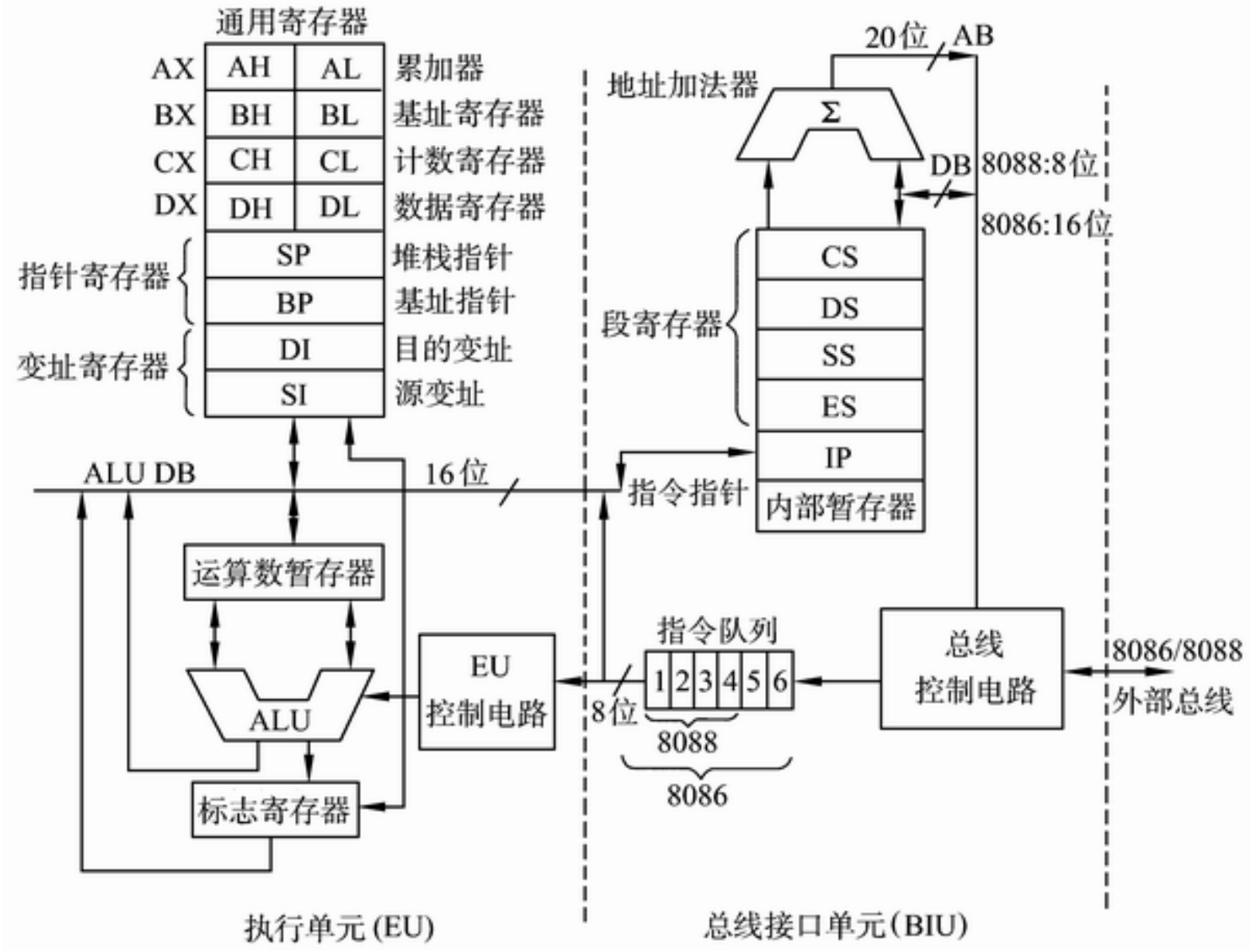
第2章 16位微处理器Intel 8086

- 1978年，Intel公司推出8086：
- 16位微处理器，兼容8085
- CISC结构
- 单一+5V供电
- 频率4.77MHz-10MHz
- 内部数据总线和外部数据总线都是16位，地址总线为20位，可最大寻址1MB的存储空间
- 双列直插式封装DIP，40个引脚
- 内存采用分段的管理方式。

- 1979年，推出成本更低的8088。
- 8088芯片最早于1981年用于IBM PC/XT。

2.1 8086微处理器的构成

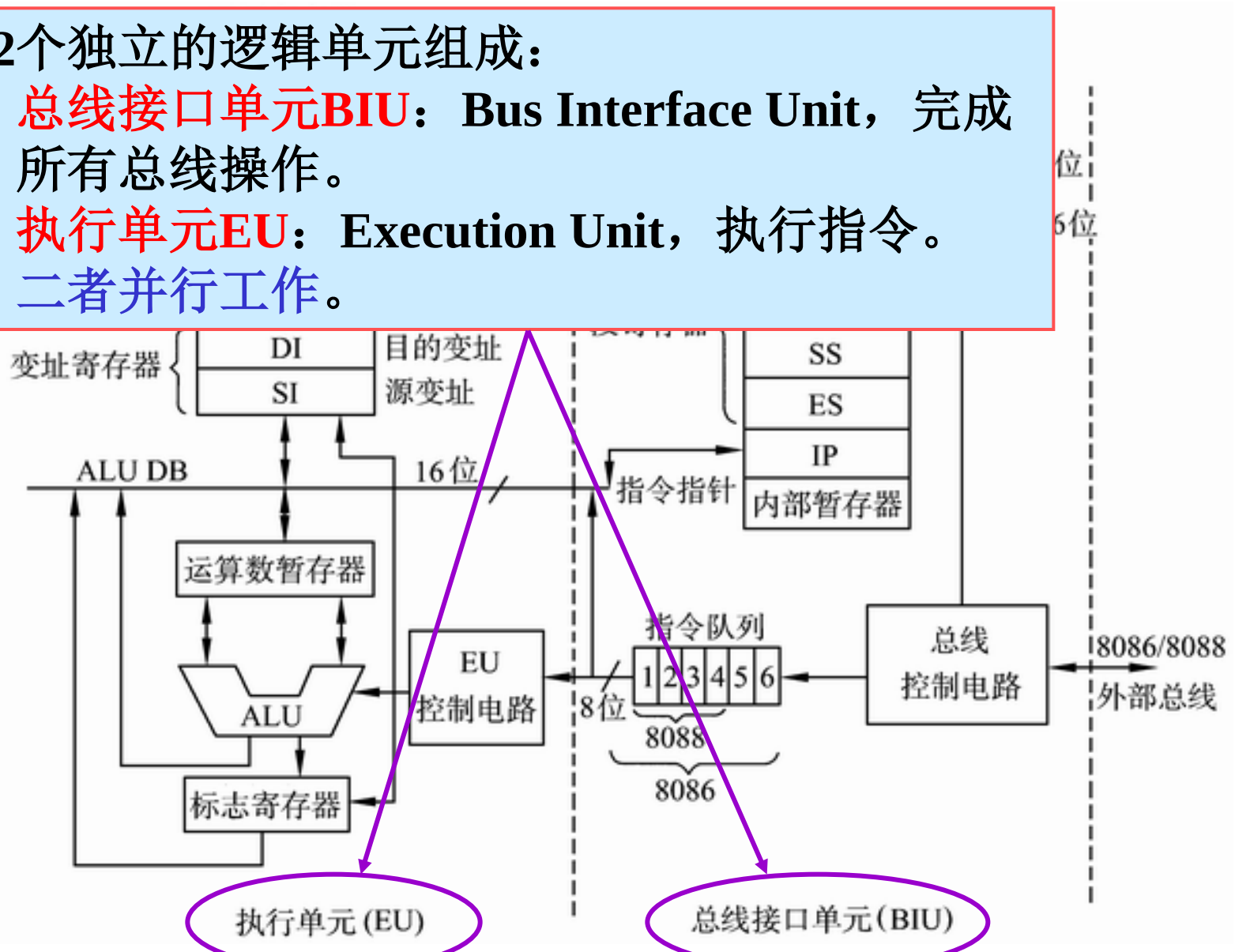


8086内部结构

由2个独立的逻辑单元组成：

1. 总线接口单元BIU：Bus Interface Unit，完成所有总线操作。
2. 执行单元EU：Execution Unit，执行指令。

➢ 二者并行工作。

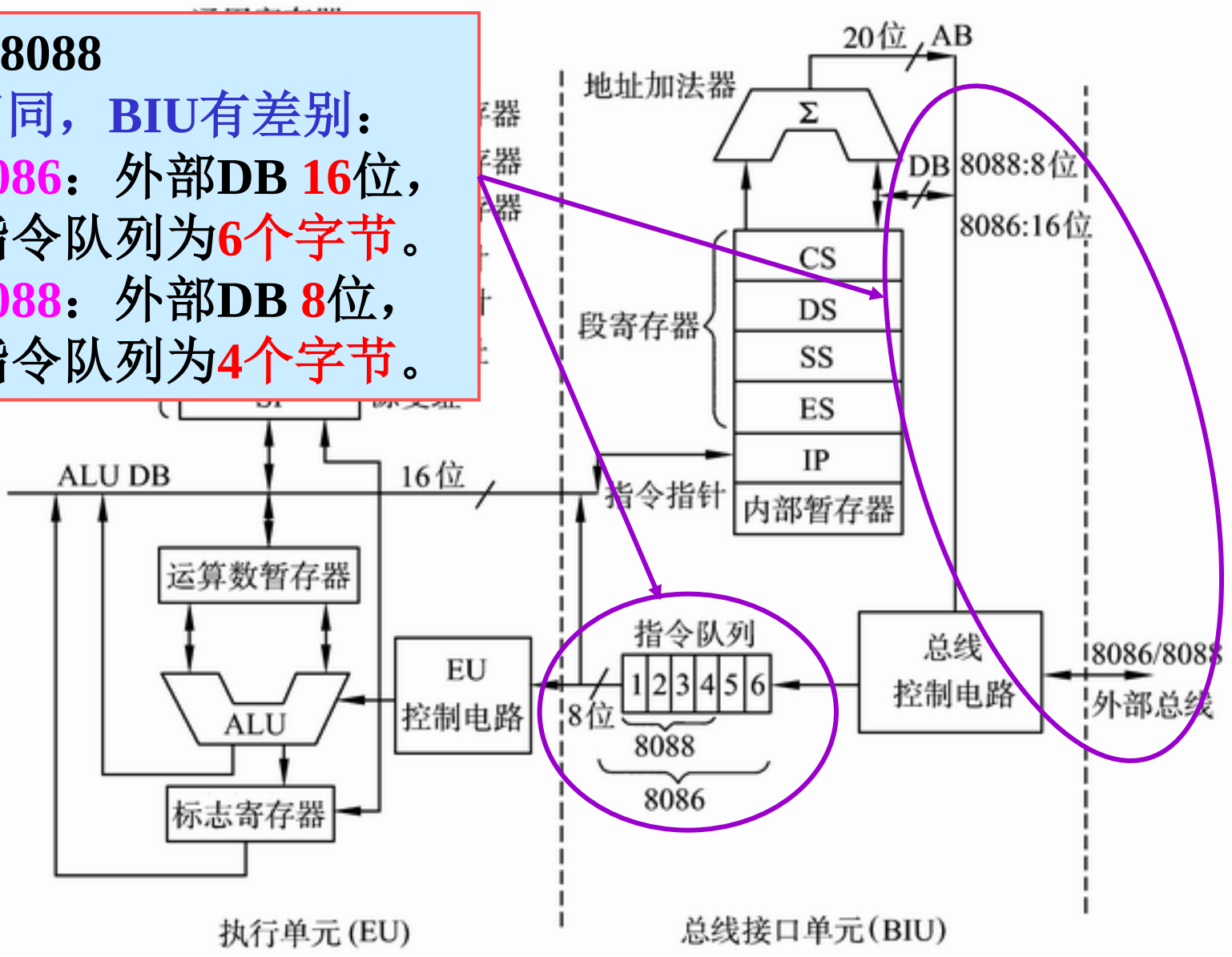


8086/8088差别

8086/8088

EU相同，BIU有差别：

1. **8086：外部DB 16位，指令队列为6个字节。**
2. **8088：外部DB 8位，指令队列为4个字节。**



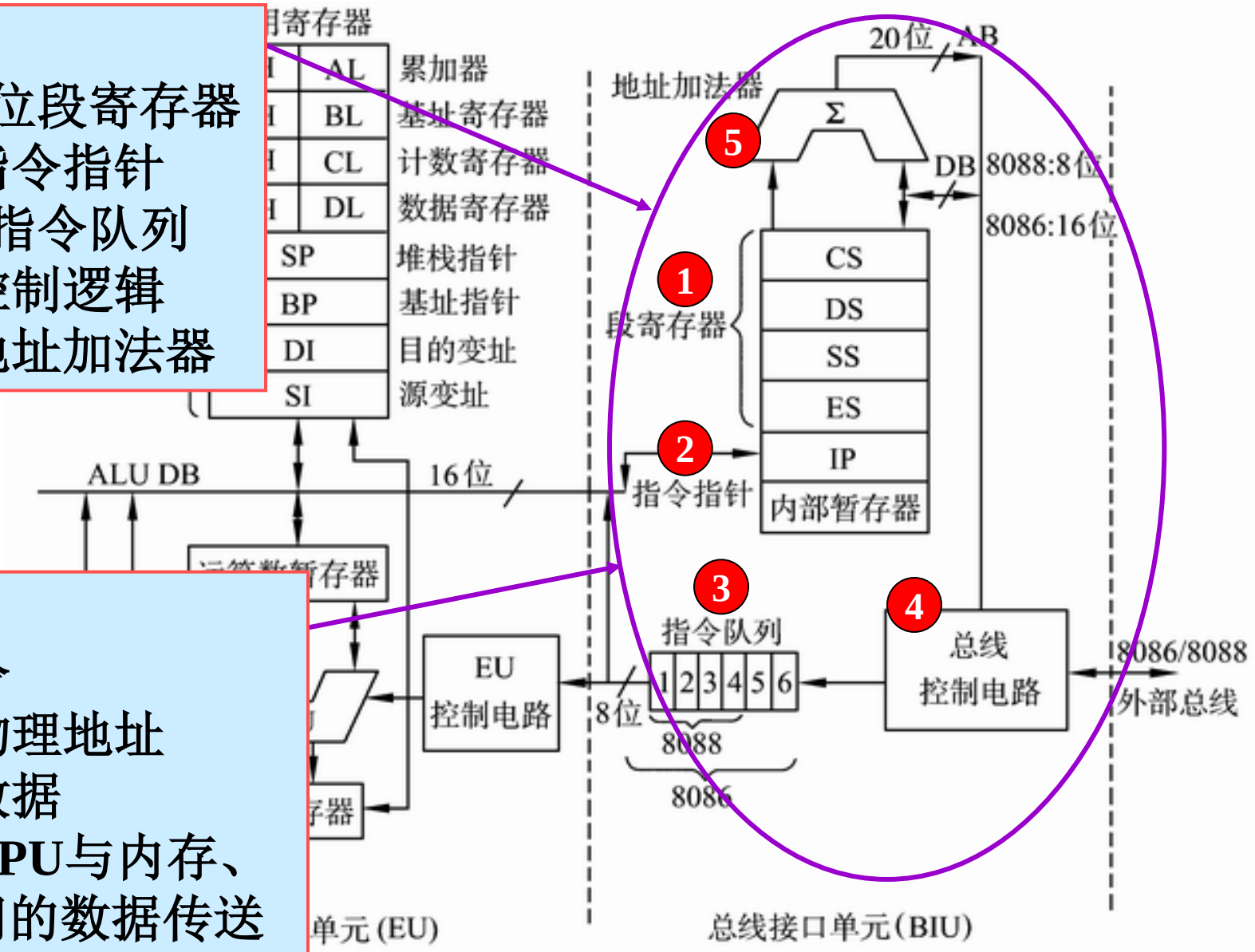
总线接口单元BIU

BIU组成：

1）4个16位段寄存器
2）16位指令指针
3）6字节指令队列
4）总线控制逻辑
5）20位地址加法器

BIU功能：

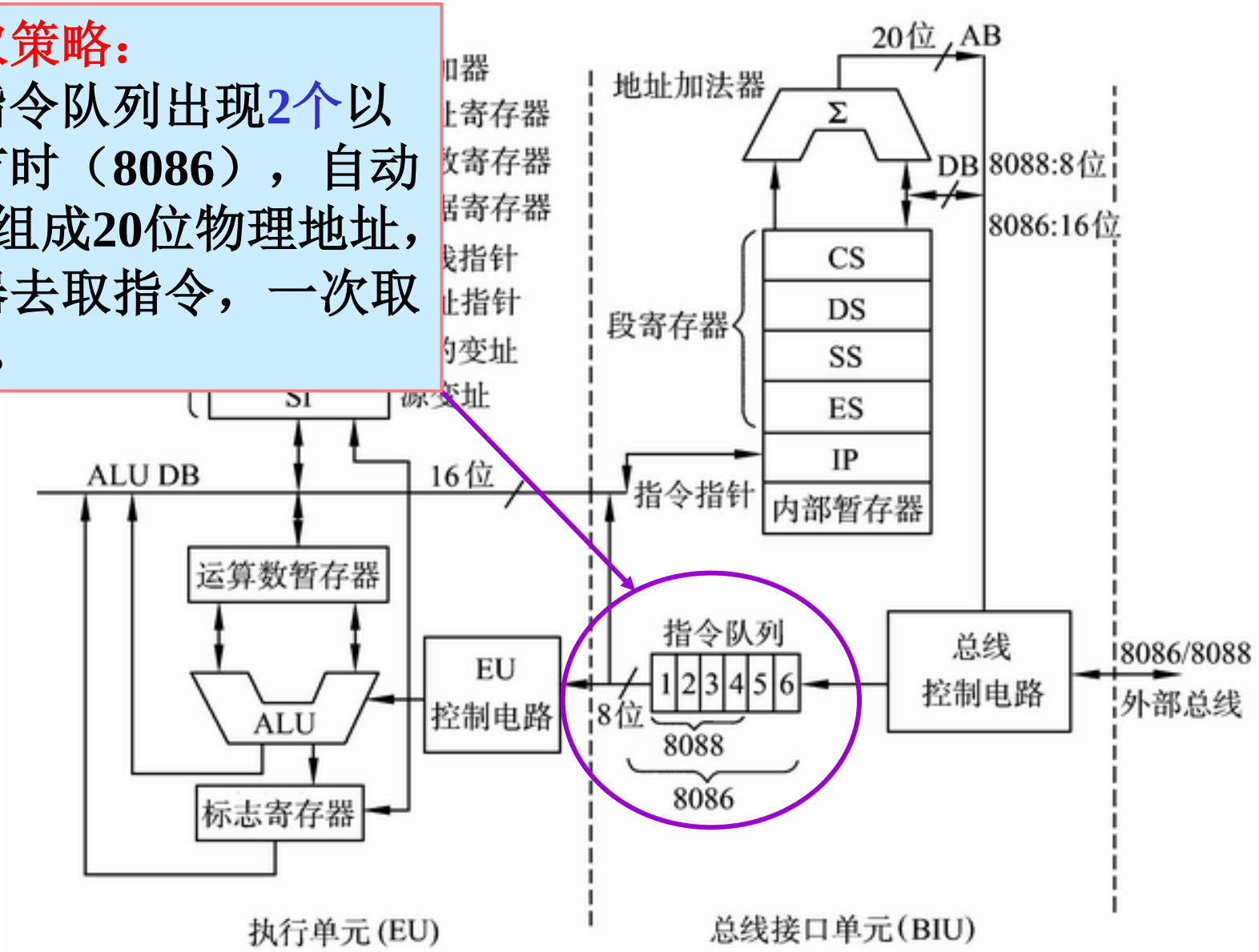
1）取指令
2）形成物理地址
3）传送数据
实现CPU与内存、I/O端口间的数据传送



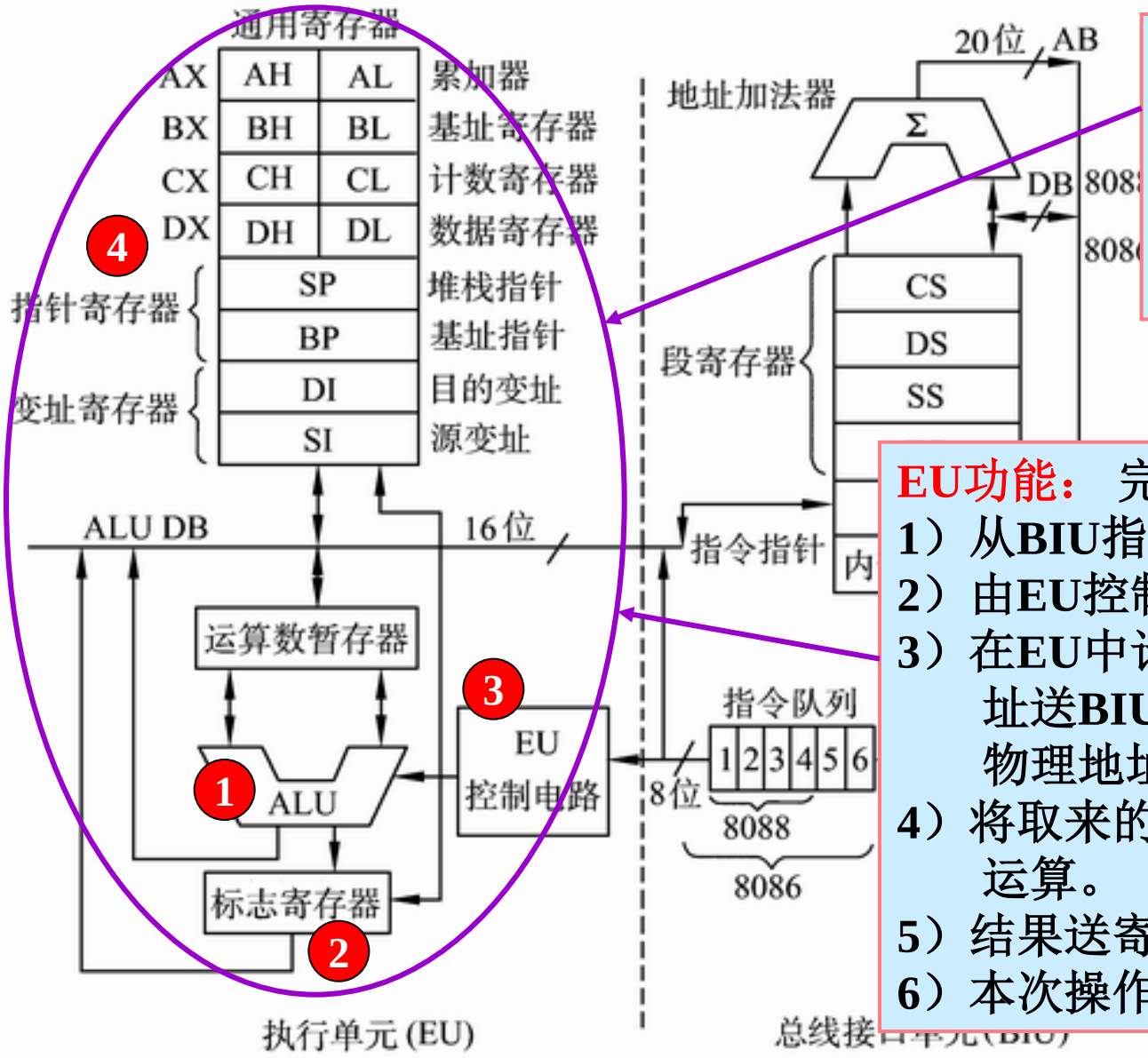
指令预取策略

指令预取策略：

当指令队列出现2个以上空字节时（8086），自动按CS:IP组成20位物理地址，到存储器去取指令，一次取2个字节。



执行单元EU



EU组成：

1）16位ALU
2）16位标志寄存器
3）EU控制电路
4）8个16位通用寄存器

EU功能： 完成指令所规定的操作

1）从BIU指令队列读取指令。
2）由EU控制电路译码分析。
3）在EU中计算操作数的16位偏移地址送BIU，由BIU的∑形成20位物理地址。
4）将取来的操作数送ALU进行指令运算。
5）结果送寄存器或送BIU放到内存
6）本次操作状态放到标志寄存器中

8086微处理器结构特点

8086以前的处理器

CPU	取指 1	执行 1	存结果 1	取指 2	执行 2	取指 3	取操作数 3	执行 3
BUS	忙	闲	忙	忙	闲	忙	忙	闲

8086处理器

EU		执行 1	执行 2		执行 3	执行 3	执行 4	
BIU	取指 1	取指 2	存结果 1	取指 3	取操作数 3	取指 4	存结果 3	取指 5
BUS	忙	忙	忙	忙	忙	忙	忙	忙

8086内部由BIU、EU两个独立单元，可独立完成总线操作和执行指令的任务，即两个单元可重叠工作，总线利用率提高，降低存储器的存取速度要求，这种重叠技术以前只在大型机中使用。

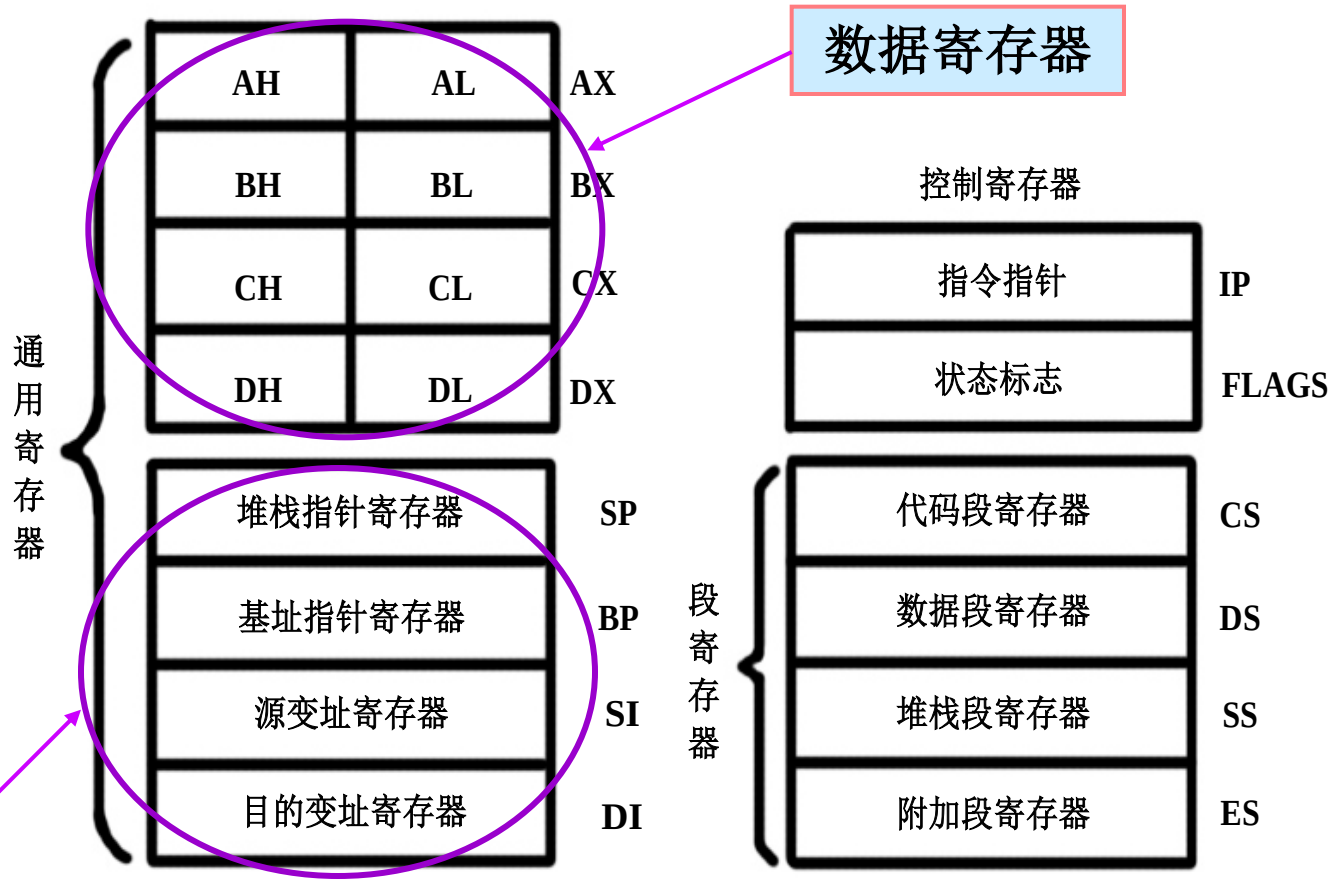
8086结构特点：

1）由BIU、EU两个独立单元组成。

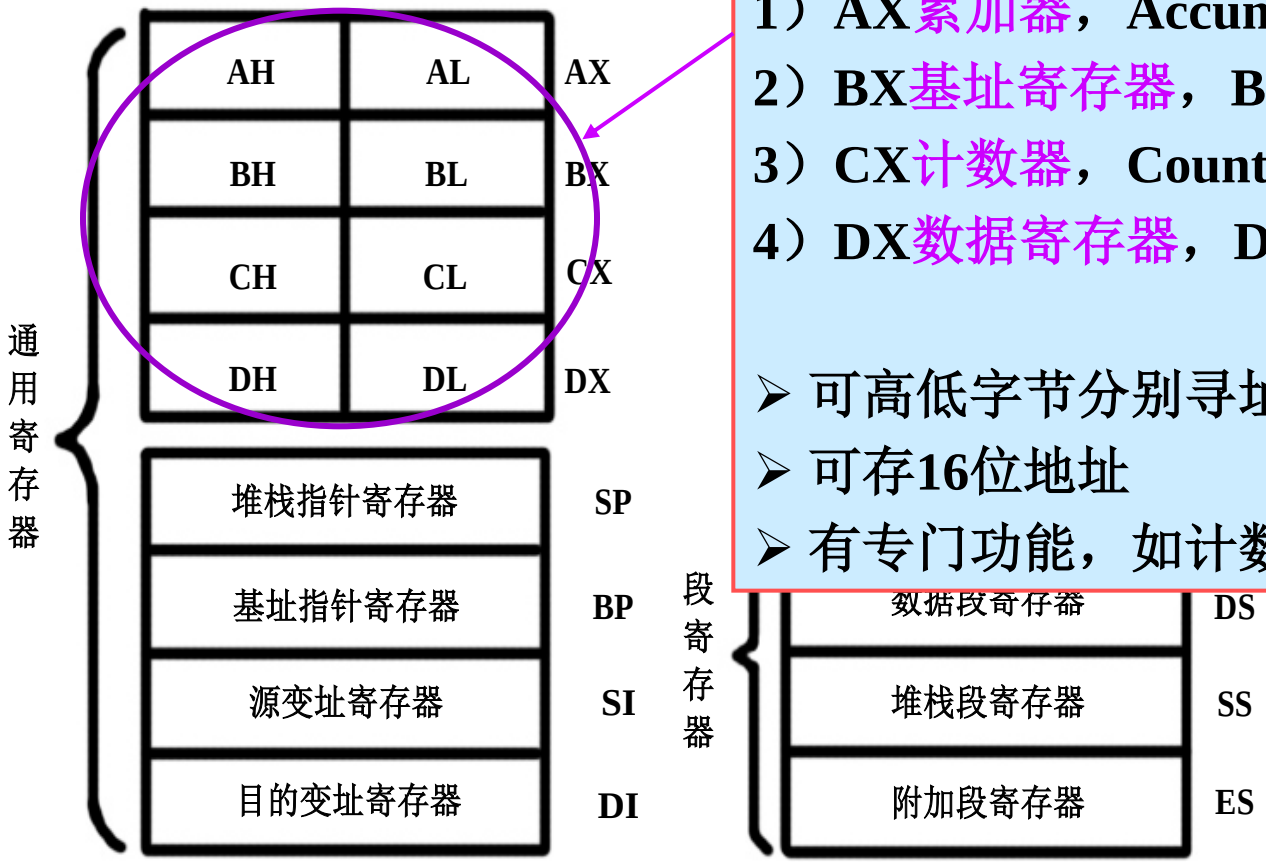
2）取指和执行重叠。

2.2 8086的编程结构

8086 CPU有14个16位寄存器



数据寄存器



数据寄存器：存放操作数据及中间结果。

1）AX累加器，Accumulator Register

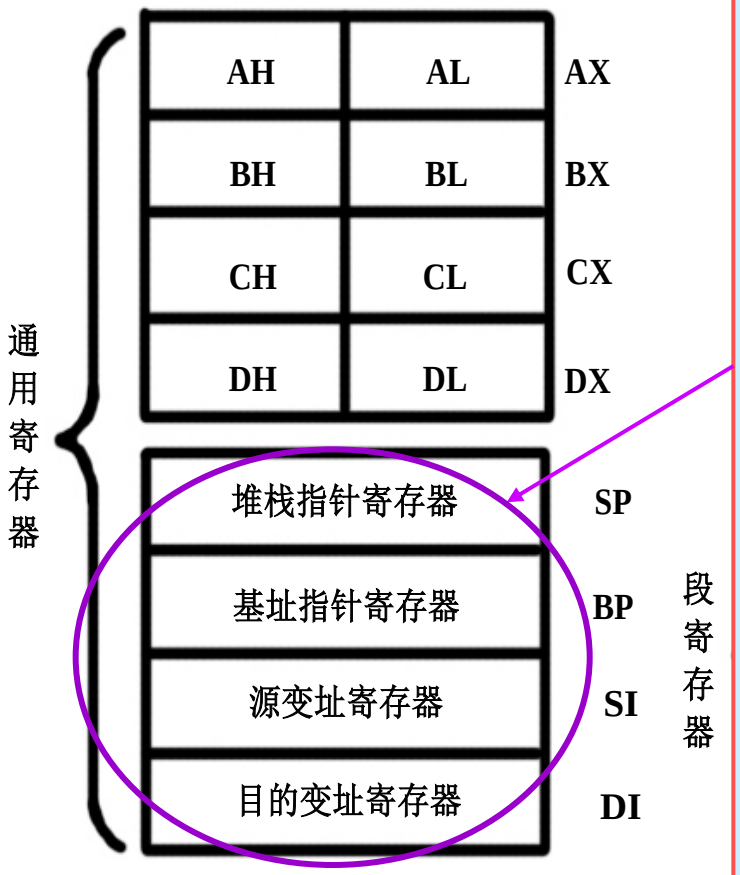
2）BX基址寄存器，Base Register

3）CX计数器，Counter

4）DX数据寄存器，DataRegister

- 可高低字节分别寻址
- 可存16位地址
- 有专门功能，如计数器、累加器等

指针及变址寄存器



指针及变址寄存器：存放逻辑地址的偏移量，是20位物理地址的组成部分

1） SP堆栈指针，Stack Pointer，指示在堆栈段中栈顶的位置，用于数据进栈、出栈的位置指向。

2） BP基址指针，Base Pointer，指示在堆栈段中一个数据区的基址位置，用于访问堆栈段中的某个数据。

3） SI源变址寄存器，Source Index，源串指针，指示在数据段中一个源串操作数的位置。

4） DI目的变址寄存器，Destination Index，目的串指针，指示在附加段中一个目的串操作数的位置。

➢ SI、DI用于数据串操作，且隐含使用。

段寄存器

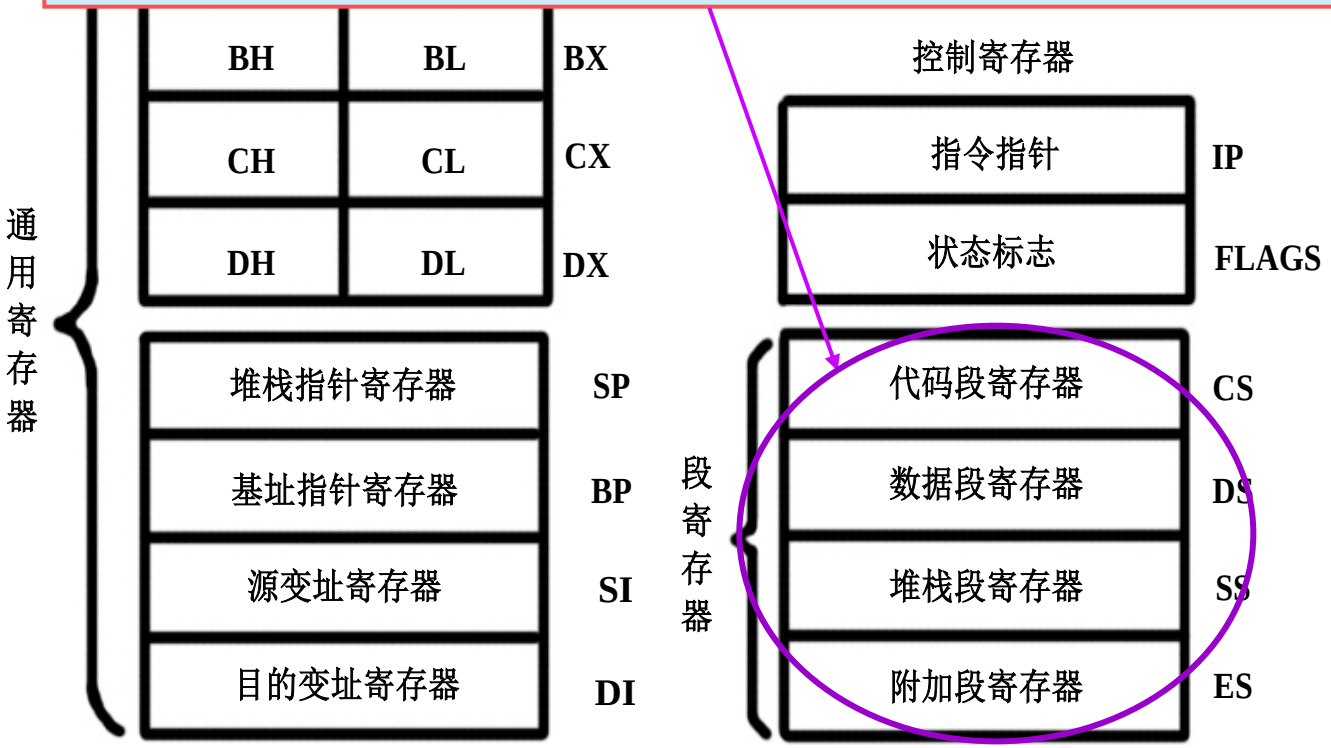
段寄存器：存放当前段的段起始地址

CS代码段寄存器，Code Segment，存放当前执行程序的段起始地址

SS堆栈段寄存器，Stack Segment，存放当前堆栈段的段起始地址

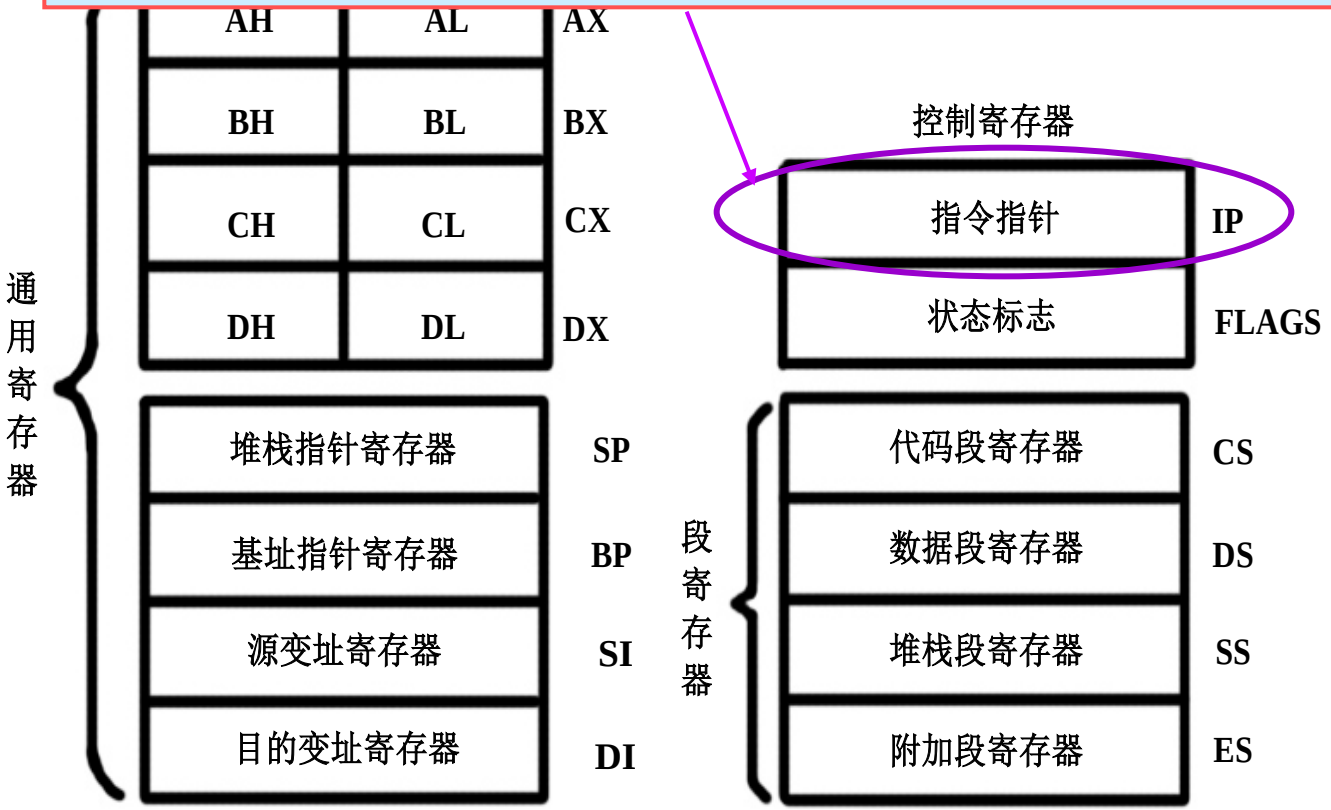
DS数据段寄存器，Data Segment，存放当前数据段的段起始地址

ES附加段寄存器，Extra Segment，存放当前附加段的段起始地址



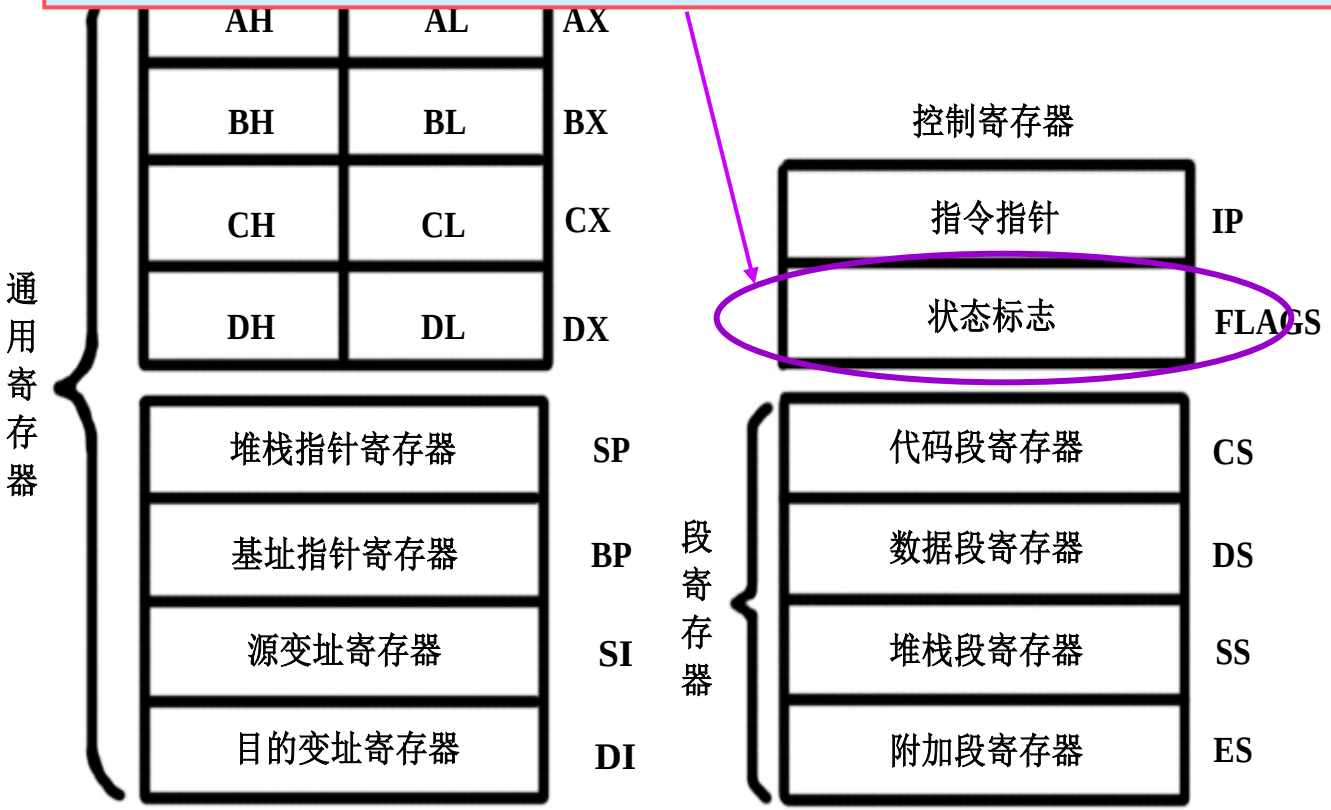
指令指针

- IP指令指针，Instruction Pointer，存放下一次要取出的指令的偏移地址
- CS:IP指示下一条要取出的指令的实际地址
- IP不能被程序直接存取，由BIU来修改，类似于程序计数器PC（Program Counter）



标志寄存器

Flag标志寄存器：存放ALU状态和对CPU的控制状态
16位，定义9位，分2类：
1）状态标志：6位，CF、OF、ZF、SF、PF、AF，上次操作后ALU状态。
2）控制标志：3位， IF、DF、TF，人为设置，控制CPU操作。



标志寄存器中各位含义-状态位

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

1）CF进位，Carry Flag：结果有进位或借位，CF=1，否则CF=0。

2）PF奇偶，Parity Flag：低8位结果为偶数个1，PF=1，否则PF=0。

3）AF辅助进位，Auxiliary Carry Flag：D3有进位或借位，AF=1，否则AF=0。

4）ZF零标志，Zero Flag：结果为0，ZF=1，否则ZF=0。

5）SF符号位，Sign Flag：结果为正，SF=0，否则SF=1。

6）OF溢出标位，Overflow Flag：有符号数算术运算，结果超出其所能表示的数值范围，OF=1，否则OF=0。

标志寄存器中各位含义-控制位

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

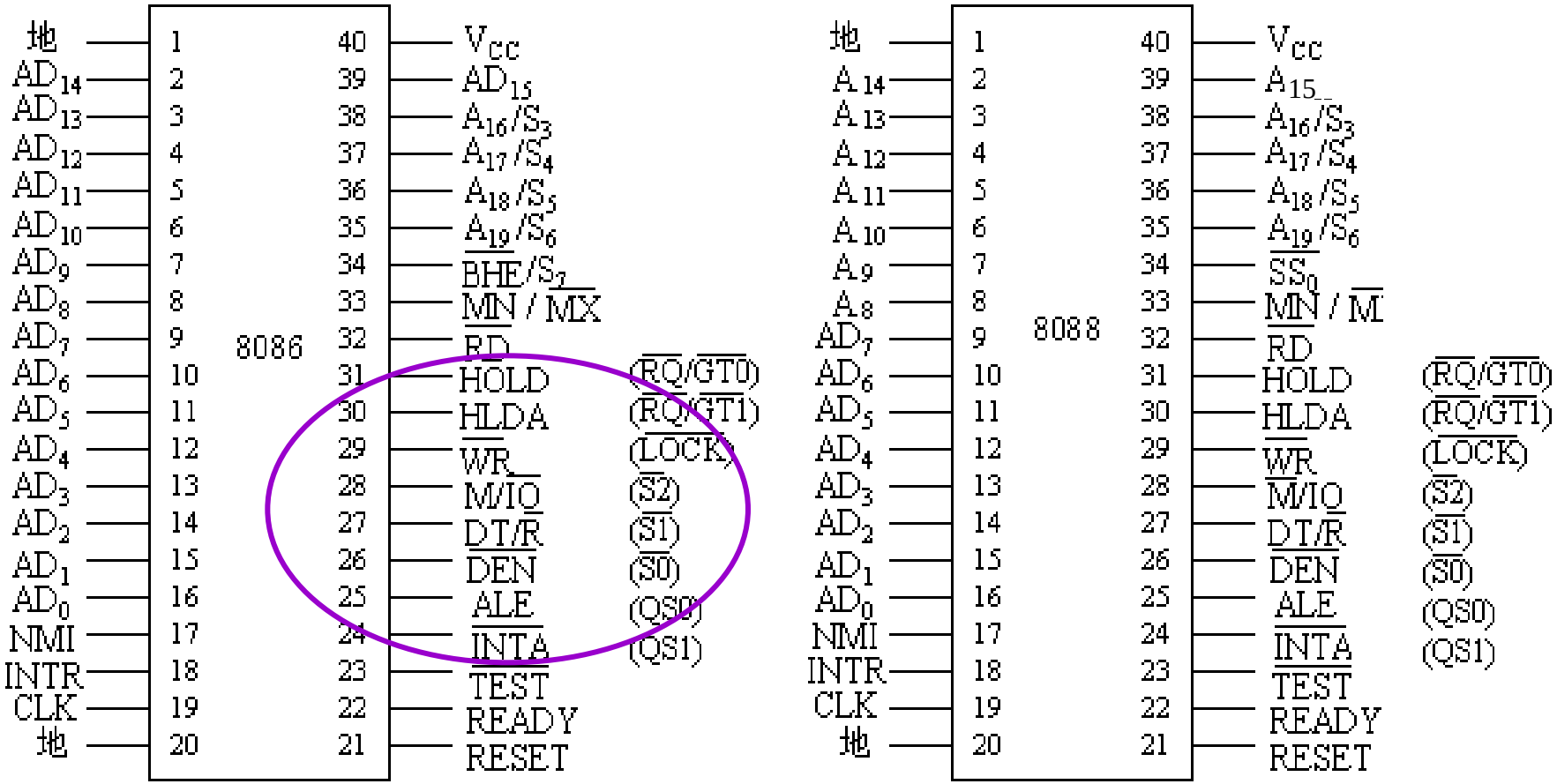
1）DF方向标志，Direction Flag：数据串操作地址修改方向控制，增址DF=0，减址DF=1。

2）IF中断允许，Interrupt Enable Flag：IF=1，允许可屏蔽中断（开中断），IF=0，关中断。

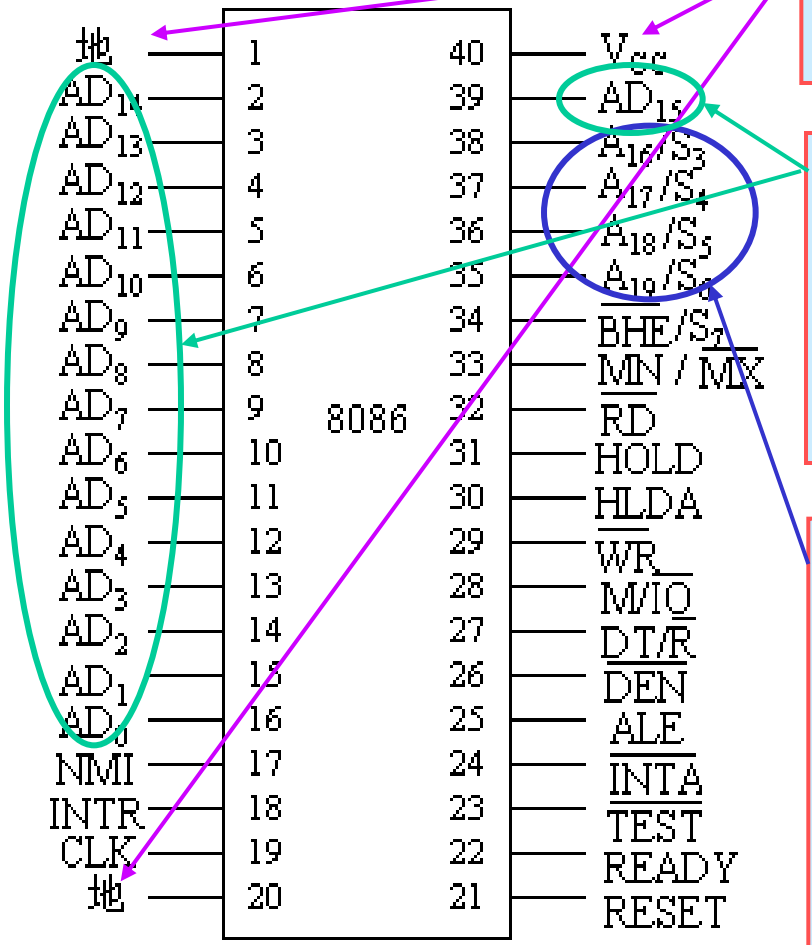
3）TF陷阱标志，Trap Flag：TF=1，单步方式，每执行一条指令自动产生一次类型为1的内部中断，使操作者可以逐条指令检查。 TF=0，正常。

2.3 8086外部引脚

8086采用40引脚双列直插式封装。
8086有两种工作模式：最大模式、最小模式。
有8个引脚在不同的模式下功能有所不同。



2.3.1 两种模式下功能相同的引脚



（1）**VCC、GND：** 电源、地，单一+5V电源，2个接地。

（2）**AD15-AD0：** 地址/数据，分时输出低16位地址及数据信号。经地址锁存器输出对应的地址信号为A15-A0。

（3）**A19/S6-A16/S3：** 地址/状态，分时输出高4位地址及状态信息。

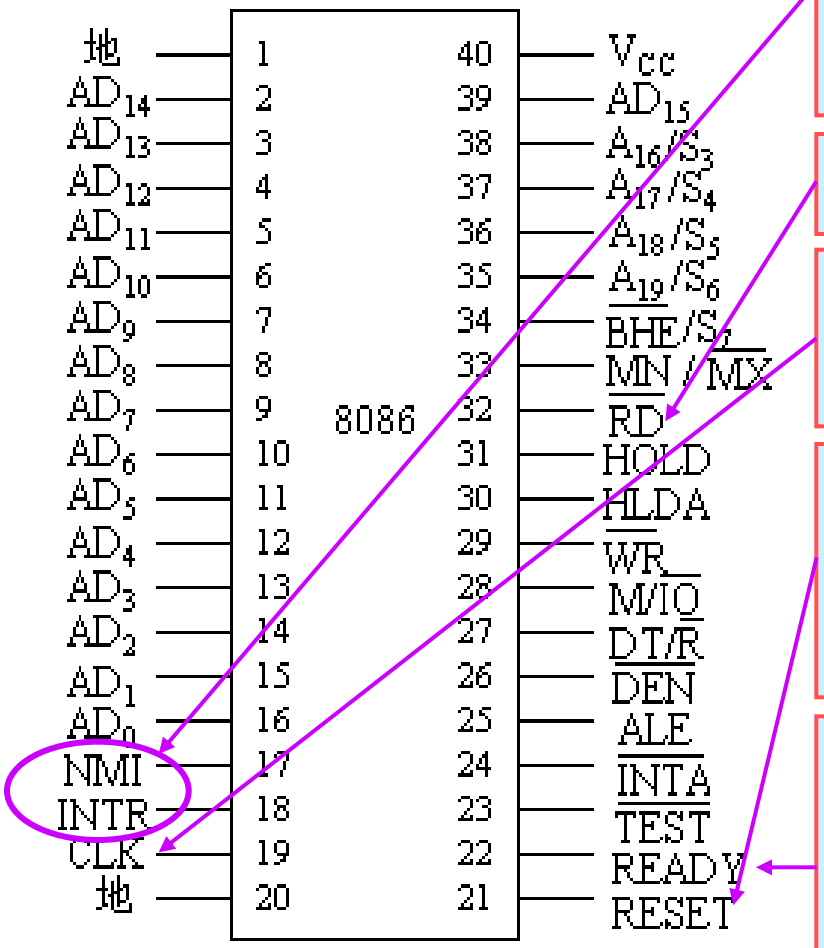
S6为0：CPU与总线连通

S5：等价于IF

S4、S3：使用哪个段寄存器

00=ES，01=SS，10=CS，11=DS

2.3.1 两种模式下功能相同的引脚



（4） **INTR：** 可屏蔽中断请求信号，高电平有效； **NMI：** 非屏蔽中断请求信号，上升沿有效。

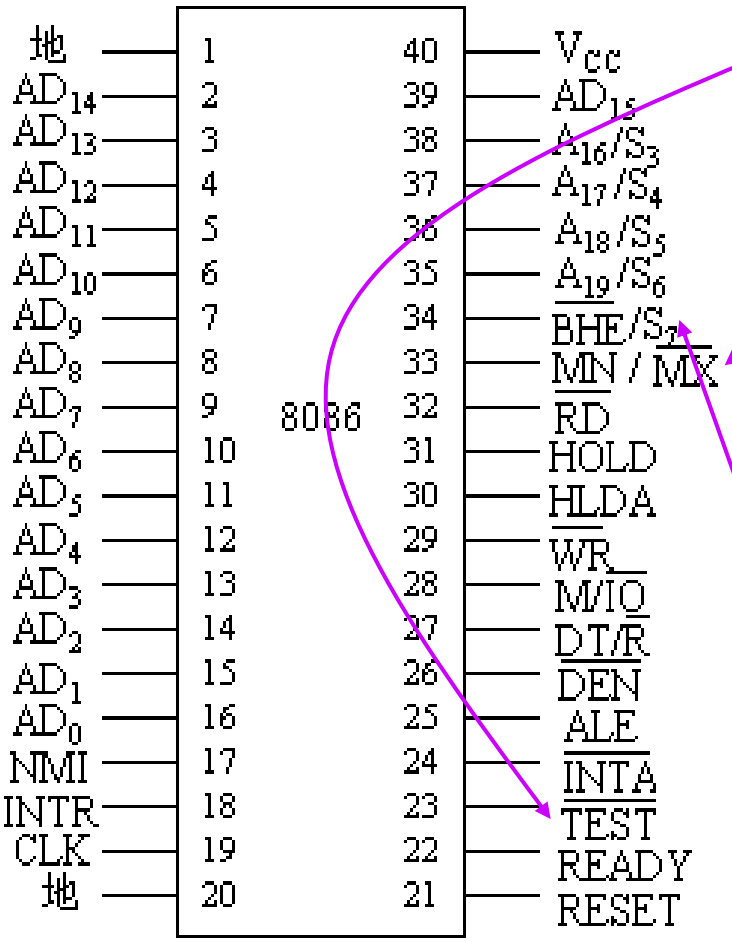
（5）**$\overline{RD}$：** 读选通。

（6）**CLK：** 时钟，占空比1/3，即1/3高电平、2/3低电平。

（7）**RESET：** 复位，4个T周期，清0标志寄存器、IP、DS、SS、ES、指令队列，CS=FFFFH。

（8）**READY：** 准备就绪，有效时表示主存或I/O接口准备好，可以进行数据传输。T3采样，若READY=0则插入TW。

2.3.1 两种模式下功能相同的引脚



（9）**TEST**：测试，由WAIT指令检查，使CPU与外部硬件同步，TEST=0继续，否则等待。

（10）**MN/MX**：最小/最大模式，接+5V最小模式，接地最大模式

（11）**BHE/S7**：高8位数据允许/状态，T1时BHE=0，AD15-AD8数据有效，S7未定义。

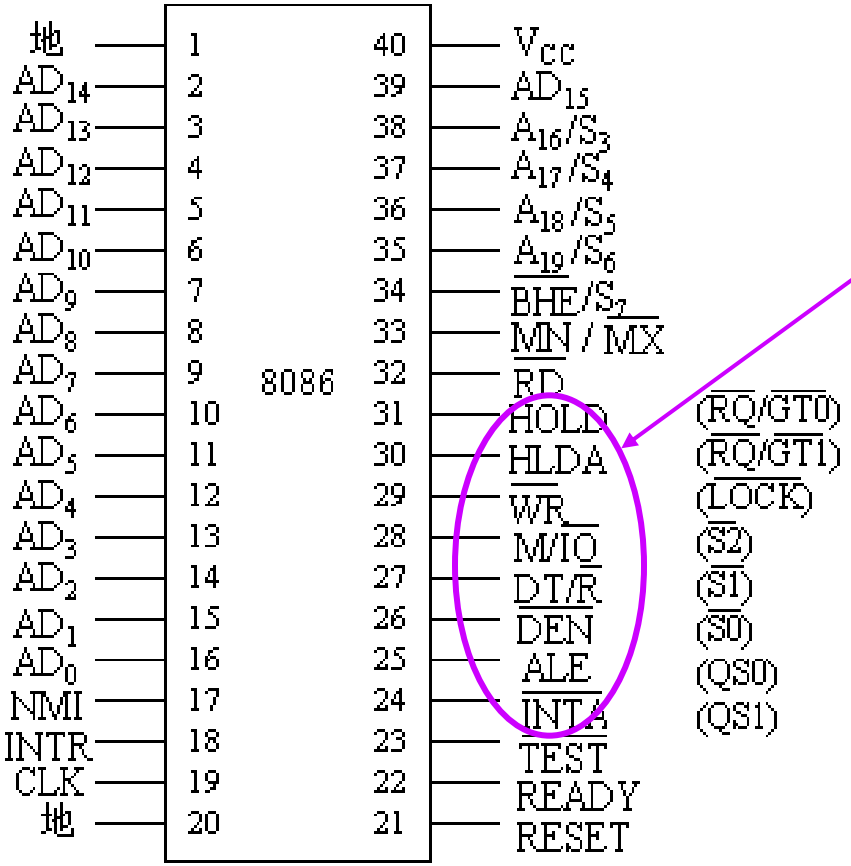
利用BHE信号和A0信号，可知系统当前的操作状态。

表2.1 BHE和A0的代码组合和对应的操作

$\overline{BHE}$ 和A0的代码组合和对应的操作			
$\overline{BHE}$	A0	操作	所用数据引脚
0	0	从偶地址单元开始读/写一个字	$AD_{15} \sim AD_0$
0	1	从奇地址单元或端口读/写一个字节	$AD_{15} \sim AD_8$
1	0	从偶地址单元或端口读/写一个字节	$AD_7 \sim AD_0$
1	1	无效	--
0	1	从奇地址开始读/写一个字（在第一个总线周期将低8位数据送到AD15 ~AD8，下一个周期将高8位数据送到AD7 ~AD0 ）	$AD_{15} \sim AD_0$
1	0		

2个周期

2.3.2 两种模式功能不同的引脚



最小模式：

INTA： 中断响应

ALE： 地址锁存允许

DEN： 数据允许

DT/R： 数据发送/接收

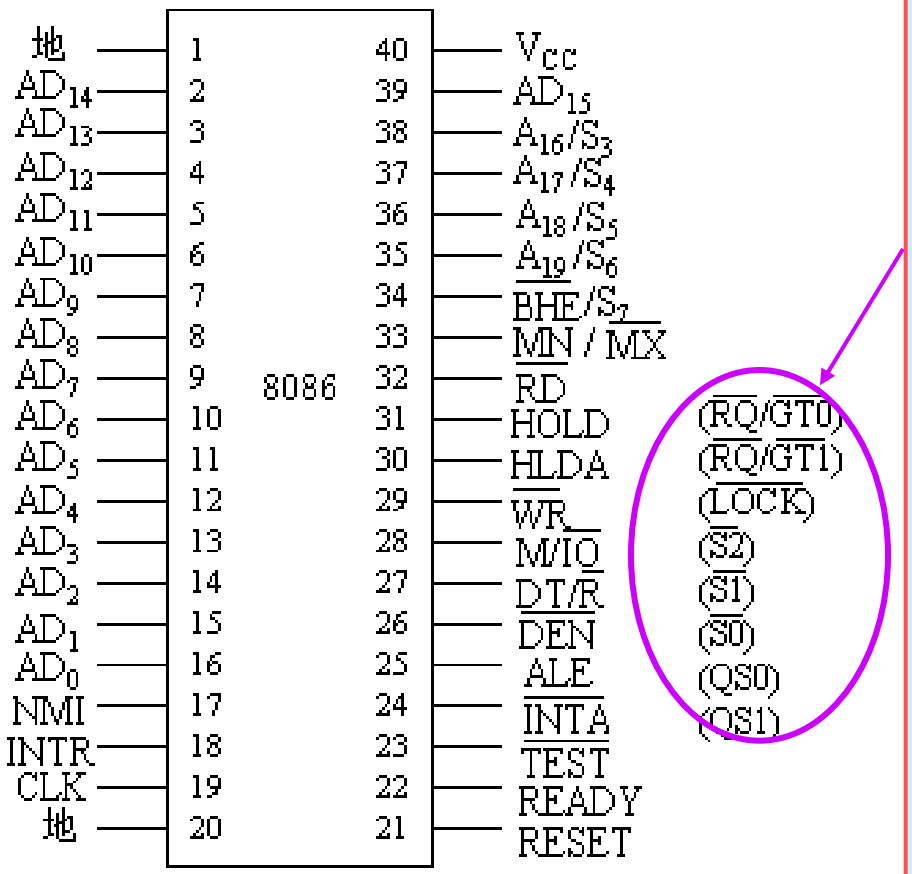
M/IO： 存储器/输入输出控制

WR： 写信号

HOLD： 总线请求

HLDA： 总线响应

2.3.2 两种模式功能不同的引脚



最大模式：

QS1、QS0： 指令队列状态，提供前一个T状态中指令队列的状态，允许外部跟踪CPU内部指令队列。

$\overline{S2}$、$\overline{S1}$、$\overline{S0}$： 总线周期状态，提供给8288。

$\overline{LOCK}$： 总线封锁，低电平时，别的设备不能获得总线控制权。由前缀指令“LOCK”产生。

$\overline{RQ}/\overline{GT1}$、$\overline{RQ}/\overline{GT0}$： 总线请求/允许，双向，$\overline{RQ}/\overline{GT1}$优先级高。

表2.4 S2、S1、S0的状态编码

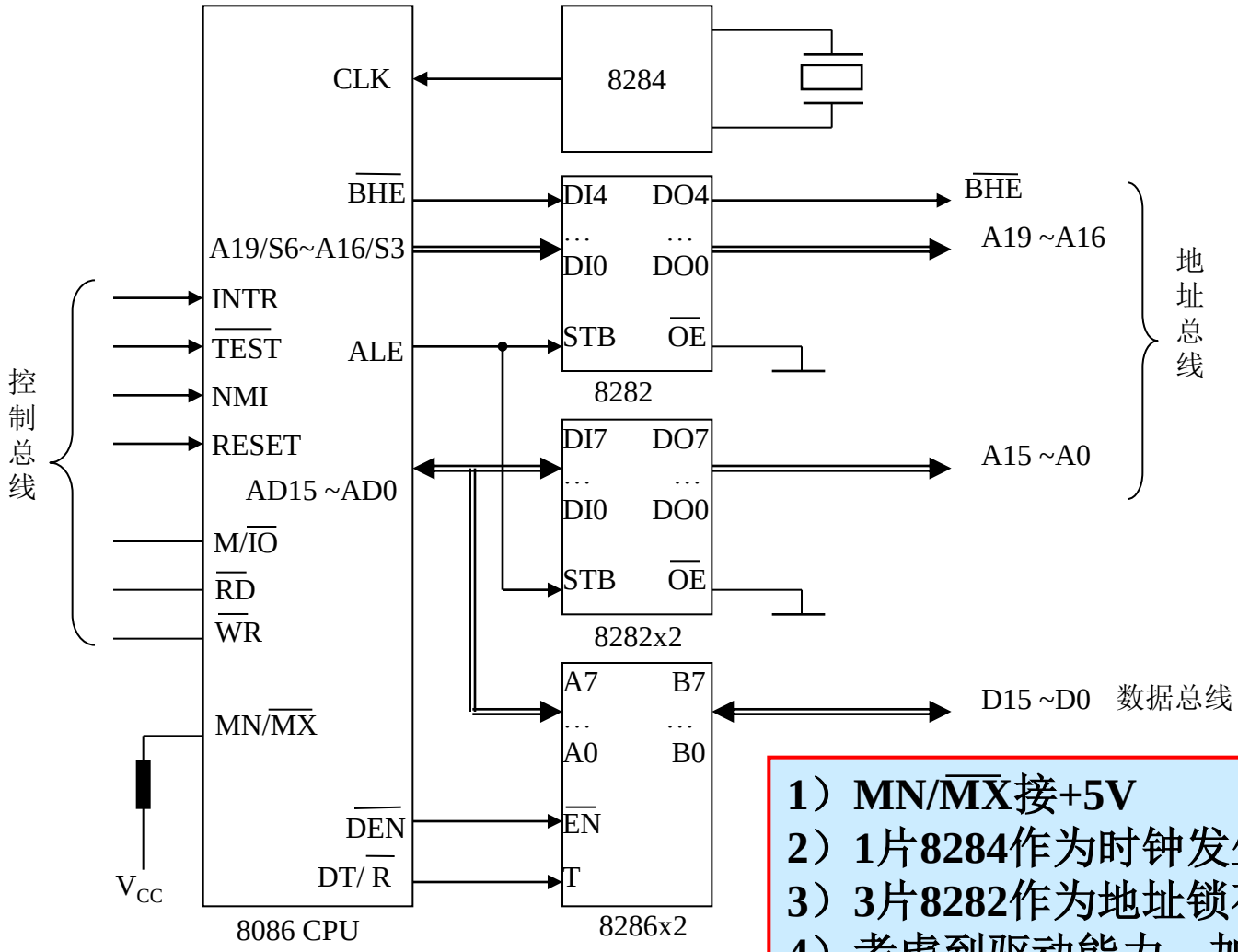
表2.4 S2、S1、S0的状态编码			
S0	S1	S2	性能
1	0	0	中断响应
1	0	1	读I/O端口
1	1	0	写I/O端口
1	1	1	暂停
0	0	0	取指
0	0	1	读存储器
0	1	0	写存储器
0	1	1	无作用

2.4．8086的两种组成模式

- 最小模式：单机系统，只有一个8086 CPU，所有控制信号由该CPU产生。系统中的控制电路可减到最小。

- 最大模式： 多机系统，有两个以上CPU，一个主8086 CPU，其他为协处理器（如8087、8089），控制信号由总线控制器8288产生。

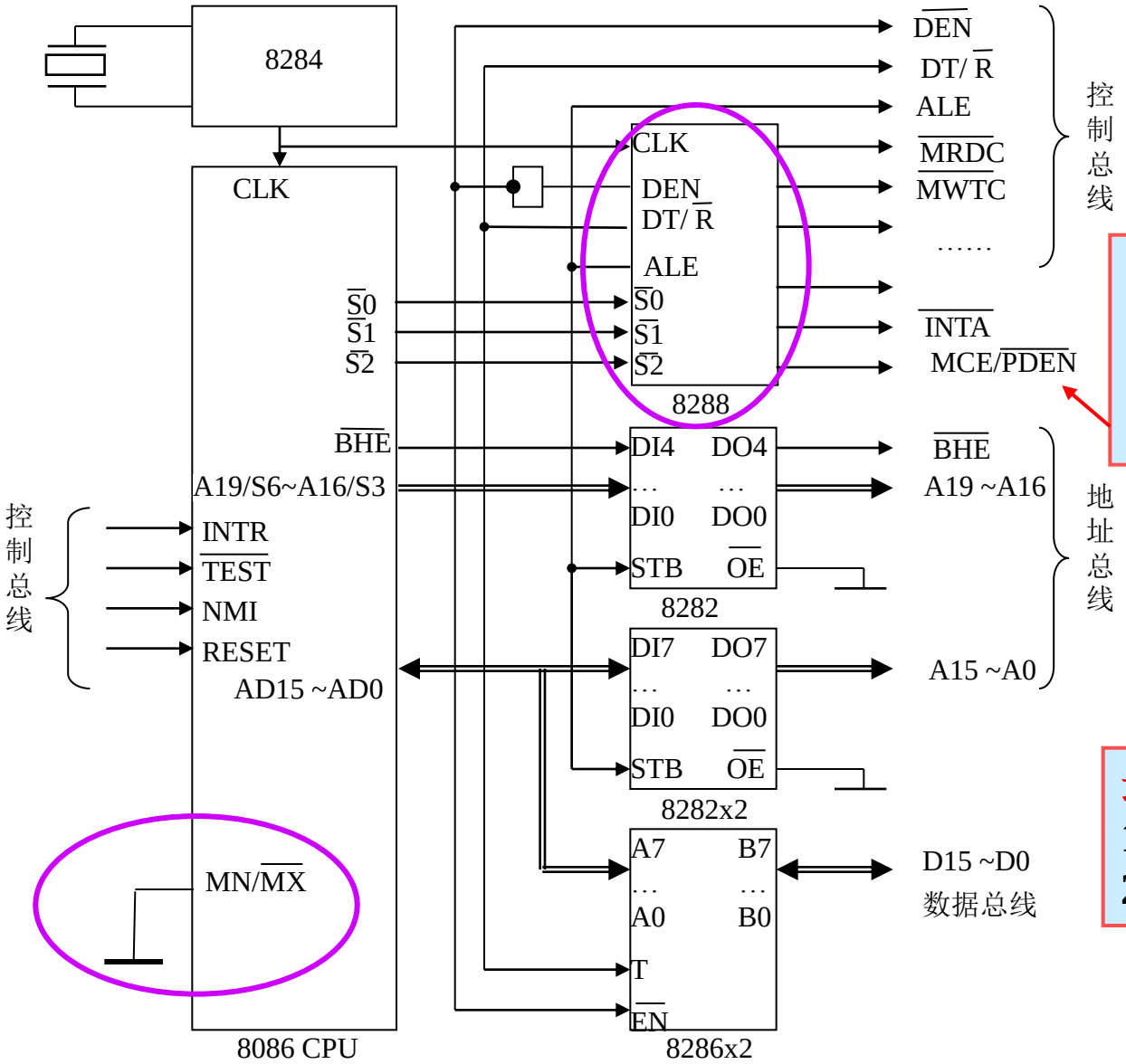
2.4.1 8086的最小模式



硬件特点

1）MN/$\overline{MX}$接+5V
2）1片8284作为时钟发生器
3）3片8282作为地址锁存器（或74LS373）
4）考虑到驱动能力，加2片8286作为总线驱动器（或74LS245）

2.4.2 8086的最大模式



MCE： 主控级联允许，
IOB=0， 用于系统总线方式
PDEN： 外围数据允许，
IOB=1， 用于I/O总线方式。

最大模式与最小模式区别
1）MN/$\overline{MX}$接地
2）1片8288总线控制器

最大模式下，总线共享的3种情况

- 最大模式下，对总线的共享可能有3种情况：

- 1）8086 CPU本身占用总线，此时，由总线控制器8288发出相应命令。

- 2）DMA请求，此时，封锁8288命令输出，由DMA控制传送。

- 3）协处理器占用总线，通过RQ/GT向8086请求，然后8086释放总线。

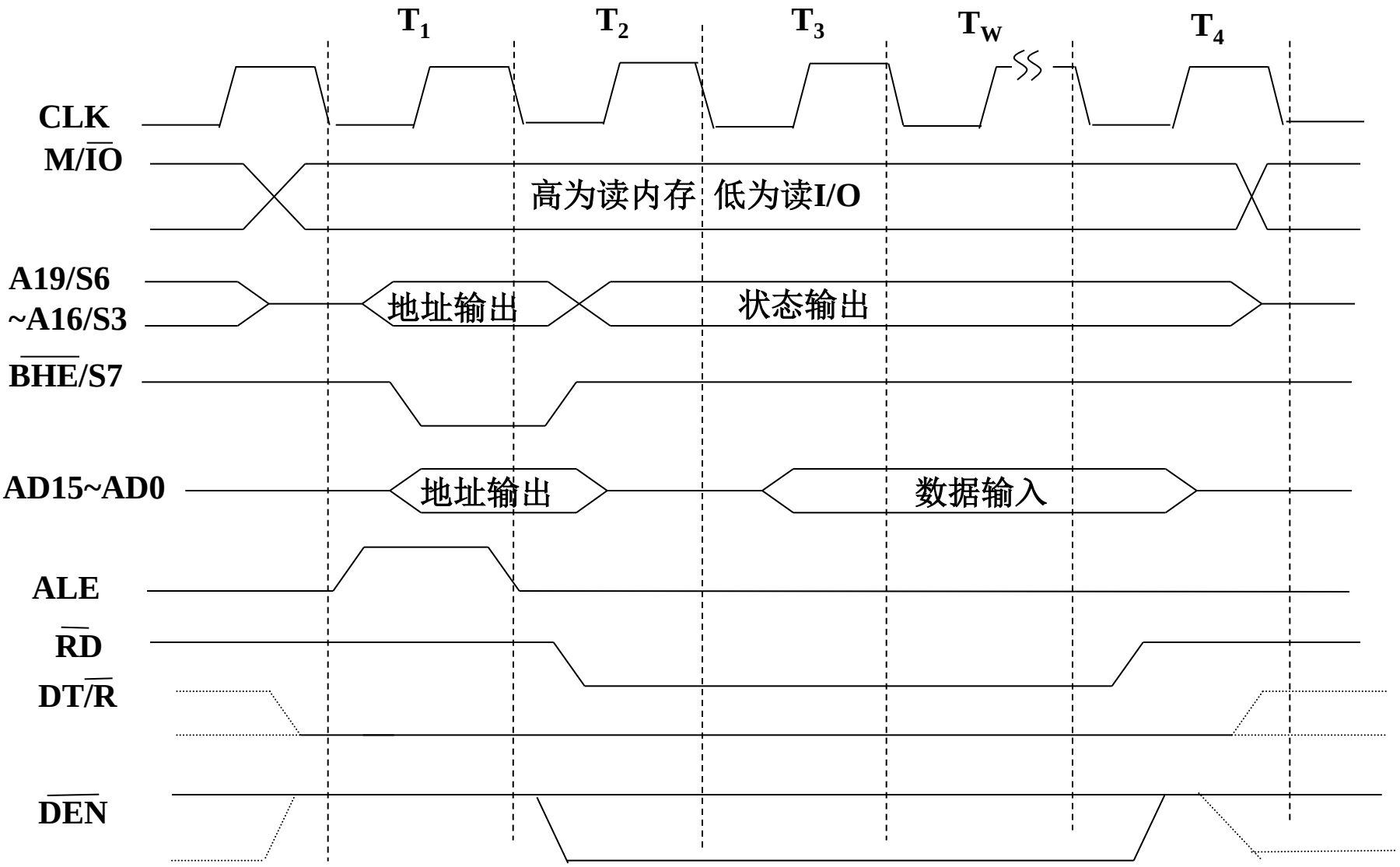
2.5 8086/8088 CPU的总线周期

- 工作时序：微处理器按照一定的时序工作。
- 时钟周期：CPU主频每个时钟脉冲的持续时间。用一个“T”表示。
- 总线周期：CPU通过总线进行一次读或写的过程。总线读操作包括取指令、读存储器、读I/O接口，总线写操作包括写存储器、写I/O接口。还有一些特殊总线周期。
- 指令周期：执行一条指令所需要的时间。
- 总线请求响应：利用总线传输数据时，总线控制部件必须获得总线的控制权。总线请求是总线控制部件发出的占用总线的请求信号，总线响应是当前控制总线的部件或总线控制器在接收到总线请求后，给出的应答信号。
- 中断响应周期：中断响应是CPU接受中断请求后的处理过程。在响应中断时，CPU在当前指令结束后，插入两个总线周期，发出中断应答，并通过总线获取中断类型码。

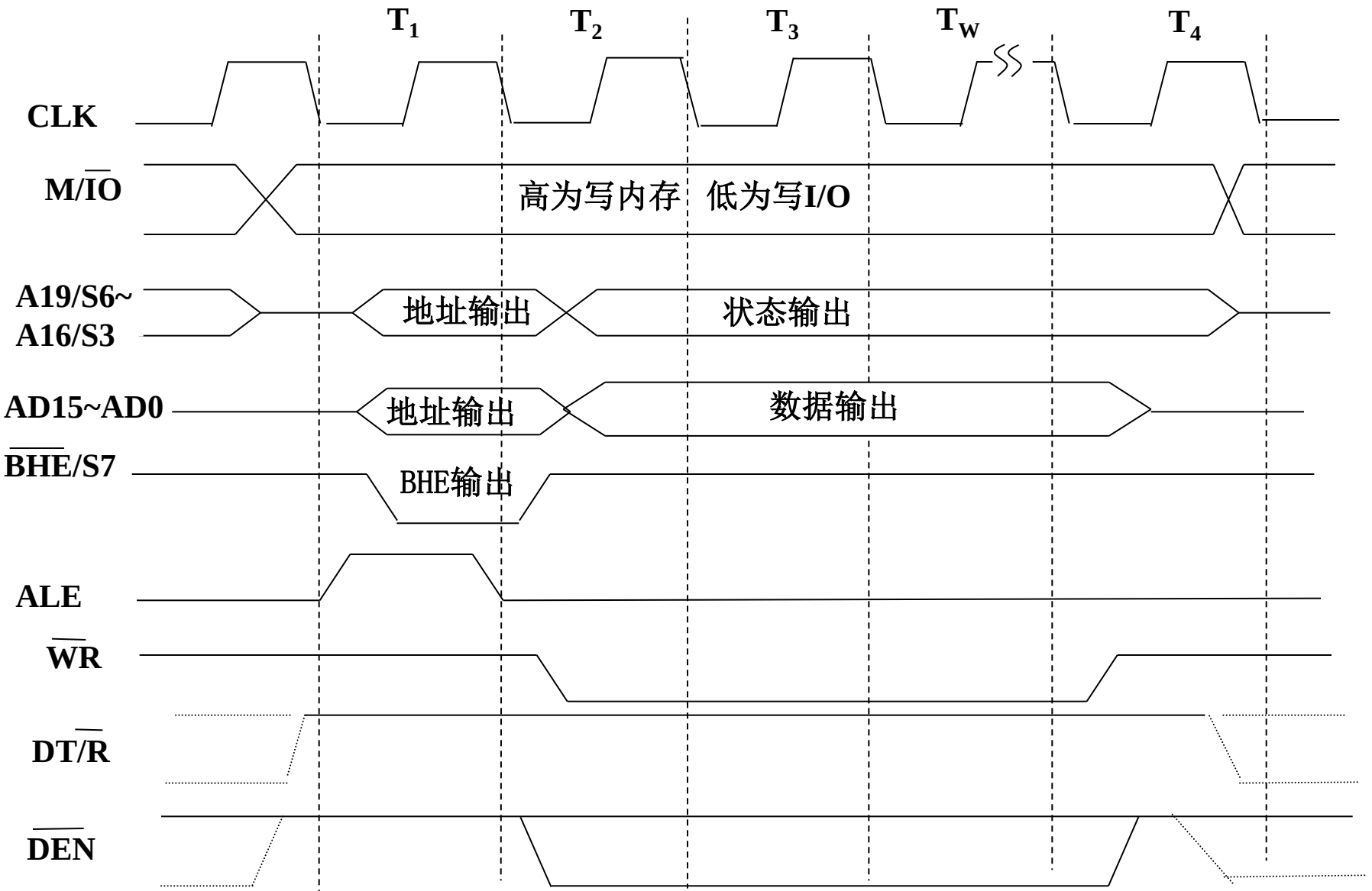
8086时钟状态

- 8086一个基本的总线周期由4个时钟周期组成。每个时钟称为T状态，用T1、T2、T3、和T4表示。
- T1状态： CPU发出地址。CPU将存储器地址或I/O端口的地址送上地址总线。
- T2状态：撤地址，发控制信号。进行读写准备，CPU撤销地址/数据、地址/状态复用线上的地址，地址/数据复用线浮置，地址/状态复用线输出状态，即复用信号在这个期间切换，读写控制信号有效。
- T3状态：地址/数据线上出现数据。写操作，CPU提供数据；读操作，等待存储器或I/O提供数据。检查READY信号，未就绪，插入TW。TW的操作与T3相同。
- T4状态：完成数据读/写，控制信号无效。结束总线操作。

8086总线读周期-最小模式



8086总线写周期-最小模式



2.6 8086的存储器组织

- 地址空间：8086有20位地址线，按字节单元编址，直接寻址能力1M字节，地址范围为00000H-FFFFFH。每个存储单元对应一个20位的地址，这个直接通过地址线给出的地址称为物理地址。
- 8086支持字节数据、字（16位）数据的存取。
- 字存放在任意连续的两个单元中，低字节放在低地址中（小端方式），低字节对应的地址为这个字数据的地址。
- 对齐字：在偶数地址中开始存放。读写时，只需一个总线周期。
- 非对齐字：在奇数地址中开始存放。读写时，需两个总线周期。

存储器的分段管理

- 20位地址，1M空间，8086CPU内部寄存器都是16位，无法直接提供20位地址，故分段管理，每段64K，段内地址连续，各段独立。
- 为了寻址每一个段，应当指明段的起始位置，这个起始位置就放在段寄存器中。

- 而段寄存器是16位的，因而规定每个段的起始位置的实际地址应被16整除，即该地址低4位为0。这样，一个段20位起始地址的高16位就是该段的段地址，段内任一个单元的地址，用相对于段起始地址的偏移量表示，即段内偏移量，16位，64K。

逻辑地址与物理地址

- 每一个存储单元都有2类地址：

  物理地址：信息在存储器中实际存放的地址。地址通过地址线给出。

  逻辑地址：编程中使用的地址。

- 逻辑地址的格式：

  段地址：段内偏移量

- 段地址：决定该段第一个字节的位置。
- 段内偏移量：该储存单元相对于该段起点字节的距离。指令中称为有效地址EA。
- 物理地址计算：

  物理地址=段寄存器×16 + 段内偏移量

  （20位） （16位）（10H） （16位）

- 同一个物理地址可以用不同的逻辑地址表示。

段寄存器的使用约定

存储器操作类型	隐含段	超越段	段内偏移量
取指令	CS	无	IP
堆栈操作	SS	无	SP
数据变量	DS	CS、SS、ES	有效地址EA
源串变量	DS	CS、SS、ES	SI
目的串变量	ES	无	DI
基址BP指针	SS	CS、SS、ES	有效地址EA

2.7 8086的I/O组织

- I/O 设备不能直接与CPU总线相连，需要通过I/O接口芯片。
- 每个I/O芯片都有一个端口或几个端口，一个端口往往对应芯片内部的一个寄存器或者一组寄存器。
- 每个端口分配一个地址，称为端口地址。
- 8086使用低16位地址线对I/O端口进行编址，最多允许有65536个8位的I/O端口，两个编号相邻的8位端口可以组合成一个16位端口。
- 8086采用独立的端口编址方式，指令系统中既有访问8位端口的输入/输出指令，也有访问16位端口的输入/输出指令。
- PC机使用A9-A0，有1024个端口。

2.8 8086的中断系统

- 8086可以处理256种不同类型的中断，每一种中断都规定一个唯一的中断类型编码。CPU根据中断类型编码来识别中断源。
- 8086 的中断分类：
  - 外部中断：外部中断源引起的中断，INTR，NMI。
  - 内部中断：由内部软件中断指令产生，或者在某些特定条件下，由CPU本身触发产生。
- 内部中断有两类：
  - 软件中断：中断指令INT n，溢出中断指令INTO，断点中断指令INT3。
  - 软件陷阱：除法错误中断，单步中断。
- 中断向量：中断服务程序的起始地址。一个中断向量占4个存储单元。

  中断向量地址=中断类型码×4
- 软件中断特点：
  1）不受IF控制
  2）不读取中断向量
  3）优先权高
- 中断优先级：（高）除法错误→软件中断→NMI→INTR→单步（低）

第2章 结 束