

汇编语言与微机原理



第二章 微处理器及其架构

2.1 微处理器概述

2.2 微处理器内部结构

2.3 微处理器寄存器结构及数据类型

2.4 微处理器主要引脚信号与总线时序

2.5 实模式存储器管理

2.6 本章小结



2.1 微处理器概述

➤ IA-32 (Intel Architectures)

字长为32位的微处理器体系架构统称为IA-32，包括：80386、80486、Pentium、MMX Pentium、Pentium Pro、Pentium II、Pentium III及Pentium 4。为了保持兼容，有时也将8086/8088以及80286作为IA-32的特殊形式看待。

➤ IA-32E (Intel 64)

为了支持64位操作，Intel采用EM64T技术（来源于AMD）引入增强型的32位处理器体系结构，成为IA-32E（支持64位地址空间，又称为Intel 64。包括以Core架构为核心的处理器（最新的酷睿i7处理器也是该架构）。

➤ IA-64

Intel公司的真正64位架构（与HP共同开发），不兼容IA-32E，于IA-32完全不兼容，包括安腾系列Itanium和Itanium 2等。



历史性一刻！AMD CPU在国内终于超越Intel

2018-11-17 00:19:52 来源: 快科技

 举报

 1984



易信



微信



QQ空间



微博



更多

AMD Ryzen锐龙这两年的表现着实亮眼，Intel这边则是多年挤牙膏之后终于尝到了苦果，偏偏又赶上14nm工艺产能吃紧，局面还是很紧张的。

市调机构Mercury Research最近发布的数据显示，2018年第三季度的x86处理器市场上，AMD已经拿到13%的份额，同比增加2.1个百分点。

虽然整体规模距离Intel依然很远，但别忘了AMD是去年才开始真正反击的，Intel可是拥有多年雄厚的群众基础。



2.1.1 微处理器的性能指标

- 字长
- 主频
- 外频
- FSB频率
- 工作电压
- 制造工艺
- 地址线宽度/数据线宽度
- 协处理器
- 流水线技术
- 超标量结构
- L1/L2/L3 Cache
- SIMD
- 核心架构
- 功耗



性能=主频*每个周期执行指令的条数

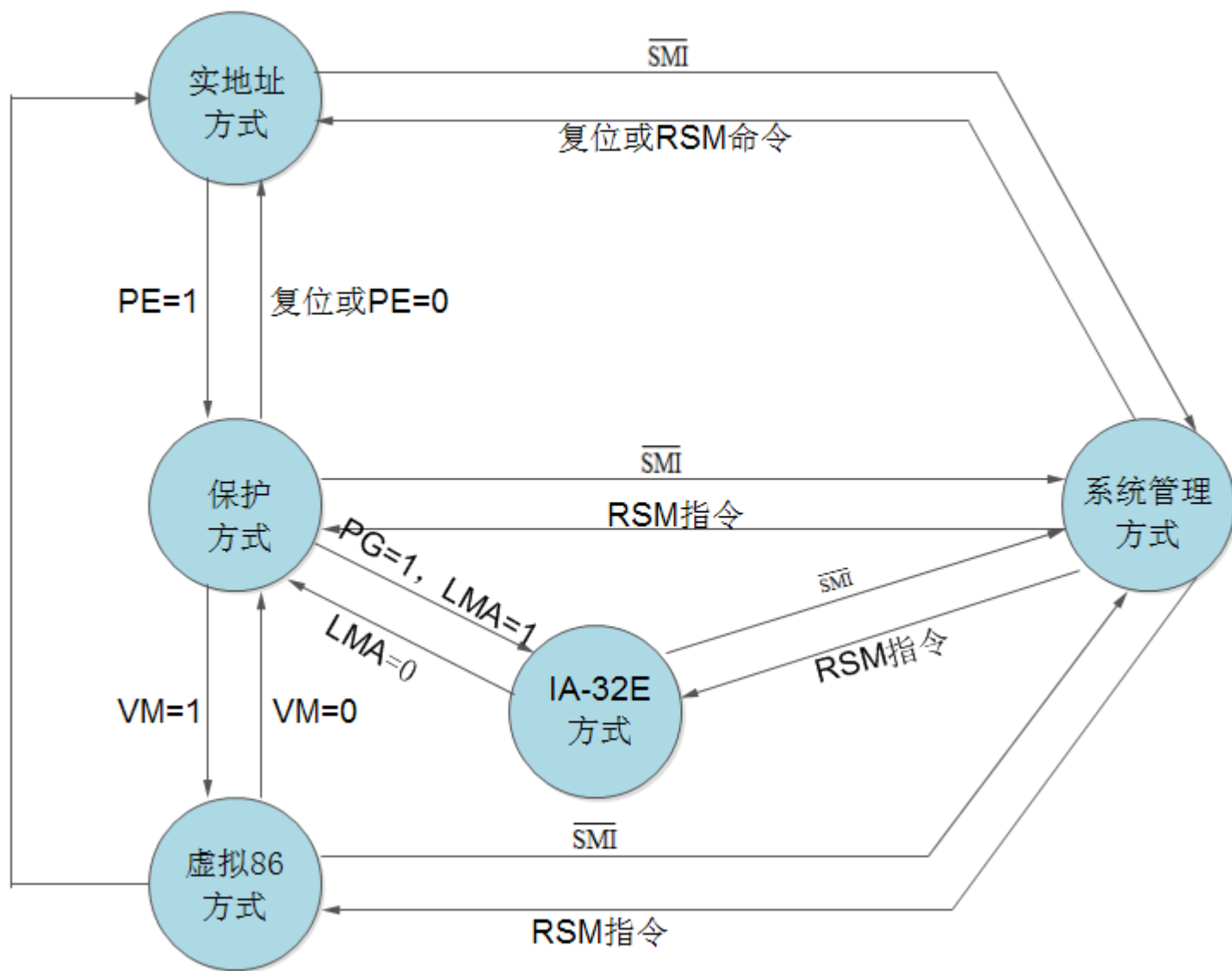
功耗=动态电容*电压*电流*频率



2.1.2 微处理器的工作方式

➤ Intel处理器5种工作方式

- 实地址方式 (Real Address Mode)
- 保护地址方式 (Protected Virtual Address Mode)
- 虚拟86方式 (Virtual 86 Mode)
- 系统管理方式 (System Management)
- IA-32E方式 (只有支持EMT64技术的处理器才有)



Intel处理器工作方式转换图



2.1.3 微处理器流水线及超标量技术

➤ 流水线技术

包括指令流水线和地址流水线技术，通常所说的流水线是**指令流水线**。

➤ 初级流水线

Intel处理器从16位的8086/8088开始就有了流水工作方式，只是非常初级的流水作业。

➤ 超级流水线

通常把具有RISC技术的5级以上流水线称为超级流水线，具有该技术的intel处理器是从80486开始的（1989年）。

在没有设计指令流水线的微处理器中，一条指令必须要等前1条指令完成了这5个步骤之后，才能进入下一条指令的第1个步骤。然而在采用指令流水线的微处理器结构中，当指令1经过取指令（取指）后，进入译码阶段的同时，指令2便可以进入取指阶段，即采取并行处理的方式。

[illegible]



- 5级指令流水线微处理器执行指令的过程

时间段	1	2	3	4	5	6	7	8	9	10	11	12
指令1	预取	译码	生成地址	执行指令	回写							
指令2		预取	译码	生成地址	执行指令	回写						
指令3			预取	译码	生成地址	执行指令	回写					
指令4				预取	译码	生成地址	执行指令	回写				
指令5					预取	译码	生成地址	执行指令	回写			
指令6						预取	译码	生成地址	执行指令	回写		
指令7							预取	译码	生成地址	执行指令	回写	
指令8								预取	译码	生成地址	执行指令	回写

采用5级指令流水线技术在10个时间段里可执行6条命令，在12个时间段里可执行8条指令，而没有采用流水线技术的10个时间段内只能执行2条指令。



➤ 超标量结构

在CPU中有一条以上的超级流水线，并且每时钟周期内可以执行一条以上的指令，这种设计技术称为超标量技术，对应处理器的结构，称为超标量结构。

处理器	80486	Pentium	Pentium Pro	Pentium II	Pentium III	Pentium 4		Core 2 Duo	Core Quad
流水线级数	5	5	12	12	12	20 (478)	31 (775)	14	14
流水线个数	1	2	3	3	3	3 (2个倍速)		4*2	4*4
单周期执行指令条数	1	2	3	3	3	5 (1+2*2)		8	16

在某个范围内，级数多效率高，但级数多带来内部结构的复杂度增加，功耗上升，效率反而降低。



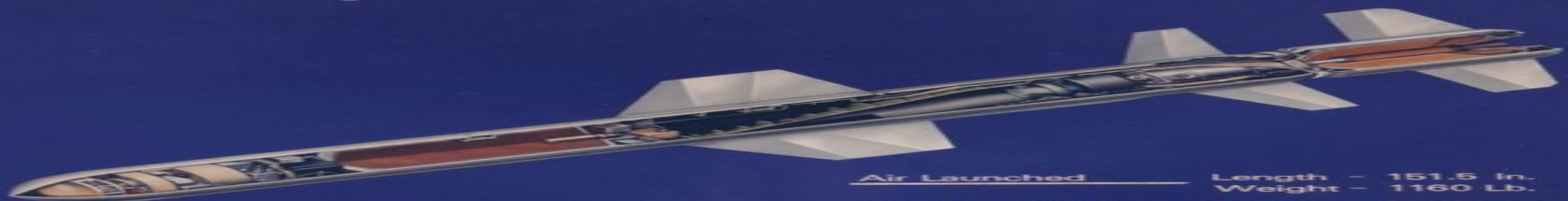
2.2 微处理器内部结构

- 80X86架构
8086/8088、80286、80386和80486
- P5架构
Pentium, Pentium MMX
- P6架构
Pentium Pro、Pentium II和Pentium III
- NetBurst架构
Pentium4
- Core架构
Pentium Dual-Core、Core2 Duo、Core2 Quad和Core2 Extreme
- Itanium架构

时间



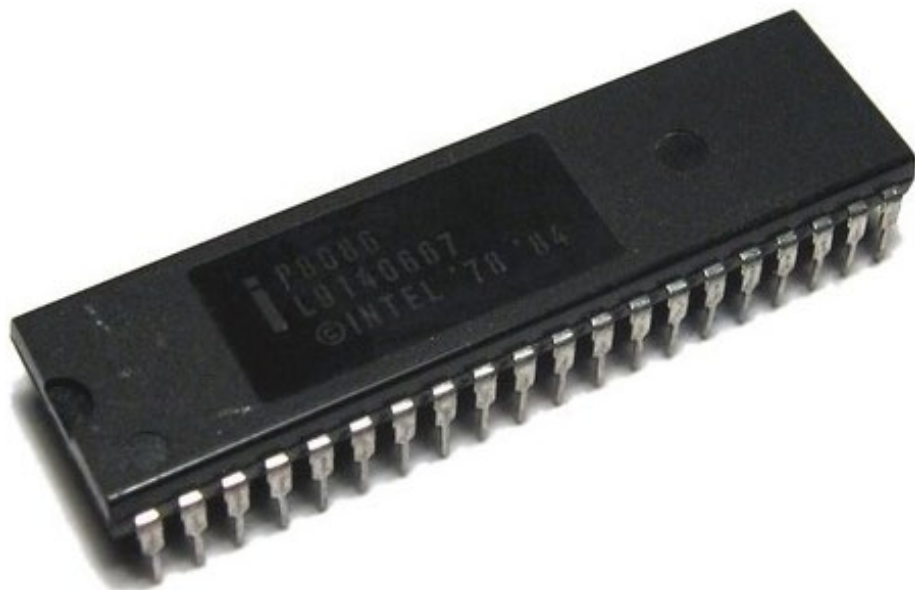
Harpoon Missile

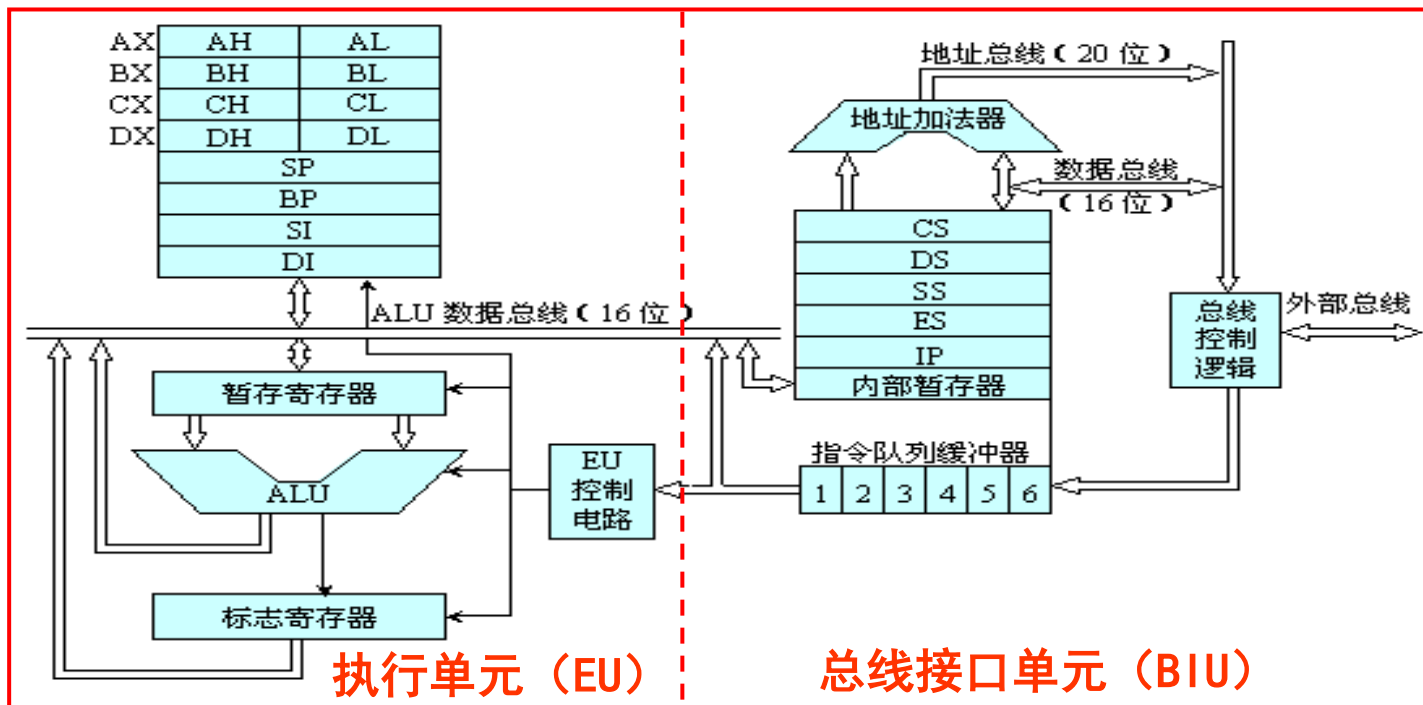


Air Launched	Length - 151.5 in.
	Weight - 1160 Lb.
Ship Launched	Length - 180.5 in.
	Weight - 1460 Lb.



2. 2. 1 8086/8088内部结构



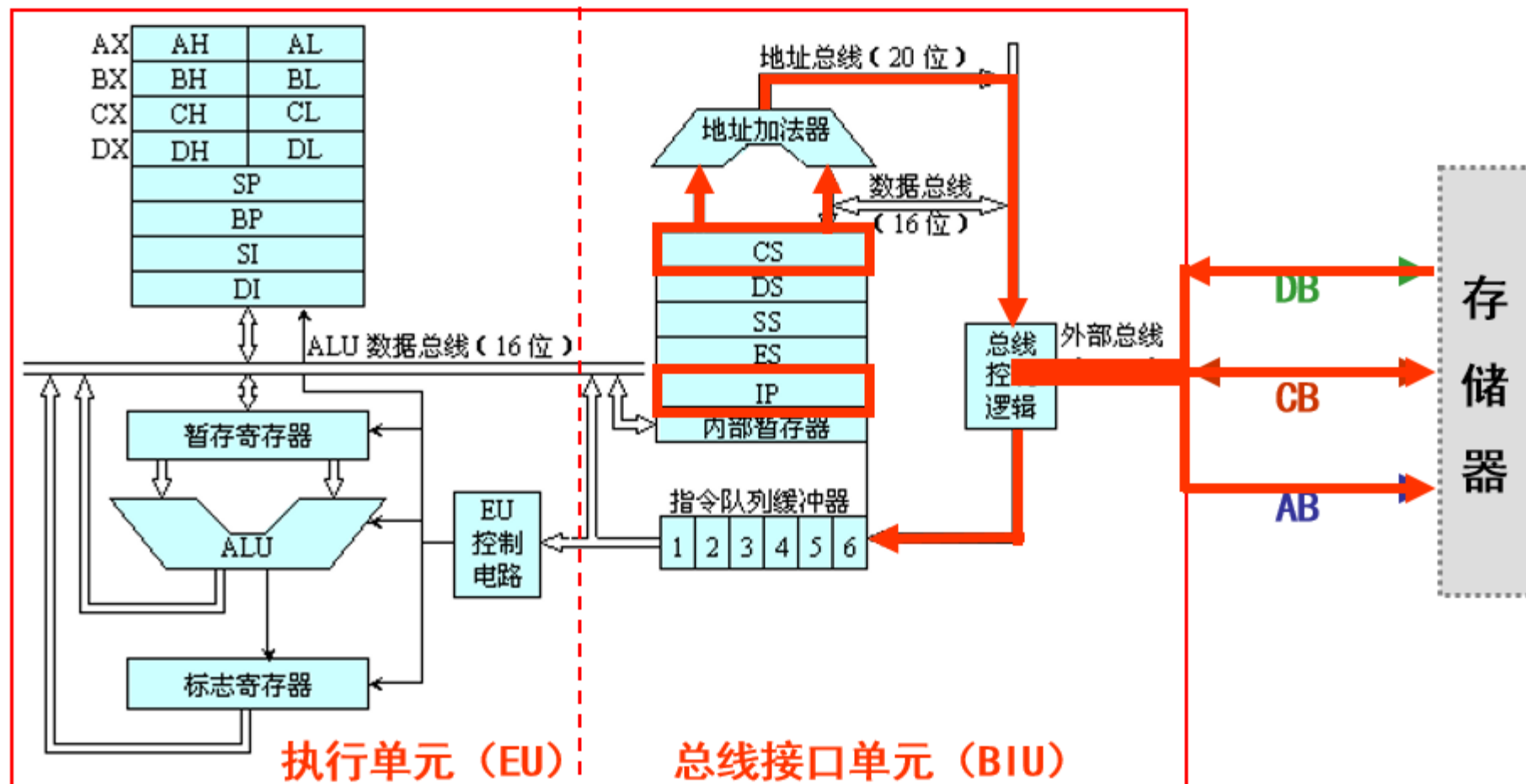


1、BIU

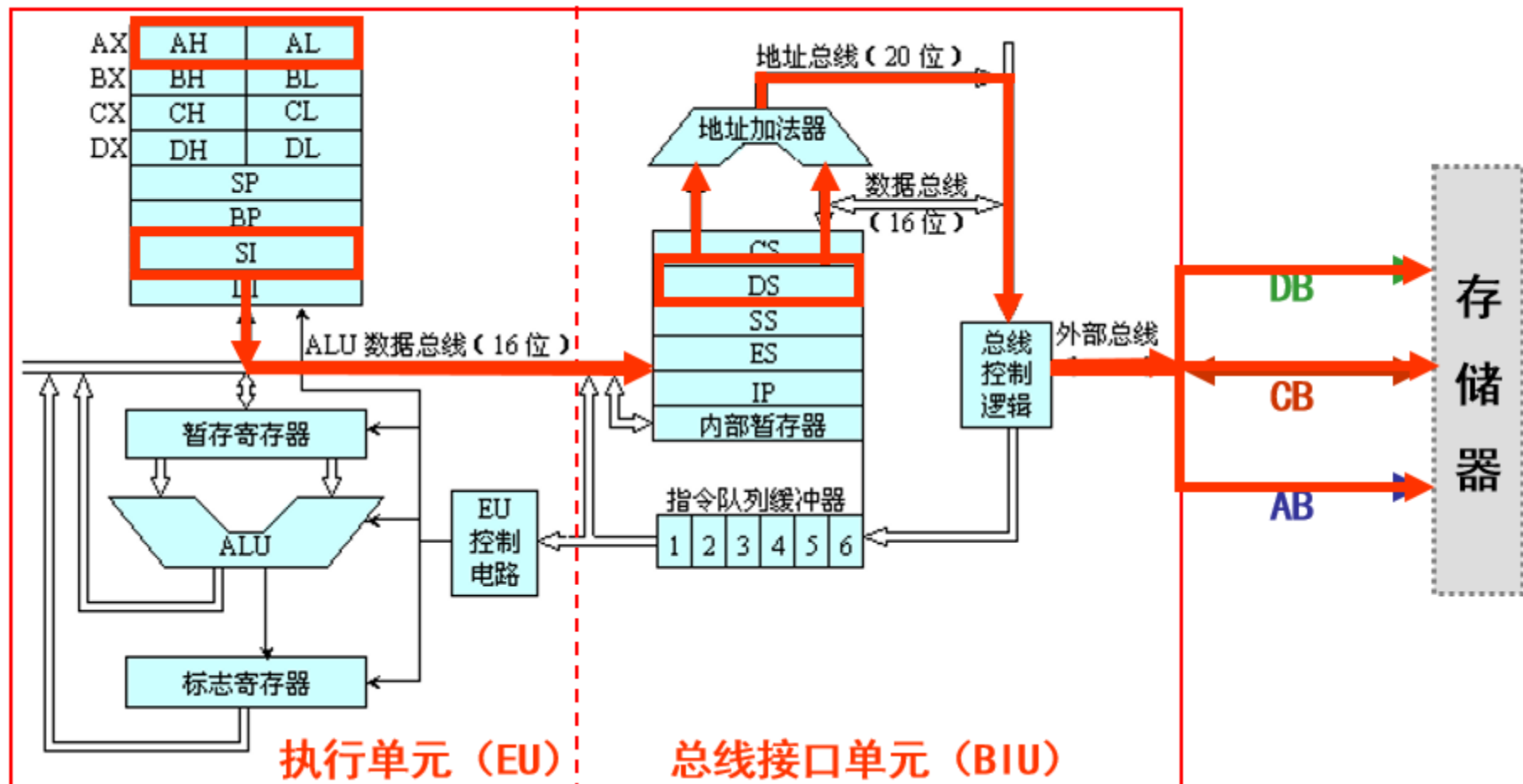
- ① 总线控制逻辑
- ② 指令指针寄存器IP
- ③ 段寄存器
- ④ 地址加法器
- ⑤ 指令队列缓冲器

2、EU

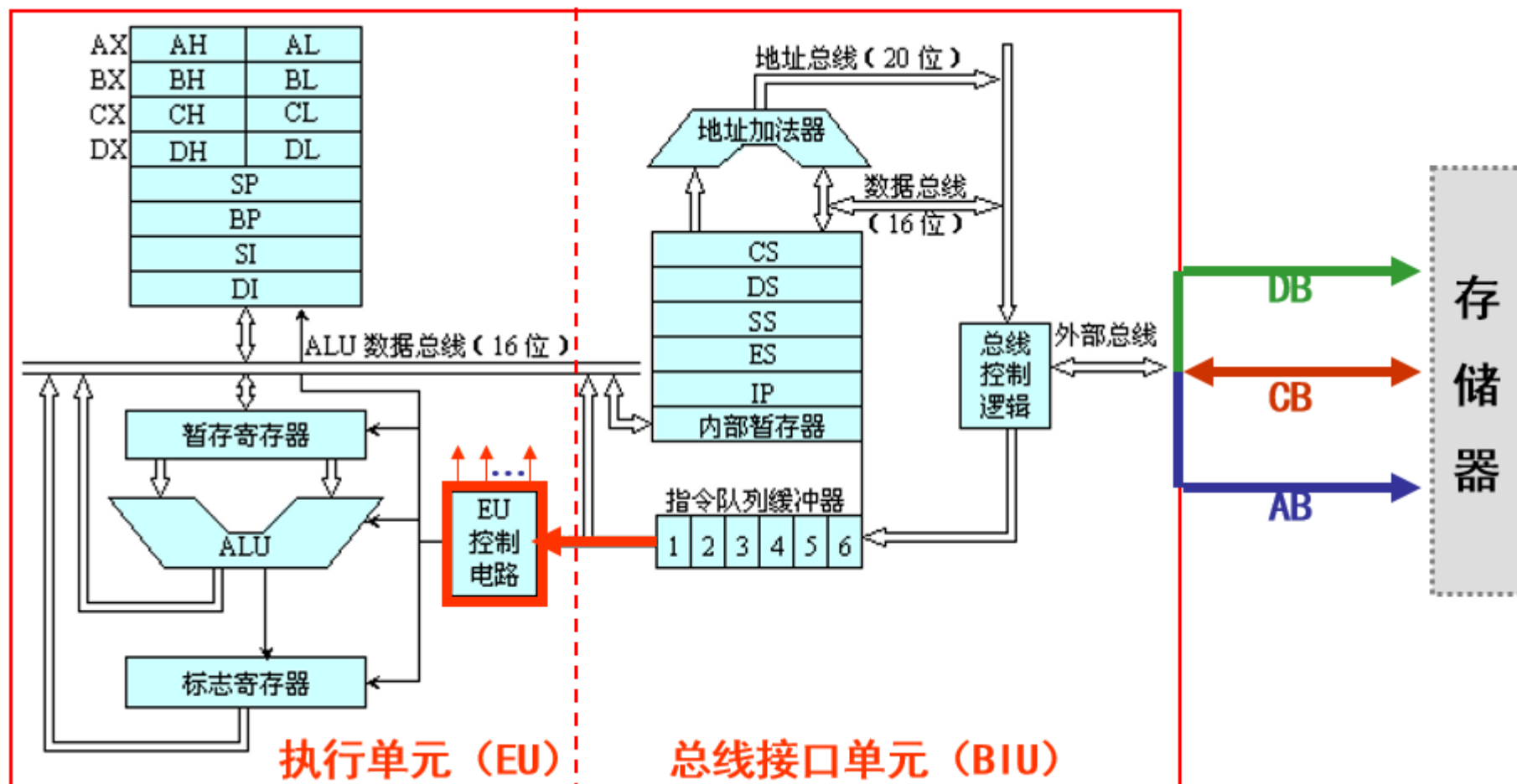
- ① 算术逻辑单元 (ALU)
- ② EU控制电路
- ③ 寄存器组



❖ CPU取指过程



❖ CPU向内存写数过程

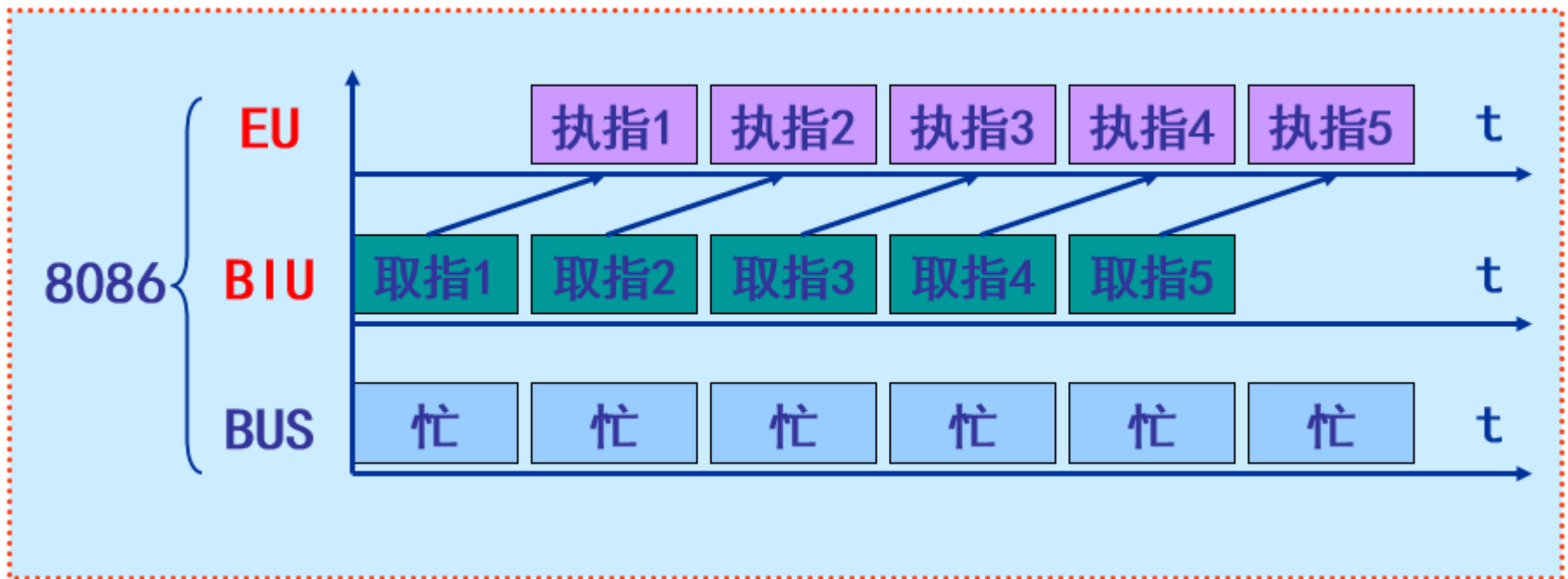


❖ EU取指、译码过程



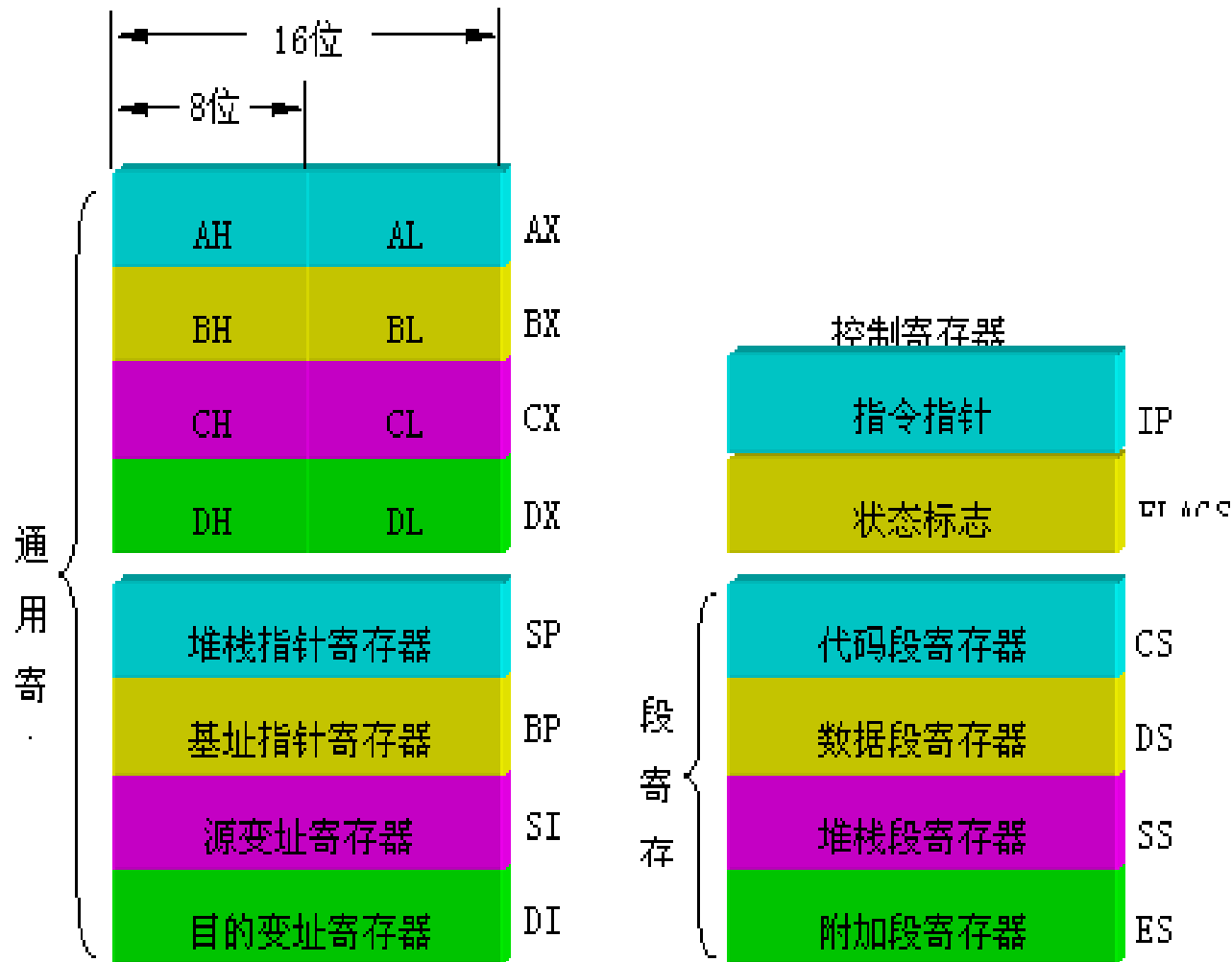
8086的内部结构特点：

指令的提取与执行分别由BIU和EU两部件完成，二者既独立又相互配合——并行工作。





2.3 微处理器寄存器结构及数据类型



8088/8086内部有14个16位的寄存器。按其功能可分为三大类：通用寄存器（8个），段寄存器（4个）和控制寄存器（2个）



➤ 通用寄存器

数据寄存器 (AX, BX, CX, DX)

地址指针寄存器 (SP, BP)

变址寄存器 (SI, DI)



➤ 数据寄存器

- 8088/8086含4个16位数据寄存器，它们又可分为8个8位寄存器，即：
 - AX **→ AH, AL**
 - BX **→ BH, BL**
 - CX **→ CH, CL**
 - DX **→ DH, DL**



➤ 数据寄存器特有的习惯用法

- AX：累加器。所有I/O指令都通过AX与接口传送信息，中间运算结果也多放于AX中；
- BX：基址寄存器。在间接寻址中用于存放基地址；
- CX：计数寄存器。用于在循环或串操作指令中存放计数值；
- DX：数据寄存器。在间接寻址的I/O指令中存放I/O端口地址；在32位乘法运算时，存放高16位数。



➤ 地址指针寄存器

- SP：堆栈指针寄存器，其内容为栈顶的偏移地址；
- BP：基址指针寄存器，常用于在访问内存时存放内存单元的偏移地址。



➤ BX与BP在应用上的区别

- 作为通用寄存器，二者均可用于存放数据；
- 作为基址寄存器，用BX表示所寻找的数据在数据段；
用BP则表示数据在堆栈段。

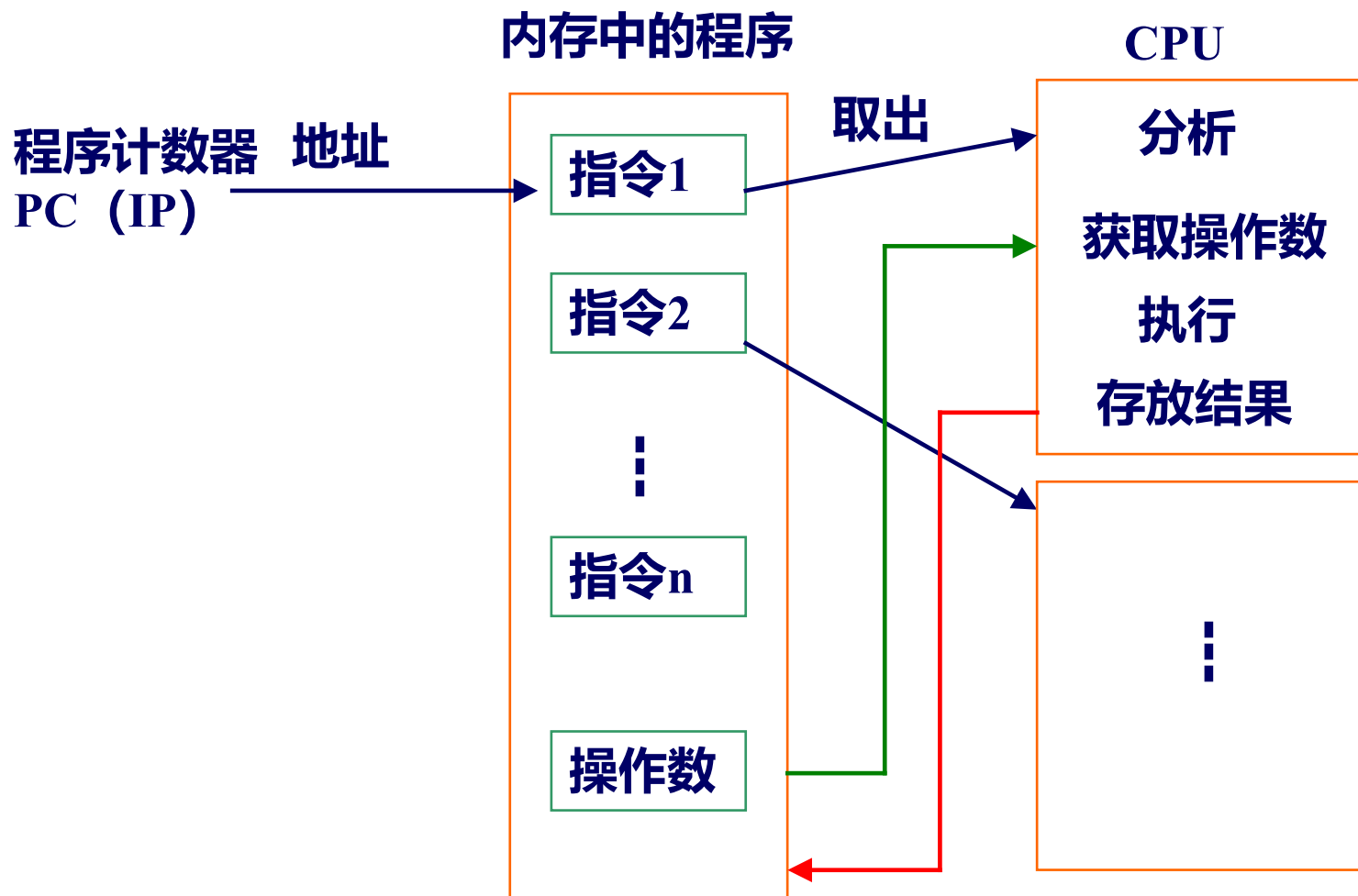


➤ 变址寄存器

- SI：源变址寄存器
- DI：目的变址寄存器在指令中常用于存放数据在内存中的地址。

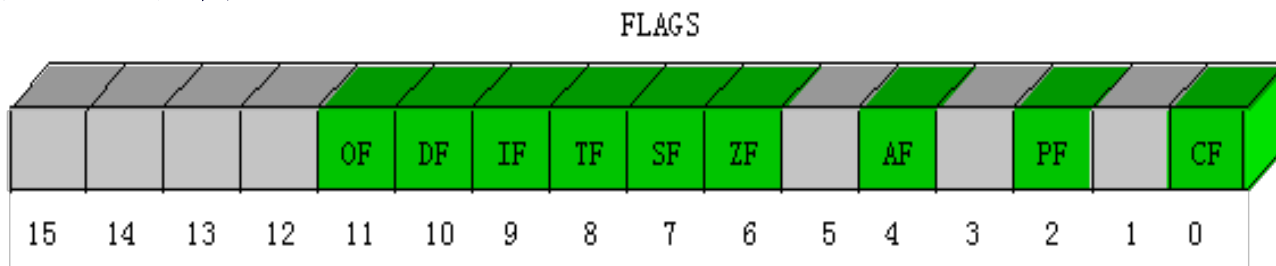


➤ 控制寄存器





➤ 标志寄存器FLAGS



- CF 进位标志 (Carry Flag)：有进/借位（字节运算时时第7位，字运算为第15位）为1，无进借位为0。另循环指令也使CF=1。
- AF 辅助进位标志 (Auxiliary Carry Flag)：低4位向高位有进/借位（即第3位向第4位进位）为1，否则AF=0。
- PF 奇偶标志 (Parity Flag)：运算结果若低8位所含1的个数为偶数，则PF=1，否则PF=0。
- ZF 全零标志 (Zero Flag)：当运算结果使有效位数的各位全为零时ZF=1，否则ZF=0。
- SF 符号标志 (Sign Flag)：当运算结果为负时SF=1，否则SF=0。SF的值就是有符号数的最高位(符号位)。
- OF 溢出标志 (Overflow Flag)：当运算结果超出了机器所能表示的范围时，则OF=1，表示溢出，否则OF=0。

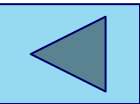


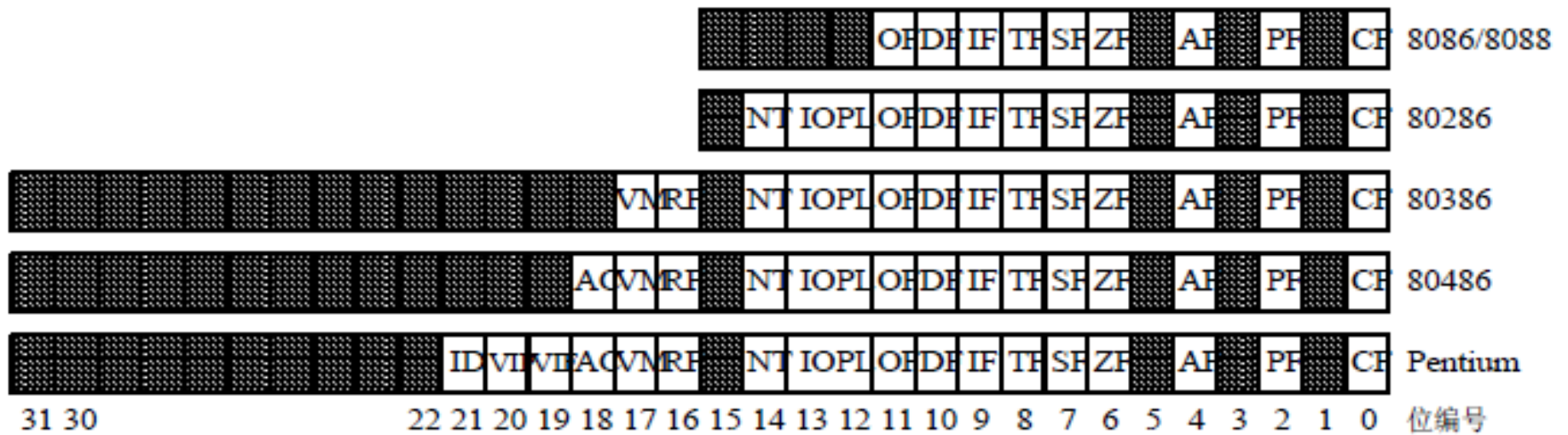
控制标志

- DF 方向标志 (Direction Flag) : 如果DF为0, 则串操作过程中地址会不断增值, 反之, 如果为1, 则会不断减值。
- IF 中断允许标志 (Interrupt Enable Flag) : 如果为0, 则CPU不能响应可屏蔽中断请求, 反之, 可接受可屏蔽中断请求。
- TF 单步标志 (Trap Flag) : 如果为1, 则CPU按跟踪方式执行命令。

$$\begin{array}{r} 0010\ 0011\ 0100\ 0101 \\ +0011\ 0010\ 0001\ 1001 \\ \hline 0101\ 0101\ 0101\ 1110 \end{array}$$

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





80x86的标志寄存器



➤ 段寄存器

- CS:代码段寄存器，存放代码段的段基地址。
- DS:数据段寄存器，存放数据段的段基地址。
- ES:附加段寄存器，存放附加数据段的段基地址。
- SS:堆栈段寄存器，存放堆栈段的段基地址

段寄存器的值表明相应逻辑段在内存中的位置



2.4 微处理器主要引脚信号与总线时序

最小工作模式：单微处理器模式，系统中只有一个8086微处理器，所有的总线控制信号均由CPU直接产生，因此系统中的总线控制电路少，控制信号较少，一般不必接总线控制器。

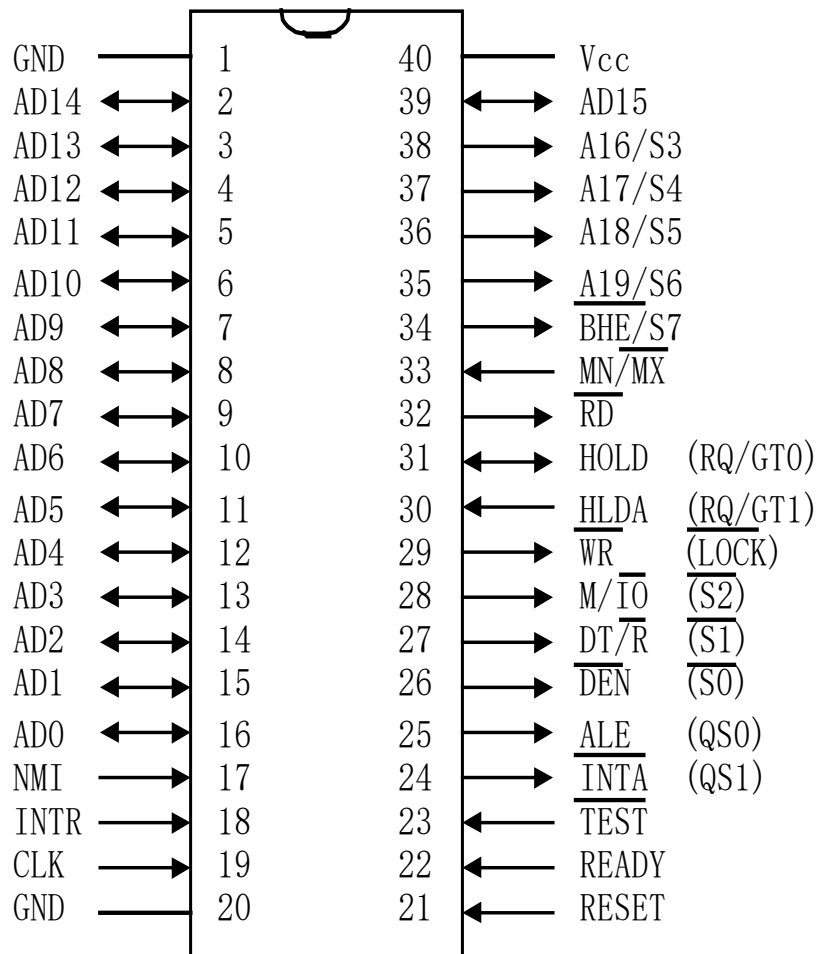
最大工作模式：多微处理器模式，系统中可包含2个或2个以上微处理器，其中一个8086为主处理器，其他的微处理器为协处理器（如8087、8089），系统中控制信号较多，需要通过总线控制器与总线相连，控制总线驱动能力较强。

$MN/\overline{MX}=0$ ——工作于最大模式

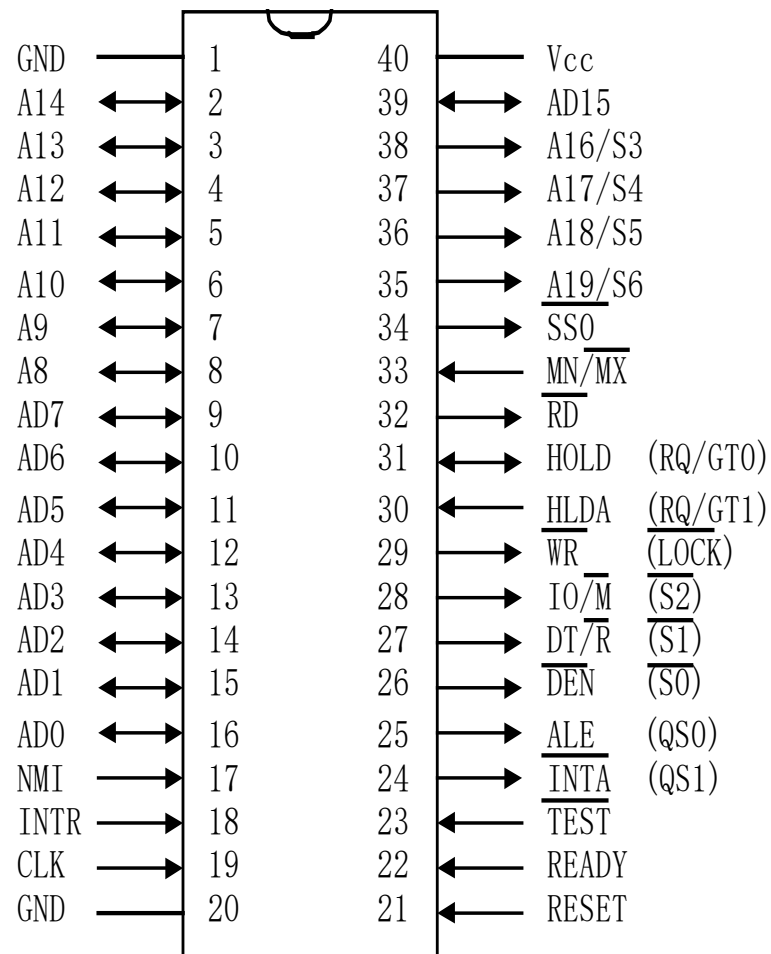
$MN/\overline{MX}=1$ ——工作于最小模式

2.4.1 8086/8088引脚信号

8086



8088





➤ 两种模式的共用引脚

分时复用就是一个引脚在不同的时刻具有两个甚至多个作用

● 数据总线和地址总线

8086有20条地址线和16条数据线，8088有20条地址线和8条数据线。数据线和低位地址线采用分时复用的方法。

8086：16条双向、三态数据线AD15~AD0，20条单向、输出地址线A19/S6、A18/S5、A17/S4、A16/S3、AD15~AD0，**低16条地址线与数据线复用。**

8088：8条双向、三态数据线AD7~AD0，20条单向、输出地址线为A19/S6、A18/S5、A17/S4、A16/S3、AD15~AD8、AD7~AD0，**低8条地址线与数据线分时复用。**

AD19~AD16与状态信号分时复用。S6始终为低，S5与标志寄存器中的中断允许标志IF一致，S5为高电平，表示IF=1，允许可屏蔽中断，S4、S3表示当前使用的段寄存器。



S4	S3	含义
0	0	当前正在使用ES
0	1	当前正在使用SS
1	0	当前正在使用CS，或者未使用任何段寄存器
1	1	当前正在使用DS

● 控制总线

8086共有16条控制总线，其中有8条共用的控制线。

$\overline{\text{MN}}/\overline{\text{MX}}$	高电平为最小模式，低电平为最大模式	决定处理器的工作模式
$\overline{\text{RD}}$	低电平有效，表示CPU从外部读入信息	从存储器或IO读取数据或指令
NMI	上升沿有效，有效时CPU进入中断	中断型号固定为2
INTR	高电平有效，有效时表示外部设备有中断请求	如果IF=1，CPU才有可能响应中断
RESET	高电平有效，有效时CPU从FFFF0H开始取指令	按RESET按钮或开机时该信号都有效
READY	高电平有效，有效时外部设备准备好	
$\overline{\text{TEST}}$	低电平有效	此信号一般用于硬件测试
$\overline{\text{BHE}}/\text{S7}$		仅限8086，对8088该脚为 $\overline{\text{SS0}}$



- 其它信号线

VCC: 单一的正5V电压; GND: 电源地; CLK: 时钟输入。

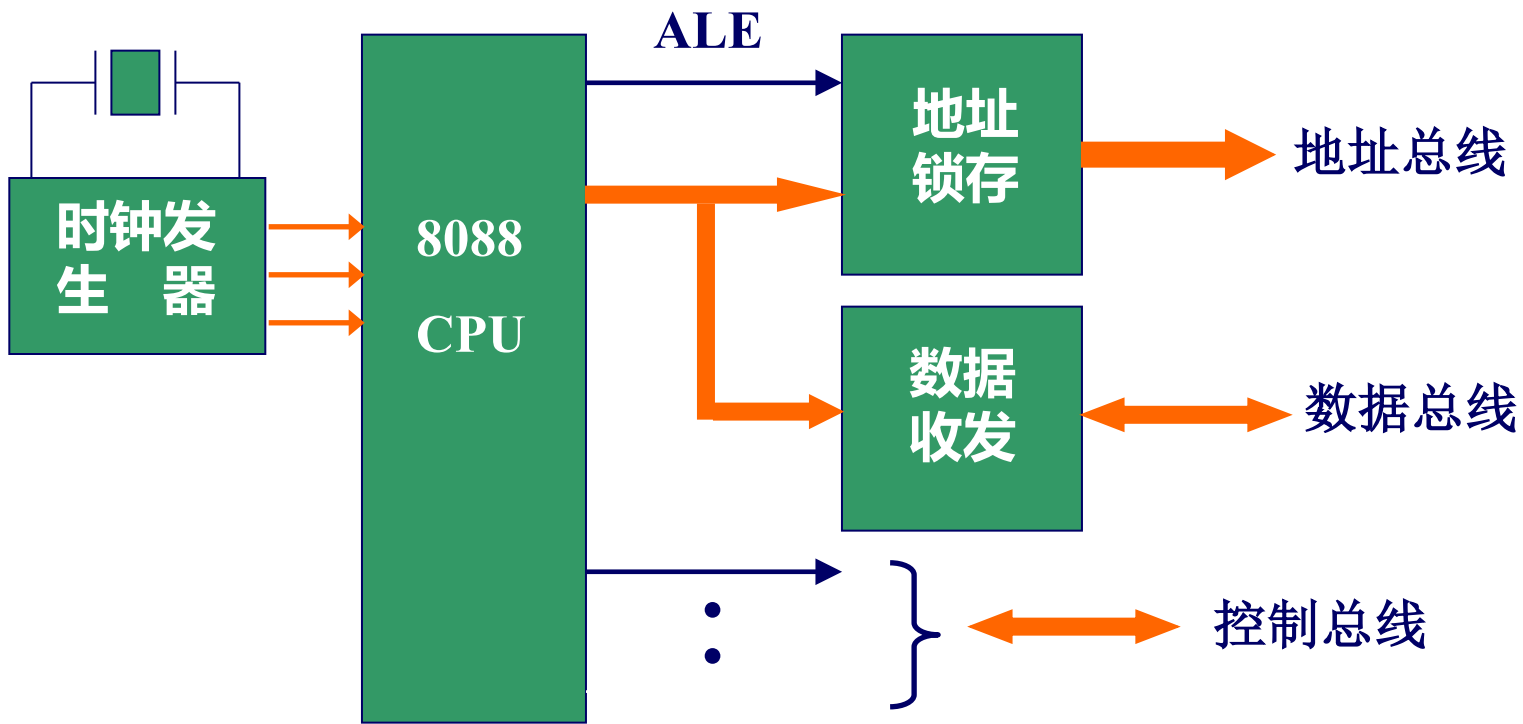


➤ 最小模式专用引脚

$\overline{MN}/\overline{MX}$ 接+5V时，系统处于最小模式。

- 在最小模式下，8086的 $\overline{M}/\overline{IO}$ 、 \overline{RD} 、 \overline{WR} 以及 $\overline{DT}/\overline{R}$ 组合决定总线周期的操作类型。

$\overline{DT}/\overline{R}$	$\overline{M}/\overline{IO}$	\overline{RD}	\overline{WR}	对应的操作	指令示例
0	0	0	1	读I/O 接口	IN AL,DX
0	1	0	1	读存储器	MOV AX,[1000H]
1	0	1	0	写I/O 接口	OUT DX,AL
1	1	1	0	写存储器	MOV [2000H],AL
1	×	0	1	非法操作	无
0	×	1	0	非法操作	无
×	×	1	1	无读写操作	无



最小模式下的总线连接示意图



➤ 最大模式专用引脚

最大模式有8个专用控制引脚。

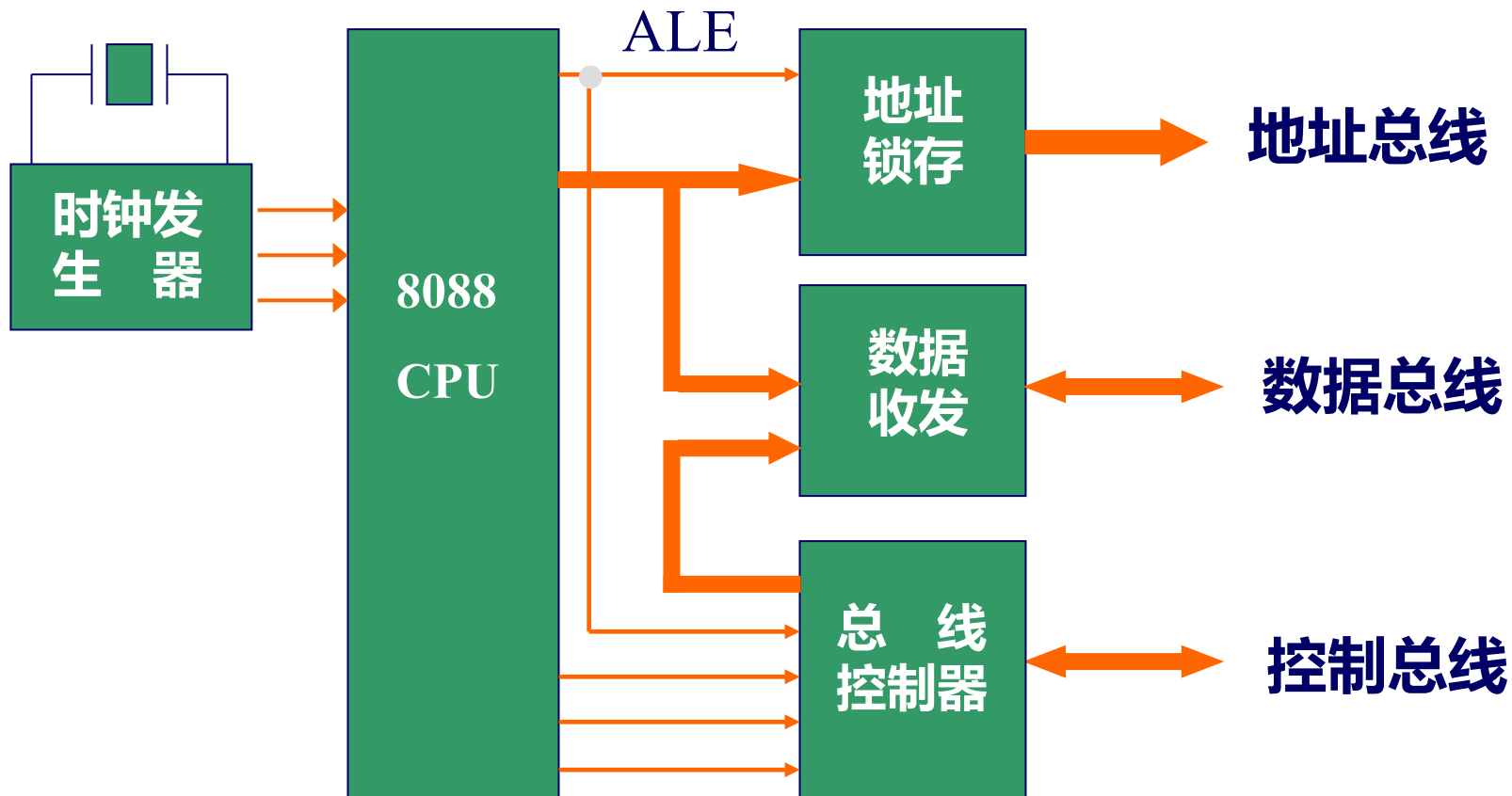
QS和QS0：指令队列状态信号：三态，输入/输出。

$\overline{S2}$ 、 $\overline{S1}$ 、 $\overline{S0}$ ：总线周期状态信号：三态，输出，在最大模式下，8086/8088借助于这3个控制信号，通过外接总线控制器8288产生多个不同的控制信号。

$\overline{S2}$	$\overline{S1}$	$\overline{S0}$	通过 8288 产生的信号	具体操作状态	有关指令示例
0	0	0	\overline{INTA}	发中断响应信号	无
0	0	1	\overline{IORC}	读 I/O 接口	IN AL, DX
0	1	0	\overline{IOWC} , \overline{AIOWC}	写 I/O 接口	OUT DX, AL
0	1	1	无	暂停	NOP
1	0	0	\overline{MRDC}	取指令	无
1	0	1	\overline{MRDC}	读内存	MOV AX, [1234H]
1	1	0	\overline{MRDC}	写内存	MOV [DI], CX
1	1	1	\overline{MWTC} , \overline{AMWC}	无效状态	无
			无		



- $\overline{\text{LOCK}}$ 总线封锁：三态，输出，低电平有效，该信号有效时，不允许其它主控部件使用总线。有两种操作都可使该信号有效，一是使用指令LOCK，二是INTR有效。
- $\overline{\text{RQ}}/\overline{\text{GT1}}, \overline{\text{RQ}}/\overline{\text{GT0}}$ 总线请求/允许：三态，输入/输出，低电平有效。这两个信号供CPU以外的两个协处理器用来发出使用总线的请求，接收CPU对总线请求信号的回答信号。



最大模式下的总线连接示意图



➤ 8088和8086CPU引线功能比较

- 数据总线宽度不同

8088的外部总线宽度是8位，8086为16位。

- 访问存储器和输入输出控制信号含义不同

8088—— $\overline{IO}/M=0$ 表示访问内存，为1表示访问接口；

8086—— $\overline{IO}/M=1$ 表示访问内存，为0表示访问接口；

- 其他部分引线功能的区别



2.4.2 系统复位与启动操作

8086的复位和启动操作是通过RESET引脚的触发信号执行的，8086要求复位信号至少维持4个时钟周期的高电平，若是初次上电则要求至少维持50 μ s的高电平

当RESET信号一进入高电平，8086就会结束现行操作，进入内部复位状态，并将CPU内部将全部寄存器置为初值

标志寄存器	0000H
指令指针IP	0000H
CS寄存器	FFFFH
DS寄存器	0000H
SS寄存器	0000H
ES寄存器	0000H
指令队列	空
其它寄存器	0000H

系统复位重新启动时，便从内存的FFFF0H单元开始执行程序。一般在FFFF0H处存放一条JMP指令，转移到系统程序入口处，保证系统一启动就可以自动进入系统程序



2.4.3 微处理器时钟信号

- 在微型计算机中，所有操作都要靠时钟信号统一协调，系统中有多时钟的称呼，其中包括内部时钟、处理器时钟、外部时钟、总线时钟、系统时钟等。
- 不同的时钟以时间为度量单位又称为时钟周期，以频率为度量单位又称为时钟频率，时钟频率是时钟周期的倒数。
- 时钟周期的单位有s、ms、 μ s和ns，对应的时钟频率单位有Hz、kHz、MHz和GHz。
- 内部时钟：也就是处理器时钟，是指为处理器内部工作定时的时钟，对应的内部时钟频率又称为主频，通常所说的时钟频率实际上就是主频。
- 外部时钟：由引脚上的外部时钟发生器产生，为协调外部工作的定时时钟称为外部时钟或总线时钟。



- ❖ 时序（Timing）描述各信号随时间变化的规律及信号间相互关系。
- ❖ 总线时序描述总线操作中相关信号的时序；它由总线主控设备产生。
- ❖ 总线时序决定了CPU控制总线期间，系统各部件应遵守的定时规则。

什么是总线操作？

- 总线操作是指CPU通过总线对外进行的各种操作



8086/8088工作周期

时钟周期：微处理器的最小定时单位，等于主频倒数。

机器周期：完成一个独立的操作所需时间。

总线周期：完成一次总线操作（存储器或I/O访问）所需的读/写机器周期。

指令周期：一条指令从取出到执行完毕所需要的时间。通常需要一个或多个总线周期

8086/8088一个基本总线周期包括4个时钟周期的时间。习惯称为4个状态，分别记为T1、T2、T3和T4状态。

T1状态：输出地址信息并锁存。

T2状态：撤消地址，为传送数据作准备。

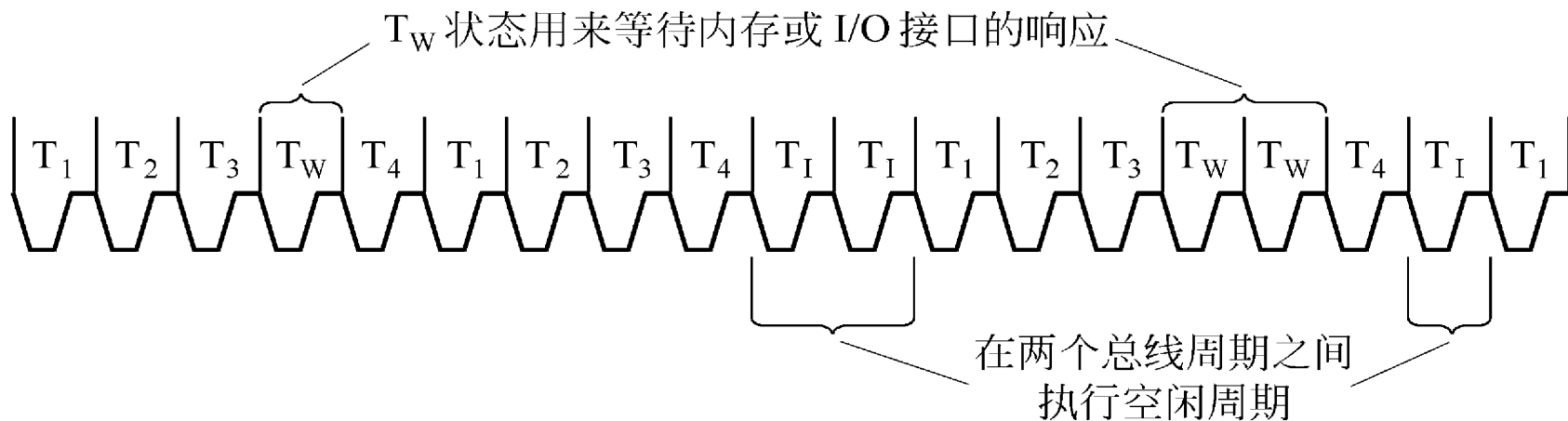
T3状态：如果外部准备好，则数据稳定在总线上。

T4状态：读写总线上的数据，总线周期结束。

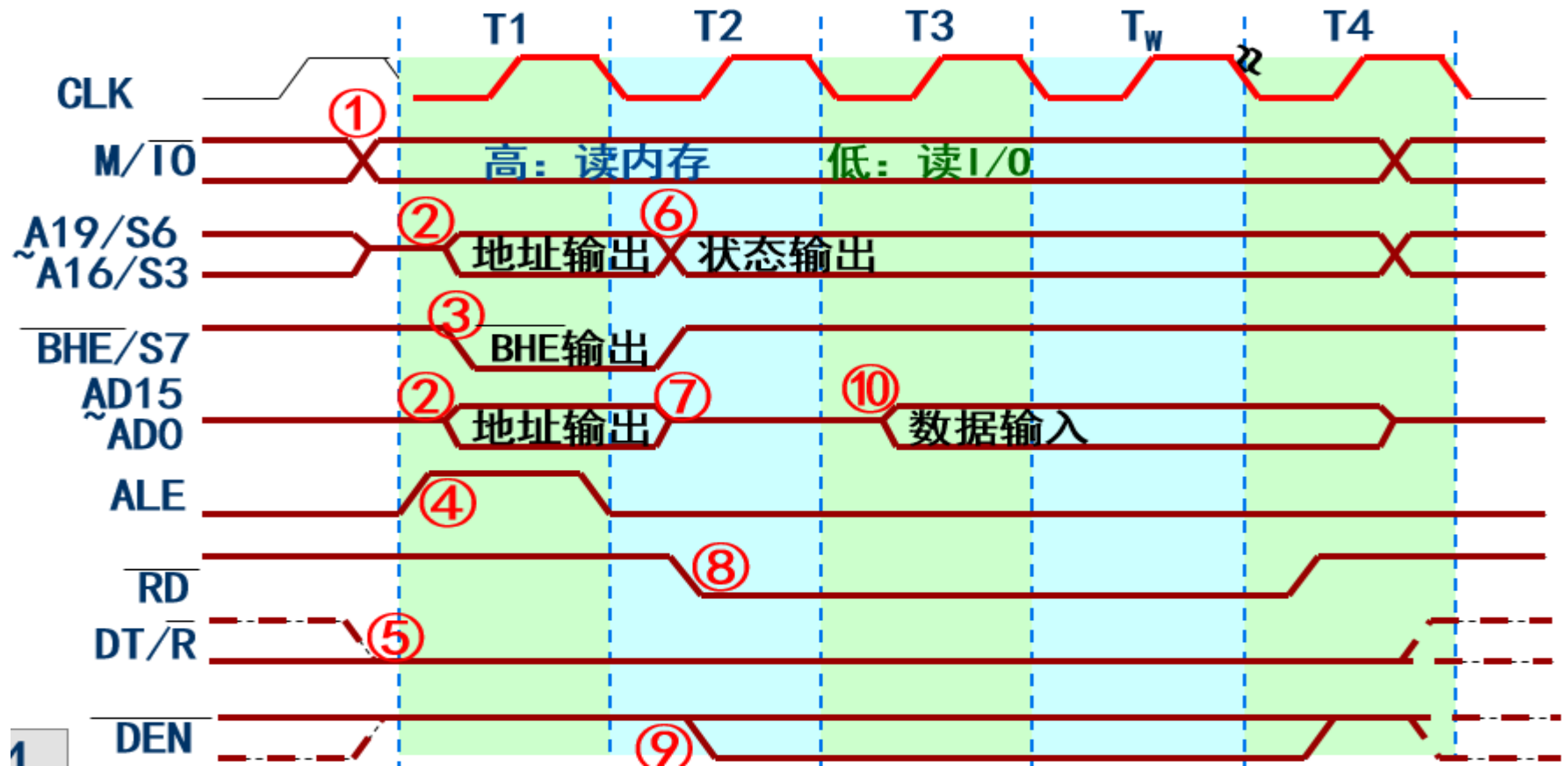
Tw：等待周期 T_w 处于T3与T4之间。同步时序通过插入等待周期，来使速度有差别的两个部件保持同步。



● 典型8086总线周期序列



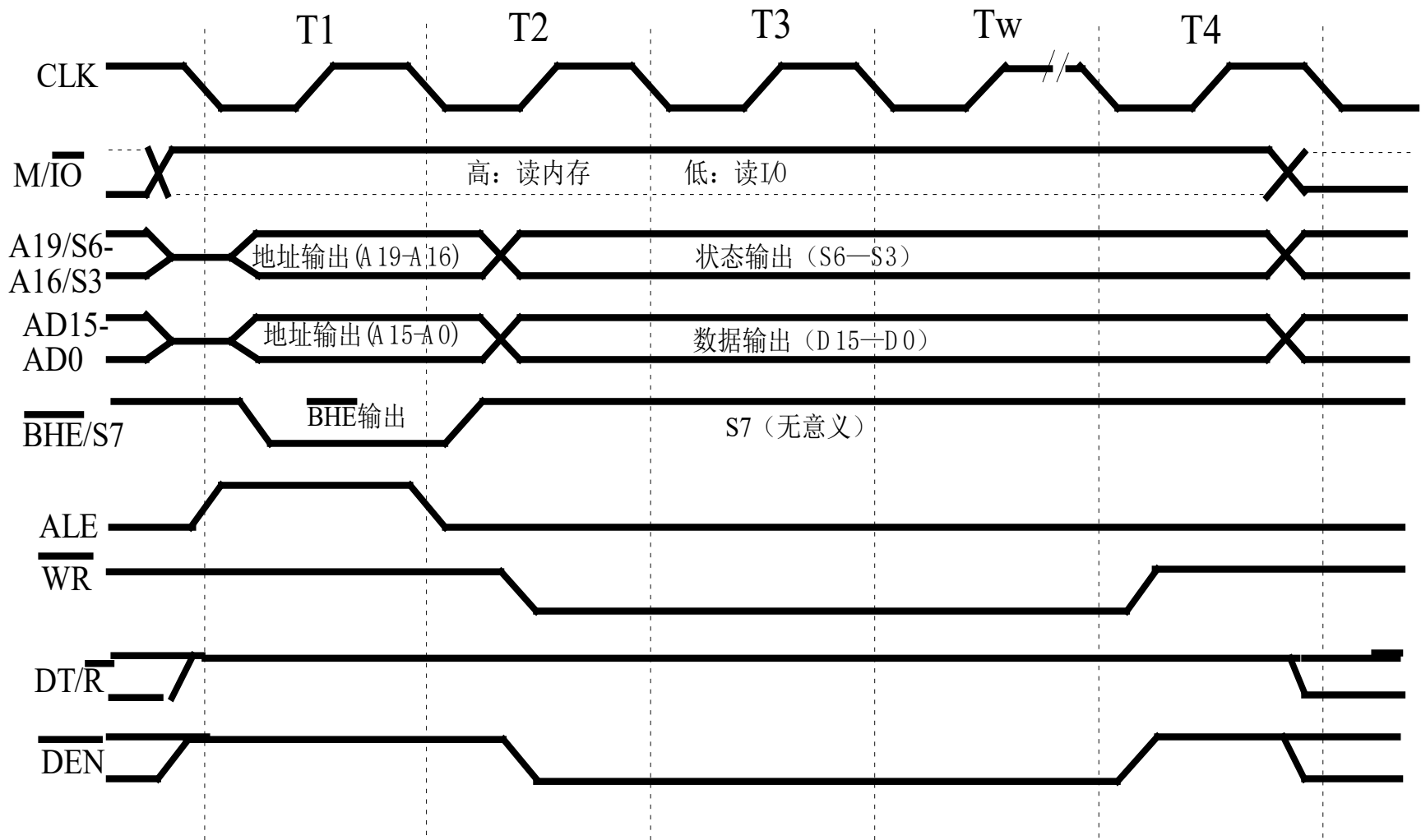
需要指出，只有在CPU和内存或I/O接口之间传输数据，以及填充指令队列时，CPU才执行总线周期。如果在1个总线周期之后，不立即执行下一个总线周期，那么，系统总线就处在空闲状态，此时，执行空闲周期。



8086最小模式下的读总线周期时序



- T_1 状态——输出20位存储器地址 $A_{19} \sim A_0$ ， IO/\overline{M} 输出低电平，表示存储器操作； ALE 输出正脉冲，利用其下降沿锁存复用总线上的输出地址
- T_2 状态—— WR^* 信号有效， \overline{DEN} 信号有效以输出数据 $D_{15} \sim D_0$ ；
- T_3 和 T_w 状态—— \overline{WR} 、 DEN^* 、 IO/M^* 、 DT/R^* 等控制信号持续有效， T_3 时钟下降沿检测 $READY$ 信号，决定是否插入 T_w ； T_w 期间，各信号延续状态。
- T_4 状态——完成数据传送，并准备过渡到下一操作。 WR^* 、 DEN^* 转为无效。



8086最小模式下的写总线周期时序



- T_1 状态——输出20位存储器地址 $A_{19} \sim A_0$ ， IO/M^* 输出低电平，表示存储器操作； ALE 输出正脉冲，表示复用总线输出地址
- T_2 状态——输出控制信号 RD^* ，选通存储器； DEN^* 信号，选通数据收发器
- T_3 和 T_w 状态——检测数据传送是否能够完成
- T_4 状态——前沿读取数据，完成数据传送



- 最大模式下8086的读总线操作与最小模式下的读操作在逻辑上是完全一样的，只不过在分析具体时序时，最大模式下需要考虑CPU和总线控制器两者产生的信号。



2.5 实模式存储器管理

➤ 8086/8088存储空间组织

8086/8088有20条地址线，可直接对1M个存储单元进行访问。每个存储单元存放一个字节型数据，且每个存储单元都有一个20位的地址，这1M个存储单元对应的地址为00000H~FFFFFFH。

一个存储单元中存放的信息称为该存储单元的内容。如下图所示，00001H单元的内容为9FH，记为：(00001H)=9FH。



● 线性地址

① 20条AB, 寻址1M空间

② 按字节组织, 每个字节唯一地址

字节: 顺序存放

③ 字: 低位字节放在低地址中, 高位字节放在高地址中

④ 双字: 低位字是偏移量, 高位字是段地址

⑤ 规则字: 低位字节存放在偶数地址

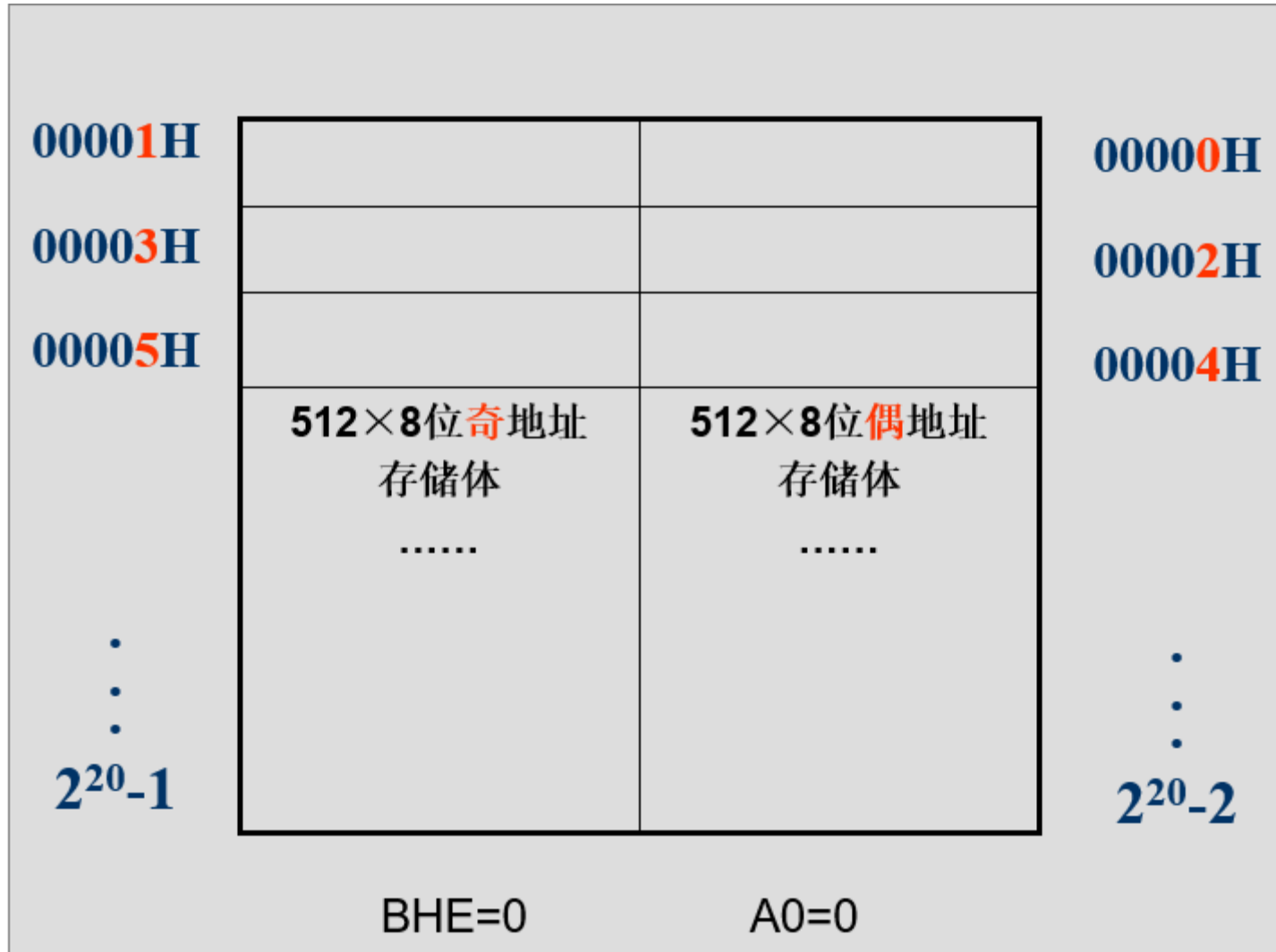
⑥ 非规则字: 低位字节存放在奇数地址

00000H	
00001H	
00002H	
00003H	
00004H	
00005H	
00006H	
00007H	
...	...

内存存储器编址



● 分体结构

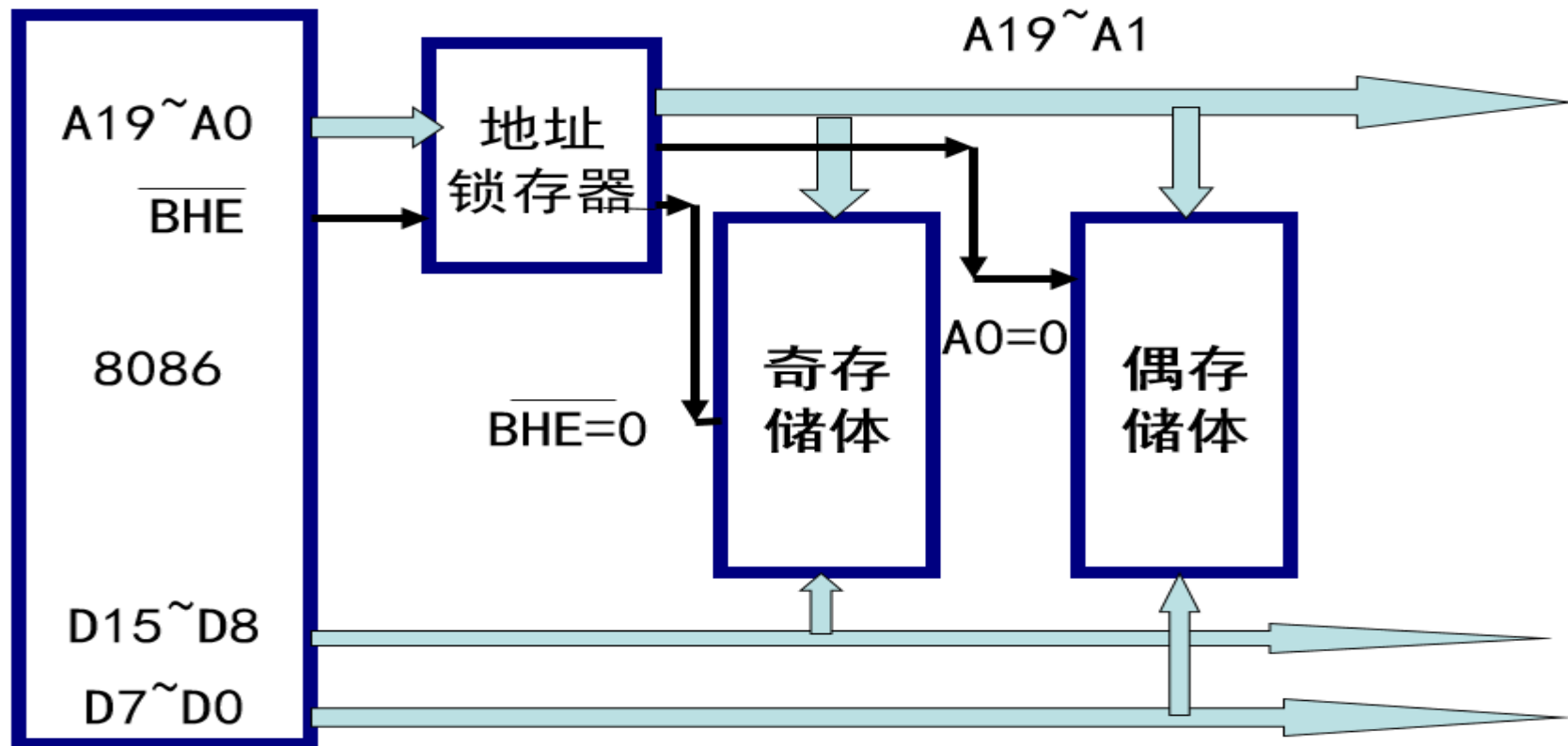




8088存储器组织：一个完整1MB存储体

8086存储器组织：1MB分成两个512KB

(一个奇地址+一个偶地址)





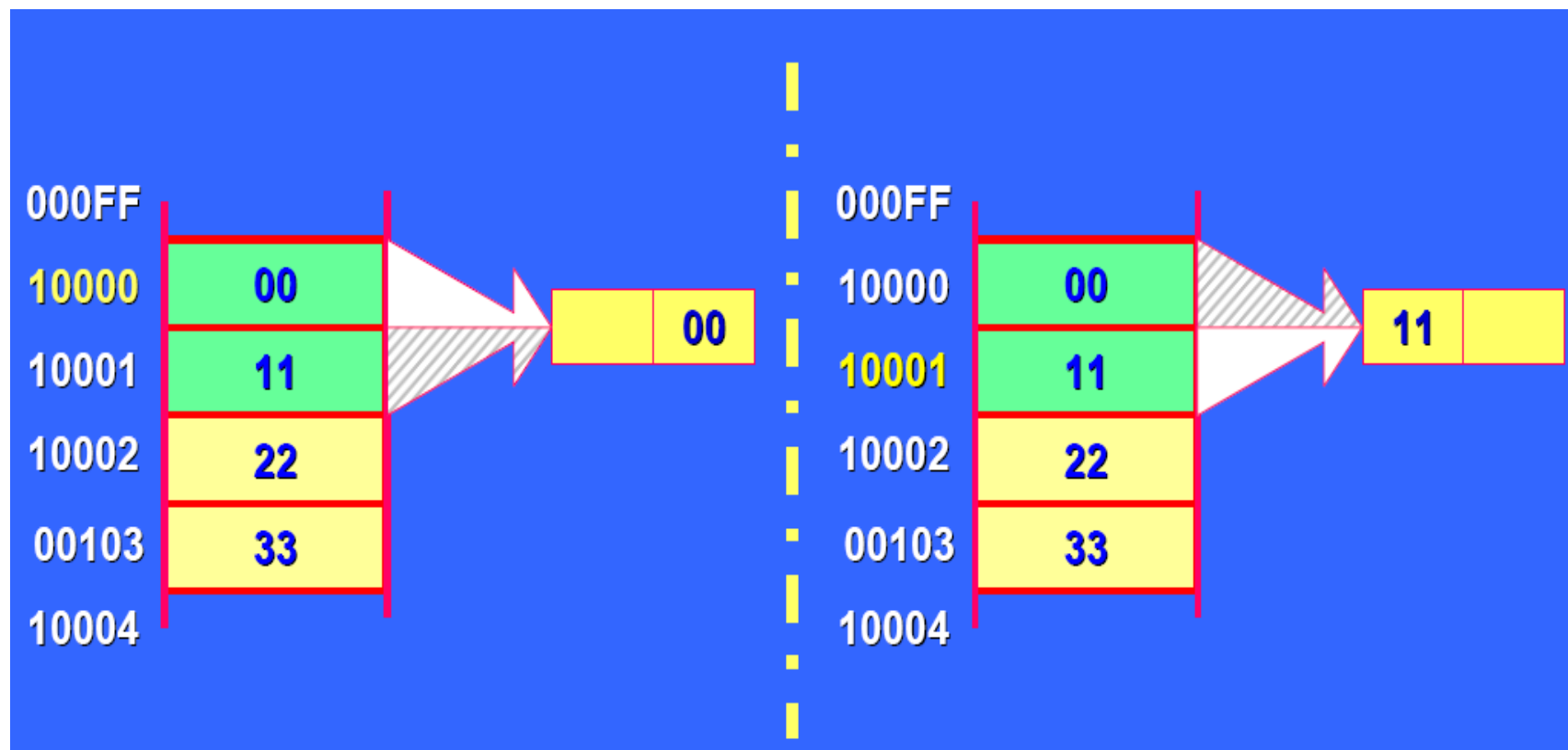
BHE与地址引脚 A_0 编码的含义

A_0	\overline{BHE}	操作	使用的数据线
0	0	访问偶地址开始的一个字单元	$D_{15} \sim D_0$
0	1	访问偶地址的一个字节单元	$D_7 \sim D_0$
1	0	访问奇地址的一个字节单元	$D_{15} \sim D_8$
1	0	访问奇地址开始的一个字单元， 第一个总线周期访问奇地址字 节，第二个总线周期访问偶地址 字节	$D_{15} \sim D_8$
0	1		$D_7 \sim D_0$

一个字可以从偶地址存放也可以从奇地址存放，由于CPU访问存储器都是以字为单位，并从偶地址开始。

➤ 分体结构读写操作

8086CPU访问存储器时，对于字节型数据，无论它存放在偶地址的低位库，还是奇地址的高位库，均可通过一个总线周期完成数据的读/写操作。

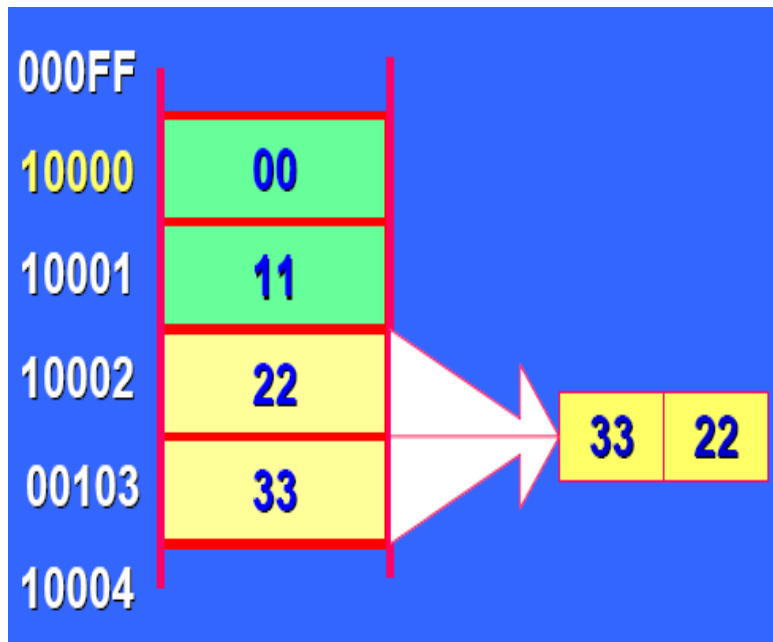


从偶地址读一个字节

从奇地址读一个字节

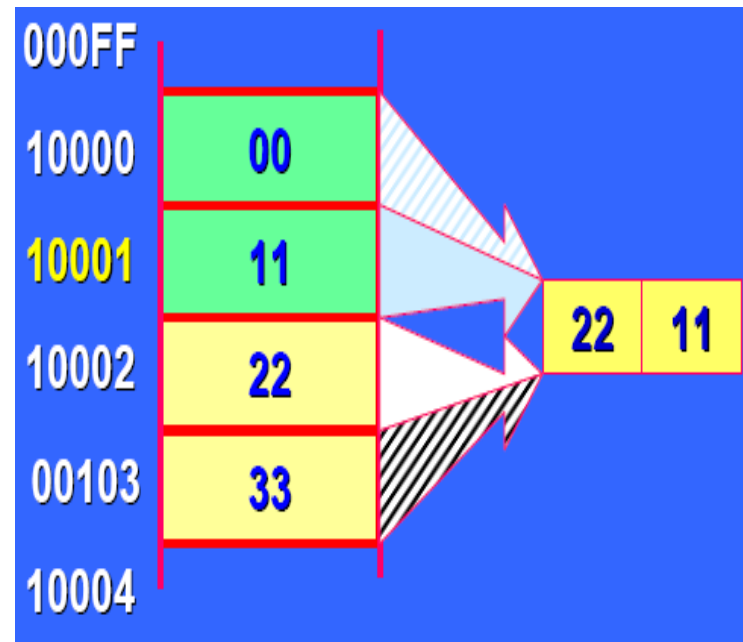


- 若字单元地址从偶地址开始，只需访问一次存储器



从偶地址读一个字

- 若字单元地址从奇地址开始，需访问两次存储器



从奇地址读一个字



➤ 存储器的编址（1）

- 每个内存单元在整个内存空间中都具有惟一的地址 → 物理地址
- 每个内存单元的地址码都由两部分组成：
 - 段（基）地址
16位
 - 段内地址 → 相对地址/偏移地址
16位

8088为16位结构，所以段地址和偏移地址均为16位



➤ 存储器的编址（2）

- 段基地址：

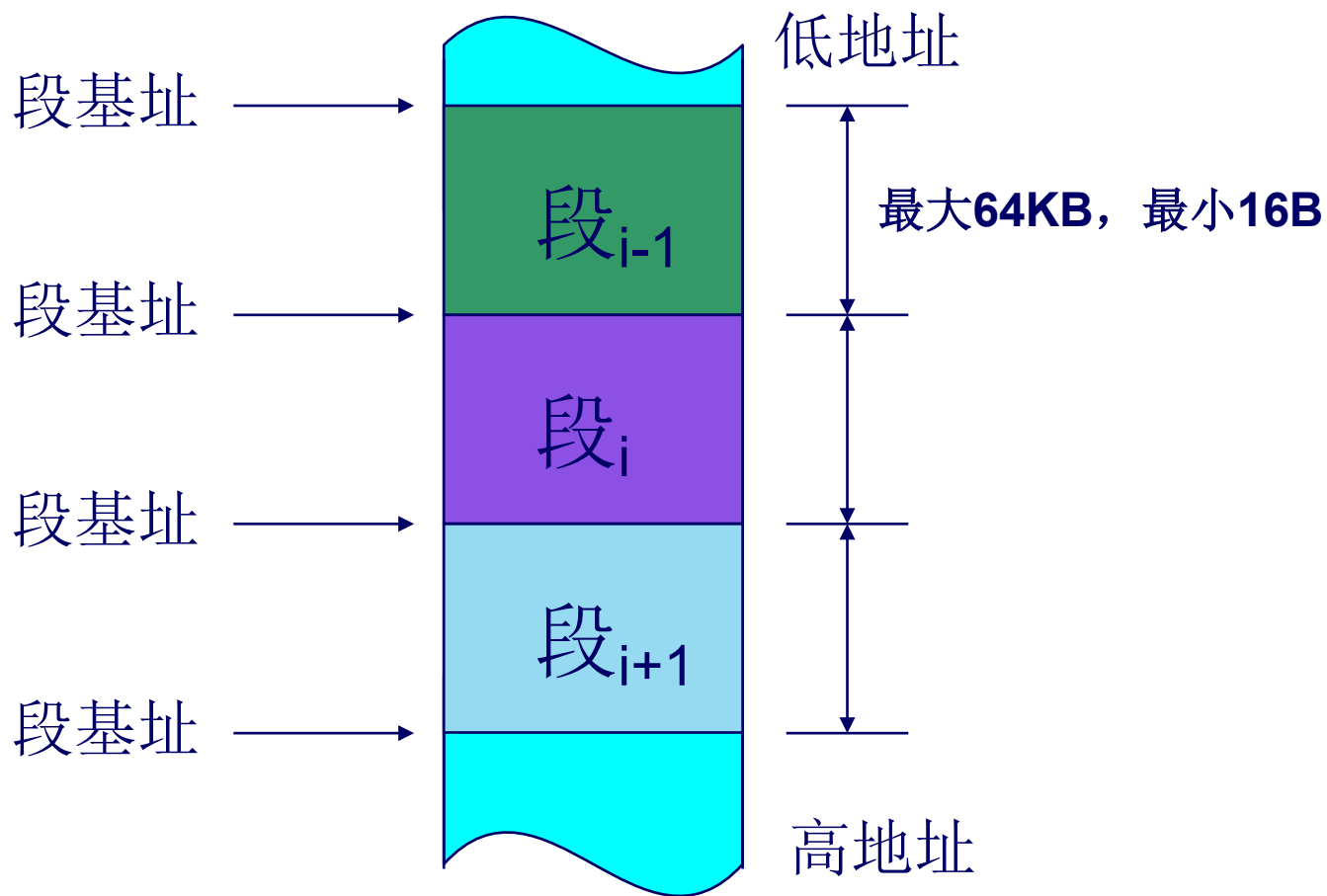
- 决定存储单元在内存中的位置

- 相对地址（偏移地址）

- 决定该存储单元相对段内第一个单元的距离

- 逻辑段的起始地址称为段首

- 每个逻辑段内的第一个单元 → **段首的偏移地址 = 0**





➤ 存储器的编址（3）

■ 物理地址：

- 内存单元在整个内存空间中的唯一地址

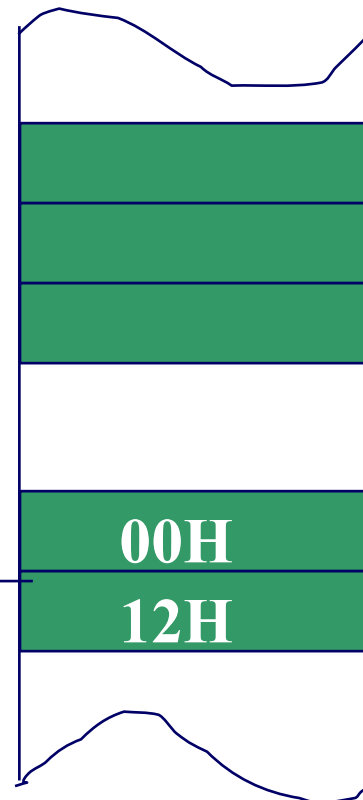
❖ 例：

- 段基地址 = **6000H**
- 段首地址
- 偏移地址 = **0009H**
- 物理地址

60000H

9

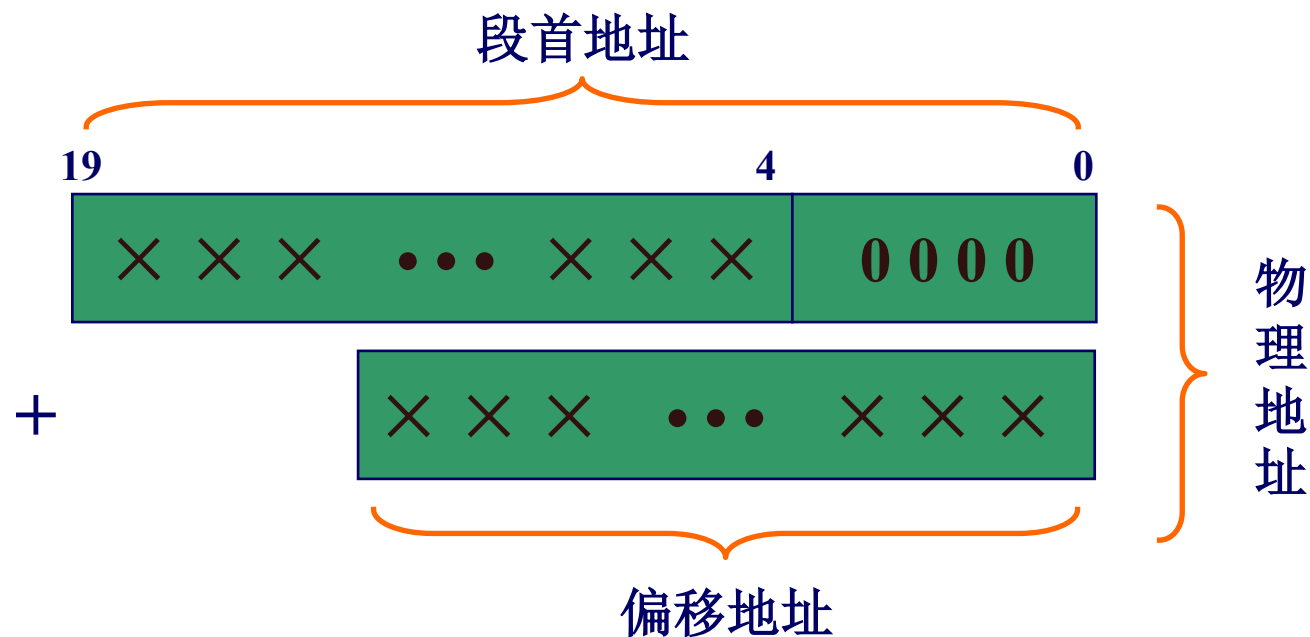
60009H



数据段

➤ 实地址模式

- 内存物理地址由段基地址和偏移地址组成



物理地址=段基地址×16+偏移地址



➤ 例：已知 CS=1055H, DS=250AH, ES=2EF0H, SS=8FF0H
，画出各段在内存中的分布。

➤ CS=1055H

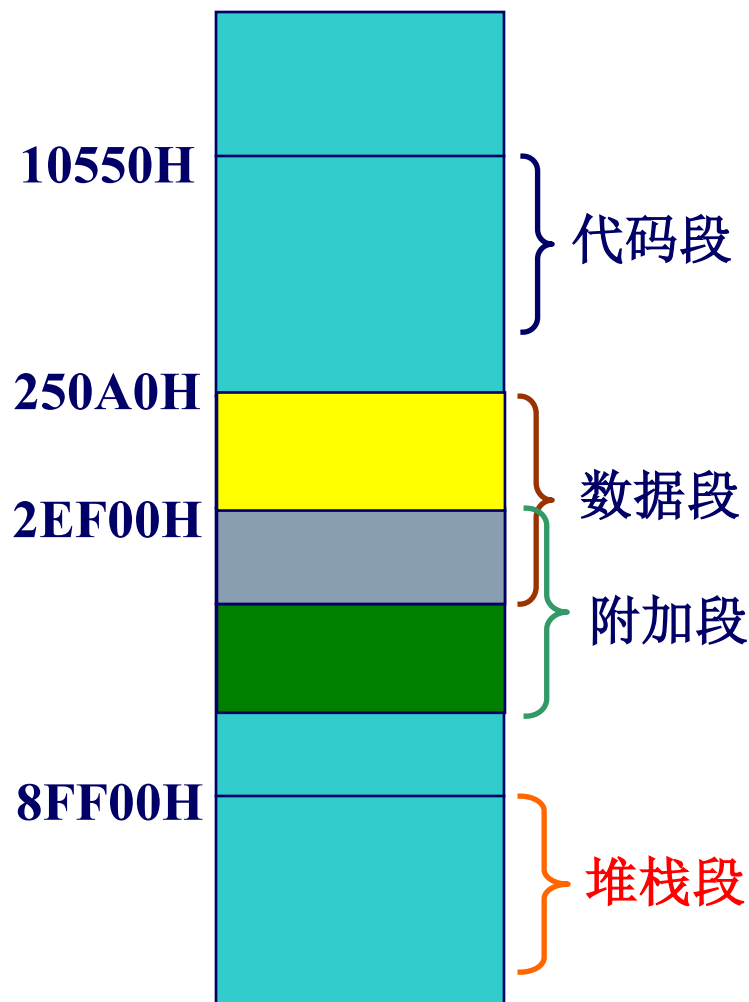
● 段首地址=10550H

➤ DS=250AH

● 段首地址=250A0H

➤ ES=2EF0H

➤ SS=8FF0H





➤ 例：设某操作数存放在数据段，DS=250AH，数据所在单元的偏移地址=0204H。则该操作数所在单元的物理地址为：

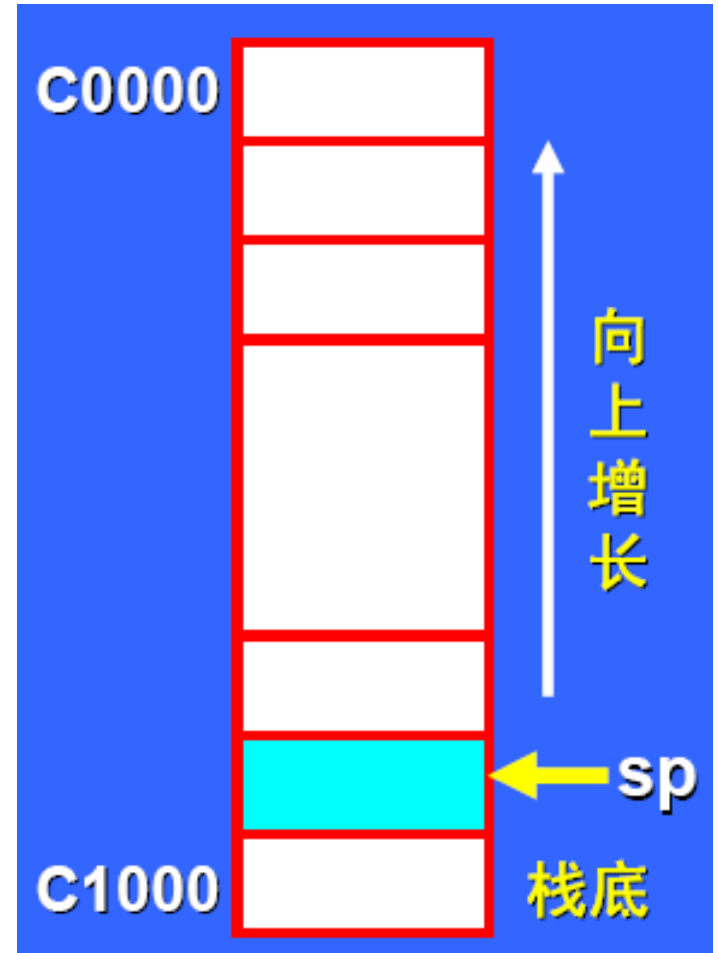
$$\blacksquare 250AH \times 16 + 0204H = 252A4H$$



➤ 堆栈及堆栈段的使用

● 堆栈:

- ❑ 内存中一个按**FILO**方式操作的特殊区域。
- ❑ 每次压栈和退栈均以**WORD**为单位
- ❑ **SS**存放堆栈段地址（段基址），**SP**存放段内偏移（指向栈顶），**SS:SP**构成了堆栈指针
- ❑ 堆栈用于存放返回地址、过程参数或需要保护的数据
- ❑ 常用于响应中断或子程序调用





- ① 工作方式：“先进后出”，指令为PUSH、POP，栈顶指针SP的变化由CPU自动管理。
- ② 操作特点：以字为单位进行操作，每访问一次堆栈就能压入 / 弹出一个字。
- ③ 地址增长方式一般是向上增长：栈底设在存储器的高地址区，堆栈地址由高向低增长。



- 执行PUSH指令时，CPU自动修改指针 $SP-2 \longrightarrow SP$ 。使SP指向新栈顶，然后将低位数据压入(SP)单元，高位数据压入(SP+1)单元。
- 当执行POP指令时，CPU先将当前栈顶SP(低位数据)和SP+1(高位数据)中的内容弹出，然后再自动修改指针，使 $SP+2 \longrightarrow SP$ ，SP指向新栈顶。



➤ 例:

● 已知

□ $SS=1000H$, $SP=0100H$

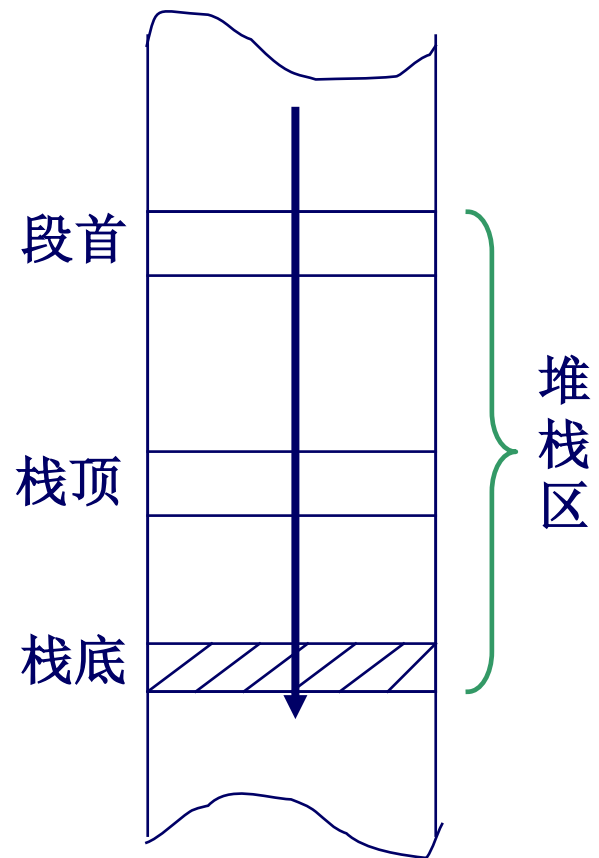
● 则:

□ 堆栈段的段首地址= **10000H**

□ 栈顶（偏移）地址= **0100H**

● 若该段最后一个单元地址为 $10200H$ ，则:

□ 栈底偏移地址= **0200H**





压栈指令

PUSH src ; src为16位操作数

例：PUSH AX ; 将AX内容压栈

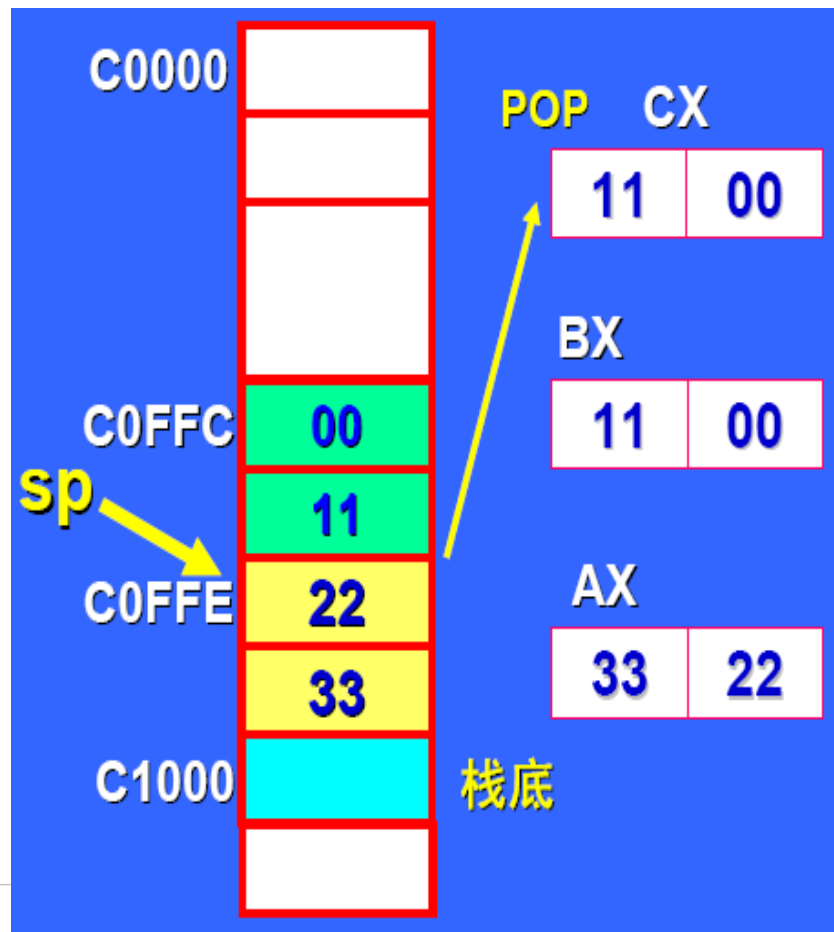
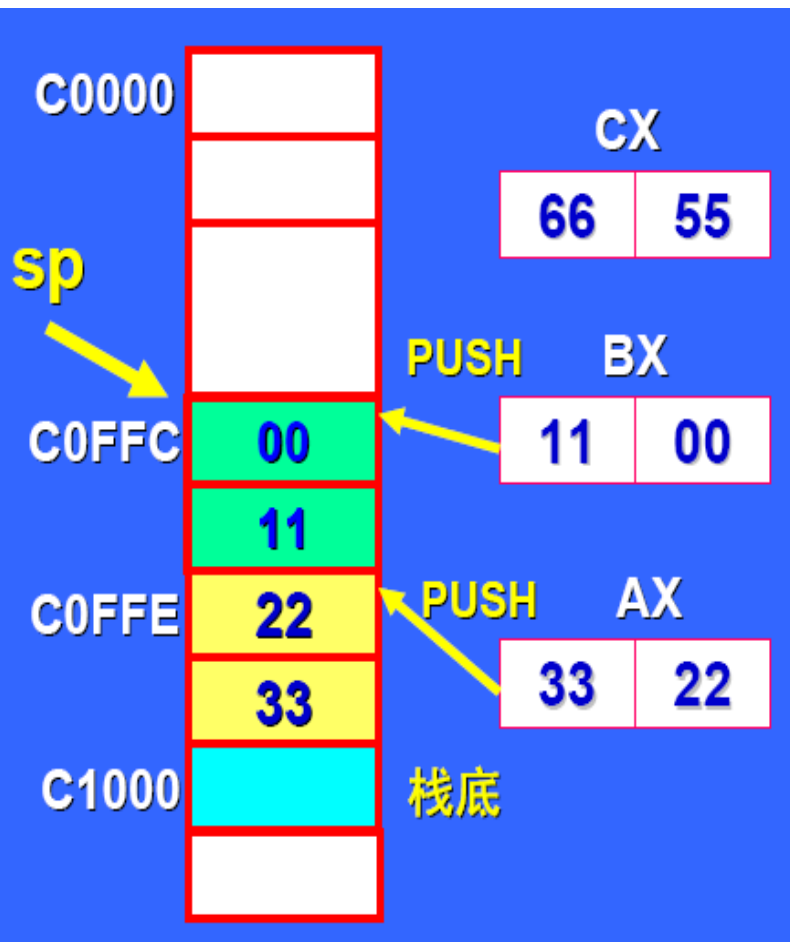
执行操作： (SP) - 1 ← 高字节AH

(SP) - 2 ← 低字节AL

(SP) ← (SP) - 2



【例】. 假如当前 $SS=C000H$, $SP=1000H$, 若 $AX=3322H$, $BX=1100H$, $CX=6655H$, 执行指令 *PUSH AX*, *PUSH BX*, 再执行指令 *POP CX*, 此时堆栈中内容发生什么变化, AX , BX , CX 中的内容是什么?





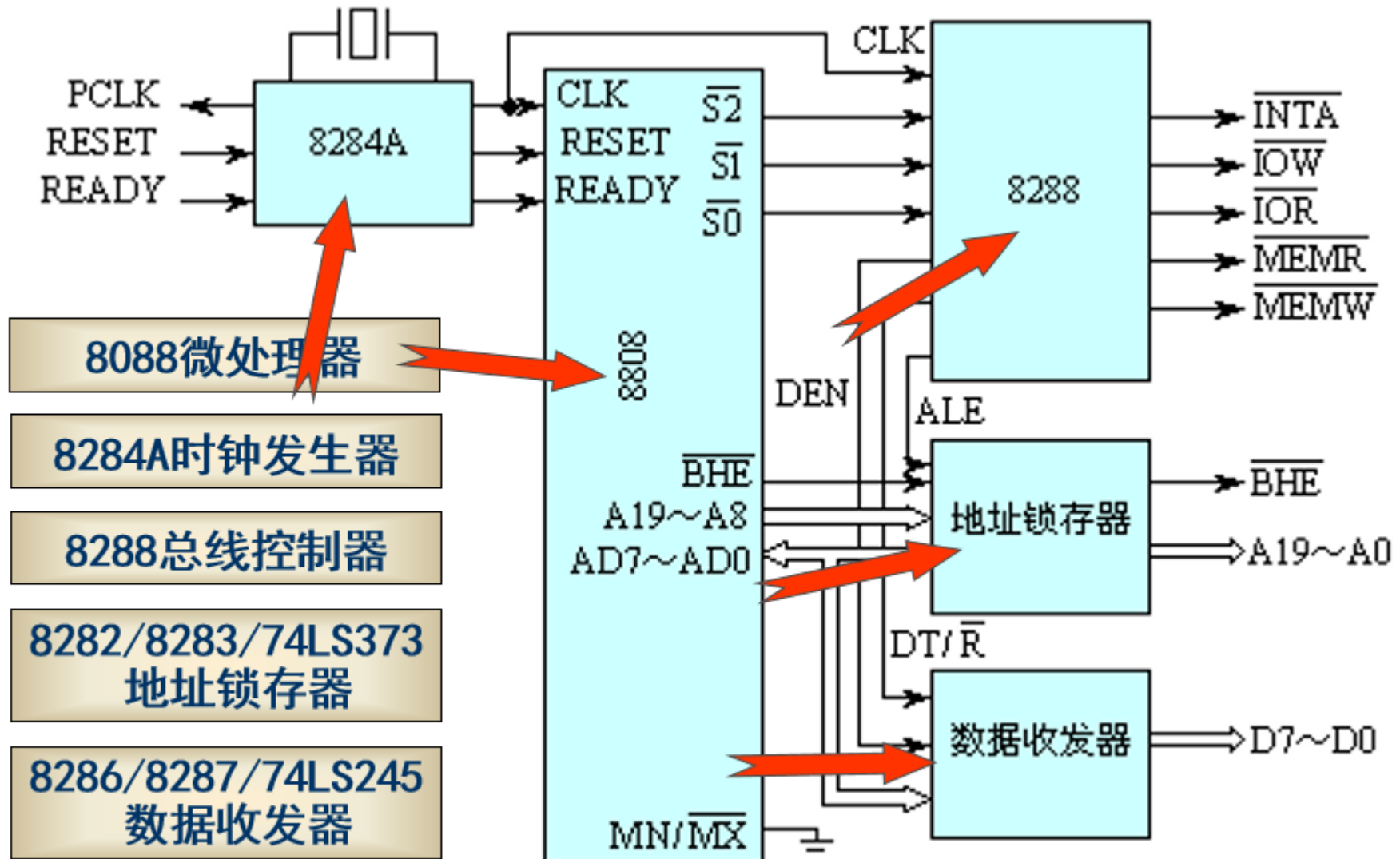
2.6 IBM PC/XT微型计算机系统简介

❖ IBM PC/XT的核心是安装在机箱底部的系统主板，
系统主板上的电路可以分成四个主要功能模块：

- 1、CPU及辅助器件构成的CPU子系统
- 2、ROM和RAM构成的存储器子系统
- 3、各种I/O芯片构成的接口部件子系统
- 4、连接各种外设适配器以及存储器扩充板的PC总线
I/O扩展槽

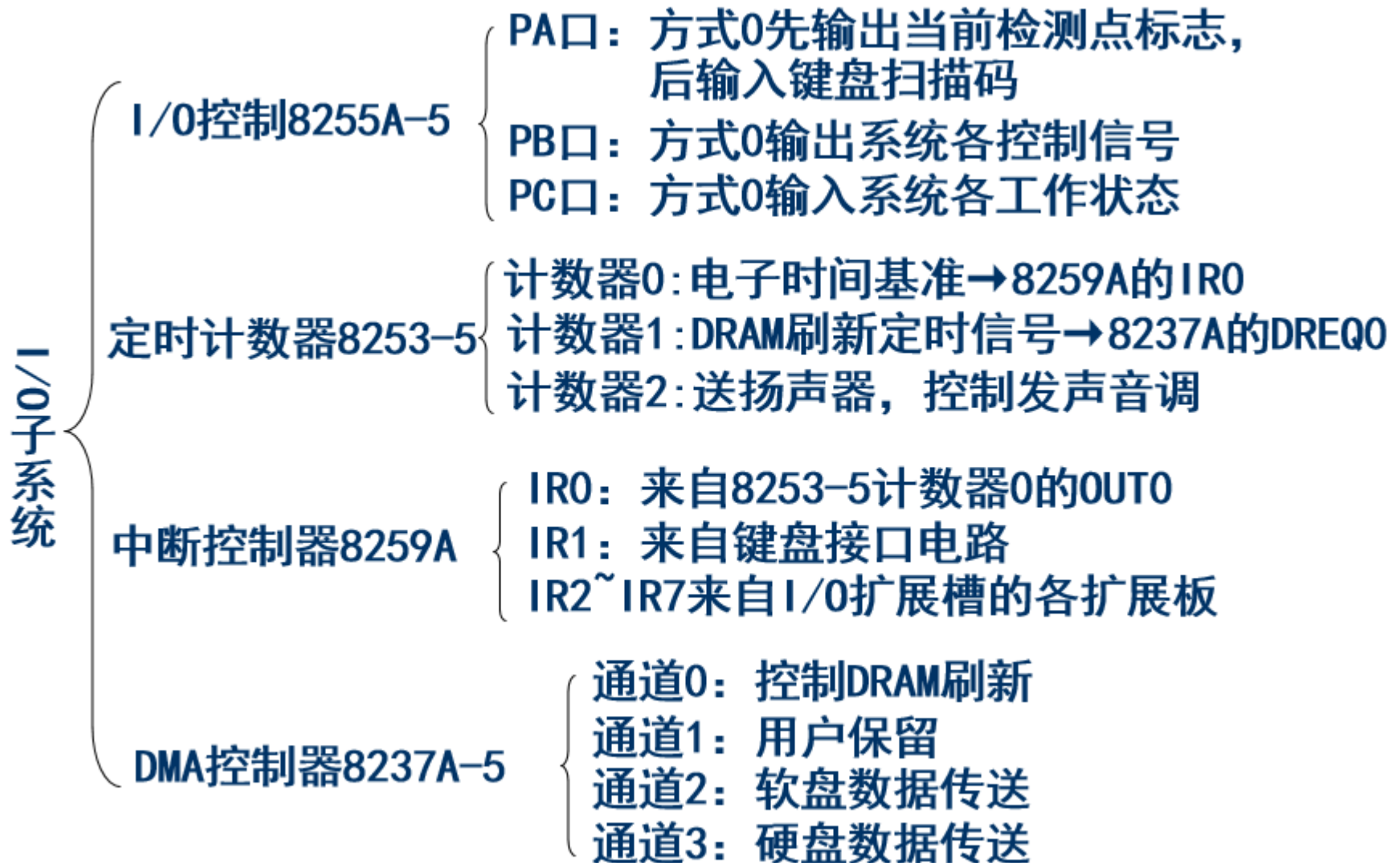


2.6.1 CPU子系统



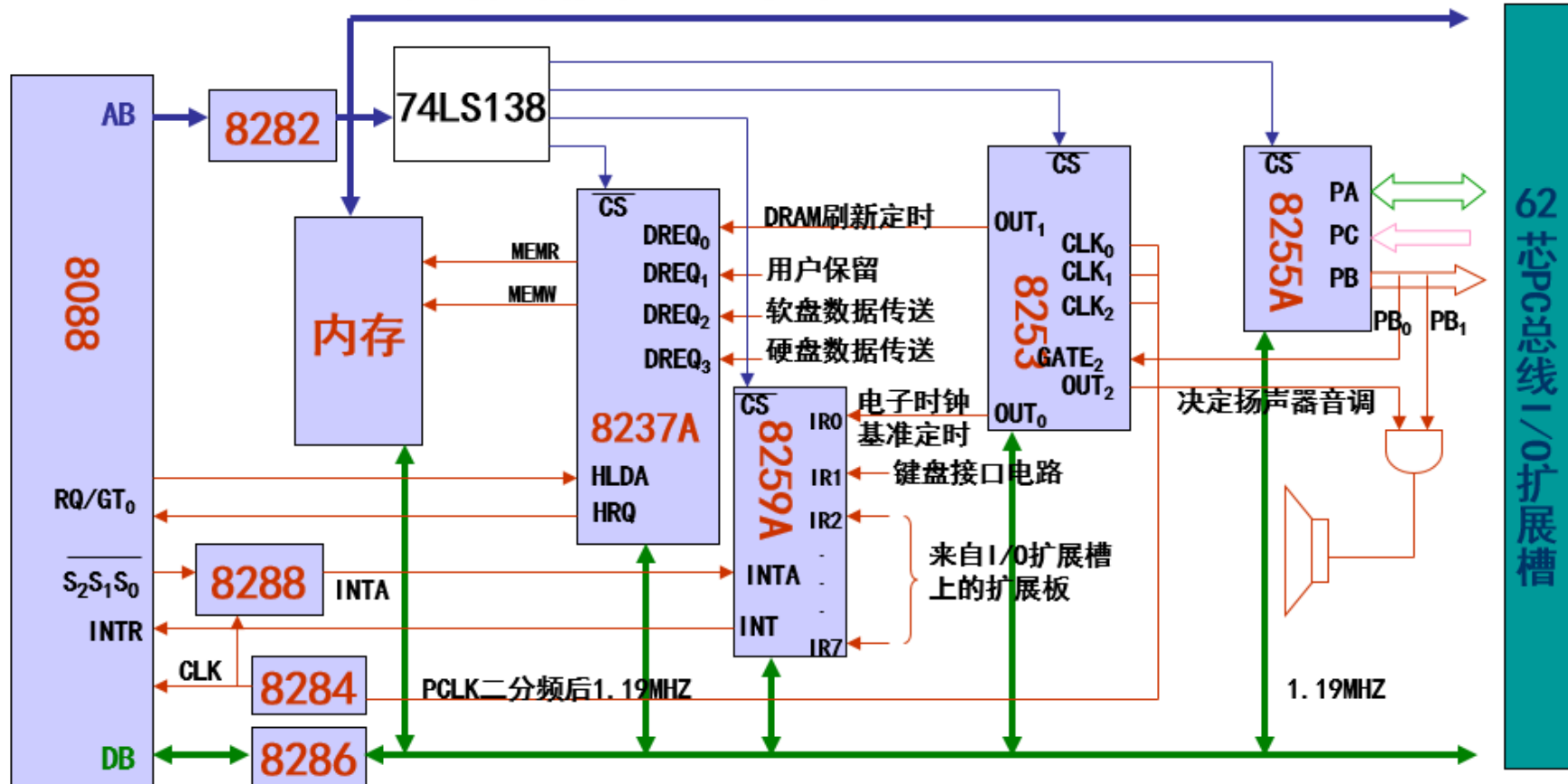


2.6.2 I/O子系统



2.6.3 系统板部件框图

IBM-PC/XT系统板部件框图





2.7 本章小结

- 对微处理器相关概念及技术进行了概述。
- 介绍了80X86架构（8086/8088、80286、80386、80486）、P5架构（Pentium、Pentium MMX）、P6架构（Pentium Pro、Pentium II和Pentium III）、NetBurst架构（Pentium4）、Core架构、Itanium架构等处理器。
- 着重围绕8086/8088介绍了其内部结构、引脚时序、存储管理。

