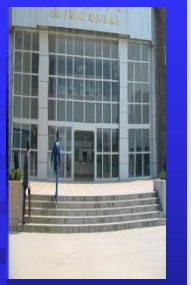
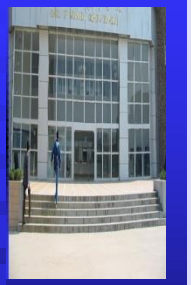


第12章 32位机工作原理

一 寄存器

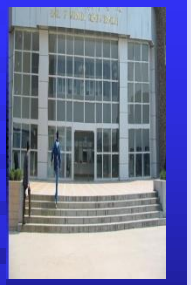
二 保护模式下的内存管理



第12章 32位机工作原理

一 寄存器

二 保护模式下的内存管理



一 寄存器

32位机有三大类寄存器

1）用户级寄存器

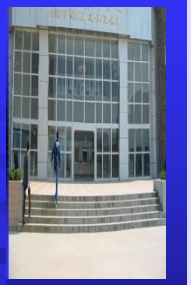
- 设计应用程序时要用的寄存器。

2）系统级寄存器

- 控制寄存器，操作系统使用。
- 支持存储器管理的段表寄存器，系统软件间接使用，是程序不可见寄存器。

3）程序调试寄存器

用户级寄存器



EAX		AH	AX	AL	累加器
EBX		BH	BX	BL	基址寄存器
ECX		CH	CX	CL	计数寄存器
EDX		DH	DX	DL	数据寄存器
ESP			SP		堆栈指针
EBP			BP		基址指针
ESI			SI		源变址寄存器
EDI			DI		目的变址寄存器

(a) 通用寄存器

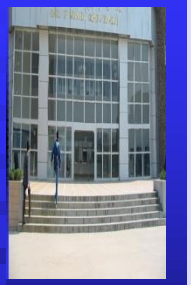
EIP		IP	指令指针
EFLAGS		FLAGS	标志寄存器

(b) 指令指针和标志寄存器

CS	代码段寄存器
DS	数据段寄存器
ES	附加段寄存器
SS	堆栈段寄存器
FS	
GS	

(c) 段寄存器

图 13.4　用户级寄存器



■ 段寄存器

增加了FS、GS

保护模式下，用它们存放段选择子，指向存放在全局描述符表GDT或局部描述符表LDT中的段描述符，来获得段的全部信息。

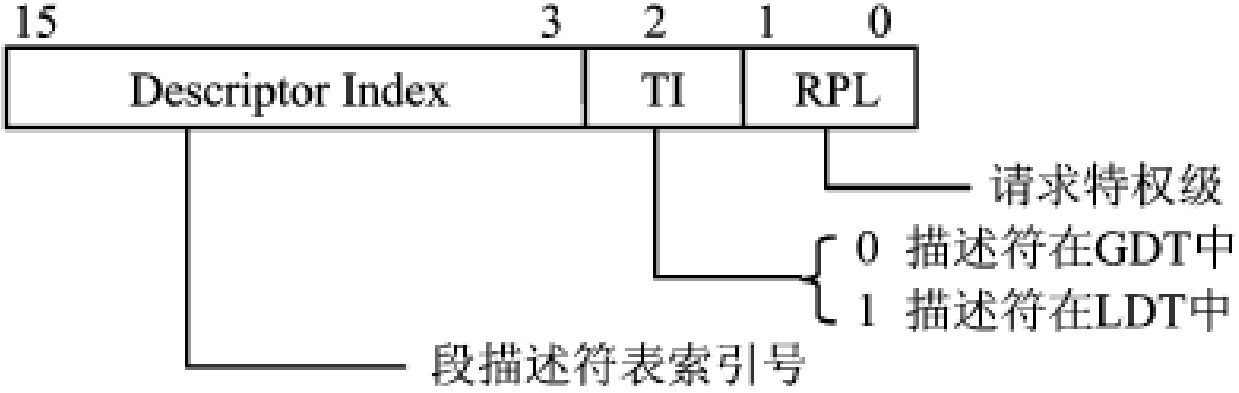
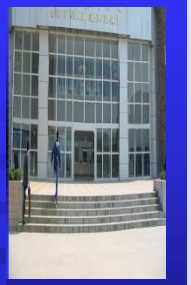


图 13.6 段选择子格式



系统级寄存器

■ 控制寄存器

决定CPU的操作模式和现行执行任务的特征状态，由操作系统使用。CR4是Pentium以上新增的。

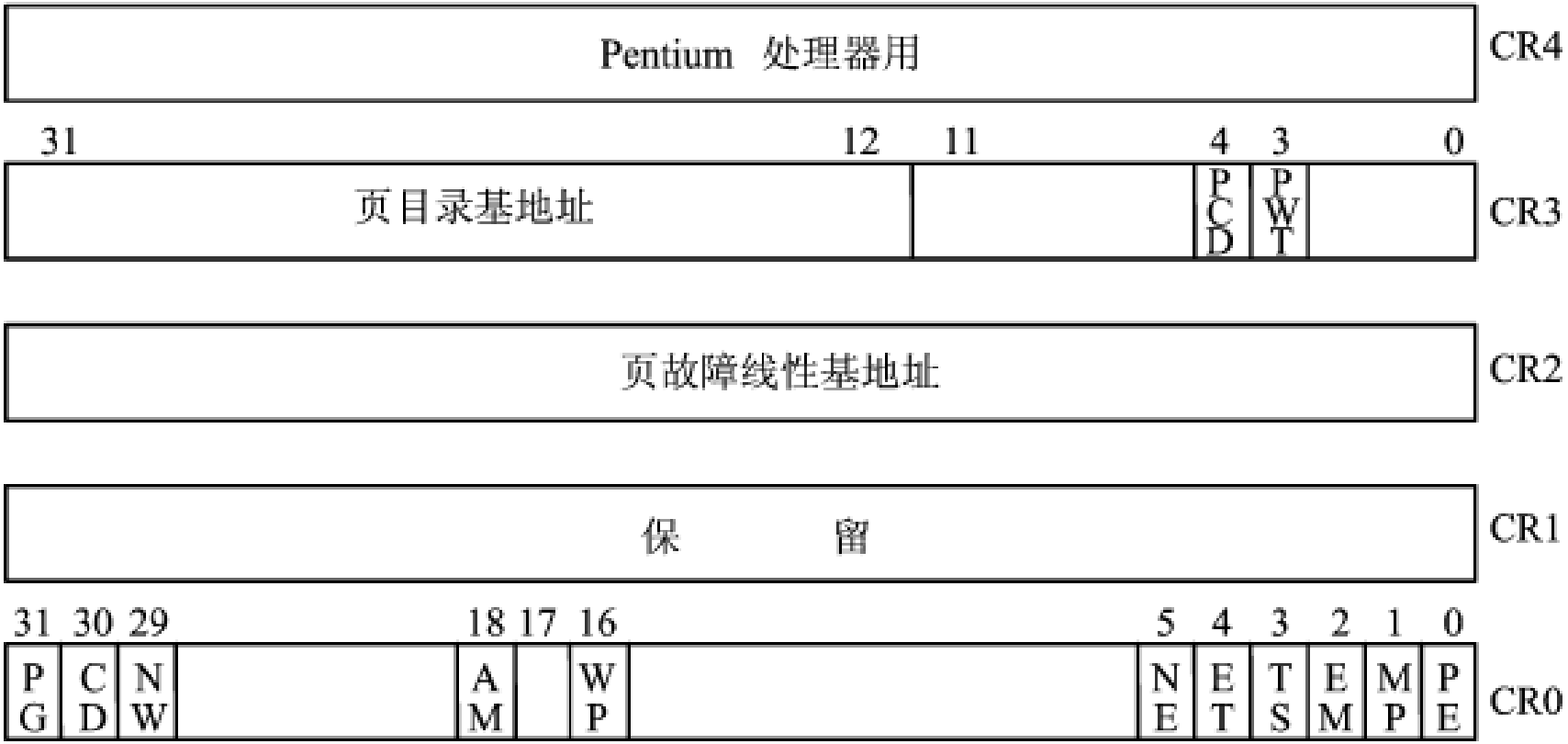
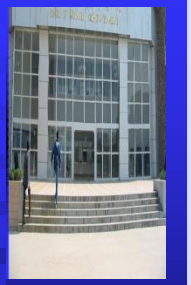


图 13.7 控制寄存器



➢**CR0**

包含控制操作模式的控制标志位，并给出处理器的状态位。控制标志位可用指令设置，其中包括决定工作环境的保护模式允许位**PE**，允许分页位**PG**等。

表 13.1　PE 和 PG 组合提供的工作环境

PG	PE	工作环境
0	0	实地址方式，与 8086 兼容
0	1	不分页保护方式，有分段功能，但无分页功能
1	0	未定义
1	1	分页保护方式，具有分页、分段功能

➢**CR1**

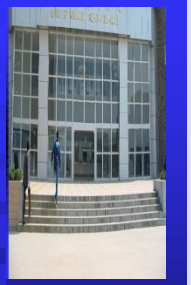
保留。

➢**CR2**

页故障线性基地址寄存器（**Page Fault Linear Address**），存放发生故障中断（异常**14**）之前所访问的最后一个页面的线性地址。

➢**CR3**

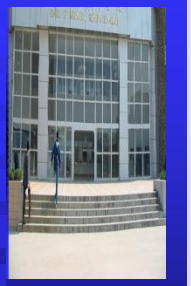
页目录基地址寄存器，存放页目录表的物理基地址。



系统段表寄存器

1）各种描述符表

- 段描述符长8字节，保存段的基地址、长度和属性。
- 描述符中的类型标志位S指定不同的段描述符：
  - S=1，内存段描述符，如代码段、数据段、堆栈段
  - S=0，系统段描述符，门描述符以及TSS（任务状态段）描述符，保存正执行任务的地址、段限长和属性。
- 这些段描述符由编译、连接、OS等生成，存放在三种描述符表中：
  - 全局描述符表（GDT）面向所有任务，只有1个GDT
  - 中断描述符表（IDT） 面向所有任务，只有1个IDT
  - 局部描述符表（LDT）面向某一任务，有多个LDT



2）系统段表寄存器组成

寻找和定义描述符表和TSS段的寄存器，包括

- **系统表寄存器**

  GDTR全局描述符表寄存器，IDTR中断描述符表寄存器

- **系统段寄存器**

  LDTR局部描述符表寄存器，TR任务状态段寄存器

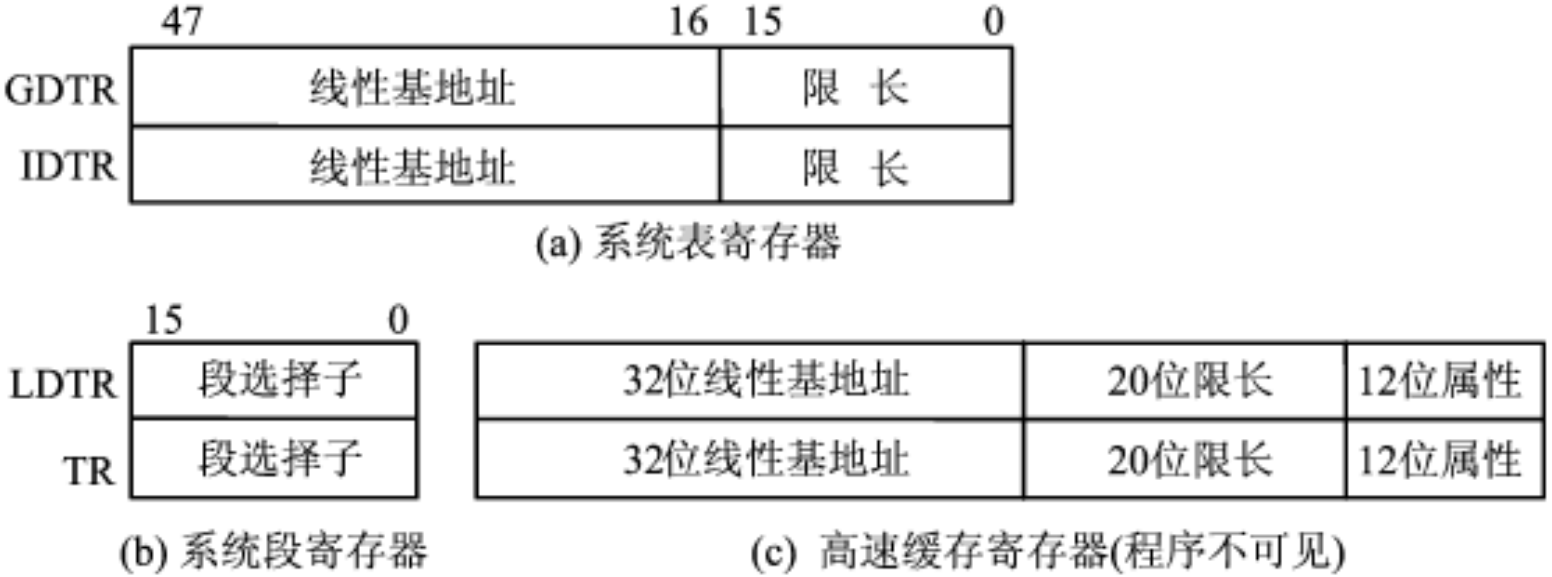
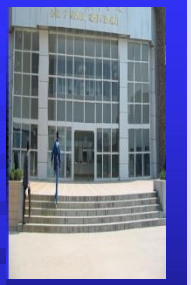
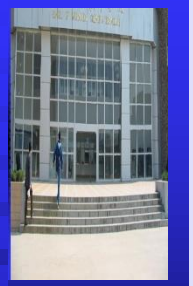


图 13.8 段表寄存器



3）GDTR和IDTR寄存器功能

- **GTDR**
  - 48位, 定义GDT表基址(指向表头)和段限长(表的长度)。
  - GDT表最多可放$2^{13}=8192$个段描述符。
  - 可用“LGDT mem48”指令装载GTDR，指定GDT表的基址和段限长，对它定位。
- **IDTR**
  - 48位, 指向IDT表的基址和限长。IDT表中存放中断或异常的描述符，最多可放256项。
  - 可用“LIDT mem48”指令装载IDTR，指定IDT表的基址和段限长。



a)根据GDTR中的基地址找到GDT表的位置，根据限长确定GDT表的大小。

b)根据IDTR的内容确定IDT表的位置与大小。

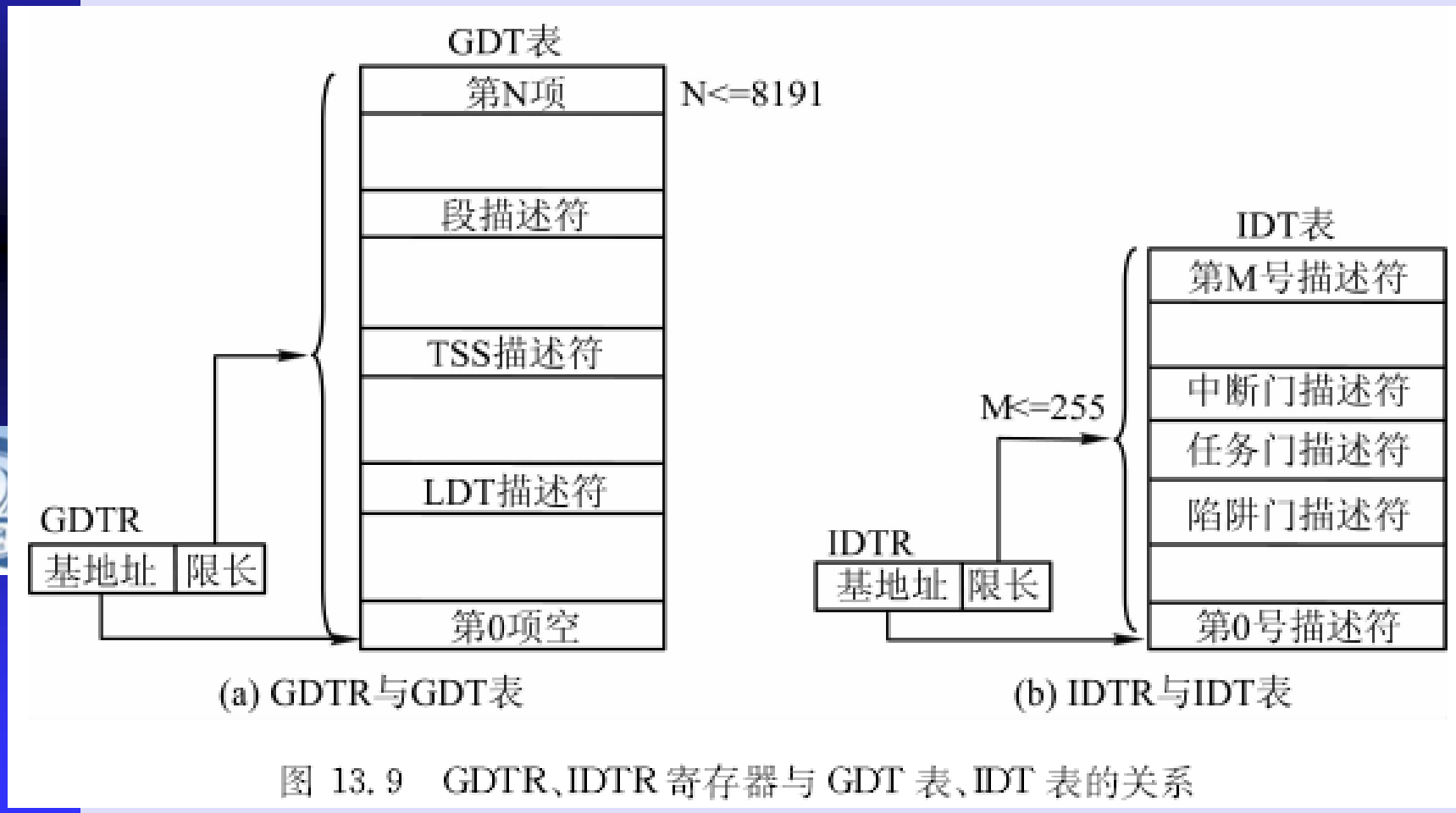
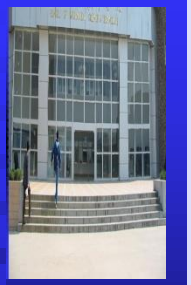


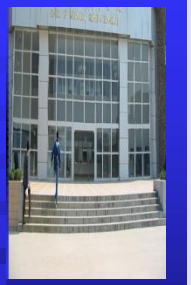
图 13.9 GDTR、IDTR 寄存器与 GDT 表、IDT 表的关系



4）LDTR寄存器的功能

- 每个任务均有1个LDT表，表中存放与该任务相关联的全部段描述符，而LDT表本身所在的段描述符则放在GDT中。
- LDTR用来定位LDT，它由16位选择子+64位高速缓存（程序员不可见）组成。
- 可用LLDT指令，把当前任务的16位LDT段选择子装进LDTR:

  LLDT　reg16/mem16
- 同时，LDT描述符（64位，含32位基址、20位限长及属性）也被装入LDTR的高速缓存。这样，就不需要根据选择子查找GDT表，可直接从高速缓存中获得64位段描述符信息，提高速度。
- 进行任务切换时，处理器会把新任务的LDT段选择子和段描述符自动载入LDTR。



- 执行任务时先要从GDT中找出64位LDT描述符，定位LDT表。可执行LLDT reg16/mem16指令，将LDT描述符的16位段选择子加载进LDTR寄存器，段描述符内容同时被装入64位高速缓存。程序取出其中内容，就可确定该LDT表的始址、长度和属性。

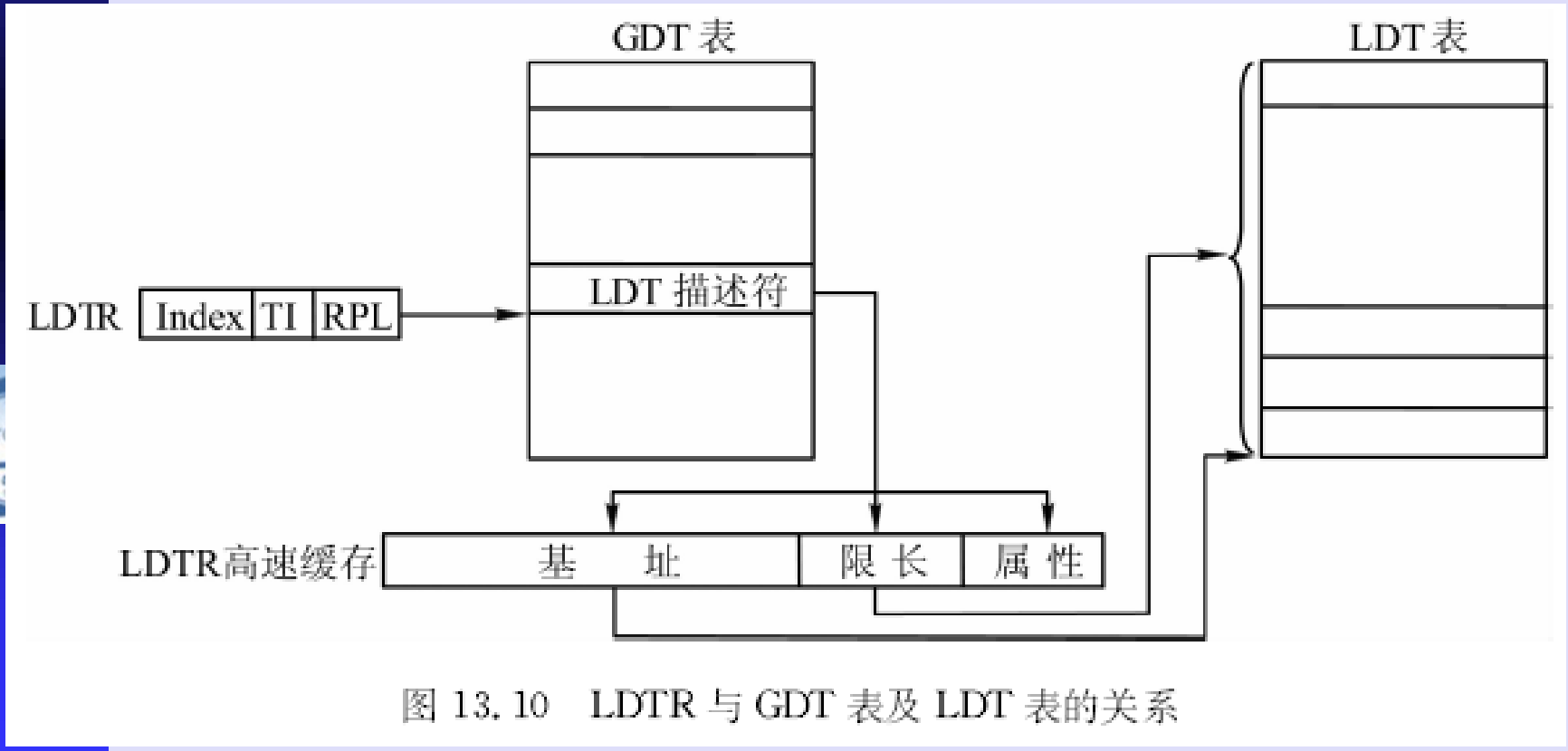
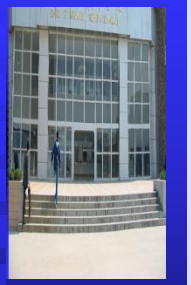


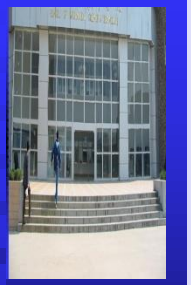
图 13.10 LDTR 与 GDT 表及 LDT 表的关系



5）TR寄存器的功能

- 多任务操作系统中每个任务都有个任务状态段(TSS)，保存任务的环境，如寄存器、内存和I/O地址空间等信息。
- TSS位于内存中，用任务状态段寄存器TR和TSS描述符来定位。
- TR和LDTR那样，由16位选择子和64位高速缓存组成，并用LTR指令装入选择子的值：

```
LTR     reg16/mem16
```



先由TR的16位选择子在GDT表中找到当前任务的TSS描述符，读出TSS段的基址、限长和属性，并加载进64位缓存。以后每次用TR的16位选择子访问内存时，就不再到GDT中读取TSS描述符，提高了任务执行效率。

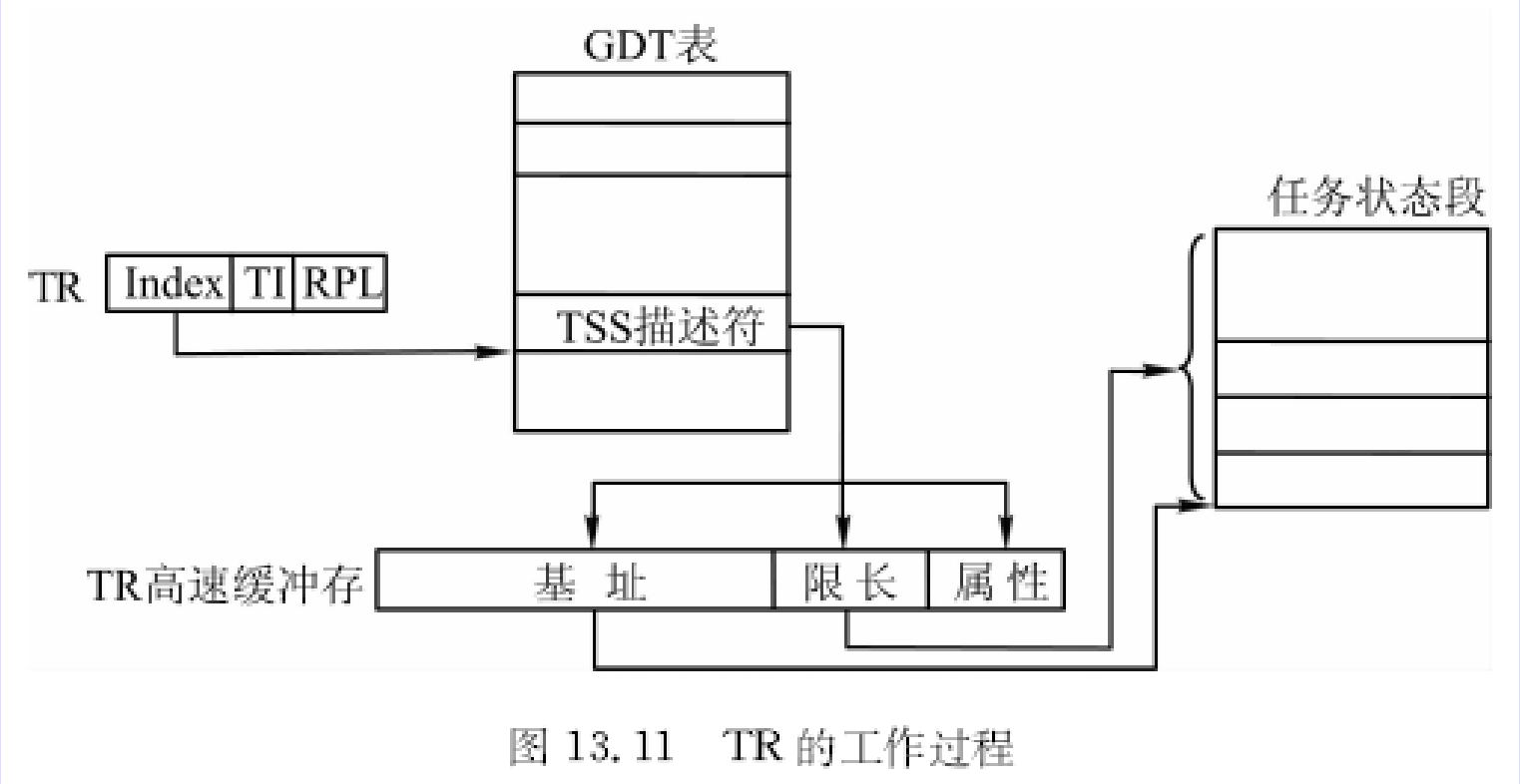
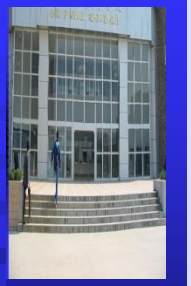


图 13.11　TR 的工作过程



■ 段寄存器和描述符表的关系

如图通过段选择子找到特定描述符，通过GDT表来定位LDT表。

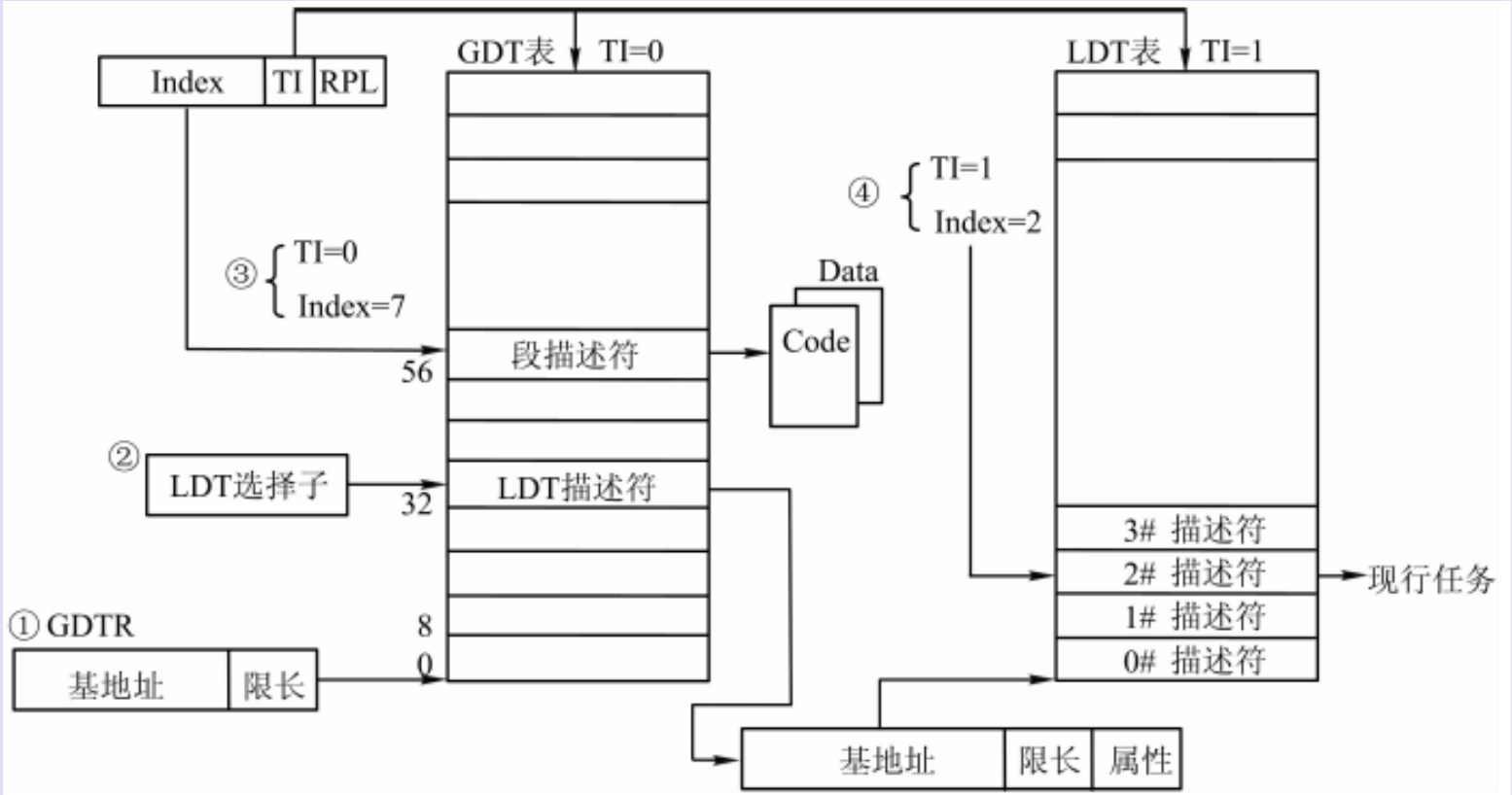
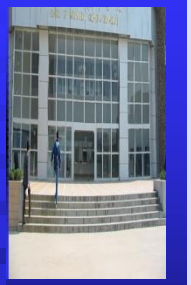


图 13.12　段寄存器与描述符表的关系

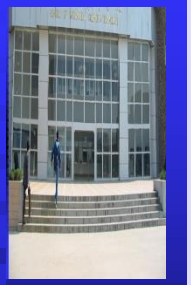


①GDTR寄存器给出GDT表的基地址和限长，定位GDT表。

②任务切换时，任务状态段TSS中有一个16位的LDT段选择子，其TI＝0，Index＝4，选中GDT表中的4号段描述符，它相对于GDT基址的偏移量为4×8=32，再从该单元开始取出LDT描述符，得到基地址、限长和属性信息，来定位LDT表。

③若某个段寄存器(如CS)装入的段选择子的TI＝0，Index＝7，即指向GDT表中的7号描述符(偏移量为7×8= 56)，由该描述符指向一个代码段或数据段。

④若某个段寄存器的TI＝1，Index＝2，则访问LDT表中的2号描述符，由此描述符指向现行任务段。



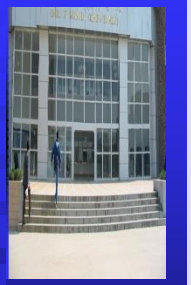
程序调试寄存器

80386前的CPU提供：INT 01H单步中断，INT 03H断点中断这两个调试接口，80386仍支持它们。

80386引入了8个调试寄存器DR7~DR0，支持指令断点和数据断点。

断点0的线性地址	DR0
断点1的线性地址	DR1
断点2的线性地址	DR2
断点3的线性地址	DR3
保　　留	DR4
保　　留	DR5
调试状态寄存器	DR6
调试控制寄存器	DR7

图 13.13　调试寄存器



程序调试寄存器

- **DR3~DR0　断点调试寄存器**

  存放4个断点的线性地址。只需将断点指令的线性地址写入相应寄存器，即能构造指令断点。80386可支持4个断点。

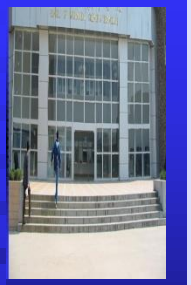
- **DR6　调试状态寄存器**

  当产生调试异常（01H号中断）时，CPU会在DR6中给出异常类型，如单步异常、Debug异常等。

- **DR7　调试控制寄存器**

  规定每个断点寄存器的使能、断点类型（指令/数据断点）及所有断点寄存器的保护。

- **DR4、DR5　保留**



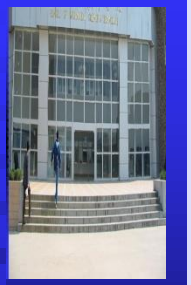
第12章 32位机工作原理

一 寄存器

二 保护模式下的内存管理

保护模式下的内存管理

- 在实模式下，80386的段内存管理与8086相同，每个段长度都是64KB，用段基址+偏移量求物理地址。只能访问1MB内存空间。
- 在保护模式下，采用全新的分段分页技术管理内存，能寻址4GB地址空间。
- 采用可变长的分段技术，每段大小：1～2^{32}字节(4 GB) 分段时还对每段赋予属性和保护信息，进行4级保护
- 采用分页管理技术，与分段技术相结合，使虚拟存储空间大大超过物理地址，可达到64TB
- 多任务系统中有了分页功能，还可明显地提高存取数据的效率，并有效利用内存碎片

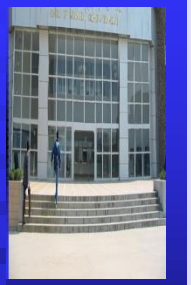


第12章 32位机工作原理

二 保护模式下的内存管理

1 段内存管理技术

2 分页内存管理技术



1 段内存管理技术

1. 逻辑地址、线性地址和物理地址

- 逻辑地址 Logic Address

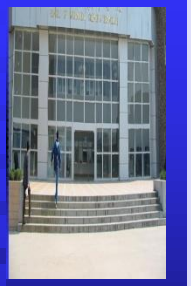
对段内存空间寻址的地址称为逻辑地址，也叫虚拟地址，逻辑地址可表示为“段选择子：偏移地址”

➢**段选择子**：16位，存于CS，DS等段寄存器中，通过它能从GDT或LDT中找到64位段描述符，获得段的全部信息。

➢**偏移地址**：32位，即有效地址EA，计算方法

偏移地址＝基址＋变址×比例因子＋位移量

- 基址：存放于32位通用寄存器中
- 变址：存放于除ESP外的32位通用寄存器中
- 比例因子：1，2，4，8
- 位移量：8位或32位立即数



1 段内存管理技术

1.逻辑地址、线性地址和物理地址

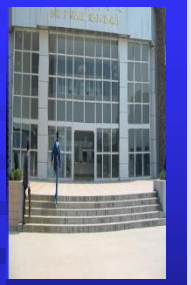
- 物理地址 Physical Address

物理地址是指内存芯片阵列中每个阵列所对应的唯一的地址，32位地址线可直接寻址 2^{32}＝4GB内存单元。

- 线性地址 Linear Address

是沟通逻辑地址与物理地址的桥梁，32位CPU的分段部件将逻辑地址空间转换成32位的线性地址。

方法：段选择子中的Index指向GDT或LDT中的一个段描述符，从中可读出线性基地址，加到32位偏移地址上，即形成线性地址。



1 段内存管理技术

1. 逻辑地址、线性地址和物理地址

● 地址转换

分段部件先将逻辑地址➔线性地址。若分页功能禁止，则线性地址就是物理地址；如允许分页，则分页部件再将线性地址转换成物理地址。

TI：0：全局描述符表　1：局部描述符表

RPL：申请特权级 0~3

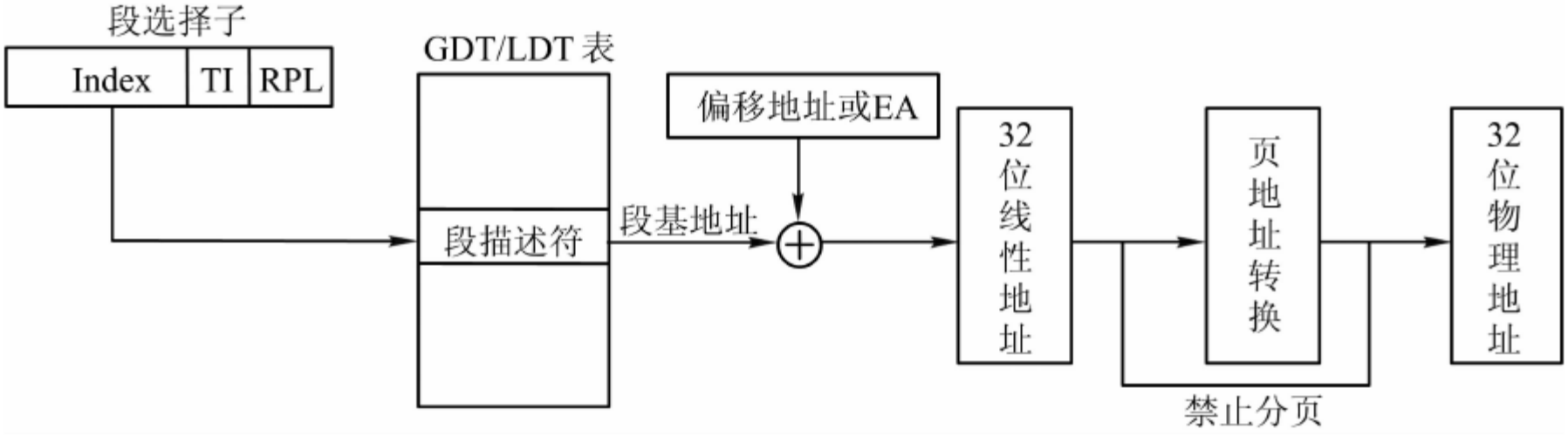
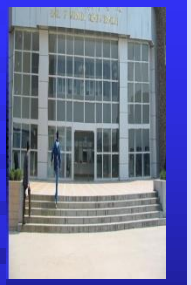


图 13.14　逻辑地址、线性地址和物理地址转换框图



2. 段描述符

共有三类段描述符，用段描述符中的b12位S来区分：

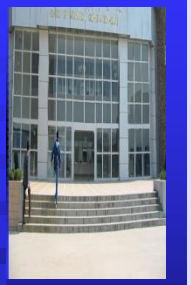
S=1，内存段描述符，对应段为代码段、数据段或堆栈段。

S=0，系统段描述符和门描述符。

- **内存段描述符**

内存段描述符都是8字节，分为：段基地址(Base Address)，段限长(Limit)和段属性(Attribute)等 3部分。

图 13.17　段描述符的格式



1 段内存管理技术

1）段基地址

- 基址是32位线性地址A_{31}~A_0，指出段的起始位置，可以是32位线性地址(4GB)中的任一个地址。

2）段限长

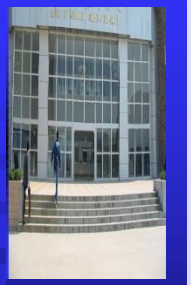
- 限长为20位，决定段的可寻址范围
- 要与b23位G结合计算实际限长：

  G=0，限长高12位为全0；

  G=1，段限长＝描述符中的20位限长×1000H＋FFFH

3）段内存属性

- 除G、S外，尚有多种属性，尤其是4位的type属性是程序员必须熟悉的。



●系统段描述符和门描述符

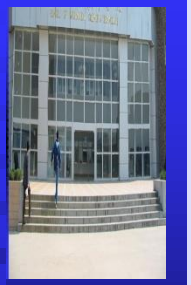
系统段描述符描述有关80386操作系统的表和任务等信息。任务状态段TSS和局部描述符表LDT都看作系统段。

1）任务状态段TSS

- 每个任务都有TSS，用在任务切换时保存任务的环境。
- 低位部分由CPU定义，对应一个任务的各种信息，存放各种寄存器值，占用104（0~67H）字节。
- 高位部分由OS创建任务时定义。
- CPU通过任务状态寄存器TR和TSS描述符来定位内存中的TSS段。8字节的TSS描述符存放在GDT中，定义了任务状态段在内存中的基址、限长和类型。格式

基址15~0					限长15~0					
基址31~24	G	0 0	AVL	限长19~16	P	DPL	0	TYPE	基址23~16	+4

图 13.20　TSS 描述符的格式



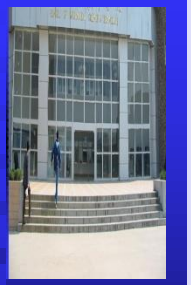
2）门(gate)

- 是一种转换机构。当程序的控制由一个代码段（源代码段）转到另一个目标代码段时，通过门来实现。门设置在目标代码段入口处，控制对该目标代码段访问的权限。
- 门描述符为8字节，格式：

段选择子15~0					偏移量15~0	0
偏移量 31~16	P	DPL	0	TYPE	参数计数器	+4

图 13.21　门描述符格式

- **4、5字节是门描述符属性，其中TYPE定义门的类型和长度（16/32位）。**
- 调用门，段选择子和偏移量指向要调用子程序的目标代码起始地址；任务门只有段选择子，指向GDT中的一个TSS描述符。中断门和陷阱门，选择子和偏移量构成中断处理子程序或陷阱处理程序的入口地址。

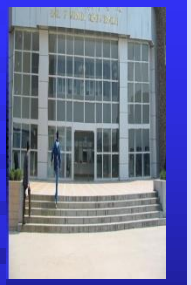


第12章 32位机工作原理

二 保护模式下的内存管理

1 段内存管理技术

2 分页内存管理技术



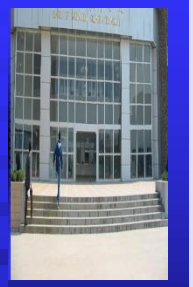
2 分页内存管理技术

分页内存管理以Page(页)为单位，将内存空间映射到磁盘空间，便于实现虚拟存储器管理，还能提高存取效率，有效利用内存碎片。是80386对8086的主要功能扩充。

80386以4KB为1页，4GB内存分成2^{20}个页，在被4KB整除的地址（后3位=000H）处分页。

分段技术将逻辑地址➔线性地址

- CR0的PE=0禁止分页时，线性地址即物理地址；
- CR0的PE=1允许分页时，分页部件把将线性地址➔物理地址。



1. 页目录与页表

●采用页目录和页表两层表实现分页管理

页表项共有2^{20}=1024×1024个，每项4字节。如果只是一级页表，要占4MB内存。采用两级页表，每级10位，2张表只需占(1024项×4字节/项)×2＝1024×8＝8KB内存空间。

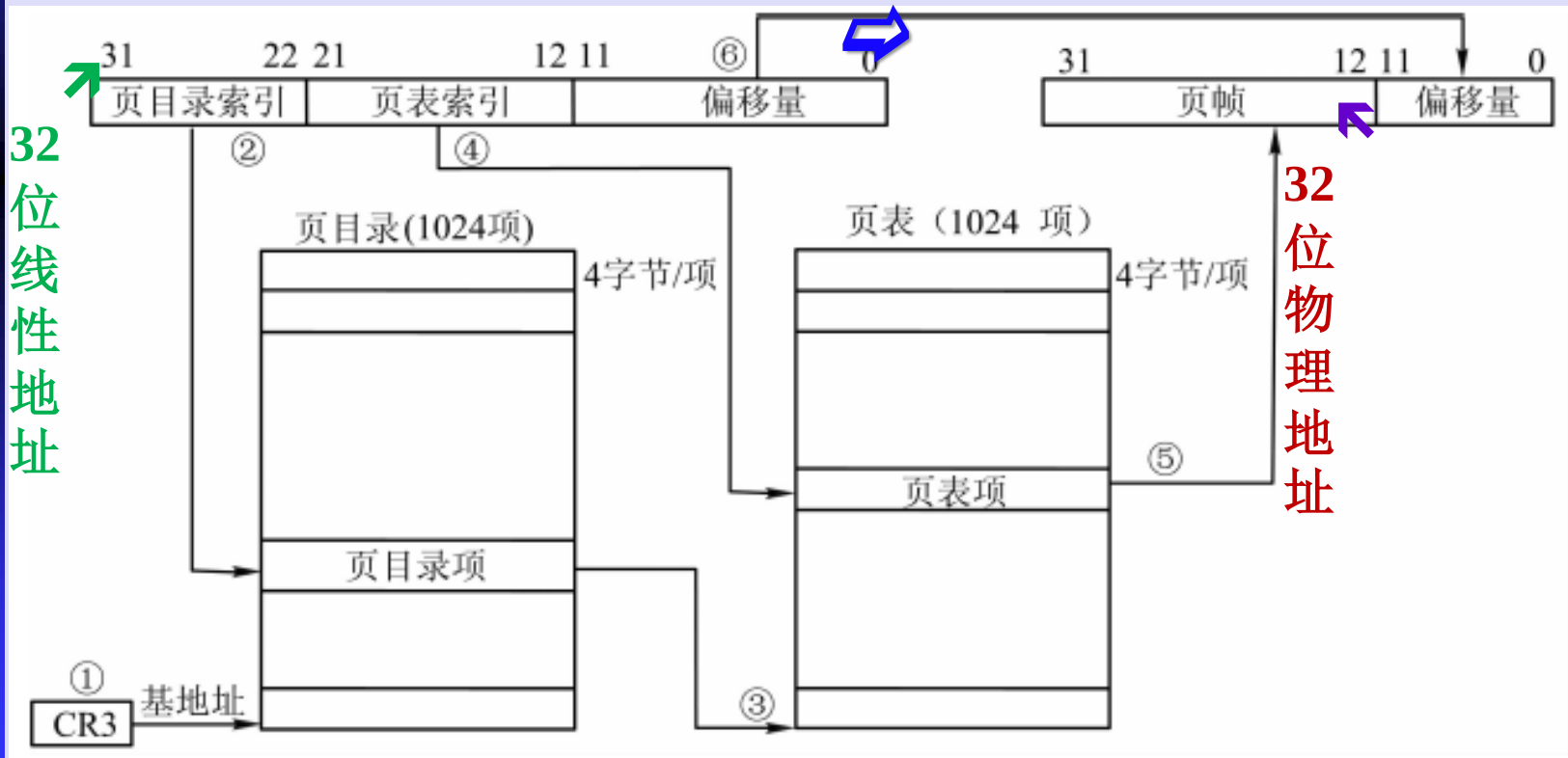
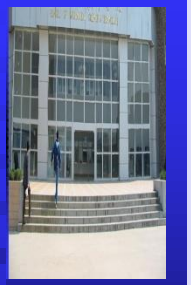


图 13.22　页目录和页表结构

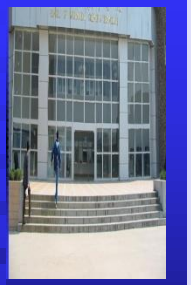


2 分页内存管理技术

1. 页目录与页表

- 采用页目录和页表两层表实现分页管理
- 页目录表含2^{10}＝1024个页目录项，每项4字节。
- 页表含1024个页表项，每项4字节对应4KB(1页)内存。
- 每个页目录项最多可以对应1024个页表项。
- 故整个页目录最多可映射物理地址空间为

  1024页目录项×1024页表项/页目录项×4KB/页表项＝4GB



2 分页内存管理技术

1. 页目录与页表

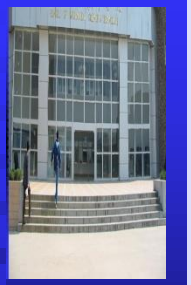
- 采用页目录和页表两层表实现分页管理
- 分页机制将32位线性地址分成3部分：

1)高10位(b31～22)作页目录索引，指向2^{10}=1024个页目录项中的1项。页目录项每项长32位，高20位是页表基址，低12位为其属性。

2)中间10位(b21～12)作页表索引，指向1024个页表项中的1项。页表项每项长度也是32位，高20位对应物理地址的高20位，也称页帧(Page Frame)，低12位为其属性

3)低12位(b11～0)作页面偏移地址，即物理地址的低12位

- 利用两层表和CR3寄存器，就可从32位线性地址求得32位物理地址



2 分页内存管理技术

2. 页目录项和页表项格式

即2张表里的内容，都是4字节，格式基本相同：

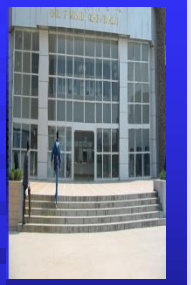
- 高20位：页帧地址

  1页物理地址的起始基地址，也就是物理地址的高20位

- 低12位：属性

31 … 12	11 9	8 7	6	5	4	3	2	1	0
页帧地址	AVL	0 0	D	A	PCD	PWT	U/S	R/W	P

图 13.23　80486 和 Pentium 的页表项格式



3. 举例说明线性地址转换成物理地址的过程

设系统的线性地址为1234 5678H，CR3＝0000 8000H，求对应的物理地址。

如图，将32位线地址1234 5678H分成3部分：

最高10位00 0100 1000B＝048H，作页目录索引号；

中间10位11 0100 0101B＝345H，作页表索引；

最后12位678H，直接作物理地址的低12位。

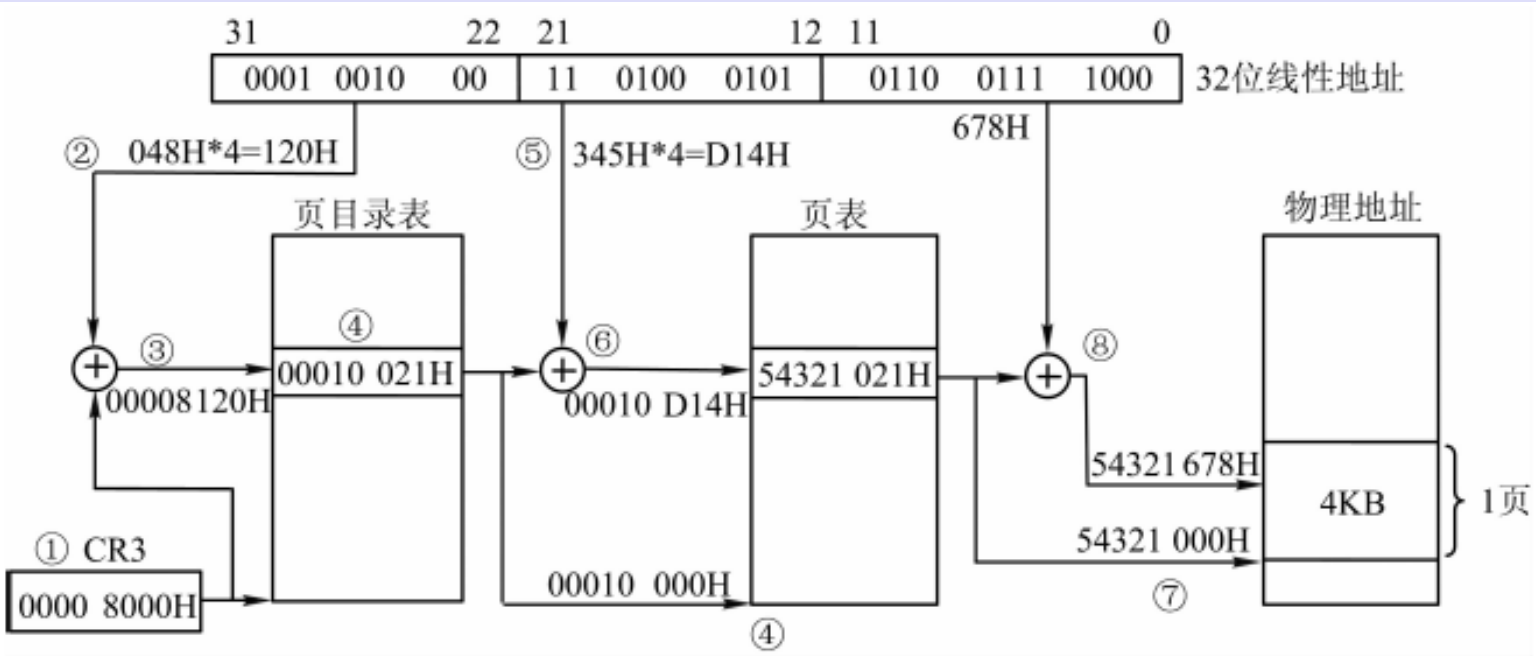
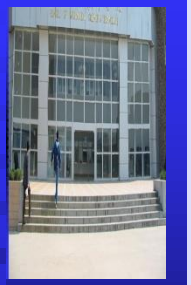


图 13.24 线性地址转换成物理地址的实例



①查询CR3，得CR3＝0000 8000H，作页目录表物理基址

②取线性地址高10位048H，作为页目录索引号。每个目录项4字节，要将索引号乘以4，即048H×4＝120H，才是页目录项的偏移地址

③求页目录项始址的物理地址。其值＝CR3中的基址＋页目录项的偏址＝0000 8000H＋120H＝0000 8120H

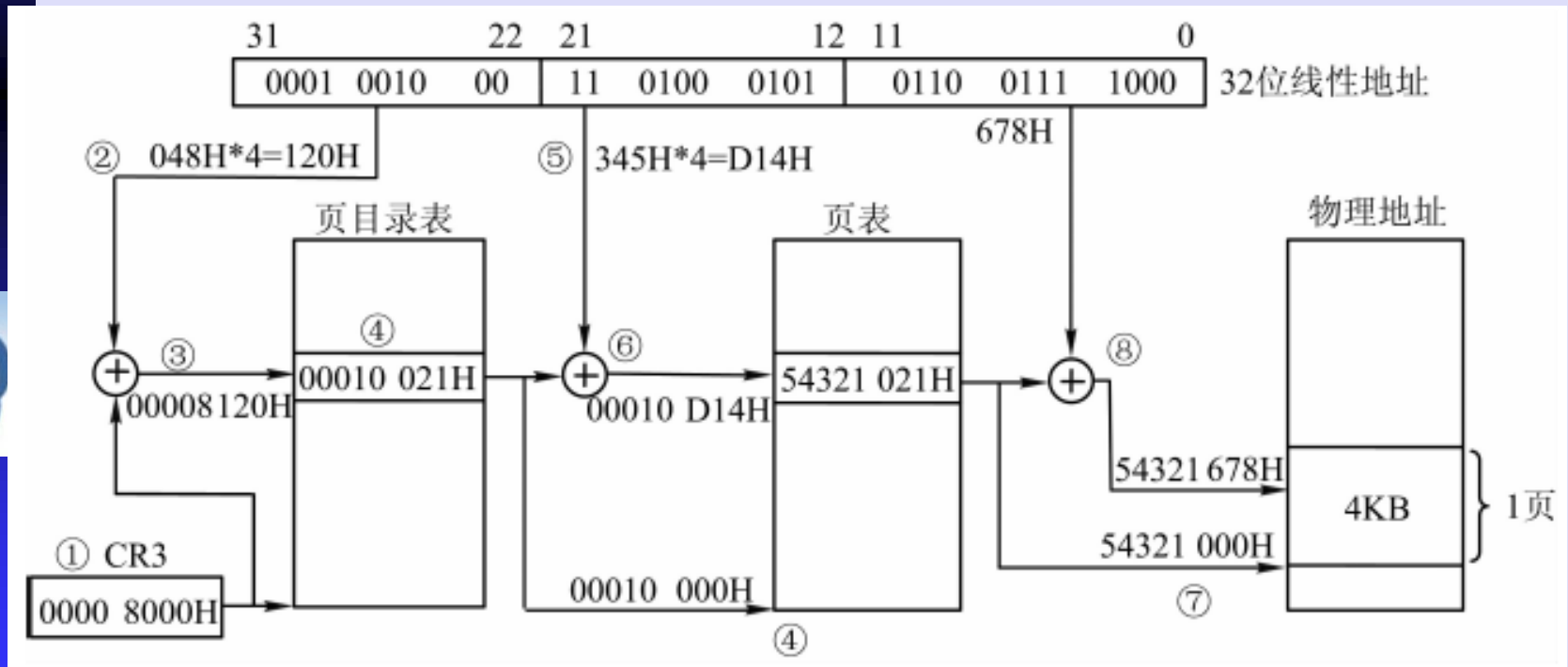


图 13.24　线性地址转换成物理地址的实例

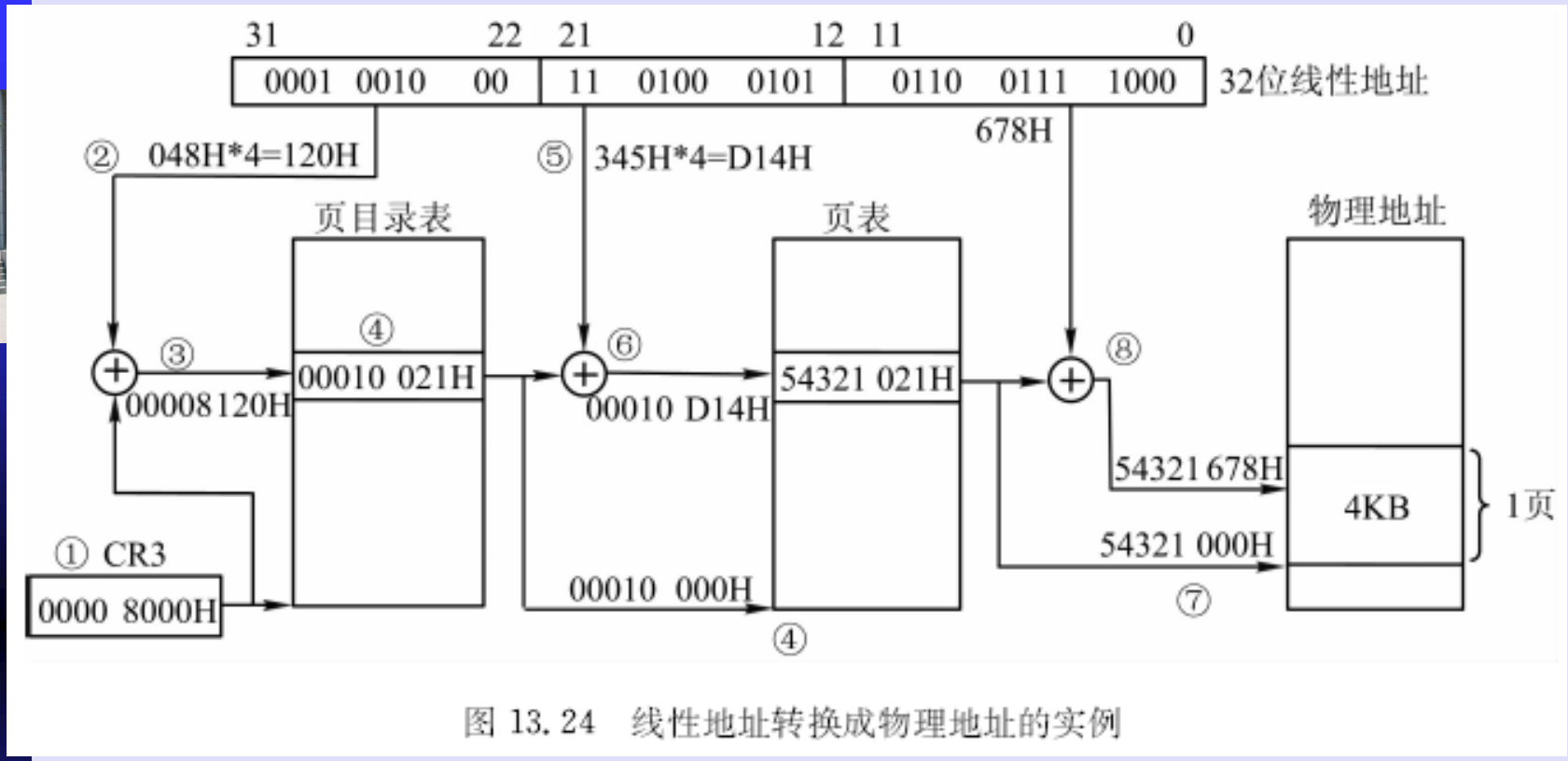
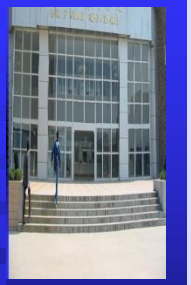


图 13.24　线性地址转换成物理地址的实例

④查页目录项的内容。设（08120H）＝0001 0021H，其中高20位为00010H，它是页表基址的高20位。低12位为属性=021H

⑤线性地址的中间10位=345H，是页表索引号。同样，页表项长4字节，该页表项的偏址为345H×4＝0D14H

⑥该页表项的物理地址=基地址＋偏移量＝0001 0000H＋0D14H＝0001 0D14H。



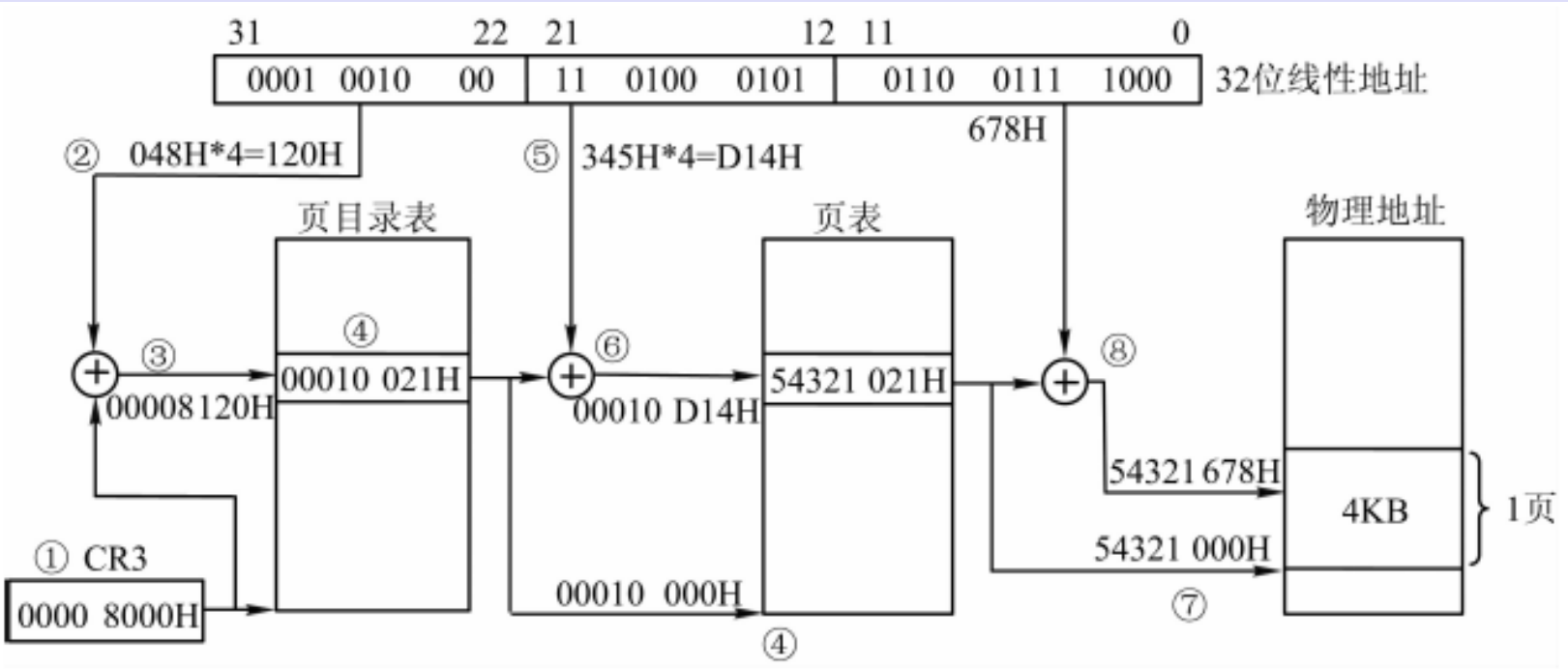
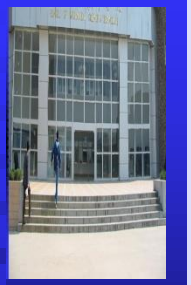


图 13.24　线性地址转换成物理地址的实例

⑦查页表项得(010D14H)＝54321 021H。其高20位54321H为页帧地址，即该页的基址为5432 1000H。后12位为属性。

⑧求物理地址＝页帧＋线性地址低12位＝5432 1000H＋678H＝5432 1678H。至此，从线性地址求得了物理地址

在保护模式下，对于Pentium Pro及以上的处理器，通过物理地址扩展后可以访问36位即64GB的物理地址空间，每页大小可以是4KB、2MB或4MB。



完

谢谢大家！