



第2章 80x86微处理器

2.1 8086微处理器

2.2 8086 CPU的存储器组织及I/O结构

2.3 8086微处理器的外部引脚及工作模式

2.4 8086 微处理器的总线操作与时序

2.5 Intel的其他微处理器



2.1 8086微处理器

对Intel 8086微处理器的了解:

- ✓ 典型的**16**位微处理器;
- ✓ 采用**HMOS**工艺制造, 芯片上集成了**29000**个晶体管;
- ✓ 用单一的**+5V**电源, 被封装在标准**40**引脚的双列直插式(**DIP**)管壳内;
- ✓ 时钟频率**5MHz~10MHz**, 内、外部数据总线均为**16**条;
- ✓ 地址总线**20**条, 可寻址的存储空间为 **$2^{20}=1\text{MB}$** , 其中的**16**条地址总线, 可以访问 **$2^{16}=64\text{KB}$** 的I/O端口

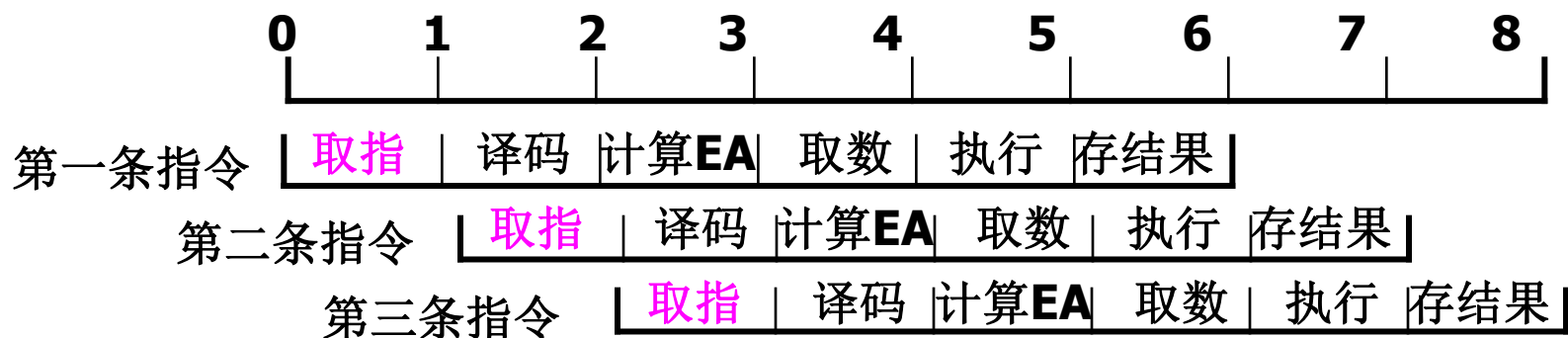


2.1 8086微处理器

一、8086 CPU的内部结构

采用**流水线处理技术**：将指令执行部件（**EU**）和总线接口部件（**BIU**）分为两个独立的部分，并可并行操作。

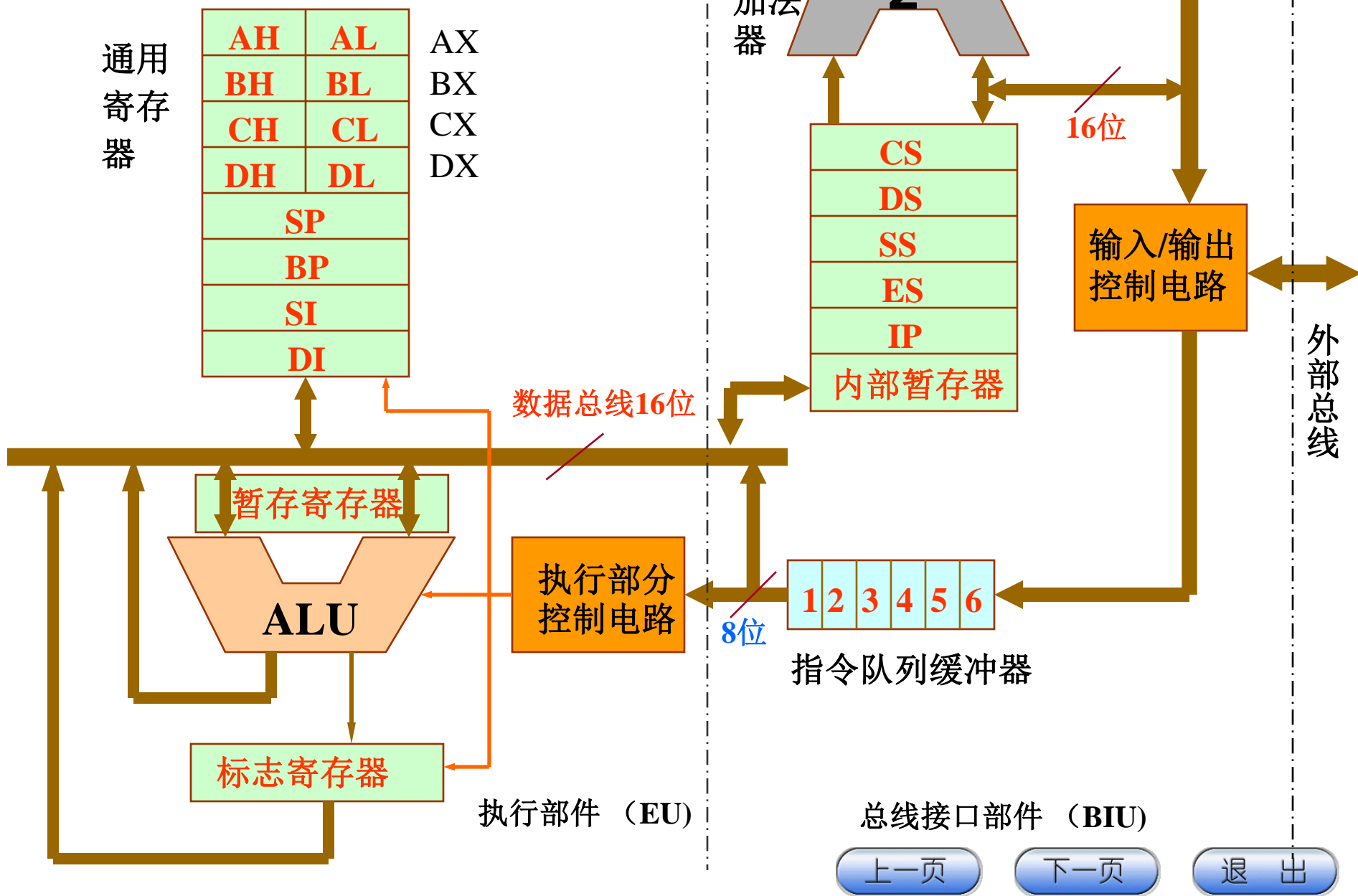
流水技术：在执行指令时预取下一条指令的技术，是一种实现多条指令重叠执行的重要技术



可见，3条指令共需8个时间单位，即可全部执行完；如果完全串行执行，则需 $3 \times 6 = 18$ 个时间单位



8086CPU内部结构图





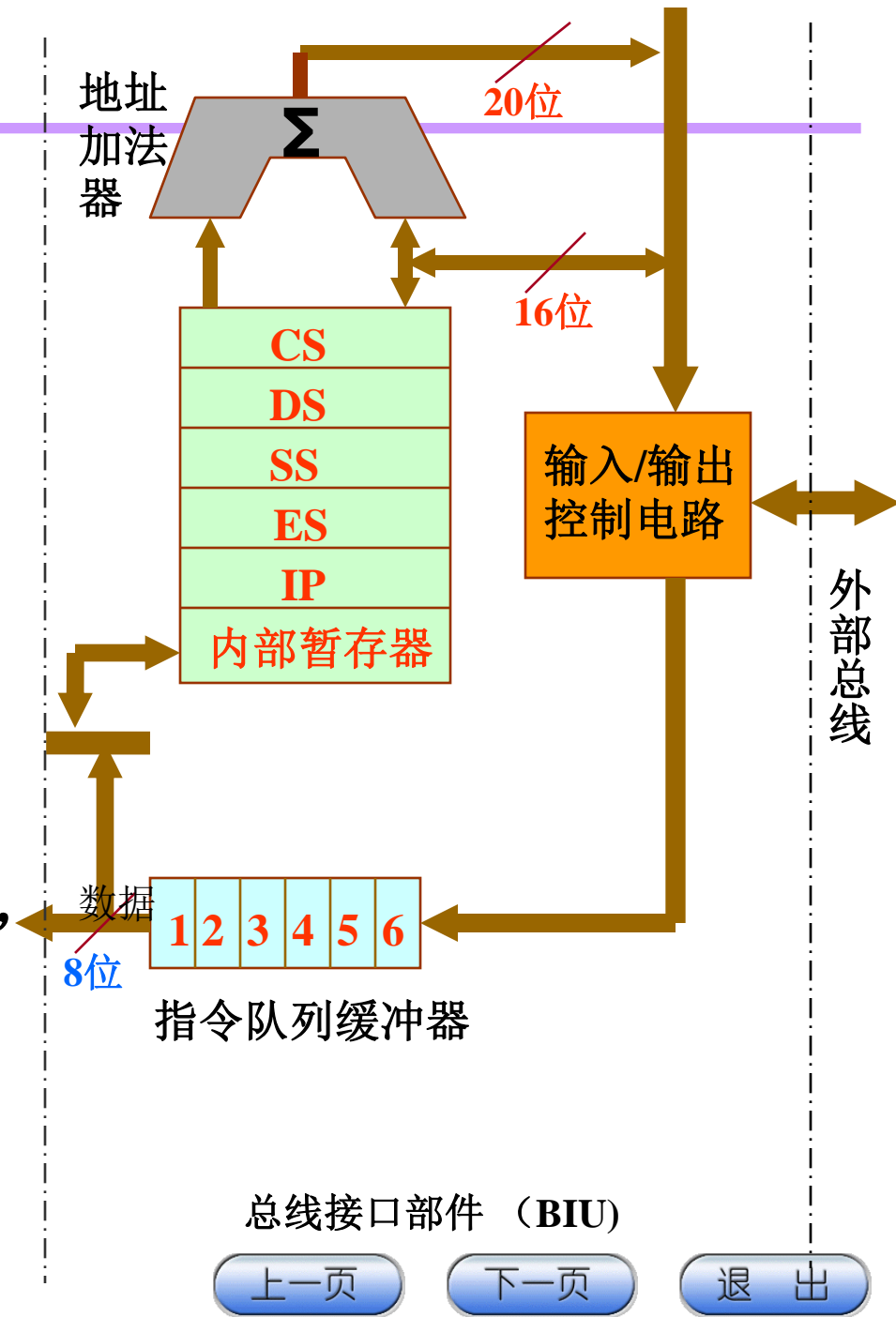
8086CPU内部结构图

BIU: CPU与存储器及I/O的接口，内部由20位地址加法器、指令队列缓冲器、内部寄存器和总线控制逻辑等组成；完成取指令、指令排队、读/写操作数、地址转换与总线控制等工作。

20位地址加法器： 将逻辑地址转换成20位物理地址，完成地址加法操作

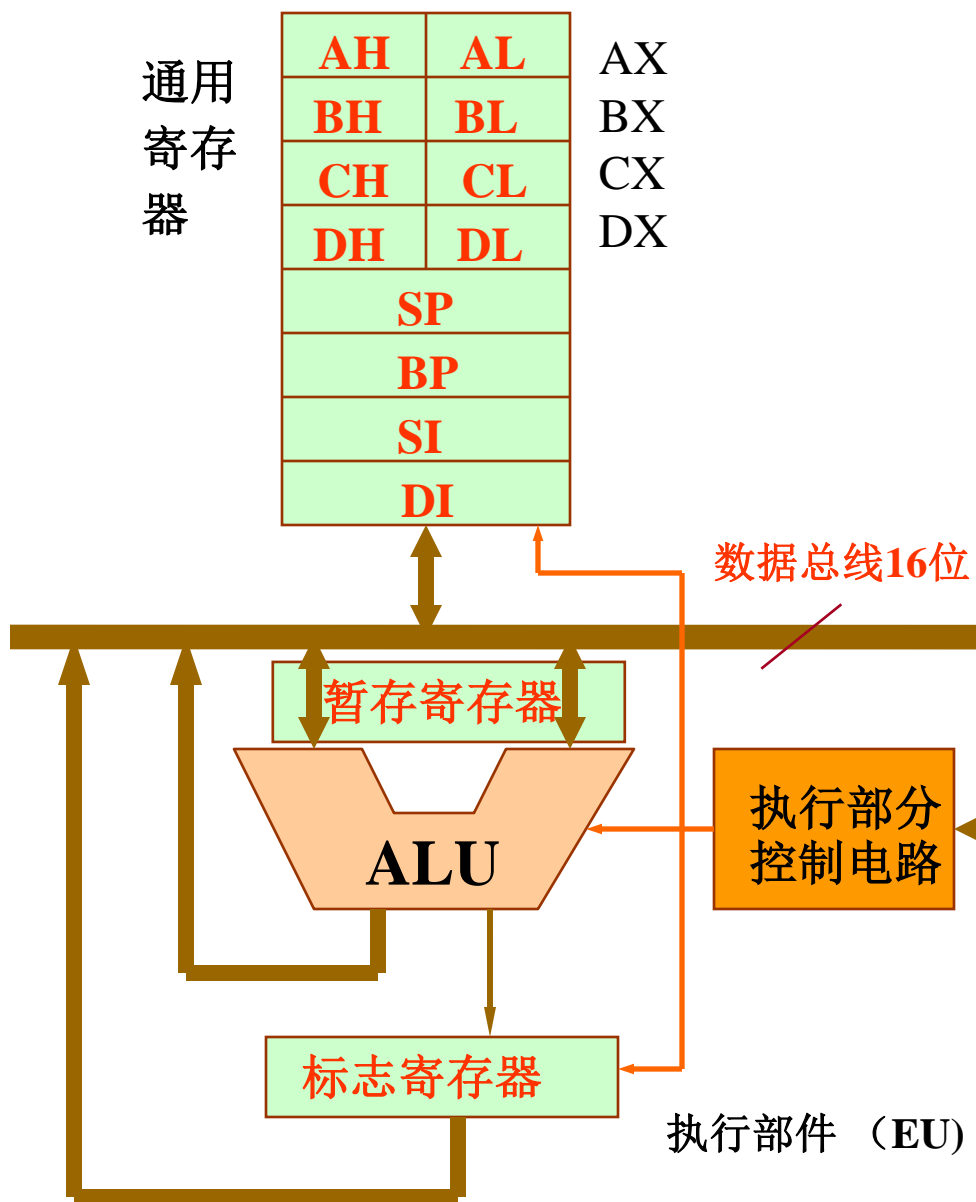
指令队列缓冲器： 具有6字节的“先进先出”的RAM存储器，按顺序存放CPU要执行的指令，并送入EU中去执行

总线控制电路： 产生并发出总线控制信号，实现对存储器或I/O端口的读/写控制





8086CPU内部结构图

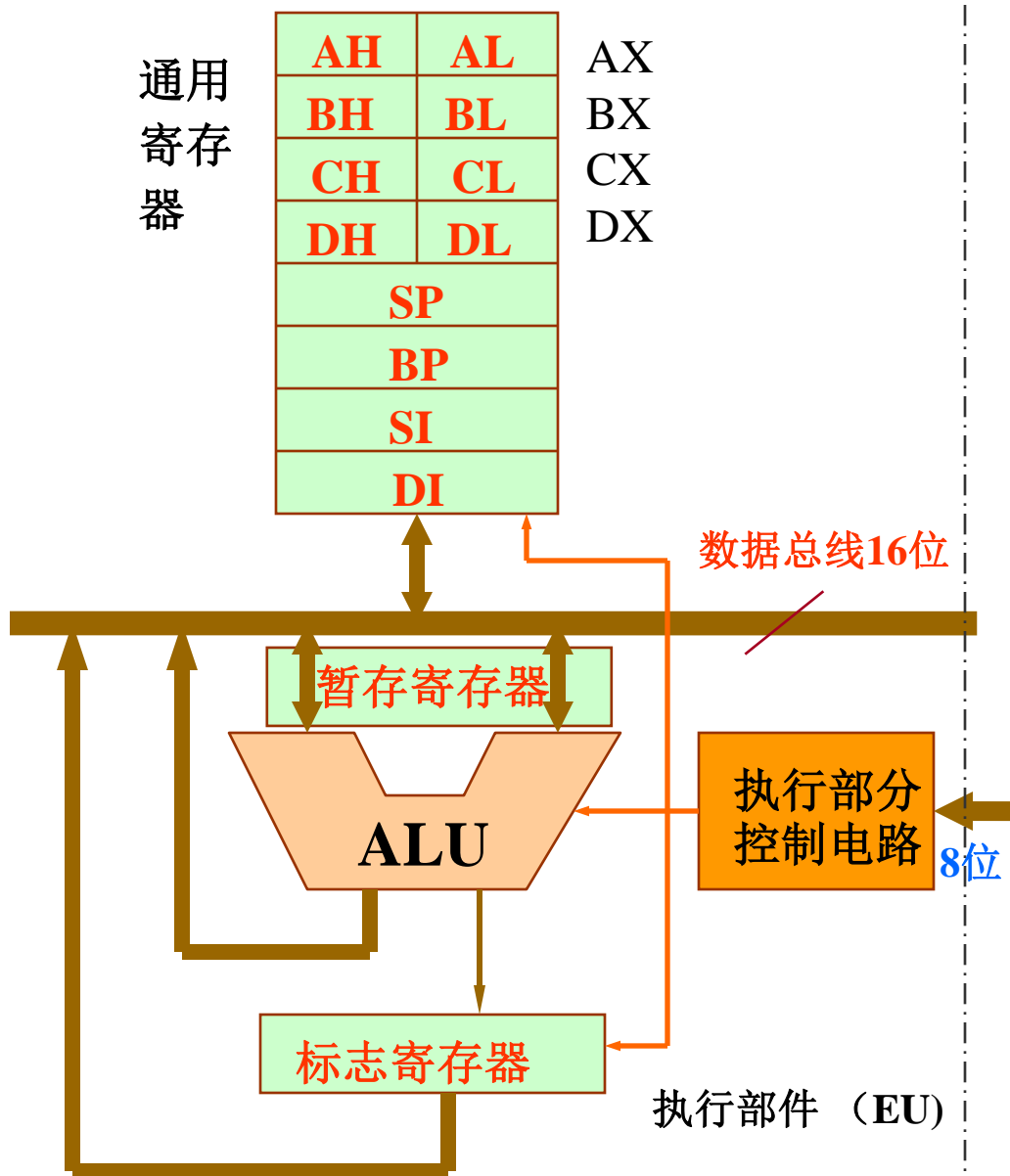


EU: 接收从BIU指令队列中取来的指令代码，经过分析、译码后形成各种实时控制信号，向EU内各功能部件发送相应的控制命令，完成每条指令所规定的操作

EU由ALU、标志寄存器、8个通用寄存器、暂存器、队列控制逻辑与时序控制逻辑（EU控制器）等组成；



8086CPU内部结构图



算术逻辑单元ALU：16位算术逻辑运算部件，实现对16/8位二进制操作数的算术、逻辑运算

暂存寄存器：16位寄存器，主要功能是暂时保存数据，并向ALU提供参与运算的操作数

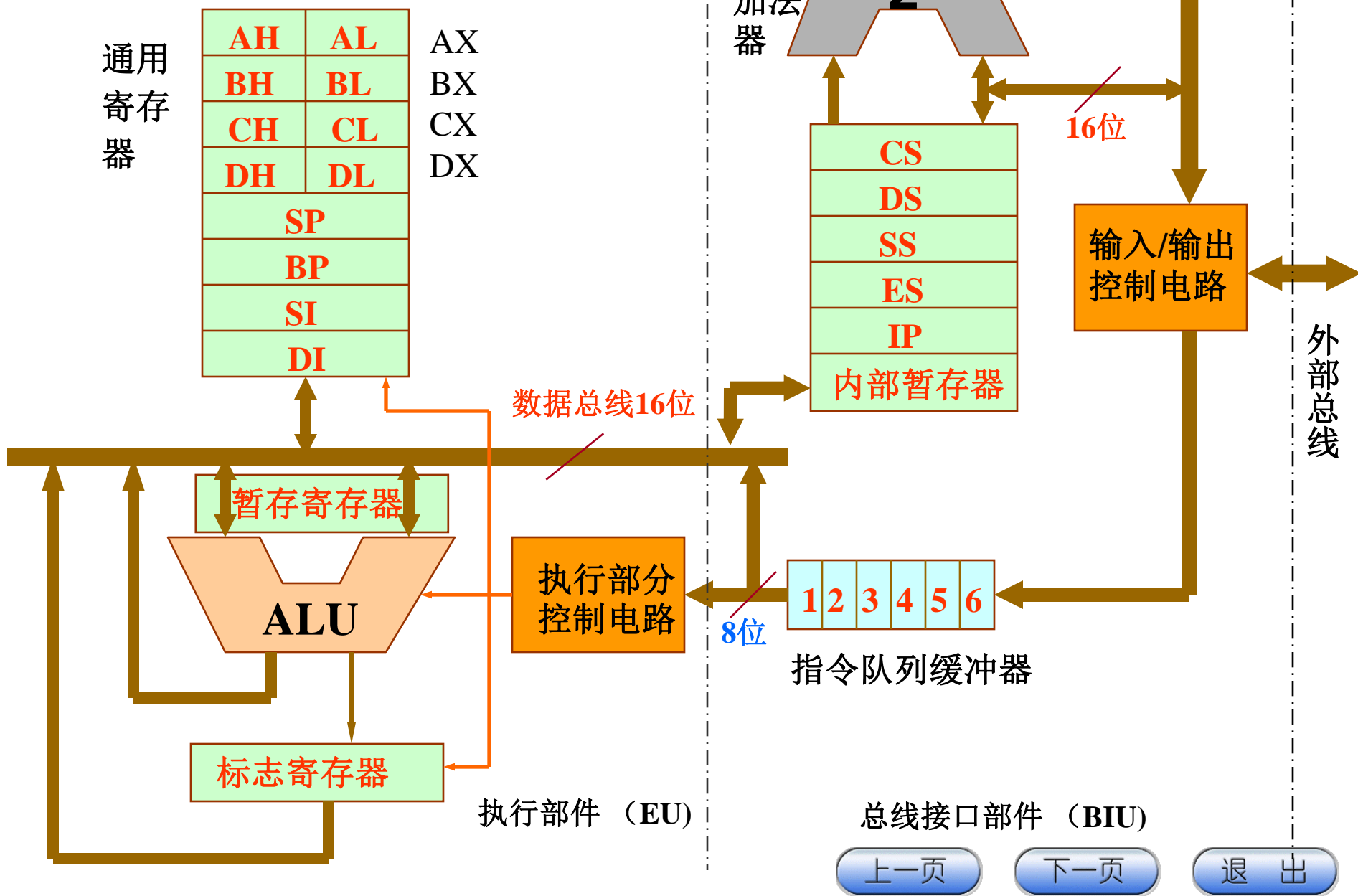
标志寄存器：一个16位的寄存器

通用寄存器：8个16位的寄存器

EU控制器：接收从BIU指令队列中取来的指令代码，经分析、译码后形成各种实时控制信号，向EU内各功能部件发送相应的控制命令，以完成每条指令所规定的操作。



8086CPU内部结构图





2.1 8086微处理器

二、8086的寄存器结构

寄存器：是CPU内部临时存放数据的部件。



8086 CPU 的寄存器结构



2.1 8086微处理器

1、数据通用寄存器

包括AX, BX, CX和DX四个16位寄存器。其中任意一个寄存器均可分为两个8位寄存器使用，它们的特殊用途和隐含性质如下表所示

通用寄存器中的 4 个数据寄存器可以分成8个8位寄存器使用：

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

程序员可利用数据寄存器的这种“可分可合”的特性，灵活地处理字/字节的信息。

用途：存放8位或16位操作数或中间结果，以提高CPU的运算速度（减少存取MEM的时间）

其中，AX是CPU使用最多的一个寄存器，功能最强。

AX的作用 { 算术逻辑运算之前保存一个操作数，之后保存结果。
CPU与 I/O、MEM交换数据所用到的最多的寄存器。



2.1 8086微处理器

特殊用法:

AX — 算术运算、I/O传输的主要寄存器（Accumulator）

BX — 计算地址时用作基址Reg.（Base）

CX — 计数器，循环或移位时用。（Counter）

DX — I/O 操作时用于保存I/O端口地址，
或字的乘除法运算时用来存放高16位。（Data）

8086 CPU 中通用寄存器的特殊用途和隐含性质

寄存器名	特殊用途	隐含性
AX, AL	在输入/输出（IN/OUT）指令中作数据寄存器用	不能隐含
	在乘法指令中存放被乘数或乘积，在除法指令中存放被除数或商	隐含
AH	在 LAHF 指令中，作目的寄存器用	隐含
	在十进制运算指令中作累加器用	隐含
AL	在 XLAT 指令中作累加器用	隐含
BX	在寄存器间接寻址中作基址寄存器用	不能隐含
	在 XLAT 中作基址寄存器用	隐含
CX	在串操作指令和 LOOP 指令中作计数器用	隐含
CL	在移位/循环移位指令中作移位次数计数器用	不能隐含
DX	在字乘法/除法指令中存放乘积的高一半或者是被除数的高一半或余数	隐含
	在间接寻址的输入/输出（IN/OUT）指令中作端口地址用	不能隐含



2.1 8086微处理器

2、指针和变址寄存器

经常用于存储器寻址时，提供**偏移地址**。

SP(堆栈指针寄存器):与 **SS**(**堆栈**段寄存器)一起确定栈顶的当前位置；

BP(基址指针寄存器): 与**SS**用来确定堆栈段中的某一存储单元的**地址**；

SI和**DI**: 主要用于存放存储单元在段内的偏移地址，可实现多种存储器操作数的寻址方式，为以不同的地址形式访问存储单元提供方便。



2.1 8086微处理器

堆栈：按照“**先进后出**”的原则存取数据

INTEL系列的**CPU**用内存作为堆栈，段地址存放在**SS**中，栈顶地址存放在**SP**中；**8086 CPU**堆栈操作以**字**为单位，把数据推入堆栈称“**压入（PUSH）**”操作，从堆栈中取数据称“**弹出（POP）**”操作

为什么要用堆栈？

主程序调用子程序（子程序调用子程序，或处理中断服务程序），**CPU**必须把主程序调用子程序指令的**下一条指令的地址（或中断时的断点）即PC值保留下来**，才能保证子程序（或中断服务程序）执行完之后正确返回到主程序继续执行；另外，调用子程序之前的有关**REG**、标志位也要及时正确的保存下来。



2.1 8086微处理器

3、指令指针寄存器IP

存放**代码段**中的偏移地址。在程序运行过程中，始终指向下一条指令的首地址，与**CS**一起形成取下一条指令的实际地址

4、标志寄存器 Flags

D ₁₅	D ₁₄	D ₁₃	D ₁₂	D ₁₁	D ₁₀	D ₉	D ₈	D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
				OF	DF	IF	TF	SF	ZF		AF		PF		CF

8086 CPU 标志寄存器 Flags

Flags：共**16**位，一般把**每一位**分别使用，**8086**使用其中的**9**位，用于存放当前**程序执行**的状况和**运算结果**的特征。**9**个标志位分为二组：

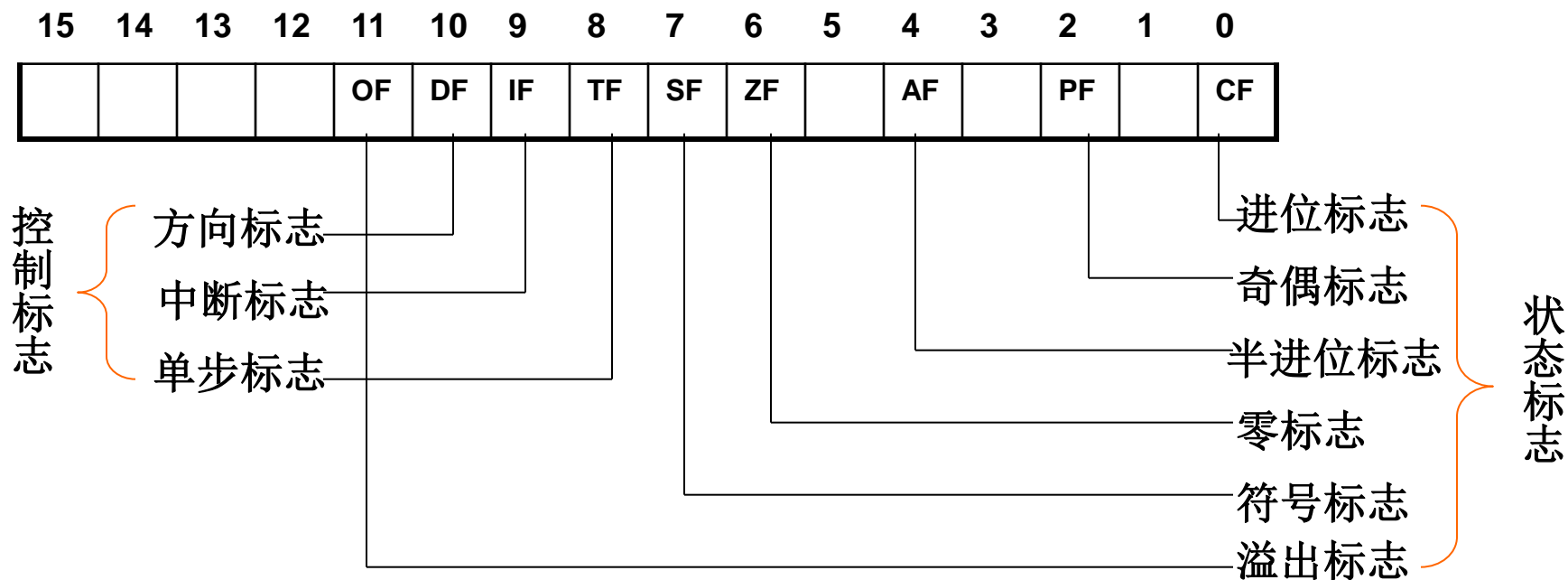
状态标志：反映指令执行的结果（受算术运算等指令结果的影响）；

控制标志：通过指令来设置，以控制某些指令的执行方式。



2.1 8086微处理器

标志寄存器的格式及各位的含义





2.1 8086微处理器

状态标志

进位标志CF(Carry Flag): 当结果的最高位(D_{15} 或 D_7)产生一个进位或借位, 则 $CF=1$, 否则 $CF=0$ 。

溢出标志OF(Overflow Flag): 当带符号数的运算结果超出 $-2^{n-1} \sim 2^{n-1} - 1$ 时, 溢出, $OF=1$, 否则 $OF=0$ 。

符号标志SF(Sign Flag): 结果的最高位(D_{15} 或 D_7)为1, 则 $SF=1$, 否则 $SF=0$ 。

零标志ZF(Zero Flag): 若运算的结果为0, 则 $ZF=1$, 否则 $ZF=0$ 。

奇偶标志PF(Parity Flag): 若运算结果的低8位中 '1'的个数为偶数, 则 $PF=1$, 否则, $PF=0$ 。

辅助进位标志AF(Auxiliary Flag): 在进行8位或16位数运算时, 由低4位向高4位(D_3 向 D_4)有进位或借位, 则 $AF=1$, 否则 $AF=0$ 。

控制标志

方向标志DF(Direction Flag): $DF=1$, 串操作时地址自动减量;
 $DF=0$, 串操作时地址自动增量。

中断允许标志IF(Interrupt Enable Flag): $IF=1$, 允许CPU接收外部中断请求, $IF=0$, 屏蔽外部中断请求

单步标志TF(Trap Flag): $TF=1$, 使处理进入单步方式, 以便于调试。



2.1 8086微处理器

示例1：将数**5439H**与数**456AH**相加，并分析相加运算对标志位的影响

$$\begin{array}{r} 0101\ 0100\ 0011\ 1001 \\ +\ 0100\ 0101\ 0110\ 1010 \\ \hline 1001\ 1001\ 1010\ 0011 \end{array}$$

CF=0, PF=1, AF=1, ZF=0, OF=1, SF=1



2.1 8086微处理器

例：执行2345H - 3219H，分析对状态标志位的影响。

	0	0	1	0	0	0	1	1	0	1	0	0	0	1	0	1
+	1	1	0	0	1	1	0	1	1	1	1	0	0	1	1	1
<hr/>																
	1	1	1	1	0	0	0	1	0	0	1	0	1	1	0	0

标志：

运算结果最高位为1

∴SF=1

运算结果本身≠0

∴ZF=0

低8位中1的个数为奇数个

∴PF=0

最高位没有进位（无进位，有借位）

∴CF=1

D₃位向D₄位无进位（无进位，有借位）

∴AF=1

次高位向最高位没有进位，最高位向前没有进位 ∴OF=0



2.1 8086微处理器

5、段寄存器

专门存放段地址的寄存器，8086共有4个：

CS—代码段段寄存器；

DS—数据段段寄存器；

ES—附加段段寄存器；

SS—堆栈段段寄存器。

段寄存器都为16位，用于存放相应逻辑段的段地址



2.2 8086CPU存储器组织及I/O结构

一、存储单元的地址和内容

1、存储单元的地址

物理地址：每个存储单元的唯一地址。

- (1) 物理地址的空间呈线性增长，从**0**开始编号，顺序每次加**1**。
- (2) 在**机器里**，物理地址用**无符号二进制数**表示，**书写格式**为十六进制数。如**23450H**、**0A209FH**。
- (3) **8086**的内存以字节编址（数据在内存中以**字节为单位**进行存储），每个内存单元地址唯一，存放一个字节。字节是**8086**中最基本的数据**存取**单位。



2.2 8086CPU存储器组织及I/O结构

内存储器	A ₁₉	A ₁₈	A ₁₇	A ₁₆	A ₁₅	A ₁₄	A ₁₃	A ₁₂	A ₁₁	A ₁₀	A ₉	A ₈	A ₇	A ₆	A ₅	A ₄	A ₃	A ₂	A ₁	A ₀	十六进制地址
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000H
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	00001H
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	00002H
3AH	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	00003H
56H																					
FFH	...																				
00H																					
	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	FFFFDH
	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	FFFFEH
	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	FFFFFH

物理地址顺序排列示意图

存取**1个字**或**双字**的方法:

➤ 1个字占据**两个相邻的内存单元**，低字节放在低地址单元，高字节放在高地址单元；**字的地址由其低地址来表示**。

双字类似字。

➤ 对于同一个内存地址，既可以看作是字节单元的地址，也可以看作是字或双字单元的地址，取决于存取的数据类型。



2.2 8086CPU存储器组织及I/O结构

2、存储单元的内容：存储单元中所存放的信息。

说明:

- ✓ 在指令中，若 X 表示某单元的地址，则 $[X]$ 表示 X 单元的内容，并且可以嵌套。
- ✓ 存储器的特性：取之不尽，新信息存入覆盖原有信息。



2.2 8086CPU存储器组织及I/O结构

二. 存储器的分段与物理地址的形成

8086/8088率先打破微处理器只能访问64KB存储空间的限制，可寻址1MB。

1、存储器的分段

分段原因：8086 CPU有20根地址线，16根数据线。CPU内部寄存器都是16位的。存放地址的寄存器是16位的，只可表示16位的地址。为了能寻址1MB地址空间，8086对存储器进行分段管理。

段的分法：把1MB内存空间划分成若干个逻辑段（简称为段），每个段的要求为：

- 段的起始地址必须是16的倍数，即最低4位二进制全为0；
- 每个段最大为64KB (2^{16})



2.2 8086CPU存储器组织及I/O结构

逻辑段之间关系：可以重叠(如B和C段)，可以连续(如C和D段)可以不连(如A和B段)。段的起始地址的计算和分配通常由操作系统完成

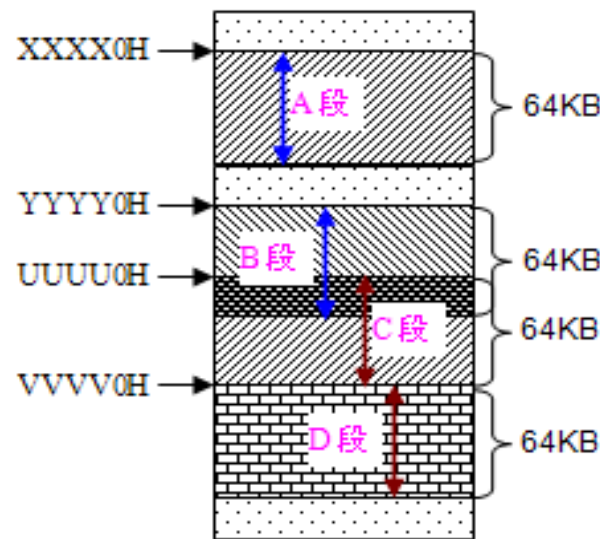
说明：段地址表示段的起始地址(XXXXH)，

只有形式为XXXX0H的地址才能作

段的起始地址，该起始地址存入16位寄存

器时，将0省略即可。存放在CS、DS、SS、

ES段寄存器中



8086 内存分段管理示意图



2.2 8086CPU存储器组织及I/O结构

2、逻辑地址与物理地址

存储器分段后，对内存的访问就可以使用逻辑地址和物理地址。

逻辑地址：程序设计时所使用的地址。表示为：

段地址:偏移地址

偏移地址（段内偏移或有效地址或偏移量）：所访问的存储单元距段的起始地址之间的**字节距离**。

“段地值”和“偏移地址”均**16位**。“段地值”由段寄存器**CS、DS、SS、ES**提供；“偏移地址”由**BX、BP、SP、SI、DI、IP**或以这些寄存器的组合形式提供。

注意：SP与IP没有提供给用户编程使用



2.2 8086CPU存储器组织及I/O结构

物理地址：数据在**存储器中实际存放**的地址，**CPU**访问存储器时实际输出的地址。**8086 CPU**的存储空间 $2^{20}=1\text{MB}$ ，地址范围**00000H**到**FFFFFFH**。

逻辑地址：用户**程序设计**时采用的地址；同一个物理地址可对应多个逻辑地址。对逻辑地址通过转换，能找到所对应的物理地址。

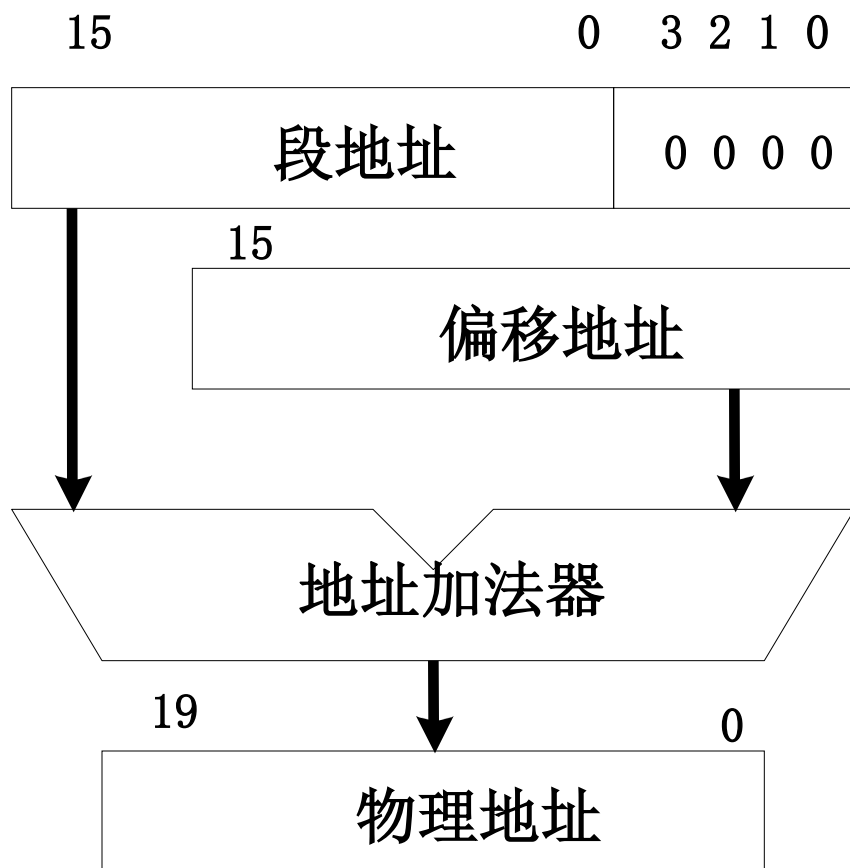
转换方法：

物理地址=段地值×16+偏移地址

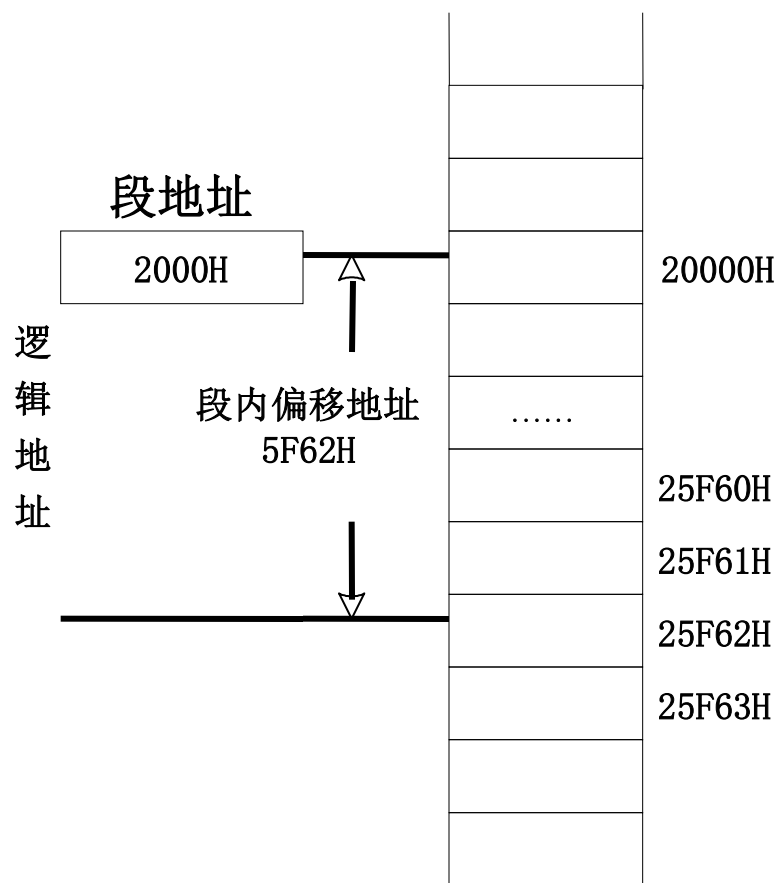


2.2 8086CPU存储器组织及I/O结构

存储器的分段结构和物理地址的形成



物理地址的形成



逻辑地址与物理地址



2.2 8086CPU存储器组织及I/O结构

3、段寄存器的引用

段寄存器为分段管理而设置。4个段寄存器（**CS**、**DS**、**ES**、**SS**）可同时访问4个不同含义的段。

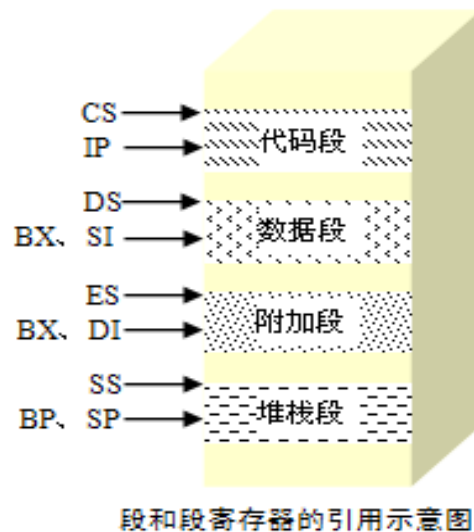
段寄存器及其偏移地址的引用关系

CS: IP 取下次要执行的指令。

SS: SP 访问栈顶单元。

SS: BP 可访问整个堆栈。

DS数据段和**ES**附加段，存取操作数时，二者之一与一个偏移地址合并可得到存储单元的物理地址。该偏移地址可以是具体数值、符号地址和指针寄存器的值等之一，具体由指令的寻址方式决定。



段和段寄存器的引用示意图



2.2 8086CPU存储器组织及I/O结构

段寄存器及其指针寄存器的引用关系

段寄存器及其指针寄存器的引用关系

访问存储器方式		缺省的段寄存器	可选用的段寄存器	偏移地址
取指令		CS		IP
堆栈操作		SS		SP
一般取操作数		DS	CS、ES、SS	有效地址
串操作	源操作数	DS	CS、ES、SS	SI
	目的操作数	ES		DI
使用指针寄存器 BP		SS	CS、DS、ES	有效地址



2.2 8086CPU存储器组织及I/O结构

三、8086 CPU的I/O结构

8086 CPU用地址总线的低16位 $A_{15} \sim A_0$ 寻址端口地址，可访问的I/O端口地址有 $2^{16} = 64K$ 个，地址范围为 $0000H \sim FFFFH$ ，实际上只使用了 $A_9 \sim A_0$ 共10条地址线作为I/O端口的寻址线，故最多可寻址1024个端口地址，地址范围为 $0000H \sim 03FFH$ 。

2.3 8086 微处理器的外部引脚及工作模式

一、总线周期的定义

时钟周期:主时钟的两个脉冲前沿的时间间隔，又称为**T状态**（即一个**T状态**就是一个时钟周期）。是**CPU**执行操作的最小时间单位，如**8086**的时钟频率为**5MHz**，则一个**T状态**为**200NS**。

总线周期:又称为**机器周期**或**M周期**，是**CPU**从存储器或**I/O**端口读写一个字节或字所需要的时间。总线周期通常包含**4个T状态**: T_1 , T_2 , T_3 , T_4 。

指令周期:一条指令（的取出和）执行所需的时间。一条指令执行需**1~5个指令周期**。

结论:

(1) **8086**中不同指令的指令周期是不等长的。

(2) 各条指令的指令周期虽各不相同，但它们仍然由一些基本的总线周期组成: 存储器读或写总线周期; 输入/输出端口的读或写总线周期; 中断响应周期

2.3 8086 微处理器的外部引脚及工作模式

二、8086 CPU的两种工作模式

根据所连的存储器和外设的规模，8086可以有**两种**不同的工作模式。

最小模式：系统中只有一个8086微处理器，所有的总线控制信号都直接由8086 CPU产生；

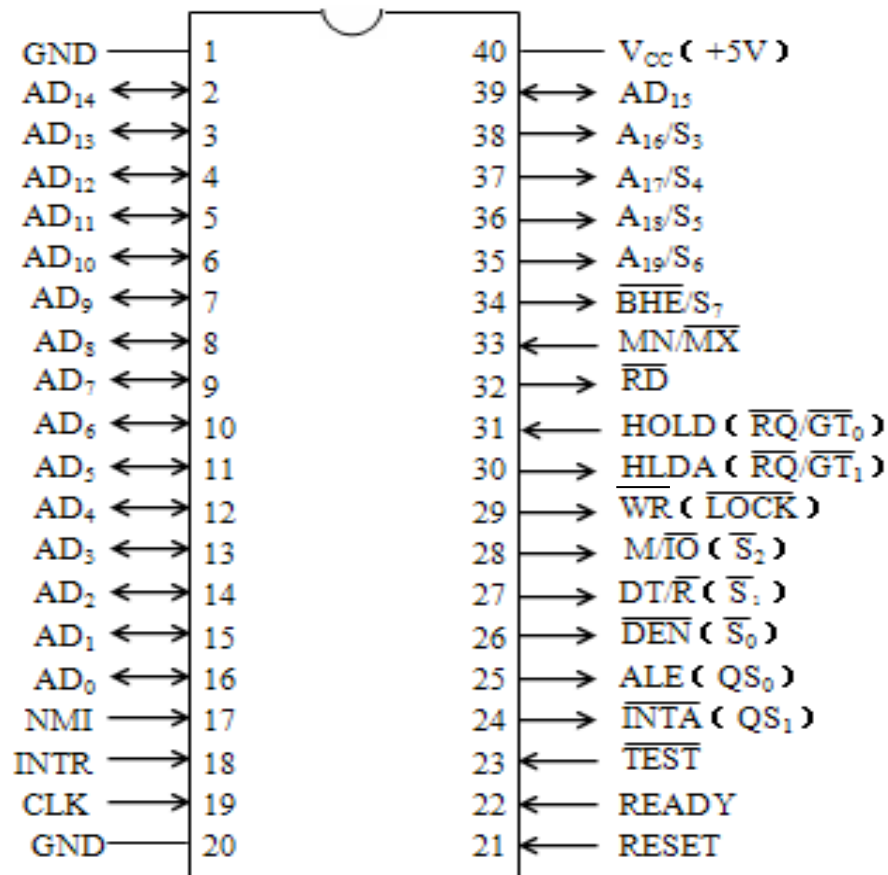
最大模式：系统中总是包含**两个或多个**微处理器，其中主处理器是8086，其它的处理器称为协处理器

□ 引脚**33**决定工作模式：

□ 两种模式下引脚**24~31**有不同的名称和意义

2.3 8086 微处理器的外部引脚及工作模式

8086微处理器是一个双列直插式、具有**40个**引脚的器件，受封装引线数目限制，涉及某些引线用来传送多于一种类型的信号



8086 CPU 引脚图

2.3 8086 微处理器的外部引脚及工作模式

1. 两种工作模式下具有相同功能的引脚

(1) **GND**、**V_{CC}**，地和电源

第1、20脚为地；第40脚为电源，8086用单一的+5V电压

(2) **AD₀~AD₁₅**(双向，三态)：地址/数据复用引脚线。

① 在每个总线周期的T₁，作地址总线低16位A₁₅~A₀，用于寻址MEM或I/O端口。

② 之后，经转换开关→数据总线D₁₅~D₀，传送数据，直到总线周期结束。

③ 在**DMA方式**时，这些引脚成**浮空**状态。

可见：复用信号使用**时间**加以划分。要求在T₁状态线出现16位地址时，用**地址锁存器**加以锁存。

2.3 8086 微处理器的外部引脚及工作模式

(3) $A_{19}/S_6—A_{16}/S_3$ 地址/状态复用引脚，输出

T_1 状态：表示最高4位地址线，需地址锁存器对高4位地址锁存。

$T_2 \sim T_4$ 状态提供状态信息。其中：

S_6 ： $S_6=0$ 指示8086当前与总线相连，在 $T_2 \sim T_4$ 状态， $S_6=0$ （低）

S_5 ：中断允许标志IF的状态， $S_5=1$ 表示允许可屏蔽中断请求；

$S_5=0$ 禁止一切可屏蔽中断。

S_4 和 S_3 ：组合起来指出当前正在使用的段寄存器，具体规定

S_4 和 S_3 的代码组合及对应段寄存器情况表

S_4	S_3	当前正在使用的段寄存器
0	0	附加段寄存器 ES
0	1	堆栈段寄存器 SS
1	0	对存储器寻址时使用代码段寄存器 CS，对 I/O 或中断向量寻址时未使用任何段寄存器
1	1	数据段寄存器 DS

2.3 8086 微处理器的外部引脚及工作模式

(4) NMI 非屏蔽中断输入引脚

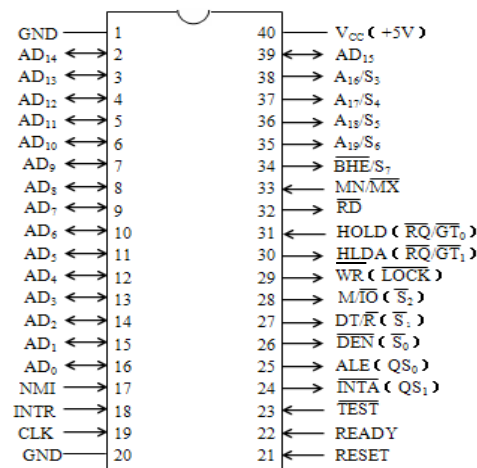
不受IF影响，也不能用软件屏蔽。每当NMI端进入一个正沿触发信号时，CPU就会在结束当前指令后执行对应于中断类型为2的非屏蔽中断处理程序。

(5) INTR 可屏蔽中断请求信号，输入

CPU在执行每条指令的最后一个时钟周期后对该信号进行采样，IF=1且接收到INTR信号，CPU就会在结束当前指令后，响应中断请求，执行中断处理程序。

(6) \overline{RD} 读信号，输出

与 $\overline{M/\overline{IO}}$ 联合，决定是对内存或I/O端口的读操作。



8086 CPU 引脚图

2.3 8086 微处理器的外部引脚及工作模式

(10) $\overline{\text{TEST}}$ 测试信号输入

与指令**WAIT**结合起来使用，在**CPU**执行**WAIT**指令时，**CPU**处于空转状态进行等待；当**TEST**有效时，等待状态结束。

(11) $\overline{\text{MN}}/\overline{\text{MX}}$ 最小/最大组态控制信号输入

固定接**+5v**，**CPU**处于**最小模式**；接**地**，**CPU**处于**最大模式**。

说明：以上是**8086**工作在最小/最大模式时都用的。

第**24—31**脚**8**个信号，在最小模式和最大模式下有不同的名称和定义。

2.3 8086 微处理器的外部引脚及工作模式

2、最小模式下的引脚信号

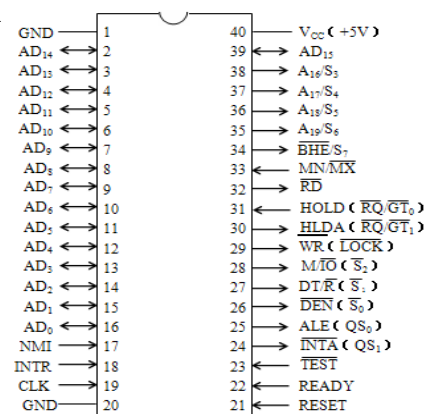
(1) $\overline{\text{INTA}}$ 中断响应信号，输出

是对外设中断请求的响应信号。中断响应周期由两个连续的总线周期组成，在每个响应周期的 T_2 ， T_3 和 T_w 状态， $\overline{\text{INTA}}$ 均有效；在第二个中断响应周期，外设端口往数据总线上发送中断类型号，CPU根据中断向量而转向中断处理程序。

(2) ALE 地址锁存允许信号，输出，高电平有效
 T_1 状态，当 $\text{AD}_{15} \sim \text{AD}_0$ 和 $\text{A}_{19}/\text{A}_6 \sim \text{A}_{16}/\text{S}_3$ 上出现地址信号时， ALE 有效，将地址信息锁存到地址锁存器中。

(3) $\overline{\text{DEN}}$ 数据允许信号

在用8286/8287作为数据总线收发器时， $\overline{\text{DEN}}$ 为收发器提供一个控制信号，表示CPU当前准备发送或接收一个数据。



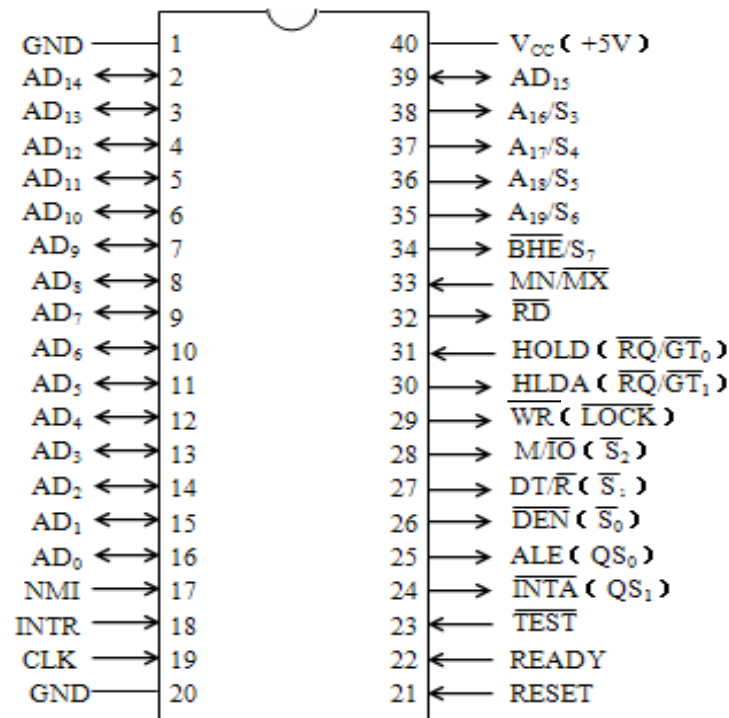
8086 CPU 引脚图

2.3 8086 微处理器的外部引脚及工作模式

(4) $\overline{\text{DT/R}}$ 数据收/发信号，输出
 DT/R 用于控制8286/8287的数据传送方向。 DT/R 为高电平表示数据发送； DT/R 为低表示数据接收。DMA方式时， DT/R 为高阻态。

(5) $\text{M}/\overline{\text{IO}}$ 存储器/输入输出控制信号，输出
决定CPU是对存储器还是输入/输出访问的控制信号。高电平表示与存储器之间；低电平表示与输入/输出端口之间。

(6) $\overline{\text{WR}}$ 写信号，输出
与 $\text{M}/\overline{\text{IO}}$ 信号联合，表示CPU当前正在进行的是存储器还是I/O的写操作。



8086 CPU 引脚图

2.3 8086 微处理器的外部引脚及工作模式

(7) **HOLD** 保持请求信号，输入，高电平有效。

当DMA操作或外部处理器要求通过总线传送数据时，**HOLD**信号为高，表示外界请求主CPU让出对总线的控制权。

(8) **HLDA** 保持响应信号，输出，高电平有效。

①当CPU同意让出总线控制权时，输出**HLDA**高电平信号，通知外界可以使用总线。同时，现有主CPU所有具“三态”的线，都进入浮空状态；

②当**HOLD**变为低电平时，主CPU也把**HLDA**变为低电平，此时它又重新获得总线控制权。

2.3 8086 微处理器的外部引脚及工作模式

3、最大模式下的引脚信号

(1) $\overline{S_2}$ 、 $\overline{S_1}$ 、 $\overline{S_0}$ 总线周期状态信号（输出，低电平有效）

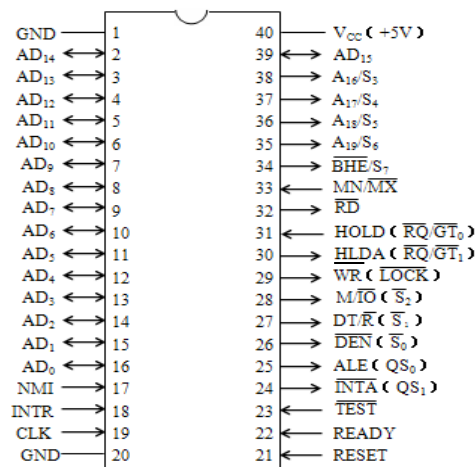
总线周期状态信号输出，分别对应最小方式的**DEN, DT/R, M/IO**

①多CPU模式下，它们的不同组合，表示CPU总线周期的状态。

② 此组信号→ 8288总线控制器，8288利用它们的不同组合，产生访问MEM或I/O的控制信号或中断响应信号。

$\overline{S_2}$ 、 $\overline{S_1}$ 、 $\overline{S_0}$ 组合产生的总线控制功能

$\overline{S_2}$ $\overline{S_1}$ $\overline{S_0}$	操作过程	$\overline{S_2}$ $\overline{S_1}$ $\overline{S_0}$	操作过程
0 0 0	发中断响应信号	1 0 0	取指令
0 0 1	读 I/O 端口	1 0 1	读内存
0 1 0	写 I/O 端口	1 1 0	写内存
0 1 1	暂停	1 1 1	无源状态



8086 CPU 引脚图

2.3 8086 微处理器的外部引脚及工作模式

四、8086微处理器两种模式下的系统结构

CPU的 $\overline{\text{MN}}/\overline{\text{MX}}$ 引脚接+5V时，8086工作于最小模式。

特点：所有总线控制信号直接由8086产生，系统中的总线控制逻辑电路被减到最少

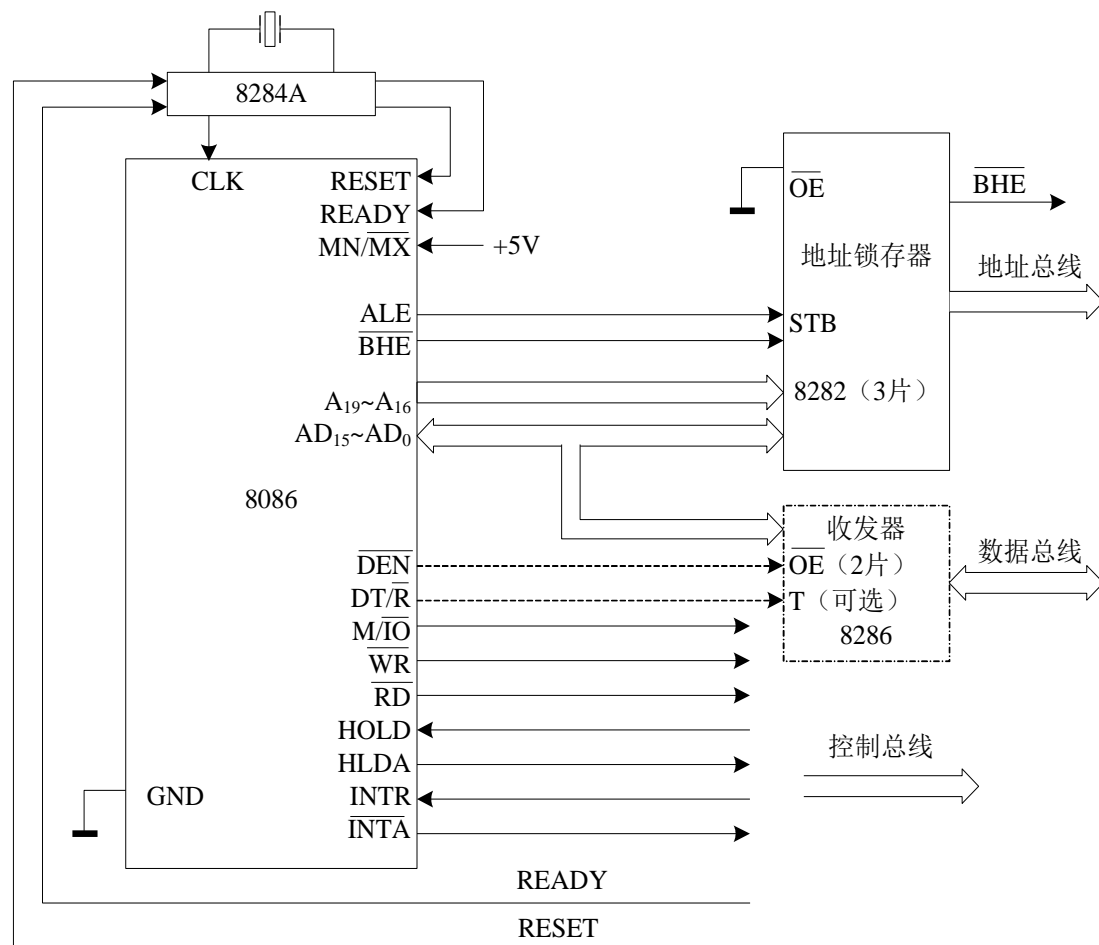


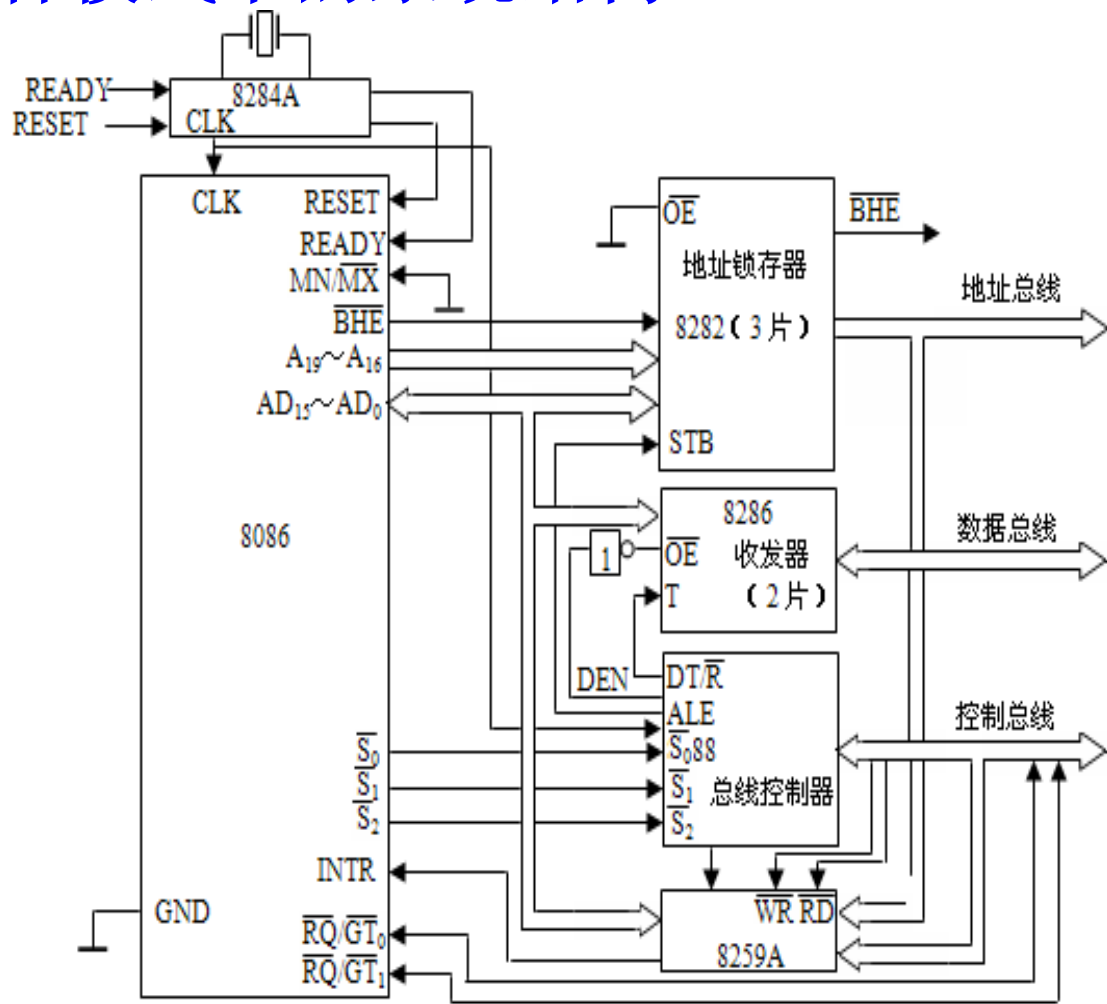
图2.11 8086最小工作模式的典型配置

2.3 8086 微处理器的外部引脚及工作模式

四、8086微处理器两种模式下的系统结构

CPU的 $\overline{\text{MN}}/\overline{\text{MX}}$ 引脚接地时，8086工作于最大模式。

特点：系统的许多控制信号由总线控制器8288对8086发出的控制信号进行变换和组合，从而得到各种系统控制信号





2.5 Intel的其他微处理器

一、Intel 80486 微处理器

80486是Intel的第二款**32**位微处理器，是为了支持多处理机系统而设计的

1、80486 CPU的主要特点

- (1) **80486 CPU**具有保护、存储器管理、任务转换、分页功能、片内高速缓存器和浮点数运算部件；
- (2) 能运行**Windows、DOS、OS/2**和**UNIX V/386**等操作系统
- (3) 具有完整的**RISC**内核，使得常用指令的执行时间都只需要一个时钟周期
- (4) 采用**8KB**统一的代码**Cache**和数据**Cache**
- (5) 内部的自测试功能包括执行代码和访问数据时的断点陷阱

2.5 Intel的其他微处理器

2、80486 CPU内部结构

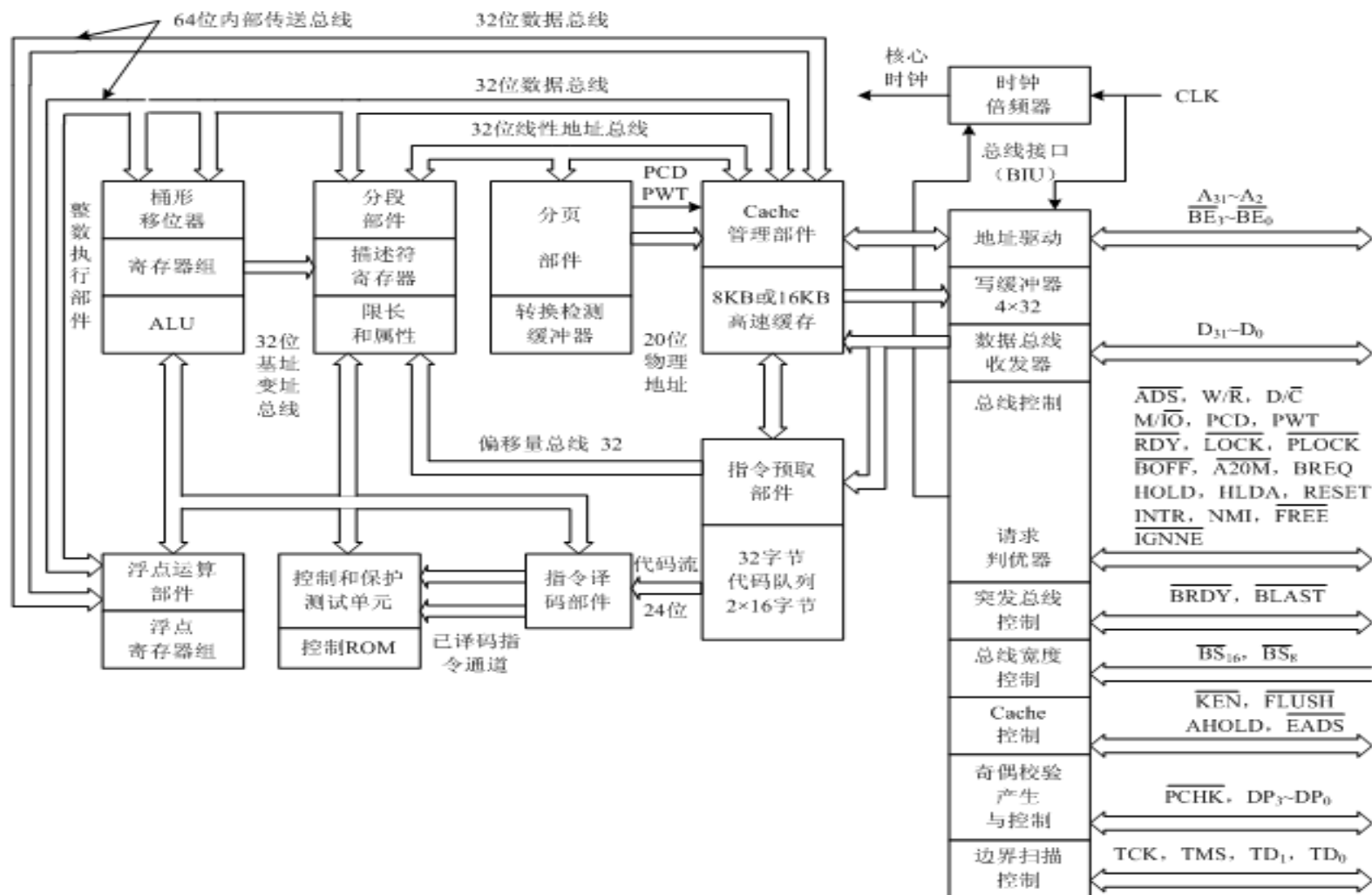


图2.19 80486微处理器内部结构

2.5 Intel的其他微处理器

2、80486 CPU内部结构

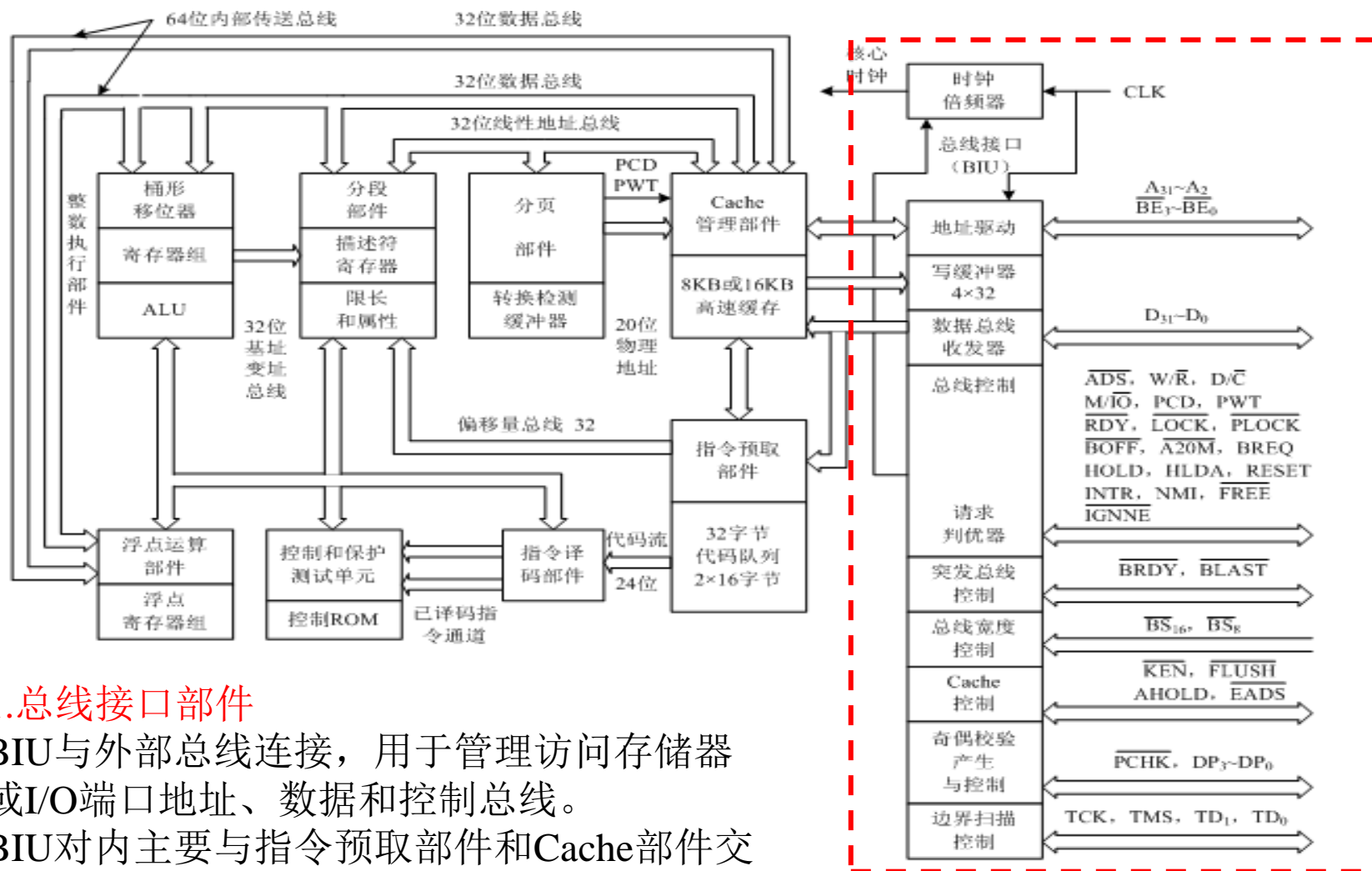


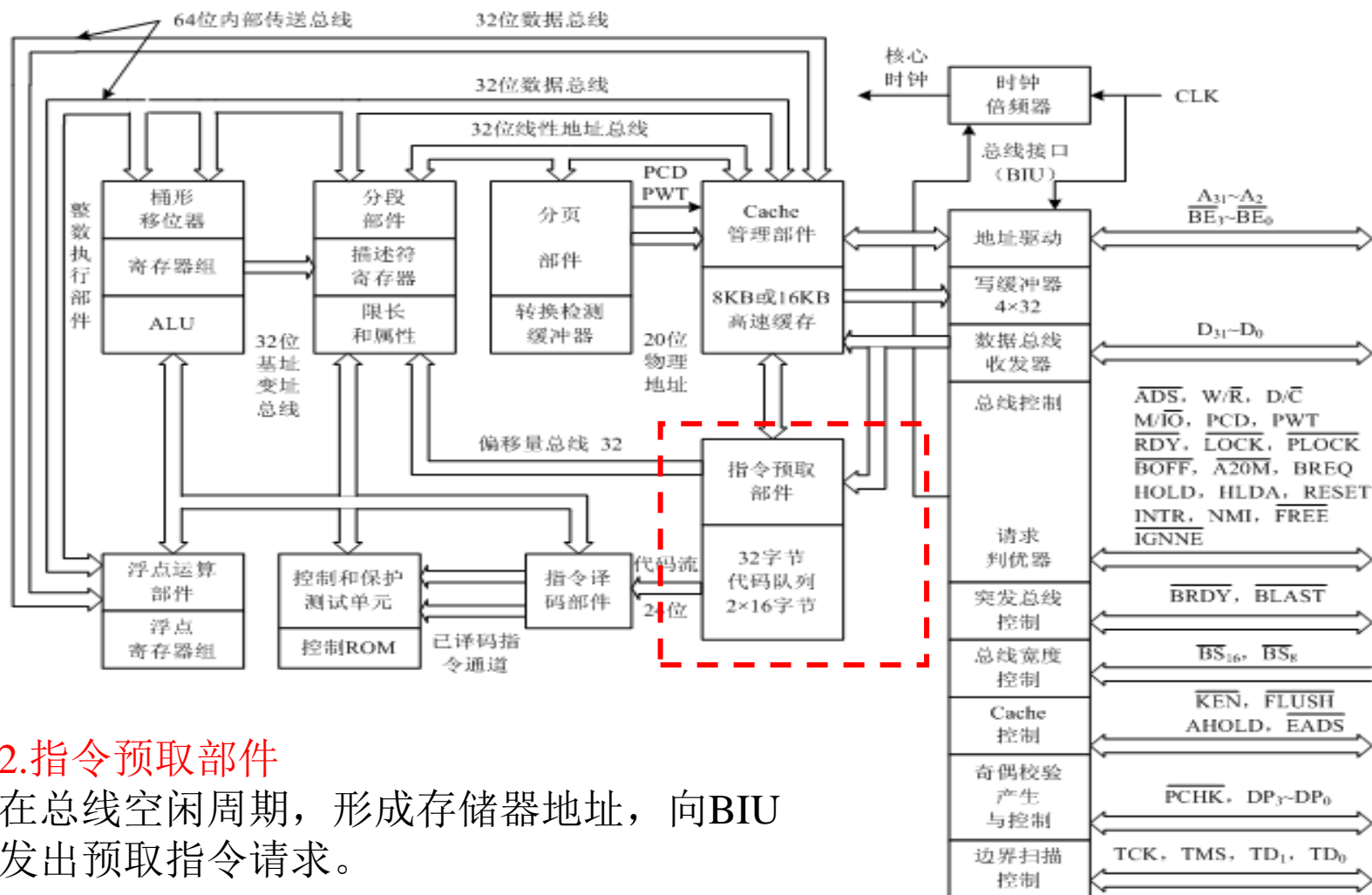
图2.19 80486微处理器内部结构

1.总线接口部件

BIU与外部总线连接，用于管理访问存储器或I/O端口地址、数据和控制总线。
BIU对内主要与指令预取部件和Cache部件交换信息。

2.5 Intel的其他微处理器

2、80486 CPU内部结构



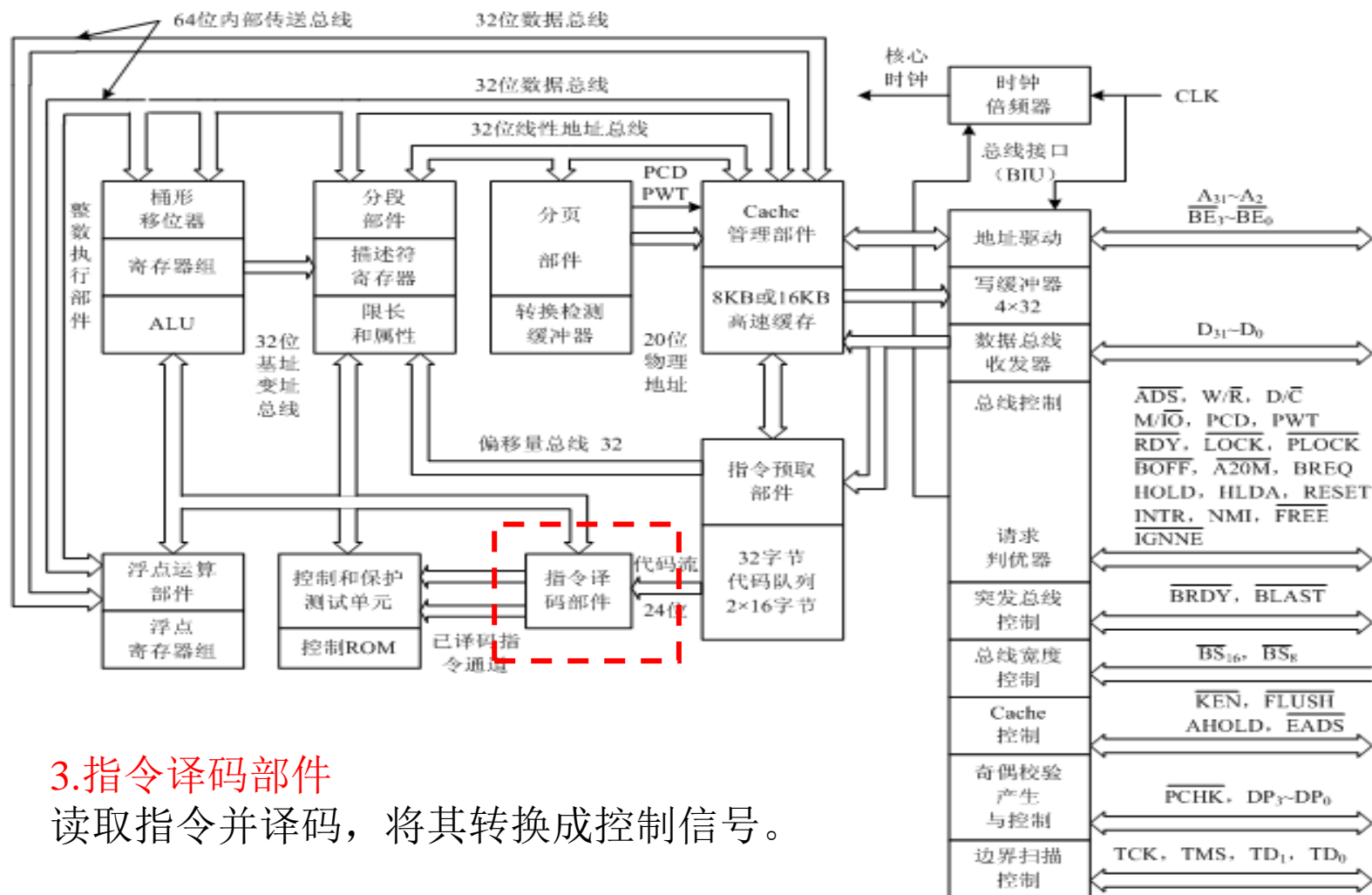
2.指令预取部件

在总线空闲周期，形成存储器地址，向BIU发出预取指令请求。

图2.19 80486微处理器内部结构

2.5 Intel的其他微处理器

2、80486 CPU内部结构



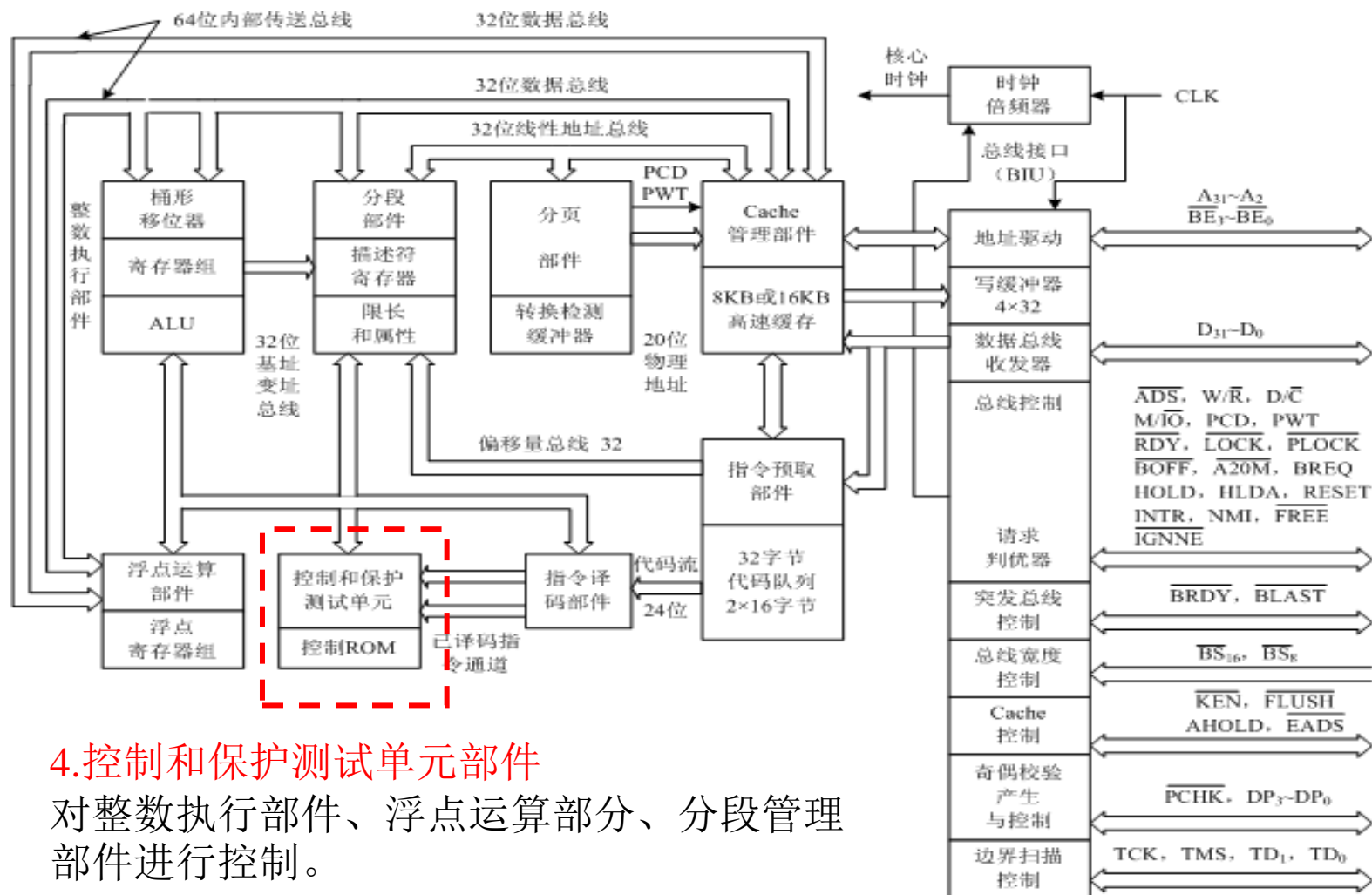
3.指令译码部件

读取指令并译码，将其转换成控制信号。

图2.19 80486微处理器内部结构

2.5 Intel的其他微处理器

2、80486 CPU内部结构



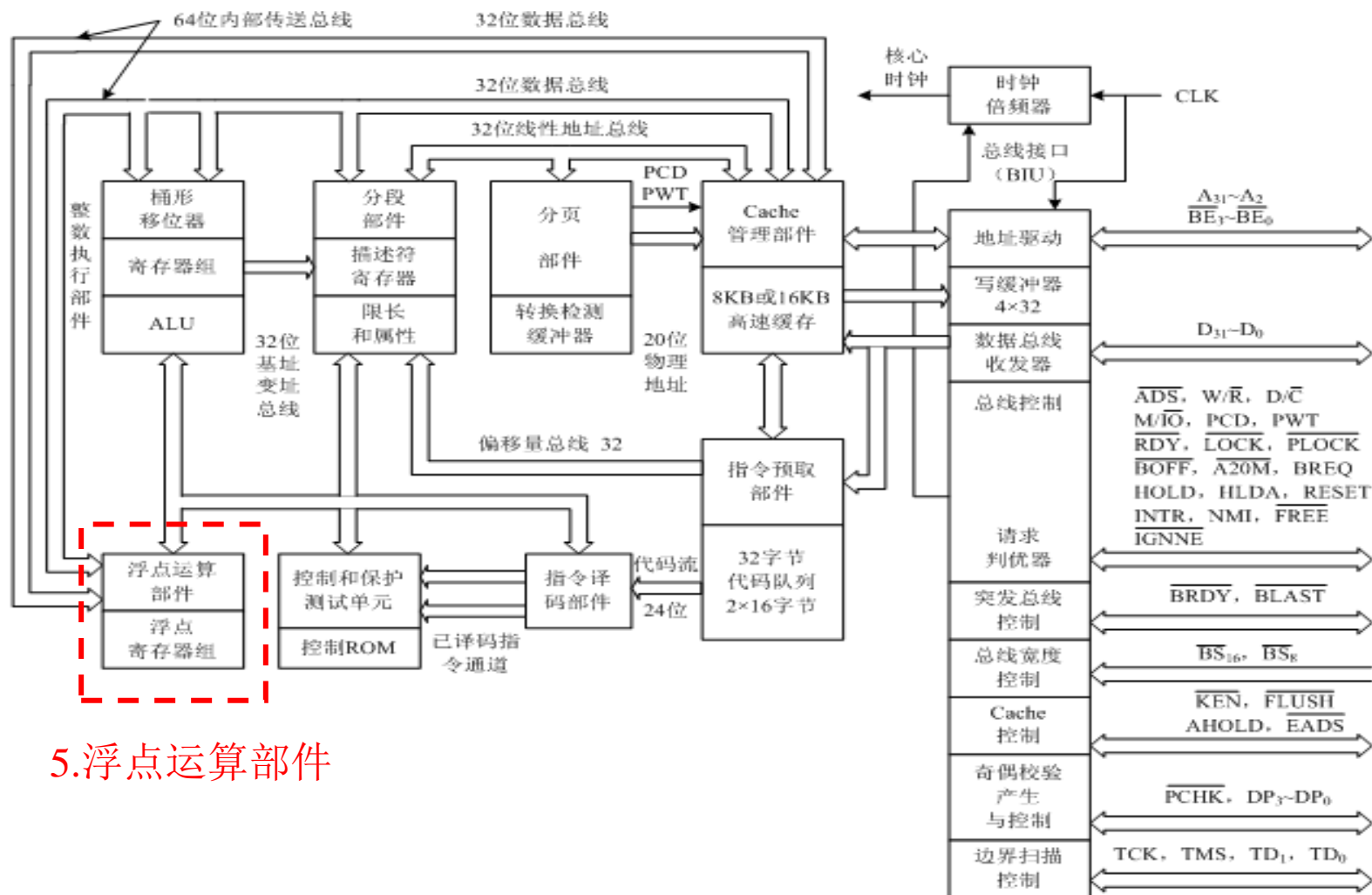
4.控制和保护测试单元部件

对整数执行部件、浮点运算部分、分段管理部件进行控制。

图2.19 80486微处理器内部结构

2.5 Intel的其他微处理器

2、80486 CPU内部结构

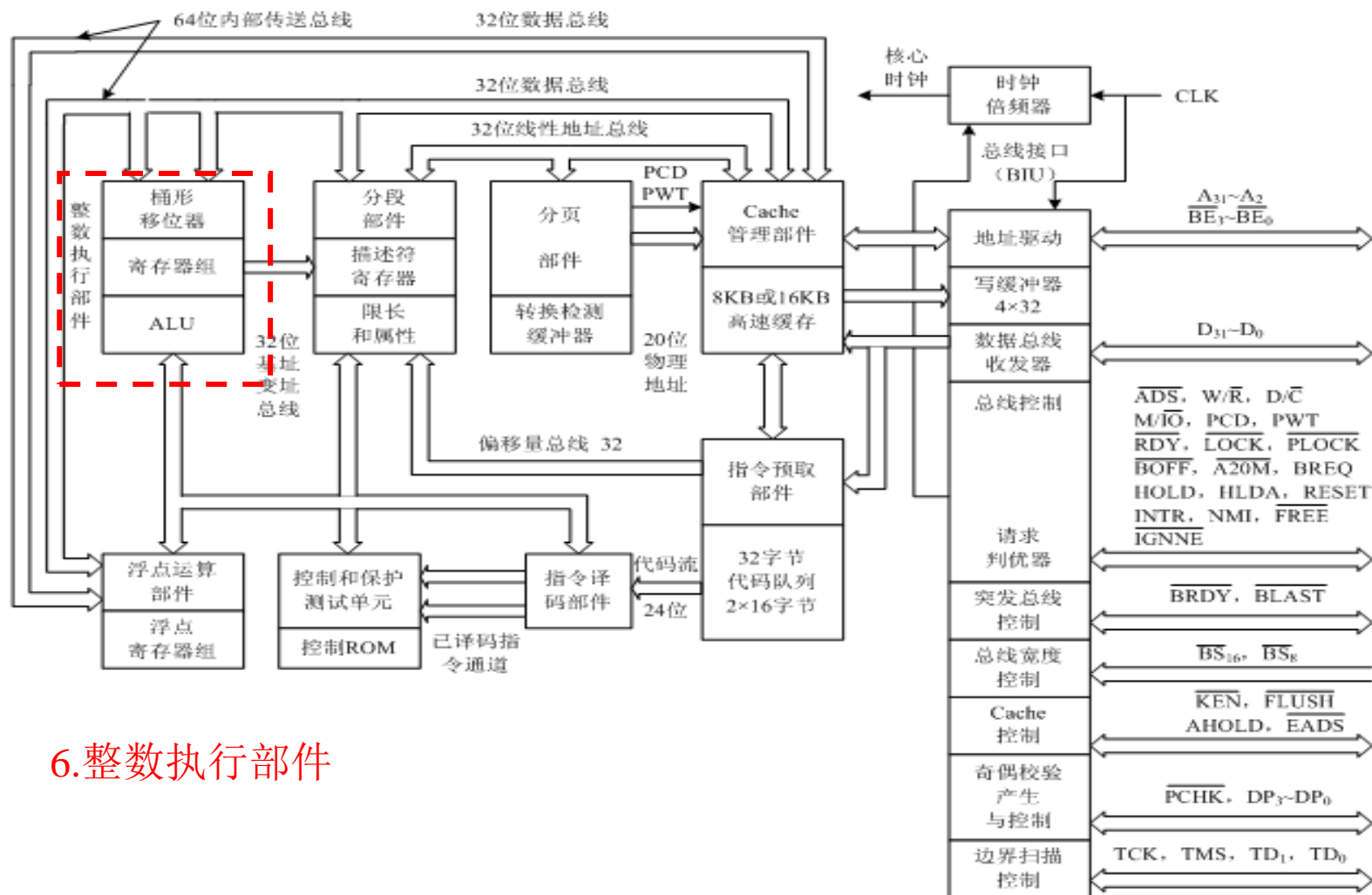


5.浮点运算部件

图2.19 80486微处理器内部结构

2.5 Intel的其他微处理器

2、80486 CPU内部结构

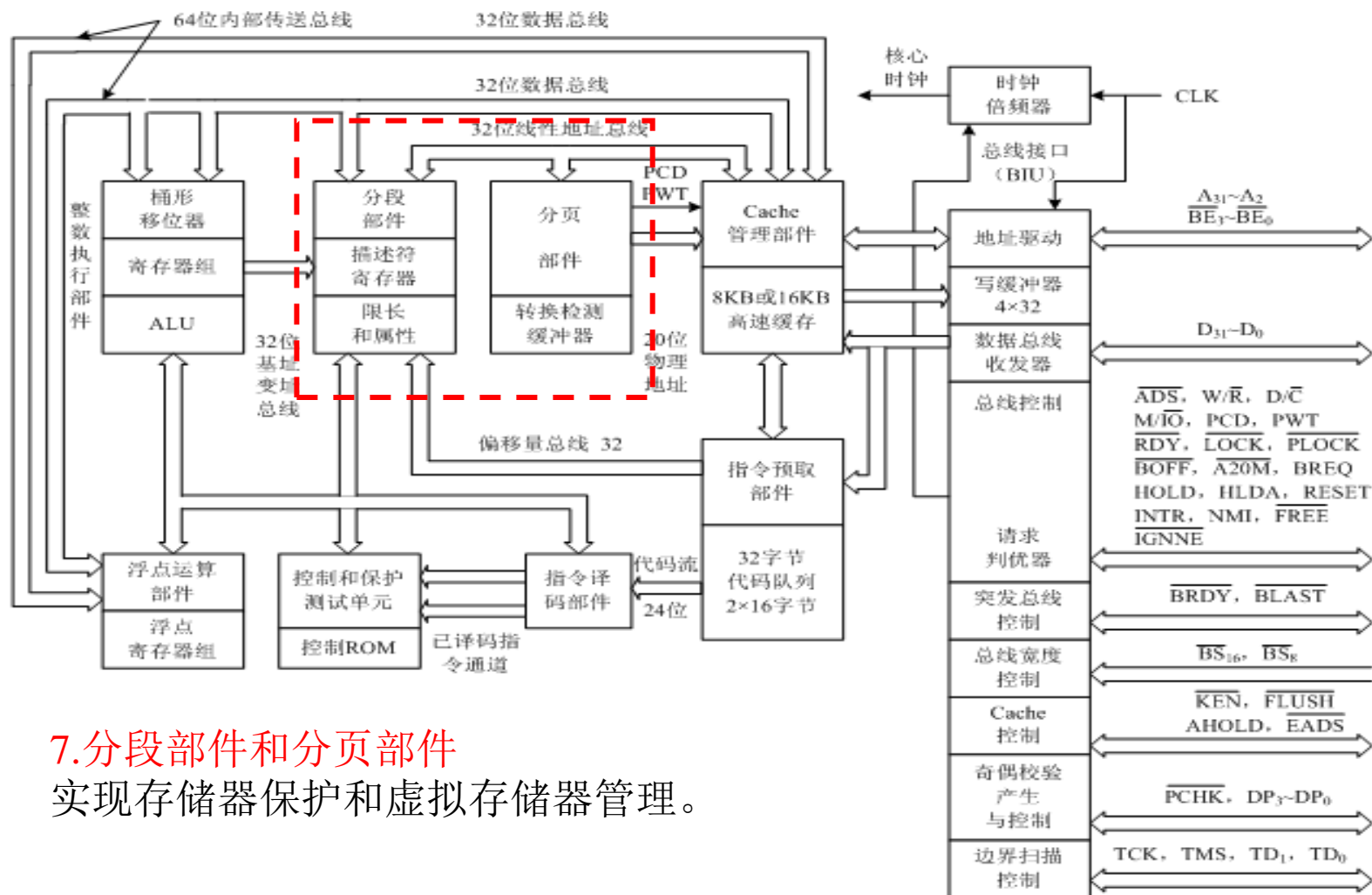


6. 整数执行部件

图2.19 80486微处理器内部结构

2.5 Intel的其他微处理器

2、80486 CPU内部结构



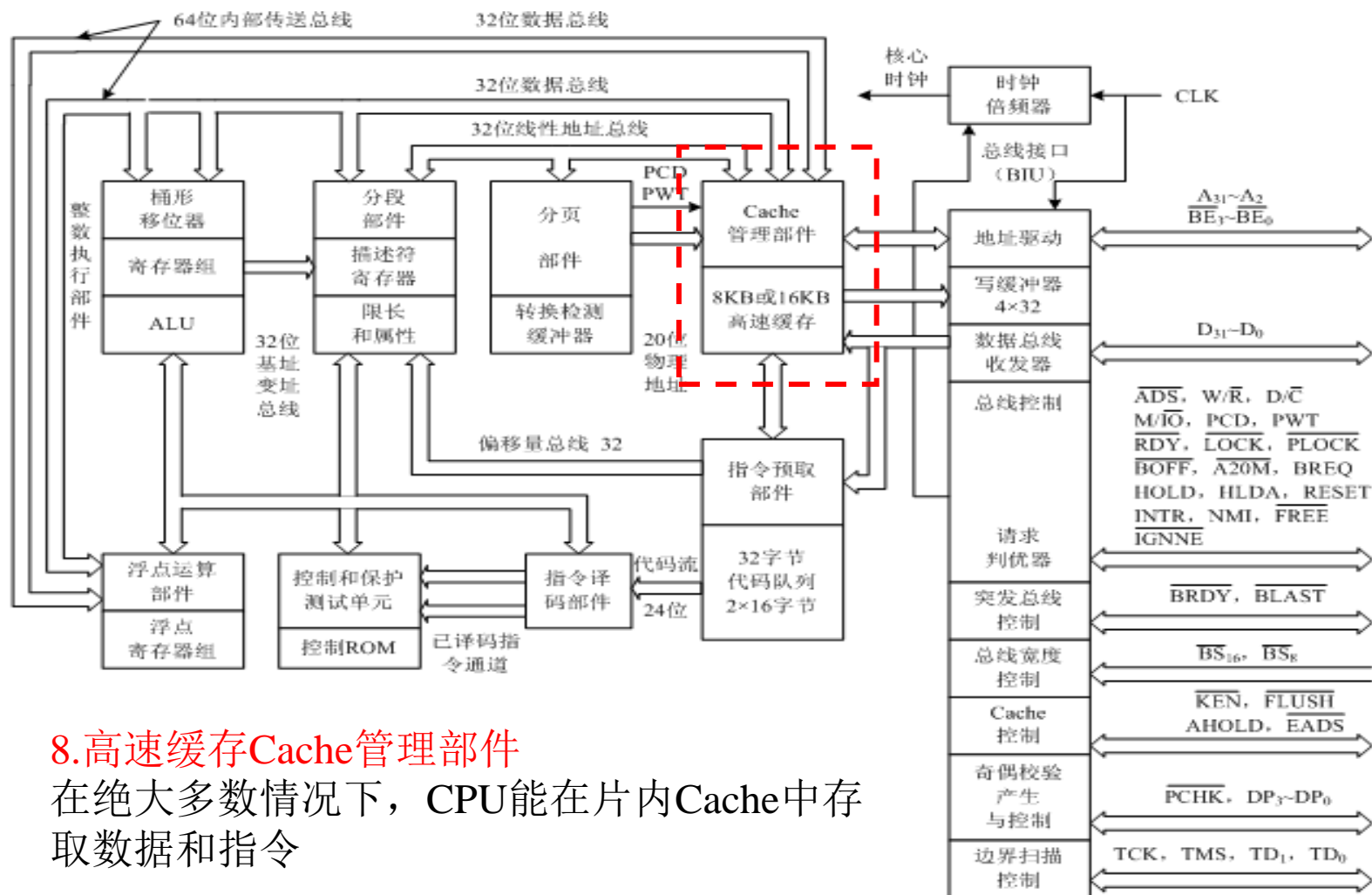
7.分段部件和分页部件

实现存储器保护和虚拟存储器管理。

图2.19 80486微处理器内部结构

2.5 Intel的其他微处理器

2、80486 CPU内部结构



8.高速缓存Cache管理部件

在绝大多数情况下，CPU能在片内Cache中存取数据和指令

图2.19 80486微处理器内部结构

2.5 Intel的其他微处理器

3、80486 CPU的寄存器结构

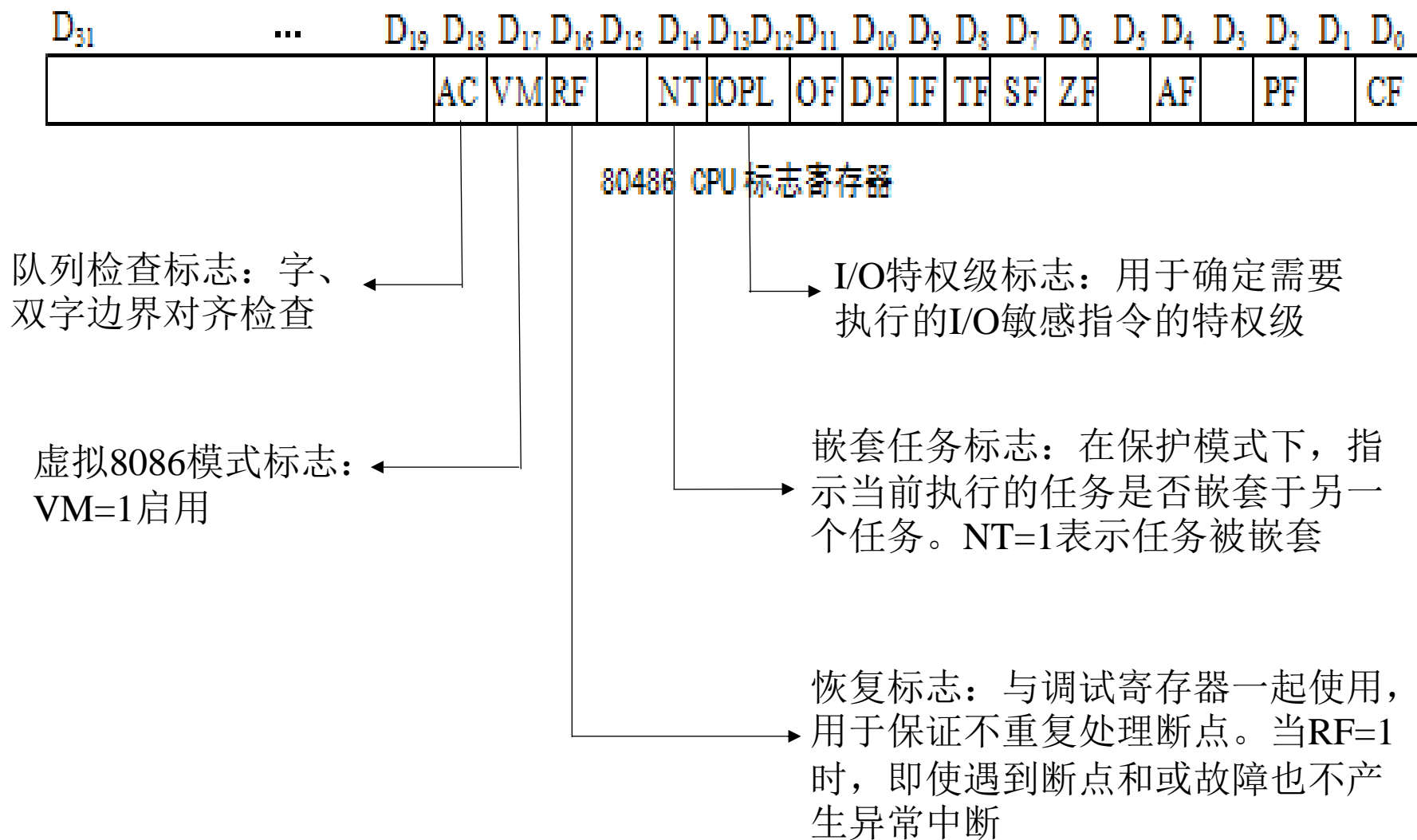
1) 基本寄存器



80486 CPU 的基本寄存器组

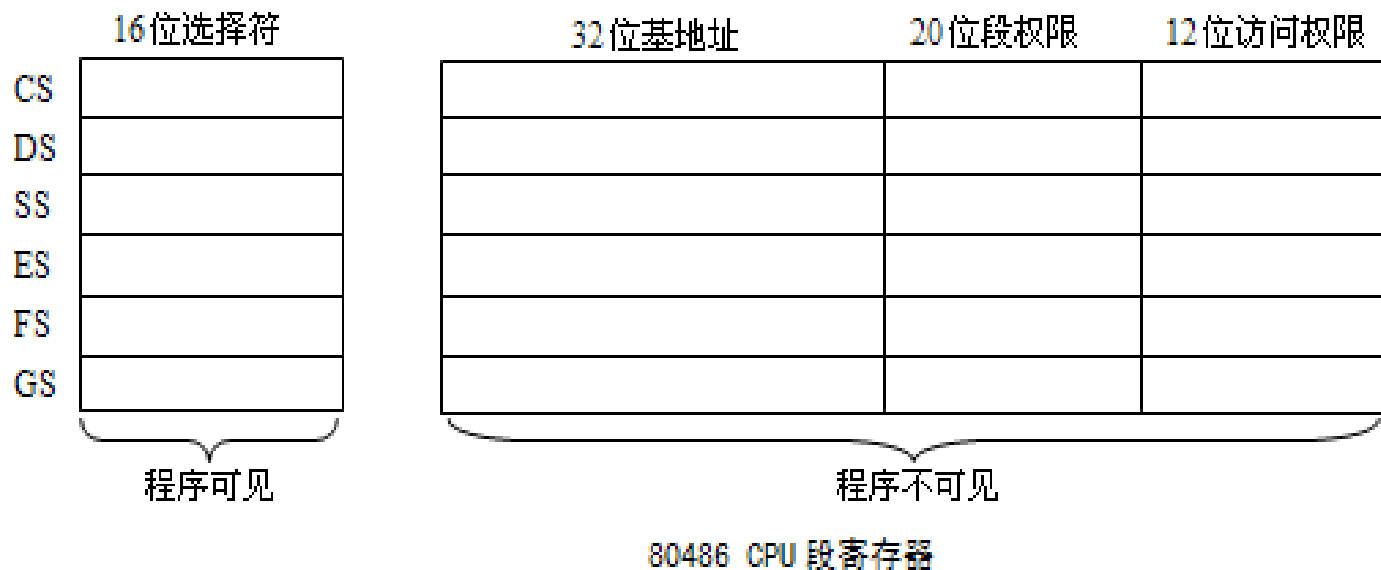


2.5 Intel的其他微处理器





2.5 Intel的其他微处理器



- ◆ 6个16位段寄存器存放段地址（实地址模式）或选择符（保护模式）
- ◆ 在保护模式下每个段寄存器都有一个相对应的段描述符



2.5 Intel的其他微处理器

2) 系统寄存器-系统地址寄存器

	16位选择符	32位基地址	16位段限	16位访问权限
TR	段选择符	基地址	段限	属性
LDTR	段选择符	基地址	段限	属性

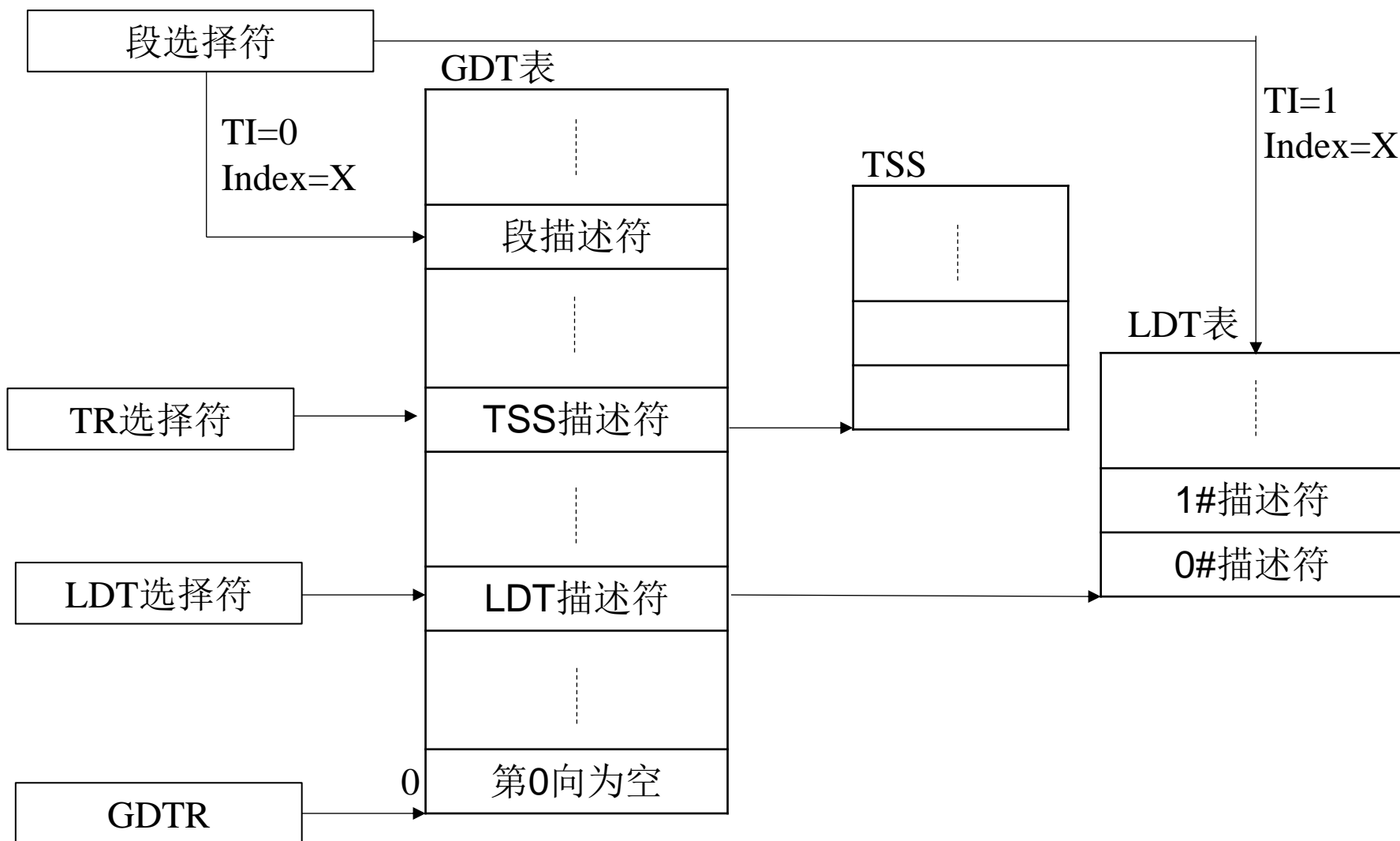
	32位基地址	16位段限
GDTR	基地址	段限
IDTR	基地址	段限

80486 CPU 系统地址寄存器

- ◆ 全局描述符表寄存器GDTR：全局描述符表GDT中包含了操作系统使用的、各任务公用的描述符。GDTR中含有GDT的32位线性基址和16位限长。
- ◆ 中断描述符表寄存器IDTR：所有中断描述符集中放在中断描述符表IDT中。IDTR中含有IDT的基址和限长。
- ◆ 局部描述符表寄存器LDTR：局部描述符表LDT包含了某一任务专用的描述符。LDTR中存放当前任务的LDT选择符，对应的LDT描述符防止LDT描述符寄存器中
- ◆ 任务寄存器TR：每一个任务都有一个任务状态段TSS。当前任务的TSS选择符存放在TR中，TSS描述符存放在TSS描述符寄存器中



2.5 Intel的其他微处理器





2.5 Intel的其他微处理器

- 2)系统寄存器-控制寄存器
- 3)调试寄存器和测试寄存器
- 4)浮点寄存器

2.5 Intel的其他微处理器

4、80486 CPU主要引脚信号

80486 CPU采用**PGA**封装形式，共有**168**个引脚信号，其中包括**30**个地址引脚信号、**32**个数据引脚信号、**35**个控制引脚信号、**24**个**VCC**引脚信号、**28**个**VSS**引脚信号和**19**个空脚

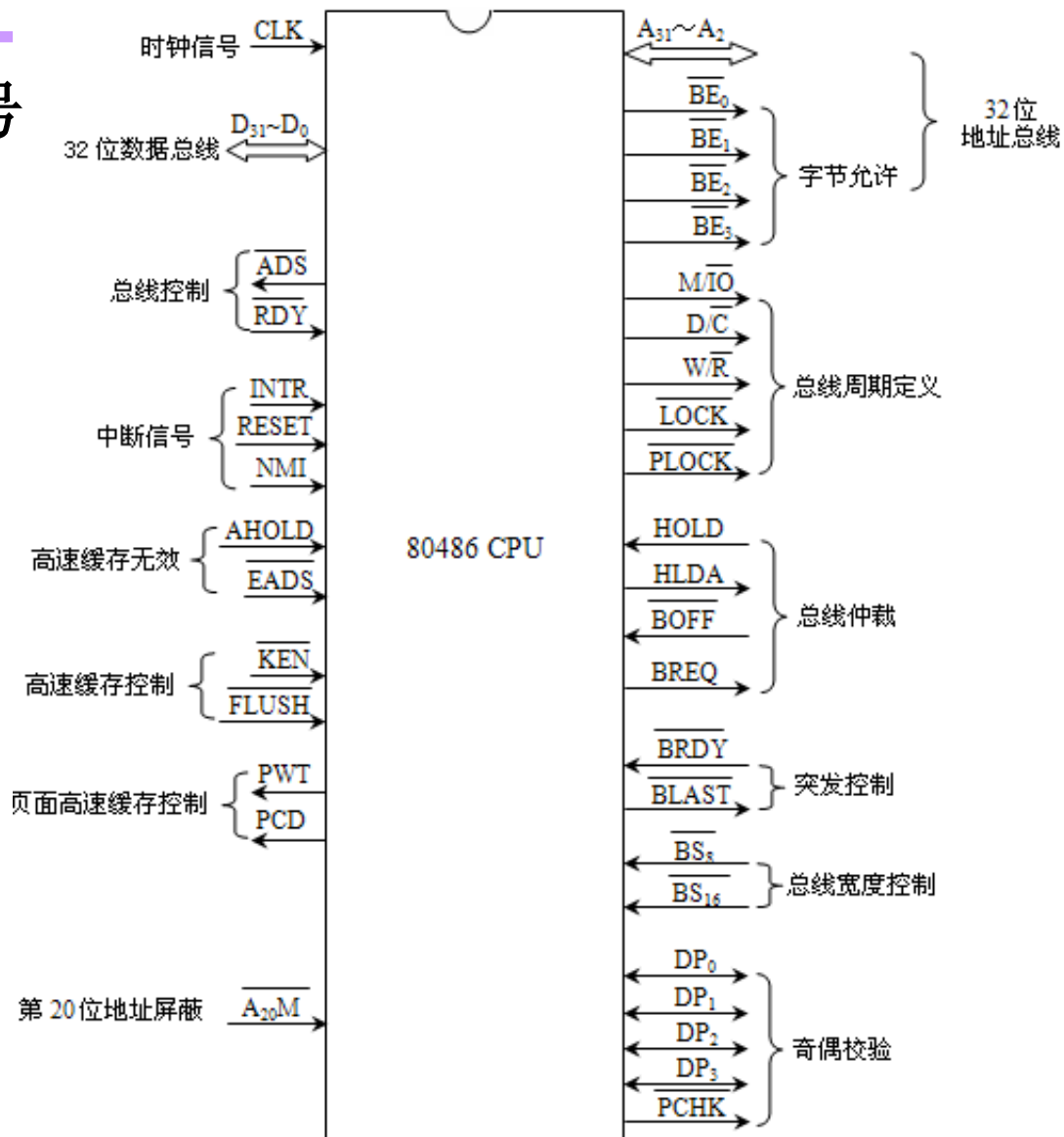
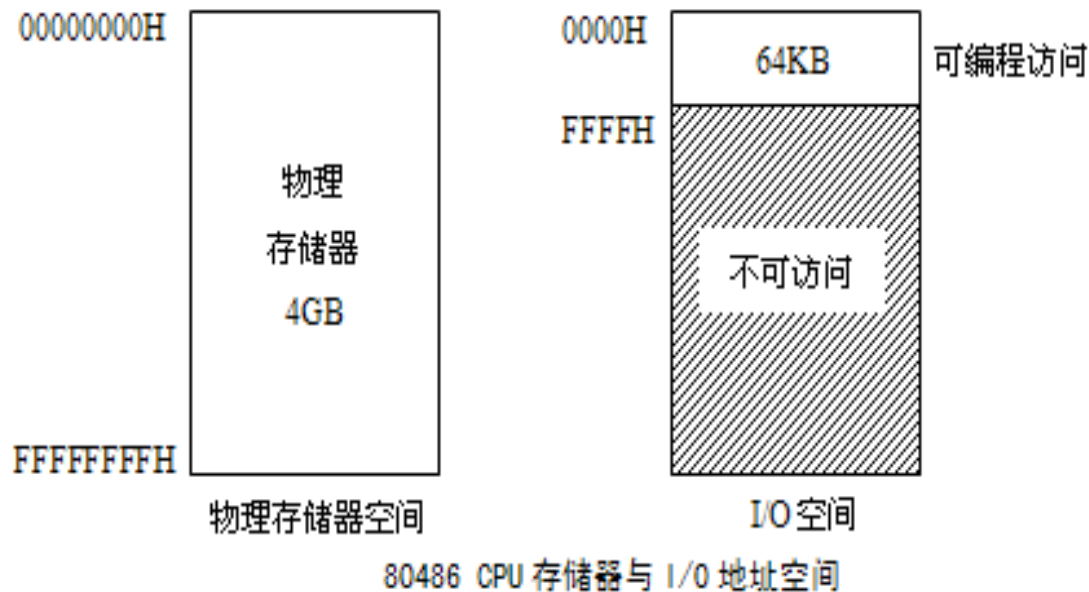


图 2.24 80486 CPU 引脚信号

2.5 Intel的其他微处理器

5、80486 CPU的存储器组织及I/O结构

80486 CPU有32条地址总线，可寻址 $2^{32}=4\text{GB}$ 的存储器空间，地址范围是00000000H~FFFFFFFFH。32条地址总线中的低16位地址用作对64KB I/O端口的寻址，地址范围是0000H~FFFFH

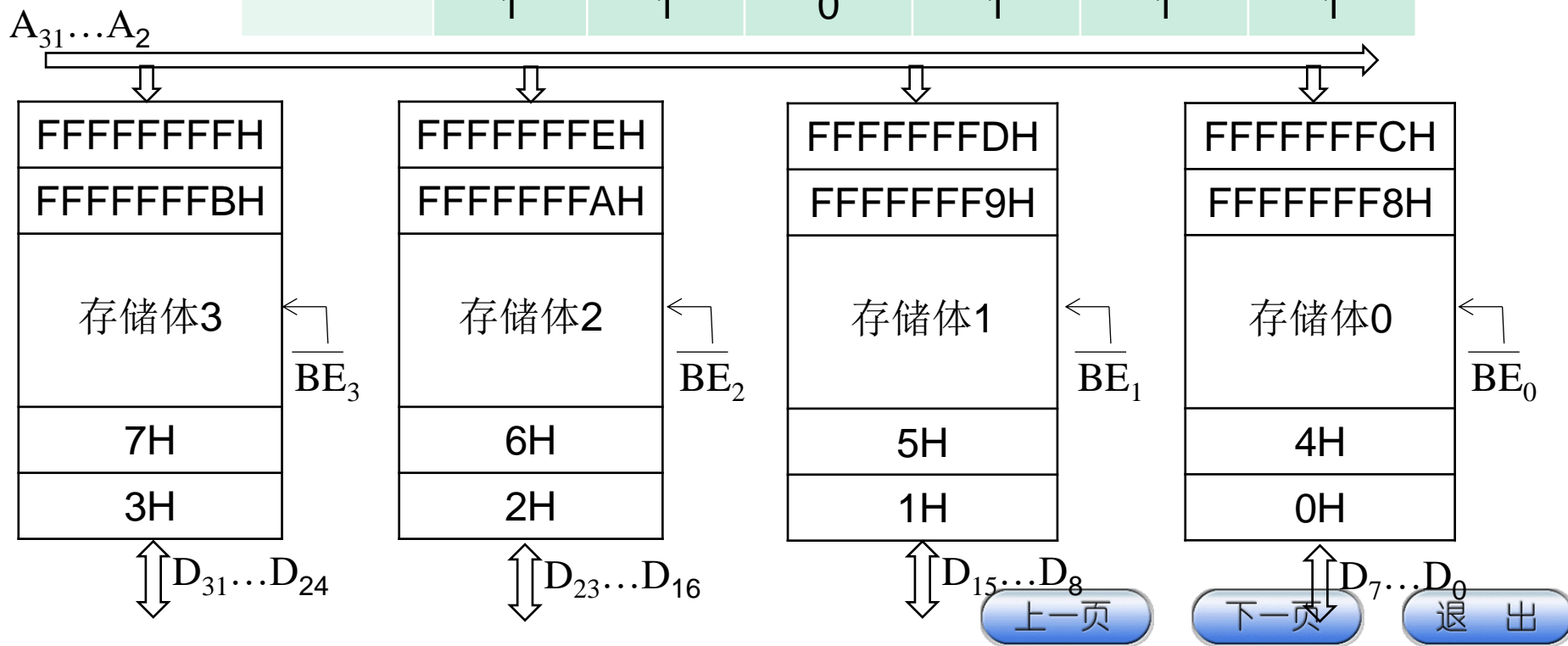




2.5 Intel的其他微处理器

存储器寻址

物理地址			输出信号			
$A_{31} \dots A_2$	A_1	A_0	\overline{BE}_3	\overline{BE}_2	\overline{BE}_1	\overline{BE}_0
$A_{31} \dots A_2$	0	0	X	X	X	0
	0	1	X	X	0	1
	1	0	X	0	1	1
	1	1	0	1	1	1





2.5 Intel的其他微处理器

6、80486 CPU的工作模式

(1) 实地址模式

实地址模式是最基本的工作方式，与**16位微处理器8086**的实地址模式保持兼容

(2) 保护模式

保护模式是在程序运行过程中，为了防止以下情况的发生而引入的。

- ① 应用程序破坏系统程序；
- ② 某一应用程序破坏了其它应用程序；
- ③ 错误地把数据当作程序运行

(3) 虚拟8086模式

虚拟**8086**模式是一种既有保护功能又能执行**16位微处理器**软件的工作方式，其工作原理与保护模式相同，但程序指定的逻辑地址与**8086 CPU**相同，可以看做是保护模式的一种子方式