



微型计算机原理与接口技术

第二章 80X86微处理器的结构



第一节 16位微处理器8086

一、8086的编程结构 (*)

微处理器的性能指标

字长:CPU能同时处理的数据位数（数据总线宽度）。字长16位，即为16位机

主频: CPU的时钟频率

8086: 16位机; 8088: 准16位机

8086/8088: 20根地址总线, 寻址空间1MB



编程结构：就是指从程序员和使用者的角度看到的结构。这种结构与**CPU**内部物理结构和实际布局是有区别的。

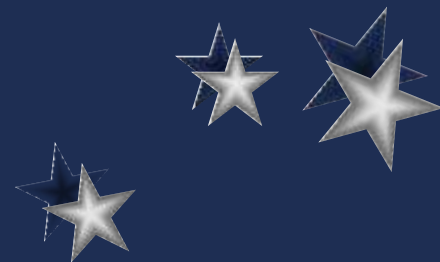
（一）**CPU**物理结构（*）

1、算逻部件**ALU**

2、寄存器组（*）

包括：通用寄存器组；指示变址寄存器；段寄存器组；程序计数器**IP**；标志寄存器

3、预取指令队列缓冲器



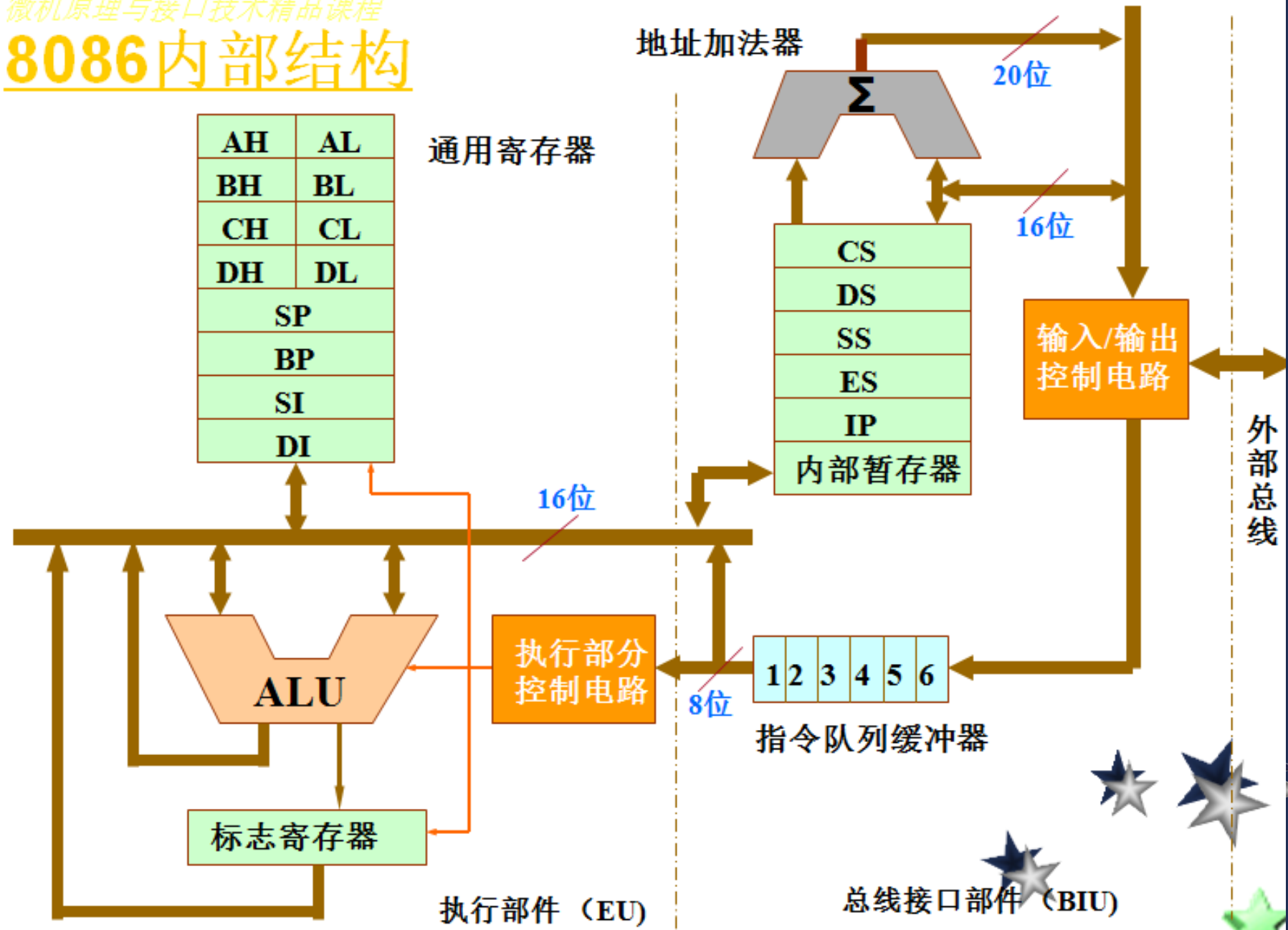
4、地址形成部件

5、控制部件 (*)

包括：指令寄存器、指令译码及总线周期
编码器、定时和控制部件、总线周期



8086内部结构



(二) 8086的编程结构

从功能上看，8086分为：总线接口部件BIU，
执行部件EU

1、总线接口部件BIU

功能：与存储器、I/O端口传送数据

取指：内存—> 指令队列

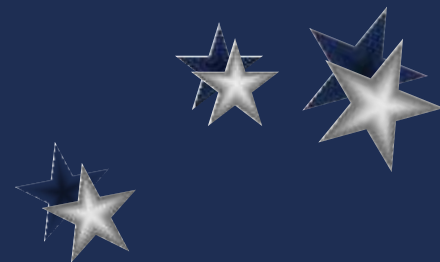
执行指令：内存或外设端口—> ^{取数} EU

EU的操作结果—> 内存

单元或外设端口



- 组成：1、4个16位段地址寄存器，即CS、DS、ES、SS
- 2、16位的指令指针寄存器 IP(Instruction Pointer);
- 3、20位的地址加法器;
- 4、6字节的指令队列缓冲器（8088为4个字节）



注：1、8086/8088在执行指令的同时，从内存中取下面1条指令或几条指令，取来的指令就放在指令队列缓冲器中。通常，8086/8088执行完一条指令就可以立即执行下一条指令，从而提高了CPU的效率。

2、地址加法器用来根据16位寄存器提供的信息计算出20位的物理地址（*）

8086/8088存储器如何分段？（*）

程序以逻辑地址编址，而不是用物理地址



2、执行部件EU：负责指令的执行（*）

（1）4个通用寄存器，即AX，BX，CX，DX；

（2）4个专用寄存器，即基数指针寄存器BP，堆栈指针寄存器SP，源变址寄存器SI，目的变址寄存器DI；

（3）标志寄存器；

（4）算术逻辑部件ALU



标志寄存器 (*) :

OF DF IF TF SF ZF AF PF CF

状态标志6个: **SF**、**ZF**、**PF**、**CF**、**AF**和**OF**

控制标志3个: **DF**、**IF**、**TF**



3、BIU和EU的动作管理 (*)

并非同步工作

- (1) 每当8086的指令队列中有2个空字节，或8088指令队列中有1个空字节时，BIU就会自动把指令取到指令队列中。
- (2) 取指和访存的冲突
- (3) 指令队列满
- (4) 执行转移、调用和返回指令时，指令队列的操作



8086/8088系统与传统计算机工作方式的区别 (*)

传统计算机的工作步骤—〉取指和执行是串行的(*)

8086/8088的工作步骤—〉取指和执行是并行的(*)



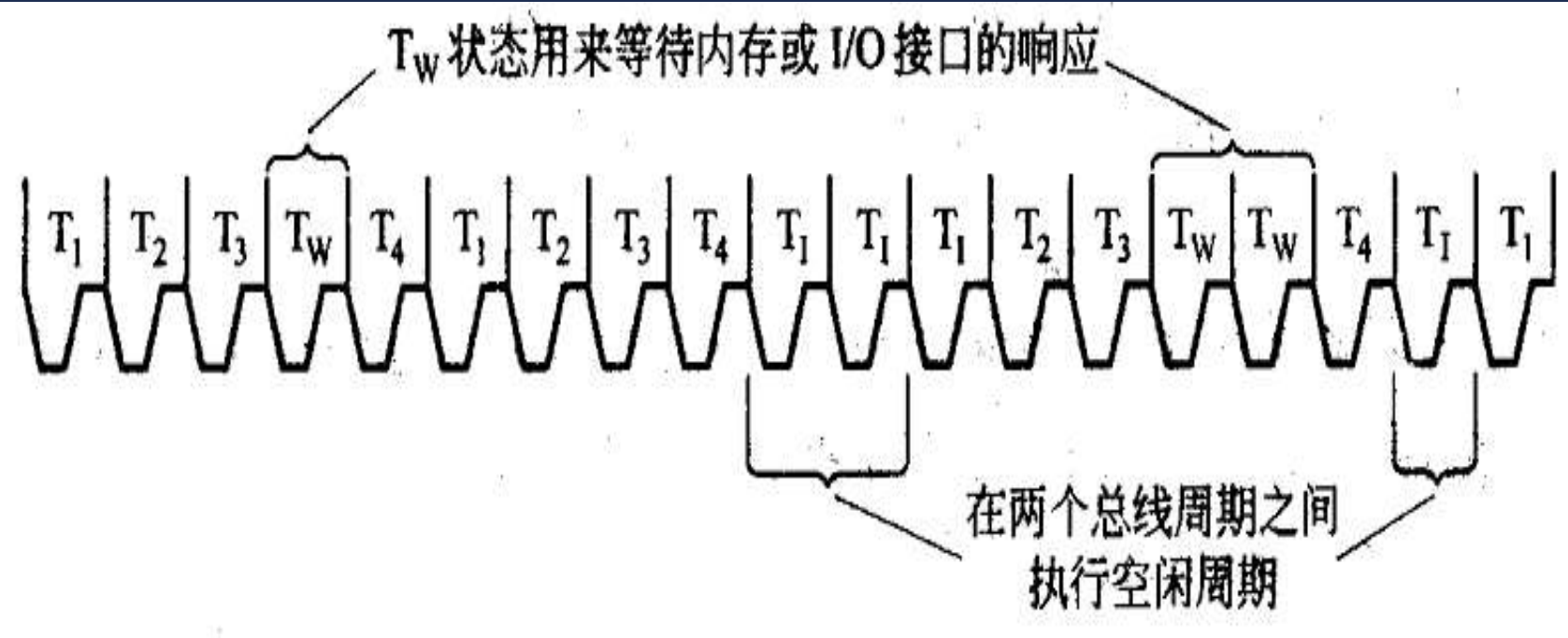
3、8086总线周期的概念 (*)

取指令或传送数据，就需要**CPU**的总线接口部件执行一个总线周期

在**8086/8088**中，一个基本的总线周期由**4**个时钟周期组成，时钟周期是**CPU**的基本时间计算单位，它由计算机主频决定。

4个时钟周期分别称为**4**个状态，即**T1**状态，**T2**状态，**T3**状态，**T4**状态





(s)



二、8086的引脚信号和工作模式

(一) 最大模式和最小模式的概念

最小模式：就是在系统中只有8086或者8088一个微处理器。在这种系统中，所有的总线控制信号都直接由8086或8088产生，因此，系统中的总线控制电路可减到最少。

最大模式：用在中等规模或者大型的8086/8088系统中。在最大模式系统中，总是包括有两个或多个微处理器，其中一个主处理器就是8086或者8088，其他的处理器称为协处理器，协助主处理器工作。



协处理器（*）：数值运算8087；

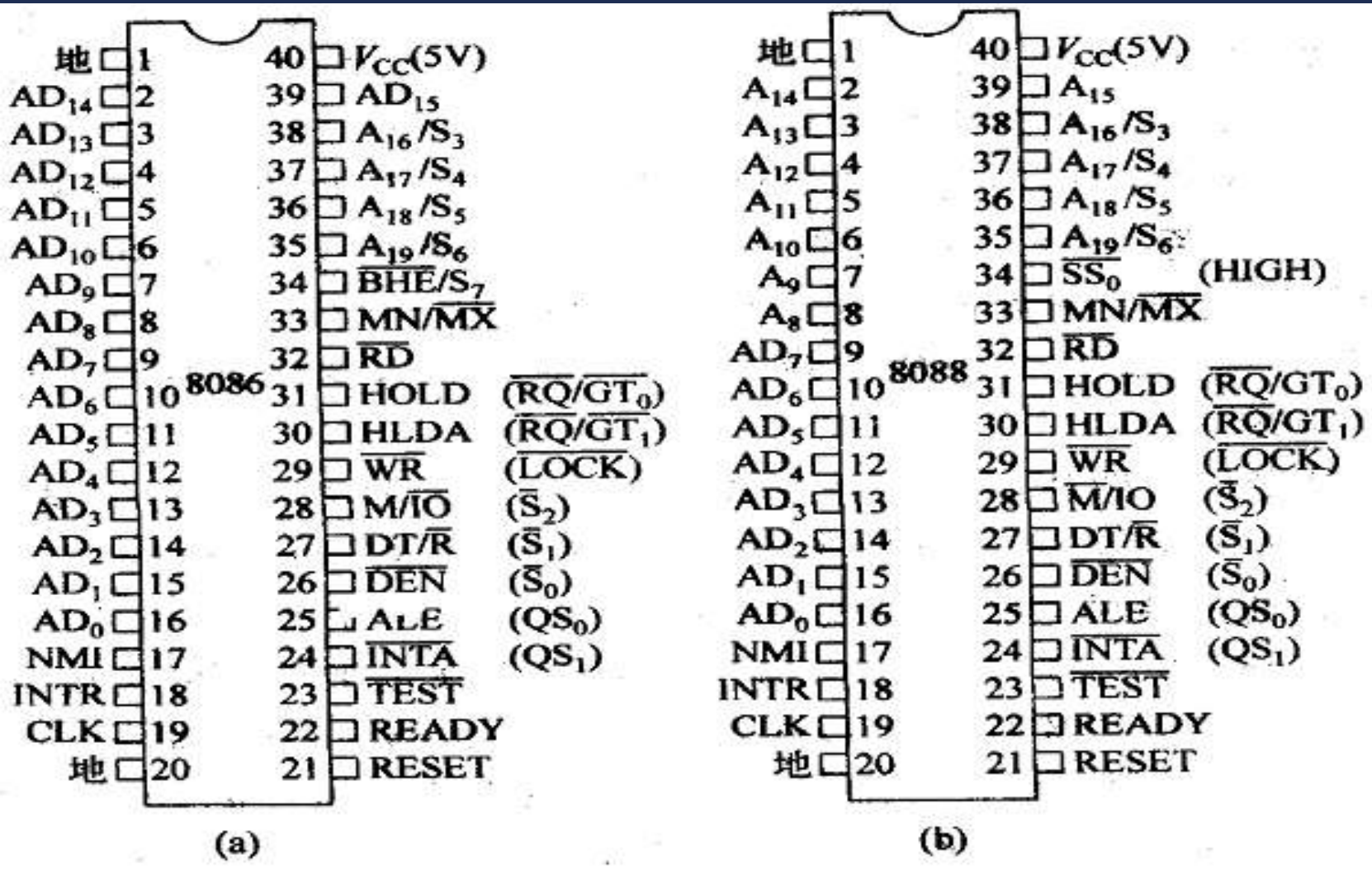
I/O操作8089

8086/8088到底工作在最大模式还是最小模式由硬件决定

（二）8086/8088的引脚信号和功能



(二) 8086/8088的引脚信号和功能



注（*）：1、8086/8088的数据线和地址线是复用的。

2、8086/8088的引脚差别（*）

（1）8086有16个地址/数据复用引脚

8088有8个地址/数据复用引脚

（2）28脚和34脚定义不同



1、各引脚信号 (*) :

(1) GND, VCC

(2) AD15~AD0: 地址/数据复用引脚,双向工作 (*)

在总线周期的T1状态用来输出要访问的存储器或I/O端口的地址

T2, T3状态,若为读周期, 则处于浮空状态, 若为写周期, 则用来传输数据。

注 (*) : 在8086系统中,常将AD0信号作为低8位数据的选通信号



(3) **A19/S6~A16/S3**地址/状态复用引脚，
输出 (*)

总线周期的T1状态用来输出**地址的最高4位**，在总线周期的T2，T3，Tw和T4状态时，用来输出**状态**信息

(4) **/BHE /S7**：高8位数据总线允许/状态复用引脚，输出 (*)

(5) **NMI**

(6) **INTR**



(7) /RD

(8) CLK

(9) **Reset**: 在复位的时候, 代码段寄存器 **CS** 和指令指针寄存器 **IP** 分别初始化为 **FFFFH** 和 **0000H**。所以, **8086/8088** 在复位之后再重新启动时, 便从内存的 **FFFF0H** 处开始执行指令。因此, 一般在 **FFFF0H** 处存放一条无条件转移指令, 转移到系统程序的入口处。这样, 系统一旦被启动, 便自动进入系统程序。



(10) Ready

(11) /TEST

(12) MN/MX

上述信号是8086/8088工作在最小模式和最大模式时都要用的。此外，8086/8088第24~31脚还有8个控制信号，它们在最小模式下有不同的名称和定义。



(三) 最小模式 (*)

1、 $\overline{\text{INTA}}$: 中断响应信号输出 (s)

2、 ALE : 地址锁存允许信号输出

在任何一个总线周期的T1状态, ALE 输出有效电平, 以表示当前在地址/数据复用总线上输出的是地址信息。 ALE 端不能浮空

3、 $\overline{\text{DEN}}$

4、 DT/R

5、 $\text{M}/\overline{\text{IO}}$

6、 $\overline{\text{WR}}$

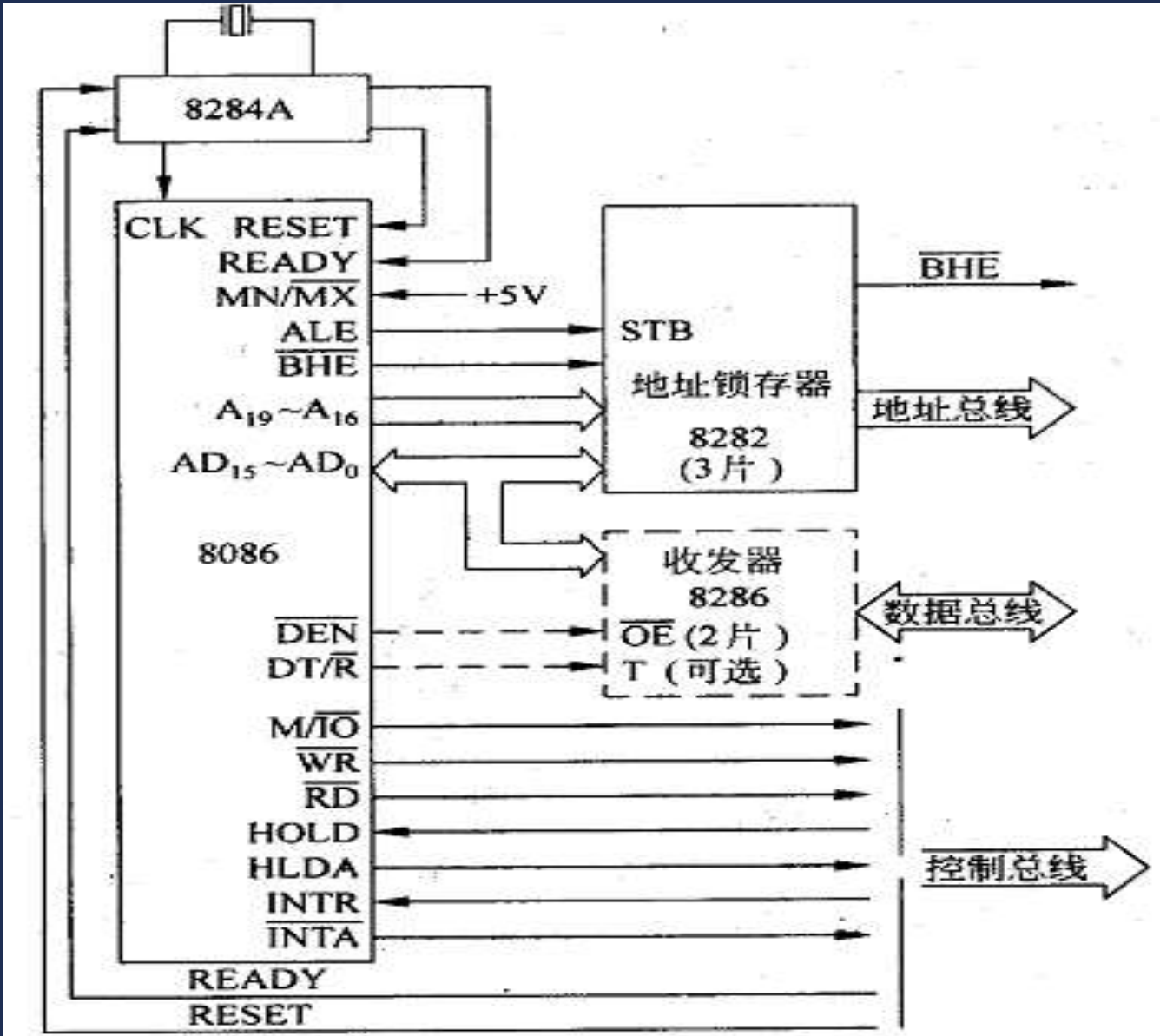


7、HOLD

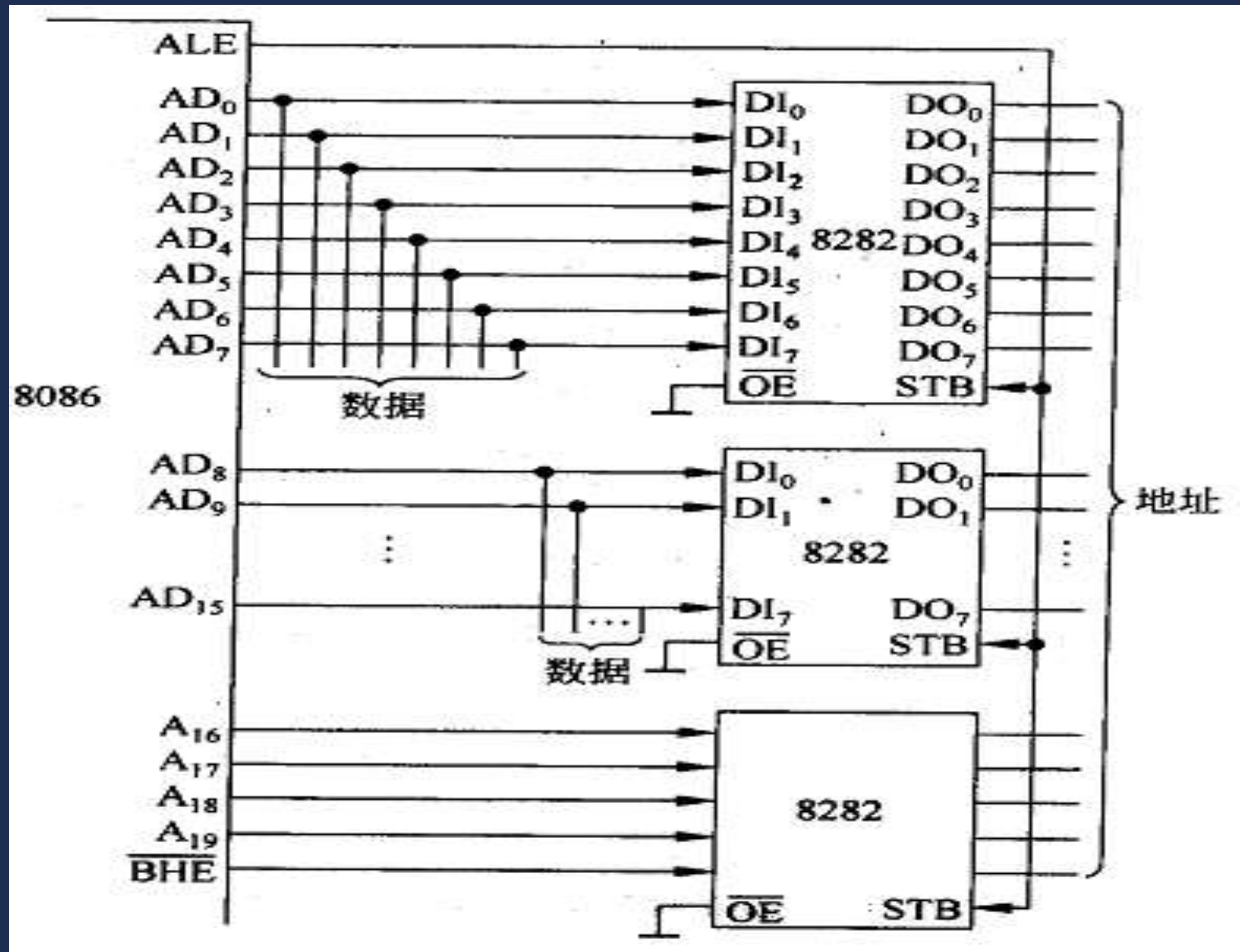
8、HLDA



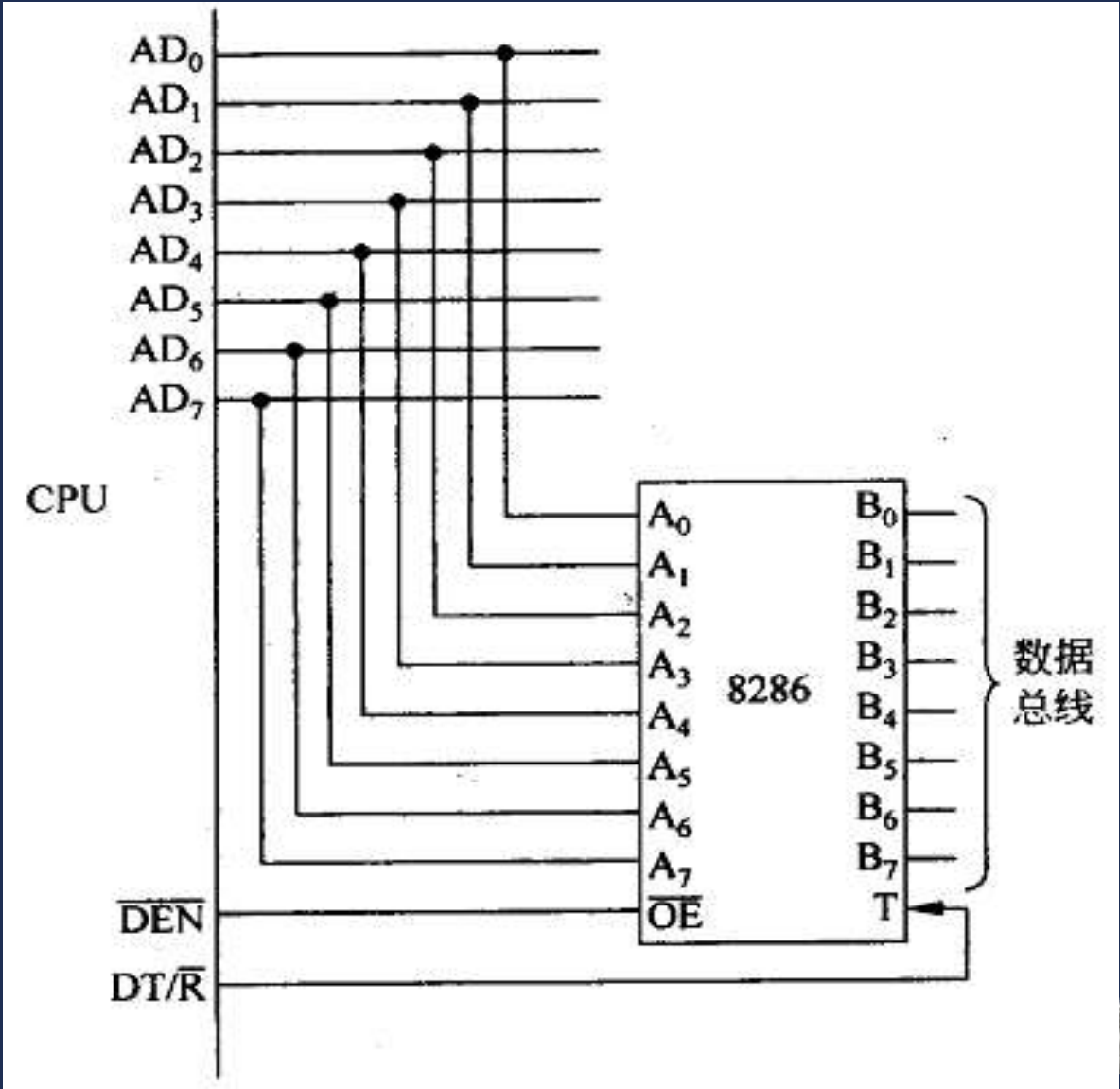
最小模式下的系统配置 (S)



8282 锁存器和8086 的连接 (S)



8286收发器和8088的连接（S）



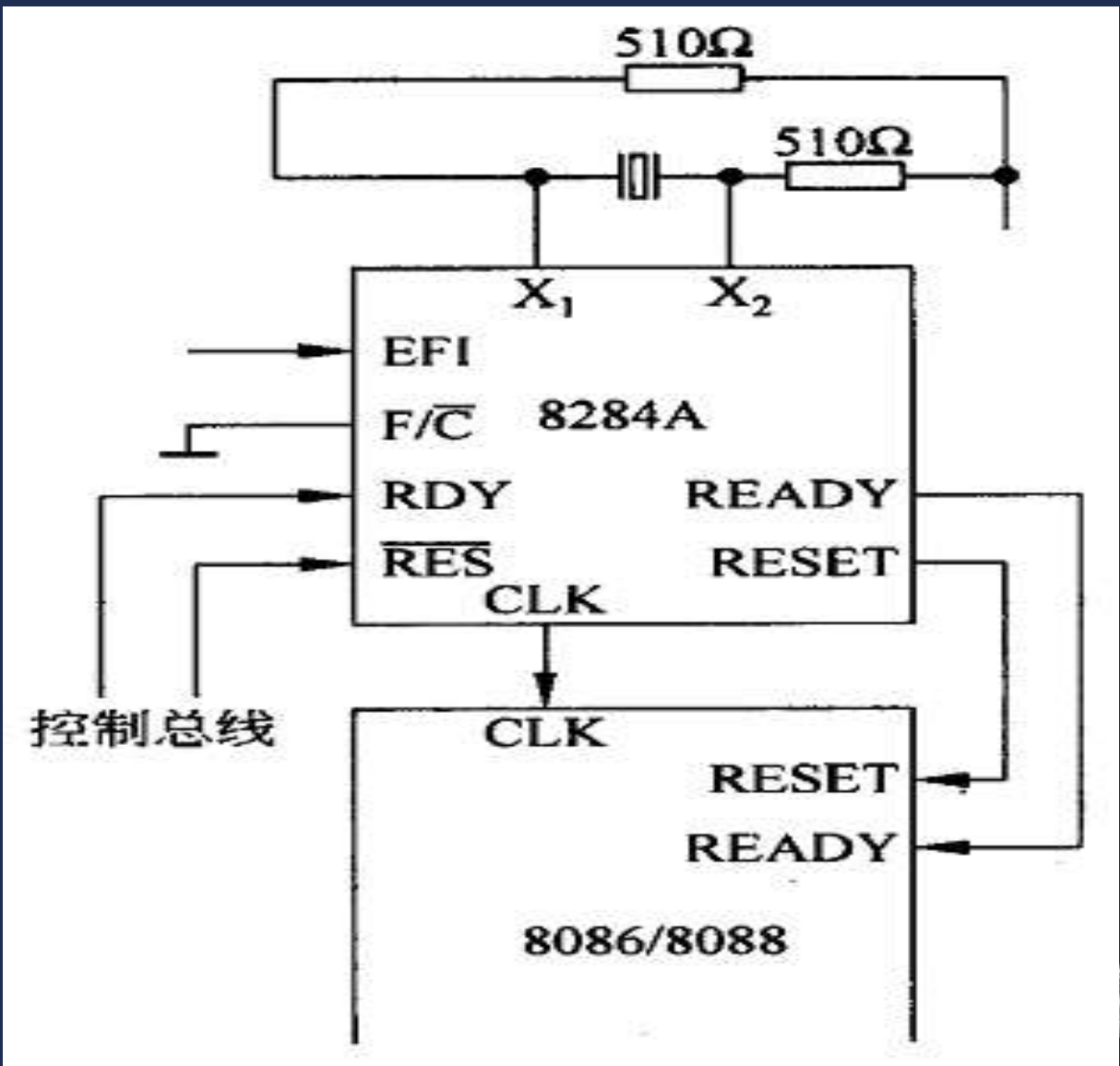
(s) 书P22表2-3

注：1、当系统中**CPU**以外的总线主部件对总线有请求时 (*)

2、若在设计总线时，想给各部件的数据信号的相位正好和**CPU**的数据信号相位相反或需要将外设数据信号反一个相位再提供给**CPU**，则可采用**8287** (*)



8284A 和8086 / 8088的连接 (s)



(四) 最大模式 (*)

24~31引脚有新的定义

1、**QS1, QS0**: 指令队列状态信号输出 (*)

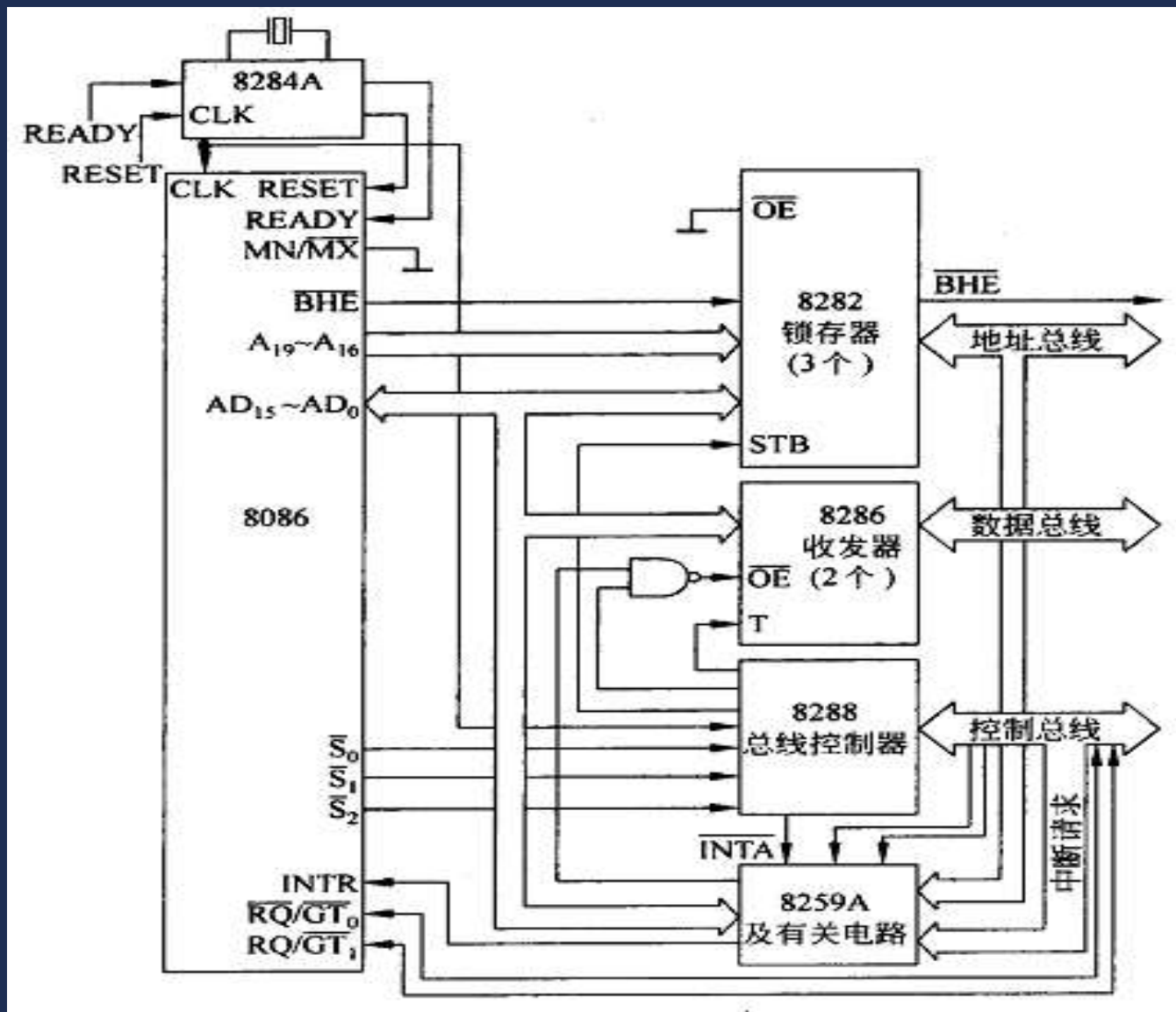
2、**S2, S1, S0**: 总线周期状态信号输出
(*)

3、**LOCK**: 总线封锁信号输出

4、**RQ/GT1, RQ/GT0**: 总线请求信号输入/
总线授权信号输出



8086在最大模式下的典型配置 (S)



二、8086的操作和时序

(一) 系统的复位和启动操作

表 2.6 复位时各内部寄存器的值

标志寄存器	清 零
指令指针(IP)	0 0 0 0 H
CS 寄存器	F F F F H
DS 寄存器	0 0 0 0 H
SS 寄存器	0 0 0 0 H
ES 寄存器	0 0 0 0 H
指令队列	空
其他寄存器	0 0 0 0 H

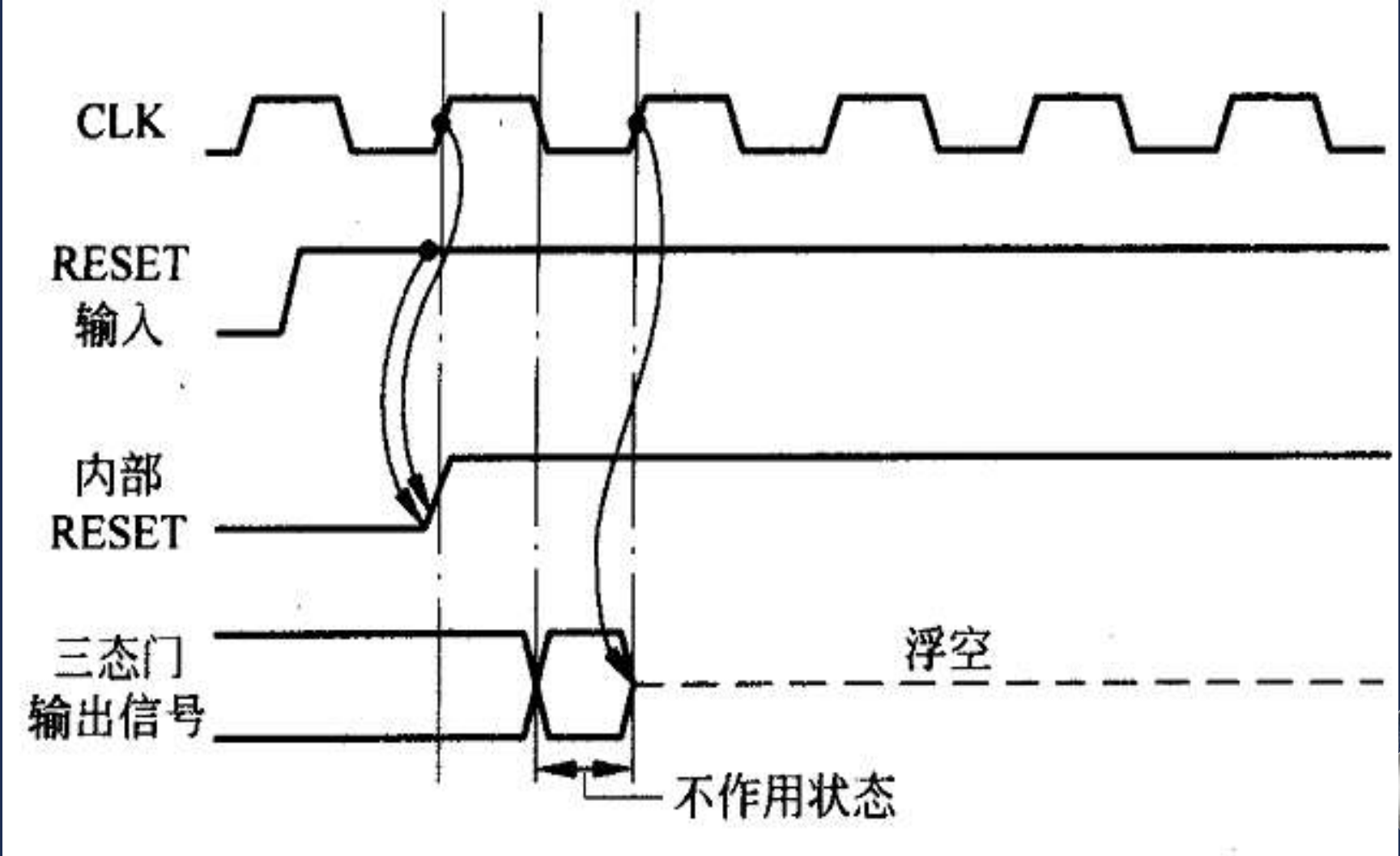
注：

1、复位后，从FFFF0H处开始执行程序。

2、由于标志寄存器被清零， $IF=0$ ，从INTR引脚进入的可屏蔽中断都得不到允许。



8086的复位操作时序 (S)

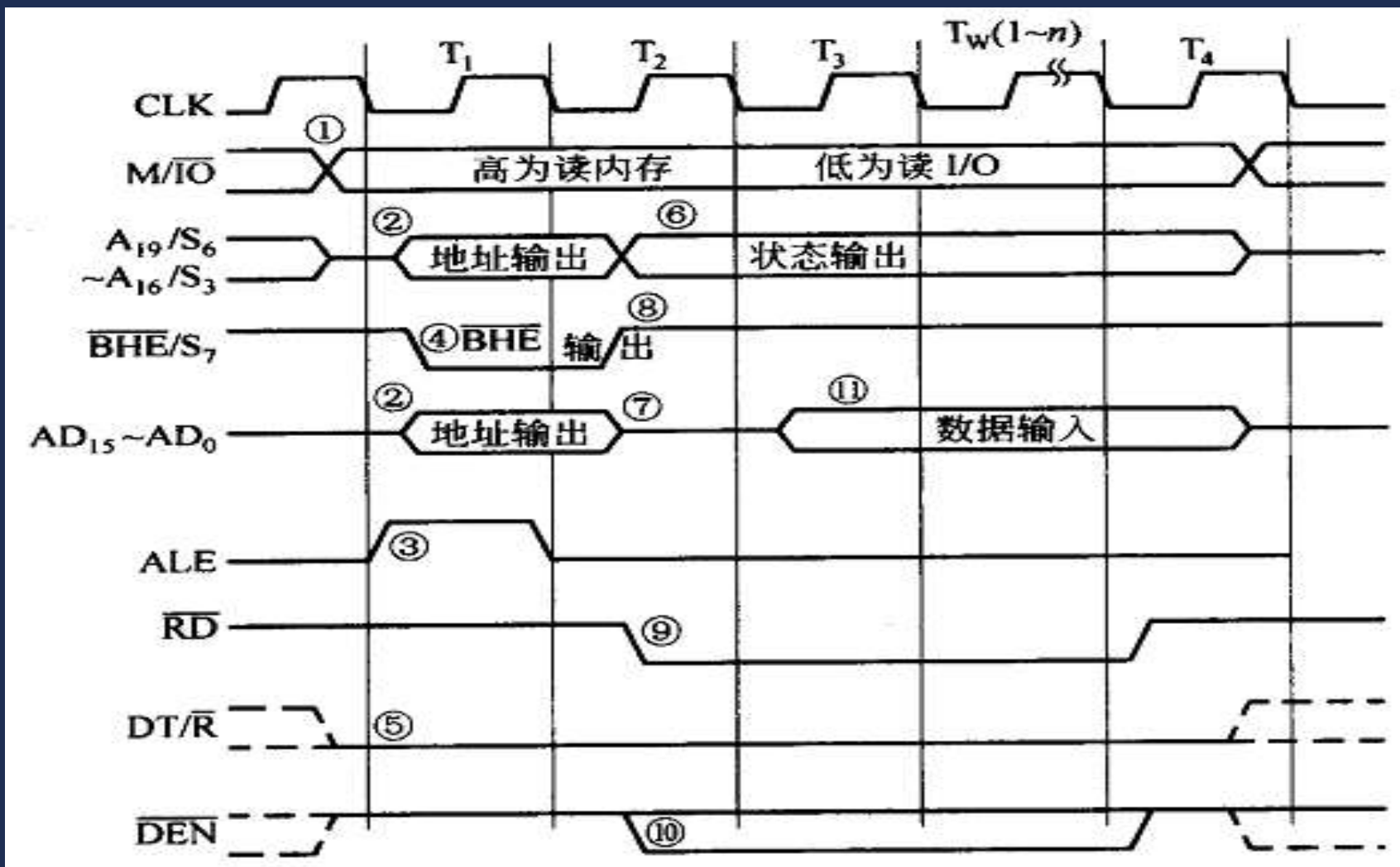


复位操作时8086的总线信号 (S)

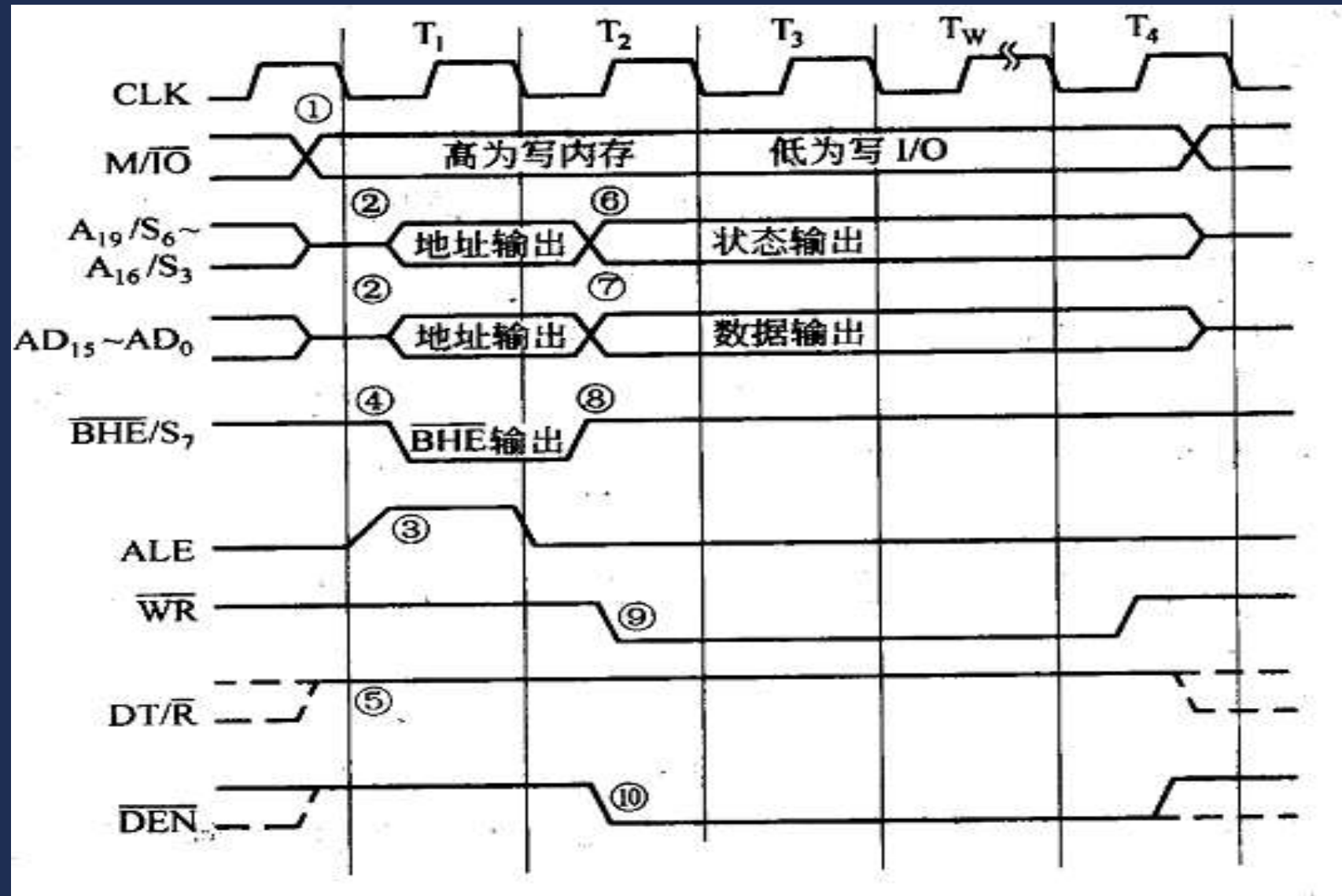
信 号	状 态
AD ₁₅ ~ AD ₀	先置成不作用状态，再进入三态。 不作用状态占进入三态前的半个时钟周期（即时钟为低电平期间）
A ₁₉ /S ₆ ~ A ₁₆ S ₃	
$\overline{\text{BHE}}/\text{S}_7$	
S ₂ (M/ $\overline{\text{IO}}$)	
S ₁ (DT/ $\overline{\text{R}}$)	
S ₀ ($\overline{\text{DEN}}$)	
$\overline{\text{LOCK}}(\overline{\text{WR}})$	
$\overline{\text{RD}}$	
$\overline{\text{INTA}}$	
ALE	
HLDA	低 低 高 高 低 低
$\overline{\text{RQ}}/\overline{\text{GT}}_0$	
$\overline{\text{RQ}}/\overline{\text{GT}}_1$	
QS ₀	
QS ₁	

(二) 总线操作（分为读操作和写操作）

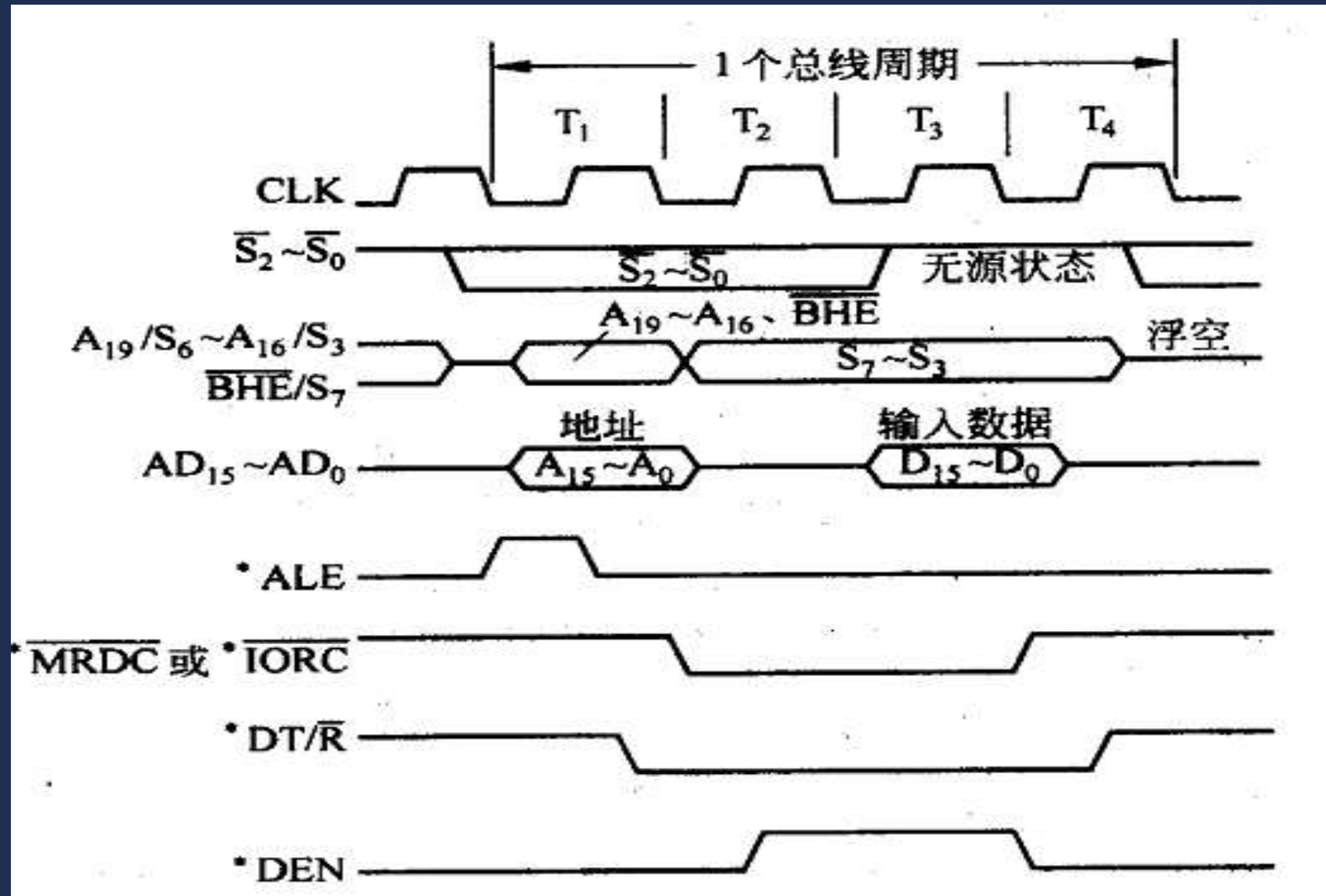
1、最小方式下的总线读操作（s）



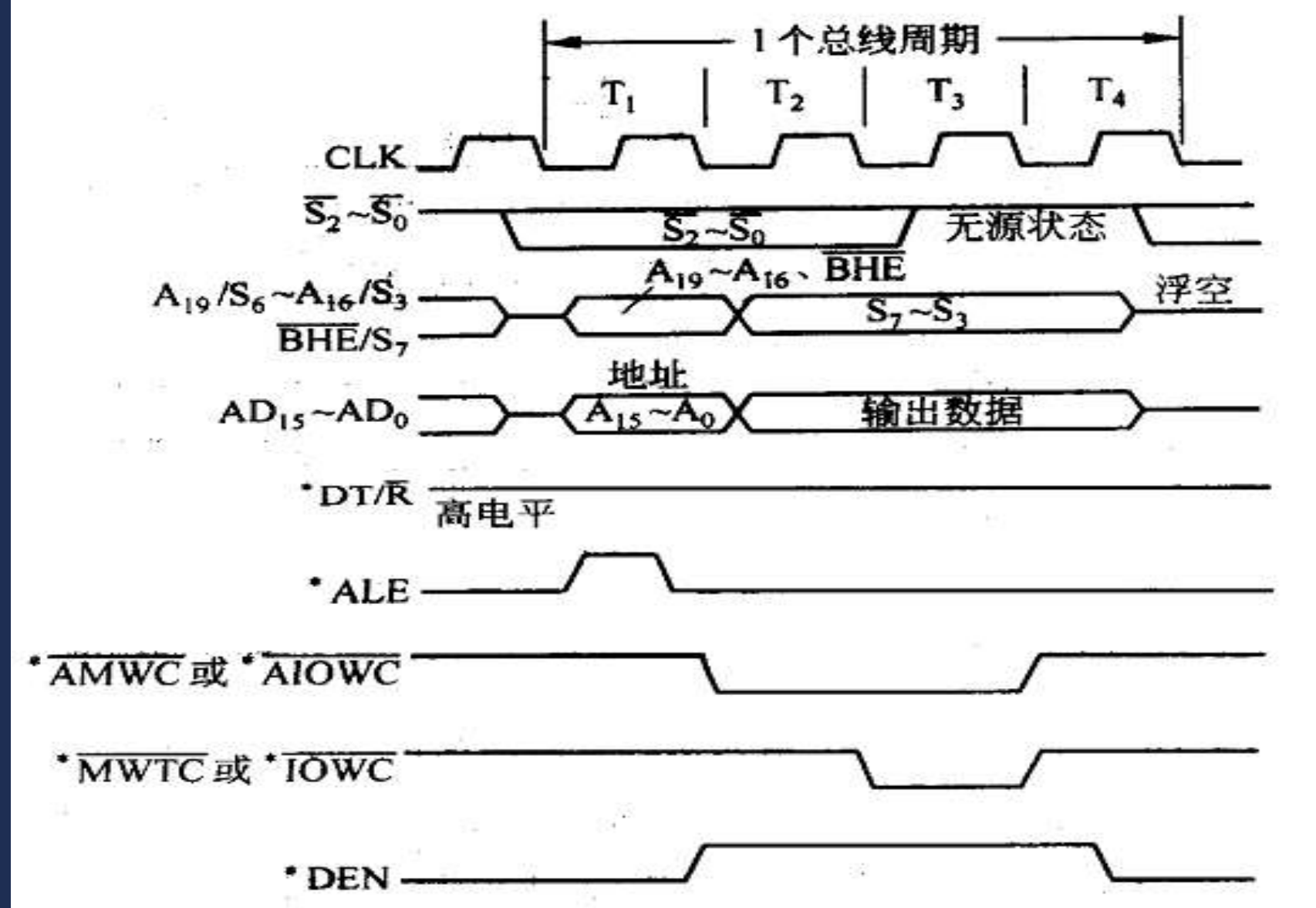
2、最小方式下的总线写操作 (s)



3、最大模式下的总线读操作时序 (s)



4、最大模式下总线写操作时序 (s)



5、总线空操作

只有在**CPU**和内存及**I/O**接口之间传输数据时，**CPU**才执行总线周期。**CPU**在不执行总线周期时，总线接口部件就不和总线打交道，此时，进入总线空闲周期**T1**。

在空闲周期中，尽管**CPU**对总线进行空操作，但在**CPU**内部，仍然进行着有效的操作。



(三) 中断操作和中断系统 (*)

1、8086的中断分类

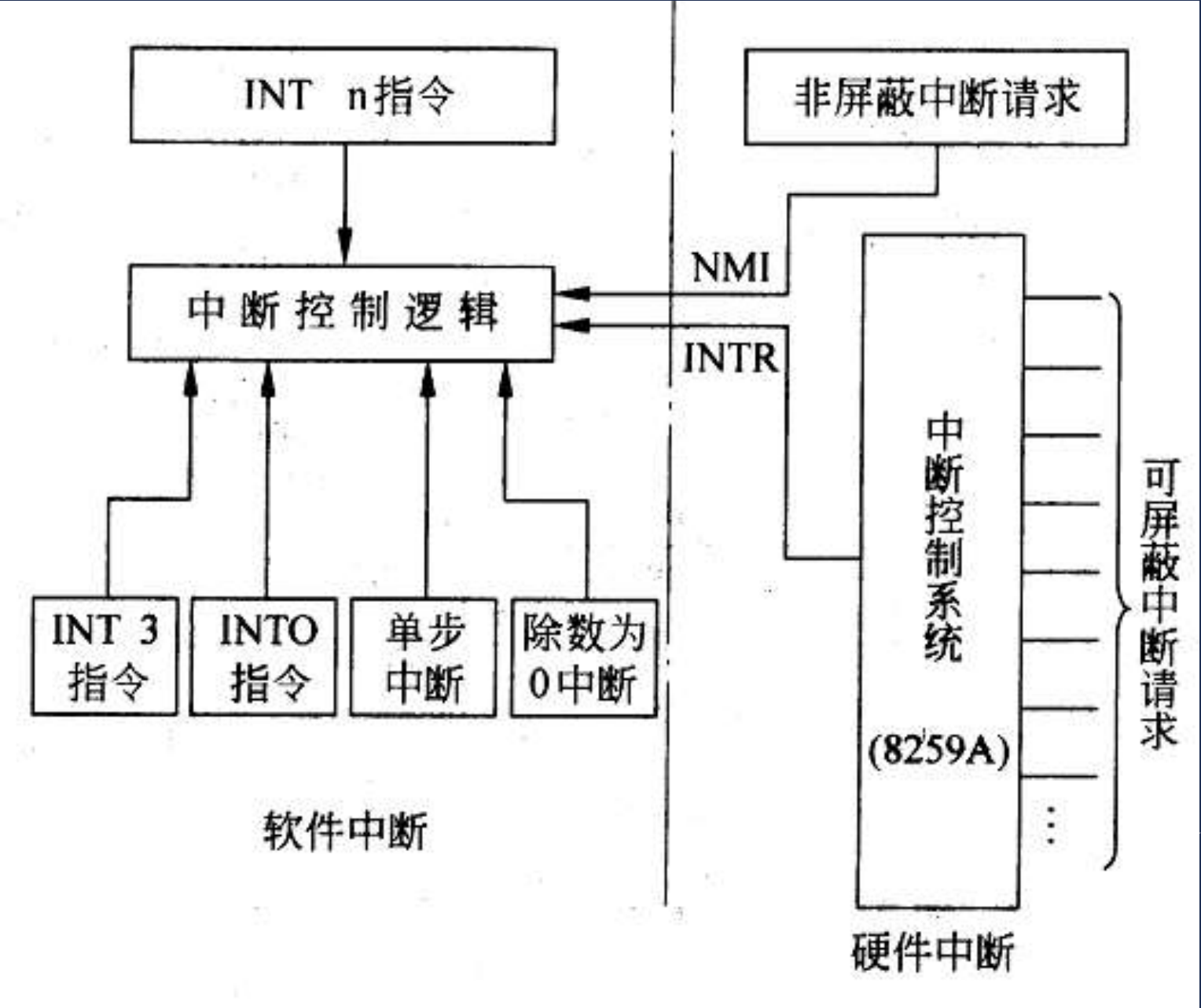
8086/8088可以处理256种不同的中断，每个中断对应一个类型码，所以，256种中断对应的中断类型码为0~255。

(1) 硬件中断

a、非屏蔽中断 NMI b、可屏蔽中断 INTR

(2) 软件中断





2、中断向量和中断向量表 (*)

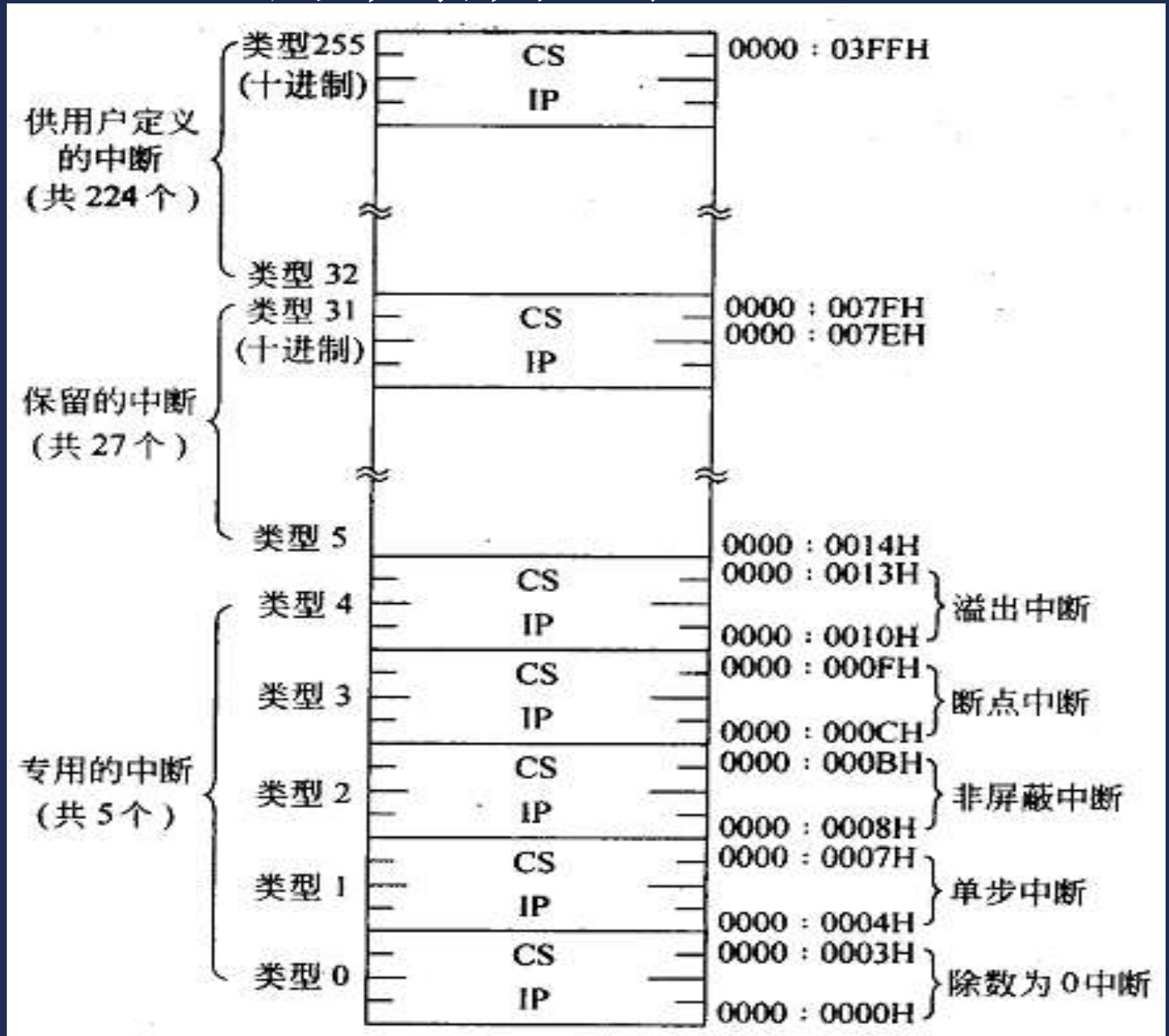
中断向量：就是中断处理子程序的入口地址，每个中断类型对应一个中断向量。一个中断向量占4个存储单元。

其中，**前2个单元**存放中断处理子程序入口地址的**偏移量 (IP)**，**低位在前，高位在后**，**后2个单元**存放中断处理子程序入口地址的**段地址 (CS)**，同样也是**低位在前高位在后**。

$256 \times 4 = 1024$ ，占用0段的**0000~03FFH**区域



8086/8088的中断向量表



注：（1）前5个是专用中断，它们有着固定的定义和处理功能。

（2）从类型5到类型31（1FH）共27个中断为保留的中断，是提供给系统使用的，用户一般不应该对这些中断自行定义。

（3）其余类型的中断原则上可以由用户定义。但是，有些中断类型目前已经有了固定的用途，比如21H类型的中断是操作系统MS-DOS的系统调用。



3、硬件中断（外部）（*）

NMI：引入非屏蔽中断。类型号为2，非屏蔽中断处理子程序的入口地址放在0段的0008H、0009H、000AH和000BH这4个单元中。其优先级最高。

INTR：引入可屏蔽中断。IF标志决定是否响应中断。



4、硬件中断的响应和时序 (*)

可屏蔽中断的响应过程:

INTR引脚用于接收**可屏蔽**中断请求信号**输入**
 （高电平有效）

INTA引脚用于中断**响应**信号**输出**（低电平有效），连续两个周期中的两个负脉冲。（*）



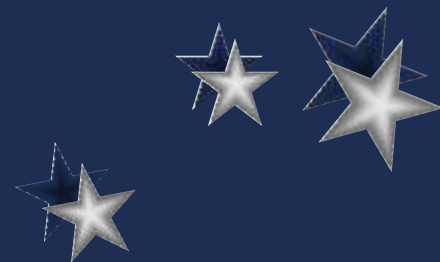
响应过程 (*) :

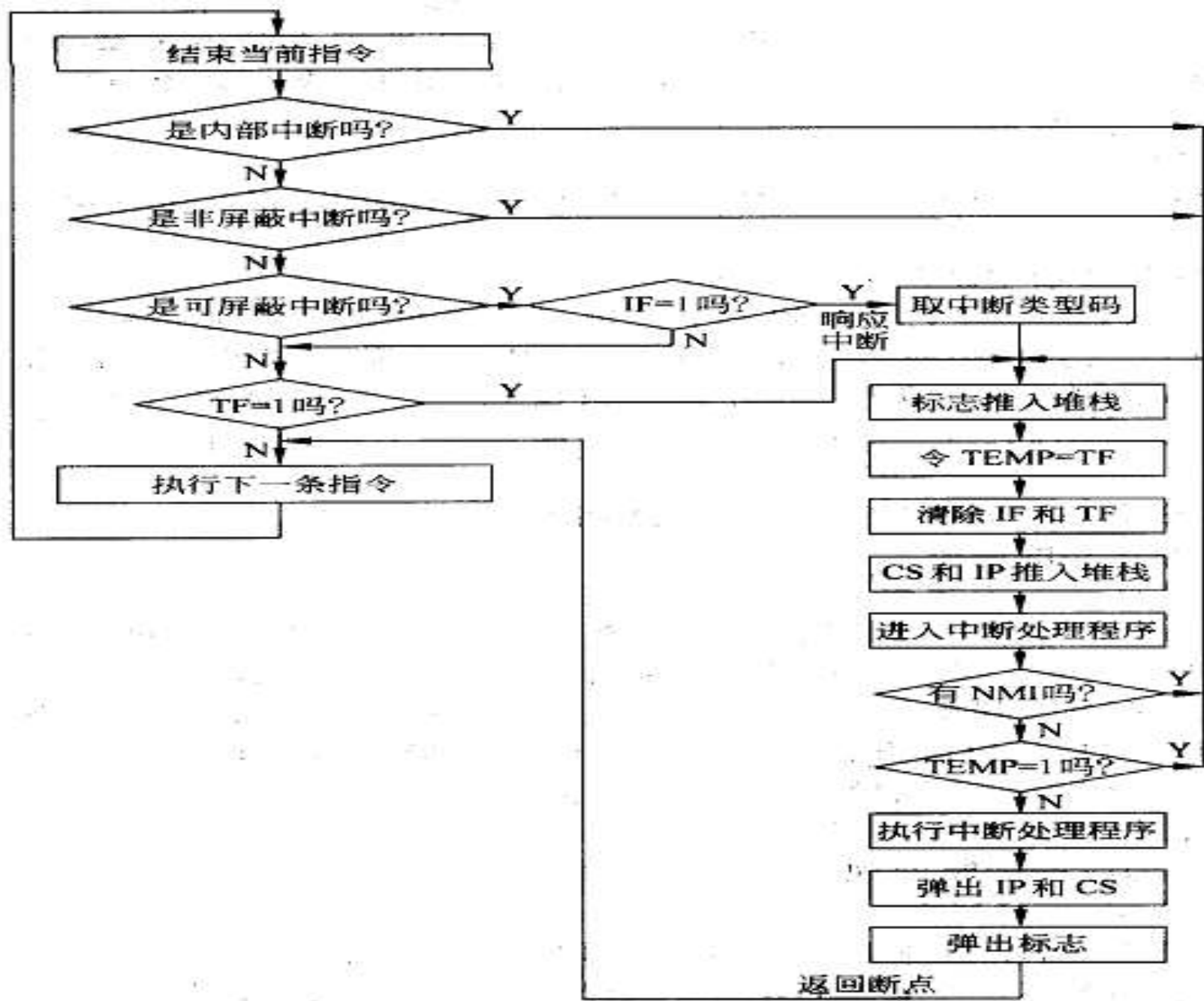
- 1、从数据总线上读取中断类型码，将其存入内部寄存器。
- 2、将标志寄存器的值推入堆栈。
- 3、IF，TF清零
- 4、断点入堆栈
- 5、根据目前得到的中断类型码，到内存0000段的中断向量表中找到中断向量，再根据中断向量转入相应的中断处理子程序。



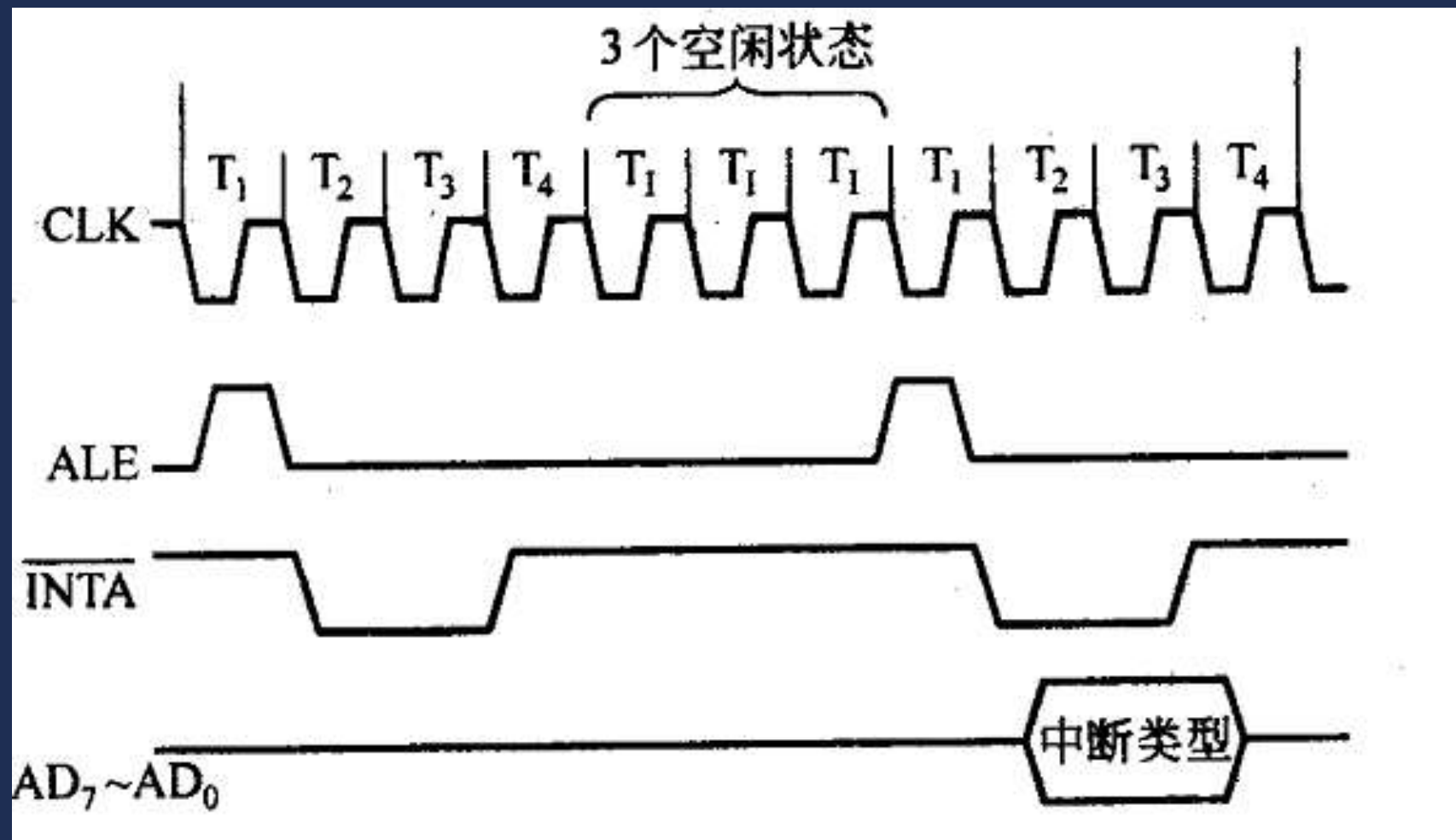
注(*)：1、可屏蔽中断请求信号是一个电平触发信号，即高电平就触发。且INTR信号的高电平必须维持到CPU响应中断才结束。

2、NMI请求信号是一个上升沿触发信号。
CPU不从外部设备读取中断类型码





8086中断响应的总线周期 (*)



5、中断处理子程序

保护现场——开中断——处理任务——关中断——恢复现场——开中断——返回

6、软件中断(*)：通过中断指令来使CPU执行中断处理子程序的方法。

中断子程序和一般子程序只是返回指令不同特点(*)：

1) 用一条指令进入中断处理子程序, 中断类型码由指令提供, 无需从DB读, 无需执行中断响应总线周期。

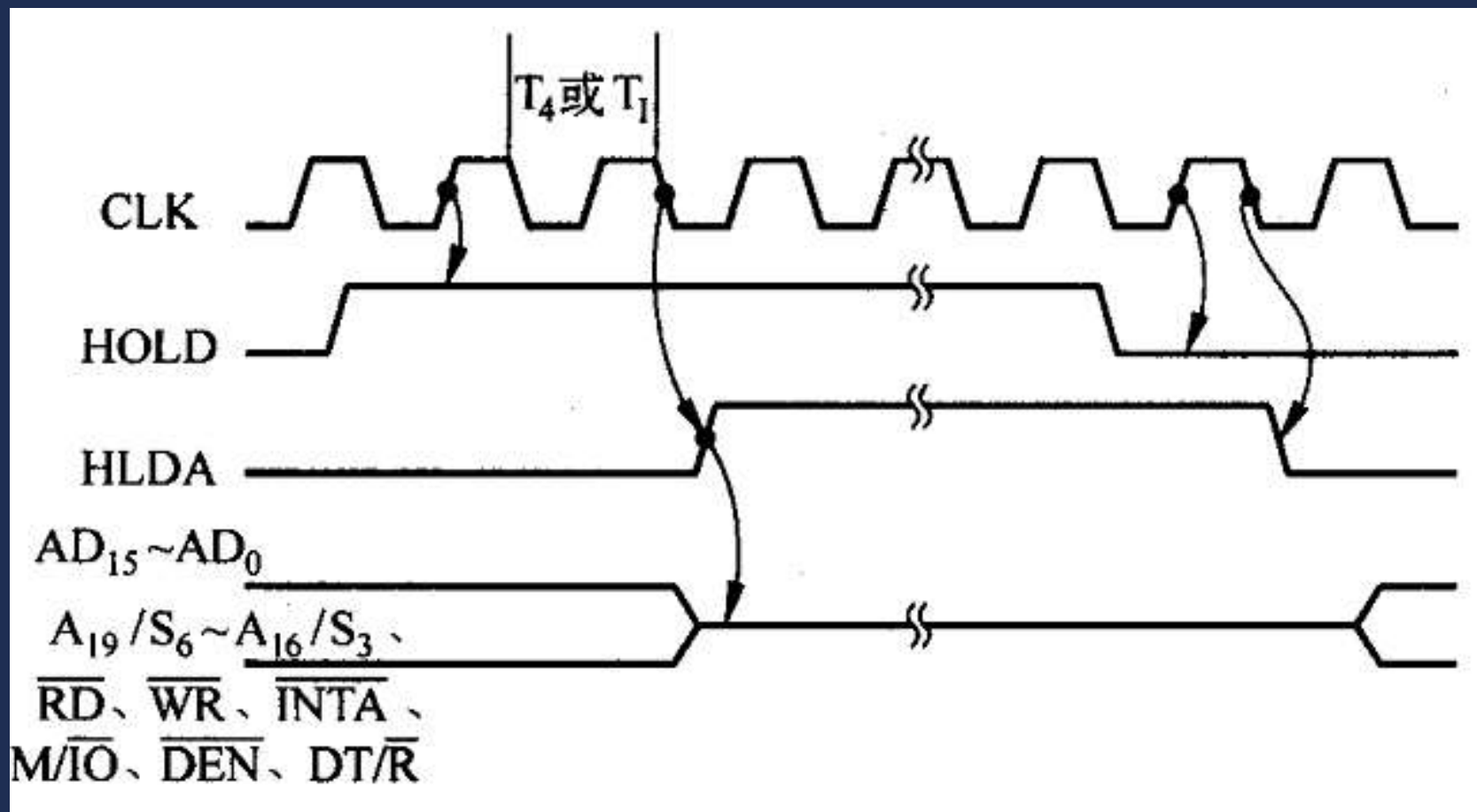


- 2) 不受中断允许标志IF的影响
 - 3) 硬件中断优先级>软件中断
 - 4) 无随机性, 可和主程序进行参数的传递。
- 而硬件中断的中断子程序和主程序互相独立

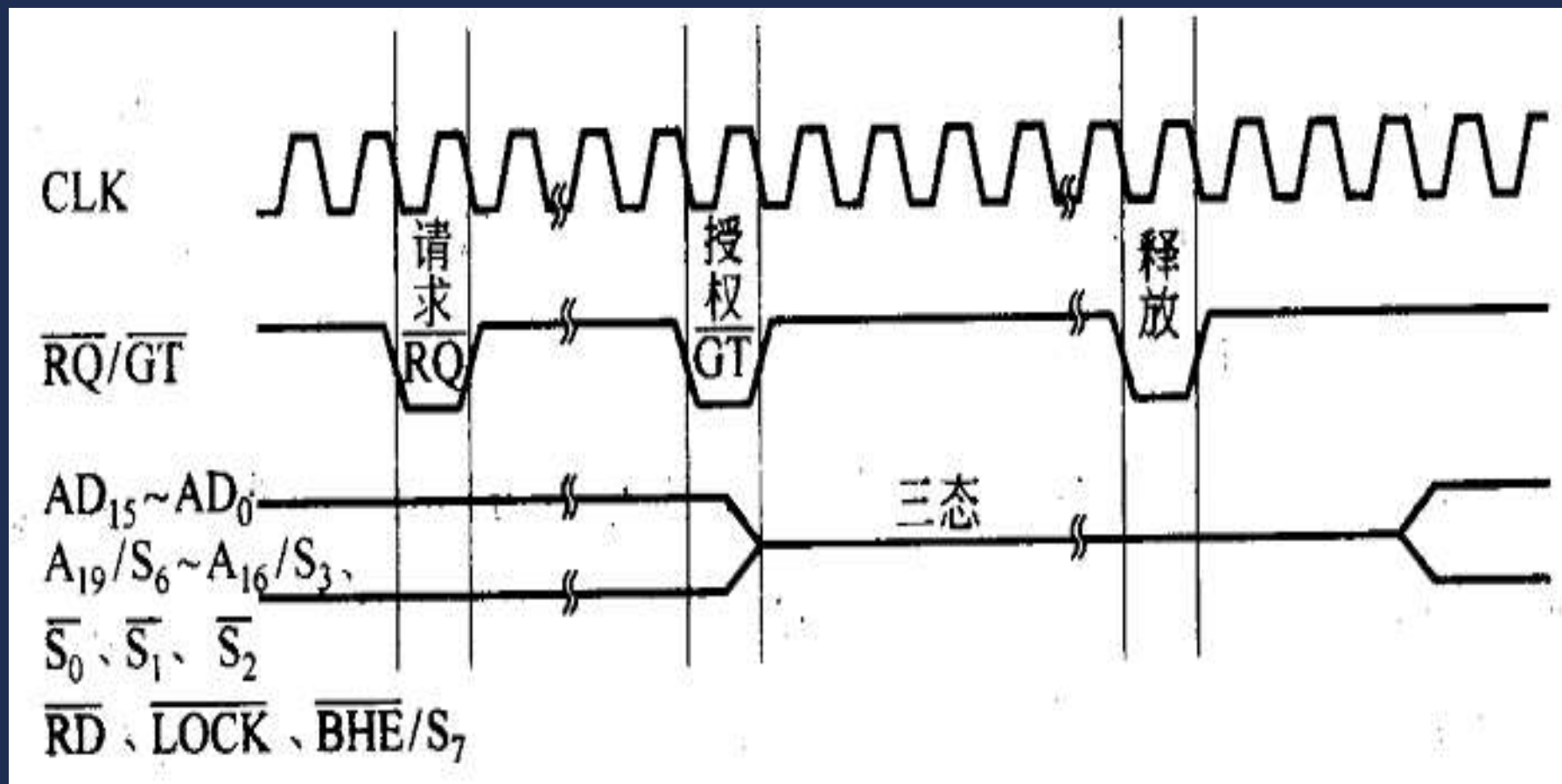
注: 常把一些常用的较大型子程序设计为中断处理子程序, 再在程序中用软件中断的方法调用它们。所以, 软中断实际上就是一种特殊的子程序调用。



四、最小模式下的总线保持 (*)



五、最大模式下总线请求 / 允许 / 释放时序 (*)



六、8086的I/O编址 (*)

- 1、什么是端口？
- 2、什么是端口地址？
- 3、编址方式
 - 1) 单独编址
 - 2) 统一编址



小结

- 1、BIU和EU。相互独立且相互配合，实现并行流水线，是8086/8088CPU体系结构最突出的特点。
- 2、引脚分时复用。
- 3、两种工作模式
- 4、8086/8088的各种操作时序
- 5、Reset后各寄存器的情况
- 6、总线空操作

