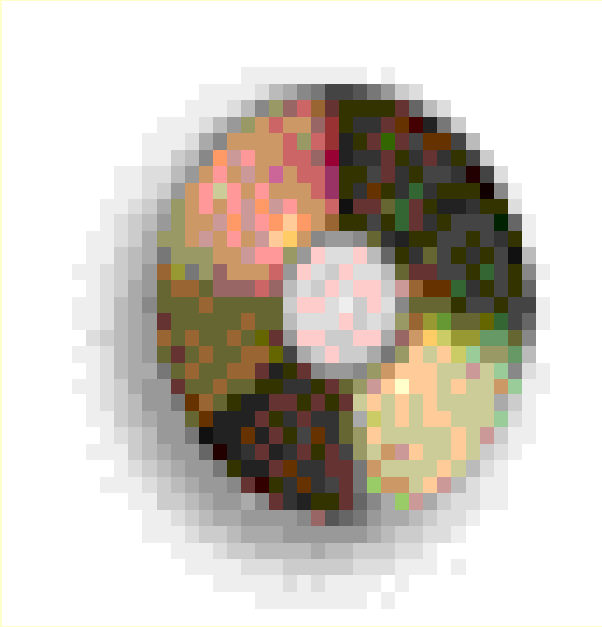


# 第十一章 DMA技术



一、直接数据通道传送

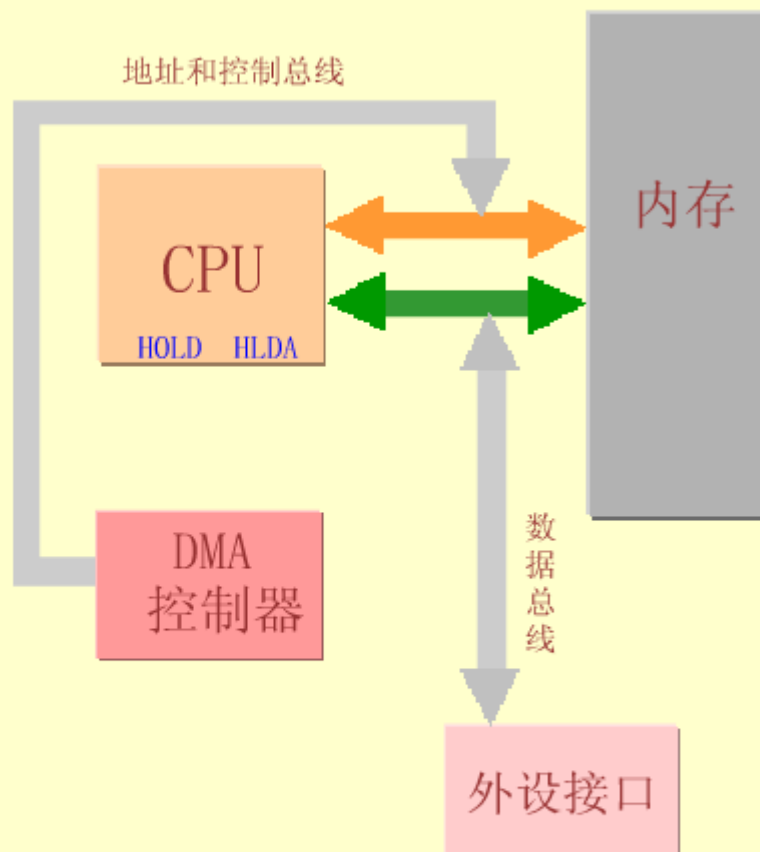
二、常用DMAC芯片— Intel 8237

三、IBM PC/XT中的 DMA 控制逻辑

四、应用举例

# 一、直接数据通道传送（直接存储器存取）

## 1、DMA方式传送过程



说明：

播放 停止



## **DMA 可以实现：**

内存与I/O之间；

内存与内存之间；

I/O 与 I/O 之间                      的数据传送。

## **DAM方式的特点：**

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。

如磁盘与存储器之间的传输速度每秒高达几兆字节，这是采用中断方式（用CPU执行指令传输数据）所实现不了的。

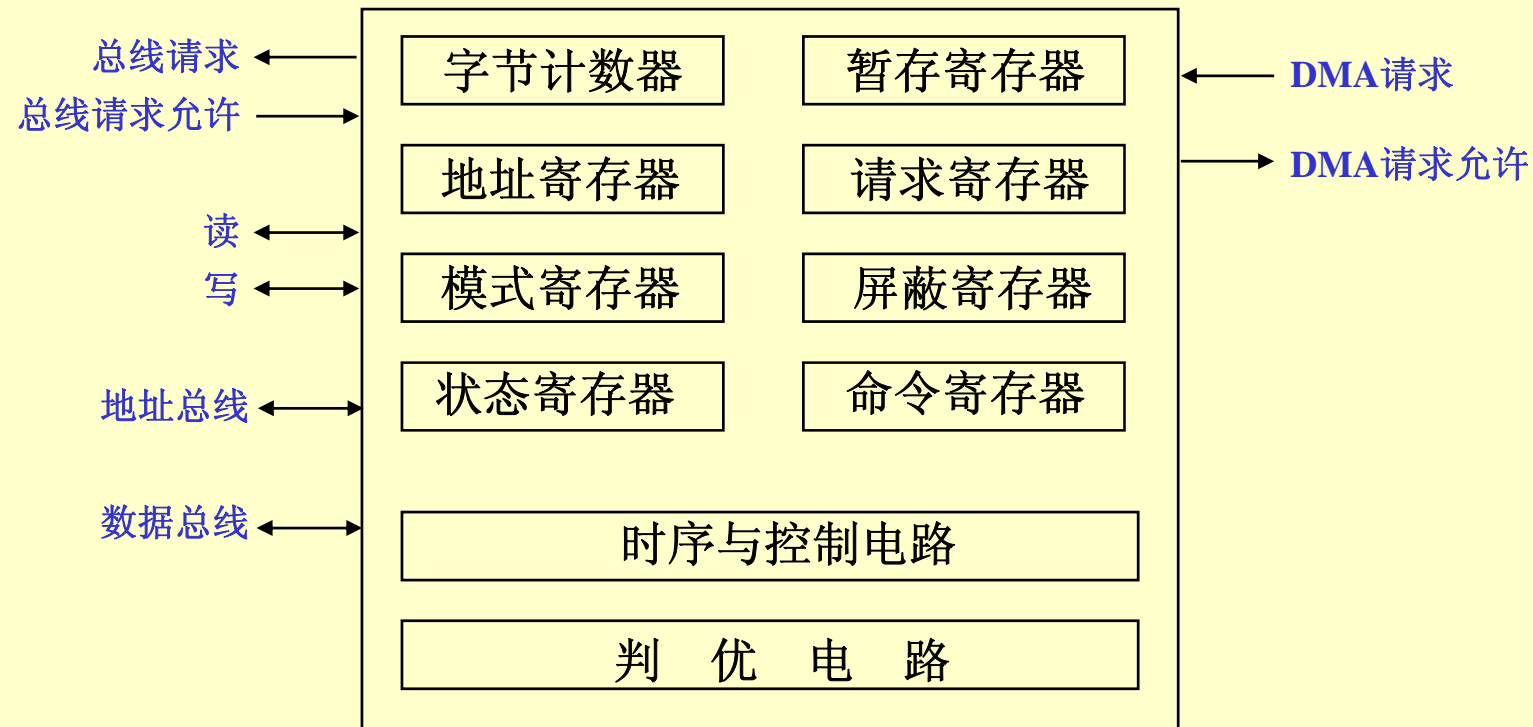


## 2、DMA控制器 (DMAC) 的功能

1. 接收CPU发出的读写信号——初始化;
  2. 接收外设发来的DMA请求;
  3. 向CPU发出总线请求信号;
  4. 接收CPU发出的总线请求允许信号;
  5. 向外设发出DMA请求允许信号;
  6. 发出地址信息, 能对存储器寻址及修改地址指针;
  7. 能发出读/写等控制信号;
  8. 能提供数据的传输数量;
  9. 能撤销总线请求信号, 使CPU恢复正常工作。
- } 控制总线



### 3、DMAC 的基本组成



**模式寄存器：**表示DMA的工作模式：单次传送；数据块传送；DMA校验等。

**命令寄存器：**表示完成存储器与I/O的传送还是其他传送（存储器与存储器之间）；  
**DMA请求信号高电平还是低电平有效等。**

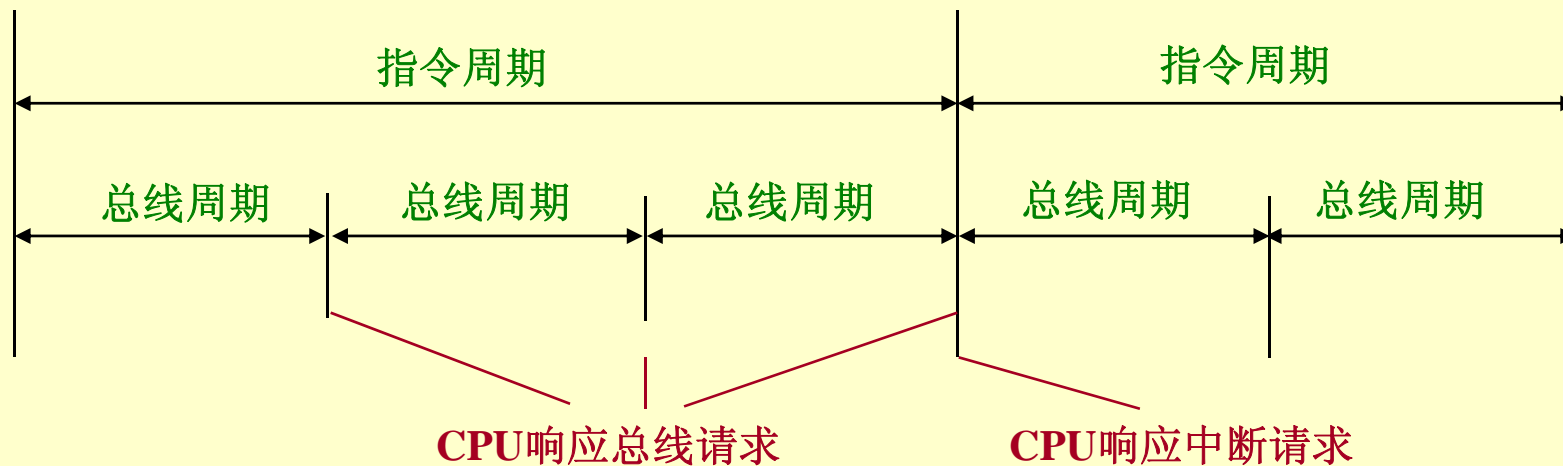
**请求寄存器：**DMAC通常可允许多路DMA请求，请求寄存器寄存各路的DMA请求。



#### 4、CPU对DMAC总线请求的响应

当要进行DMA传送时，DMAC向CPU提出总线请求，那么CPU什么时候给予相应？

——在每个总线周期的最后一个节拍，响应DMAC的总线请求。



由于一个指令周期通常包括几个总线周期，因此，CPU对总线请求的响应比对中断请求的响应快得多。

## 5、DMAC占用总线的方式

- (1) CPU停止使用总线：DMAC从CPU手中获得总线控制权，使CPU停止使用总线。

由于内存速度通常高于外设，故在传送过程中总线将有很多空闲，故此方法适用于高速外设成组传送数据。

- (2) 周期挪用：DMAC取得控制权后，I/O设备挪用一到几个总线周期（CPU不访问存储器的那些周期）进行数据传送。

I/O设备每挪用一次总线周期，都要经历请求、使用、归还总线的过程，故适用于I/O读写周期大于内存周期的情况。

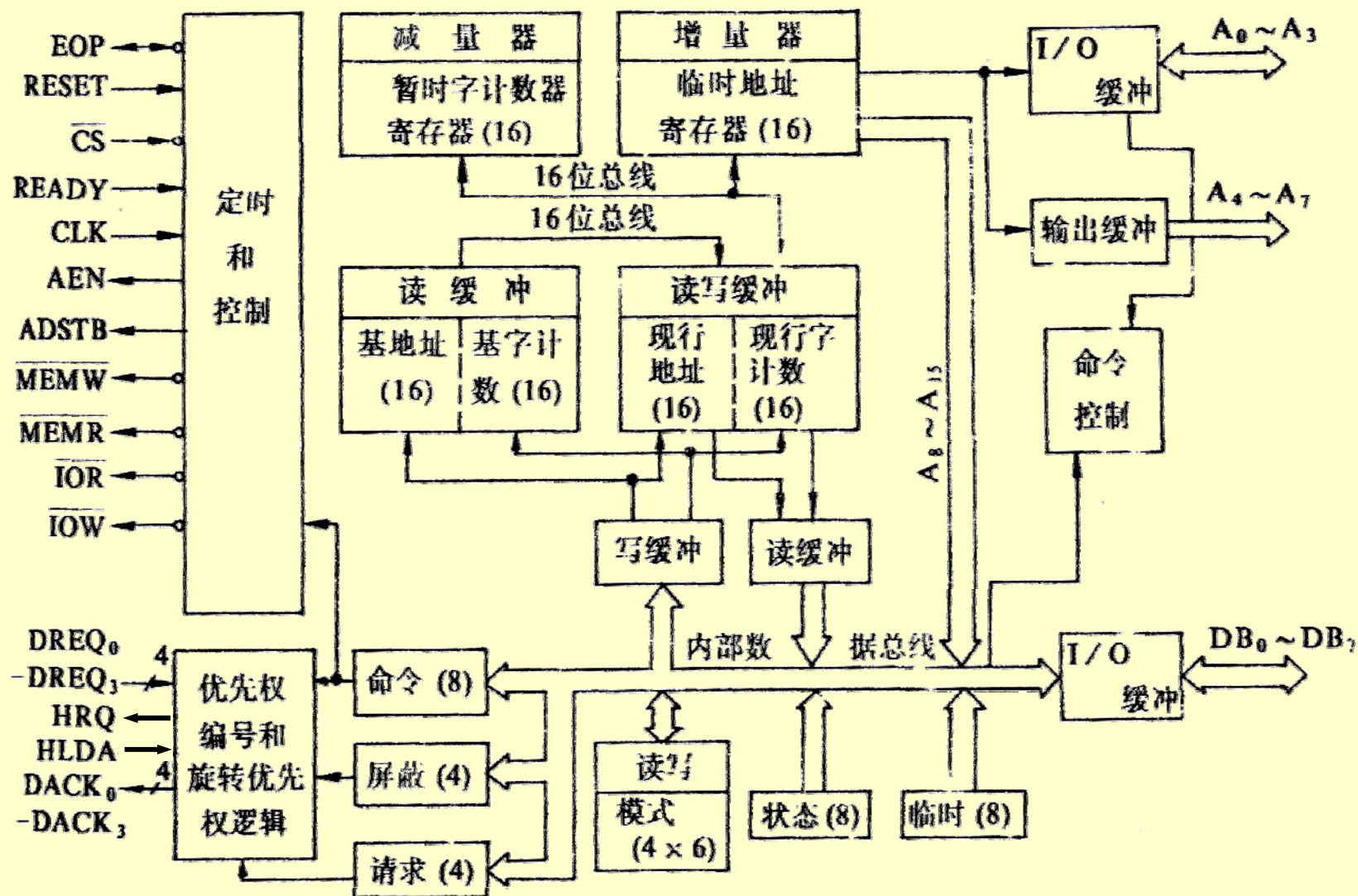
- (3) 周期扩展：将提供给CPU的时钟周期加宽，在加宽的时钟周期中进行DMA操作。

在这些方式中，CPU停止使用总线是最常用也是最简单的一种DMA传送方式。



## 二、常用DMAC芯片 —— Intel 8237

### 1. 基本结构及功能



### 2. 工作模式及其它





## (1) 时序和控制逻辑

- 作用：
- 接受外部时钟及片选信号；
  - 产生内部时序控制及读、写信号，地址信号。

信号： ADSTB —— 地址选通输出。将芯片内数据缓冲器送出的高8位地址送到外部地址锁存器锁存。

AEN —— 地址允许输出。将锁存的高8位地址送入系统地址总线，与低8位地址组成16为地址。

$\overline{\text{EOP}}$  —— DMA过程结束信号。 ① 外部输入； ② 内部传输结束产生。

$\left. \begin{array}{l} \overline{\text{MEMR}} \\ \overline{\text{MEMW}} \end{array} \right\}$  —— 存储器读、写信号。

$\left. \begin{array}{l} \overline{\text{IOR}} \\ \overline{\text{IOW}} \end{array} \right\}$  —— ① 空闲周期：用于CPU将编程信息送入芯片的内部寄存器。  
② 有效周期：从外设读出数据送入内存（IOR、MEMW）；  
从内存读出数据送入外设（IOW、MEMR）。



## (2) 优先级编码逻辑

作用： 对同时提出申请的多个通道进行优先级排队判优。

8237有两种优先级编码：

- 固定优先级编码 —— 0通道优先级最高，其次是1，2，3通道。
- 循环优先级编码 —— 最近一次服务的通道在下次循环中变为最低，依次循环。

\* 无论在哪种优先级编码中，经判优某个通道获得服务后，其他通道无论优先级高低，均被禁止，直至已服务的通道结束为止。

信号：  $DREQ_{0-3}$  —— DMA通道请求信号，外设向8237发出。

$DACK_{0-3}$  —— DMA应答输出信号，8237通知外设。

HRQ —— 总线请求输出信号，  
8237发出，向CPU请求总线控制权。

HLDA —— 总线保持响应输入信号，  
CPU的响应信号。



### (3) 程序命令控制逻辑

作用：对CPU送来的程序命令进行译码，

- 在芯片空闲周期，通过I/O地址缓存器送来的地址  $A_3-A_0$ ，分别对内部寄存器进行预置。

$A_3$ $A_2$ $A_1$ $A_0$	读操作	写操作
1 0 0 0	读状态寄存器	写命令寄存器
1 0 0 1	无效	写请求寄存器
1 0 1 0	无效	写单个通道屏蔽
.....		

- 在DMA有效周期，对模式控制字的最低两位 $D_3$ 、 $D_2$  进行译码，  
以确定DMA的操作类型。  
写传送、读传送、校验传送。



#### (4) 数据地址缓冲器组

由三部分组成:

**A<sub>0</sub>-A<sub>3</sub>**: 最低4位地址线, 在芯片空闲周期, 用作芯片内部寄存器寻址, 由CPU装入或读出数据。  
在芯片有效周期, 作为输出线提供低4位地址。

**A<sub>4</sub>-A<sub>7</sub>**: 在DMA有效周期, 提供高4位地址。

**DB<sub>0</sub>-DB<sub>7</sub>**: 芯片空闲周期, 在IOW命令下, 由CPU对内部寄存器编程写入 ;  
在IOR命令下, 将内部寄存器的值送至总线上。

芯片有效周期, 作为最高8位的地址信号经缓冲器输出到系统总线, 由ADSTB选通到外部锁存器, 与A<sub>0</sub>-A<sub>7</sub>组成16位地址



## (5) 内部寄存器组

### 8237 内部有12 种寄存器

#### 1. 基本字节数寄存器：16位、4个（每通道一个）

保存需传送数据的原始字节数，8237初始化时由CPU写入。

#### 2. 基地址寄存器：16位、4个（每通道一个）

保存数据传送的初始地址，8237初始化时由CPU写入。

#### 3. 现行地址寄存器：16位、4个（每通道一个）

保存DMA传送期间的地址值，每次传送后，地址自动加1或减1。

#### 4. 现行字节数计数器：16位、4个（每通道一个）

保存当前字节数，每传送一个字节后自动减1。

#### 5. 临时地址寄存器：16位，1个

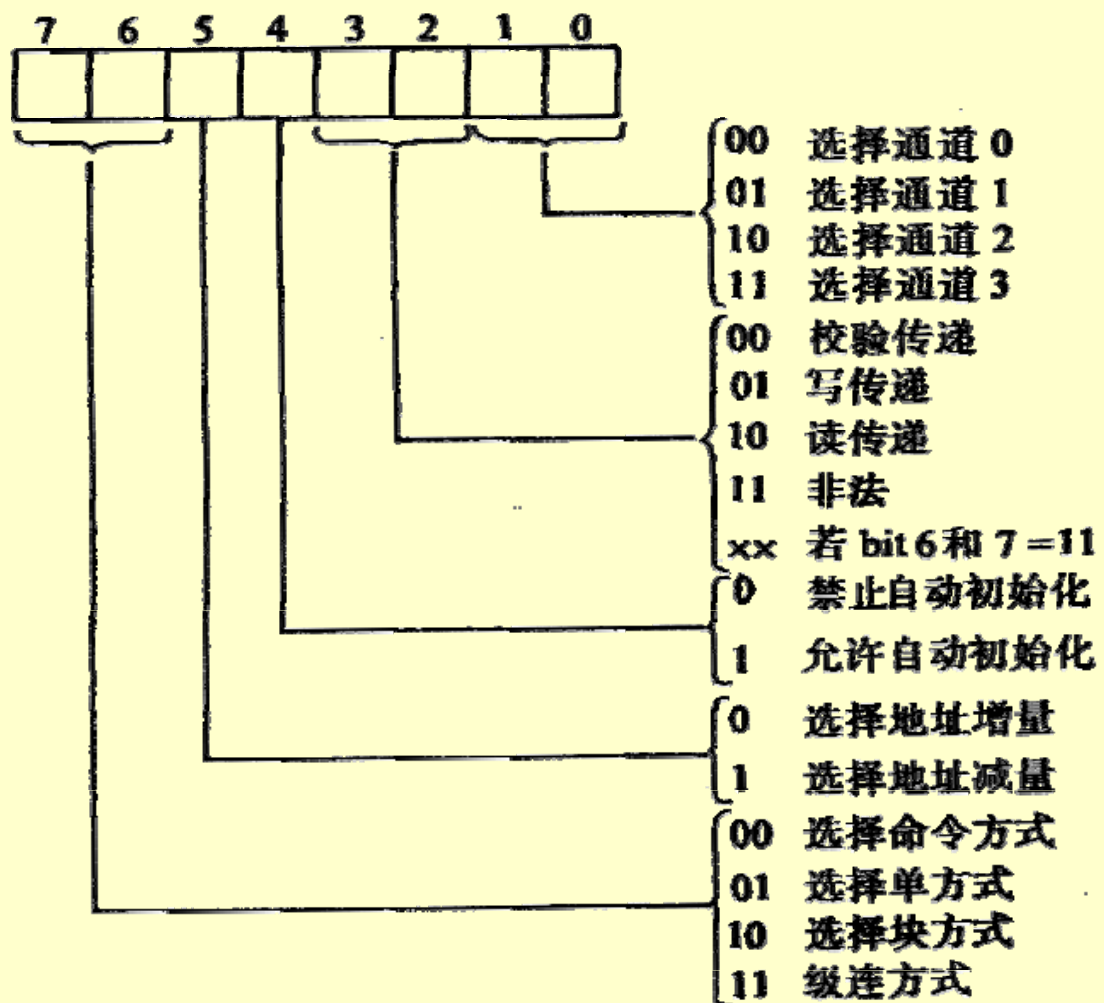
#### 6. 临时字节数寄存器：16位，1个

暂存当前地址和字节数，与CPU没有直接联系。



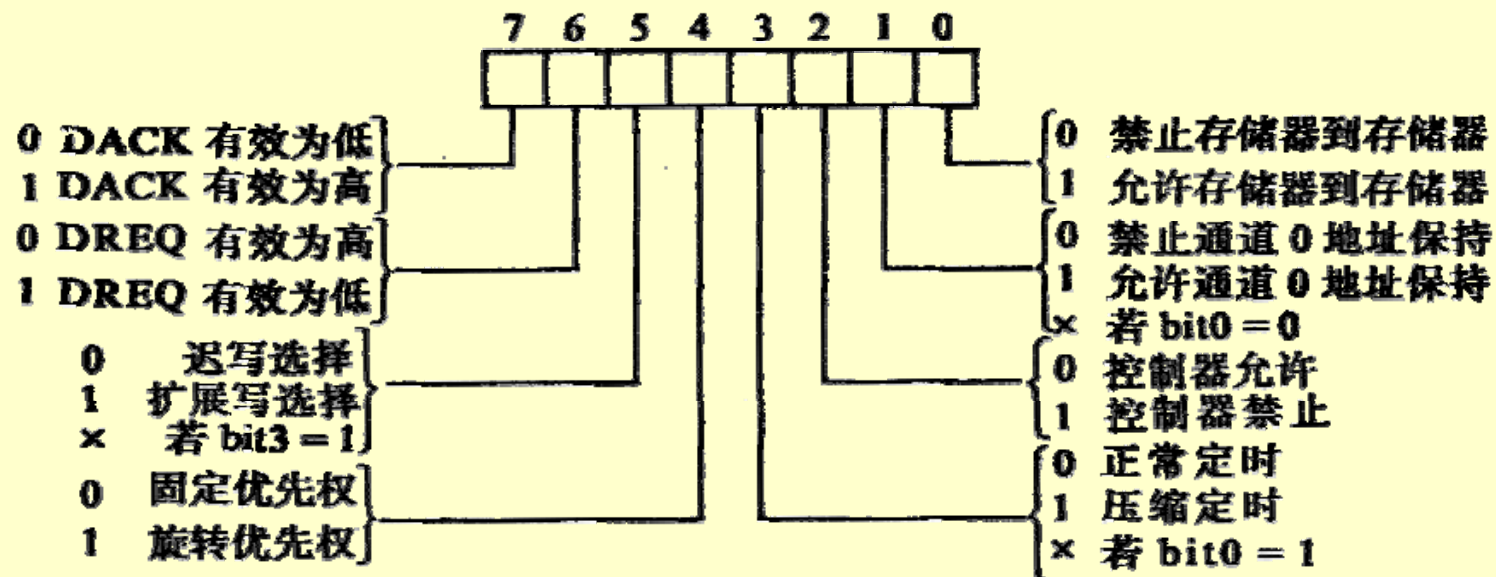
## 7. 模式寄存器: 6位、4个 (每通道一个)

寄存相应通道的模式控制字, 模式控制字初始化时由CPU写入, 以规定相应通道的操作模式。



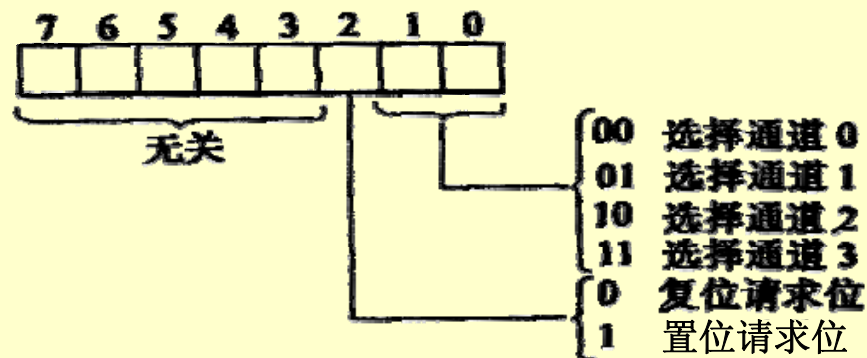
## 8. 命令寄存器: 8位, 1个

初始化时由CPU写入, 以规定8237的操作方式。



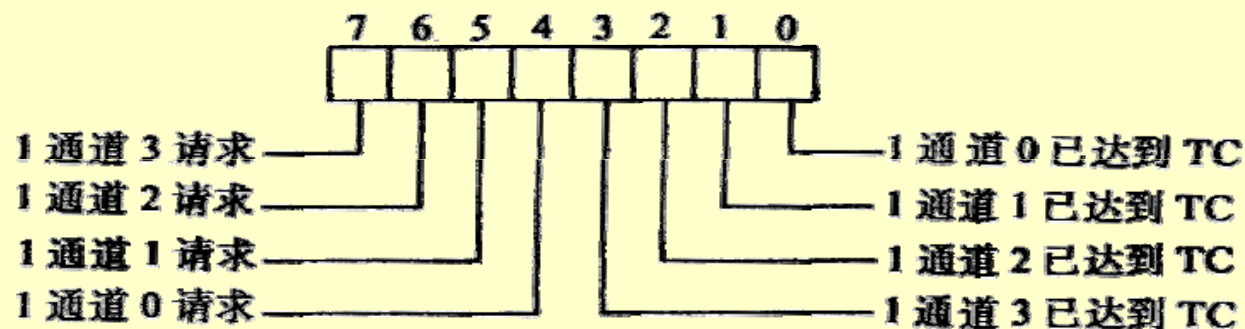
### 9. 请求寄存器: 4位, 1个

初始化时由CPU写入, 以便在软件控制下产生一个DMA请求, 如同外部设备产生DREQ请求一样。



### 10. 状态寄存器: 8位, 1个

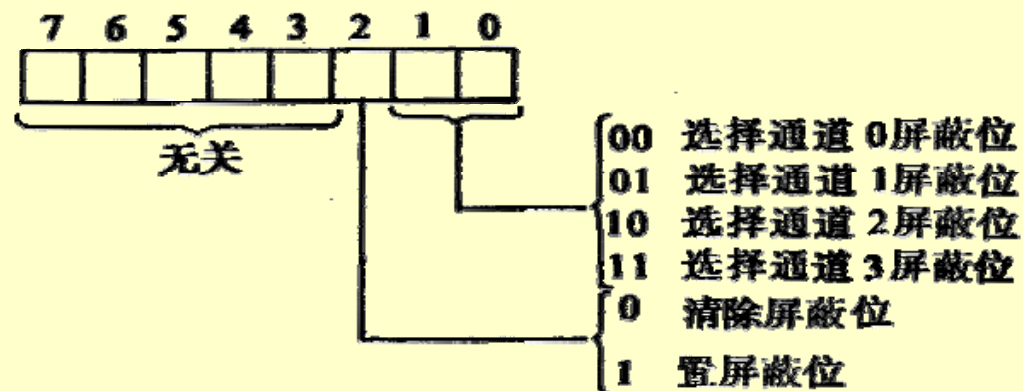
高4位表示哪个通道还有未处理的DMA请求, 低4位表示哪个通道已收到终止计数信号。



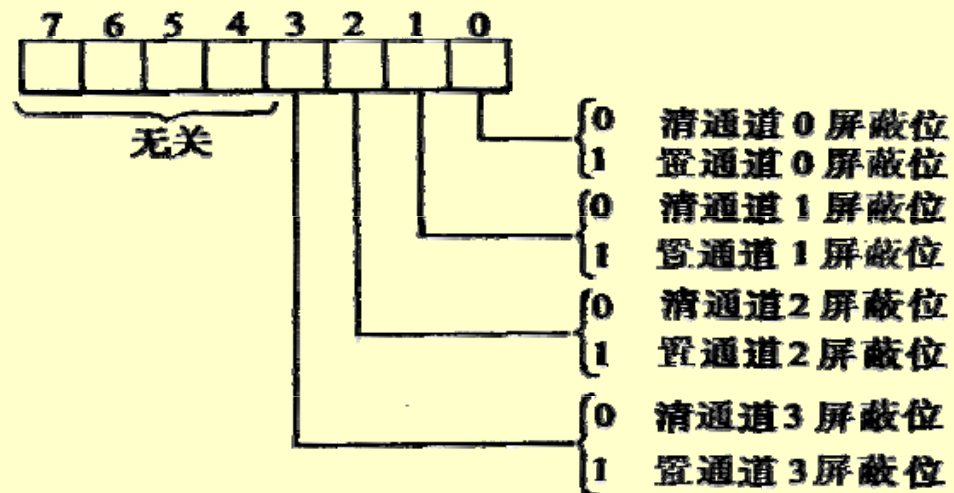


### 11. 屏蔽寄存器: 4位, 1个

用于选择允许或禁止各通道接受DMA请求信号DREQ。



也可使用其对各通道同时操作



### 12. 临时寄存器: 8位, 1个

仅用于存储器到存储器间的传输操作, 暂存其传输数据。



## 2. 8237的工作模式

8237在DMA传送时有四种工作模式：

### (1) 单字节传送模式

每次只传送一个字节，传送后，字节数寄存器减 1，地址寄存器加 1或减 1，并释放总线，将控制权交还 CPU。

### (2) 块字节传送方式

可多字节传输，直至字节数计数器由 0 减到 FFFFH时，产生一个终止计数信号 T/C，或由外界输入一个过程结束信号  $\overline{\text{EOP}}$  为止。

在PC机中，此方式不能采用，因为在这种方式下，在传送期间 8088 不能占用总线，因而别的DMA请求被禁止，这样系统动态存储器刷新等无法进行。

### (3) 请求传送方式

只要没有计数结束信号 T/C，或外界过程结束信号  $\overline{\text{EOP}}$ ，或者 DREQ 信号一直有效，DMA 传送可一直进行，直到外部设备传送完全部字节。

### (4) 级连方式

连接多个8237芯片，用于系统扩展。



### 3. 8237的传送类型

在单字节传送、块字节传送、请求传送中，均有三种不同的传送类型——读、写、校验。

**读传送**——通过有效的 $\overline{\text{MEMR}}$ 和 $\overline{\text{IOW}}$ ，将数据从存储器传送到I/O设备；

**写传送**——通过有效的 $\overline{\text{MEMW}}$ 和 $\overline{\text{IOR}}$ ，将数据从I/O设备传送到存储器；

**校验传送**——伪传送。实际上用来校验8237芯片内部的读写功能。

在这种传送中，8237芯片的操作如同读、写传送一样，产生地址及对EOP的响应等，但所有的存储器和I/O设备的控制信号均为无效。



## 4. 8237的工作周期

8237有两种工作周期——空闲周期和有效周期。

### (1) 空闲周期:

当没有I/O设备或软件请求DMA传送时，8237处于空闲周期。

在空闲周期内，芯片在每个时钟周期内都要进行两种检测：

- $\overline{CS}$  是否被选中. 以判断CPU是否要对8237内部寄存器进行读写操作(对8237进行初始化).  
若CS被选中，则8237变为CPU的一个外部设备，CPU可对其进行初始化，控制其工作方式。
- 有无 DMA 请求信号：若测出 DREQ 请求，则8237进入有效周期。

### (2) 有效周期:

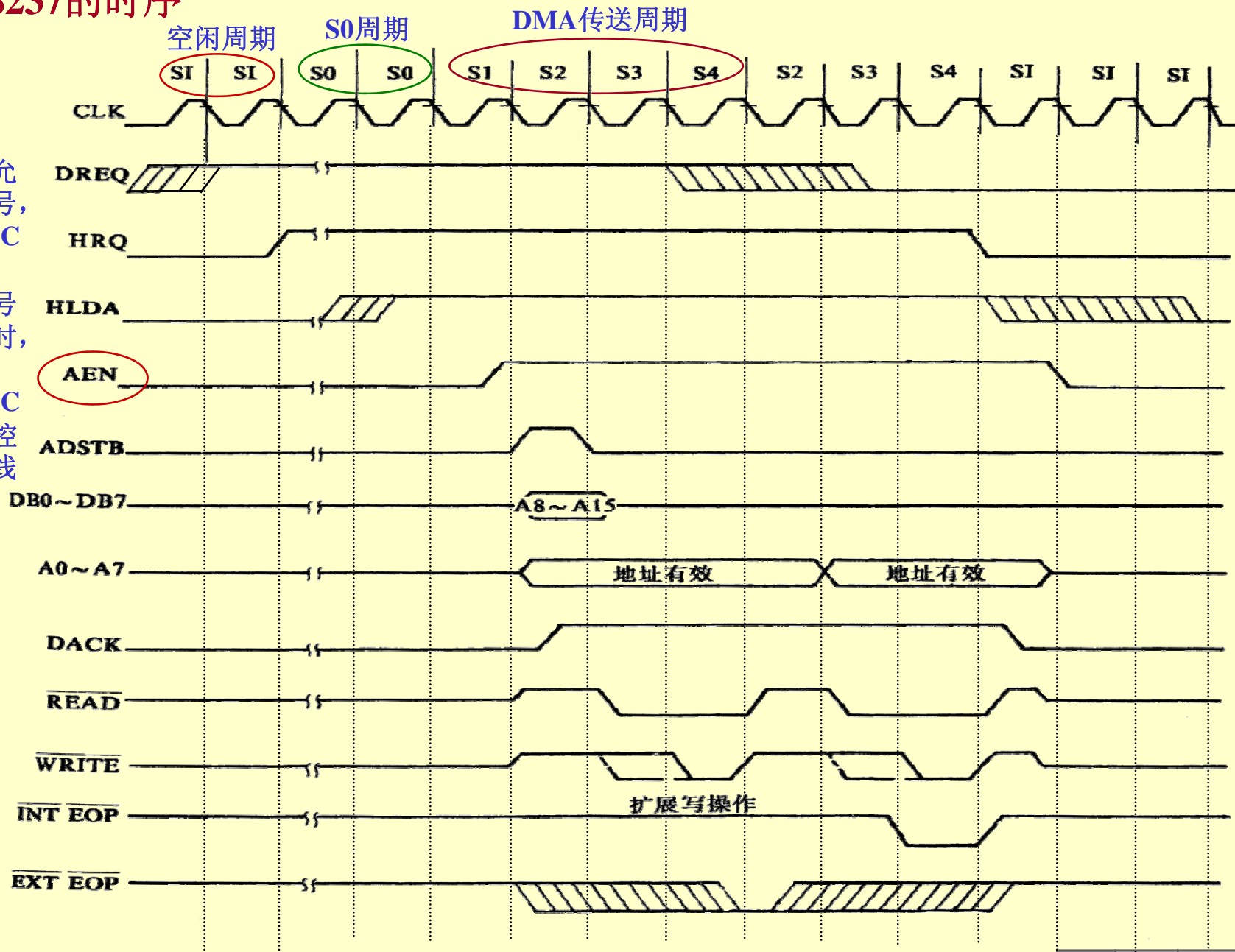
8237向CPU发出总线请求信号 HRQ，

- 只要 CPU 还没有送来有效的 HLDA 信号，8237仍处于初始化状态；
- 当 CPU 回答有效的 HLDA 信号以后，便开始 DMA 传送操作；传送结束后，8237进入空闲状态，等待下一个 DMA 传送请求或接受 CPU 的读写操作。



## 5. 8237的时序

地址允许信号, DMAC发出, 该信号有效时, 表示DMAC已经控制总线



(1) SI周期 —— 空闲周期: 8237 在复位以后, 即进入空闲周期, 再此周期内:

- CPU 可对 8237 编程, 进行初始化。
- 8237 采样 DREQ 线。

(2) S0周期 —— 在 SI 周期只要检测到有 DREQ 信号, 即进入S0 周期。

在 S0 周期 8237 等待 CPU 的响应信号, 只要未收到 HLDA 信号, 则 8237 一直处在S0 周期。

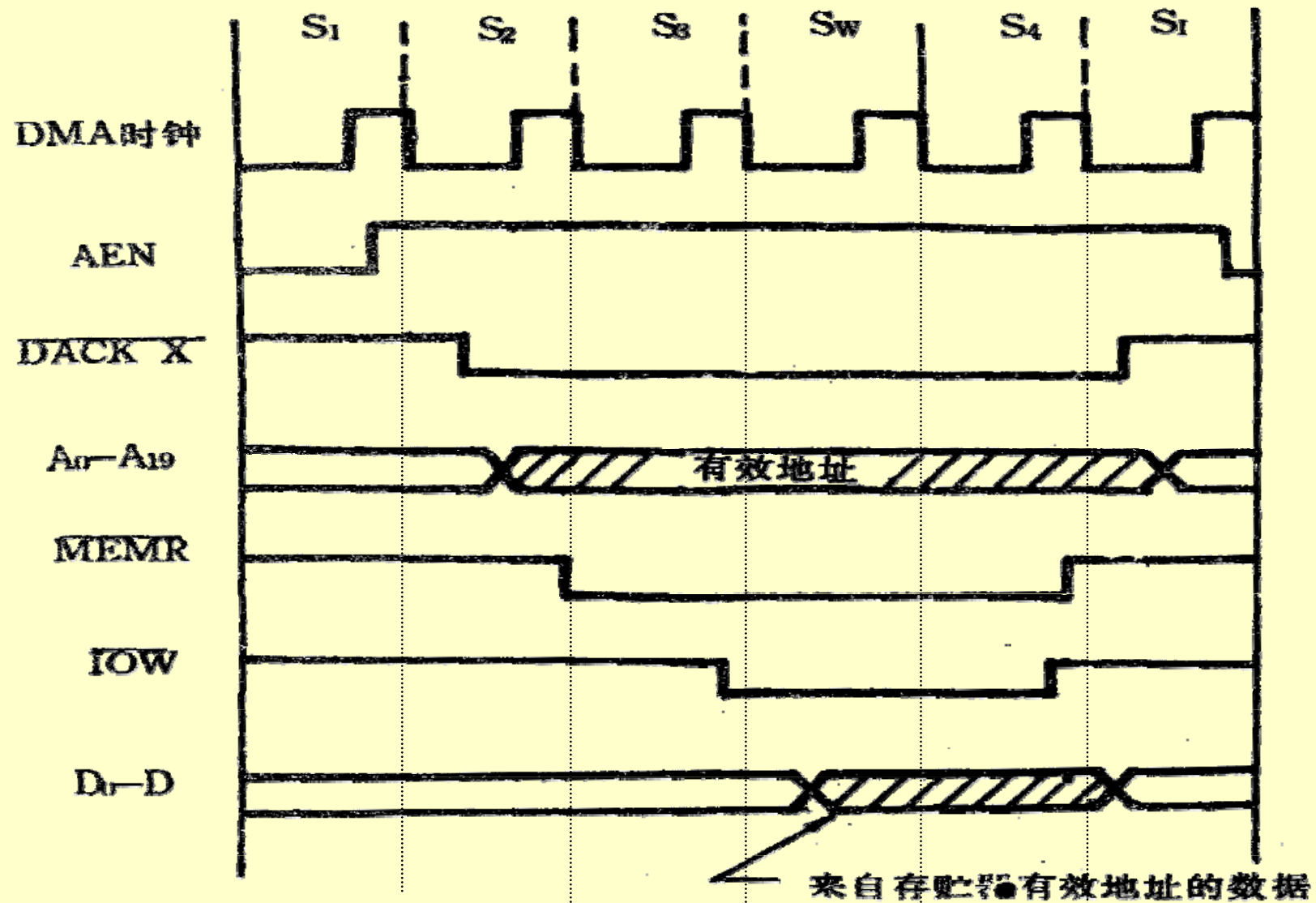
(3)  $S_1S_2S_3S_4$  —— DMA传送周期。

(\*) 在块传送方式下,  $S_4$  以后应传送下一个字节。在通常情况下, 地址的高 8 位不变, 只是低 8 位增量或减量, 所以输出和锁存高 8 位地址的  $S_1$  周期不需要了, 直接进入  $S_2$  周期。

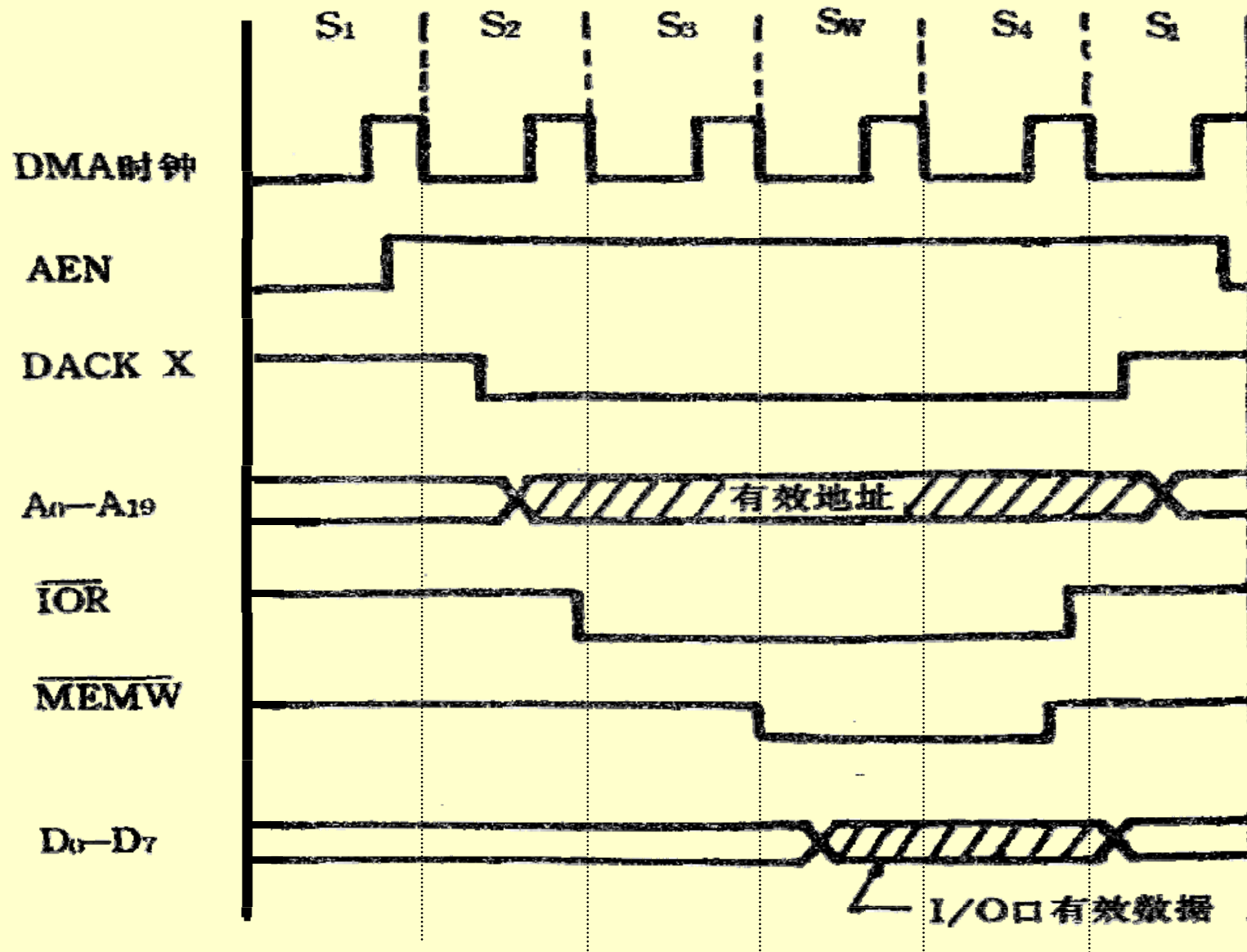


## 6. DMA操作下的总线周期

- DMA操作下 I/O 读和存储器写周期



- DMA操作下 I/O 写和存储器读周期





## 7. 8237的初始化编程

### (1) 8237 的软件命令

8237在编程状态还有两种软件命令，软件命令不需要通过数据总线写入控制字，而由8237直接控制。

- 清除高/低触发器

高/低 触发器用于控制读出或写入16位寄存器的是高字节还是低字节。

触发器为 ‘0’，则为高字节； 为 ‘1’，则为低字节。

复位后，触发器清零，每当对16位寄存器进行一次操作，则触发器改变一次状态。

- 主清除命令

该命令与 RESET 信号有同样的功能。

该命令使 命令、状态、请求、临时寄存器及内部高/低触发器清零；

使 屏蔽寄存器全部置 1；

使 8237 进入空闲周期，以便进行编程。



## (2) 8237初始化编程步骤:

1. 输出主清除命令;
2. 将传送数据的地址写入基与现行地址寄存器;
3. 将传送数据的字节数写入基与现行字节数寄存器;
4. 将传送模式写入模式寄存器;
5. 写入屏蔽寄存器;
6. 写入命令寄存器;
7. 写入请求寄存器。

## (3) 8237 各寄存器地址分配



$A_3 A_2 A_1 A_0$	端口地址	寄存器 ( $\overline{IOR}$ )	寄存器 ( $\overline{IOW}$ )
0 0 0 0	00	通道0当前地址寄存器	通道0基址与当前地址寄存器
0 0 0 1	01	通道0当前字数寄存器	通道0基字数与当前字数寄存器
0 0 1 0	02	通道1当前地址寄存器	通道1基址与]当前地址寄存器
0 0 1 1	03	通道1当前字数寄存器	通道1基字数与当前字数寄存器
0 1 0 0	04	通道2当前地址寄存器	通道2基址与当前地址寄存器
0 1 0 1	05	通道2当前字数寄存器	通道2基字数与当前字数寄存器
0 1 1 0	06	通道3当前地址寄存器	通道3基址与当前地址寄存器
0 1 1 1	07	通道3当前字数寄存器	通道3基字数与当前字数寄存器
1 0 0 0	08	状态寄存器	命令寄存器
1 0 0 1	09	——	请求寄存器
1 0 1 0	0A	——	写屏蔽寄存器单个屏蔽位
1 0 1 1	0B	——	工作模式寄存器
1 1 0 0	0C	——	清除先后触发器
1 1 0 1	0D	临时寄存器	主清除命令
1 1 1 0	0E	——	清屏蔽寄存器
1 1 1 1	0F	——	写屏蔽寄存器全部位



#### (4) 8237 编程举例

例：利用通道 0，由外设（磁盘）输入 32K字节的一个数据块，送至内存 8000H开始的区域，采用块连续传送的方式，传送完不自动初始化，外设的DREQ 和 DACK 均为高电平有效。

- 首先确定端口的地址：高位地址  $A_7-A_4$  由系统确定，用于选中芯片（CS），在此设为5；低位地址  $A_3-A_0$  用于8237内部寄存器的选址。
- 设定各模式、命令字

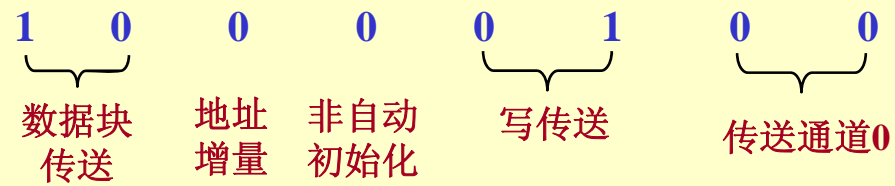
初始化程序如下：

```
OUT 5DH, AL    ; 输出主清命令
MOV AL, 00H
OUT 50H, AL    ; 输出基和现行地址的低8位
MOV AL, 80H
OUT 50H, AL    ; 输出基和现行地址的高8位
MOV AL, 00H
OUT 51H, AL    ; 输出基和现行字节数的低8位
MOV AL, 80H
OUT 50H, AL    ; 输出基和现行字节数的高8位
```

```
MOV AL, 84H
OUT 5BH, AL    ; 输出模式字
MOV AL, 00H
OUT 5AH, AL    ; 输出屏蔽字
MOV AL, 0A0H
OUT 58H, AL    ; 输出命令字
```



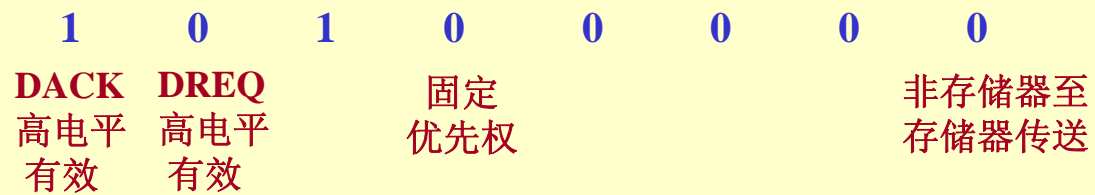
### (1) 模式控制字



### (2) 屏蔽字



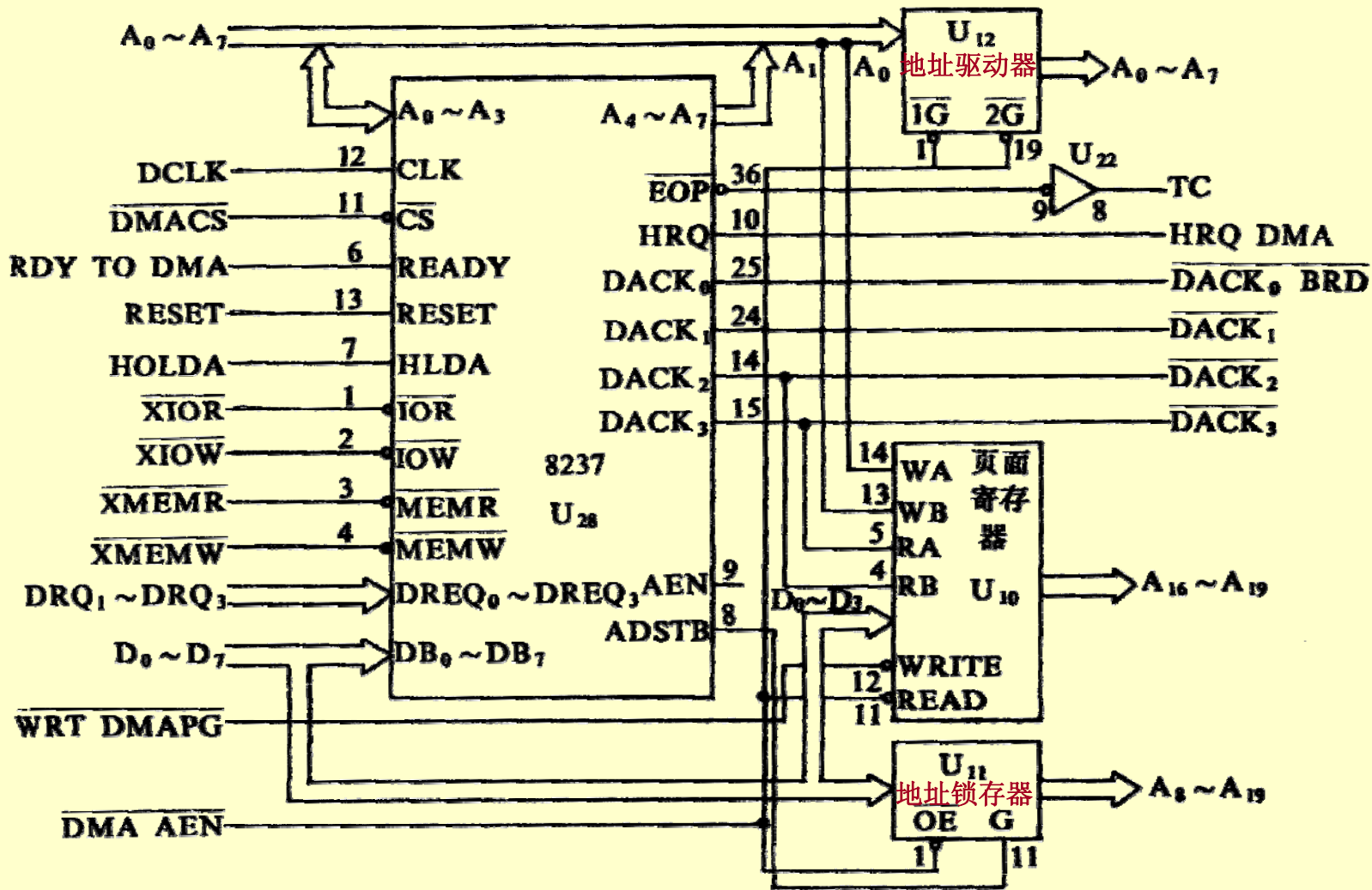
### (3) 命令字



### 三、IBM PC/XT中的 DMA 控制逻辑

IBM PC/XT中 DMA 控制逻辑由DMA控制电路和DMA应答电路组成。

#### 1. DMA 控制电路



- (1) 8237有4个DMA通道, CH<sub>0</sub> 动态RAM刷新 (在系统板上)
- |                             |   |          |
|-----------------------------|---|----------|
| CH <sub>1</sub> 为用户保留的DMA通道 | } | (在扩展选件上) |
| CH <sub>2</sub> 软磁盘         |   |          |
| CH <sub>3</sub> 硬磁盘         |   |          |

- (2) 系统地址总线20根, 寻址1M字节空间, 而DMA只能发出16位地址, 怎样寻址20位地址空间呢?

—— 使用页面寄存器, 由系统提供高4位地址。

页面寄存器采用4个4位的寄存器堆, 存放4个DMA通道的高4位地址A<sub>19</sub>~A<sub>16</sub>。

- 将地址写入页面寄存器: 在对8237进行初始化时, 由CPU写入。端口地址为:

WRITE	WB(A <sub>1</sub> )	WA(A <sub>0</sub> )	地址	DMA通道	功能
0	0	0			写入0组寄存器
0	0	1	81H	CH <sub>2</sub>	写入1组寄存器
0	1	0	82H	CH <sub>3</sub>	写入2组寄存器
0	1	1	83H	CH <sub>1</sub>	写入3组寄存器
1	*	*			寄存器内容不变

例: MOV AL, 0FH  
OUT 83H, AL



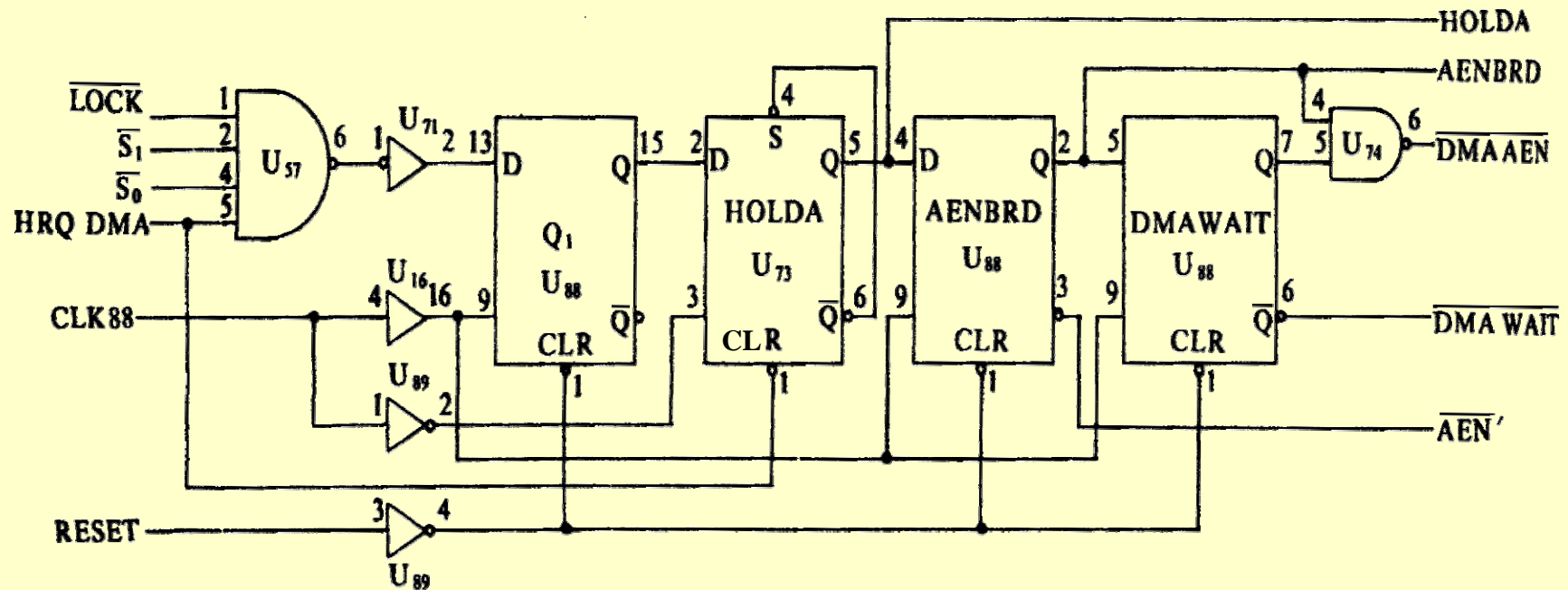
- 从页面寄存器读出地址：在DMA操作期间，8237从页面寄存器中读出数据，与它输出的16位地址一起组成20位物理地址。

工作通道	页面寄存器读出条件	RA(DACK <sub>3</sub> )	RB(DACK <sub>2</sub> )	寄存器组
CH <sub>2</sub>	$\overline{\text{DMAAEN}} \cdot \overline{\text{DACK}}_2$	0	1	1组寄存器
CH <sub>3</sub>	$\overline{\text{DMAAEN}} \cdot \overline{\text{DACK}}_3$	1	0	2组寄存器
CH <sub>1</sub>	$\overline{\text{DMAAEN}} \cdot (\overline{\text{DACK}}_3 \cdot \overline{\text{DACK}}_2)$	1	1	3组寄存器
CH <sub>0</sub>	—	—	—	—





## 2. DMA 应答电路



### (1) 总线应答条件:

HRQDMA——8237 的DMA请求。

$\overline{LOCK}$ ——锁存输出信号。在CPU工作的大部分周期中，该信号为高电平。当其为低时，表示8088不允许总线上的其它主控制器占用总线。 $\overline{LOCK}$ 信号由前缀指令LOCK使其生效，且在下一条指令完成以前保持有效。例如：**LOCK MOV AX,DX**

$\overline{S_1}$ 、 $\overline{S_0}$ ——CPU的状态信号，为11时表示8088或者处于暂停状态，或者进入总线周期结束状态，可以交出系统总线，进行DMA传送。



## (2) CPU交出总线控制权

**HOLDA**——由**HOLDA**触发器产生对**DMA**请求的应答信号，允许**8237**进行**DMA**操作；

**HRQDMA**——接至**HOLDA**触发器的复位端，在整个**DMA**操作期间它始终为高，故在**DMA**传送期间**HOLDA**始终为高电平。

**AENBRD**， $\overline{\text{AEN}}'$ ——经**AENBRD**触发器产生**AENBRD**（高电平）信号及 $\overline{\text{AEN}}'$ （低电平）信号；

**AENBRD**控制**CPU**地址锁存器和**8288**总线控制器的工作，其为高电平时，**CPU**地址锁存器禁止输出，**8288**禁止工作。

$\overline{\text{AEN}}'$ 信号送至**8288**的命令允许端**CEN**，禁止**8288**输出控制数据收发器工作的命令信号。

至此，**CPU**交出了系统总线的控制权。

## (3) DMA控制器接管总线

$\overline{\text{DMAWAIT}}$ ——**DMAWAIT**触发器产生，送至**8284**时钟信号发生器的**RDA1**端，使**8284**产生低电平的**READY**信号，则**CPU**一直工作在 $T_w$ 状态，直至**DMA**传送完毕。



**$\overline{\text{DMAAEN}}$**  —— 这个信号使 **8237** 接过系统总线控制权，它使**DMA**地址锁存器、地址驱动器、页面寄存器，把**20**位的地址信息送至系统地址总线；  
同时还控制双向驱动器把 **8237** 发出的读写命令送至系统的控制总线。

#### (4) CPU恢复对总线的控制

**HRQDMA** —— 完成传送后，**HRQDMA**恢复低电平；  
这个低电平信号使**HOLDA**触发器清零，**HOLDA**信号为低，结束对**DMA**的应答；  
使**AENBRD** 变为低电平、**AEN'**、 **$\overline{\text{DMAAEN}}$**  变为高电平，**8237** 放弃对总线的控制； **CPU**接管总线控制权。



## 四、应用举例

例：将软盘上的数据送到内存，入口参数：**DH**=扇区个数；**ES:BX**=写入数据的初始地址。

### DMA\_SETUP PROC NEAR

CLI	;关中断
OUT DMACS+12,AL	;清先后触发器
MOV AL,46H	;单字节传送，地址增量，禁止自动预置，读传送，选择通道2
OUT DMACS+11,AL	;设置模式控制字
MOV AX,ES	;取段基址
MOV CL,4	
ROL AX,CL	
MOV CH,AL	;最高4位地址保存在CH的低4位中
AND AL,0F0H	;最高4位清零
ADD AX,BX	;形成16位地址（段地址左移4位+偏移地址形成20位物理地址）
JNC J33	;无进位，转
INC CH	;有进位，最高位加1
J33: OUT DMACS+4,AL	;预置通道2的基地址和当前地址寄存器
MOV AL,AH	



OUT DMACS+4,AL	
MOV AL, CH	;最高4位地址预置到页面寄存器 81H 中
AND AL, 0FH	
OUT 81H, AL	
MOV AH, DH	;传送的扇区个数
SUB AL, AL	;乘256 (AL=0)
SHR AX, 1	;右移1位, 乘128 (一扇区128字节), 得传送字节数
DEC AX	;传送字节数减1
OUT DMACS+5, AL	;置通道2基字节数及当前字节数计数器,先低位后高位
MOV AL, AH	
OUT DMACS+5, AL	
STI	;开中断
MOV AL, 2	;开放通道2
OUT DMACS+10, AL	;输出清除屏蔽位
RET	
DMA_SETUP ENDP	

