

2.2 32位微处理器

- 80386微处理器
- 80486微处理器
- Pentium微处理器
- Pentium的存储组织

1 80386微处理器

- 80386主要特点

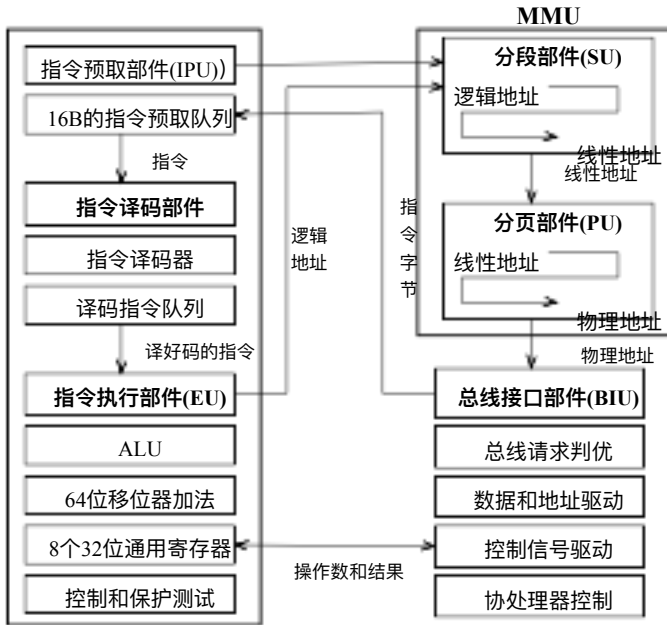
- 先进的CHMOS工艺，芯片内集成275000个晶体管
- 时钟频率12MHZ/16MHZ工作，指令执行速度比286快3倍
- 32条地址线，实模式下寻址4GB物理存储器空间，保护模式下可寻址64GB的虚拟存储器空间
- 高速缓冲器结构，提高了指令的执行速度及工作效率

80386内部结构

总线接口部件BIU

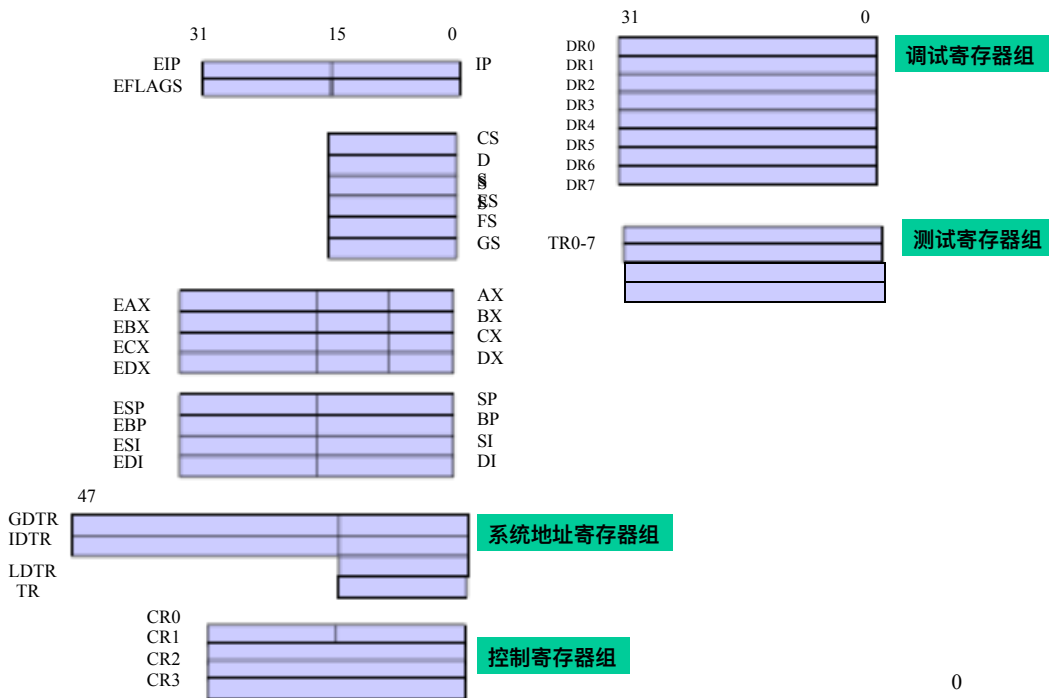
中央处理部件CPU

存储器管理部件MMU



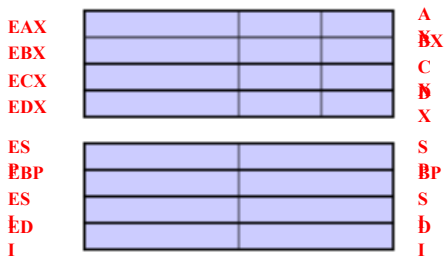
CPU

80386内部寄存器组



- 通用寄存器组

8个32位通用寄存器



可工作在8位寄存器的是：AH, AL, BH, BL, CH, CL, DH, DL

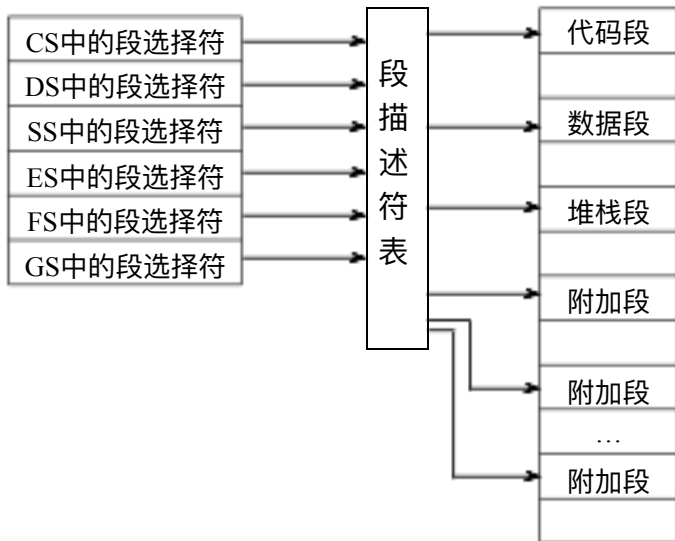
可工作在16位寄存器的是：AX, BX, CX, DX, SP, BP, SI, DI

可工作在32位寄存器的是：EAX, EBX, ECX, EDX, ESP, EBP, ESI, EDI

- 段寄存器组

6个段寄存器（16位）：**CS DS SS ES FS GS**

段寄存器中存储段选择符——由段选择符找到段描述符，段描述符中含有段基址和一些其他信息



- 段描述符

80386采用**段描述符**来描述存储器段的位置、大小和使用情况，每个描述符长8个字节。

描述符用于表示存放在内存中的程序或数据的属性。

段基地址B0—B15					段界限值L15—L0						
段基地址 B31—B24	G	D	0	0	段界限值 L19—L16	P	DP L	S	段类 型	A	段基地址 B23—B16

A：1 表示访问过此描述符；0表示未访问过此描述符

S：1 表示此描述符为数据段或程序段描述符；0 表示为系统描述符

P：1表示此描述符所指的段已装入内存；0 表示未装入内存

D：表示默认操作数的大小（仅在代码段描述符中识别，1表示32位；0表示16位）

G：粒度标志，0 表示段长度为字节粒度；1表示的段长度为页粒度

DPL：段的特权级别，0为最高级别

段类型

段类型			段的特性		
E	C	R			
1	0	0	代码段可 执行不可 写	按特权规则	不可读
		1			可读
	1	0		例外的可独立访问特权	不可读
		1			可读
	ED	W			
0	0	0	数据段不 可执行， 可读	向地址增大的方向扩展，此 时，偏移量 \leq 段界值	不可写
		1			可写
	1	0		向地址减小的方向扩展，此 时，偏移量 $>$ 段界值	不可写
		1			可写

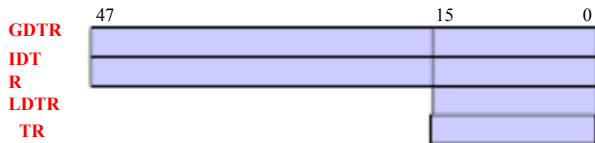
- **段描述符表**

段描述符存储在描述符表中，描述符表包括全局描述符表GDT、局部描述符表LDT、中断描述符表IDT、任务状态段TSS。

- **全局描述符表**：系统中公用段都可以访问的段描述符，通常包含操作系统使用的代码段，数据段和堆栈段，系统中所有的LDT表的描述符等等。
- **局部描述符表**：用于每一个任务，包含了此任务中所使用的代码段，数据段和堆栈段描述符。
- **中断描述符表**：存放中断描述符。

描述符表的起始地址来自系统地址寄存器**GDTR/LDTR/IDTR**。

- 系统地址寄存器



GDTR: 存放全局描述符表的32位线性基地址和16位界限地址

IDTR: 存放中断描述符表的32位线性基地址和16位界限地址

LDTR: 存放局部描述符表的16位段选择符

TR: 存放任务状态段表的16位段选择符

最大、小、中、大

● 标志寄存器EFLAG

肯定会考试

8086

V
M

RF

NT

IO
PL

O
F

DF

IF

TF

SF

ZF

AF

PE

CF

0: 所有调试故障排除
1: 所有调试故障忽略

方向
标志

零标志

奇偶
标志

进位
标志

符号位标志

辅助进位标志

溢出标志

单步标志

中断允许标志

IO特权标志 (2位): 限制I / O指令的使用特权级, 00 01 10 11

0: 任务无嵌套

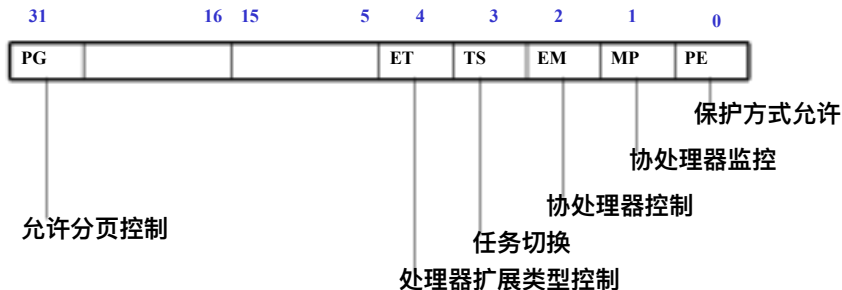
1: 当前任务嵌套于另一任务中

0: 工作于实地址方式

1: 工作于保护虚地址方式

● 控制寄存器组

CR0寄存器



PE = 1/0 系统进入保护/实地址模式。

MP 若TS=1 且MP=1 则CPU在执行WAIT指令时，会产生一个协处理器无效信号。

EM = 1 则会使所有机器指令都产生一个协处理器无效信号，只有EM=0，才会使协处理器指令在协处理器上运行。

ET=0 使用80287

=1 使用80387

PG = 1 启动80386片内分页部件工作

PG	PE	工作环境
0	0	实地址方式，和8086兼容
0	1	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。带有存储体分段和分页功能，即真正的保护方式。

- CR2和CR3寄存器

CR2	页面故障线性地址					
CR3	页组目录表地址	保留				保留

CR2: 保存产生缺页中断之前最后访问页面的32位线性地址

OS, 根据CR2中的数据来确定线形地址空间中的哪一页引起本次异常。

CR3: 高20位保存页目录基地址的高20位, 物理地址向CR3装入一个新值的时候, 低 12位必为0, 从CR3中取值的时候, 低12位被忽视。

● 调试寄存器

DR 0	断点0的32位线性地址																																
DR 1	断点1的32位线性地址																																
DR 2	断点2的32位线性地址																																
DR 3	断点3的32位线性地址																																
DR 4	Intel保留																																
DR 5	Intel保留																																
DR 6																	BT	BS	BD	0	0	0	0	0	0	0	0	0	0	B3	B2	B1	B0
DR 7	LEN3	RW3	LEN2	RW2	LEN1	RW1	LEN0	RW0	0	0	GD	0	0	GE	LE	G3	L3	G2	L2	G1	L1	G0	L0										

DR6是调试状态寄存器，存放上次异常中断时异常的状态，进入调试状态，指令试图读/写调试寄存器，进入单步调试状态发生任务切换。

DR7为调试控制寄存器，可以控制断点的操作，包括断点允许和禁止位，设置断点的条件等。

- 测试寄存器组

TR0~TR5: 保留寄存器

TR6: 测试寄存器, 测试转换后备缓冲区TLB

TR7: 测试寄存器, 保存测试TLB后的结果

更复杂

P305

2 80486微处理器

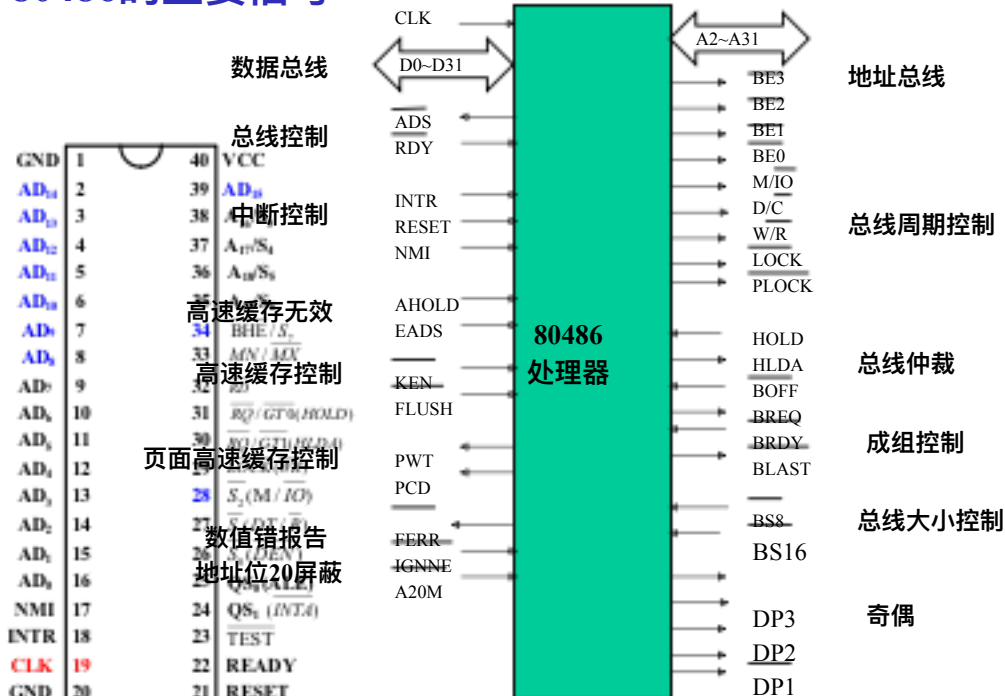
- 80486内部结构

- 沿袭80386的体系结构
- 内含高速缓存和浮点处理器
- 面向多处理器结构

- 80486内部寄存器组 与 80386 区别

- CR3新定义：PCD（cache使能）和PWT（通写/回写）
- CR0增加WP页面保护，AM配合页面检查
- EFLAG增加AC，对界检查
- 增加TR3-5，测试片上高速缓存

● 80486的主要信号



几组重要信号：

地址相关信号：

- **A31~A2**：32位地址总线，三态、双向，对外输出内存地址，对内输入cache地址，用于定义存储器和I/O端口地址
- **A20M**：第20位地址屏蔽信号，输入，低电平有效，有效时，将屏蔽A20及以上地址，使32位微处理器仿真8086CPU的1MB存储器地址。只有CPU工作在实模式下才有意义
- **BE0~BE3**：标识当前传送操作中涉及32bit中的哪些字节被驱动有效
- **ADS**：地址状态输出，总线周期已启动
- **NA**：下一个地址有效，总线是否按流水线工作

数据相关:

引Pmp

- **D31~D0** : 32位双向数据总线, 可以传输8位、16位、32位数据
- **PCHK、DP0~DP3** : 奇偶校验
- **BRDY、BLAST** : 成组准备就绪, 成组输出信号
- **FERR、IGNNE** : 浮点数据出错报告
- **BS8、BS16** : 控制数据总线的宽度

Cache相关:

- **KEN、FLUSH** : cache允许、强制清洗
- **AHOLD、EADS** : cache无效性控制
- **PWT、PCD** : 页面回写/通写、cache禁止

总线周期定义相关：

- **W/R**：读/写信号、存储器或I/O访问信号、数据/控制信号、总线锁定信号与80486微处理器中这些信号功能相同
- **M/IO**：数据传输对象
- **D/C**：数据传输方向
- **LOCK**：总线封锁
- **PLOCK**：伪锁定输出
- **ADS、RDY**：总线周期的状态

控制信号：

- **RESET**：复位信号，输入，高电平有效。当RESET有效时，内部cache全部无效，数据cache中的修改行不再回写
- **中断请求**：可屏蔽中断请求**INTR**信号、非屏蔽中断请求**NMI**信号
- **总线仲裁**：Pentium微处理器中的总线请求信号**HOLD**、总线请求响应信号**HLDA**、内部总线请求信号**BREQ**、强制CPU放弃系统总线信号**BOFF**

P43 几段话必须！
填空/选择/大题

Pentium的总线操作

T1 地址和状态信号有效

T2 数据出现在数据总线上

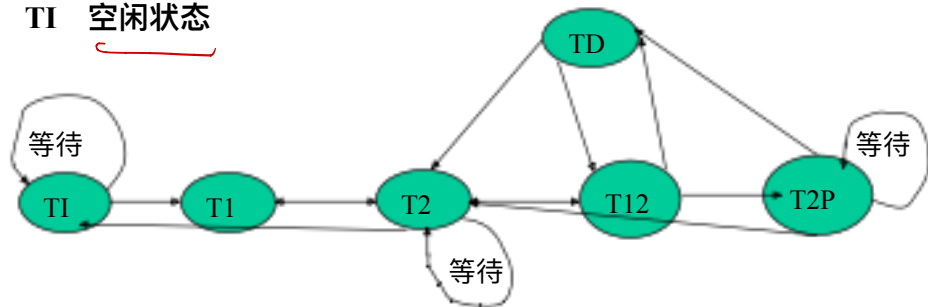
T12 流水线式特有状态

T2p 流水线式特有状态

TD 过渡状态

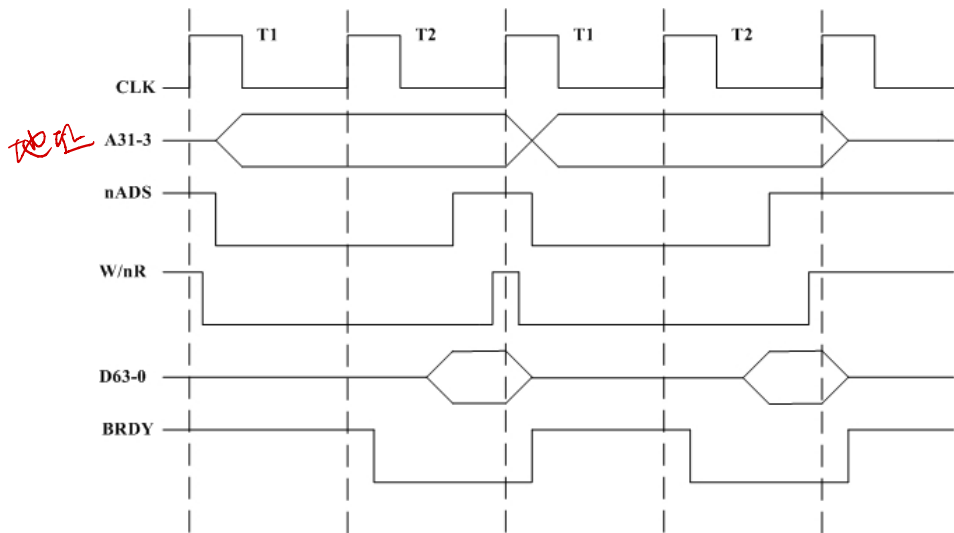
TI 空闲状态

状态变化图

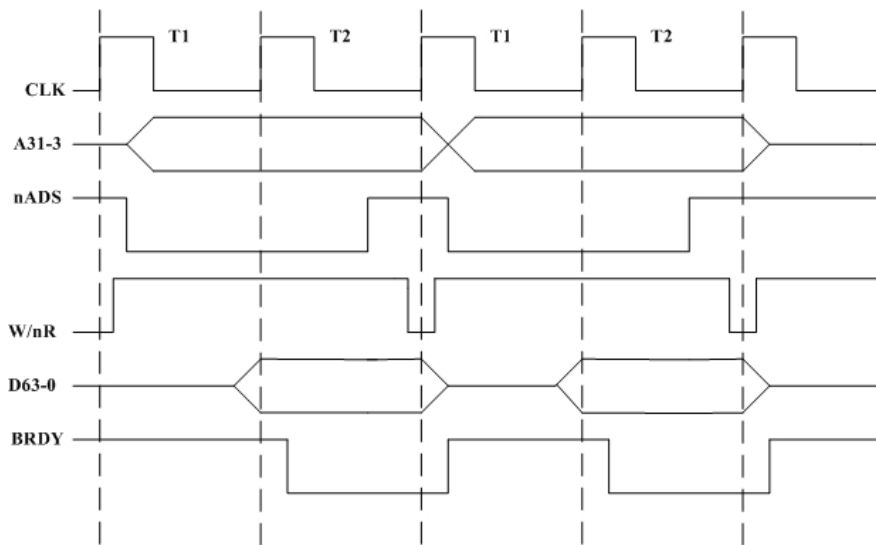


总线周期时序

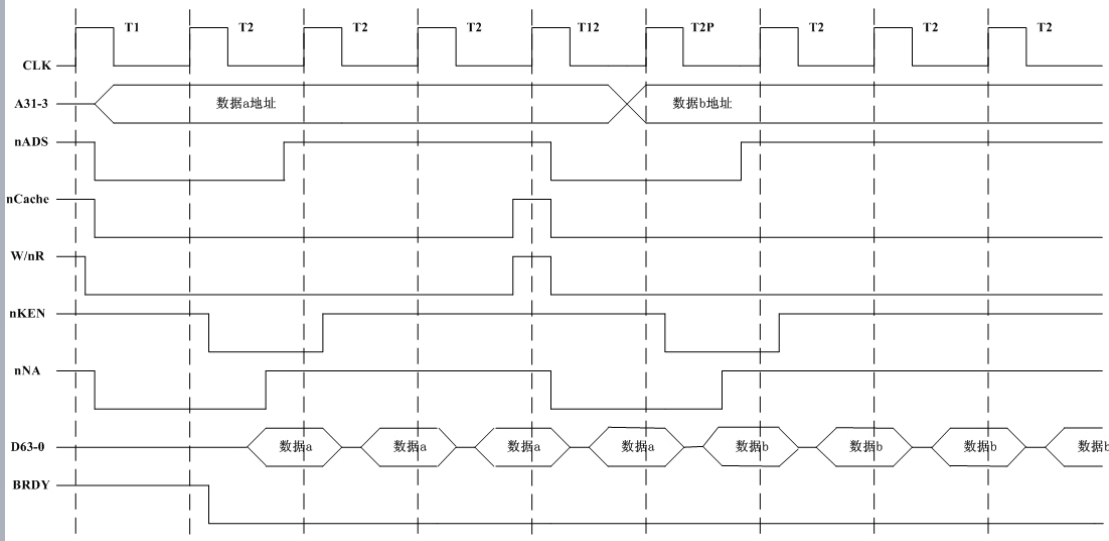
非流水线式读周期



非流水线式写周期

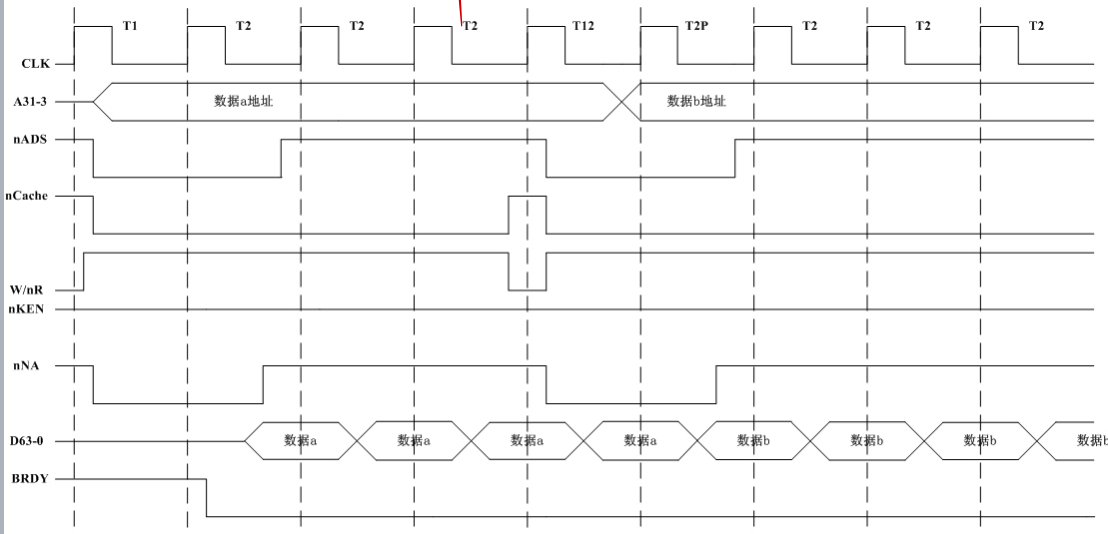


流水线式读周期



流水线式写周期

3个



● 80486的工作模式

1) 实地址模式

243

实地址方式与8086工作方式基本相同。

- 实地址方式下，物理地址的形成机制与8086相同，所有段的长度最大为64KB，并且均执行读、写操作。
- 实地址方式下，保留了两个固定的存储区域：系统初始化区和中断向量区。中断向量区地址和中断向量的方式与8086相同。系统初始化区地址范围为FFFFF0H~FFFFFFFH，大小为16B。
- 复位后，首先从初始化区的FFFFF0H处取指令执行远程跳转，然后A31~A20地址线将变为低电平，从系统ROM内取指令，执行初始化程序并启动系统。

2) 保护模式

- 支持多任务操作；
- 存储器采用虚拟/线性/物理地址空间方式描述，虚拟地址空间 64TB，具有存储保护(空间和操作)功能；
- 可使用分段和分页技术管理存储空间；
- 采用分级(4级)管理，用户程序一般在3级。

在实地址方式下，相当于一个快速的8086，并未充分发挥其自身的许多高性能作用。要利用CPU486所有功能，必须工作于保护方式。

3) 虚拟8086模式

虚拟8086方式是一种既能利用保护方式功能，又能执行8086代码的工作方式。它允许同时运行多个8086任务、任务和任务之间彼此不会相互干扰。微处理器的分页机构还可为每个8086任务分配一个受保护的1MB的地址空间。

虚拟8086方式特点如下：

- 允许多任务运行；
- 段均为64KB；
- 代码和数据段具有保护机制；
- 支持任务之间以及特权级的数据和程序保护

3 Pentium微处理器



P4系列 478根

P4系列 775根

P4 笔记本系列 479根

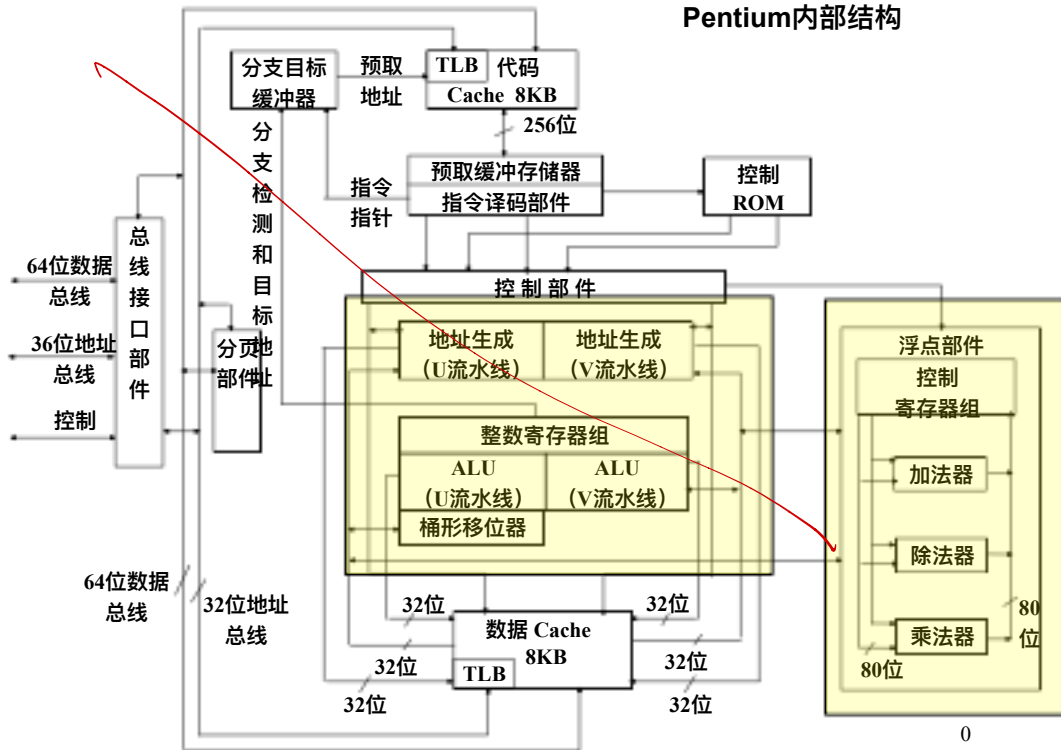
P3 志强 603根

P4 志强 604根

酷睿2 2核 775

酷睿i7 4核 1150

Pentium内部结构



● Pentium的先进技术

1) 先进的体系结构

- 外部采用64位数据总线，内部32位总线；
- 片内有指令Cache和数据Cache；
- 使用了二条指令流水线并行执行指令；
- 内部集成了增强型浮点处理部件(FPU—Floating processor unit)；
- ADD、MUL、INC、DEC、PUSH、POP、JMP、CALL和LOAD等常用指令采用硬件实现；
- 增加了系统管理方式，增强了对系统的管理，比如：对操作系统的管理、对正在运行的程序的管理、对电源的管理等。

2) CISC和RISC相结合

大多数指令是简化指令，但仍然保留了一部分复杂指令，对这部分指令采用硬件来实现。

3) 超标量流水线技术

① 超标量和流水线的概念

Pentium指令流水线由总线接口部件、指令预取部件、指令译码部件和执行部件构成。

超标量：配置多个执行部件和指令译码电路，能同时执行多条指令。

② 整数流水线

- **PF**(预取)：处理器从代码cache中预取指令
- **D1**(译码阶段1)：处理器对指令译码确定操作码和寻址信息。在这个阶段还进行指令的成对性检查和分支预测
- **D2**(译码阶段2)：产生访问存储器的地址
- **EX**(执行)：处理器或者访问数据cache，或者利用ALU、筒型移位器或其他功能单元计算结果
- **WB**(写回)：利用指令运行结果更新寄存器和标志寄存器

③ 浮点流水线

八个阶段的浮点流水线：

- PF 预取
- D1 译码阶段1
- D2 译码阶段2
- EX 取操作数
- X1 浮点执行步骤1
- X2 浮点执行步骤2
- WT 写浮点数

~~浮点流水线中的前五个步骤~~与整数流水线中的五个步骤是同步执行的，只是多出三个步骤

Pentium的超标量流水线



- 指令配对

U、V两条流水线并行执行的要满足一些前提条件，Pentium定义了指令配对规则，如没有写后读和写后写的依赖关系，不能同时既包含位移量又包含立即数，带前缀的指令只能出现在U流水线中。

- 分支预测技术

增加流水线的效率，是一种“猜测”。

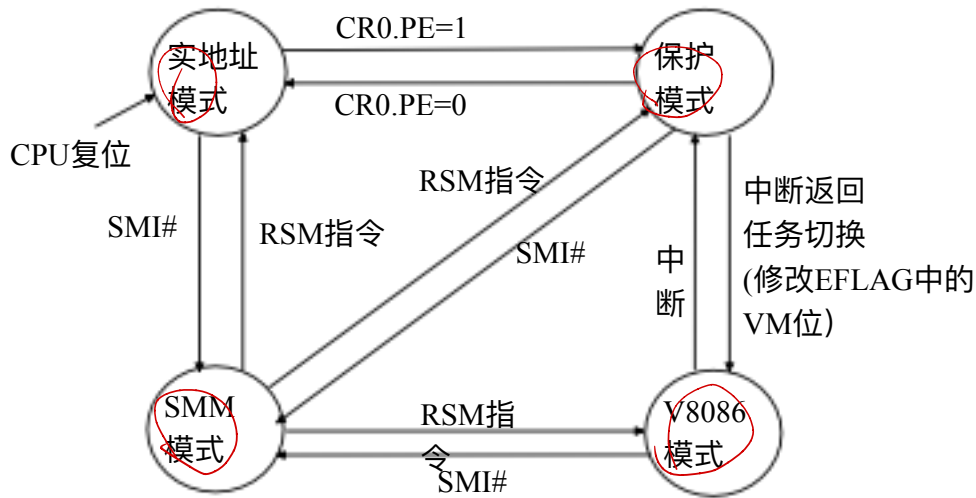
BTB——分支目标缓冲器

- 1KB的Cache，可容纳256条转移指令的目标地址和历史状态
- 动态预测，检测分支指令的执行状态并据此预测分支目标地址

两个预取缓冲存储器

- 按照BTB预测结果预取指令
- 以预测排除的方向预取指令

● Pentium的工作方式



说明: $SMI\#$ 是来自外部系统管理中断引脚的信号

RSM 指令是从系统管理中断服务程序中返回的指令

● 32位微处理器的虚拟存储机制

虚拟存储技术

虚拟存储器是指程序使用的逻辑存储空间,它可以比物理存储器大的多,其对应的地址叫**虚拟地址**,也叫**逻辑地址**。

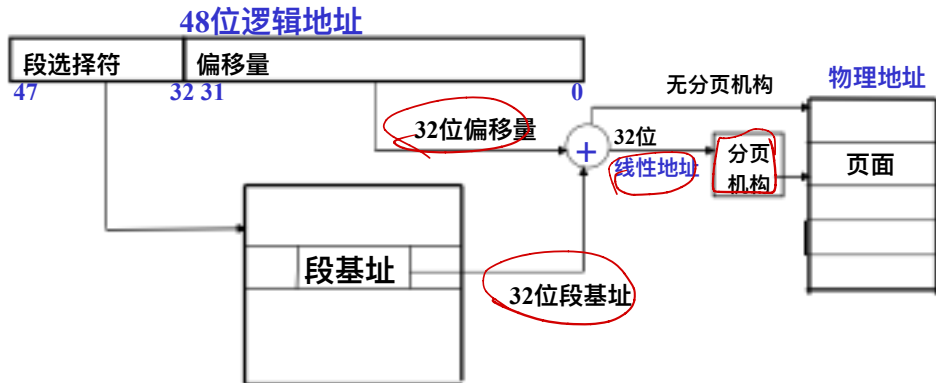
虚拟存储机制由主存储器、辅助存储器和存储管理部件共同组成。

逻辑地址: 程序中使用的地址,也称虚拟地址。

线性地址: 由段基地址(32位)和段内偏移量(32位)组成的地址。

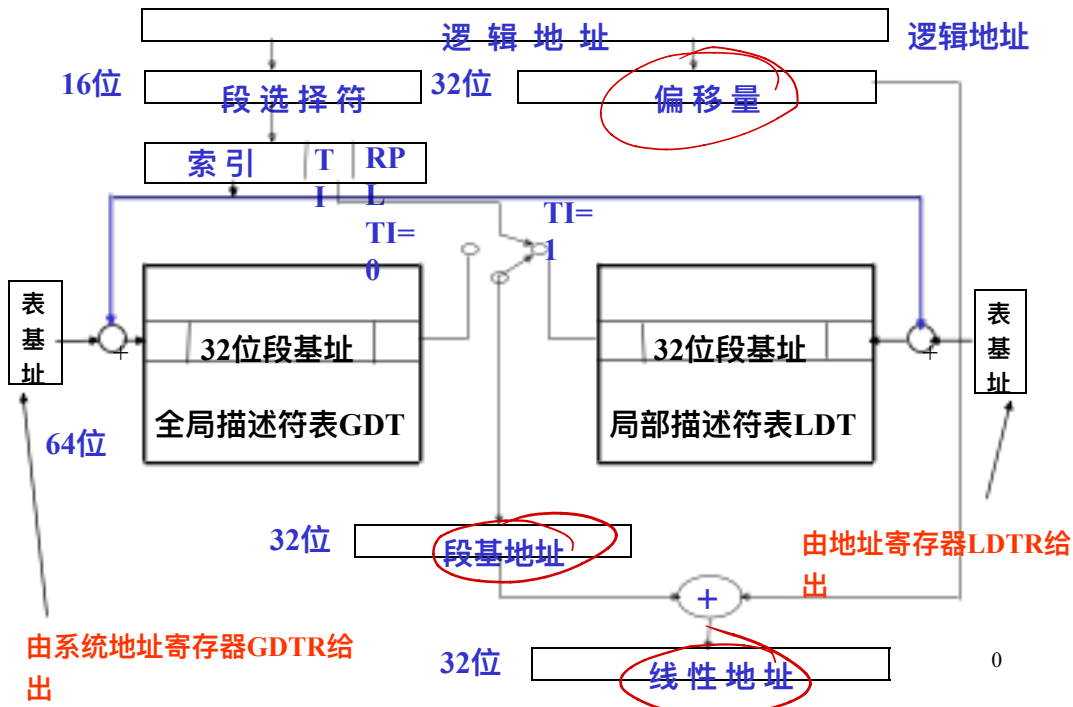
物理地址: 指由地址总线直接访问的存储器的地址。

逻辑地址、线性地址和物理地址关系

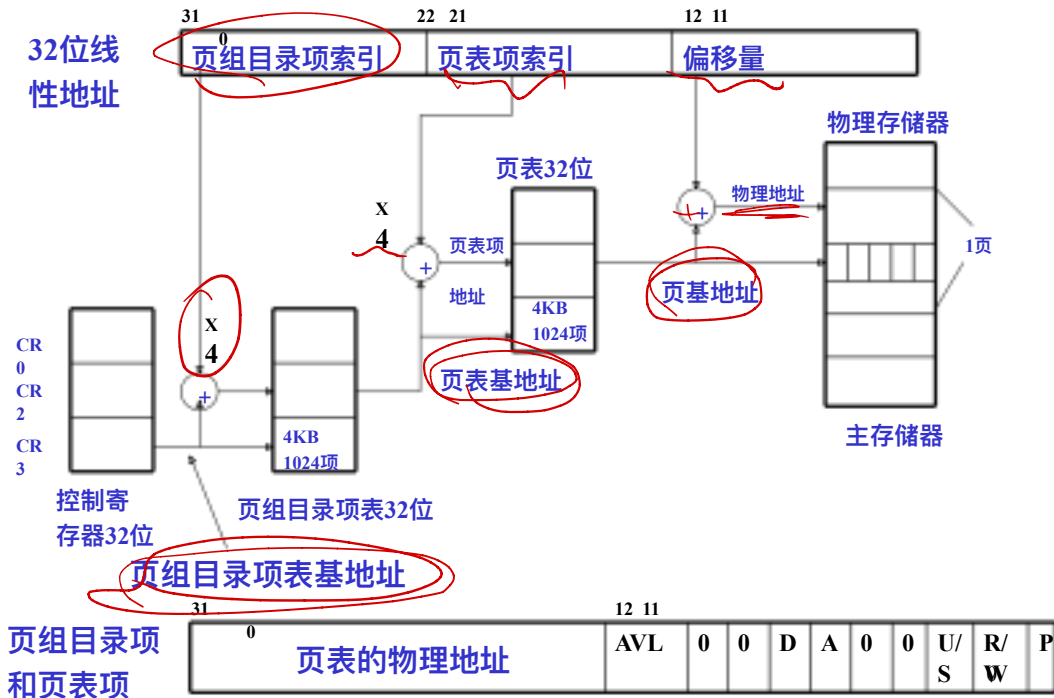


分段部件实现从逻辑地址到线性地址的转换

48位



分页机构实现从线性地址到物理地址的转换



页表项

页面的物理地址	AVL	0	0	D	A	0	0	U/S	R/ W	P
---------	-----	---	---	---	---	---	---	-----	---------	---

A =1 访问过此页表

=0 未访问过此页表

D =1 该页被修改过

=0 该页未被修改

U/S =0 该页面为系统级

=1 该页面为用户级

P = 0 未装入内存

= 1 已装入内存

R/W = 0 只读

= 1 可读写

AVL: 保留，一般操作系统用于统计该页访问频度

地址转换过程

微处理器需要访问内存数据时，逻辑地址到物理地址的转换：

- 1) 根据当前指令的性质确定使用哪一个段寄存器，例如：转移指令中的地址在代码段，而取数指令中的地址在数据段。
- 2) 根据段寄存器的内容，找到相应的地址段描述结构。
- 3) 从地址段描述结构中得到段基地址。
- 4) 将指令中发出的地址作为位移，与段描述结构中规定的段长度相比，看看是否越界。
- 5) 根据指令的性质和段描述符中的访问权限来确定是否越权。
- 6) 将指令中发出的地址作为位移，与基地址相加而得出线性地址。
- 7) 如无分页，则该线性地址即物理地址；如有分页，则根据线性地址查找页表，得到物理地址。

Pentium操作系统工作：

- a) 建立初始页组目录表和页表，完成线性地址到物理地址的转换
- b) 完成存储数据的交换，缺页故障时调度
- c) 周期性检测页表项的访问位A
- d) 确保TLB的内容和分页部件两个表相符

- **Pentium的保护技术**

为了支持多任务，Pentium处理器要对存放在存储器中的代码及数据的保护与共享提供支持。Pentium的保护机制能有效地实现不同任务之间的保护和同一任务内的保护。

- **使用局部地址空间，实现任务的隔离**

任务A和任务B并存，那么任务A和任务B必须隔离，以免相互影响。

两个不同任务的虚拟地址转换为不同的物理地址。仅由一个任务占有的虚拟地址空间部分，称为局部地址空间。局部地址空间包含的代码和数据，是任务私有的，需要与系统中的其它任务相隔离。

- **使用全局地址空间，实现资源的共享**

任务A和任务B可能要共享部分代码和数据，如对操作系统的共享。

各个任务公用的虚拟地址空间部分，称为全局地址空间。对全局地址空间中同一虚拟地址的访问，在所有任务中都转换为同样的物理地址，从而支持公共的代码及数据的共享。

1 段页二级保护机制

自己看一下

1) 存储器的段级保护

A. 段类型提供读/写保护

数据段的段描述符中，用W控制可否向此段写入信息；
代码段的段描述符中，用R控制可否从此段读取信息。

B. 界限和粒度提供范围保护

段描述符中，“段界限”字段表示段长度；
“粒度G”为0，则段大小为4KB；为1，则大小为4GB。

C. 特权级对操作系统和驱动程序提供保护

段描述符中用DPL字段设置4个特权级0~3，0级最高，3级最低，用于限制对存储器段进行访问。

段基地址B0—B15					段界限值L15—L0						
段基地址 B31—B24	G	D	0	A V L	段界限值 L19—L16	P	DP L	S	段类 型	A	段基地址 B23—B16
		/									
		B									

2) 存储器的页级保护

➤ 页的特权级提供页保护

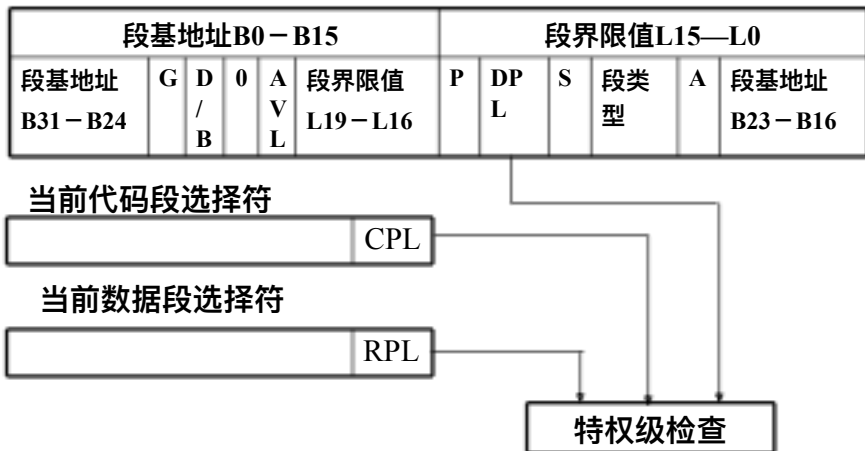
页表中，U/S字段，表示页面特权级，分为管理级和用户级

➤ 标志R/W提供页面写保护

页面是否可以进行写操作

	AVL	0	0	D	A	0	0	U/S	R/ W	P
--	-----	---	---	---	---	---	---	-----	---------	---

2 程序运行中的保护



描述符特权级DPL：数据段的DPL每次被访问时受到检查,以确定程序是否有权访问该数据段

当前特权级CPL：表示当前正在运行的程序的特权级。

请求特权级RPL：只有当代码段的CPL的级别不低于RPL时才能使访问顺利进行。

