

§ 5.7 直接存储器存取DMA

- ❖ 1 DMA传输
- ❖ 2 DMA操作
- ❖ 3 DMA控制器
- ❖ 4 DMA系统
- ❖ 5 实模式下DMA应用程序的开发

1. DMA传输

1.1 DMA传输的特点

DMA 传输数据的数率高

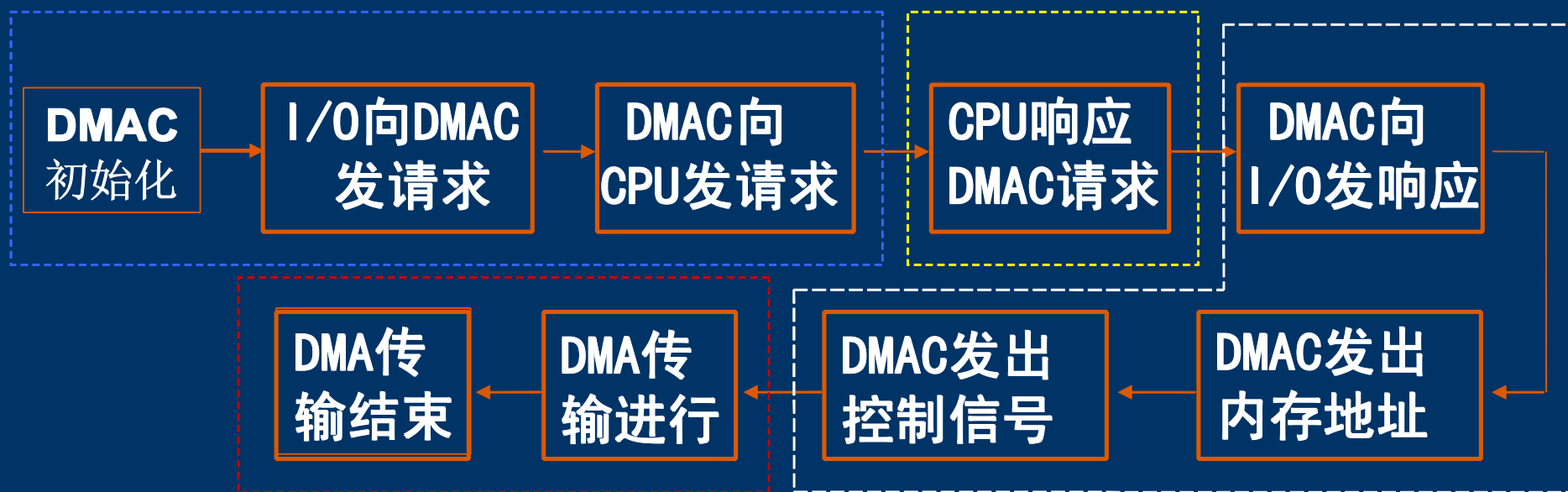
- ❖ DMA方式不用处理器干预,完成存储器与I/O间、存储器与存储器间的数据传送。
- ❖ DMA期间系统总线由DMA控制器模块控制(驱动)。
- ❖ DMA控制器模块提供系统的地址及控制信号。
- ❖ DMA控制器与处理器配合可实现系统的DMA功能。

DMA传输方式实现：

存储器与外设、外设与外设、存储器与存储器之间直接交换数据，不需经过**CPU**的累加器中转，在传送过程中，不需**CPU**控制，由**DMA**控制器负责对**DMA**传送的全过程控制。

1.2 DMA传输的过程

申请阶段 → 响应阶段 → 数据传输阶段 → 传输结束阶段



2 DMA操作

2.1. DMA 操作类型

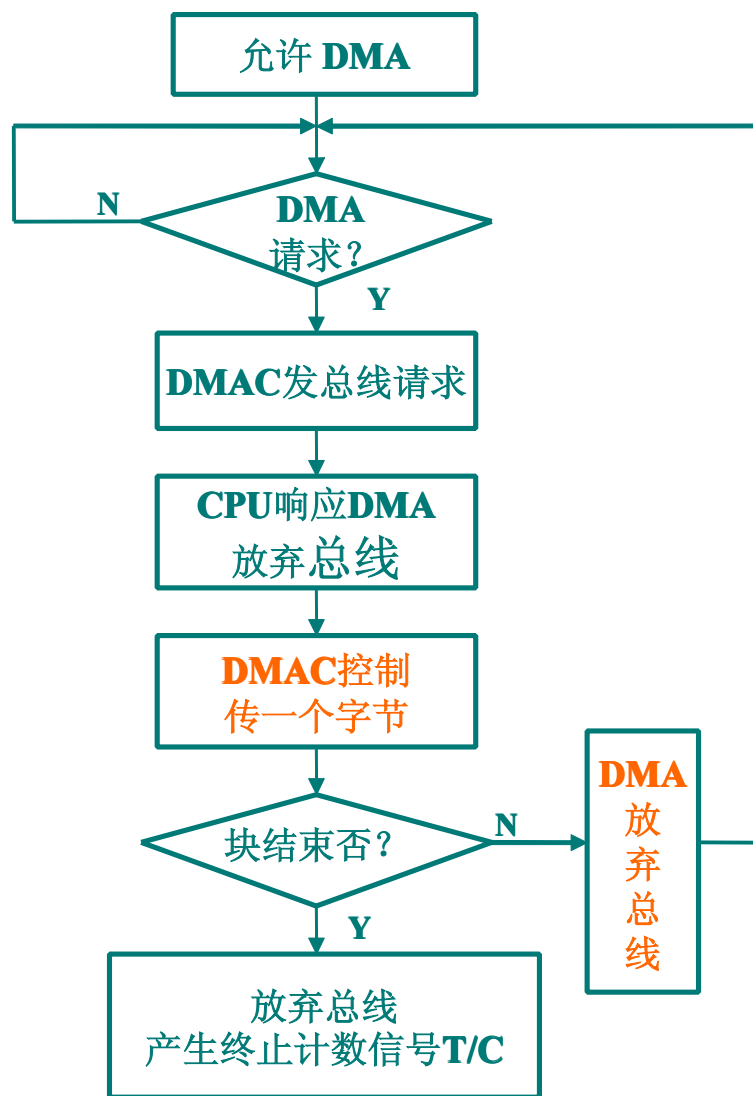
- 数据传输：
 - DMA读**：从存储器读出
 - DMA写**：向存储器写入
- 数据校验：过程同于发送，但不发读写控制信号。
- 数据检索：过程同于校验，但是在指定的内存区域内查找数据。

以存储器为
参照点

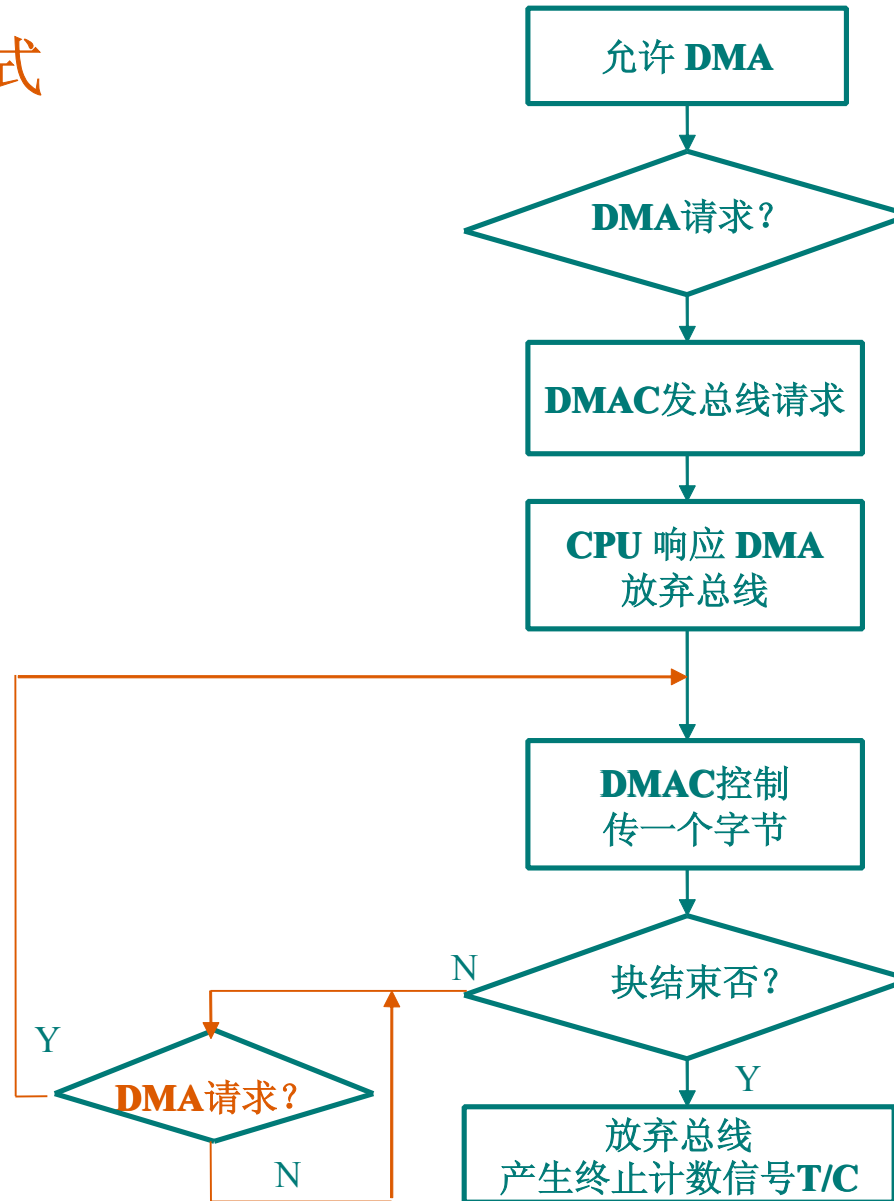
2.2 DMA操作方式

- ❖ 单字节方式： 每次**DMA**操作仅传送一个字节
- ❖ 连续（块字节）方式： 每次**DMA**操作针对一个数据块
- ❖ 请求（询问）方式： 外部有**DMA**请求时， **DMA**才占用总线，实现块操作。

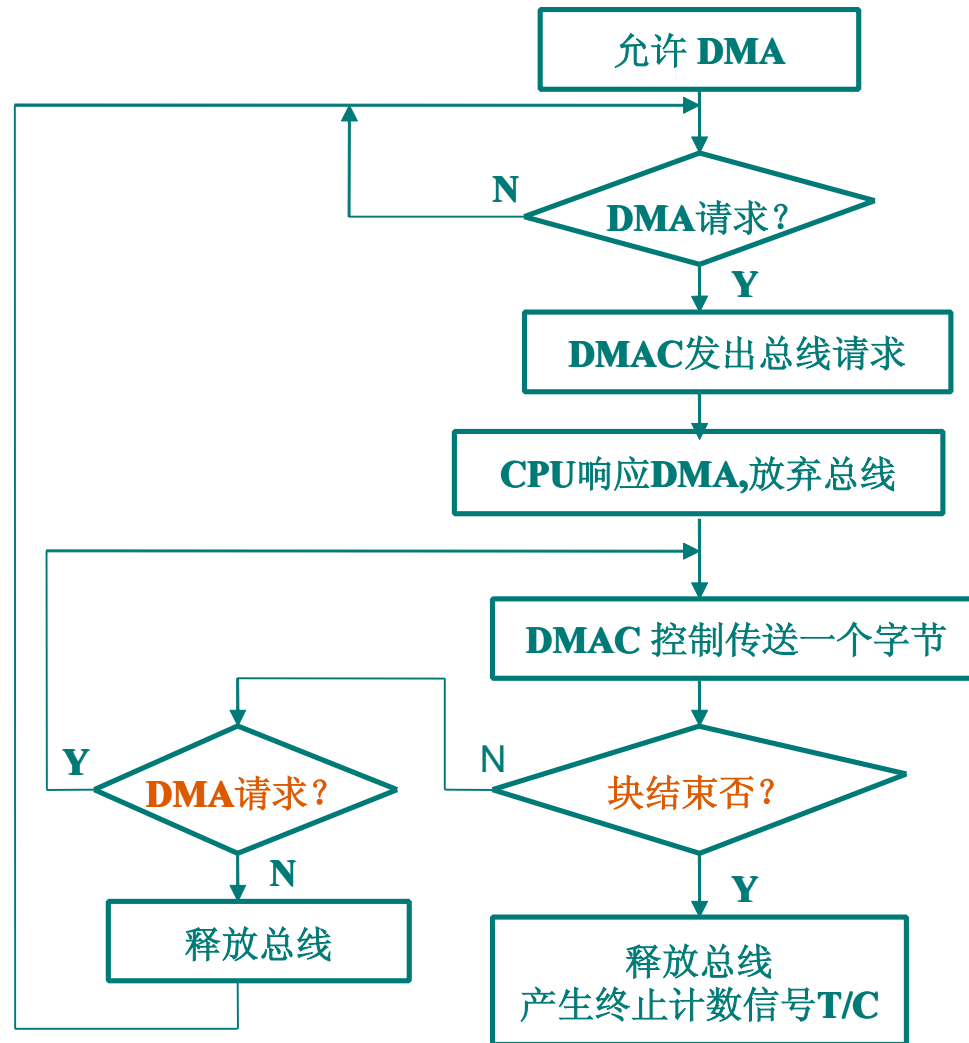
单字节方式



连续方式



请求方式



3 DMA控制器

3.1 DMA控制器在系统中的工作状态

1. DMA控制器的两种工作状态

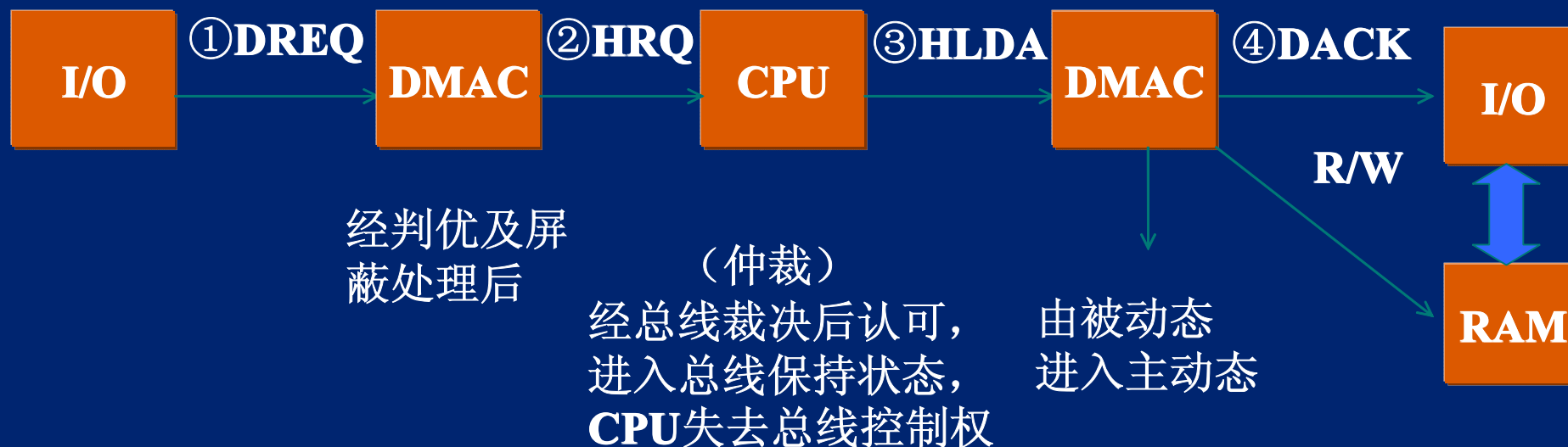
1) 被动态（DMA空闲周期）——

CPU控制总线，DMAC接受CPU对它的控制和指挥

2) 主动态（DMA有效周期）——

DMAC取得总线控制权，向存储器和外设发号施令

3.2 DMA控制器与CPU之间的总线控制权转移



3.3--3.4 DMA控制器 82C37A的外部特性 寄存器及编程命令

1. 特点:

- (1) 四个独立通道（可对4个I/O设备进行DMA服务）
- (2) 64KB寻址与计数能力
- (3) 数据传输率1.5Mb/s
- (4) 有级联方式和多种操作模式

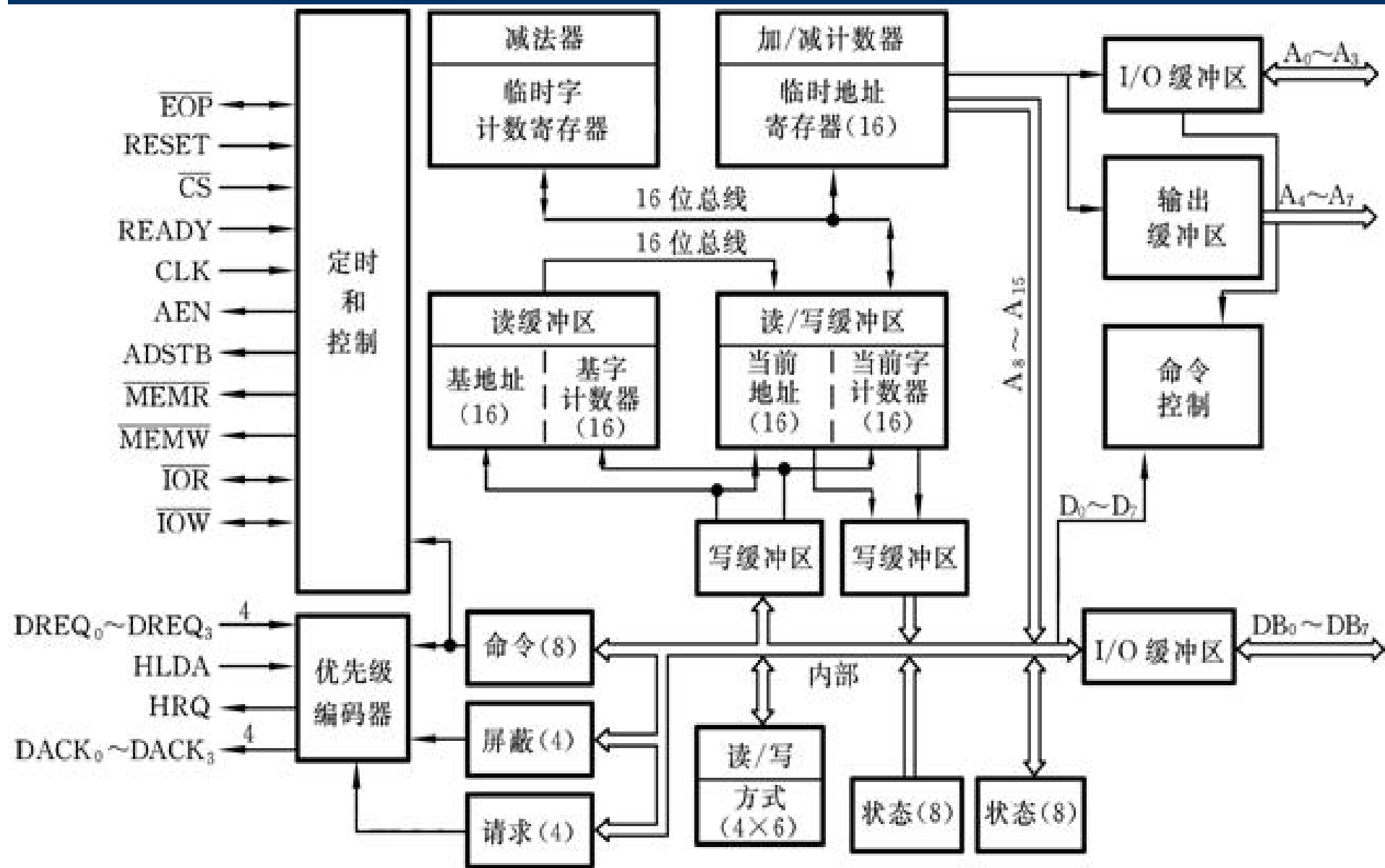
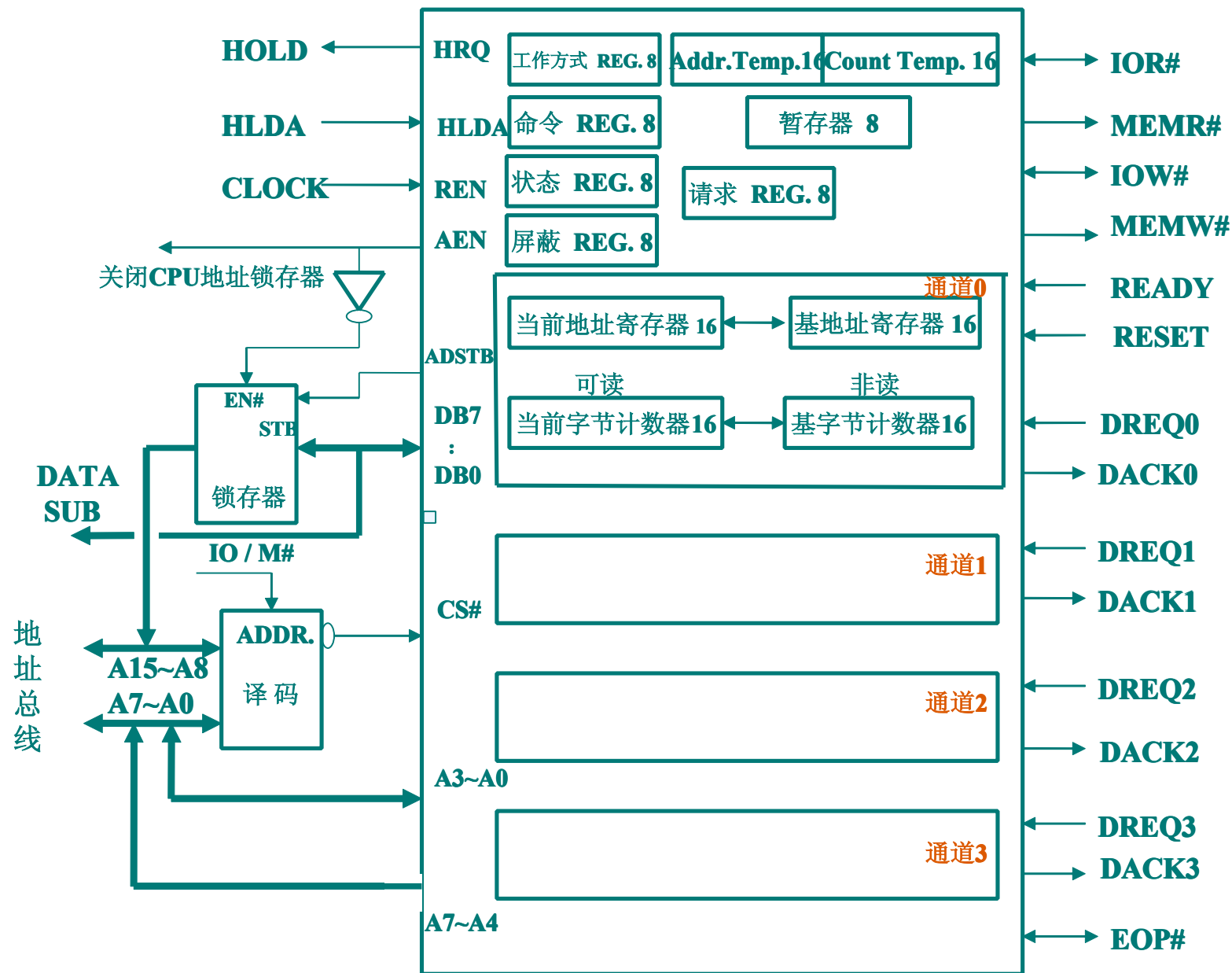


图2 82C37A内部逻辑框图



Intel 82C37A可编程DMA控制器框图

1	$\overline{\text{IOR}}$	A_7	40
2	$\overline{\text{IOW}}$	A_6	39
3	$\overline{\text{MEMR}}$	A_5	38
4	$\overline{\text{MEMW}}$	A_4	37
5	NC	$\overline{\text{EOP}}$	36
6	READY	A_3	35
7	HLDA	A_2	34
8	ADSTB	A_1	33
9	AEN	A_0	32
10	HRQ 82C37A	V_{CC}	31
11	$\overline{\text{CS}}$	DB_0	30
12	CLK	DB_1	29
13	RESET	DB_2	28
14	DACK_2	DB_3	27
15	DACK_3	DB_4	26
16	DREQ_3	DACK_0	25
17	DREQ_2	DACK_1	24
18	DREQ_1	DB_5	23
19	DREQ_0	DB_6	22
20	GND	DB_7	21

❖ 82C37A组成说明:

- 82C37A有四个独立通道。
- 每个通道都有各自的4个寄存器: 16位基地址REG, 16位当前地址REG; 16位基字节计数器, 16位当前字节计数器。
- 四个通道公用控制寄存器, 状态寄存器, 屏蔽寄存器, 请求标志寄存器及暂存器, 各寄存器均为8位。

❖ 8237A引线说明:

➤ DB7-DB0: 双向数据总线。

8237A为从模块时为数据线:

被处理器编程或读状态, 传输数据或命令字。

8237A为主模块时为地址和数据分时复用线:

地址线: 访问存储器的高8位地址A15-A8,

数据线: 在存储器到存储器传送操作时, 经
DB7-DB0, 将存储器数据送8237A暂存器。

➤ READY: 输入, 准备就绪, 主模块时控制总线周期的长度, 与慢速设备同步。

- **A3-A0**: 地址线，双向。

8237A为从模块时为**输入**，处理器寻址8237A；

8237A为主模块时**输出**低4位地址。

- **A7-A4**: 地址线，主模块时**输出**A7-A4。

- **\overline{CS}** : 片选，从模块时处理器用来寻址8237A。

- **\overline{IOR} , \overline{IOW}** : I/O读写控制，双向。

8237A在从模块时为**输入**，

在主模块时为**输出**。

- HRQ: 8237A向处理器发出的总线请求信号。
- HLDA: 处理器发给8237A的总线请求响应信号。
- AEN, ADSTB: 输出, 8位地址锁存允许及选通。在主模块时允许外部锁存器锁存8237A的高8位地址。
- MEMR#, MEMW#: 输出, 存储器读写控制, 主模块时送存储器。

➤ RESET: 输入, 复位信号, 复位时屏蔽寄存器置1, 其它寄存器置0。

➤ EOP #: 双向。

输出时, 表明内部通道传送结束;

输入时, 表明外部强迫DMA传送停止。

➤ DREQ0—DREQ3: I/O设备DMA请求输入信号。

➤ DACK0—DACK3: 输出DMA请求的响应。

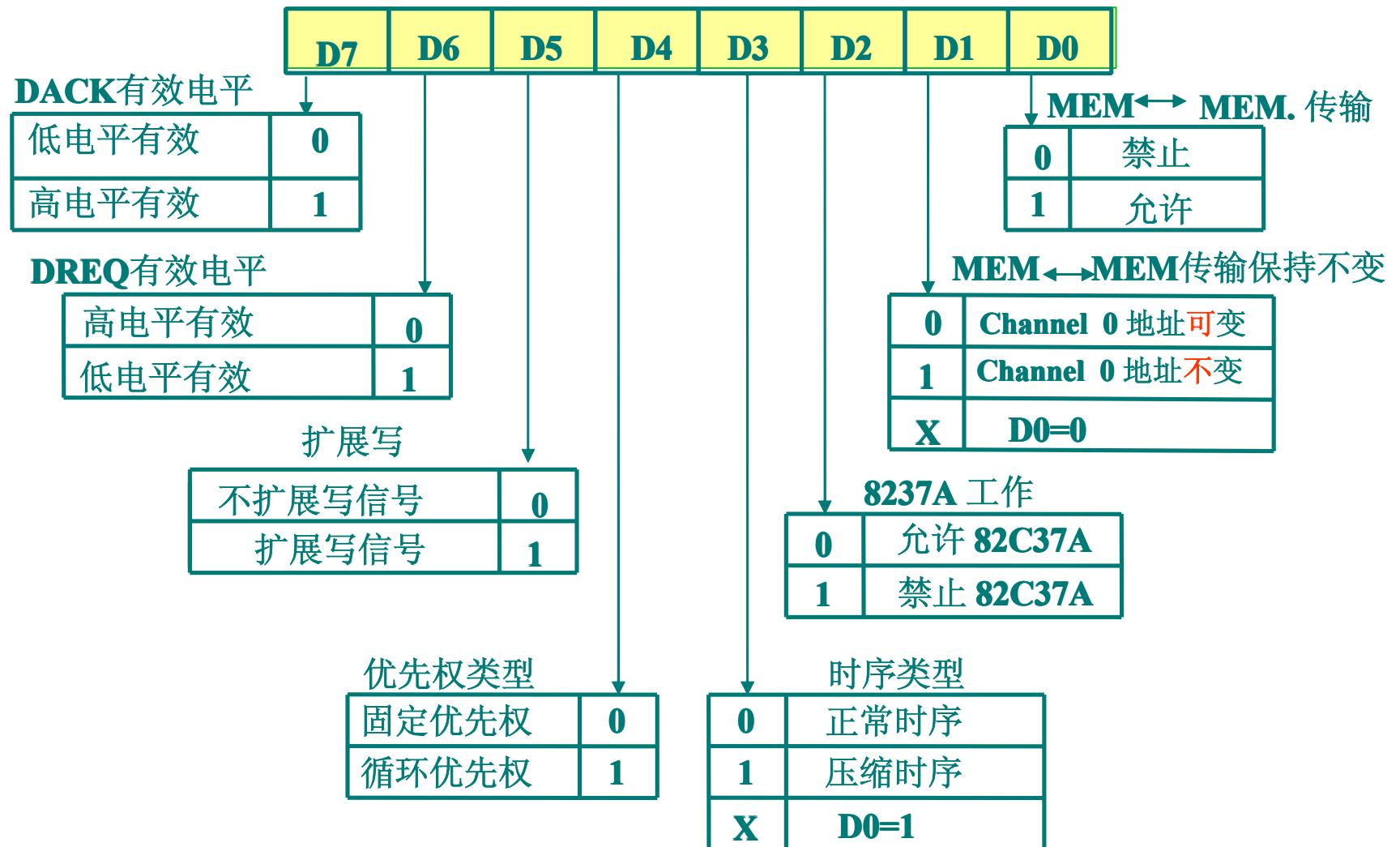
82C37A寄存器端口地址

端 口	通 道	I/O 口 地 址	寄 存 器	
			读 ($\overline{\text{IOR}}$)	写 ($\overline{\text{IOW}}$)
DMA+0	0	00	读通道 0 的当前地址寄存器	写通道 0 的基地址与当前地址寄存器
DMA+1	0	01	读通道 0 的当前字节计数寄存器	写通道 0 的基字节计数与当前字节计数寄存器
DMA+2	1	02	读通道 1 的当前地址寄存器	写通道 1 的基地址与当前地址寄存器
DMA+3	1	03	读通道 1 的当前字节计数寄存器	写通道 1 的基字节计数与当前字节计数寄存器
DMA+4	2	04	读通道 2 的当前地址寄存器	写通道 2 的基地址与当前地址寄存器
DMA+5	2	05	读通道 2 的当前字节计数寄存器	写通道 2 的基字节计数与当前字节计数寄存器
DMA+6	3	06	读通道 3 的当前地址寄存器	写通道 3 的基地址与当前地址寄存器
DMA+7	3	07	读通道 3 的当前字节计数寄存器	写通道 3 的基字节计数与当前字节计数寄存器
DMA+8		08	读状态寄存器	写命令寄存器
DMA+9		09	——	写请求寄存器
DMA+10	公 用	0A	——	写单个通道屏蔽寄存器
DMA+11		0B	——	写工作方式寄存器
DMA+12		0C	——	写清除先/后触发寄存器*
DMA+13		0D	读暂存寄存器	写总清命令*
DMA+14		0E	——	写清四个通道屏蔽寄存器命令*
DMA+15		0F	——	写置四个通道屏蔽寄存器

* 为软命令

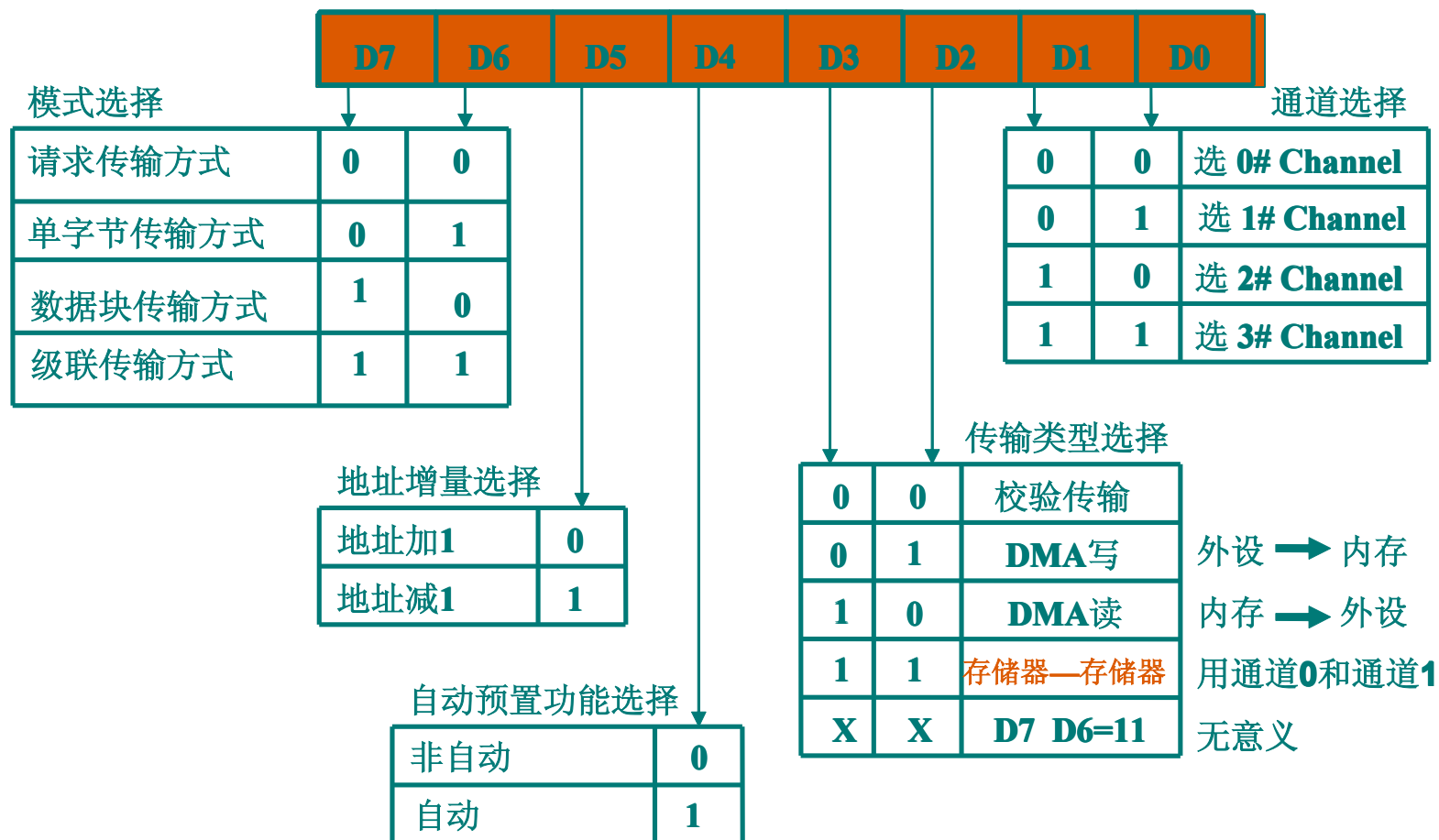
1. 命令寄存器：只写不读，用来控制8237A的操作。

格式（写8号端口）



2. 工作方式寄存器（写0B端口）：工作方式的设定：

用于设置**DMA**的操作类型、操作方式、地址改变方式、自动预置，以及选择通道。



例1. 用通道2 (D1D0=10)，单字节传送 (D7D6=01)，地址增1 (D5=0)，不用自动预置 (D4=0)，写出方式控制字。

解： 方式控制字

DMA写(存储器)	01000110	= 46H	读盘
DMA读(存储器)	01001010	= 4AH	写盘
校验	01000010	= 42H	校验盘

3. 地址寄存器:

➤ 基地址寄存器

16位，存放**DMA**传送的内存首址，传送过程中内容不变。只能写不能读

➤ 当前地址寄存器:

16位，可读可写。存放**DMA**传送过程中的内存地址，在每次传送后地址增**1**或减**1**，它的初值与基地址寄存器的内容相同。

4. 字节计数寄存器

➤ 基字节计数寄存器:

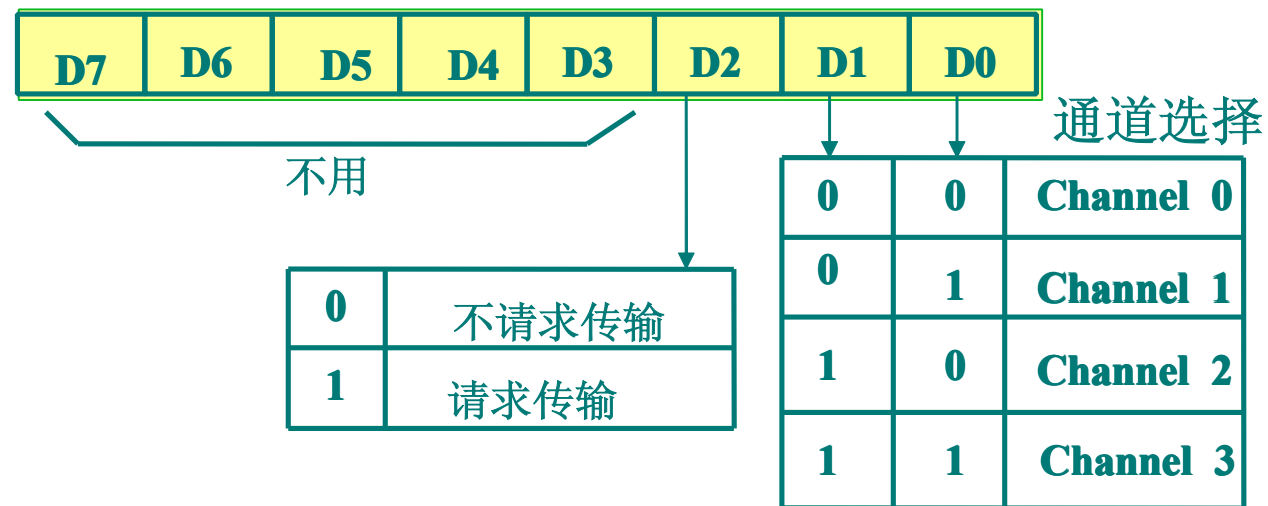
16位，只能写不能读，存放**DMA**传送的总字节数，传送过程中基字节数计数器内容不变。计数总字节数（**N-1**）。

➤ 当前字节计数器:

16位，可读可写。存放**DMA**传送过程中没有传送完的字节数。

5. 请求寄存器：只写不读。软件启动**DMA**请求，用于存储器与存储器之间传送。

格式（写**9**号端口）

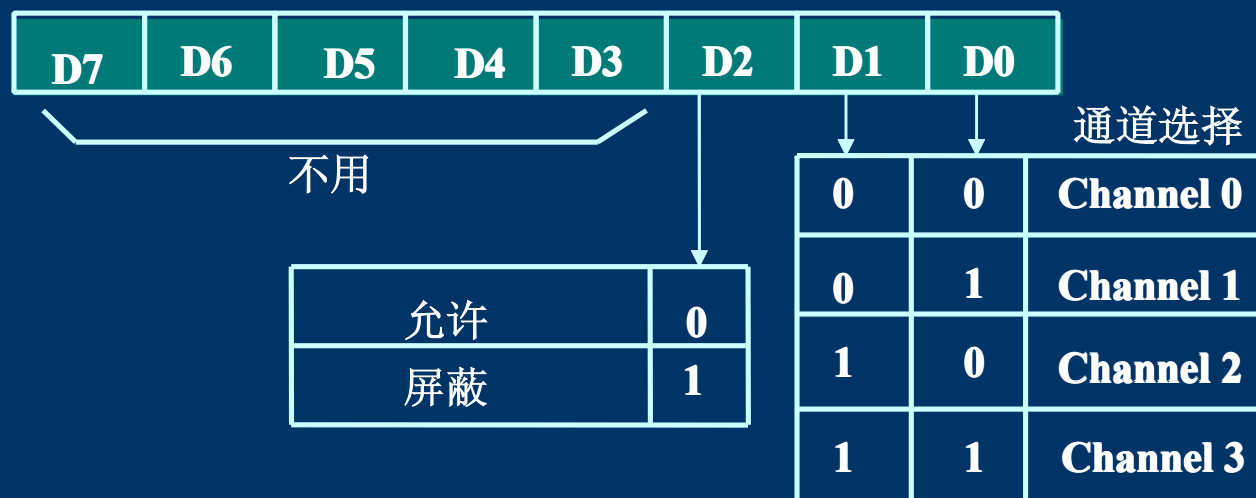


6. 屏蔽寄存器：只写不读。用来禁止或允许通道的DMA请求

有两种屏蔽命令对应两个屏蔽寄存器：

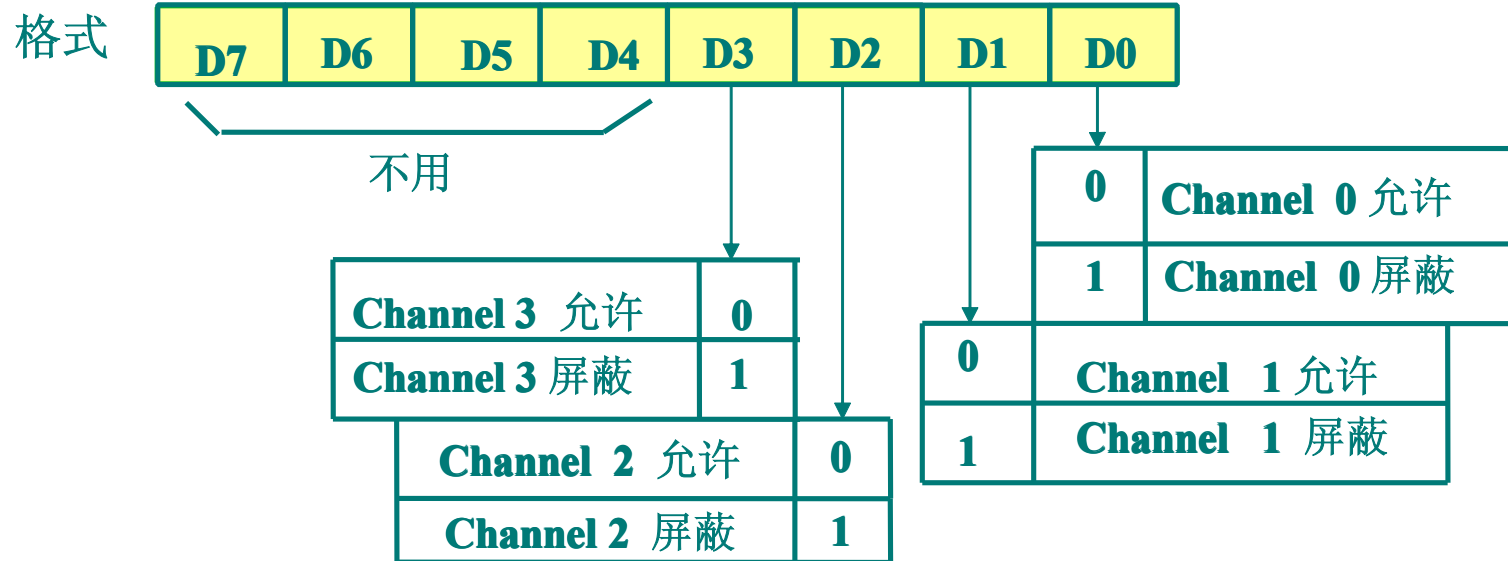
a. 单通道屏蔽寄存器（0AH 端口）

格式



```
MOV AL, 00000010 B ;开放通道2，作为响应软盘的DMA请求
OUT DMA+0AH, AL
```

b. 4个通道屏蔽寄存器（写0FH 端口）



例如：将**00000100**写入**0FH**端口，

MOV AL , 00000100B

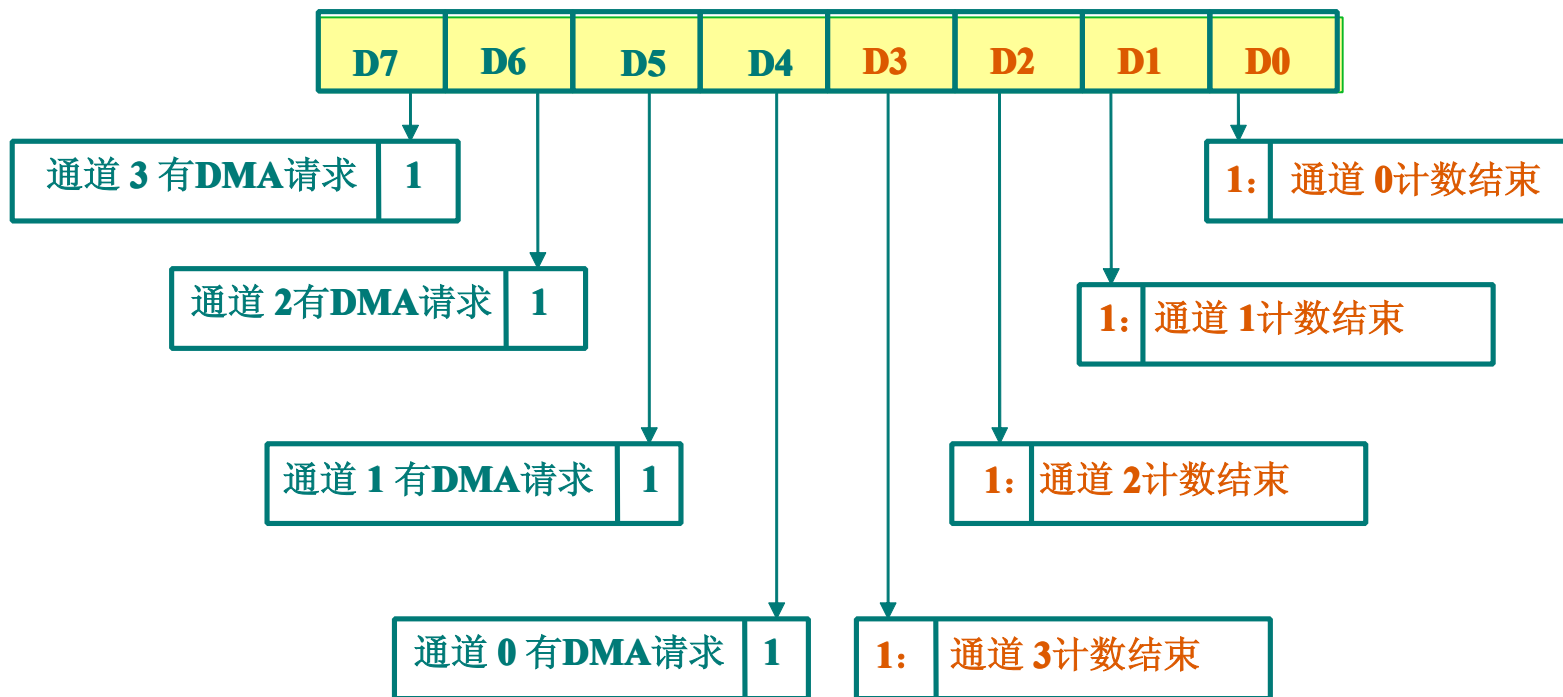
OUT DMA+0FH , AL ；将允许（开放）**0,1,3** 号通道，屏蔽**2**号通道

00000100B 若写入**0AH** 端口仅屏蔽**0** 号通道。

向**0EH** 端口写一个任意字时，将开放所有通道：**OUT DMA+0EH , AL**

7. 状态寄存器：读08H号端口，将得到状态寄存器的8位值，可监测运行状态。

格式（读8号端口）



8. 暂存寄存器

- ❖ 用于存储器对存储器传输时，暂时保存从源地址读出的数据。**RESET**信号和总清除命令可清除暂存寄存器的内容。

9. 软命令 (3个)

与端口号有关，与DB线上的内容无关。

(1) 清先/后触发器 (向0CH地址写一任意值)

用来对16位的寄存器访问次序进行控制：

清除字节指针，使将装入的顺序为：先LSB，后MSB。

0：访问低**8**位字节

1：访问高**8**位字节

MOV AL, 0AAH ; **AL**为任意值

OUT (DMA+12), AL ; 置先/后触发器为**0**态

- 对**DMA+0H—7H**端口每访问一次，该触发器就反转
- 向**DMA+0CH**端口写一个任意值，将对该触发器清“0”

(2) 总清除命令 (向0DH地址写任意值)

屏蔽寄存器为1, 其它寄存器为0。

(称软件复位)

```
MOV    AL, 0BBH          ; AL为任意值
OUT     (DMA+13), AL      ; 总清命令端口
```

(3) 清屏蔽寄存器命令 (向0EH地址写任意值) 清除4个通道的屏蔽位。

```
MOV     AL, 0CCH          ; AL为任意值
OUT      (DMA+14), AL     ; 清屏蔽寄存器命令端口
```

3.5 DMA控制器82C37的工作时序

❖ 8237A-5 有7种状态周期:

S_I S_0 S_1 S_2 S_3 S_4 和 S_W

被动态（空闲周期）



主动态（有效周期）

1. 空闲周期 S_I

上电后，未编程前，还没有DMA请求，进入空闲周期 S_I ，DMA处于被动工作方式，CPU可对DMAC编程。



2. 过渡状态 S_0

若检测到DREQ请求，DMAC即向CPU发出总线请求信号HRQ。并且，DMAC从 S_I 状态跳入 S_0 状态，并重复执行，直到收到HLDA信号（CPU）进入 S_1 状态，DMAC从被动态→主动态。

3. DMA有效周期（ $S_1 \sim S_4$ ）

- ① 在**CPU**的回答信号**HLDA**到达后，**DMAC**进入有效周期开始传送数据。
- ② 一个完整的传送周期包括**S1**，**S2**，**S3** 和**S4**四个周期，如果**I/O**速度跟不上，可在**S3** 和**S4** 之间插入等待周期**SW**。

a) S_1 周期：更新高8位地址。 $A_8 \sim A_{15}$

DMAC在 S_1 状态发出地址允许 AEN 和 $ADSTB$ 信号，将高8位地址 $A_8 \sim A_{15}$ 送到数据总线上 $DB_0 \sim DB_7$ 上。

b) S_2 周期：完成DMA的响应工作

①输出16位地址到RAM，（其它高8位已锁存）

②向I/O发请求回答信号 $DACK$ ，准备传送数据。

c) S_3 周期：读周期

发出 \overline{MEMR} （DMA读）或 \overline{IOR} （DMA写）命令，从内存或I/O准备传送数据。

d) S₄周期： 写周期

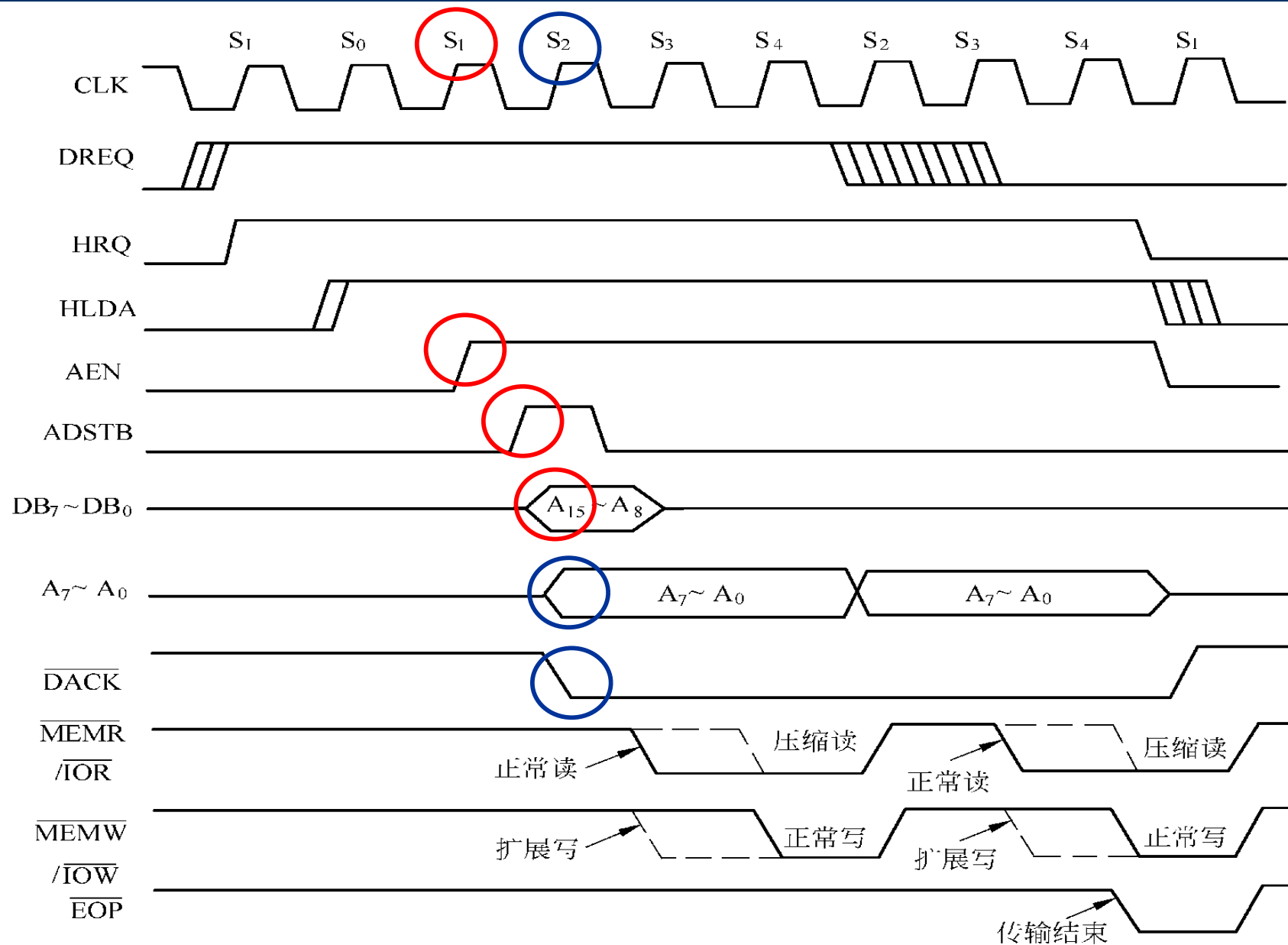
发 **IOW** (**DMA**读) 或 **MEMW** (**DMA**写) 命令，
将数据写到**RAM**或**I/O**接口。

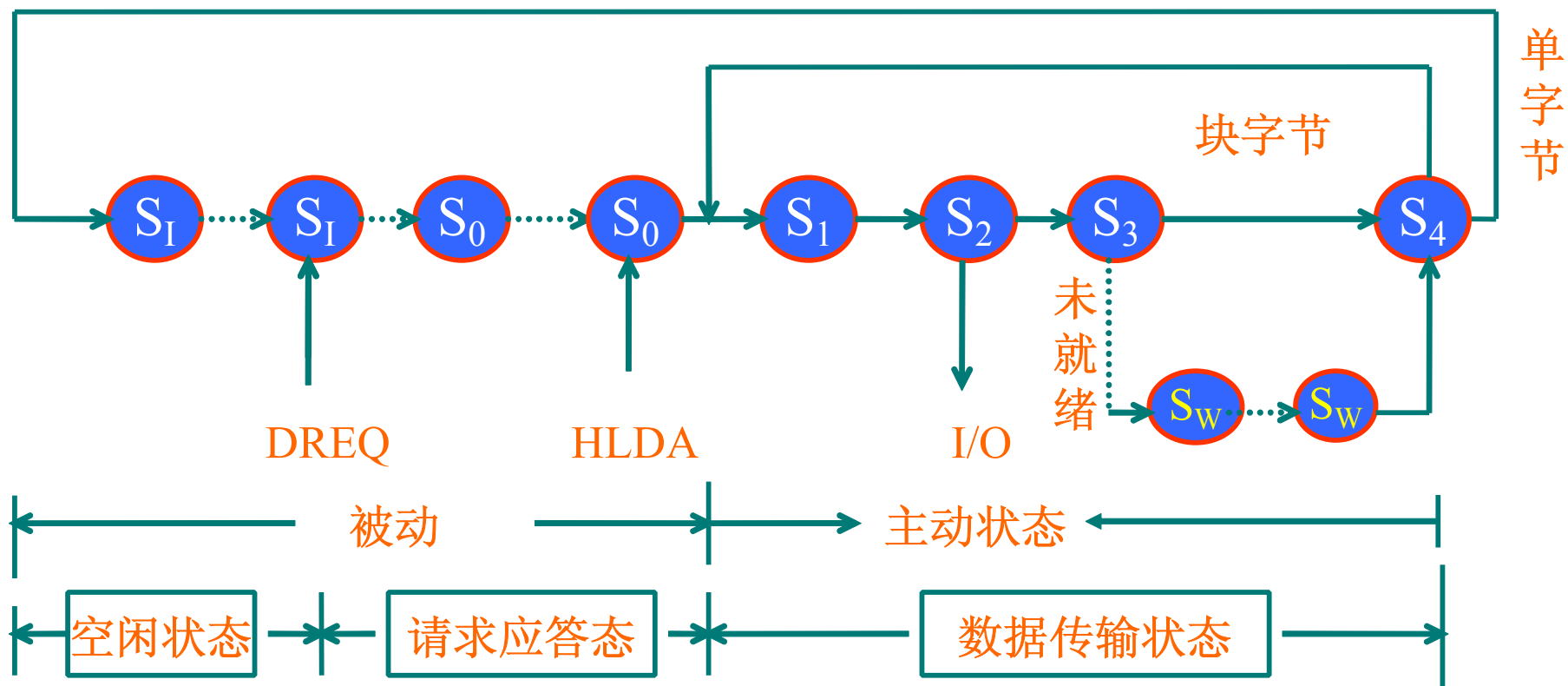
在**PC**例中，用于动态**RAM**刷新的通道**0**

0通道：有**S₁~S₄** 4个周期为**840ns**

其他： **S₁~S₃**， **S_w**， **S₄** 5个周期为 **1.02μs**

二、DMA有关的时序 (P304 图5.65)

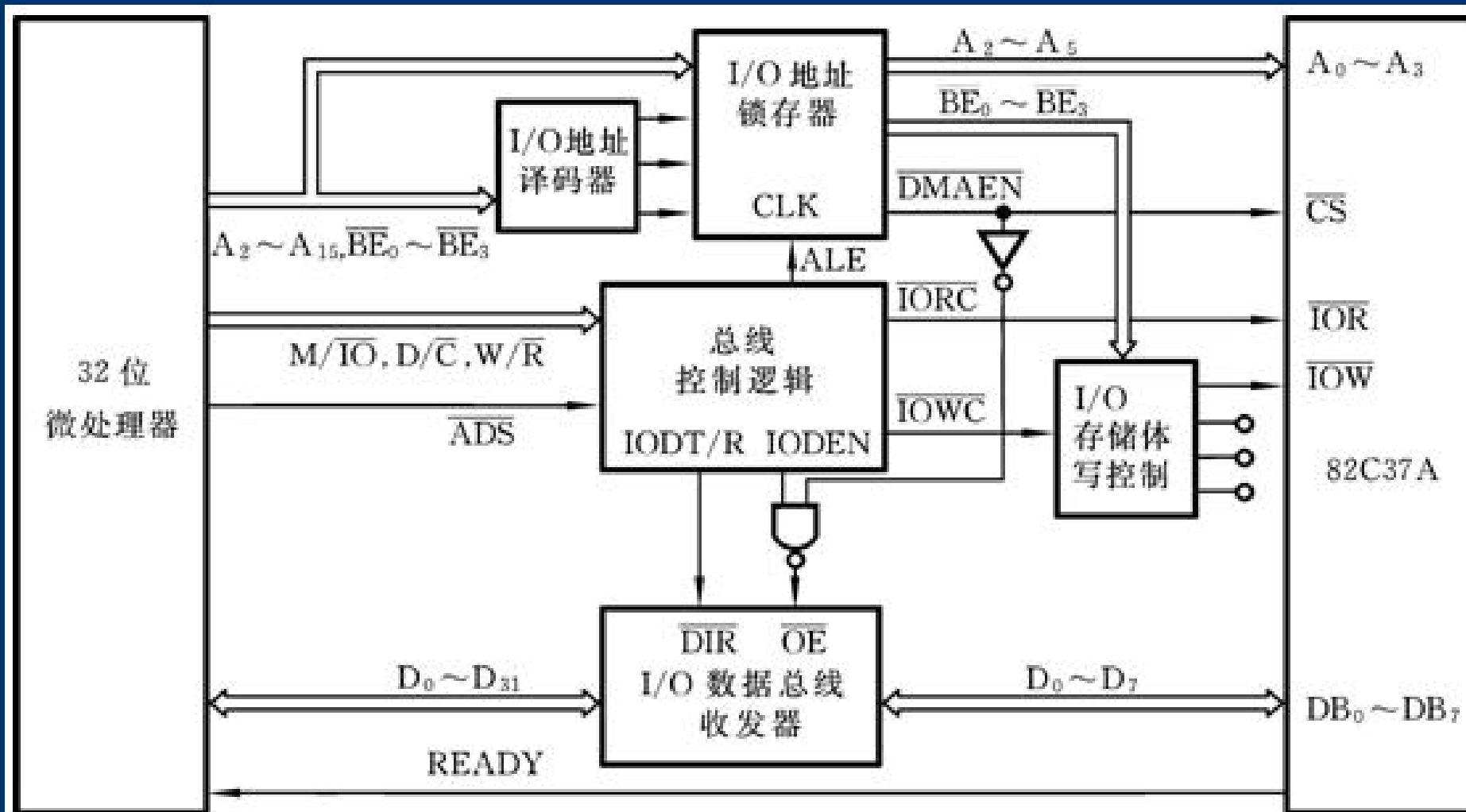




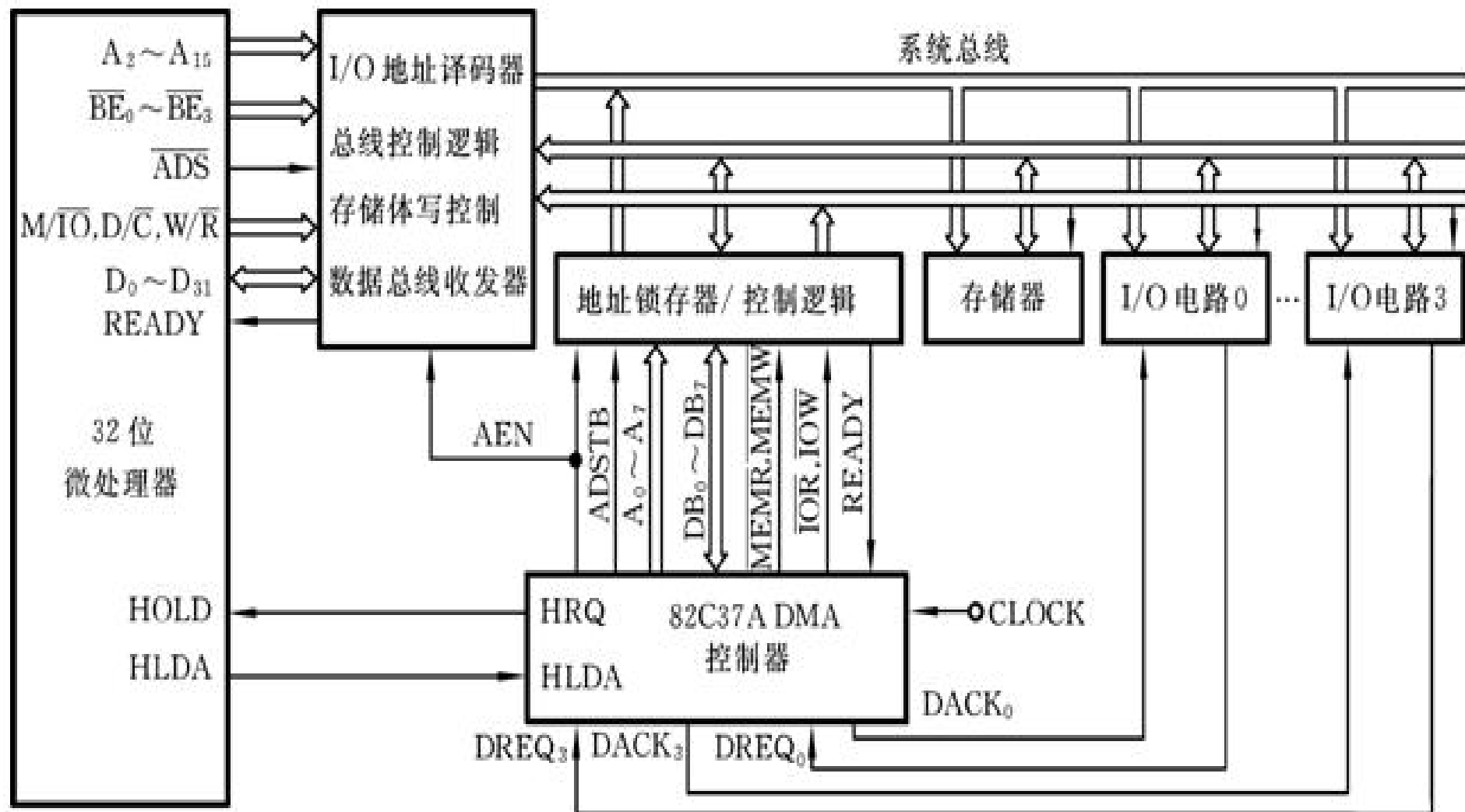
内部状态流程图

4 DMA系统

4.1 DMA控制器与微处理器的接口



4.2 DMA控制器与IO电路的接口



4.3 DMA系统中对存储器和IO设备的寻址

DMA控制器（DMAC）有效地址的生成

有效地址：是指当DMAC取得总线控制权后，作为系统主控制器，如何向存储器和I/O设备发地址信号

1. 如何扩展82C37A的寻址空间

(1) **DMAC**只能提供**16**位地址

- $A_0 \sim A_7$ （低8位）
- $DB_0 \sim DB_7$ （高8位）

(2) **20**位地址总线的系统

- ① **DMAC**提供**16**位 $A_0 \sim A_{15}$
- ② 页面地址寄存器**4**位, **$A_{16} \sim A_{19}$**

(3) **24**位地址总线的系统

- ① **DMAC**提供**16**位
- ② 页面地址寄存器**8**位(**$A_{16} \sim A_{23}$**)

4.3 DMA系统中对存储器和IO设备的寻址

- (3) 32位地址总线的系统 {
- ① DMAC提供16位 $A_0 \sim A_{15}$
 - ② 页面地址寄存器16位, $A_{16} \sim A_{31}$

2. 如何对I/O设备寻址

DMA控制器提供**DAK**信号来取代I/O设备地址选择

DAK信号、**RD**、**WR**信号同时有效, 就能完成对I/O设备端口的读/写操作, 而与I/O设备的端口地址无关

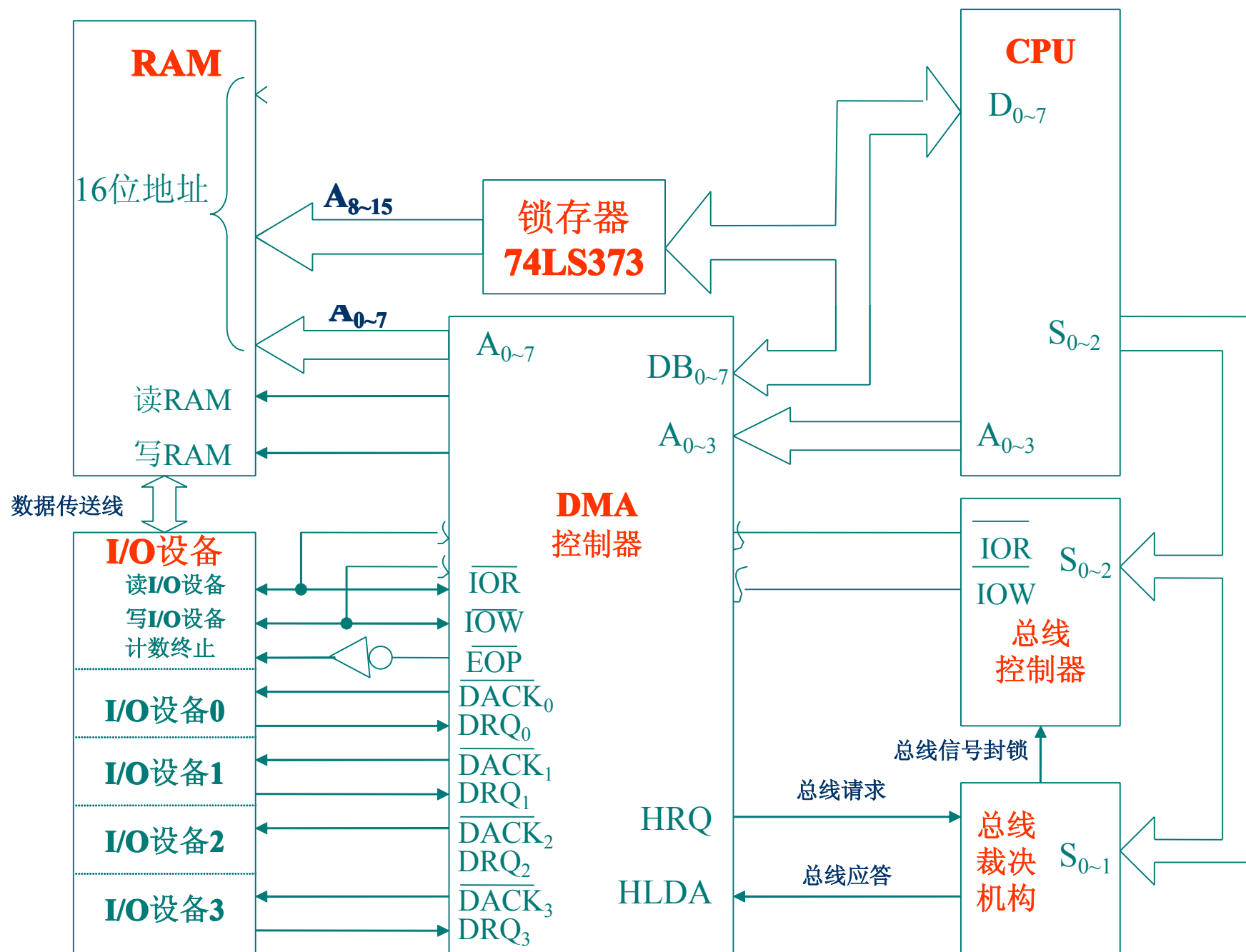


图4.5 PC系列DMA系统逻辑框图

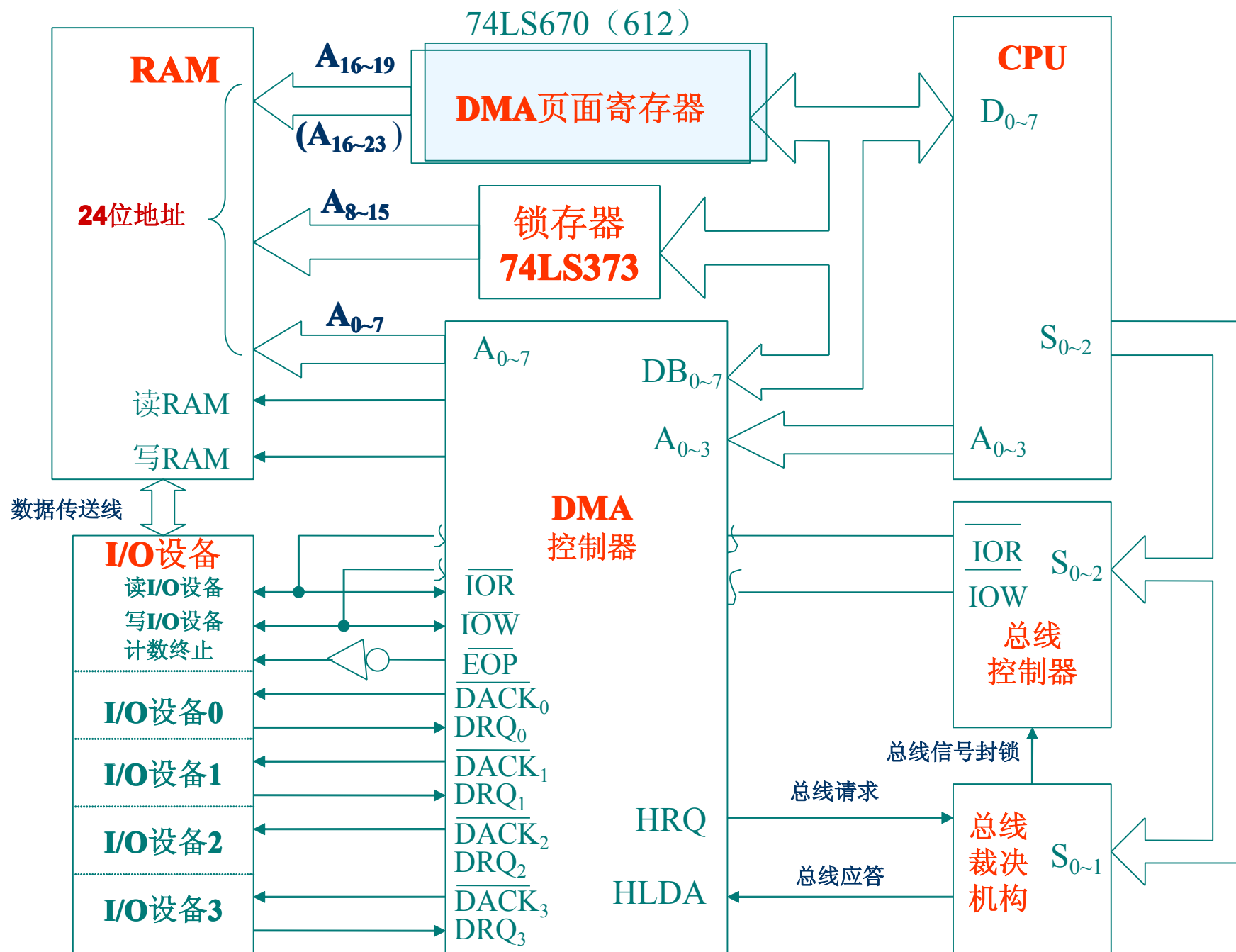
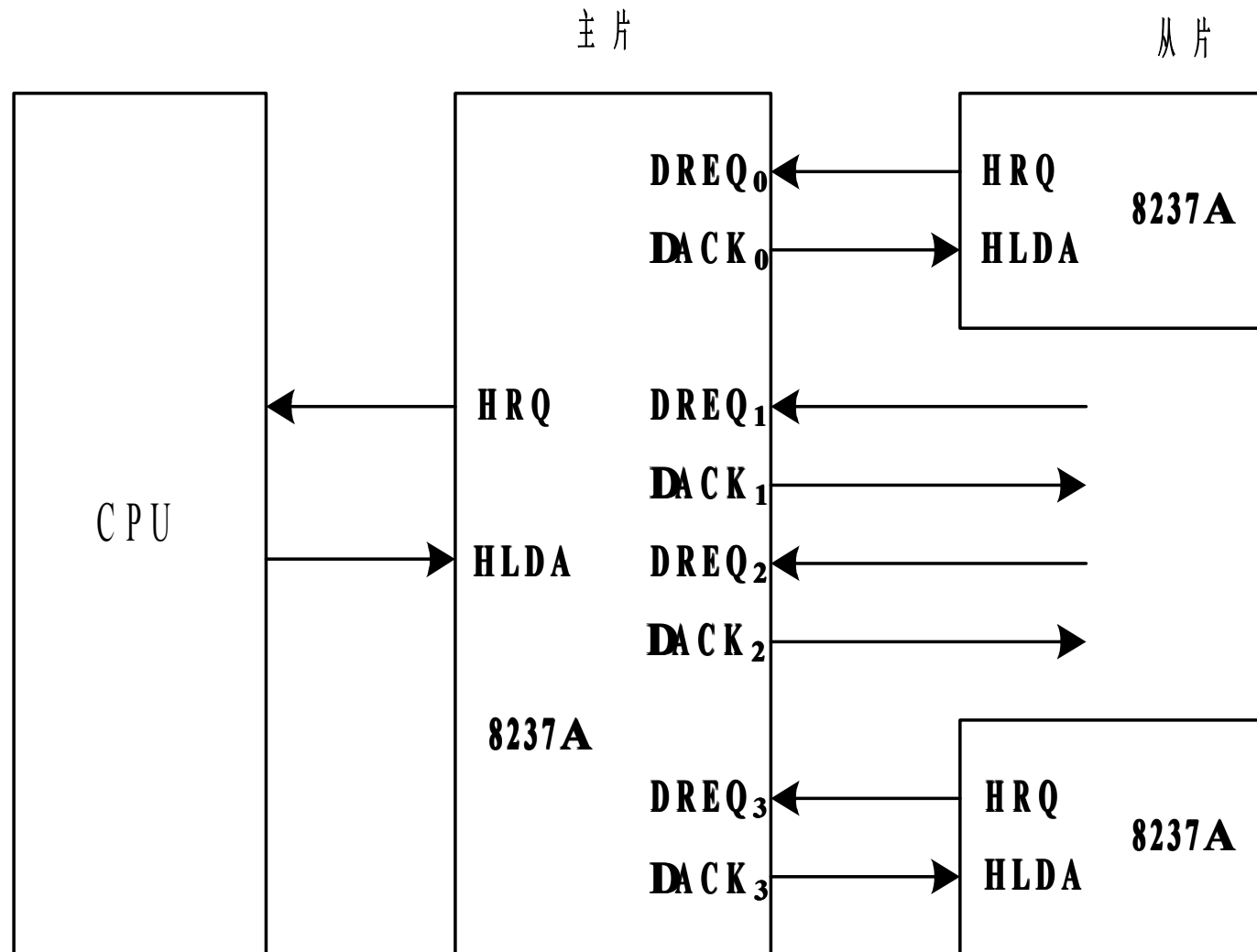


图4.5 PC系列DMA系统逻辑框图

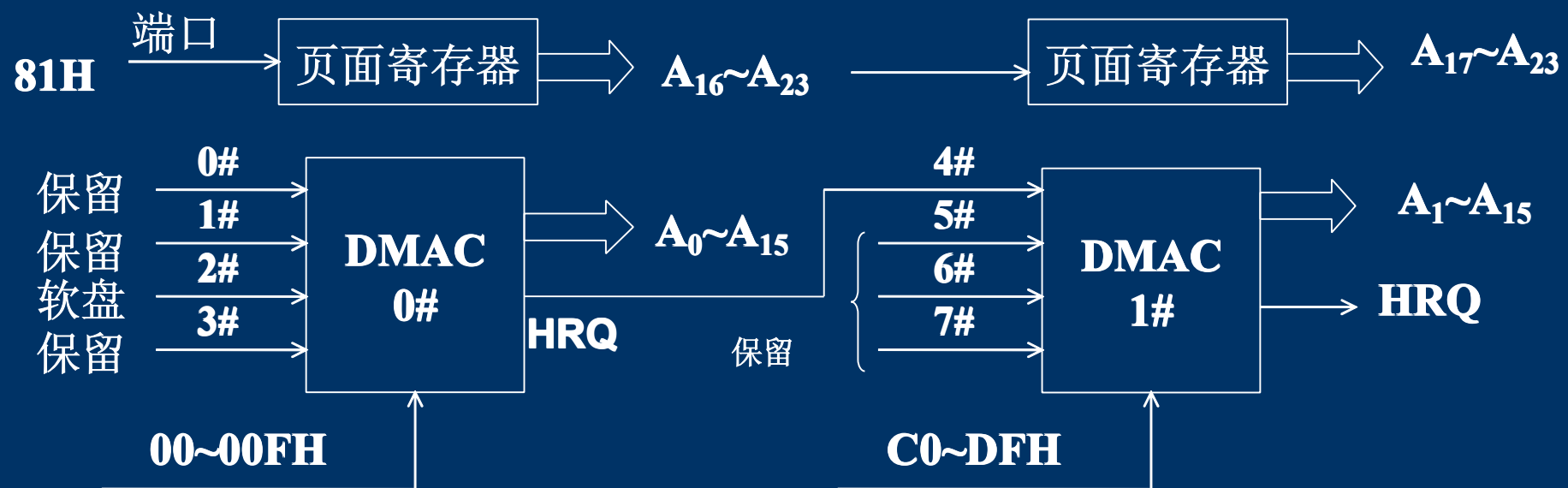
级联 方式示意图



2. 双片使用

32位微机使用2片82C37A，并在芯片组82801BA中，集成了两片82C37A.

2片8237DMAC支持7个DMA通道传送。



只能在偶地址编程，其起始端口定为00C0H，每个端口地址间隔为2

5.2 DMA编程的步骤

1. 向命令寄存器写入命令字

确定**DMA**传输的对象（存储器—存储器/存储器—I/O）、工作时序、优先级排队方式、**DREQ**和**DACK**的有效逻辑电平，以及是否处于允许工作状态。

2. 向屏蔽寄存器写入屏蔽字

为了防止正在进行初始值设置期间，由于外部另有**DMA**请求打断尚未完成的初始值设置而出错，在初始值设置之前必须先屏蔽正在初始值设置的通道，待初始值设置结束后再解除该通道的屏蔽。

3. 向方式寄存器写入方式字

确定**DMA**传输的操作类型、操作方式、地址改变方式、自动预置及通道选择。

5.2 DMA编程的步骤

4. 置先/后触发器为0态

执行一条软命令，即对**82C37A**的口地址**+0CH**，写入任何数据，可置先/后触发器为**0**态，为初始化地址寄存器和字节计数寄存器做准备。

5. 写基地址寄存器和基字节计数寄存器

把**DMA**传输的存储器首地址写入基地址寄存器和当前地址寄存器，

并把要传输的字节数减**1**，写入基字节计数寄存器和当前字节计数寄存器。

5.2 DMA编程的步骤

6. 解除屏蔽

初始值设置的最后，需向通道的单个屏蔽寄存器或**4**个屏蔽寄存器写入解除屏蔽的命令字，即可开放相应通道的**DMA**请求，以响应**DMA**请求，进入**DMA**传输周期。

7. 通过请求寄存器申请DMA传输

如果DMA传输的请求不是由外部硬件产生，而采用软件申请DMA传输，则在完成通道初始值设置后，在程序的适当位置向请求寄存器写入D2~D0=1××的命令字，即可以启动相应通道进行DMA传输。

例如，在存储器到存储器之间的DMA传输，就是采用软件申请DMA传输的。

三、系统中应用举例

通道0: DRAM刷新,

8253 OUT1定时 $15\mu s$ 产生

通道1: 给用户用,

页面寄存器地址为83H

通道2: 软驱用, 页面寄存器地址为81H

通道3: 硬盘用, 页面寄存器地址为82H

8237只有16位地址线, 产生 $A_{15} \sim A_0$, 20
位地址的高4位实现写在页面寄存器中

页面寄存器：存储DMA地址 $A_{19} \sim A_{16}$ ，用
OUT指令写，接收数据总
线上 $D_3 \sim D_0$ 的数据

例如： MOV AL , 08
 OUT 83H , AL

设置通道1页面地址为80000H。

8237 应用举例 P310--311

OUT 0CH, AL ; 清除 高/低触发器

MOV AL, 0

OUT 02, AL ; 低位地址 00H

OUT 02, AL ; 高位地址

MOV AL, 8 ; 页面地址

OUT 83H, AL ; 通道1页面寄存器

; 得内存起始地址 80000H

MOV AX, 300H ; 传输字节数

DEC AX

OUT 03, AL ; 写通道1字节数低位

MOV AL, AH

OUT 03, AL ; 写通道1字节数高位

MOV AL, 49H ; 通道1模式字:单字节读, 地址加

1

OUT 0BH, AL ; 写模式寄存器

MOV AL, 49H ; 命令字: **DACK**、**DREQ**低有效

; 正常时序, 固定优先权

No 54

OUT 08H, AL ; 写命令字

```
MOV AL, 1 ;清除通道1的屏蔽
OUT 0AH, AL ;写屏蔽寄存器
WAITF: IN AL, 08 ;读状态寄存器
AND AL, 2 ; 00000010B 通道1传输完成否
JZ WAITF ;没完成则等待
MOV AL, 5 ; 00000101B 恢复通道1的屏蔽
OUT 0AH, AL ;写屏蔽寄存器
.....
```

将内存起始地址为**80000H**的**300H**字节的内容经通道**1**直接输出给外部设备