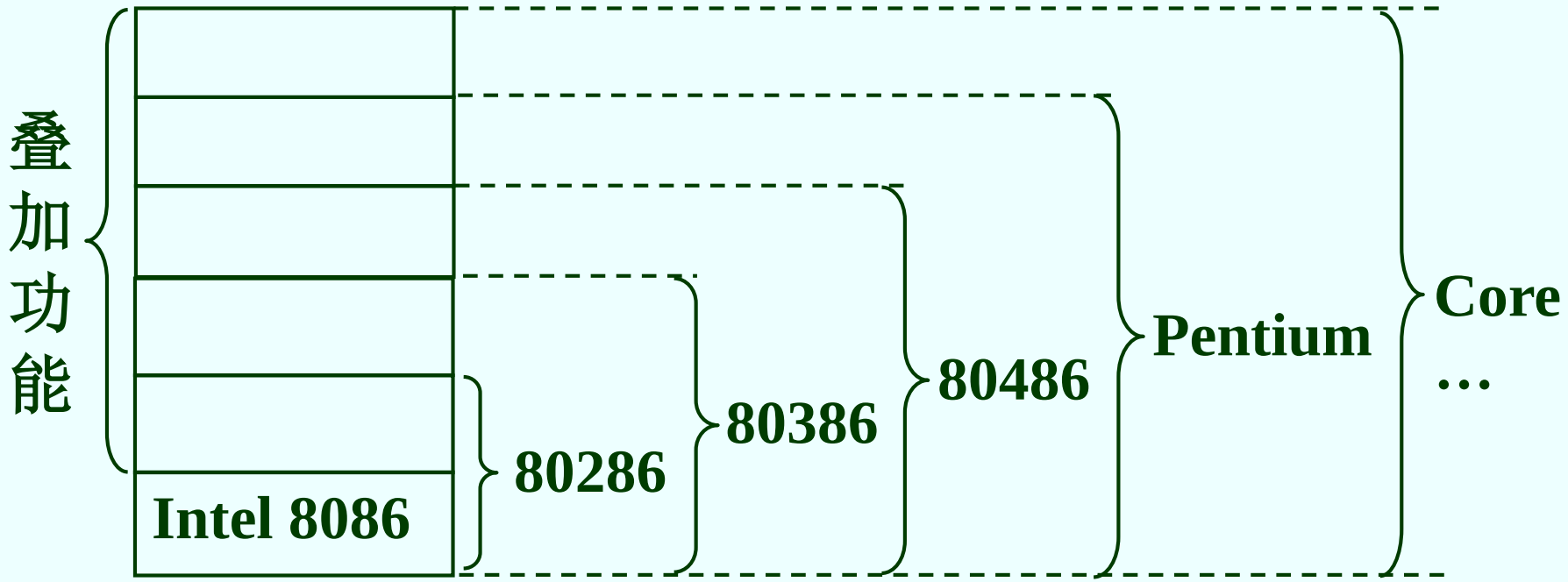
第二章 Intel 系列处理器

- Intel×86系列处理器



- 计算机领域, 在一段相对长的时间内

“产品过时, 技术不过时”

第一节 Intel 8086处理器

一、内部组成结构

(一) 主要功能模块

- (1) 执行部件 EU
  - 指令译码器
  - 运算器ALU
  - 通用寄存器和标志位寄存器
  - 控制逻辑
- (2) 总线接口部件 BIU
  - 段寄存器
  - 地址形成逻辑
  - 总线控制逻辑及总线驱动器
  - 指令队列
  - 指令指针

(二) 寄存器

通用寄存器

AX	AH	AL
BX	BH	BL
CX	CH	CL
DX	DH	DL

指针及变址寄存器

- SP 堆栈指针
- BP 基地址变址寄存器
- SI 源变址寄存器
- DI 目的变址寄存器

段寄存器

- CS 代码段基地址寄存器
- DS 数据段基地址寄存器
- SS 堆栈段基地址寄存器
- ES 附加段基地址寄存器

专用寄存器

- IP 指令指针
- FR 标志位寄存器

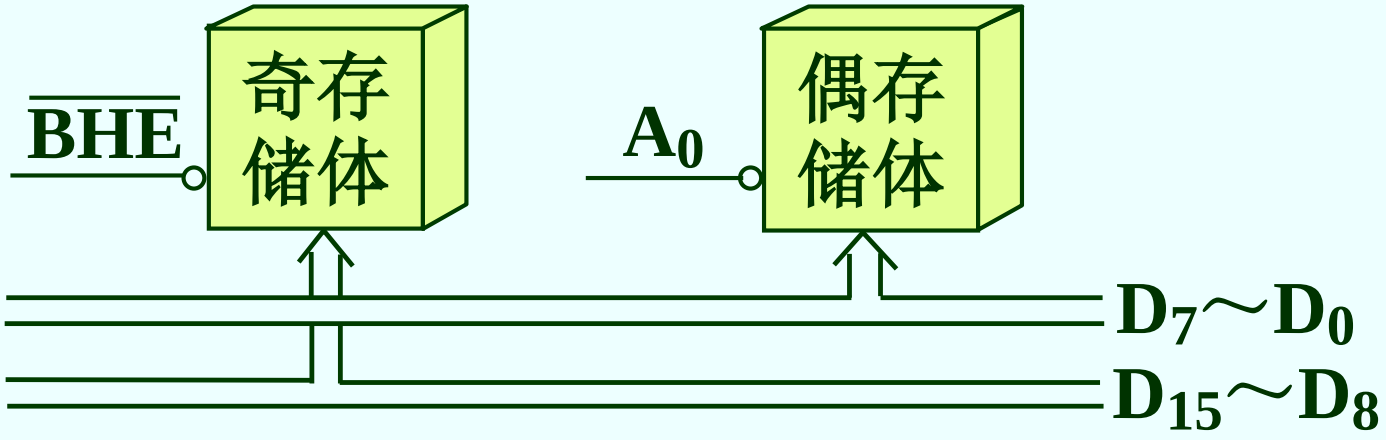
二、存储器的结构

1、数据存放的格式

对一个16位的操作数，存放方式是：存放在两个连续存储单元，低字节存放在偶数地址单元(起始地址)，高字节存放在相邻的奇数地址单元。

按上述格式存放，存/取一个16位的数据只需一个总线周期，否则，需要两个总线周期。

8086用A_0=0选择偶存储体，用$\overline{BHE}$选择奇存储体，如下图所示：



A_0	$\overline{BHE}$	操作
0	0	同时访问奇偶存储体
0	1	访问偶存储体
1	0	访问奇存储体
1	1	无效

若不按照上述格式存放数据, 则访问一个16位的字, 需要2个访存周期

例: MOV (2013H), AX; (需要2个访存周期)

将AX的16位数据存入2013H开始的单元。

指令执行时，由于地址2013H奇数，A_0=1，$AX_{7\sim0}$存入地址2013H单元(一个总线周期)，然后CPU将地址自动加1(此时A_0=0)，将$AX_{15\sim8}$存入2014H单元(再一个总线周期)。

2、存储器分段以及地址的形成

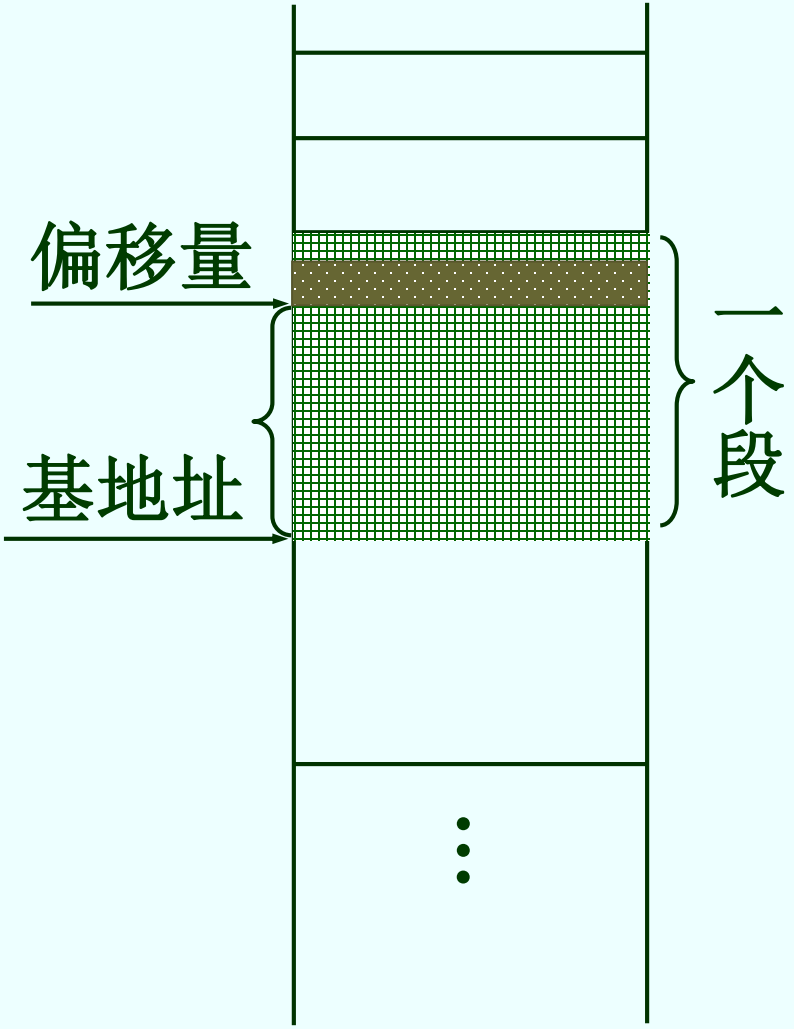
将存储器逻辑上划分为每64K为一个段

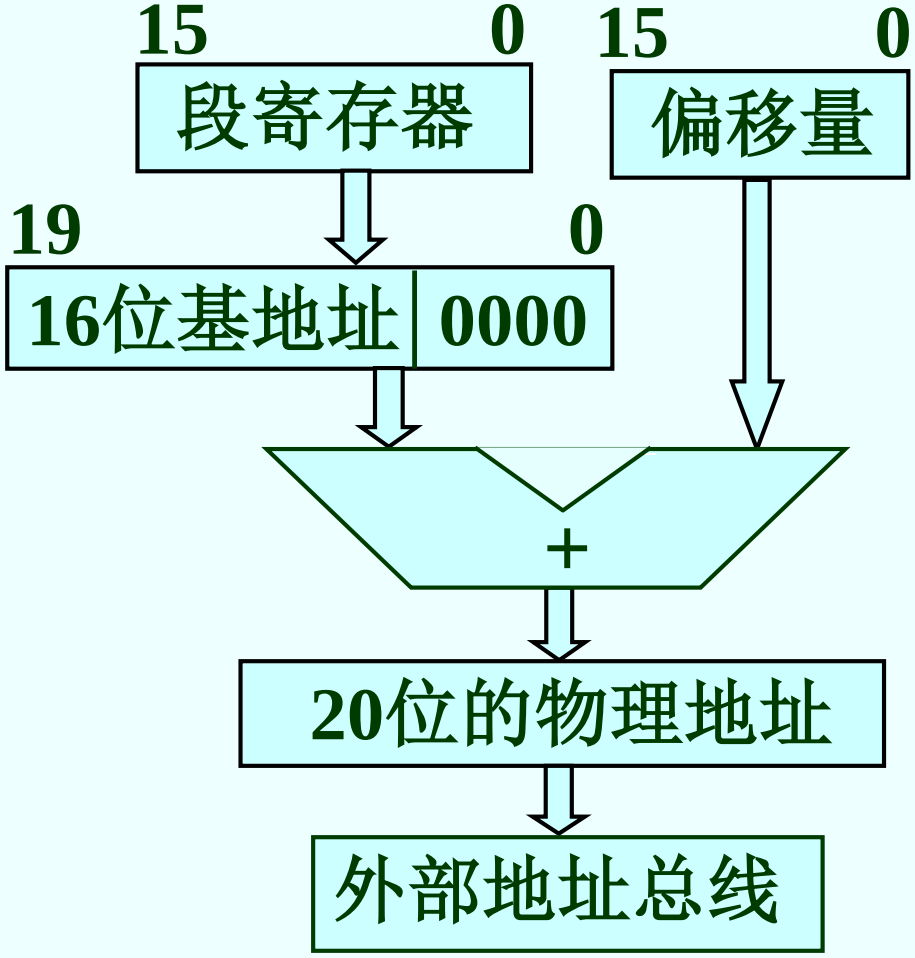
实际访问单元地址= 段基地址×16+段内偏移量

(物理地址)

↓

左移4位





为什么8086/8088的存储器要分段?

三、8086 I/O系统(中断系统)

(一) 中断源

1. 外部中断
   - INTR 屏蔽中断
   - NMI 非屏蔽中断
2. 内部中断

   处理器运行过程中, 由于其内部某种异常或错误而在内部自动产生的, 比如:
   - 除法出错中断: 如商大于目标寄存器所能表示的范围;
   - 单步中断;
   - 断点中断;
   - 溢出中断等;

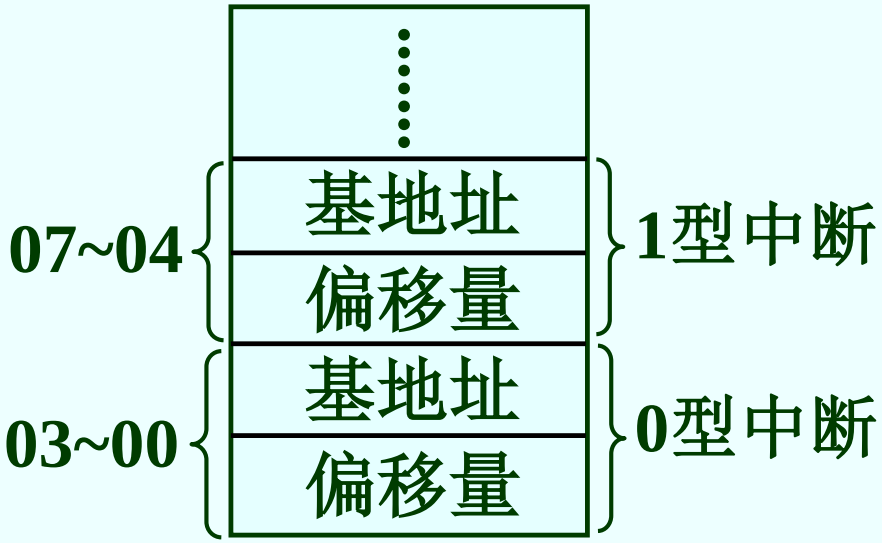
(二) 转入中断服务程序

原理上(向量中断方式):

中断请求信号INTR→处理器→中断响应周期→处理器取得中断类型码(中断向量)→将类型码转换为中断服务器程序地址→程序计数器→中断服务器程序。

对8086处理器:

将中断向量乘4，结果作为地址查找中断向量表，服务程序首地址放在该向量表中(中断向量表共1024字节)如图所示:



第二节 Intel 80286

与8086的显著区别：

减少引脚数量，便于芯片封装

1. 地址线和数据线不再分时复用，简化了硬件设计；
2. 增加了地址线的宽度，物理地址空间增加到16M
3. 增加了新的指令，以增强其控制能力。
4. 引入存储管理中的虚存管理机制。通过“虚地址”和“保护”两重功能对存储器管理提供了支持，加强了对多用户/多任务运行的管理能力。

一、80286处理器的结构

(一) 主要构成

IU 指令部件
EU 执行部件
> 相当于8086的EU

AU 地址部件
BU 总线部件
> 相当于8086的BIU

(1) AU: 按EU请求的寻址方式形成物理地址

主要构成:

- 段基地址寄存器
- 段容量寄存器
- 段限检查器
- 地址加法器
- 描述子表基地址寄存器

(2) BU: 按AU形成的物理地址, 完成EU所请求的寻址过程和数据传送。主要构成:

- 总线时序控制器
- 总线地址驱动器
- 数据收发器
- 6字节指令预取队列

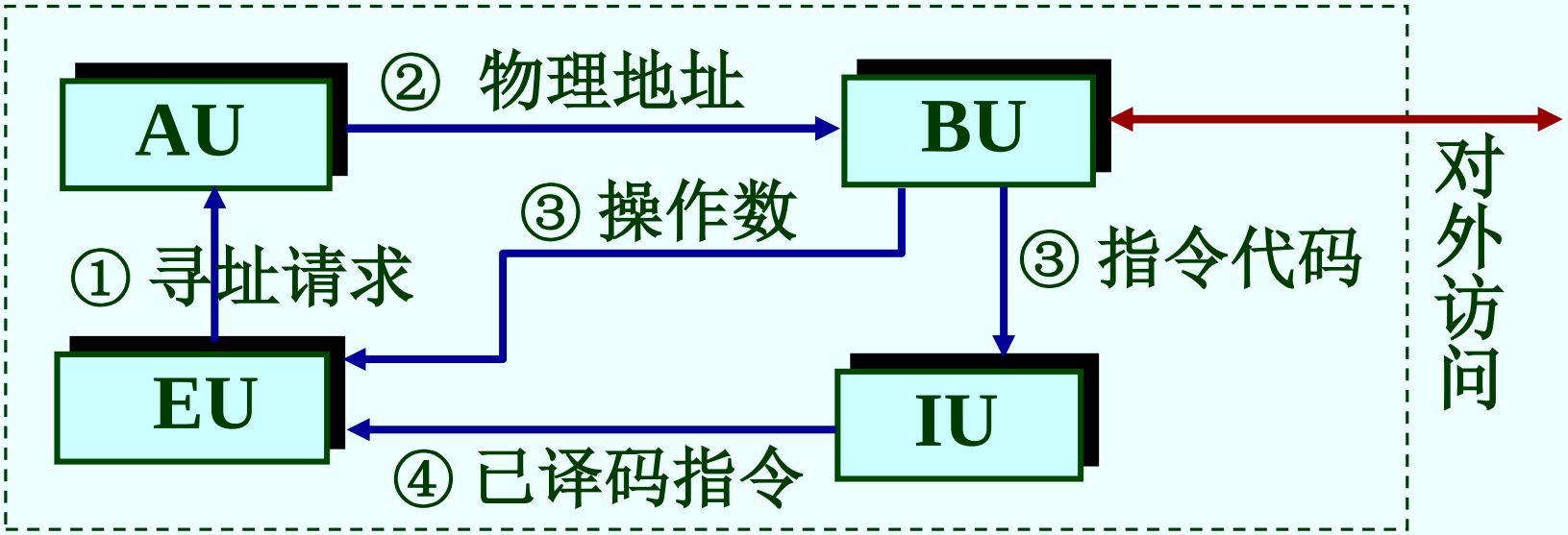
(3) IU: 完成指令译码

包含指令译码器和指令队列

(4) EU: 执行指令所要求的功能。

包括运算器、微程序控制器、寄存器以及相关时序电路

以上四个部件之间的逻辑关系是:



(二) 寄存器的结构

第一类: 通用寄存器, 与8086相同。

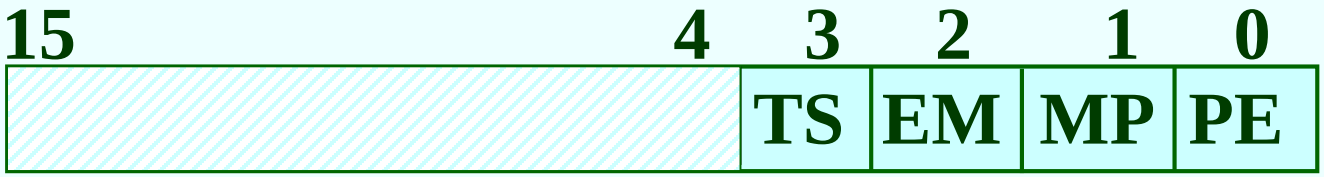
第二类: 指针和变址寄存器, 与8086相同

第三类: 段寄存器与8086相同, 用法上有所区别:

当 { 实地址模式: 与8086相同
　 { 虚地址保护模式: 存放选择子, 而非段基地址

第四类: 状态与控制寄存器

- 标志位寄存器FR: 在8086基础上增加了3位:
  - 任务嵌套标志NT (1位)
  - I/O特权级标志IOPL (2位)
- 新增机器状态字MSW (16位寄存器)



  - PE(保护模式标志)
    当PE←1, 处理器进入虚地址保护模式
  - MP(协处理器监视位)
    系统配置有80287时, 自动将MP←1,否则MP←0

- **EM(协处理器仿真位)**

  系统中没有协处理器且应用程序需要协处理, 则用软件仿真协处理器的功能, 将EM←1。

  在程序执行过程中,若出现协处理指令,并有MP=0且EM=1，系统自动产生异常中断7，进入相应的协处理仿真程序。

- **TS(任务切换标志)**

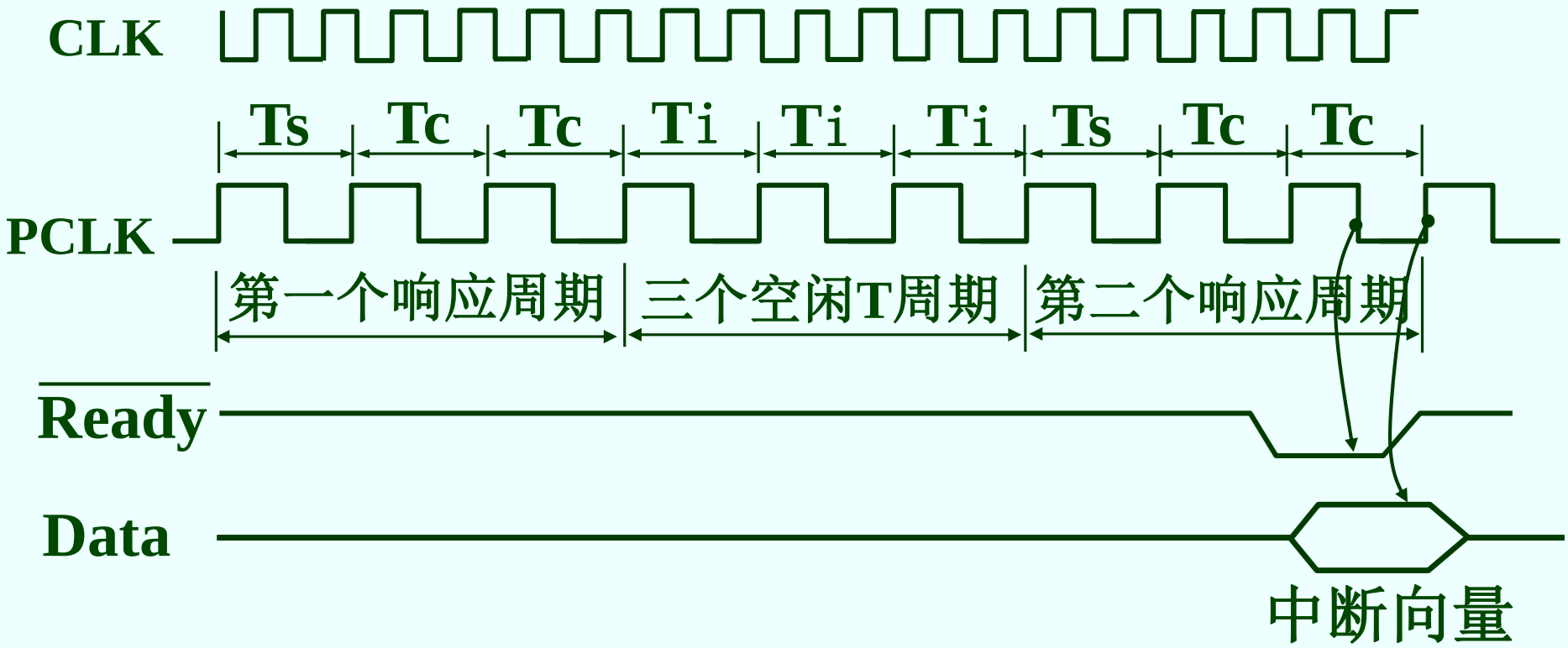
  用于标识发生了任务切换，如果新任务需要使用协处理器, 需要防止原任务的协处理程序现场被破坏

  工作过程: 发生任务切换时, 将TS←1。当MP=1时, 如果遇到协处理指令时, 则自动产生异常中断7, 在该异常中断处理程序中, 进行协处理器的现场切换。

二、80286总线周期描述

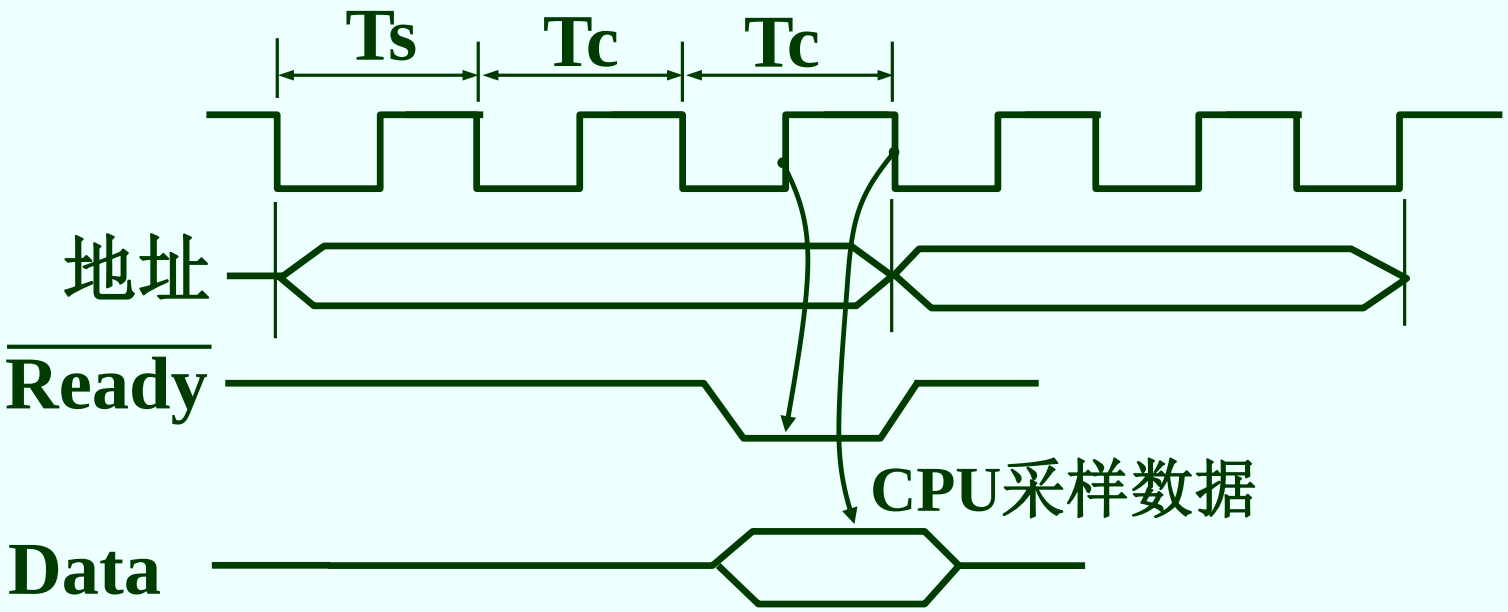
数据线: D_{15}～D_0； 地址线: A_{23}～A_0

1、中断响应周期



2、存储器读周期

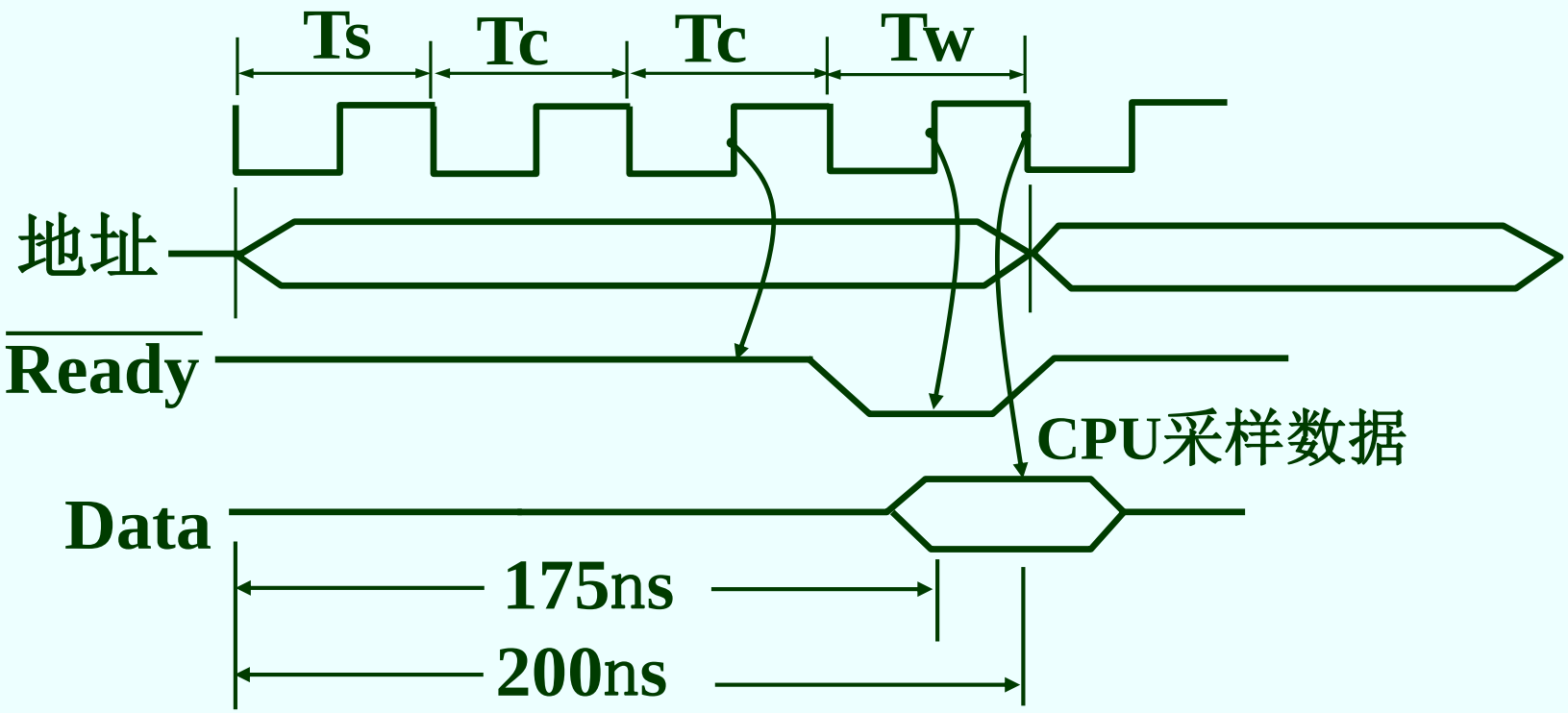
(1) 无等待周期的存储器读



注: 实际系统要求在第二个Tc上升沿之前10ns, 数据必须准备好, Ready才会有效。

(3) 需要插入等待周期的存储器读

假设: 每个T周期为50ns, 存储器读数据的建立时间(简化为读存储器的速度)140ns。因为从进入Ts到第二个Tc的上升沿为125ns, 则需要插入一个Tw。

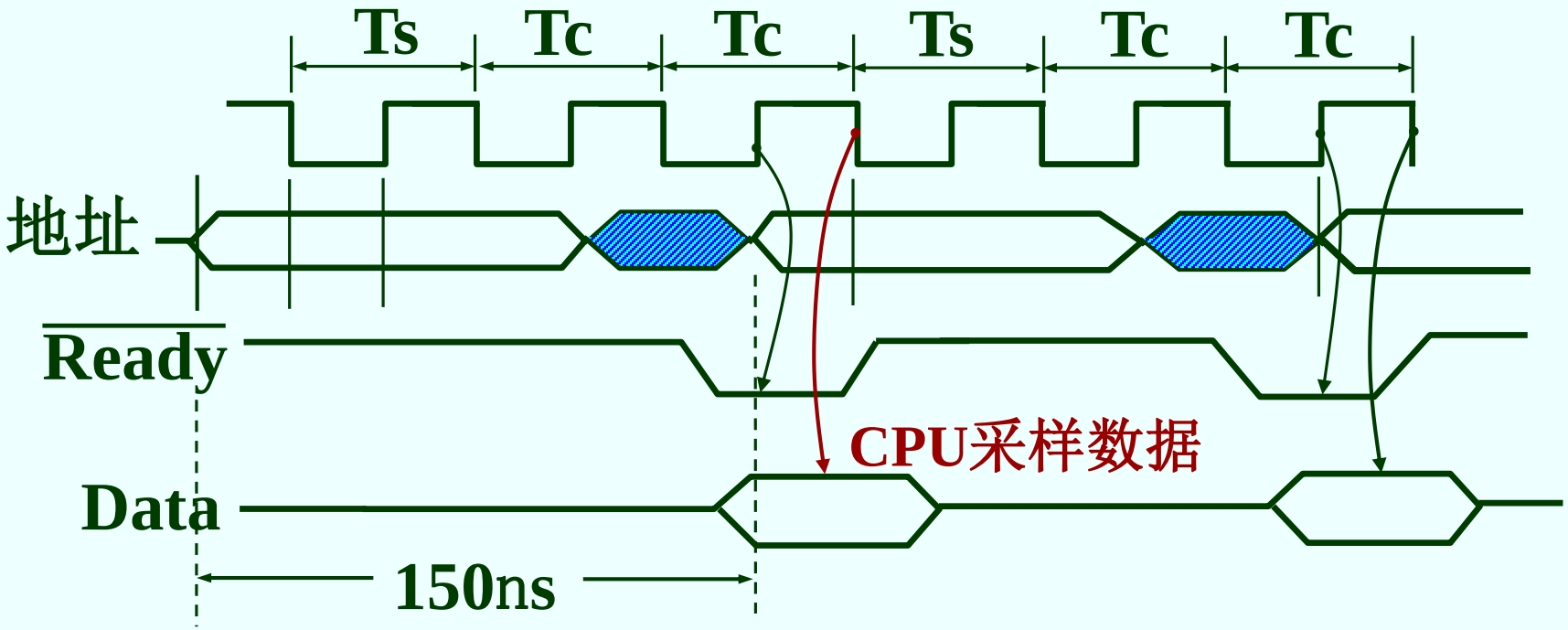


3、地址流水线

仍然假设每个T周期为50ns, 存储器读数据的建立时间为140ns

目标: 尽可能减少等待周期Tw

措施: 提前发出地址



地址流水线的简单描述:

当前周期发出下一个总线周期所需要的地址;

或 上一个周期发出当前周期所需要的地址。

结论:

采用地址流水线后, 由于地址信号的提前建立, 与非地址流水线相比, 可以尽量减少插入Tw等待周期。因而加快了访存速度。

(但并没有提高存储器的速度)。

三、80286的工作模式

(一) 实地址模式

系统开机复位时,自动进入实地址模式, A_{23}~A_{20}自动置为0, 以 A_{19}~A_0寻址1M的存储空间。

(二) 虚地址保护模式

主要针对在多任务机制中的存储管理、不同任务之间隔离与保护的。

1. 虚地址保护模式的基本概念

两个方面的含义:

(1) 虚地址: 程序设计者可以寻址一个比实际物理地址空间(16M)大得多的虚存空间(1000M)。

(2) 保护

保护什么？—对存储空间的(数据和程序)保护

为什么需要保护？— 多任务机制的引入

保护的具体内容:

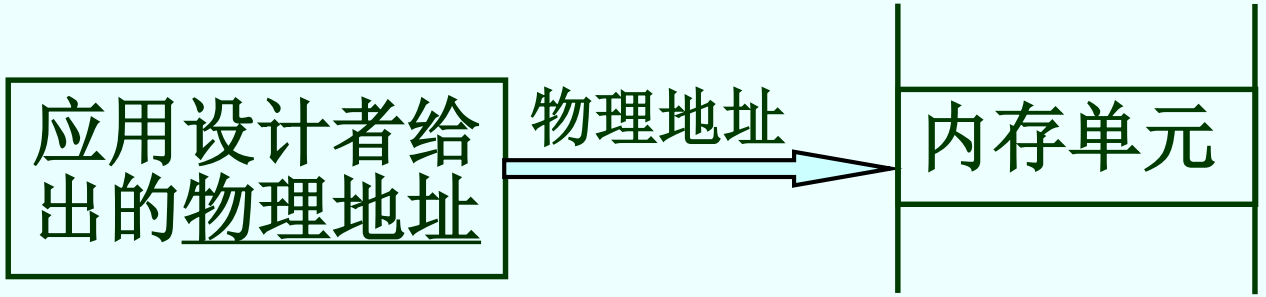
- 地址空间的保护
  避免多任务机制下的越界访问
- 特权级的保护
  如: 防止应用软件修改系统软件或系统数据
- 访问权限的保护
  如: 可读或可读/写、可执行或可读/可执行等

2. 保护模式下的寻址过程

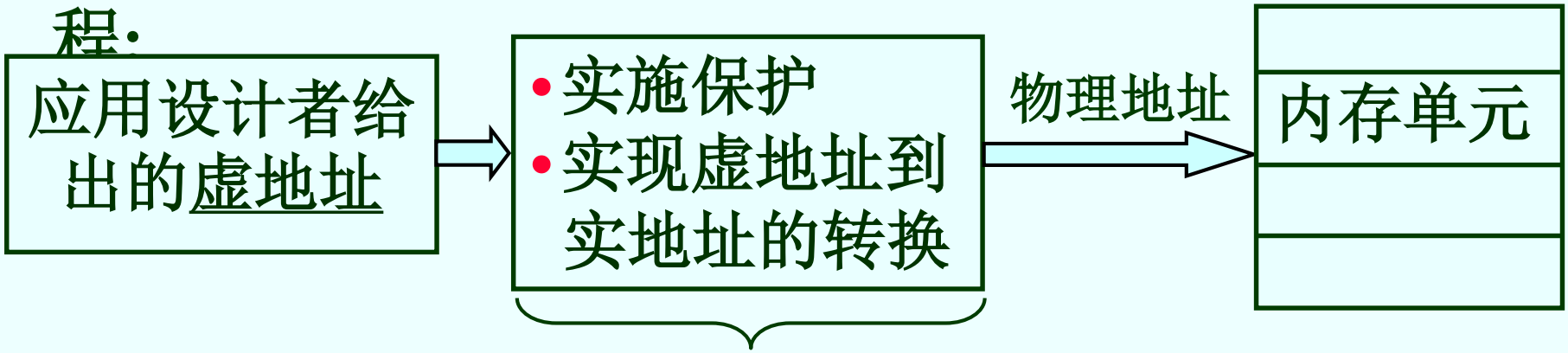
为实现“虚地址”和“保护”两大功能，系统必须提供一种“机制”或“平台”或一个“中间环节”来实施并完成上述两大功能。

实地址模式下的寻址过程：

简化为：



为实施“虚地址保护”所希望的寻址过程：



“虚地址保护”实施的中间平台

“中间平台”的核心部分描述子 (Descriptor)

描述子的作用：

刻划存储段的属性(比如一个段的保护属性), 并提供虚地址到实地址转化的信息

描述子的引入，存储空间就由若干存储段和若干存储段对应的描述子构成，存储器的组织形式就由实地址模式的单一的“存储段”变为两级结构，即：

(1) 一系列可变长的段(1 ~ 64K)

(2) 一系列的描述子

描述子分类(描述子的类型)：

① 按描述子的作用范围：局部描述子和全局描述子

- 局部描述子

  刻划某一个任务所要访问代码段或数据段的描述子，作用于该任务所要访问的范围。这些描述子的组合构成一个描述子表，称为局部描述子表LDT。每个任务都有一个LDT。(这意味着每一个代码段或数据段都对应有一个描述子)

- 全局描述子

  作用范围是系统中所有的代码段和数据段。所有这些描述子的组合构成一个全局描述子表GDT。整个系统只有一个GDT。

② 从描述子的功能来划分

- 数据/代码段描述子

  用于刻划一个存放数据或代码的存储段的各种属性。比如该段的特权级、段限、读写属性、并提供从虚地址到实地址转换的信息。

- 门描述子: 实现不同任务间的转换和同一任务的不同代码段之间的转移。

- 任务状态段描述子

按描述子的定义，在保护模式下访问存储器中数据或代码，则需要使用数据/代码段描述子来实施相应的保护功能。其寻址过程示意图：