



上海交通大学

网络学院

微机原理与应用

The Principle & Application of Microcomputer

王春香 副教授

wangcx@sjtu.edu.cn

第三章 8086/8088微处理器及其系统

主要内容

- 3.1 8086/8088微处理器
- 3.2 8086/8088系统的最小/最大工作方式
- 3.3 8086/8088的存储器
- 3.4 8086/8088的指令系统



第三章 8086/8088微处理器及其系统

- ◆ 透彻理解与熟练掌握8086/8088内部组成结构、寄存器结构与总线周期等。
- ◆ 深入理解存储器的分段设计。
- ◆ 正确理解与熟练掌握物理地址和逻辑地址关系。
- ◆ 理解堆栈及其操作。
- ◆ 理解“段加偏移”寻址机制。
- ◆ 掌握寻址方式。
- ◆ 掌握6大类指令系统的基本用法。

第三章 8086/8088微处理器及其系统

- Intel 8086/8088、Z8000 和 MC68000 为代表的 16 位微处理器是第 3 代产品，其性能已达到中、高档小型计算机的水平。
- 20 多年来，Intel 系列 CPU 一直占着主导地位。
- 虽然 8086/8088 后续的 80286、80386、80486 以及 Pentium 系列 CPU 结构与功能已经发生很大变化，但从基本概念与结构以及指令格式上来讲，仍然是经典 8086/8088 CPU 的延续与提升。
- 其它系列流行的 CPU(如 AMD 公司的 6X86MX/M II 等)也与 80x86 CPU 兼容。



3. 1 8086/8088微处理器

- 8086是Intel系列的16位微处理器。
- 8086采用HMOS工艺，集成2.9万个晶体管，单一的+5V电源，40条引脚，双列直插式封装，时钟频率5~10MHz，最快的指令执行时间为0.4 μ s。
- 8086有16根数据线，20根地址线，寻址1MB的存储单元和64KB的I/O端口。

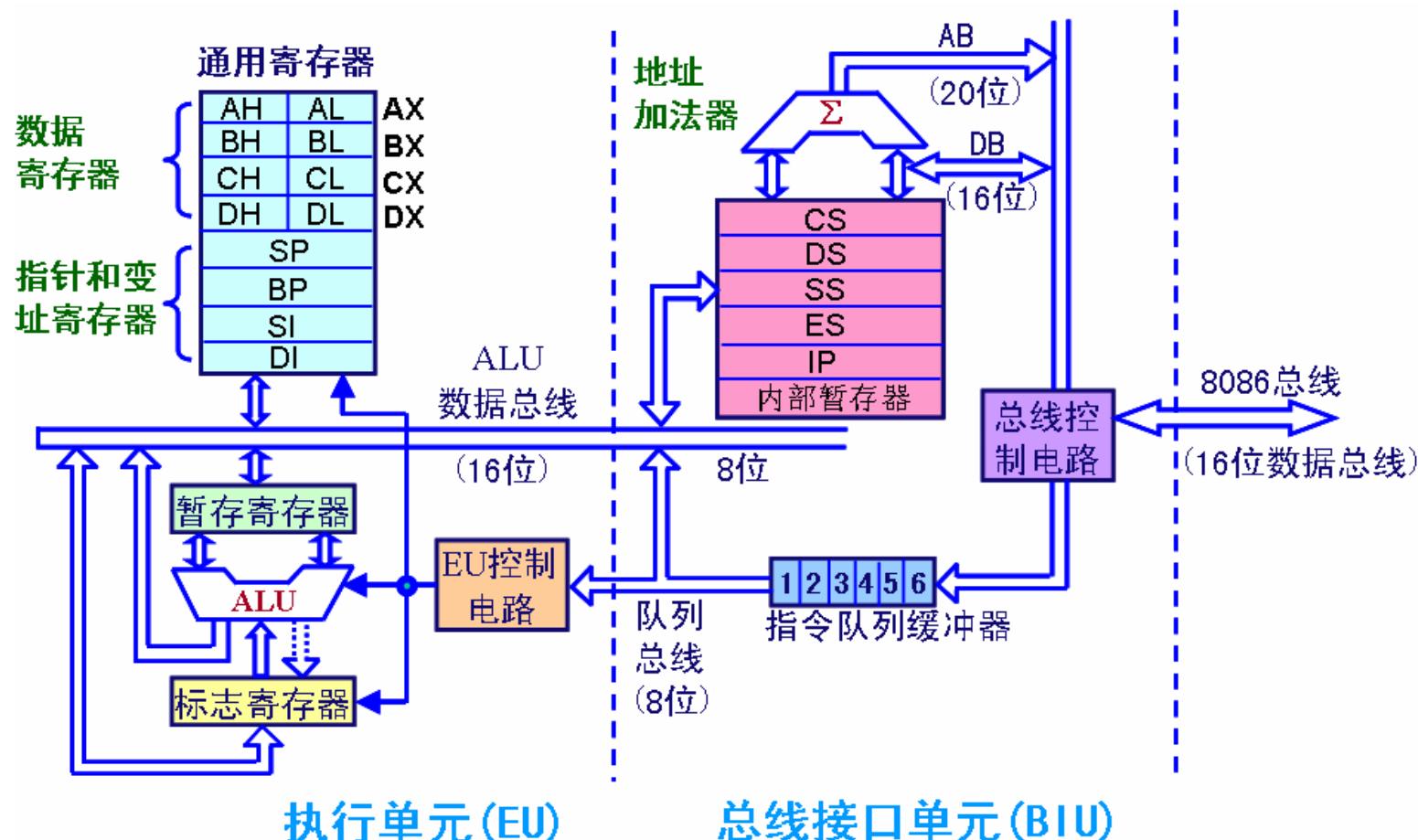
8088是准16位微处理器。内部寄存器、运算器以及内部数据总线都是16位，但外部数据总线为8条。



3.1 8086/8088微处理器

3.1.1 8086/8088 CPU的内部结构

从功能上讲，可分为两个独立的部分，并行重叠操作：





3. 1 8086/8088微处理器

◆ 总线接口单元 (Bus Interface Unit, BIU):

负责完成CPU与存储器或I/O设备之间的数据传送。即

- BIU从内存预取指令送到指令队列缓冲器；
- CPU执行指令时，BIU配合EU对指令的内存单元或I/O端口存取数据。

◆ 执行单元 (Execution Unit, EU):

负责执行指令，即

- 执行的指令从BIU的指令队列缓冲器中取得；
- 指令执行的结果或所需要的数据，由EU向BIU发出请求；
- 再由BIU对存储器或I/O端口进行存取。



3. 1 8086/8088微处理器

□ 总线接口单元组成

◆ 4个16位段地址寄存器

代码段寄存器(Code Segment): 取得CPU所执行的指令。

数据段寄存器(Data Segment): 存放程序所使用的数据。

堆栈段寄存器(Stack Segment): 堆栈操作的执行地址在此段中。

附加段寄存器(Extra Segment): 也用来存放数据。

◆ 16位指令指针寄存器 IP

◆ 6字节指令队列缓冲器

◆ 20位地址加法器

◆ 总线控制器



3. 1 8086/8088微处理器

1. 指令队列缓冲器

用于存放预取的指令。

8086指令队列为6个字节，而8088指令队列为4个字节。

在执行指令的同时，从内存中取下面1条或几条指令，取来的指令依次放在指令队列中。

采用“先进先出”的原则。



3. 1 8086/8088微处理器

1. 指令队列缓冲器（续1）

“先进先出”原则，按顺序存放，并按顺序取到EU中去执行。

- ① 取指时，当指令队列缓冲器中存满1条指令，EU开始执行；
- ② 指令队列缓冲器中只要空出两个(8088空出一个)指令字节时，BIU便自动执行取指操作，直到填满时为止。
- ③ EU执行指令时，如需对M或I/O设备存储数据时，BIU将在执行完现行取指的存储器周期后的下一个存储器周期，对内存单元或I/O设备进行存取操作，交换的数据经BIU由EU进行处理。
- ④ 当EU执行完转移、调用和返回指令时，要清除指令队列缓冲器，并要求BIU从新的地址重新执行。



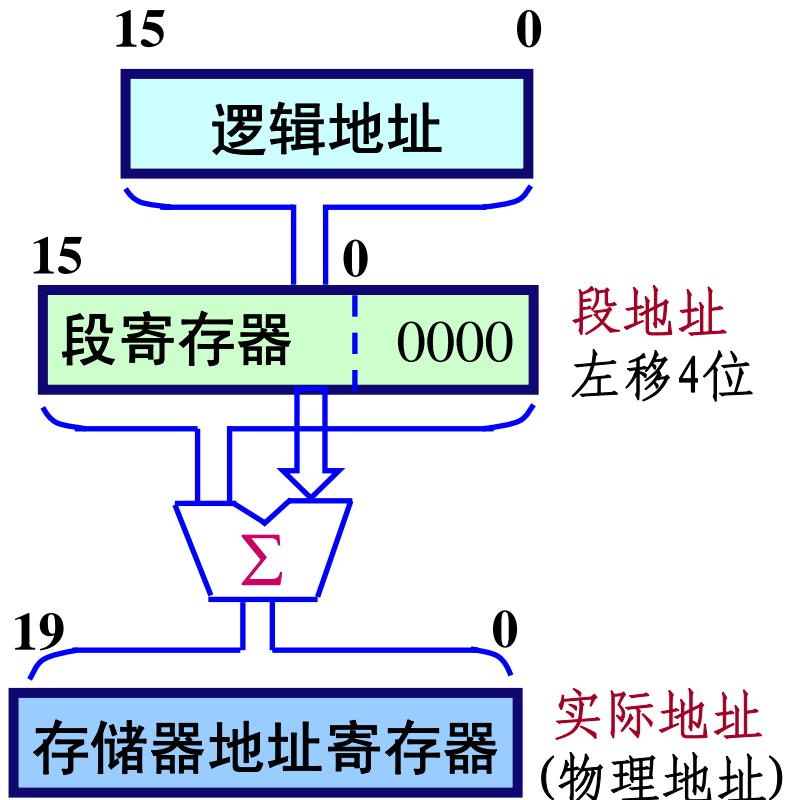
3. 1 8086/8088微处理器

2. 地址加法器和段寄存器

8086有20根地址线，内部寄存器只有16位。

“段加偏移”技术：

- 段寄存器存放确定各段起始地址的16位段地址信息。
- 由IP提供或由EU按寻址方式计算出寻址单元的16位偏移地址,也称为逻辑地址或简称偏移量。
- 将偏移地址与左移4位后的段寄存器内容同时送到地址加法器,相加后形成20位实际地址。





3. 1 8086/8088微处理器

3. 16位指令指针(Instruction Pointer, IP)

- IP中含有BIU要取的下一条指令(字节)的偏移地址。
- IP在程序运行中**自动加1**，指向要执行的下一条指令(字节)。
- **有些指令**能使IP值改变或使IP值压进堆栈，或由堆栈弹出恢复原址。



3. 1 8086/8088微处理器

□ 执行单元组成

- ◆ 16位算术逻辑单元(ALU)
算术、逻辑运算，计算16位偏移量
- ◆ 16位标志寄存器F
CPU的运算状态特征或存放控制标志
- ◆ 数据暂存寄存器
协助ALU完成运算
- ◆ 通用寄存器组
4个16位数据寄存器，4个16位指针与变址寄存器
- ◆ EU控制电路
控制、定时与状态逻辑电路



3. 1 8086/8088微处理器

□ 8088 CPU 内部结构

- BIU 指令队列长度为4个字节;
- BIU 通过总线控制电路与外部交换数据的总线宽度是8位，与专用寄存器之间的数据总线宽度也是8位。

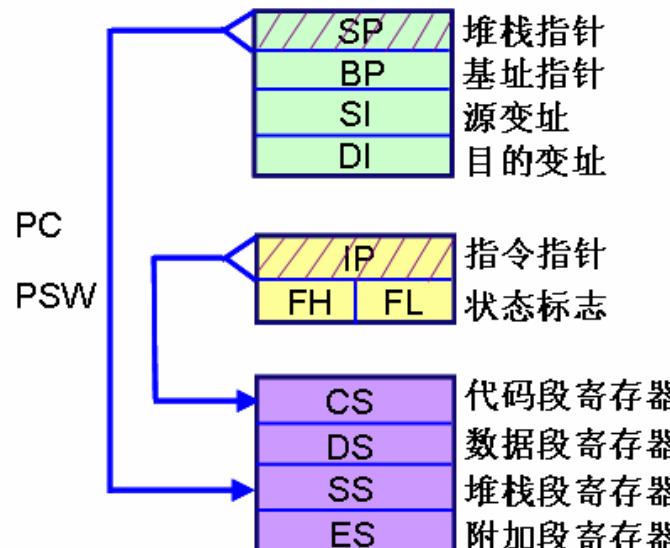
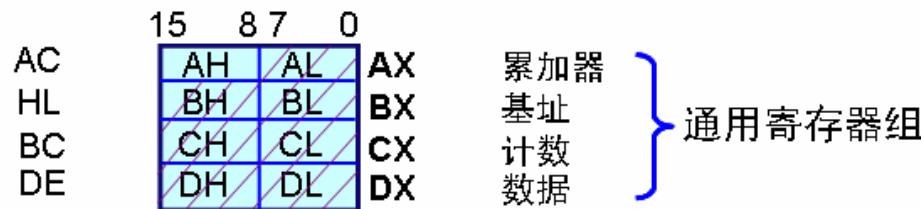


3.1 8086/8088微处理器

3.1.2 8086/8088 的寄存器结构

8086/8088的内部寄存器编程结构共有13个16位寄存器和1个只用了9位的标志寄存器。

8080/8085 相当的寄存器





3. 1 8086/8088微处理器

1. 通用寄存器

① 数据寄存器

4个16位: AX, BX, CX, DX

8个8位: AH, AL; BH, BL; CH, CL; DH, DL

- **多数情况下**, 用于算术运算或逻辑运算指令中。
- **有些指令中**, 有特定的用途。



3. 1 8086/8088微处理器

1. 通用寄存器（续1）

② 指针寄存器和变址寄存器（16位，存放偏移地址）

✓ 指针寄存器(P组):

堆栈指针寄存器SP和基址指针寄存器BP。

指示存取位于当前堆栈段中数据所在的地址。

SP(Stack Pointer): 给出栈顶的偏移地址（入栈和出栈指令时）。

BP(Base Pointer): 存放位于堆栈段中的数据区基地址的偏移地址。



3. 1 8086/8088微处理器

1. 通用寄存器 (续2)

② 指针寄存器和变址寄存器 (16位, 存放偏移地址)

✓ 变址寄存器(I组)

源变址寄存器SI和目的变址寄存器DI。

存放当前数据段的偏移地址。

SI(Source Index): 源操作数的偏移地址。

DI(Destination Index): 目的操作数的偏移地址。



3. 1 8086/8088微处理器

1. 通用寄存器（续3）

数据寄存器的隐含使用

寄存器	操作	寄存器	操作
AX	字乘, 字除, 字I/O	CL	移位, 循环移位
AL	字节乘, 字节除, 字节I/O, 查表转换, 十进制运算	DX	字乘, 字除, 间 接I/O
AH	字节乘, 字节除	SP	堆栈操作
BX	查表转换	SI	数据串操作
CX	数据串操作, 循环	DI	数据串操作



3. 1 8086/8088微处理器

2. 段寄存器

段寄存器是为实现“段加偏移”寻址基址而设置的。

4个16位段寄存器，8086/8088指令可直接访问。

CS(Code Segment): 存放代码段的段地址。

SS(Stack Segment): 存放堆栈段的段地址。

DS(Data Segment): 存放数据段的段地址。

ES(Extra Segment): 存放附加段的段地址，
用于存放处理后的数据。



3. 1 8086/8088微处理器

2. 段寄存器（续1）

8086/8088 CPU有20条地址线，具有寻址1MB存储空间。

8086/8088指令中给出的地址码仅有16位，指针寄存器和变址寄存器也只有16位长，不能直接寻址1MB大小的内存空间。

用8086/8088CPU内部一组16位段寄存器内容作为段地址，再由段寄存器左移4位形成20位的段起始地址，称为段地址或段基址。这样，8086/8088就有可能寻址1MB存储空间。

将1MB存储空间分成为若干个逻辑段，使每个逻辑段的长度为64KB（它由16位的偏移地址限定）。



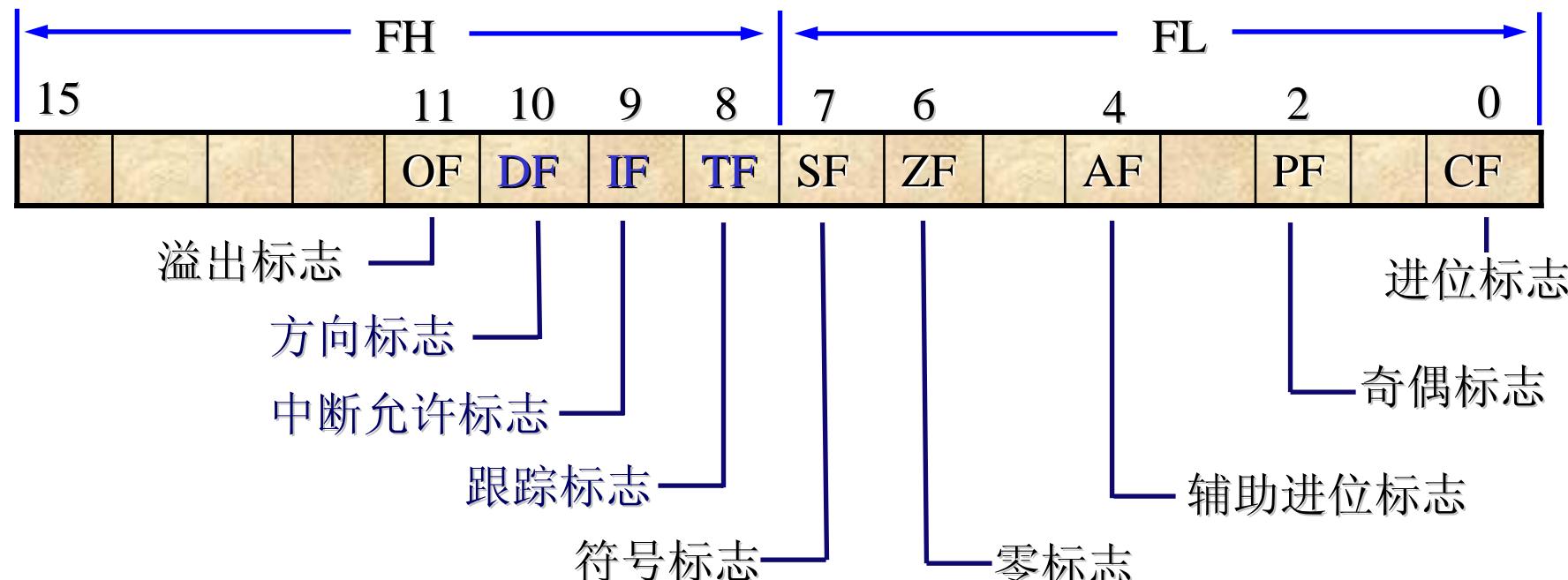
3. 1 8086/8088微处理器

3. 标志寄存器

8086/8088 标志寄存器F为16位，用了其中9位。

6个状态标志位: CF, PF, AF, ZF, SF, OF

3个控制标志位: TF, IF, DF





3. 1 8086/8088微处理器

3. 标志寄存器 (续1)

状态标志位：反映算术或逻辑运算后结果状态，记录CPU的状态特征。

① CF(Carry Flag): 进位标志，D0位。

执行结果在最高位上，产生了一个进位或借位， $CF=1$ ；无进位或借位， $CF=0$ 。

会受循环指令影响。

② PF(Parity Flag): 奇偶性标志，D2位。

执行结果的低8位中有偶数个"1"时， $PF=1$ ；否则 $PF=0$ 。用于机器中传递信息时，对产生的代码出错情况提供检测条件。现代程序设计已较少用。



3. 1 8086/8088微处理器

3. 标志寄存器 (续2)

③ AF(Auxiliary Flag): 辅助进位标志, D4位。

执行结果的低4位向高4位有进位或借位时, $AF=1$;
否则 $AF=0$ 。

一般用在BCD码运算中。

④ ZF(Zero Flag): 零标志, D6位。

如运算结果为零, $ZF=1$;

如运算结果不为零, $ZF=0$ 。



3. 1 8086/8088微处理器

3. 标志寄存器 (续3)

⑤ SF(Sign Flag): 符号标志, D7位。

如运算结果为负数, SF=1;

如运算结果为正数, SF=0。

⑥ OF(Overflow Flag): 溢出标志, D11位。

如带符号数在进行算术运算时产生了溢出, OF=1;

如无溢出, OF=0。

溢出表示运算结果已经超出机器能够表示的数值范围。



3. 1 8086/8088微处理器

3. 标志寄存器 (续4)

控制标志位: 控制CPU的操作，由程序设置或清除。

① **DF(Direction Flag):** 方向标志，D10位。

控制数据串操作指令的步进方向。

若用指令**STD**将DF=1，数据串操作过程中地址**自动递减**；

若用指令**CLD**将DF=0，则地址**自动递增**。



3.1 8086/8088微处理器

3. 标志寄存器 (续5)

② IF(Interrupt Flag): 中断允许标志, D9位。
控制可屏蔽中断的。

若用指令STI将IF=1, 允许接受外部从INTR引脚发来的可屏蔽中断请求;

若用指令CLI将IF=0, 禁止接受外部发来的可屏蔽中断请求。

IF的状态不影响非屏蔽中断(NMI)请求, 也不影响CPU相应内部的中断请求。



3.1 8086/8088微处理器

3. 标志寄存器 (续6)

③ TF(Trap Flag): 跟踪(陷井)标志, D8位。

为调试程序方便而设置的。

若TF=1, CPU处于单步工作方式;

若TF=0, 正常执行程序。



3.1 8086/8088微处理器

3.1.3 总线周期的概念

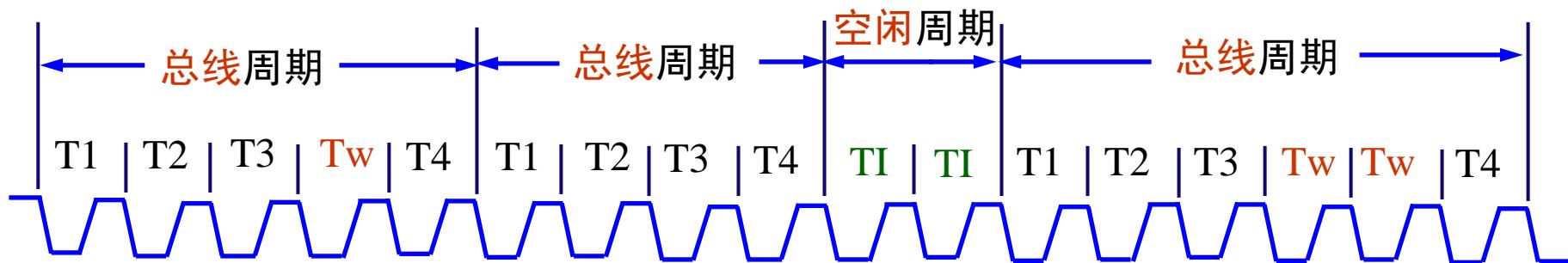
- ① **时序**：三种总线上出现的信息不但有严格的顺序，而且有准确的时间，称为定时或时序。
- ② **时钟**：时钟脉冲发生器产生具有一定频率和占空比的脉冲信号，称之为机器的主脉冲或时钟。
- ③ **主频**：时钟的频率，是机器的一个重要指标。
- ④ **时钟周期**：主频的倒数，是CPU的基本时间计量单位。
也叫一个T周期或T状态，或一个节拍。
8086主频为5MHz，则一个时钟周期为200ns。
- ⑤ **总线周期**：在取指令和传送数据时，CPU总线接口部件占用的时间称为总线周期或机器周期。



3.1 8086/8088微处理器

3.1.3 总线周期的概念（续1）

一个最基本的总线周期由4个时钟周期组成：
用 T_1, T_2, T_3, T_4 表示





3.1 8086/8088微处理器

3.1.3 总线周期的概念（续2）

- ① **T1状态**: CPU往多路复用总线上发送地址信息，选中所要寻址的存储单元或外设端口地址。
- ② **T2状态**: CPU从总线上撤销地址，并使总线的低16位浮置成高阻状态，为传送数据作准备。
- ③ **T3状态**: 总线的高4位继续提供状态信息，低16位将出现由CPU写出的数据，或CPU从存储器或者外设端口读入的数据。



3.1 8086/8088微处理器

3.1.3 总线周期的概念（续3）

- ④ **等待状态**: 有些情况下, I/O或M不能及时配合CPU传送数据, 在T3状态启动之前它会通过 READY 引脚向CPU发一个“未准备好”信号。

于是, CPU在**T3状态之后**自动插入若干个时钟周期 T_w 。直至CPU接受到“准备好”信号, 自动脱离 T_w 状态进入T4。

- ⑤ **T4状态**: 总线周期结束。

- ⑥ **空闲周期 T_I** : 两个总线周期之间, 若干个。



3.1 8086/8088微处理器

3.1.4 8086/8088 的引脚信号和功能

GND	1	40	Vcc(+5V)
AD ₁₄	2	39	AD ₁₅
AD ₁₃	3	38	A _{16/S₃}
AD ₁₂	4	37	A _{17/S₄}
AD ₁₁	5	36	A _{18/S₅}
AD ₁₀	6	35	A _{19/S₆}
AD ₉	7	34	BHE/S ₇
AD ₈	8	33	MN/MX
AD ₇	9	32	RD
AD ₆	10	31	HOLD(RQ)/GT ₀
AD ₅	11	30	HLDA(RQ)/GT ₁
AD ₄	12	29	WR(LOCK)
AD ₃	13	28	M(IO/S ₂)
AD ₂	14	27	DT/R(S ₂)
AD ₁	15	26	DEN(S ₀)
AD ₀	16	25	ALE(QS ₀)
NMI	17	24	INTA(QS ₁)
INTR	18	23	TEST
CLK	19	22	READY
GND	20	21	RESET

GND	1	40	Vcc(+5V)
A ₁₄	2	39	A ₁₅
A ₁₃	3	38	A _{16/S₃}
A ₁₂	4	37	A _{17/S₄}
A ₁₁	5	36	A _{18/S₅}
A ₁₀	6	35	A _{19/S₆}
A ₉	7	34	SS ₀ (HIGH)
A ₈	8	33	MN/MX
AD ₇	9	32	RD
AD ₆	10	31	HOLD(RQ)/GT ₀
AD ₅	11	30	HLDA(RQ)/GT ₁
AD ₄	12	29	WR(LOCK)
AD ₃	13	28	M(IO/S ₂)
AD ₂	14	27	DT/R(S ₂)
AD ₁	15	26	DEN(S ₀)
AD ₀	16	25	ALE(QS ₀)
NMI	17	24	INTA(QS ₁)
INTR	18	23	TEST
CLK	19	22	READY
GND	20	21	RESET



3.1 8086/8088微处理器

3.1.4 8086/8088 的引脚信号和功能 (续1)

① 地址/数据总线: AD₁₅~AD₀ (引脚39,2-16, 16根)

分时复用的地址和数据总线。

传送地址时: 单向, 三态输出

传送数据时: 双向, 三态输入输出

利用分时复用方法使8086/8088用40条引脚实现20位地址、16位数据及众多控制信号和状态信号的传输。

8088中只能传输8位数据, 因此只有AD7~AD0 8条地址/数据线, A15~A8只用来输出地址。

T1状态: 用来输出要寻址的存储器或I/O端口地址;

T2状态: 浮置成高阻状态, 为传输数据作准备;

T3状态: 用于传输数据; T4状态: 结束总线周期。

当CPU响应中断以及系统总线“保持响应”时, 复用线都被浮置为高阻状态。



3.1 8086/8088微处理器

3.1.4 8086/8088 的引脚信号和功能 (续2)

- ② 地址/状态总线: $A_{19}/S_6 \sim A_{16}/S_3$ (引脚35-38, 4根)
分时输出地址和状态信息。输出, 三态。

T_1 状态: 输出地址的最高4位;

$T_2 \sim T_4$ 状态: 输出状态信息。

访问存储器时: T_1 状态时输出的 $A_{19} \sim A_{16}$ 送到锁存器(8282)锁存, 与 $AD_{15} \sim AD_0$ 组成20位的地址信号;

访问I/O端口时: 不使用这4条引线, $A_{19} \sim A_{16} = 0$ 。

S_6 为0, 指示8086/8088当前与总线相连。在 $T_2 \sim T_4$ 状态, S_6 总等于0, 即8086/8088当前连在总线上。

S_5 表明中断允许标志位IF的当前设置。

S_4 和 S_3 用来指示当前正在使用哪个段寄存器。



3.1 8086/8088微处理器

3.1.4 8086/8088 的引脚信号和功能 (续3)

S₄、S₃的代码组合和对应的状态

S ₄	S ₃	状态
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, 或未用任何段寄存器
1	1	当前正在使用DS



3.1 8086/8088微处理器

3.1.4 8086/8088 的引脚信号和功能 (续4)

③ 电源线和地线: 3根

1个电源线Vcc(引脚40): 输入电压 $+5V \pm 10\%$

2个地线GND(引脚1,20): 均接地



3.1 8086/8088微处理器

3.1.4 8086/8088 的引脚信号和功能 (续5)

④ 控制总线: 8根

a) $\overline{\text{BHE}}/\text{S}_7$ (引脚34): 高8位数据总线允许/状态复用三态, 输出

$\overline{\text{BHE}}$ 在总线周期的 T_1 状态时输出, S_7 在 $\text{T}_2 \sim \text{T}_4$ 时输出。

8086中, 当 $\overline{\text{BHE}}/\text{S}_7$ 引脚上输出 $\overline{\text{BHE}}$ 信号时, 表示总线高8位AD15~AD8上的数据有效。

8088中, 34引脚不是 $\overline{\text{BHE}}/\text{S}_7$, 而是被赋予另外的信号: 最小方式时, 它为 SS_0 ; 最大方式时, 它恒为高电平。

S_7 在8086中未被赋予定义。



3.1 8086/8088微处理器

3.1.4 8086/8088 的引脚信号和功能 (续6)

b) \overline{RD} (引脚32)：读控制，三态，输出，低有效

$\overline{RD} = 0$ 时，表示CPU将执行一个对M或I/O端口的读操作。

对内存单元还是对I/O端口读取数据，取决于 M/\overline{IO} (8086) 或 \overline{M}/IO (8088) 信号。

在一个读操作的总线周期中， \overline{RD} 信号在 T_2 、 T_3 和 T_w 状态均为低电平，以保证CPU读有效。

在系统总线“保持响应”期间， \overline{RD} 被浮空。



3.1 8086/8088微处理器

3.1.4 8086/8088 的引脚信号和功能 (续7)

c) *READY* (引脚22)：“准备好”信号，输入，高有效
由所寻址的存储器或端口发出

READY=1时，表示所寻址的内存或I/O设备已准备就绪，
马上就可进行一次数据传输。

CPU在每个总线周期的T₃状态开始对READY信号采样。

READY=0，表示存储器或I/O设备尚未准备好，则CPU
在T3状态之后自动插入一个或几个等待状态Tw，直到
READY 变为高电平，才进入T4状态，完成数据传送过
程，从而结束当前总线周期。



3.1 8086/8088微处理器

3.1.4 8086/8088 的引脚信号和功能 (续8)

d) \overline{TEST} (引脚23) : 等待测试信号, 输入, 低有效

当CPU执行WAIT指令时, 就进入空转的等待状态, 并且每隔5个时钟周期对该线的输入进行一次测试;

若 $\overline{TEST} = 1$ 时, CPU将停止取下条指令而继续处于等待状态, 重复执行WAIT指令, 直至 $\overline{TEST} = 0$ 时, 等待状态结束, CPU才继续往下执行被暂停的指令。

等待期间允许外部中断。



3.1 8086/8088微处理器

3.1.4 8086/8088 的引脚信号和功能 (续9)

e) *INTR* (引脚18): 可屏蔽中断请求, 输入, 高有效

*INTR=1*时, 表示外设提出了中断请求, 8086/8088在每个指令周期的最后一个T状态去采样此信号。

若*IF=1*, 则CPU响应中断, 停止执行当前的指令序列, 并转去执行中断服务程序。



3.1 8086/8088微处理器

3.1.4 8086/8088 的引脚信号和功能 (续10)

f) *NMI* (引脚17): 非屏蔽中断请求, 输入, 上升沿触发

此请求不受IF状态的影响, 也不能用软件屏蔽, 只要此信号一出现, CPU就会在现行指令结束后引起中断。



3.1 8086/8088微处理器

3.1.4 8086/8088 的引脚信号和功能 (续11)

g) *RESET* (引脚21): 复位信号, 输入, 高有效

与8284A (时钟发生/驱动器) 的复位输出端相连;

8086/8088要求复位脉冲宽度不得小于4个时钟周期, 而初次接通电源时所引起的复位, 则要求维持的高电平不能小于50 μ s;

复位后, CPU的主程序流程恢复到启动时的循环待命初始状态。

表 3.3 复位后内部寄存器的状态

内部寄存器	状态
标志寄存器	清除
IP	0000H
CS	FFFFH
DS	0000H
SS	0000H
ES	0000H
指令队列缓冲器	清除



3.1 8086/8088微处理器

3.1.4 8086/8088 的引脚信号和功能 (续12)

h) **CLK** (引脚19) : 系统时钟, 输入

通常与**8284A**时钟发生器的时钟输出端**CLK**相连, 该时钟信号的低/高之比常采用**2:1**(占空度为**1/3**)。



3.1 8086/8088微处理器

3.1.4 8086/8088 的引脚信号和功能 (续13)

⑤ 其它控制线(引脚24-31): 8个

这些控制线的性能, 根据方式控制线(引脚33) MN/\overline{MX} 所处状态而定。

$MN/\overline{MX} = 1$, 8086/8088工作于**最小方式**,
全部控制信号由**CPU**本身提供;

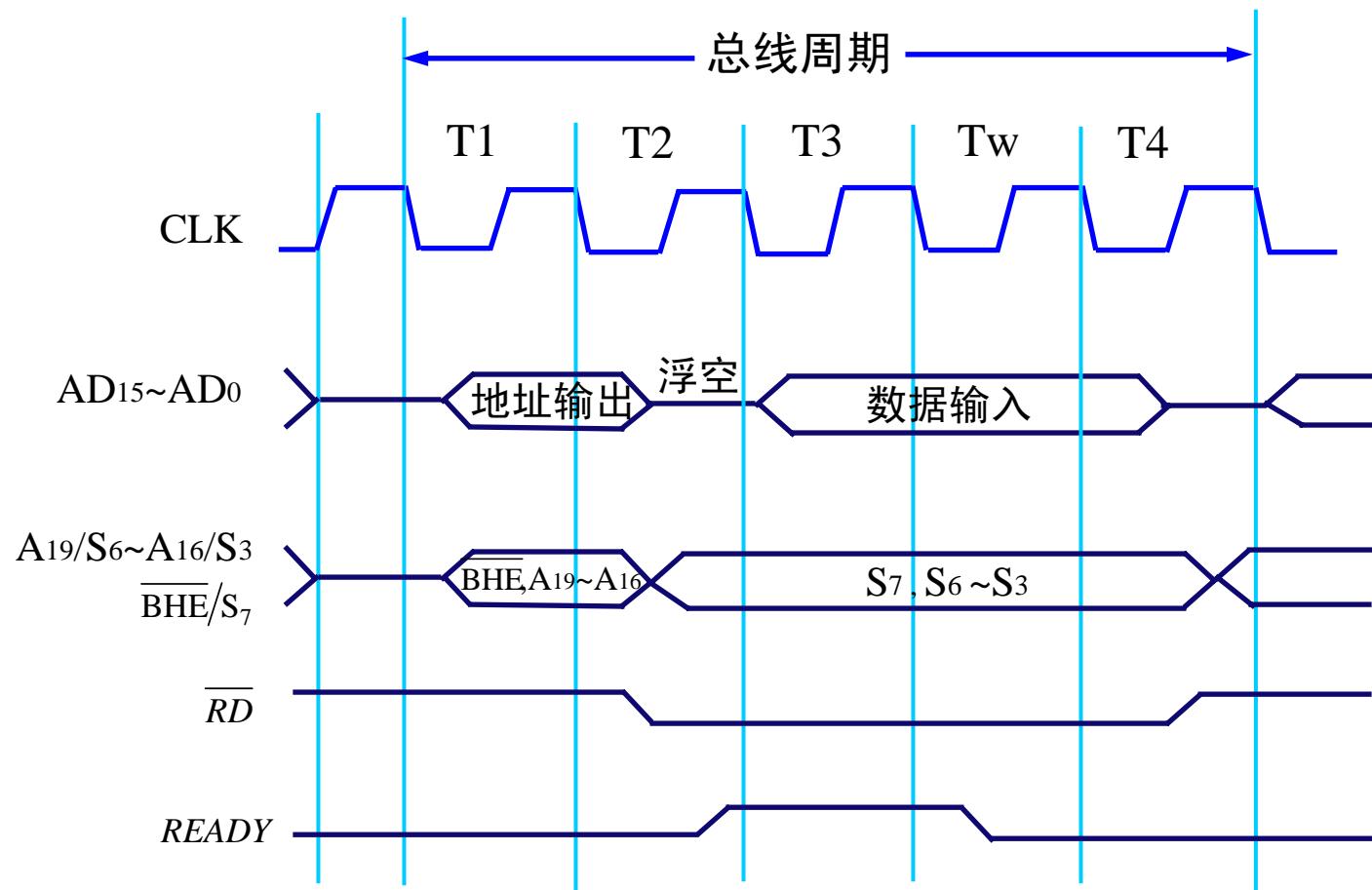
$MN/\overline{MX} = 0$, 8086/8088工作于**最大方式**,
控制信号由**8288总线控制器**提供。



3.1 8086/8088微处理器

3.1.4 8086/8088 的引脚信号和功能 (续14)

时序：是计算机操作运行的时间顺序。





3.2 8086/8088系统最小/最大工作方式

3.2.1 最小方式

MN/\overline{MX} 接电源电压时，系统工作于**最小方式**，即**单处理器系统方式**；

时钟发生/驱动器8284A：外接晶体的基本振荡频率为15MHZ，经8284A三分频后。送给CPU作系统时钟CLK。

8位地址锁存器8282：锁存后地址信号，在整个周期保持不变。

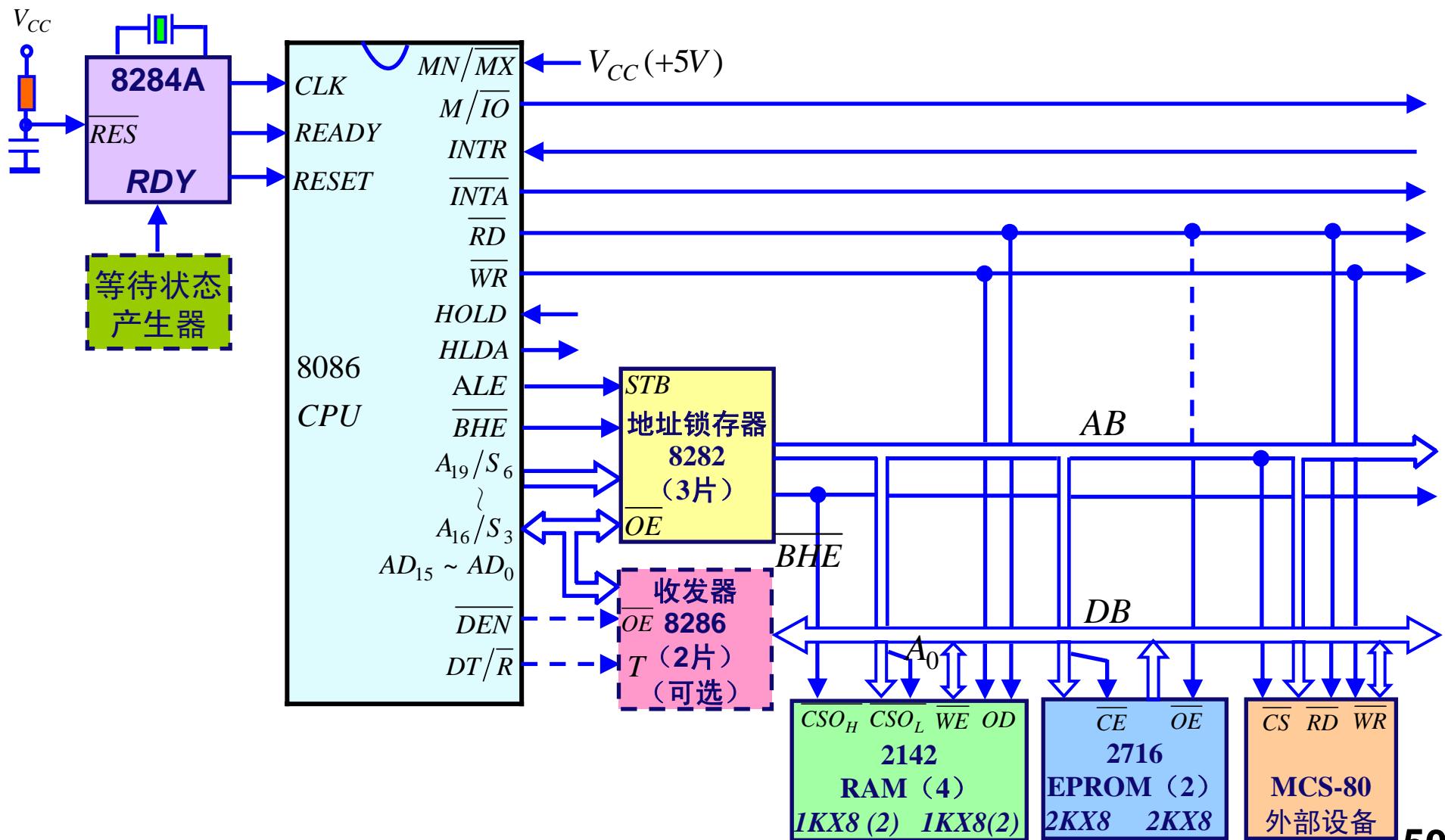
三态输出的8位数据总线收发器8286：用于需要增加驱动能力的系统。

等待状态产生电路：向8284A的 RDY端提供一个信号，经同步后，向CPU的READY线发“准备就绪”信号，通知CPU数据传送已完成，可退出当前总线周期。



3.2 8086/8088系统最小/最大工作方式

3.2.1 最小方式 (续1)





3.2 8086/8088系统最小/最大工作方式

3.2.1 最小方式（续2）

- a) \overline{INTA} (interrupt acknowledge)：引脚24
中断响应信号，输出，低有效

用于CPU对来自外设的中断请求作出响应。发出两个连续负脉冲。

第1个负脉冲：通知外设端口，中断请求被允许；
第2个负脉冲：外设往数据总线上发送中断类型码。

- b) ALE (address latch enable)：引脚25
地址锁存允许信号，输出，高有效

CPU在每个总线周期的T₁状态发出的，作为地址锁存器的地址锁存信号。



3.2 8086/8088系统最小/最大工作方式

3.2.1 最小方式（续3）

c) \overline{DEN} (data enable) : 引脚26

数据允许信号, **输出**, **三态**, **低有效**

为**总线收发器** \overline{OE} 端提供控制信号, 决定是否允许数据通过。

d) DT/R (data transmit/receive) : 引脚27

数据收发控制信号, **输出**, **三态**

控制**总线收发器**的数据**传送方向**,

高电平时: 数据**发送**;

低电平时: 数据**接收**。



3.2 8086/8088系统最小/最大工作方式

3.2.1 最小方式（续4）

e) M/\overline{IO} (memory/input and output): 引脚28

存储器/输入输出控制信号, **输出, 三态**

用于**区分**CPU进行存储器访问还是IO访问,
高电平: 访问存储器; 低电平: 访问IO。

f) \overline{WR} (write) : 引脚29

写控制信号, **输出, 三态, 低有效**

表示CPU当前进行存储器或者IO的**写操作**。



3.2 8086/8088系统最小/最大工作方式

3.2.1 最小方式（续5）

g) *HOLD* (hold request) : 引脚30

总线保持请求信号, **输入**, 高有效

是系统中的**其它**总线主控部件向CPU发出的**请求占用总线**的控制信号。

h) *HLDA* (hold acknowledge) : 引脚31

总线保持应答信号, **输出**, 高有效

与HOLD配合使用的联络信号,

有效期间, 所有与三态门连接的引脚处于浮空, 让出总线。



3.2 8086/8088系统最小/最大工作方式

3.2.2 最大方式

MN/\overline{MX} 接地时，系统工作于最大方式，含两个或多个处理器，需解决主处理器和协处理器之间协调工作，以及总线共享问题。

时钟发生/驱动器8284A：外接晶体的基本振荡频率为15MHz, 经8284A三分频后。送给CPU作系统时钟CLK。

8位地址锁存器8282：锁存后地址信号，在整个周期保持不变。

三态输出的8位数据总线收发器8286：用于需要增加驱动能力的系统。



3.2 8086/8088系统最小/最大工作方式

3.2.2 最大方式（续1）

等待状态产生电路：向8284A的 RDY端提供一个信号，经同步后，向CPU的READY线发“准备就绪”信号，通知CPU数据传送已完成，可退出当前总线周期。

总线控制器8288：对CPU发出的控制信号**变换、组合**，得到对M或I/O的读/写信号，对8282和8286的控制信号。



3.2 8086/8088系统最小/最大工作方式

3.2.2 最大方式（续2）

最小方式：控制信号 M/\overline{IO} 或 (\overline{M}/IO) 、 \overline{WR} 、 \overline{INTR} 、 ALE 、 DT/\overline{R} 、 \overline{DEN} 直接从CPU的第24 ~ 29脚送出。

最大方式：从状态信号 $\overline{S_2}$ 、 $\overline{S_1}$ 、 $\overline{S_0}$ 组合中得到。



3.2 8086/8088系统最小/最大工作方式

3.2.2 最大方式 (续3)

$\overline{S_2}$ 、 $\overline{S_1}$ 、 $\overline{S_0}$ 的代码组合和对应的操作

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

有源状态
 $\overline{S_2}, \overline{S_1}, \overline{S_0}$
至少一个
为低电平

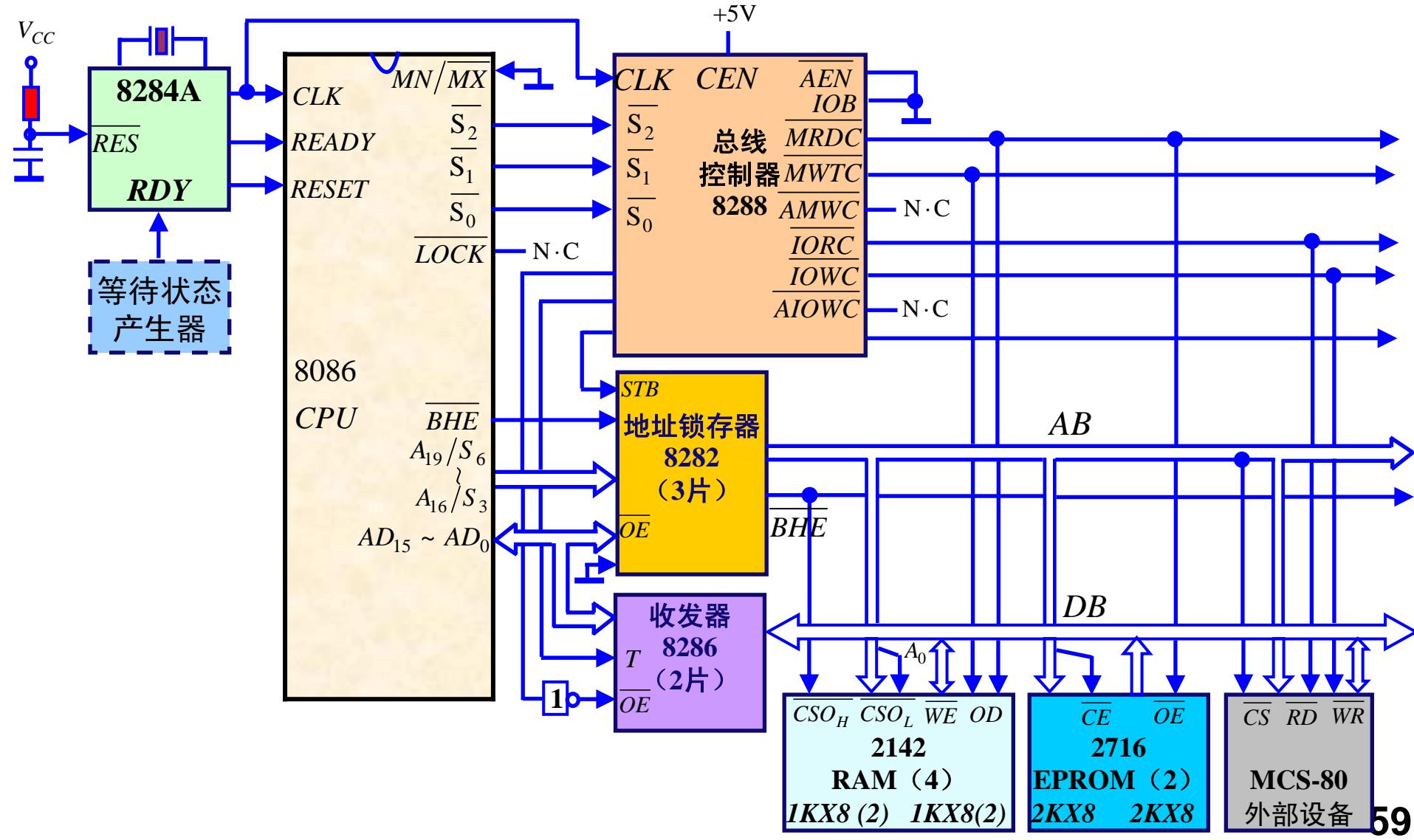
有源状态：三者只要有1个信号为低电平，对应了某一总线操作过程。

无源状态：三者均为高，此时，1个总线操作过程就要结束，另1个新的总线周期还未开始。



3.2 8086/8088系统最小/最大工作方式

3.2.2 最大方式 (续4)





3.2 8086/8088系统最小/最大工作方式

3.2.2 最大方式 (续5)

- ① QS_1 、 QS_0 (instruction queue status)，引脚24,25
指令队列状态信号，输出

提供本总线周期的前一个时钟周期中指令队列的状态，
便于外部对8086/8088内部指令队列的跟踪。

QS_1 、 QS_0 的代码组合和对应的操作

QS_1	QS_0	含义
0	0	无操作
0	1	从指令队列的第一个字节中取走代码
1	0	队列为空
1	1	除第一个字节外，还取走了后续字节中的代码



3.2 8086/8088系统最小/最大工作方式

3.2.2 最大方式（续6）

② \overline{LOCK} (lock) : 引脚29

总线封锁信号, **输出**

由**指令前缀** $LOCK$ 产生, $LOCK$ 前缀后面一条指令执行完毕, $LOCK$ 信号撤销。

低电平时: CPU独占总线, 系统中其它总线主部件就不能占用总线。



3.2 8086/8088系统最小/最大工作方式

3.2.2 最大方式 (续7)

③ $\overline{RQ}/\overline{GT_1}$ 、 $\overline{RQ}/\overline{GT_0}$ (request/grant) : 引脚30,31

总线请求输入/总线请求允许输出

供CPU以外的两个处理器，用来发出使用总线的请求信号，和接收CPU对总线请求信号的回答信号。

二者都是双向的，信号在同一引线上传输，当方向相反。 $\overline{RQ}/\overline{GT_0}$ 有更高优先权。



3.2 8086/8088系统最小/最大工作方式

3.2.2 最大方式 (续8)

④ \overline{MRDC} (memory read command) : 对存储器读

\overline{MWTC} (memory write command) : 对存储器写

\overline{AMWC} (advanced memory write command) : 超前写存储器命令

\overline{IOWC} (I/O write command) : 对I/O写

\overline{IODC} (I/O read command) : 对I/O读

\overline{AIOWC} (advanced I/O write command) : 超前写I/O命令

\overline{INTA} (interrupt acknowledge) : 中断响应信号

3.3 8086/8088 存储器



3.3.1 存储器组织

- ✓ 20条AB，寻址1M存储空间；
- ✓ 按字节组织，每个字节唯一地址；
- ✓ 字节：顺序存放
- ✓ 字：低位字节放在低地址中
高位字节放在高地址中
- ✓ 双字：低位字是偏移量
高位字是段地址
- ✓ 规则字：低位字节存放在偶数地址
- ✓ 非规则字：低位字节存放在奇数地址

MEM	
19H	0C
1AH	30
1BH	90
1CH	10
1DH	45
1EH	67
1FH	AB
20H	CD
21H	34
22H	57
23H	13
24H	59
25H	E0
26H	48
27H	4A
28H	43
29H	00
2AH	5D
2BH	3E

指令OR AL, IMMED

指令NOP

字节数据

字节数据

规则字数据

(AB67H)

字节数据

非规则字数据

(5734H)

指令ADC r, m/r (字)

指令DEC AX

指令DEC DX

指针数据

段地址: 3E5DH

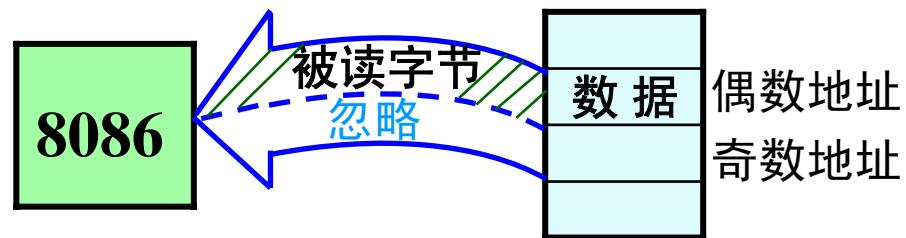
偏移量: 0043H

3.3 8086/8088 存储器

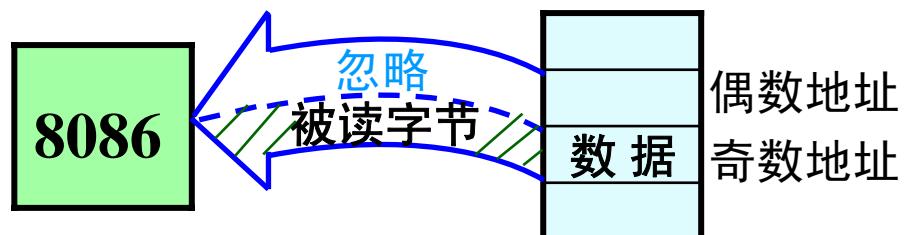


3.3.1 存储器组织 (续1)

字节的存取需要1个总线周期。



(a) 读偶数地址中的字节



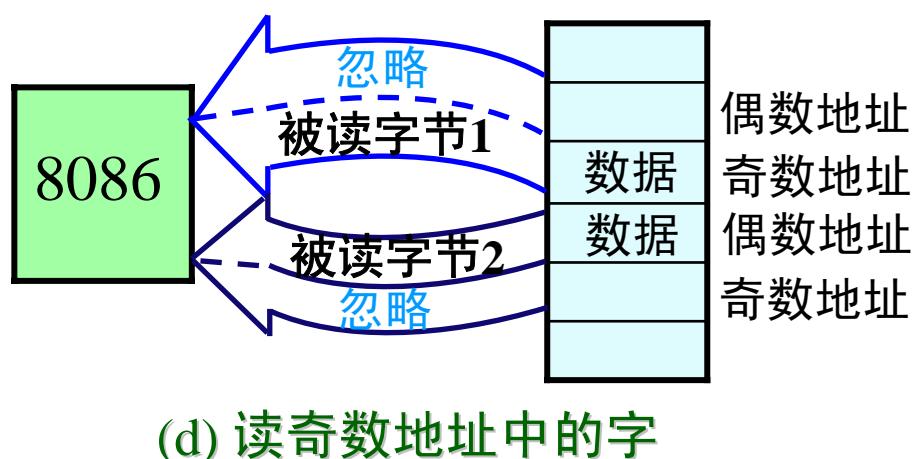
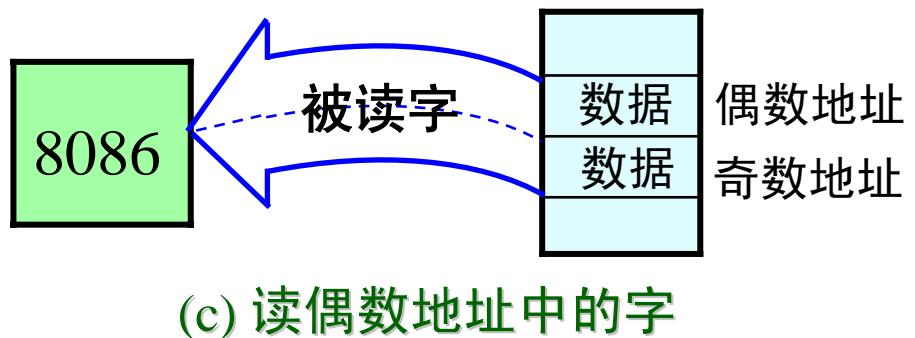
(b) 读奇数地址中的字节

3.3 8086/8088 存储器



3.3.1 存储器组织 (续2)

规则字的存取需要1个总线周期，非规则字的存取需要2个总线周期。



3.3 8086/8088 存储器

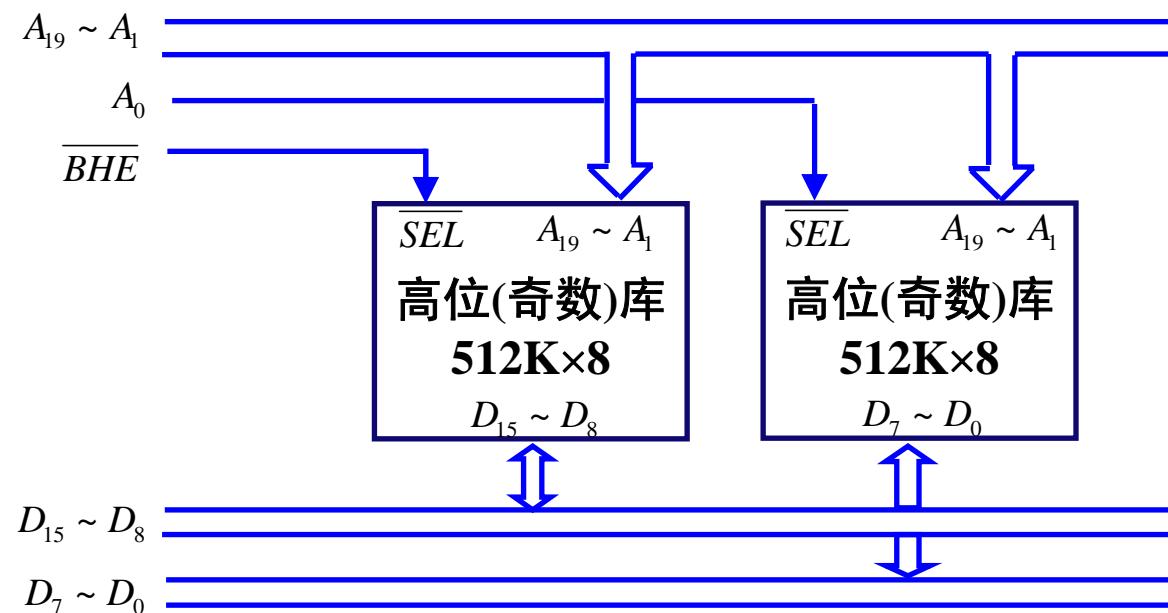


3.3.1 存储器组织 (续3)

1MB 存储空间，分成2个512 KB的存储体(存储库)；

低位库：与数据总线 $D_7 \sim D_0$ 相连，每个地址为**偶数地址**。

高位库：与数据总线 $D_{15} \sim D_8$ 相连，每个地址为**奇数地址**。



3.3 8086/8088 存储器



3.3.1 存储器组织 (续4)

地址总线 $A_{19} \sim A_1$ 可同时对高、低位库的存储单元寻址，
 A_0 或 \overline{BHE} 用于库的选择，分别接到库选择端 \overline{SEL} 。

\overline{BHE}	A_0	操作	使用的数据总线
0	0	同时读/写高低两个字节	$AD_{15} \sim AD_0$
0	1	只读/写奇地址的高位	$AD_{15} \sim AD_8$
1	0	只读/写偶地址的低位	$AD_8 \sim AD_0$
1	1	不传送	



3.3 8086/8088 存储器

3.3.1 存储器组织 (续5)

8088系统中，寻址空间1MB，单一的存储体， $1M \times 8$ 位。

20根地址线与8根数据线分别同8088CPU的对应地址线和数据线相连。

8088CPU每访问1次存储器只读/写1个字节信息。任何数据字都需要两次访问存储器才能完成读/写操作。

8088系统中，程序运行速度比在8086系统中慢些。

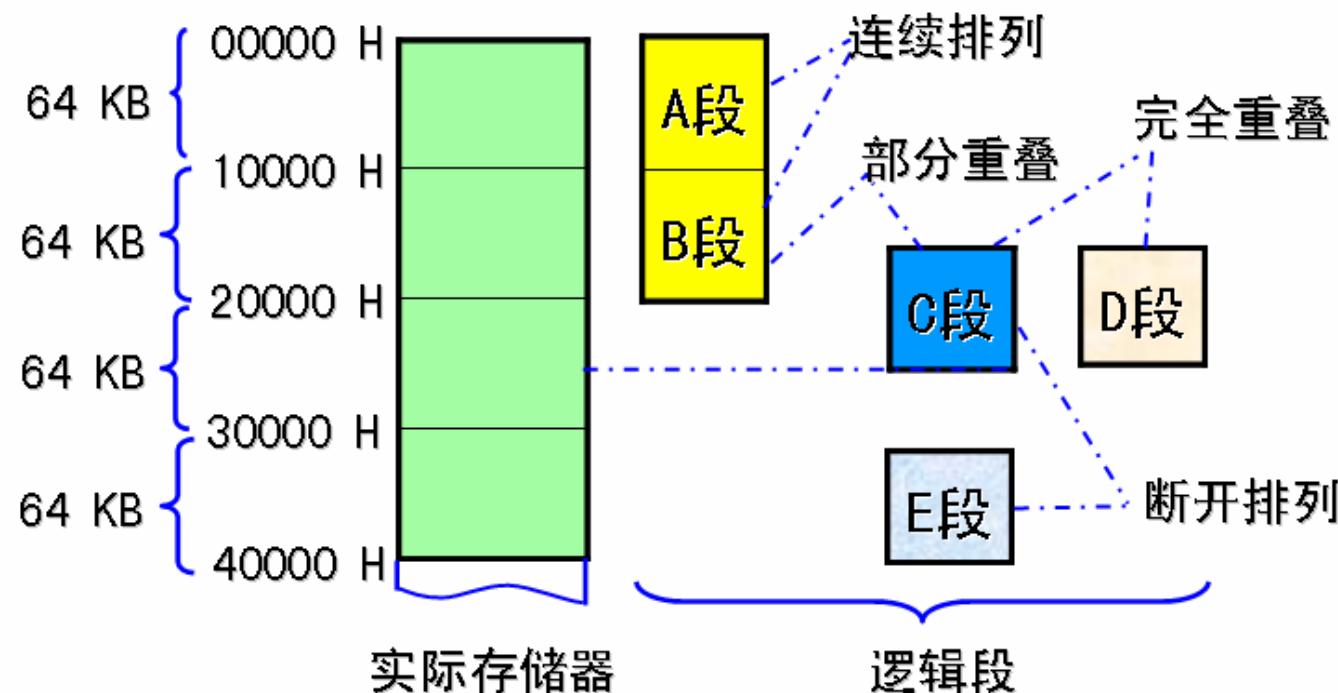
3.3 8086/8088 存储器



3.3.2 存储器分段

20根地址总线，允许寻址1MB存储空间；IP、SP、SI、DI都是16位，直接寻址大空间64KB；

为了寻址1MB存储空间，实行分段管理，每一段最长为64KB。段和段之间关系的：连续、分开、部分重叠、完全重叠；





3.3 8086/8088 存储器

3.3.2 存储器分段 (续1)

- ✓ 每个段大小可从一个字节开始，任意递增，最多包含64KB长的连续存储单元；
- ✓ 每个段的20位起始地址(段基址)，是能被16整除的数，即最后4位为零，可通过程序在段寄存器中装入16位段地址来设置；段地址是20位段基址的前16位。
- ✓ 1个程序所用的具体存储空间：1个或多个逻辑段；
- ✓ 段基址存在CS、SS、DS、ES中，程序可从中给出的逻辑段中存取代码和数据；
- ✓ 段区的分配是由操作系统完成的，系统允许程序员指定。



3.3 8086/8088 存储器

3.3.3 实际地址和逻辑地址

实际地址(也称为物理地址): CPU对存储器进行访问时的实际寻址所使用的地址，对8086/8088来讲，用20位二进制数或5位十六进制数表示。

逻辑地址: 程序和指令中表示的一种地址，由段地址和偏移地址两部分组成，用无符号的16位二进制或4位十六进制数表示。

段地址: 16位段寄存器直接给出的16位地址。

偏移地址(也称为偏移量或偏移): 由指令寻址时的寄存器组合与位移量之和，16位的偏移量。

表示所寻址的地址单元距离段起始地址之间的偏移。

3.3 8086/8088 存储器



3.3.3 实际地址和逻辑地址 (续1)

一个实际地址可对应多个逻辑地址。

实际地址11245H可从两个部分重叠的段中得到：

段地址：1123H

偏移地址：15H

实际地址：11230H+15H=11245H

段地址：1124H

偏移地址：05H

实际地址：11240H+05H=11245H

段地址	偏移量	M	实际地址
1123H	0000 H		11230 H
1123H	0001 H		11231 H
1123H	0002 H		11232 H
1123H	0003 H		
1123H	0004 H		
1123H	0005 H		
1123H	0006 H		
1123H	0007 H		
1123H	0008 H		
1123H	0009 H		
1123H	000A H		
1123H	000B H		
1123H	000C H		
1123H	000D H		
1123H	000E H		1123E H
1123H	000F H		1123F H
1124H	0010 H		11240 H
1124H	0011 H		11241 H
1124H	0012 H		11242 H
1124H	0013 H		11243 H
1124H	0014 H		11244 H
1124H	0015 H		11245 H
	FFFE H	≈	2122E H
	FFFF H	≈	2122F H

3.3 8086/8088 存储器



3.3.3 实际地址和逻辑地址 (续2)

段地址来源于: CS, DS, SS, ES

偏移地址来源于: IP, SP, BP, SI, DI

} BIU根据执行操作的种类和
要取得的数据类型来确定

存储器操作涉及的类型	正常使用段基址	可使用段基址	偏移地址
取指令	CS	无	IP
堆栈操作	SS	无	SP
变量(下面情况除外)	DS	CS,ES,SS	有效地址
源数据串	DS	CS,ES,SS	SI
目的数据串	ES	无	DI
作为基址寄存器使用的BP	SS	CS,DS,ES	有效地址



3.3 8086/8088 存储器

3.3.4 堆栈

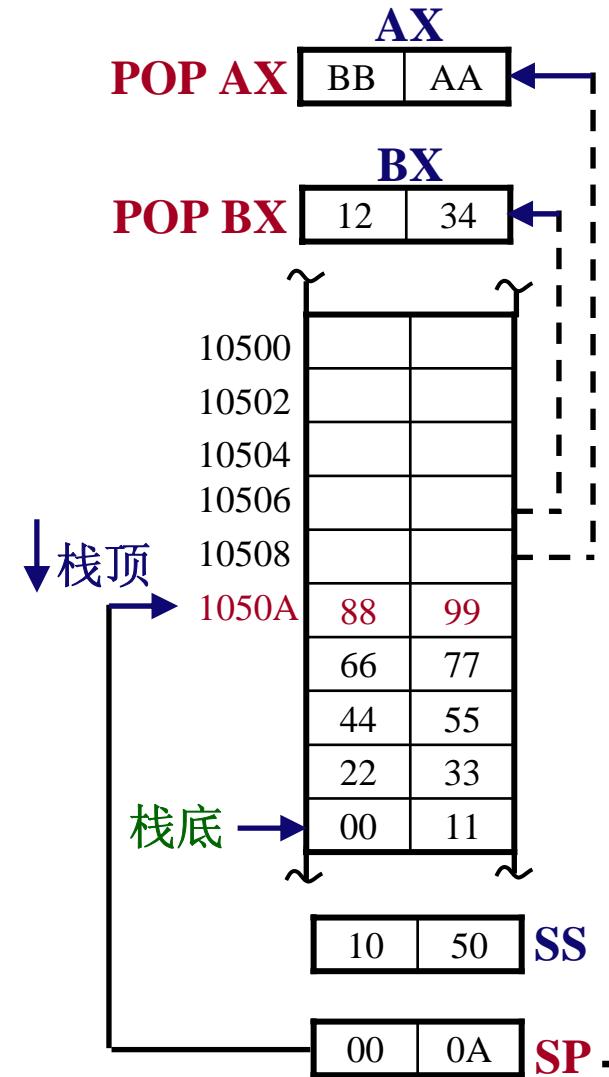
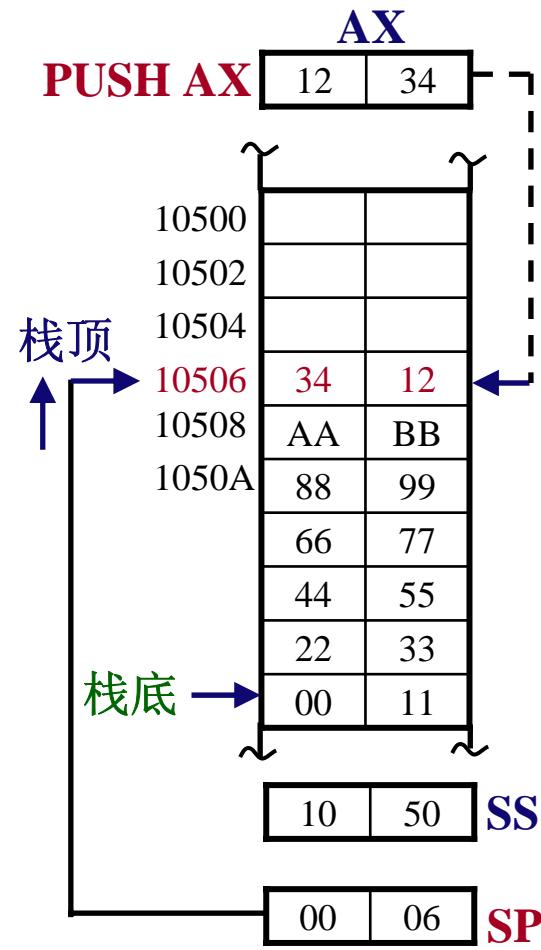
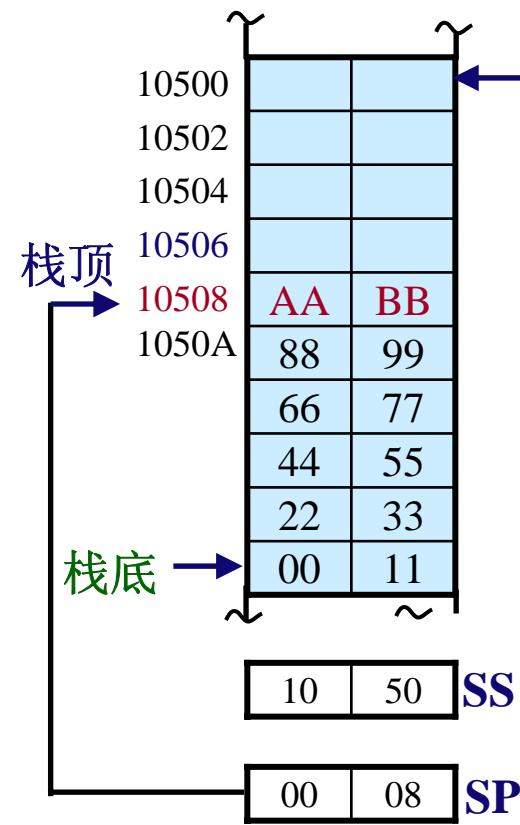
- ① 8086/8088系统中的堆栈是用**段定义语句**在存储器中定义的一个堆栈段，如同其它逻辑段，可在1MB的存储空间中浮动。
- ② 一个系统堆栈**数目**不受限制，栈长度不超过64KB。
- ③ 堆栈由段寄存器**SS**和堆栈指针**SP**来寻址
 - SS：给出堆栈段的段基址；
 - SP：给定当前栈顶，即指出从堆栈的段基址到栈顶的偏移量。
- ④ **栈顶**是堆栈操作的唯一出口，是栈地址较小的一端。

3.3 8086/8088 存储器



3.3.4 堆栈 (续1)

为加快堆栈操作的速度，均以字为单位进行。



3.3 8086/8088 存储器



3.3.5 “段加偏移”寻址机制允许重定位

- ① 重定位：一个完整的程序块或数据块，可在存储器所允许的空间内任意浮动，并定位到一个新的可寻址的区域。
- ② “段加偏移”寻址机制允许重定位(或再定位)是一种重要特性。
- ③ 原来为8086在实模式下运行所编写的程序，在以后80286以上高型号微处理器中，当系统由实模式转换为保护模式时也可运行。
- ④ 各种通用计算机系统在运行同一软件和数据时能够保持兼容性。



第三章第1部分结束！