



第三部分 微机原理与接口

11章 8088系列微处理器





11.1 8086/8088微处理器

两者之间的差异:

- ◆ 外部数据总线宽度不同

8088的外部总线宽度是8位，8086为16位。

- ◆ 访问存储器和I/O控制的信号含义不同

8088: $\text{IO}/\overline{\text{M}}=0$ 表示访问内存;

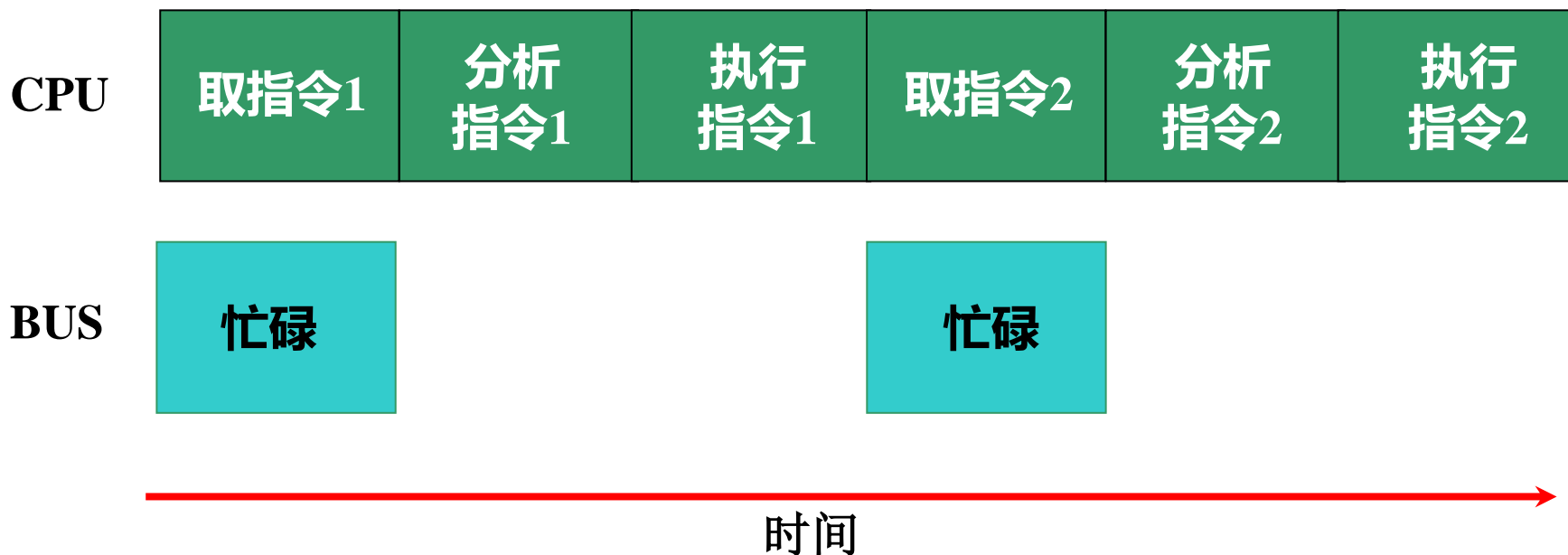
8086: $\overline{\text{IO}}/\text{M}=1$ 表示访问内存。



11.1.1 8086/8088CPU的特点

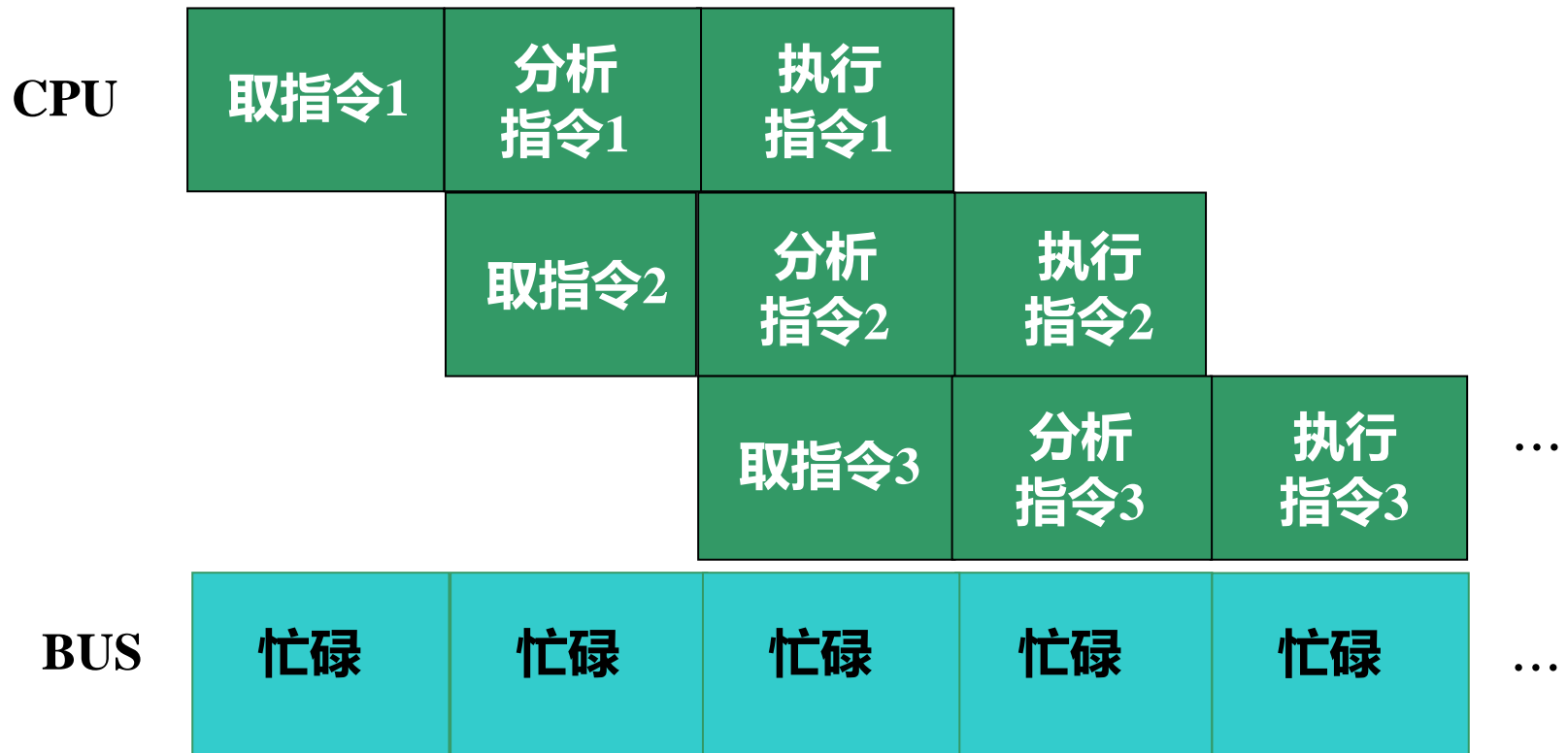
1 指令流水线

在8086/8088未出现之前，微处理器是按顺序串行完成程序的执行(假设不考虑存储器操作数)。





**8086/8088CPU并行完成程序的执行，假设
不考虑存储器操作数。**





2 内存分段管理

- ◆ 8086/8088CPU的内部结构为16位。
- ◆ 作为地址时，则只能访问64K存储单元。
- ◆ 采用分段管理的方法，将内存空间分为多个逻辑段。每个逻辑段最大64K存储单元，段内每个单元的相对地址码用16位表示。
- ◆ 每个段设置段地址，用16位寄存器存储，以区分不同的逻辑段。



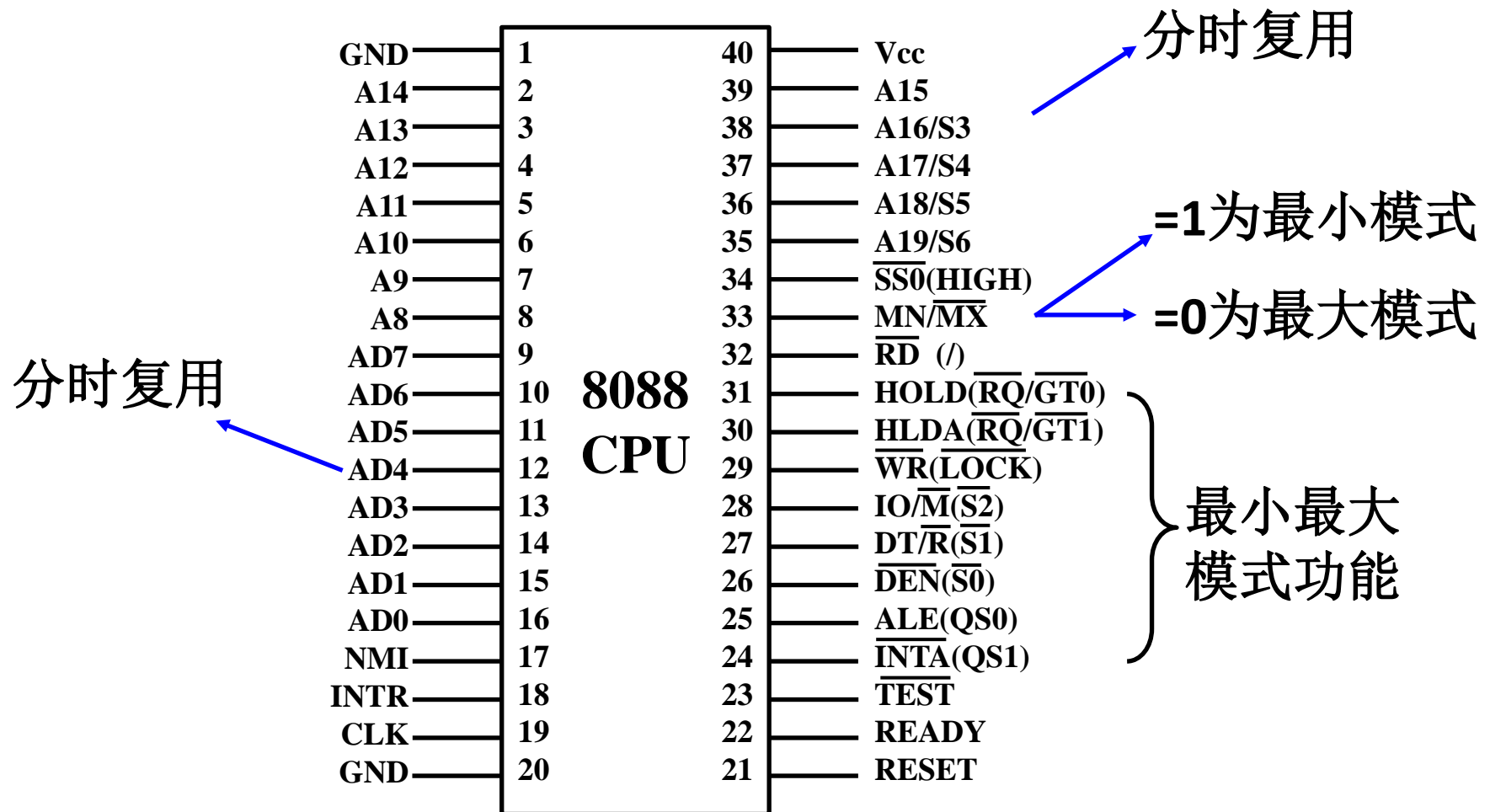
3 支持多处理器系统

8086/8088具有最小与最大两种工作模式：

- ◆ 最小模式也称为单处理器模式，系统控制总线的信号由CPU直接产生。
- ◆ 最大模式也称为多处理器模式，由总线控制器8288提供所有总线控制信号。



11.1.2 8088CPU的外部引脚及功能





1 最小模式下的引脚定义

◆ $A_{16} \sim A_{19}/S_3 \sim S_6$: 地址/状态复用。某一时刻输出最高4位地址, 另一时刻送出状态。

$S_6=0$

$S_5=IF$ (中断允许标志位)

S_4S_3 =指明CPU正在使用的段寄存器

S_4	S_3	当前段寄存器	S_4	S_3	当前段寄存器
0	0	ES	1	0	CS
0	1	SS	1	1	DS



- ◆ $A_8 \sim A_{15}$: 中8位地址。
- ◆ $AD_0 \sim AD_7$: 地址/数据分时复用, 当 $ALE=1$ 时, 输出的是地址, 当 $\overline{DEN}=0$ 时, 输出的是数据。
- ◆ IO/\overline{M} : 高电平访问IO, 低电平访问存储器。
- ◆ \overline{WR} : 写信号输出, 为低电平时, 表示对IO或存储器进行写操作。
- ◆ DT/\overline{R} : 高电平时CPU向M或IO发送数据, 低电平时CPU从M或IO接收数据。



- ◆ **$\overline{\text{DEN}}$** : 数据允许信号，当为低电平时，表示数据总线上具有有效数据，常用作数据驱动器的片选信号。
- ◆ **ALE**: 地址允许信号，当为高电平时，表示地址线上地址有效，常用作锁存器的控制器信号。
- ◆ **$\overline{\text{RD}}$** : 读写信号输出，为低电平时，表示对IO或存储器进行读操作。



- ◆ **READY:** 由M或IO发出的响应信号，当为高电平时，表示M或IO准备就绪，CPU可以进行数据传送。
- ◆ **INTR:** 可屏蔽中断请求输入信号，高电平有效。CPU在每条指令的最后一个节拍采样该信号，以决定是否进入中断响应周期。
- ◆ **NMI:** 非屏蔽中断请求输入信号，上升沿触发，CPU在当前指令执行结束后进入中断。



- ◆ **$\overline{\text{INTA}}$** : **INTR**中断响应信号，低电平有效。在响应过程中，**CPU**连续输出两个负脉冲分别用于响应信号和中断向量码的读取控制。
- ◆ **HOLD**: 总线保持请求信号输入，高电平有效。某一主设备要占用系统总线时，向**CPU**发出的请求信号。
- ◆ **HLDA**: 总线保持响应信号输出，高电平有效。当**CPU**检测到**HOLD**变低后则**HLDA**也变低，并恢复对总线的控制。

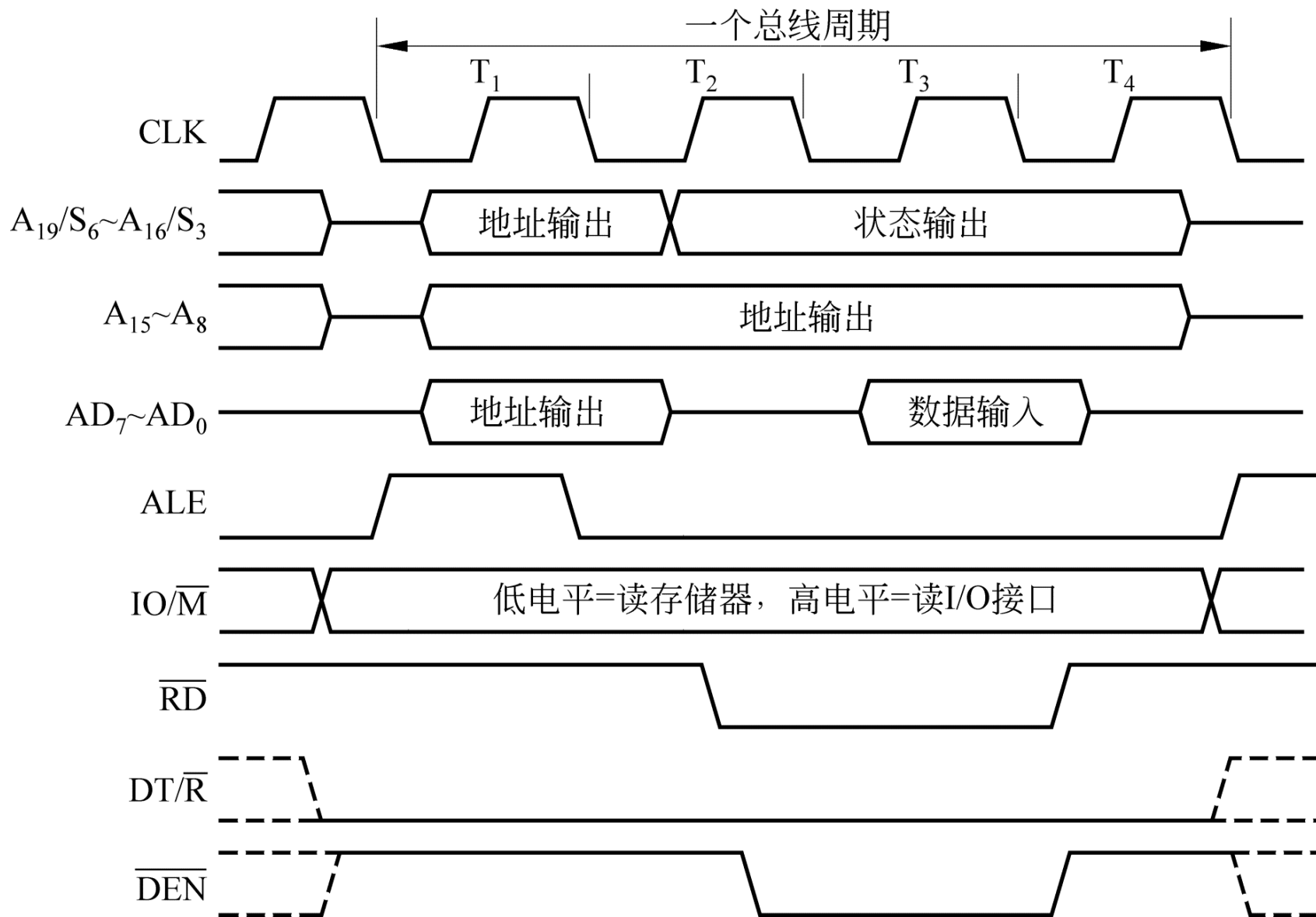


◆ \overline{SS}_0 : 系统状态信号输出。它与 IO/\overline{M} 和 DT/\overline{R} 决定了最小模式下当前总线周期的状态。

IO/\overline{M}	DT/\overline{R}	\overline{SS}_0	操作	IO/\overline{M}	DT/\overline{R}	\overline{SS}_0	操作
1	0	0	发中断响应	0	0	0	读指令
1	0	1	读IO端口	0	0	1	读内存
1	1	0	写IO端口	0	1	0	写内存
1	1	1	暂停	0	1	1	无



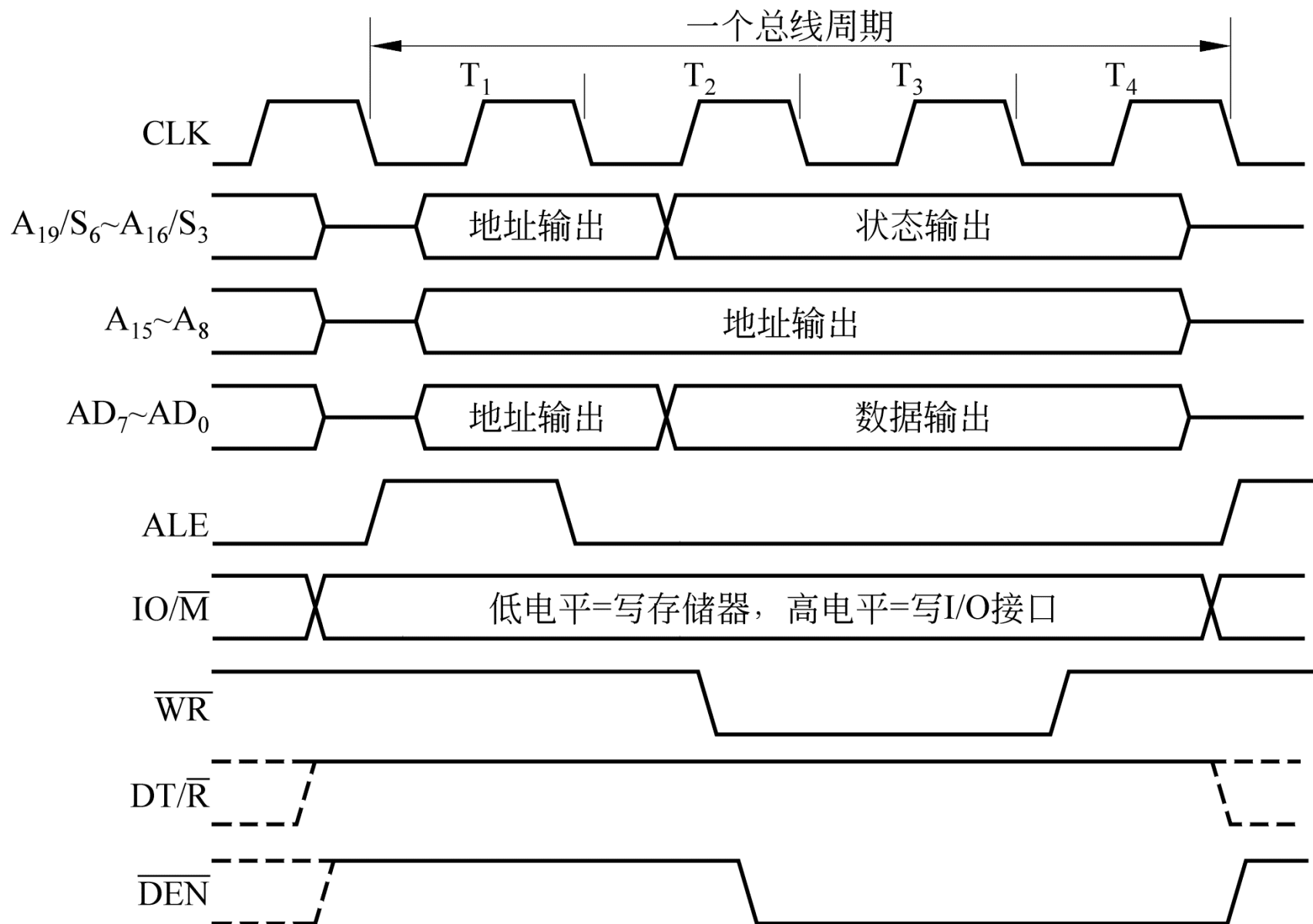
2 最小模式下的读周期





3

最小模式下的写周期





4 最大模式下的引脚定义

- ◆ \overline{S}_2 、 \overline{S}_1 、 \overline{S}_0 : 总线周期状态信号，低电平有效。它们连接到8288总线控制器的输入端，产生系统总线的各种控制信号。

\overline{S}_2	\overline{S}_1	\overline{S}_0	操作	\overline{S}_2	\overline{S}_1	\overline{S}_0	操作
0	0	0	发中断响应	1	0	0	读指令
0	0	1	读IO端口	1	0	1	读内存
0	1	0	写IO端口	1	1	0	写内存
0	1	1	暂停	1	1	1	无



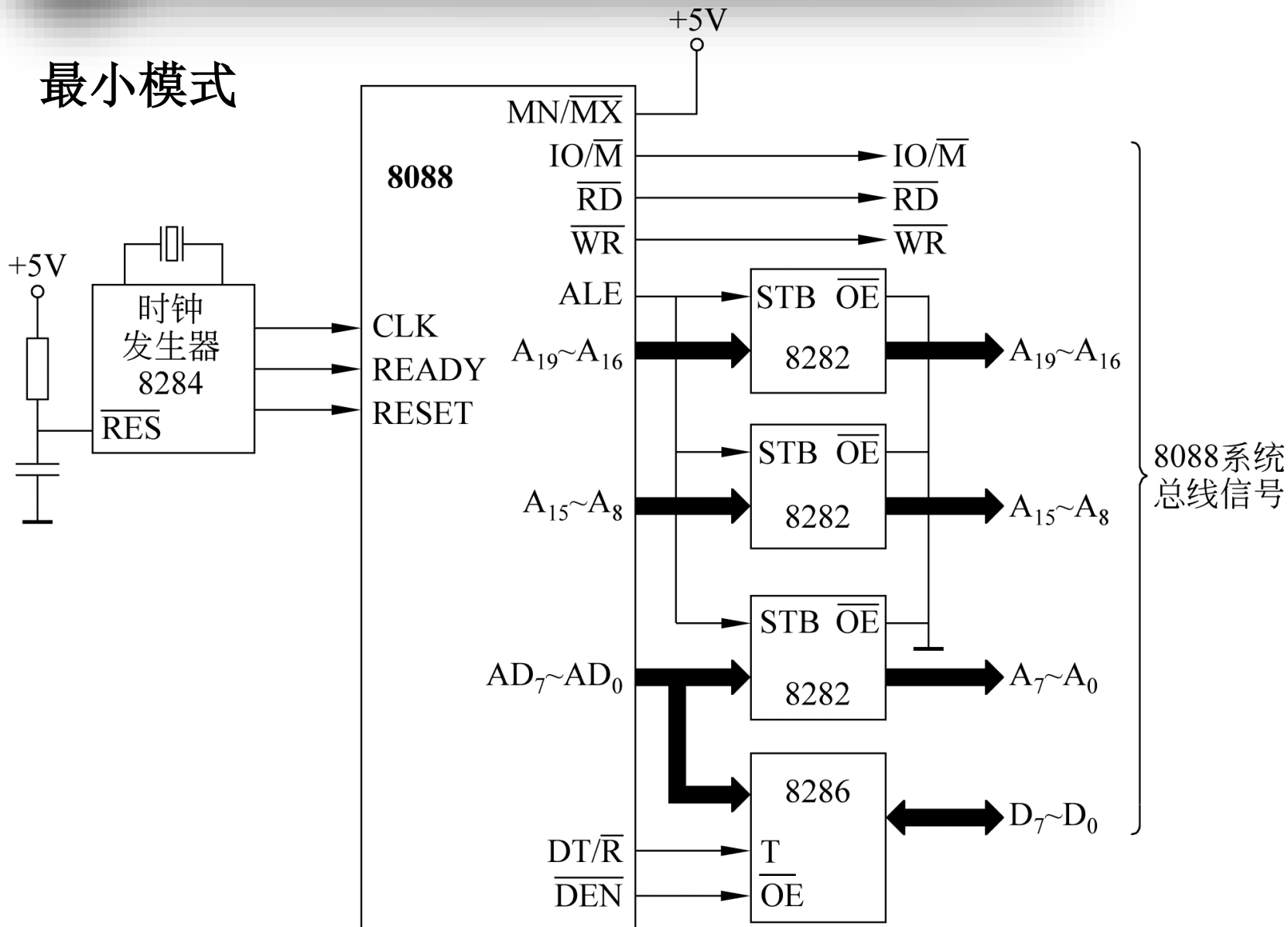
- ◆ $\overline{RQ}/\overline{GT}_1$ 、 $\overline{RQ}/\overline{GT}_0$: 总线请求/响应信号, 0级比1级高, 同时具有双向功能。
- ◆ \overline{LOCK} : 总线封锁输出信号。
- ◆ QS_1 、 QS_0 : 指令队列状态输出。

QS_1	QS_0	操作	QS_1	QS_0	操作
0	0	无操作	1	0	队列空
0	1	操作码第一字节	1	1	非第一操作码字节



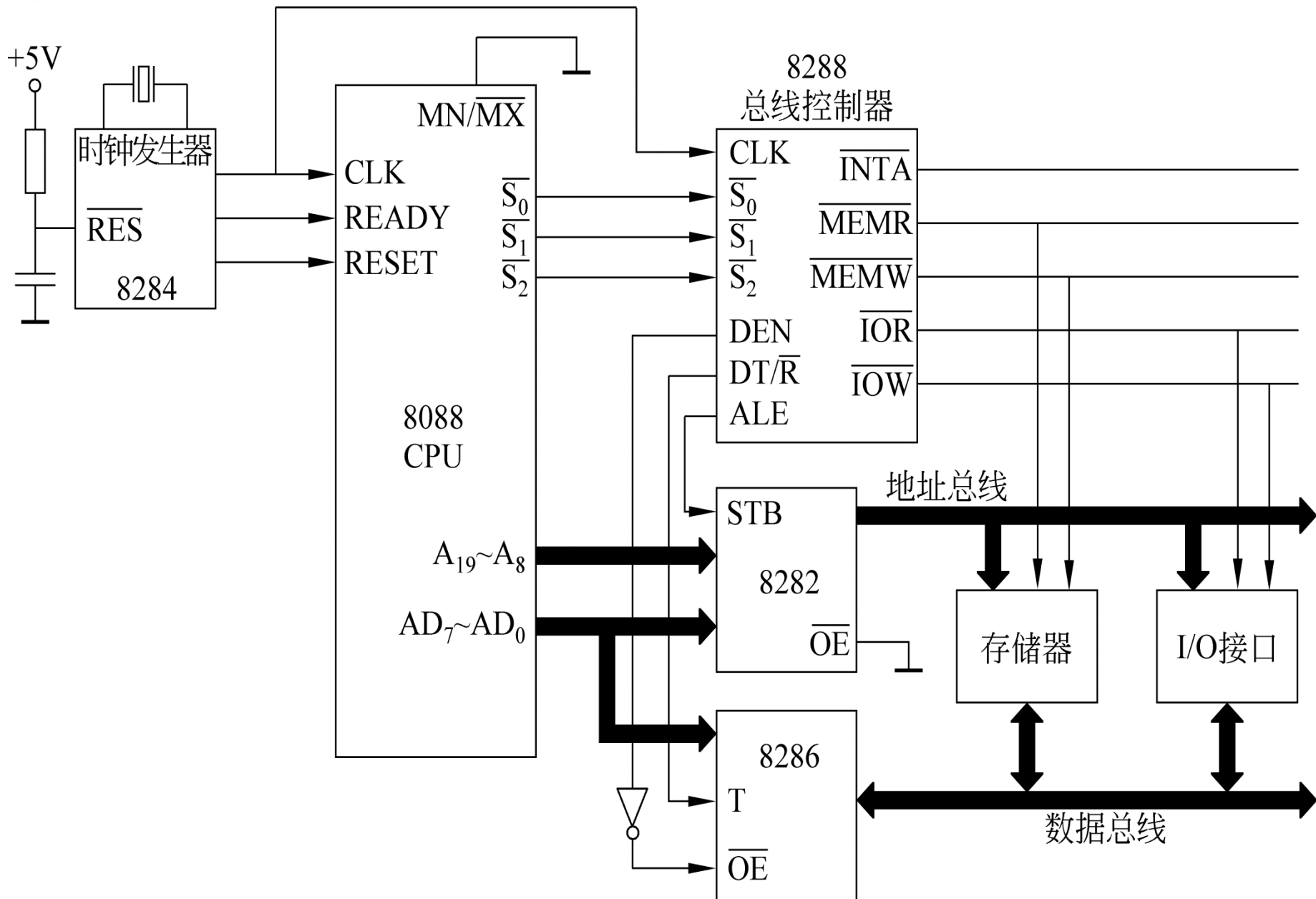
11.1.3 8088CPU的系统总线

最小模式





最大模式





11.1.4 8086/8088CPU的功能结构

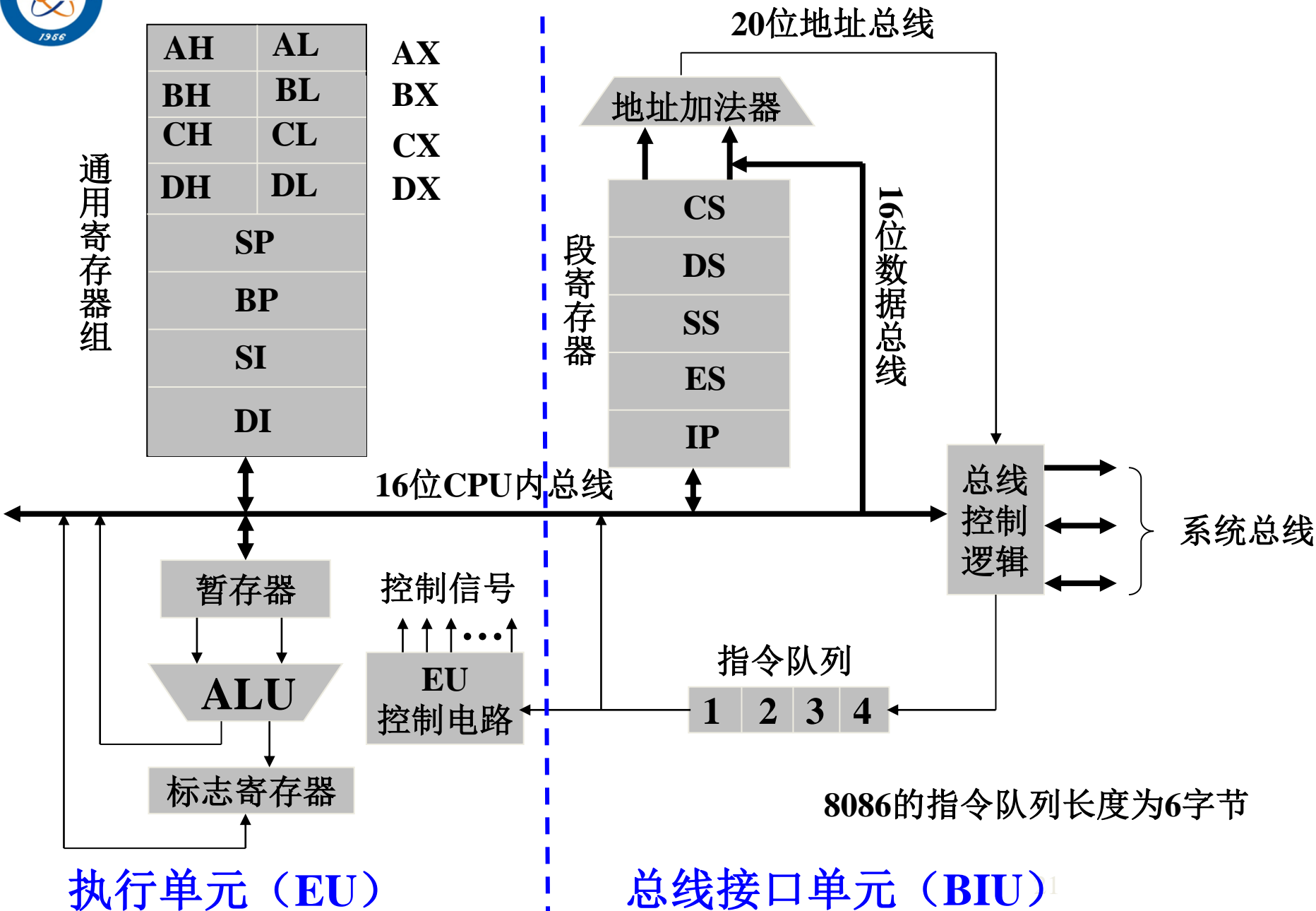
1 8086/8088CPU的内部结构

8088/8086内部由两部分组成:

{ 执行单元 (EU)
总线接口单元 (BIU)



8088CPU结构





◆ 执行单元（EU）

包括ALU、8个通用寄存器、标志寄存器
FLAGS和EU控制部件。

从指令队列中不断取出指令代码，译码后产生执行指令的控制信号。ALU完成算术和逻辑运算，通用寄存器存放操作数和运算结果，而运算结果的状态特征则保留到标志寄存器中。



◆ 总线接口单元（BIU）

包括4个段寄存器、程序计数器、指令队列、地址加法器以及总线控制逻辑。

负责CPU与存储器和IO接口之间的信息传送。由于8086/8088CPU中的寄存器均为16位，而CPU外部地址线为20根，因此，BIU中采用地址加法器产生20位的物理地址，可以寻址的存储单元为1M。



2 8086/8088CPU的内部寄存器

8086/8088CPU的内部共有14个16位寄存器，包括通用寄存器（8个）、段寄存器（4个）和控制寄存器（2个）。

1) 通用寄存器包括数据类寄存器、地址指针寄存器和变址寄存器。

（1）数据类寄存器包括AX、BX、CX、DX，常用于存放操作数或运算结果。



每个数据类寄存器分别可以作为两个独立的8位寄存器使用，从而可以方便地实现8位或16位的数据处理。

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



数据类寄存器的特殊用途:

①**AX**: 累加器, 乘除运算中的隐含操作数以及中间结果; **IO**指令中也使用**AX(AL)**进行数据传送。

②**BX**: 基址寄存器, 常用于存放被访问内存单元数据块的基地址, 默认为数据段。

③**CX**: 计数寄存器, 在循环和串操作指令中用作计数器。

④**DX**: 数据寄存器, 用于存放**IO**指令中的16位端口地址; 存放32位乘除运算中的高16位(低16位于**AX**中), 以及32位除法结果中的余数。



(2) 地址指针寄存器

包括堆栈指针寄存器**SP**和基址指针寄存器**BP**。这两个指针寄存器除可以存放操作数外，还可以作为地址指针，常用于在堆栈操作中存放偏移地址。

①**SP**：在堆栈操作中用来存放栈顶单元的偏移地址，永远指向堆栈的栈顶。

②**BP**：默认用于存放当前堆栈内某个单元的偏移地址，即可以对堆栈内任意单元的数据进行操作。



(3) 变址寄存器

变址寄存器包括源变址寄存器SI和目的变址寄存器DI，除可以存放操作数和作为地址指针外，还分别固定应用于数据的串操作指令中，提供串操作数的索引地址：

- ①SI：串操作指令中隐含作为源操作数偏移地址。
- ②DI：串操作指令中隐含作为目的操作数的偏移地址。



2) 段寄存器

CS: 代码段寄存器，当前代码段的段地址。

DS: 数据段寄存器，当前数据段的段地址。

ES: 附加段寄存器，当前附加段的段地址。

SS: 堆栈段寄存器，当前堆栈段的段地址。



3) 控制寄存器

◆ **IP：指令指针寄存器。**

用以存放预取指令的偏移地址。

CPU取指令时总是从以**CS**为段地址，**IP**为偏移地址的存储单元中取出指令代码的一个字节后，**IP**自动加1，指向指令代码的下一个字节，用户程序不能直接访问**IP**寄存器。



FLAGS: 标志寄存器或程序状态字 (PSW)。

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) 算术或逻辑运算结果的特征位。

- **CF**: 进位标志位。加(减)法运算时, 若最高位有进(借)位则**CF=1**。
- **PF**: 奇偶标志位。运算结果的低8位中“1”的个数为偶数时**PF=1**, 奇数时**PF=0**。
- **AF**: 辅助进位标志位。加(减)操作中, 低位从Bit0开始, 若Bit3向Bit4有进位(借位), 则**AF=1**。



- **ZF**: 零标志位。当运算结果为零时 **ZF=1** 。
- **SF**: 符号标志位。当运算结果的最高位为1时，则 **SF=1** 。
- **OF**: 溢出标志位。当算术运算的结果超出了带符号数的表示范围时， **OF=1** 。

例， **10110110+11110100**

$$\begin{array}{r} 10110110 \\ + 11110100 \\ \hline 1 \boxed{10101010} \end{array}$$

CF= 1 **OF= 0**

AF= 0 **PF= 1**

SF= 1 **ZF= 0**



(2) 控制标志位。

- **TF**: 陷井标志位，也叫跟踪标志位。**TF=1**时，使**CPU**处于单步执行指令的工作方式。
- **IF**: 中断允许标志位。**IF=1**使**CPU**可以响应可屏蔽中断请求。
- **DF**: 方向标志位。在数据串操作时确定操作的方向。**DF=1**，地址指针按自减方式；**DF=0**，地址指针按自增方式进行。



11.1.5 8086/8088CPU的存储器组织

1 物理地址与逻辑地址

◆ 物理地址

①指每个内存单元在整个内存空间中具有的唯一地址。

②8086/8088CPU有20根地址线，它可以产生20位的地址码，寻址范围为 2^{20} 。同时存储器按照字节进行编址，因此存储器最大容量为1MB。



在源程序中常用**5位十六进制数**或一个符号来表示一个存储单元的地址。

十六进制数地址	二进制数地址	存储单元（字节）
		70
00000H	000000000000000000000000	
00001H	000000000000000000000001	
00002H	000000000000000000000010	
⋮	⋮	⋮
0FFFFFFEH	1111,1111,1111,1111,1110	
0FFFFFFFH	1111,1111,1111,1111,1111	



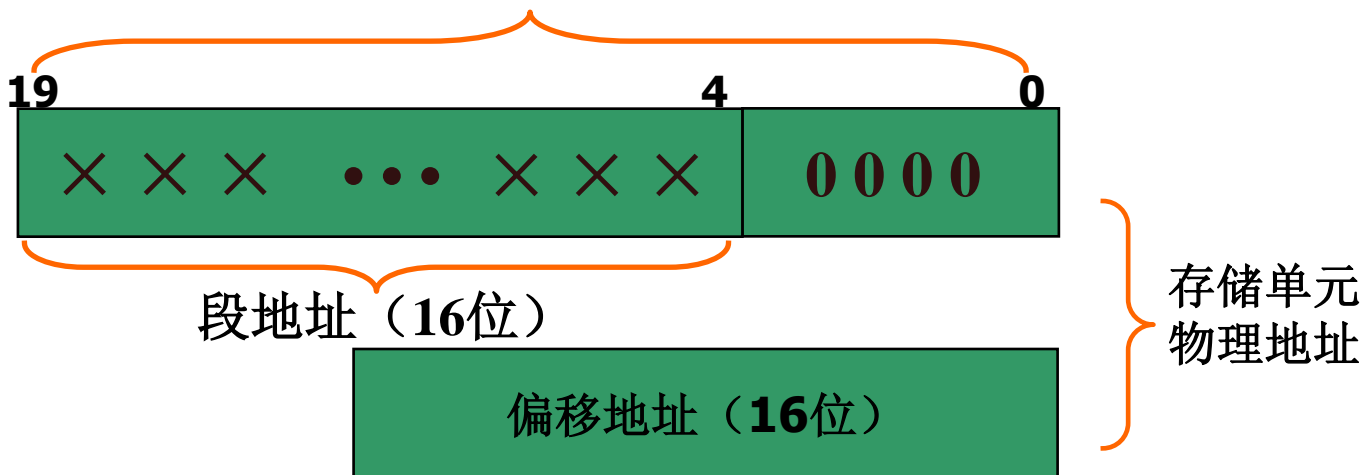
- 任何两个相邻字节单元就构成一个字单元；
- 字单元的地址为两个字节单元的低地址。
- 字数据的存放规则是低8位放在较低地址字节单元中，高8位放在较高地址字节单元中。

例如，将数据3456H放在地址为09235H的存储单元中的存储分配。

地址	存储单元
	⋮
09235H	56H
09236H	34H
	⋮



段首地址（段首的物理地址）



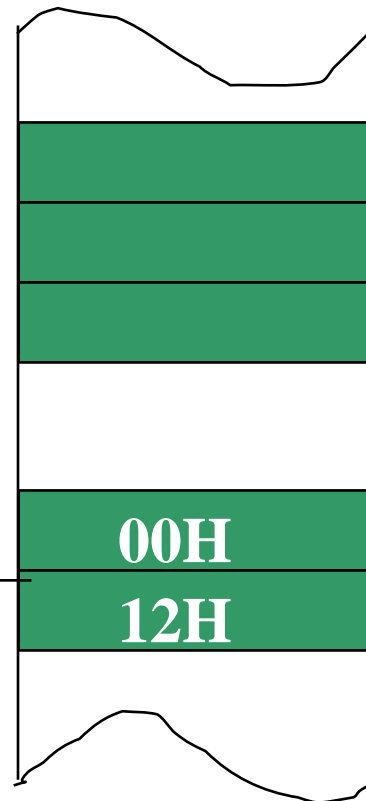
例：

- 段地址 = 1234H
- 段首地址
- 偏移地址 = 0009H
- 物理地址

12340H

9

12349H





逻辑地址

8086/8088的存储器段结构的特点：

1. 每个段最大长度为**64K（65536）**个字节单元组成。
2. 每个段的起始地址（段首地址）必须是一个小节的首址。

从**0**地址开始，每**16**个字节单元称为一个小节（Paragraph）
1MB内存就可划分为**64K**个小节。

第 0 小节：00000H, 00001H, 00002H.....0000FH

第 1 小节：00010H, 00011H, 00012H.....0001FH

⋮

⋮

⋮

⋮

⋮

第65534小节：FFFE0H FFFE1H FFFE2H.....FFFEFH

第65535小节：FFFF0H FFFF1H FFFF2H.....FFFFFH



逻辑地址通常写成, **XXXXH:YYYYH**

段地址

段内偏移地址
(相对地址)

物理地址=段地址× 16+段内偏移地址

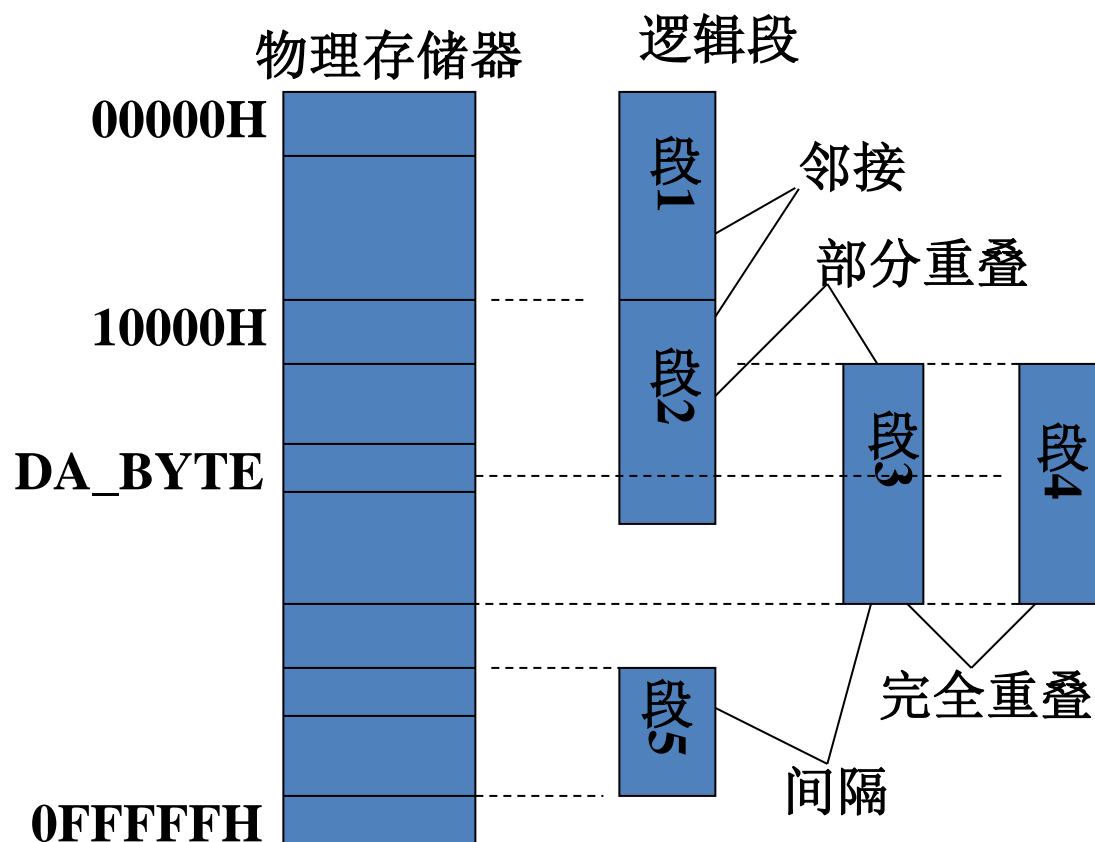
例, 设某操作数存放在数据段, **DS=250AH**, 数据所在单元的偏移地址=**0204H**。则该操作数所在单元的物理地址为:

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



3. 逻辑段在物理存储器中可以是邻接的、间隔的、部分重叠的和完全重叠的等4种情况。

内存中的一个物理存储单元可以映象到一个或多个逻辑段中



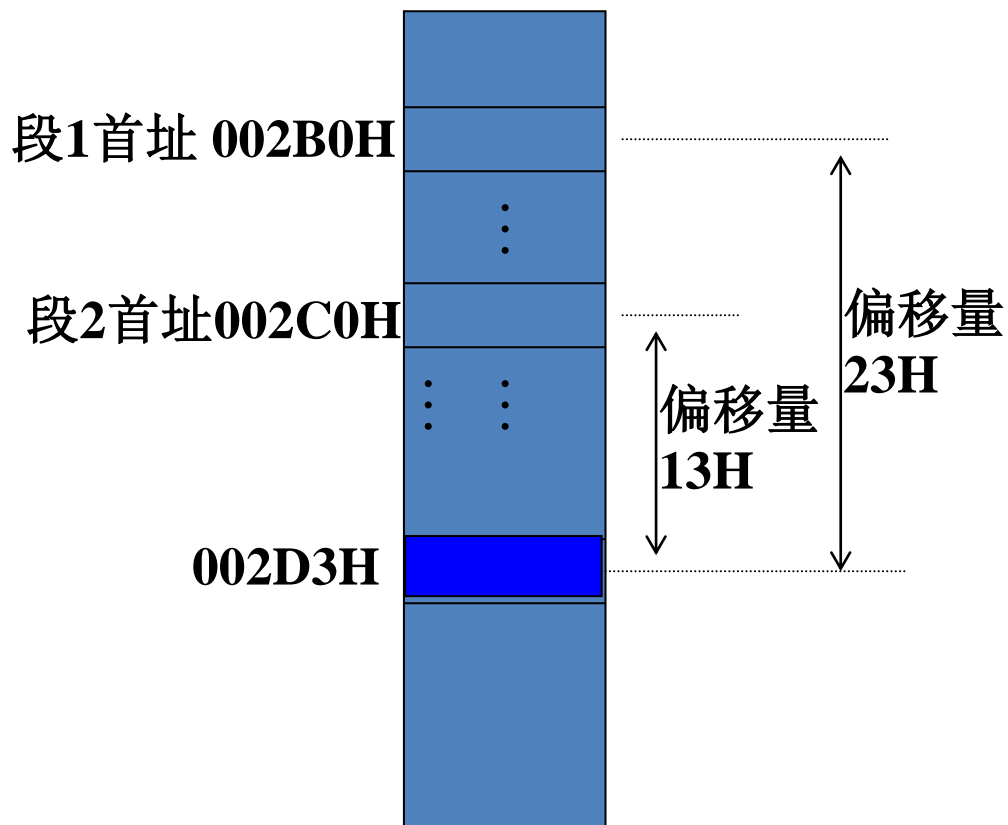
DA_BYTE物理单元可以映象到逻辑段2、段3和段4中。



例：同一个物理地址
002D3H被两个逻辑段中的
逻辑地址映射的情况。

$$002B0H + 00023H = 002D3H$$

$$002C0H + 00013H = 002D3H$$



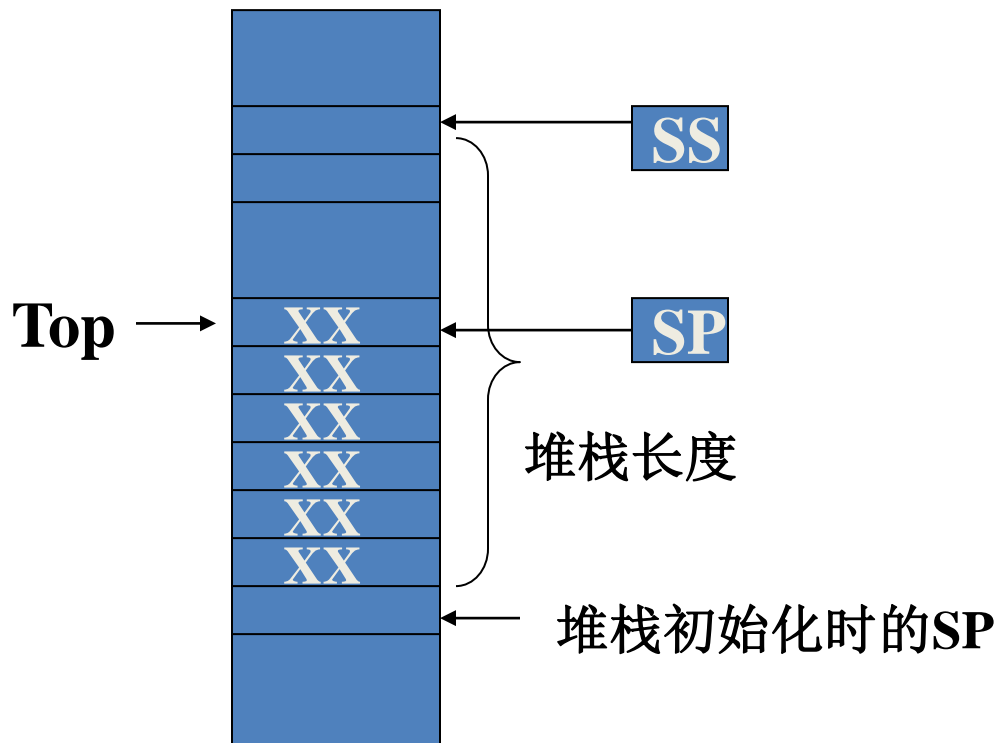
4. 在任一时刻，一个程序只能访问4个当前段中的内容。

4个当前段分别是代码段、数据段、堆栈段和附加段，它们
分别由4个段寄存器**CS**、**DS**、**SS**和**ES**提供当前段的段地址。



2 8086/8088堆栈的组织

在8086/8088微机中堆栈是由堆栈段寄存器SS指示的一段存储区（按字节编址按字存取，长度用字节数表示且为偶数）。





- 数据在堆栈中以**字**为单位存取，低8位放在较低地址单元，高8位放在较高地址单元。
- SP被初始化时的值就是堆栈的长度。由于SP是16位寄存器，因此堆栈长度< 64K字节。
- SP始终表示堆栈段首地址与栈顶之间的距离（**字节数**）。
 - 当SP为最大(初始)值时，表示堆栈为空。
 - 当SP为0时，表示堆栈全满。
- 当用户程序中要求的堆栈长度超过一个堆栈段的最大长度64KB时，可以设置多个堆栈段。