

第二章 微型计算机中的微处理器

内容:

- 8086/8088 CPU的编程结构
- 8086 / 8088 CPU的引脚信号和工作模式
- 寄存器结构
- 8086 / 8088的存储器组织
- 8086的I / O组织
- 8086 / 8088微处理器典型时序分析

Intel 8086/8088 结构

8086/8088微处理器是Intel公司推出的第三代CPU芯片，它们的内部结构基本相同，都采用16位结构进行操作及存储器寻址，两种处理器都封装在相同的40脚双列直插组件（DIP）中。

在Intel 8080与8085的基础上发展起来的。

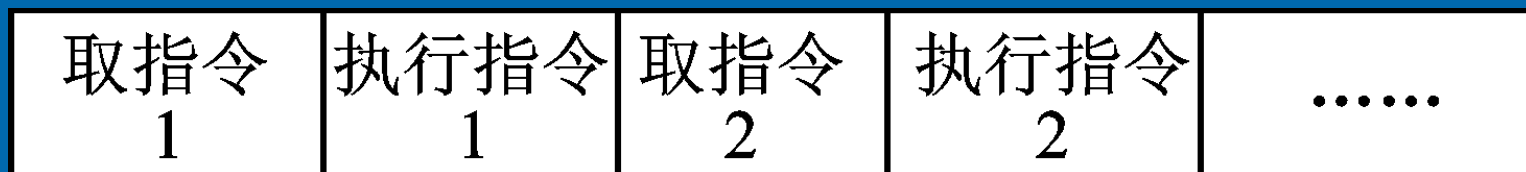
结构特点:

- (1) 内部结构是16位的内部寄存器, 内部运算部件, 内部操作按16位设计);
- (2) 外部数据总线16 (8086) /8 (8088)位, 能处理16位数据, 也能处理8位数据;
- (3) 汇编语言与8080/8085兼容, 即能执行整套8080/8085的指令, 增加了许多16位操作指令;
- (4) 20条地址总线, 直接寻址能力1M字节;
- (5) 40条引线、双列直插式;
- (6) 单相时钟;
- (7) 电源为5V。

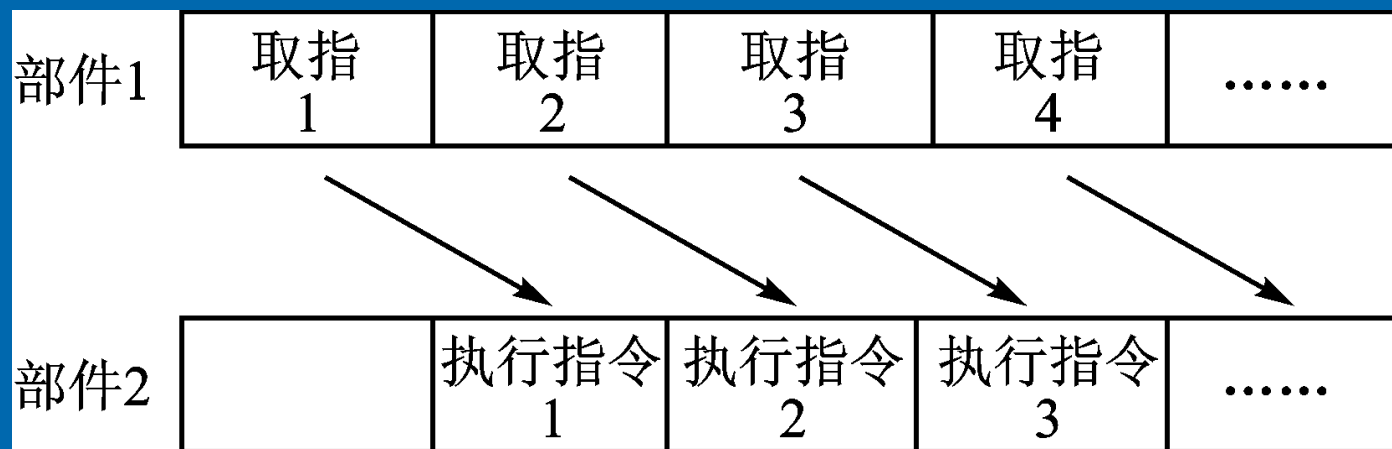
8088微处理器与8086微处理器的主要区别: 对外的数据线只有8位, 目的是为了更方便地与8位I/O接口芯片相兼容。

2.1 8086/8088 CPU的编程结构

传统结构的CPU执行程序时，取指令与执行指令交替进行：



为提高CPU的工作效率，8086 / 8088CPU采用流水线处理方式，取指令与执行指令同时进行。



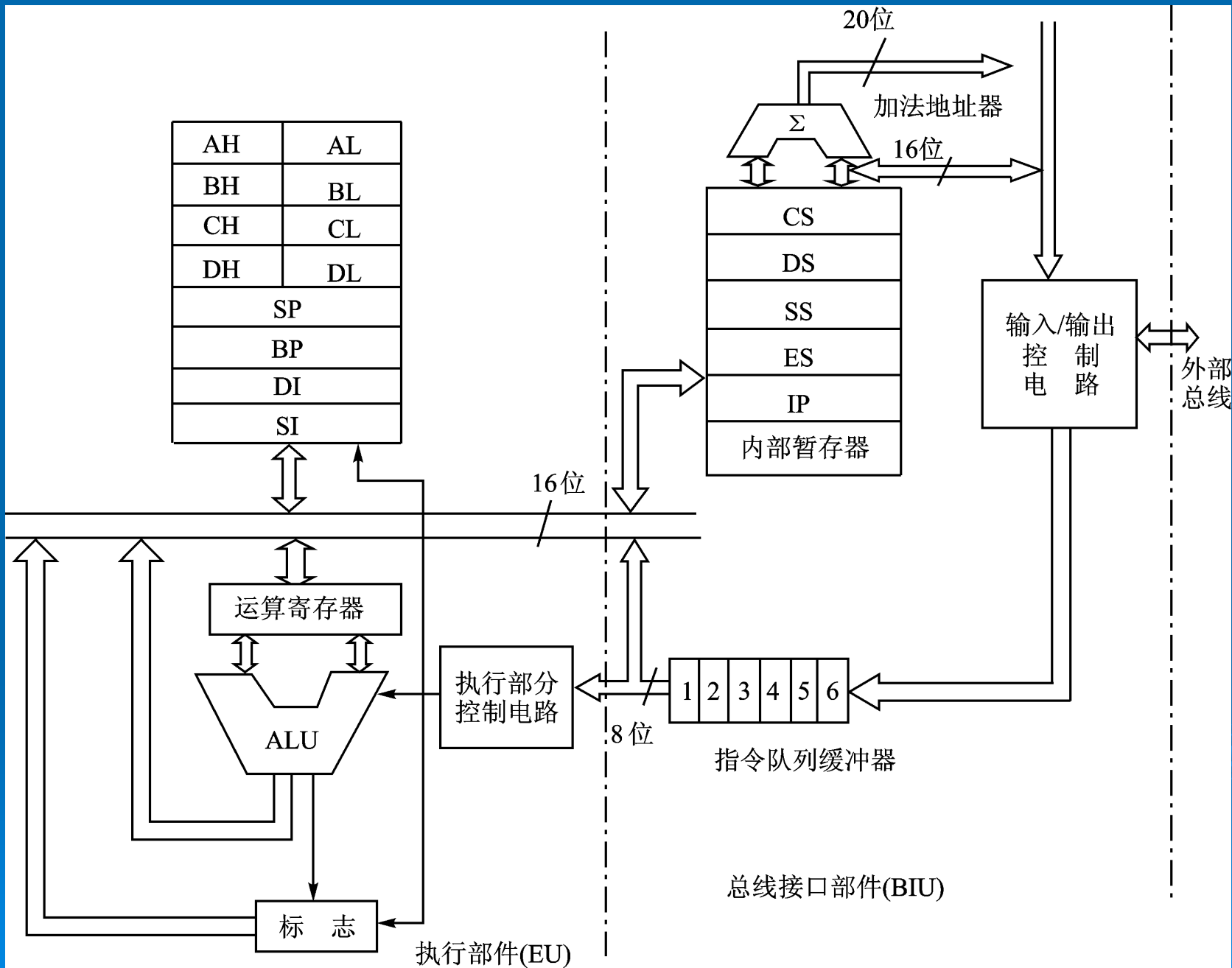
一方面提高了执行速率；

另一方面降低了与之相配的存储器的存取速度的要求。

8086/8088的编程结构分两部分：

1、总线接口单元 BIU (Bus Interface Unit)

2、执行部件 EU (Execution Unit)



8086 CPU的功能结构

一、总线接口单元 BIU (Bus Interface Unit)

1、功能：负责与 存储器、I/O 端口进行数据传送。

具体讲：

取指令：总线接口部件从内存中取出指令后送到指令队列。

预取指令。

配合EU执行指令，存取操作数和运算结果。

2、组成

- 段地址寄存器（CS、DS、ES、SS）。
- 16位指令指针寄存器IP。
- 地址加法器(形成20位物理地址),
- 6字节（8086）或4字节（8088）的指令队列,
- 内部寄存器
- 总线控制电路:

3、注：

1) 指令队列

8086 的指令队列为6个字节，

8088 的指令队列为4个字节。

2) 指令执行顺序

顺序指令执行：指令队列存放紧接在执行指令后面的那一条指令。

执行转移指令：BIU 清除指令队列中的内容，从新的地址取入指令，立即送往执行单元，然后再从新单元开始重新填满队列。

二、EU（Execution Unit）执行单元

1、功能：负责指令执行。

2、组成：

- 4个通用寄存器：AX、BX、CX、DX
- 4个专用寄存器：BP、SP、SI、DI，
- 标志寄存器（PSW）：为16位，存放指令执行结果的特征和处理器状态，如结果为0，为负，单步执行等。
- 算术逻辑单元ALU：
16位加法器。完成8位 / 16位二进制数的算术逻辑运算。
- EU 控制系统：
接受从总线接口单元的指令队列中取来的指令代码，
对其译码和向EU内各有关部分发出时序命令信号，协调执行指令规定的操作。

由此可见，8086/8088微处理器：

BIU和EU分开，取指和执行可以重迭，大大减少了等待取指所需的时间，提高CPU的利用率。

三、8086 / 8088处理器的启动和程序执行过程

1、CPU的启动

8086 / 8088系统中，CPU被启动后，处理器内部的各寄存器和标志寄存器的内容自动设置为：

CS FFFFH	DS 0000H
SS 0000H	ES 0000H
IP 0000H	指令队列空
FR 0000H	(禁止中断)

因CS=FFFFH，IP=0000，所以8086 / 8088将从地址FFFF0H开始执行指令。故8086 / 8088引导程序的入口地址在FFFF0H。

2、程序执行过程

设程序的指令代码已存放在存储器中。为执行程序，CPU按照时钟节拍，产生一系列控制信号，有规则地重复进行以下过程。

- (1) BIU从存储器中取出一条指令存入指令队列。
- (2) EU从指令队列取指令并执行指令。BIU利用总线空闲时间，从内存取第二条指令或取第三条指令存入指令队列。
- (3) EU执行下一条指令。如果前面一条指令有写存储器的要求，则通知BIU把前条指令结果写到存储器中，然后再取指令存入指令队列。

- (4) 如指令执行要求读取操作数，由BIU完成。
- (5) EU执行再下一条指令，返回（1）处继续执行上述操作过程。

所以，程序的执行过程就是CPU取指令、分析指令、执行指令，再取指令这样一个循环重复过程。

在指令执行过程中，**利用EU分析指令操作码和执行指令时不占用总线操作时间的特点**，BIU自动地通过总线读取存贮器中的指令码存入BIU指令队列，从而使BIU与EU并行工作，提高CPU执行指令的速度。

四、8086 / 8088 CPU的总线周期概念

总线周期：BIU通过系统总线对存储器或I / O端口进行一次读 / 写操作的过程称为一个总线周期。

8086 / 8088CPU的一个基本总线周期由**4个时钟周期**（ $T_1 \sim T_4$ ）组成，也称4个T状态。

CPU在每个时钟周期内完成若干个基本操作，具体是：

T_1 状态：CPU向多路复用总线上发送地址信息指出要寻址的存储单元或外设端口地址。

T_2 状态：CPU从总线上撤消地址，使总线的低16位置为高阻抗状态，为传输数据作准备。总线的高4位输出本总线周期状态信息。这些状态信息用来表示中断允许状态、当前正在使用的段寄存器等。

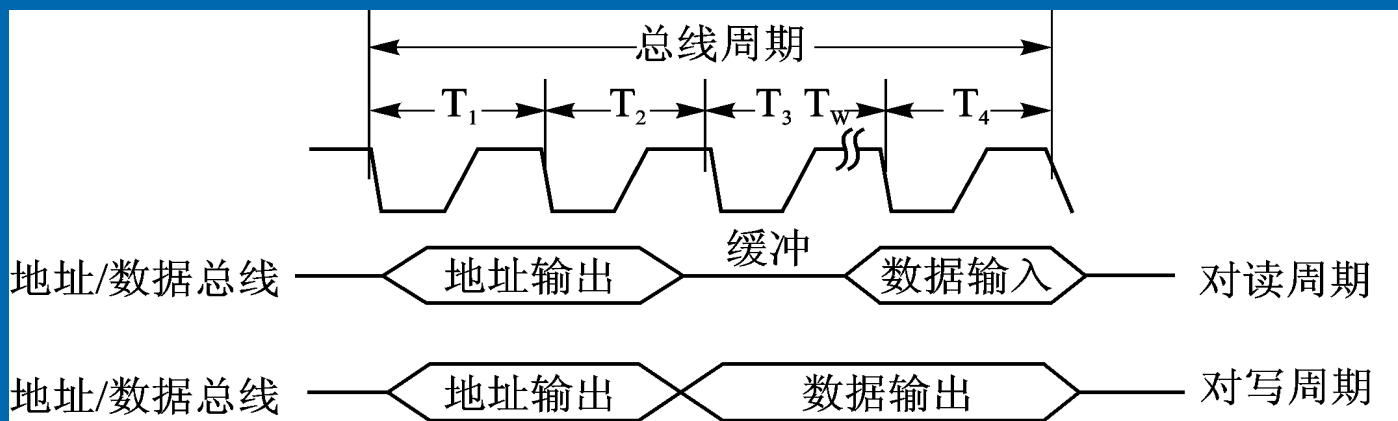
T_3 状态：CPU在总线的高4位继续输出总线周期状态信号。在总线的低16位出现由CPU写出的数据，或者从存储器或I / O端口读入的数据。

T_4 状态：总线周期结束。

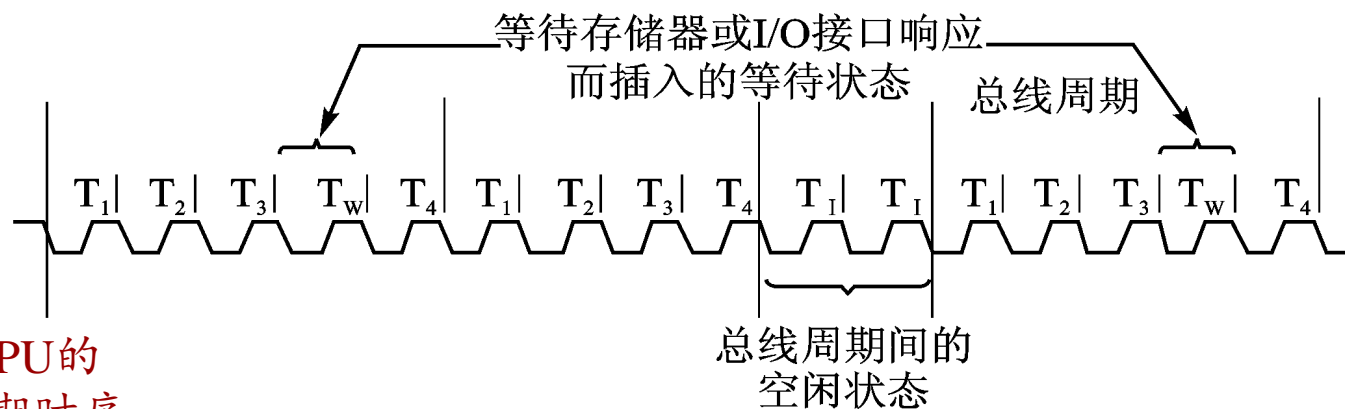
T_W 等待状态：如果存储器或I / O设备不能及时配合CPU传送数据，这时外设或存储器会通过“READY”信号线在 T_3 状态启动之前向CPU发数据“未准备好”信号，迫使CPU在 T_3 状态后插入等待状态 T_W 。 T_W 状态的总线情况与 T_3 周期的情况相同。当被选中的存储器或I / O端口有足够的时间来完成读写操作时，就发出“准备好”（Ready）信号，使CPU脱离 T_W 状态继续工作。

。

T_1 空闲状态：如果在一个总线周期之后，不立即执行下一个总线周期，或者当指令队列是满的，执行部件EU又没有访问总线的要求，这时BIU就处于空闲状态。空闲状态，可以包含一个或几个时钟周期。



(a) 典型的总线周期时序



(b) 有空闲状态的总线周期时序

8086/8088 CPU的
典型总线周期时序

2.2 8086/8088CPU的引脚信号和工作模式

一、最大和最小工作模式

最小工作模式：指系统中只有8086 / 8088一个微处理器，构成小规模的应用系统。最小模式也称单处理器模式。在最小模式系统中，所有的系统总线信号都直接由8086 / 8088CPU产生。

最大工作模式：指系统中包含有两个或两个以上的微处理。一个为主处理器（8086 / 8088CPU），其他的称为协处理器，协助主处理器工作。构成较大规模的应用系统。

常与主处理器8086 / 8088CPU配合的协处理器：一个是专用于数值运算的协处理器8087；另一个是专用于输入 / 输出操作的协处理器8089。

最大模式是一个多处理器系统，需要解决主处理器和协处理器之间的协调工作问题和对系统总线的共享控制问题。

二、8086/8088CPU的引脚信号和功能

地	1	40	V_{CC}
AD ₁₄	2	39	A ₁₅
AD ₁₃	3	38	A ₁₆ /S ₃
AD ₁₂	4	37	A ₁₇ /S ₄
AD ₁₁	5	36	A ₁₈ /S ₅
AD ₁₀	6	35	A ₁₉ /S ₆
AD ₉	7	34	BHE/S ₇
AD ₈	8	33	MN/ \overline{MX}
AD ₇	9	32	RD
AD ₆	10	31	HOLD ($\overline{RQ/GT0}$)
AD ₅	11	30	HLDA ($\overline{RQ/GT1}$)
AD ₄	12	29	WR (\overline{LOCK})
AD ₃	13	28	IQ/ \overline{M} ($\overline{S2}$)
AD ₂	14	27	DT/R ($\overline{S1}$)
AD ₁	15	26	DEN ($\overline{S0}$)
AD ₀	16	25	ALE (QS0)
NMI	17	24	INTA (QS1)
INTR	18	23	TEST
CLK	19	22	READY
地	20	21	RESET

地	1	40	V_{CC}
A ₁₄	2	39	A ₁₅
A ₁₃	3	38	A ₁₆ /S ₃
A ₁₂	4	37	A ₁₇ /S ₄
A ₁₁	5	36	A ₁₈ /S ₅
A ₁₀	6	35	A ₁₉ /S ₆
A ₉	7	34	SS ₀
A ₈	8	33	MN/ \overline{M}
AD ₇	9	32	RD
AD ₆	10	31	HOLD ($\overline{RQ/GT0}$)
AD ₅	11	30	HLDA ($\overline{RQ/GT1}$)
AD ₄	12	29	WR (\overline{LOCK})
AD ₃	13	28	IQ/ \overline{M} ($\overline{S2}$)
AD ₂	14	27	DT/R ($\overline{S1}$)
AD ₁	15	26	DEN ($\overline{S0}$)
AD ₀	16	25	ALE (QS0)
NMI	17	24	INTA (QS1)
INTR	18	23	TEST
CLK	19	22	READY
地	20	21	RESET

8088/8086 CPU的引脚

1、 $AD_{15} \sim AD_0$ (Address/Data Bus) 地址/数据复用总线
传送地址时三态输出，传送数据时三态双向输入/输出。

T_1 状态：用来输出访问存储器或I/O端口的地址。

T_2 状态：如果是读周期，则处于浮空（高阻）状态，如果是写周期，则为传送数据。在中断响应及系统总线处于“保持响应”周期时， $AD_{15} \sim AD_0$ 被置成高阻状态。

在8086系统中，常将 AD_0 为低8位数据的选通信号。

2、 $A_{19}/S_6 \sim A_{16}/S_9$ (Address/Status) 地址/状态复用线

作地址线时，高4位 ($A_{19} \sim A_{16}$) 地址，与 $AD_{15} \sim AD_0$ 构成20位访问存储器的物理地址。

作状态线时，输出状态信息 $S_6 \sim S_3$ 。

3、 $\overline{\text{BHE}}/\text{S7}$ (Bus High Enable/Status) 高8位数据总线允许 / 状态复用信号高8位数据有效信号， $\overline{\text{BHE}}$ 低电平有效。和 AD_0 结合起来，指出当前传送的数据在总线上将以何种格式出现。

4、 $\overline{\text{RD}}$ (Read) 读信号。输出、三态、低电平有效。有效时，表示CPU正在对存储器或 $\overline{\text{I/O}}$ 端口进行读操作，具体是对存储器读，还是对 $\overline{\text{I/O}}$ 端口读，由 IO/M (8088为 IO/M) 决定。

5、 READY 存储器或 $\overline{\text{I/O}}$ 口准备就绪,输入。用来使CPU和慢速存储器或 $\overline{\text{I/O}}$ 设备之间实现速度匹配的信号。

在总线操作周期中，CPU会在第3个时钟周期的前沿测试该引脚。

如果测到高有效，CPU直接进入第4个时钟周期。

如果测到无效，CPU将插入等待周期 T_w 。

CPU在等待周期中仍然要监测 READY 信号，有效则进入第4个时钟周期，否则继续插入等待周期 T_w 。

6、 $\overline{\text{TEST}}$ (Test) 测试信号，输入、低电平有效。

用于协调8086/8088CPU与协处理器的工作。

该引脚与WAIT指令配合使用。当CPU执行WAIT指令时，他将在每个时钟周期对该引脚进行测试：如果无效，则程序踏步并继续测试；如果有效，则程序恢复运行。也就是说，WAIT指令使CPU产生等待，直到引脚有效为止。

在使用协处理器8087时，通过引脚和WAIT指令，可使8088与8087的操作保持同步。

7、INTR (Interrupt Request) 可屏蔽中断请求信号,输入、高电平有效。

8、NMI (NO—Maskable Interrupt) 不可屏蔽中断请求信号，输入，高电平有效。

9、RESET (Reset) 复位信号，输入、高电平有效。

10、CLK (Clock) 主时钟输入端

11、 $\overline{\text{MN/MX}}$ (Maximum Mode Control) 最小/最大模式控制信号。

当此引脚接高电平时，CPU工作于最小模式；接低电平时，CPU工作于最大模式。

12、GND、VCC和电源

以上信号是8086/8088CPU工作在最小模式和最大模式时都要用到的。8086/8088的第24~31腿信号在不同模式下有不同的名称和定义。

两种模式下的主要区别体现在第24~31号引脚的功能定义不同。

三、最小模式

1、引腿信号

(1) \overline{IO}/M (Memory/Input and Output) 存储器/输入输出控制信号。用来区分CPU当前是访问存储器还是访问I/O端口。

对于8088CPU，该信号定义为 \overline{IO}/M ，功能相同。

(2) \overline{WR} (Write) 写信号，三态、输出、低电平有效。

有效时，表示CPU当前正在进行写操作。是写存储器还是写I/O端口，由 \overline{IO}/M 来区分。DMA方式时， \overline{IO}/M 被置成高阻态。

(3) \overline{INTA} (Interrupt Acknowledge) 中断响应信号，输出、三态、低电平有效 是对中断请求信号INTR的响应。CPU在整个中断响应周期内发出两个连续的 \overline{INTA} 负脉冲，第一个负脉冲是通知请求中断的外设，其发出的中断请求已得到响应，外设接口收到第二个负脉冲后，向数据总线上送中断类型码，信号通常用来作为读取中断类码的选通信号。

(4) ALE (Address Latch Enable) 地址锁存允许信号，输出、高电平有效。

在任何一个总线周期的 T_1 状态，ALE输出有效电平，表示当前在地址/数据复用总线上输出的是地址信息，锁存器利用它将地址锁存，ALE信号不能浮空。

(5) $\overline{DT}/\overline{R}$ (Data Transmit/Receive) 数据发送 $\overline{DT}/\overline{R}$ 接收控制信号，输出、三态。表明当前总线上数据的流向（高电平时数据自CPU输出（发送）低电平时数据输入CPU（接收））

当使用总线收发器8286 / 8287时，可用 $\overline{DT}/\overline{R}$ 信号来控制总线收发器的数据传送方向。DMA方式时 $\overline{DT}/\overline{R}$ 被置为高阻抗状态。

(6) \overline{DEN} (Data Enable) 数据允许信号，输出、三态、低电平有效。

有效时，表示当前数据总线上正在传送数据。当使用总线驱动器8286/8287时， \overline{DEN} 信号用来作为8286/8287的输出允许控制信号，使之开始传送数据。

(7) HOLD (Hold Request) 总线保持请求信号，输入、高电平有效。作为其它部件向CPU发出使用总线的请求信号。

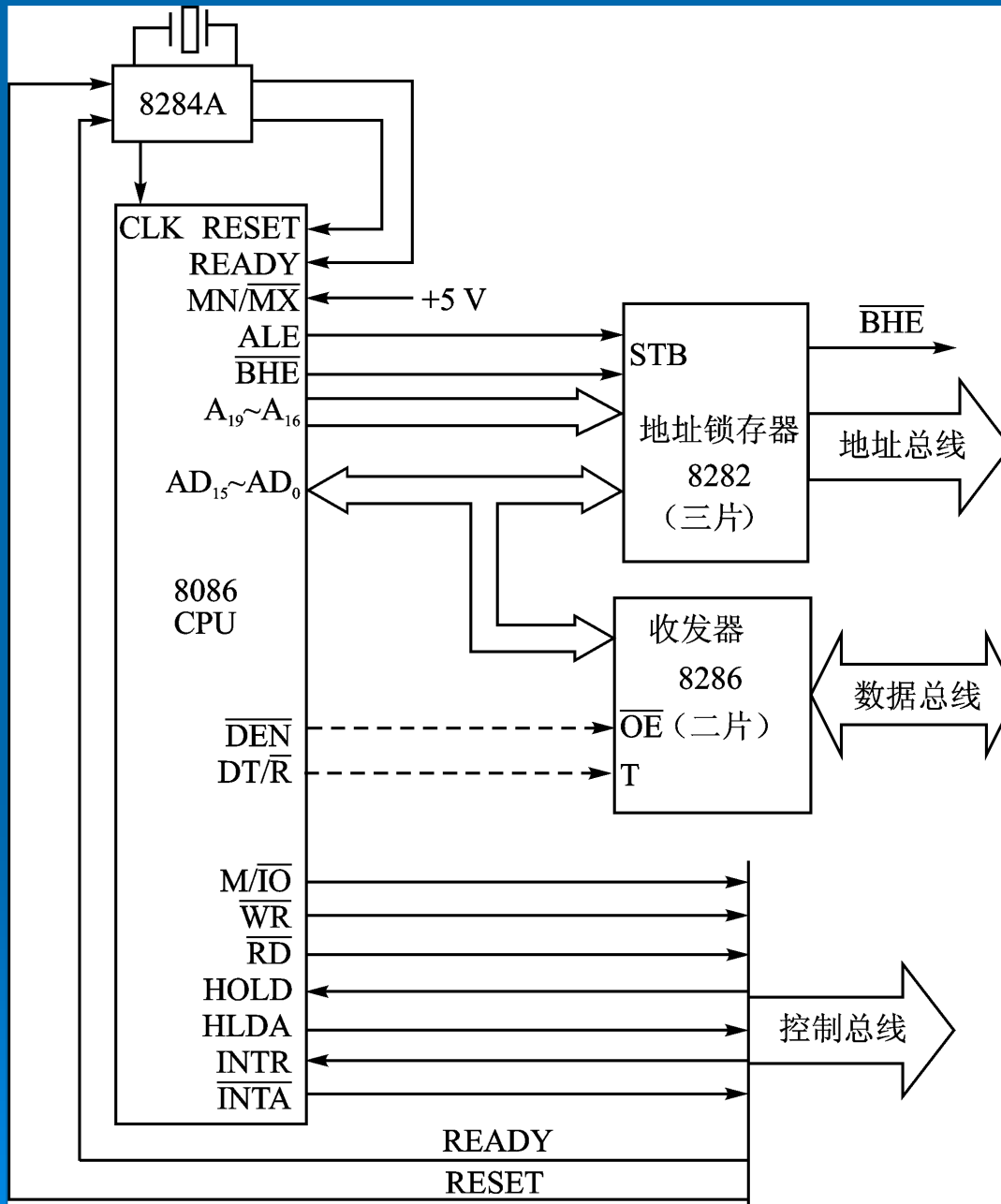
(8) HLDA (Hold Acknowledge) 总线保持响应信号，输出、高电平有效。是CPU对总线保持请求信号HOLD的响应信号。

注：在最小模式下，8088CPU只有8位数据总线，不需要 \overline{BHE} 信号。因此，34引脚定义为 $\overline{SS_0}$ 和 $\overline{IO/\overline{M}}$ 及 $\overline{DT/\overline{R}}$ 信号组合起来，决定了当前总线周期的操作。

8088中 $\overline{SS_0}$ 、 $\overline{IO/\overline{M}}$ 、 $\overline{DT/\overline{R}}$ 信号的组合及其对应的总线操作

$\overline{IO/\overline{M}}$	$\overline{DT/\overline{R}}$	$\overline{SS_0}$	含义
0	0	0	取指令
0	0	1	读存储器
0	1	0	写存储器
0	1	1	无源状态
1	0	0	发中断响应信号
1	0	1	读I/O端口
1	1	0	写I/O端口
1	1	1	暂停

2、系统配置



最小组态工作模式下
总线的形成

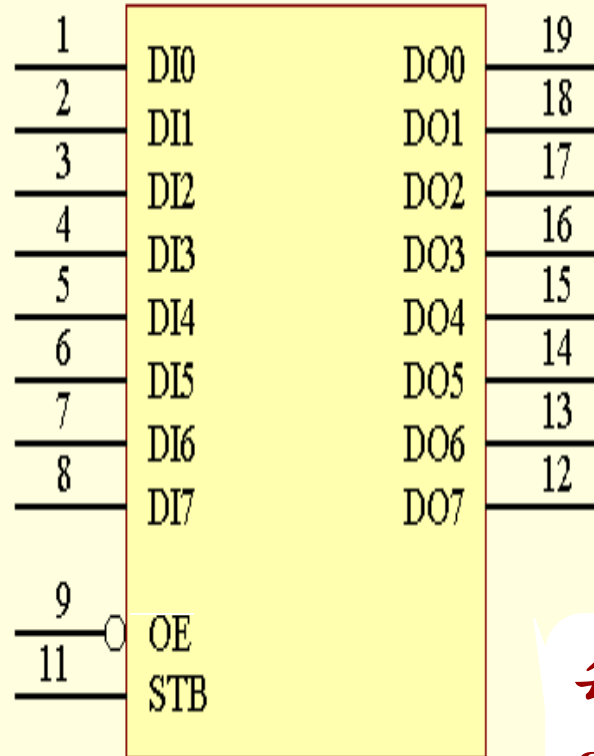
(1) 20位地址总线:
采用3个三态透明
锁存器8282进行锁存
和驱动。

(2) 16位数据总线:
采用数据收发器
8286进行驱动。

(3) 系统控制信号:
由CPU引脚直接提
供。

8086 CPU在最小模
式下的基本配置

Intel 8282



8282

具有三态输出的
TTL电平锁存器

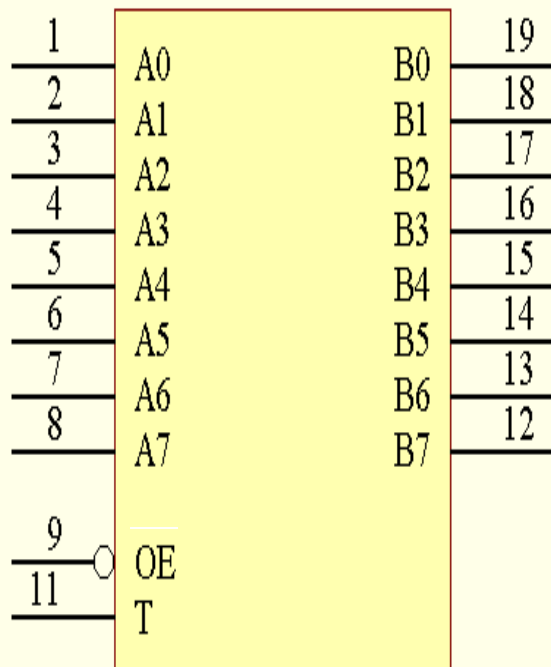
STB 电平锁存引脚
 $\overline{\text{OE}}$ 输出允许引脚

每一位都是一个三态锁存器，
8个三态锁存器的控制端连在一起

Intel 8286

8位双向缓冲器

- 控制端连接在一起，低电平有效
- 可以双向导通
- 输出与输入同相



8286

OE=0, 导通
T=1 A→B
T=0 A←B
OE=1, 不导通

四、最大模式

1、信号引脚

(1) QS_1, QS_0 (Instruction Queue Status) 指令队列状态信号, 输出。

信号 QS_1, QS_0 的组合用来指示CPU内的指令队列的当前状态, 以便外部对CPU内指令队列的动作跟踪。

(2) $\overline{S_2}, \overline{S_1}, \overline{S_0}$ (Bus Cycles Status) 总线周期状态信号输出。

状态信号的不同组合, 指出CPU在当前总线周期所进行的操作类型。

最

大模式中, 总线控制器8288利用这些状态信号进行组合, 产生访问存储器和I/O端口的控制信号。

(3) $RQ/GT_0, RQ/GT_1$ (Request/Grant) 总线请求信号输入 / 总线请求允许信号, 输出。

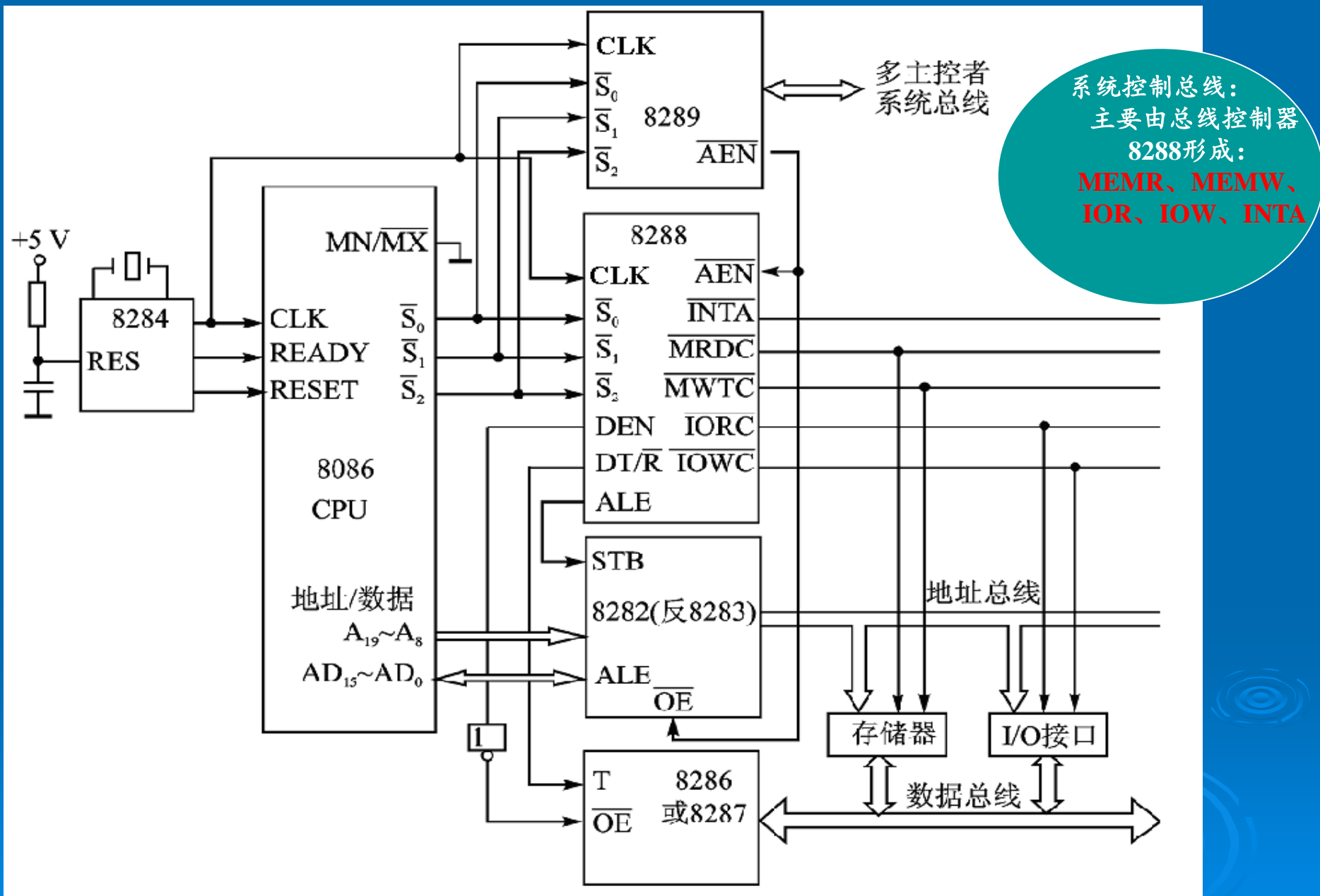
在最大模式系统中, 主CPU和其他协处理器间交换总线使用权的联络控制

信号。 GT_0 的优先级高于 GT_1 。

(4) \overline{LOCK} (lock) 总线封锁信号，输出。

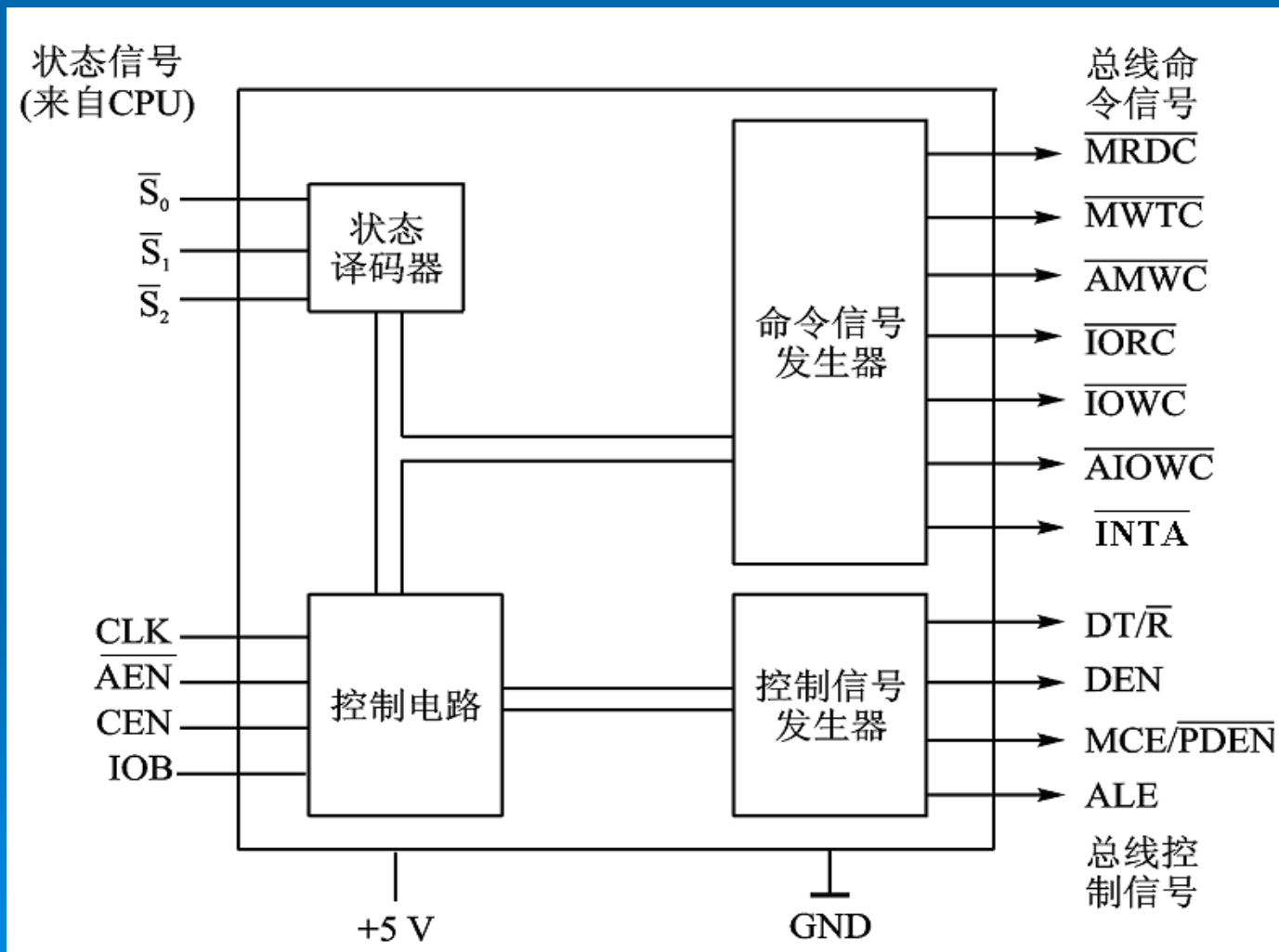
有效时，表明此时CPU不允许其他总线主模块占用总线。在DMA期间，被置为高阻抗状态。

2、系统配置



总线控制器8288芯片的引脚和内部结构。

来自8086 / 8088CPU的总线状态信号 $\overline{S_2}$, $\overline{S_1}$, $\overline{S_0}$ 经8288状态译码器译码后, 与输入控制信号 \overline{AEN} , CEN和IOB相互配合, 产生总线命令信号和总线控制信号。



2.3 寄存器结构

1、通用寄存器

通用寄存器包括：

数据寄存器、地址指针寄存器、变址寄存器。

数据寄存器包括： AX、BX、CX、DX。

地址指针寄存器包括：SP、BP。

变址寄存器包括： SI、DI。

2、段寄存器

段寄存器包括：CS、SS、DS、ES。

3、控制寄存器

控制寄存器包括：IP、PSW。

1、通用数据寄存器

AX、BX、CX、DX 作为通用寄存器。

用来暂存计算过程中所用到的操作数，结果或其它信息。

访问形式:可以用16位的访问；或者可以用字节（8位）形式访问。

高8位记作：AH、BH、CH、DH。

低8位记作：AL、BL、CL、DL。

AX——（Accumulator）累加器。

它是算术运算的主要寄存器。

所有I/O指令都使用这一寄存器与外部设备交换数据。

BX——Base用作基址寄存器使用。

在计算内存储器地址时，经常用来存放基址。

CX——Count可以作计数寄存器使用。

DX——Data可以作为数据寄存器使用。

一般在双字长乘除法运算时，把DX和AX组合在一起存放一个双字长(32位)数，DX用来存放高16位。

2、 地址指针与变址寄存器

SP、BP、SI、DI四个16位寄存器。

以字为单位在运算过程中存放操作数，经常用以在段内寻址时提供偏移地址。

段地址：只取段起始地址高16位值。

偏移地址：指在段内某内存单元物理地址相对段起始地址的偏移值。

SP: 为堆栈指针寄存器，指出当前堆栈段中栈顶的偏移地址，与SS联用。

BP: 为对堆栈操作的基址寄存器。BP中存放的是堆栈中某一存储单元的偏移地址，不是指栈顶。通常和SS联用。

SI: 为源变址寄存器，与数据段寄存器DS联合使用，确定数据段中某一存储单元的地址。

DI: 为目的变址寄存器，与数据段寄存器DS联合使用，确定数据段中某一存储单元的地址。

SI和DI具有自动增量和减量的功能。在串操作指令中，SI、DI隐含作为源变址和目的变址寄存器。对其他的指令，则没有这种限制。

3、段寄存器

段寄存器：4个16位段寄存器CS、DS、SS、ES。

用来识别当前可寻址的四个段，不可互换的使用。

CS——Code Segment Register 代码段寄存器

用来识别当前代码段（程序一般放在代码段）。

DS——Data Segment Register 数据段寄存器

用来识别当前数据段寄存器。

SS——Stack Segment Register 堆栈段寄存器，

用来识别当前堆栈段。

ES——Extra Segment Register 附加段寄存器，

用来识别当前附加段。

4、指令指针寄存器

用来存储代码段中的偏移地址；

程序运行过程中IP始终指向下一次要取出的指令偏移地址。
IP要与CS寄存器相配合才能形成真正的物理地址。

5、标志寄存器FR

由条件码标志FLAG、控制标志构成。

只用了其中9位, 6位条件码标志, 3位控制标志。

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
				溢出标志	方向标志	中断允许标志	单步标志	符号标志	零标志		辅助进位标志		奇偶校验标志		进位标志

- 状态标志位:

用来记录程序中运行结果的状态信息。

包括6位：CF、PF、AF、ZF、SF、OF。

CF 进位标志

CF=1: 从最高有效位产生进位值。

CF=0: 从最高有效位不产生进值。

PF 奇偶标志

PF=1: 结果操作数低8位中有偶数个1。

PF=0: 结果操作数低8位中有奇数个1。

AF 辅助进位标志

AF=1: 第3位（半个字节）产生进位值。

AF=0: 第3位（半个字节）不产生进位值。

ZF 零标志

ZF=1: 运算结果为0。

ZF=0: 运算结果不为0。

SF 符号标志

SF=1: 运算结果的符号为负。

SF=0: 运算结果的符号为正。

OF 溢出标志

OF=1: 在运算过程中，如操作数超过了机器表示的范围称为溢出。

OF=0: 在运算过程中，如操作数未超过了机器能表示的范围称为不溢出。

控制标志:

对控制标志位进行设置后，对其后的操作起控制作用。

IF 中断标志位

IF=1, 允许外部可屏蔽中断。

IF=0, 关闭中断。CPU禁止响应可屏蔽中断请求。

DF (Direction Flag) 方向标志，用来控制串操作指令标志。

DF=0, 地址指针为自动递增。

DF=1, 地址指针自动递减。

由STD指令可使DF置1，用CLD指令可使DF清0。

TF 单步标志。

TF=1时，CPU为单步工作方式，每执行完一条指令就自动产生一次内部中断。用在调试程序，可使用户逐条跟踪程序。

控制标志一旦设置，便对处理器的操作产生控制作用。

例1：执行两个数的加法，分析对标志位的影响。

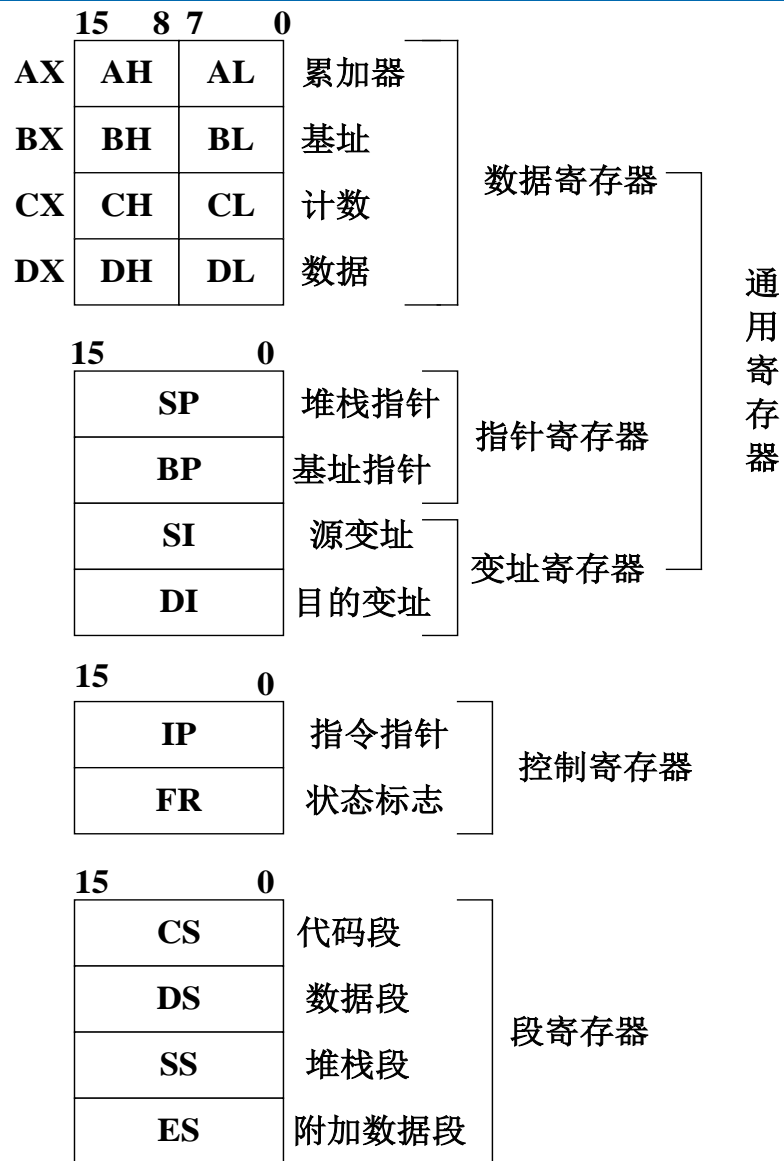
	0	0	1	0	0	0	1	1	0	1	0	0	0	1	0	1
+	0	0	1	1	0	0	1	0	0	0	0	1	1	0	0	1
<hr/>																
	0	1	0	1	0	1	0	1	0	1	0	1	1	1	1	0

标志：

\therefore SF=0; ZF=0 PF=0; CF=0 AF=0; OF=0。

例2：执行两个数的加法，分析对标志位的影响。

	0	1	0	1	0	1	0	0	0	1	1	1	0	0	1	
+	0	1	0	0	0	1	0	1	0	1	1	0	1	0	1	
<hr/>																
	1	0	0	1	1	0	0	1	1	0	1	0	0	1	1	
	OF=1												AF=1			



8086 CPU寄存器结构图

2.4 8086 存储器组织

- 1、存储单元的地址和内容
- 2、8086存储器的分体结构
- 3、存储器的分段
- 4、物理地址的形成

1、存储单元的地址和内容

存储单元地址：就是对存储单元的编号。

8086 / 8088系统中，存储器是按照字节编址的，即一个存储单元存放一个字节的內容。

存储单元内容：是指一个存储单元中的有效信息。

8086 / 8088 系统字长是16位的，由二个字节组成。所以当一个字存入存储器时需要占用相继的二个存储单元：低位字节存入低地址单元，高位字节存入高地址单元。字单元的地址采用它的低地址来表示。

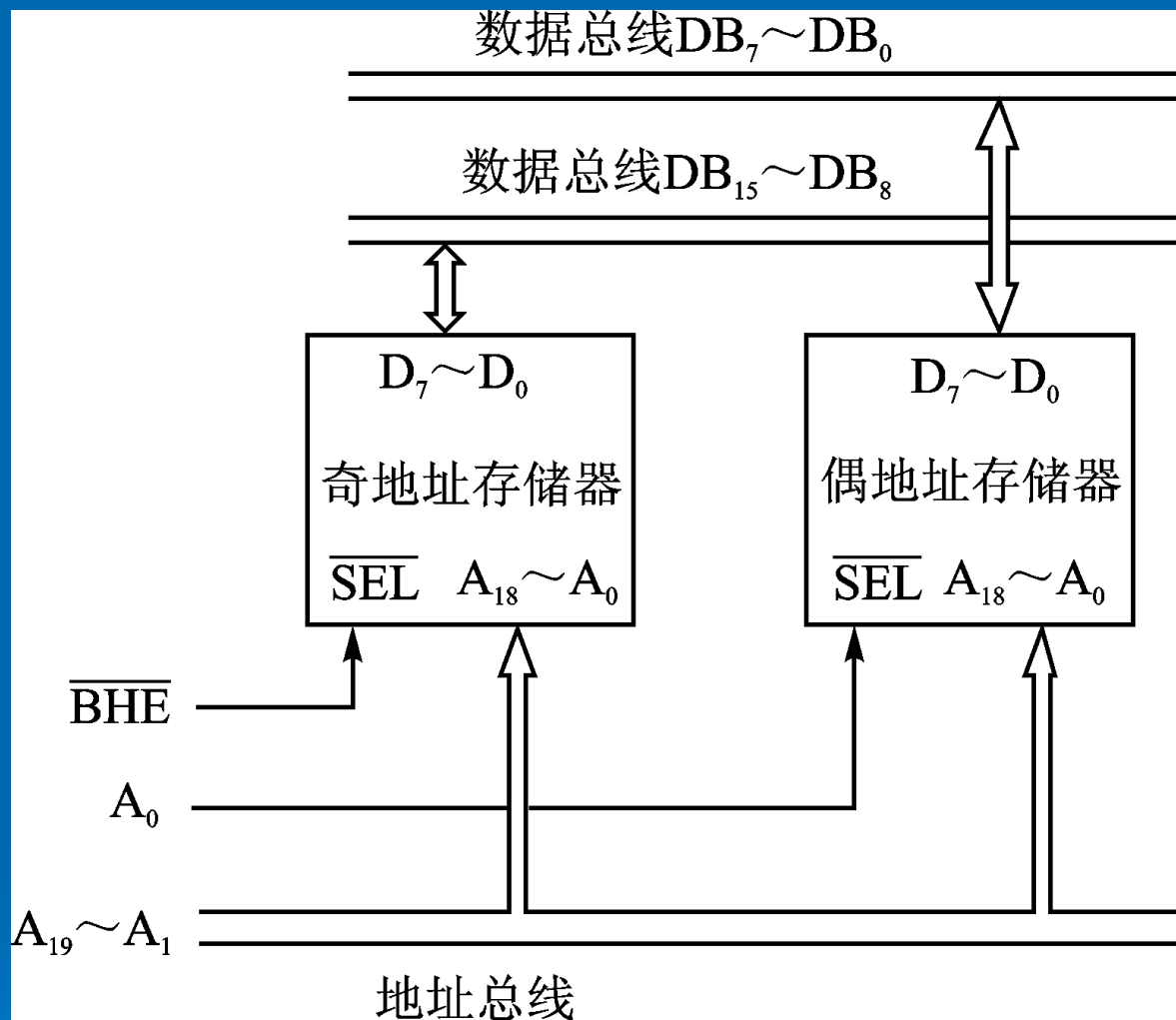
2、8086存储器的分体结构

在8086系统中，将其可寻址的1 MB存储器分为两个512 KB的存储体，即奇地址存储体和偶地址存储体，各为512 KB。

奇地址存储体与系统高8位数据总线相连，偶地址存储体与系统低8位数据总线相连。所以，访问存储器时，读 / 写偶地址体时，数据从低8位数据总线上传送。读 / 写奇地址体时，数据从高8位数据总线上传送。奇偶存储体的选择由 $\overline{\text{BHE}}$ 信号决定。



存储器分体结构单元示意图



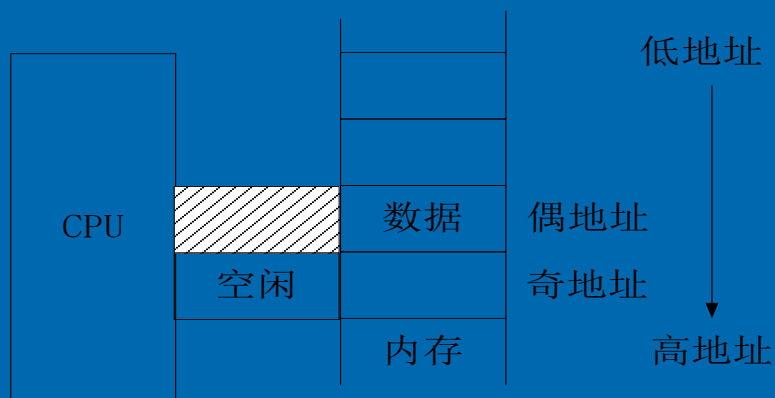
8086系统中存储器与总线的连接

内存中数据的存放规则是一个字节数据可以存放在奇地址体，也可以在偶地址体，字节地址就是存储单元的实际地址。一个字（16位）数据占连续的二个单元，高字节存放高地址，低字节存放低地址，并将低字节的地址作为该字的字地址。CPU访问存储器规则是：一次读/写一个字，并且均从偶地址开始。所以读/写字、字节就会有几种不同的情况。

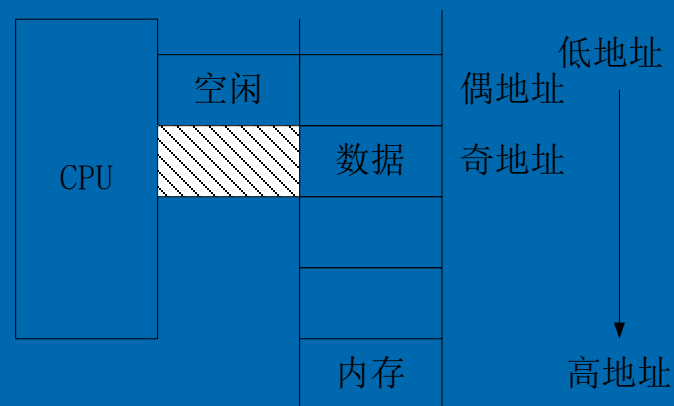
$\overline{\text{BHE}}$	A_0	操作		所用的数据总线
0	0	存取规则字（从偶地址开始读 / 写一个字）		$AD_{15} \sim AD_0$
1	0	从偶地址内存单元或I / O端口读 / 写一个字节		$AD_7 \sim AD_0$
0	1	从奇地址内存单元或I / O端口读 / 写一个字节		$AD_{15} \sim AD_8$
0 1	1 0	从奇地址开始读 / 写一个（非规则）字	第一总线周期高8位数据有效	$AD_{15} \sim AD_8$ $AD_7 \sim AD_0$
			第二总线周期低8位数据有效	

代码组合及对应的存取操作

读 / 写一个字节：只须访问某个存储体（奇体或偶体），相应的8位数据在数据总线上有效，而另一个字节的数据被忽略

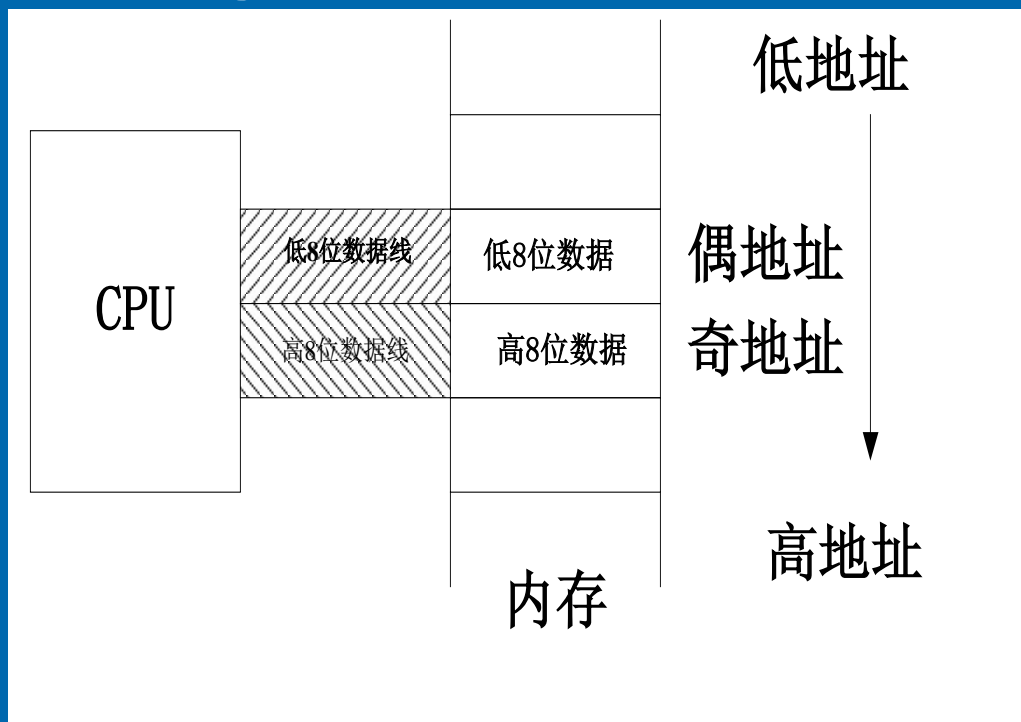


读/写偶地址字节

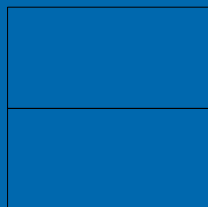


读/写奇地址字节

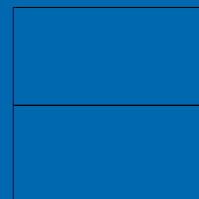
读/写一个字：若该字单元地址是从偶地址开始的，则只须执行一个总线读/写周期便可完成对该字的读/写操作；若该字地址从奇地址开始，则CPU需要执行连续的二个总线读/写周期才能完成对该字的读/写，第一次取奇地址体上数据，偶地址体上的8位数据被忽略，第二次取偶地址体上数据，奇地址体上8位数据被忽略。



读/写偶地址字



第一个总线周期



读/写奇地址字

第一个总线周期

3、存储器地址分段：

8086/8088有20条地址总线，直接寻址能力为 $2^{20}=1\text{M}$ 字节。
用16进制数表示1M字节的地址范围应为00000H~FFFFFH。

CPU中的寄存器是16位的，20位地址无法用16位寄存器表示，必须分段。

程序员在编制程序时把存储器划分成段；

段内地址16位,每个段的大小最大可达64KB。

20位物理地址形成

物理地址：

在1M字节存储器里，每个存储单元都有一个唯一的20位地址作为该存储单元的物理地址。

CPU访问存储器时，必须先确定所要访问的存储单元的物理地址才能取出（或存入）该单元中的内容。

20位物理地址形成：由16位段地址和16位偏移地址组成。

段地址：只取段起始地址高16位值。

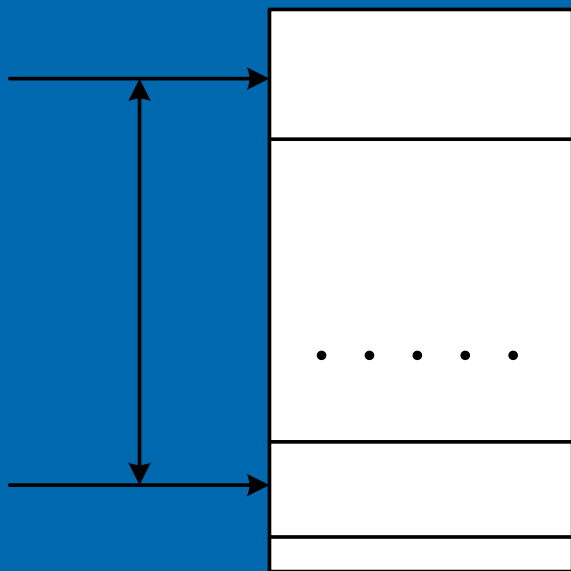
偏移地址：指在段内某内存单元物理地址相对段起始地址的偏移值。

物理地址计算方法:

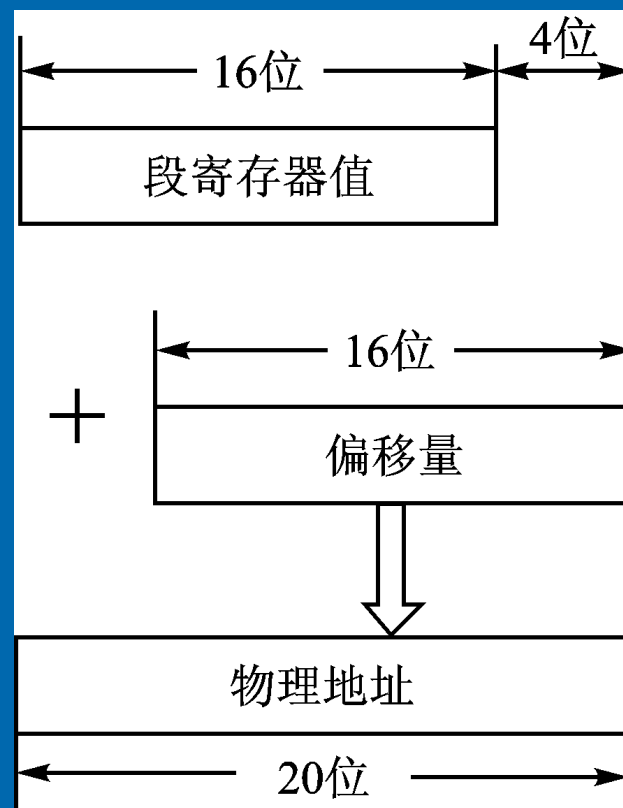
即把段地址左移4位再加上偏移地址值形成物理地址，写成：

$$\text{物理地址} = 16d \times \text{段地址} + \text{偏移地址}。$$

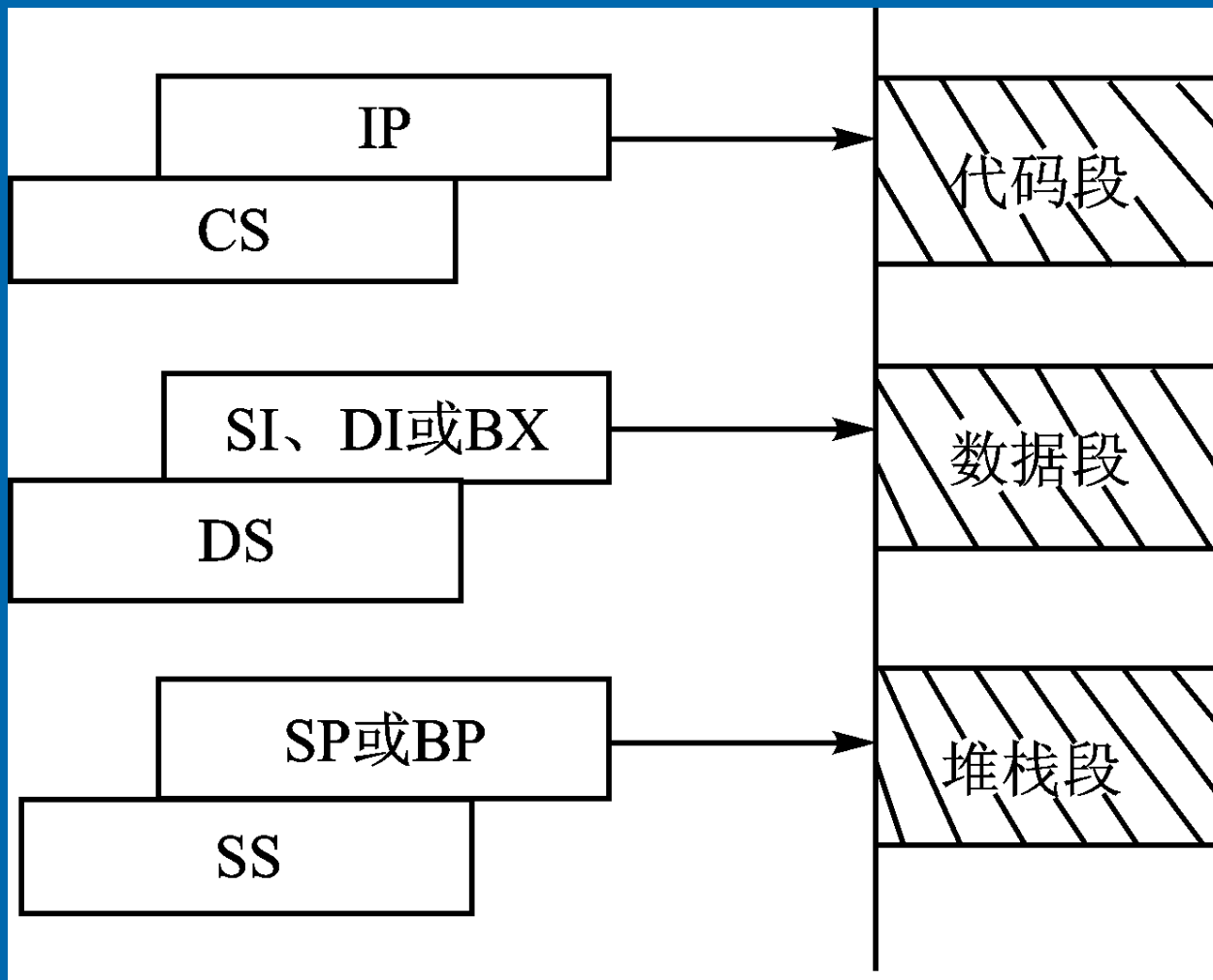
每个存储单元只有唯一的物理地址。
但可由不同的段地址和不同的偏移地址组成。



段地址与偏移地址关系示意图



存储器物理地址的计算方法



段寄存器和其他寄存器组合指向存储单元示意图

2-5 8086的I/O组织

8086系统和外部设备之间是通过I/O芯片连接的。每个I/O芯片都设置有一定数目的端口，一个端口通常对应芯片内部的一个寄存器或者一组寄存器。微型机系统为每个端口分配一个地址，叫**端口号**。各端口号不能重复。

8086可以访问64K个8位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。

CPU在执行访问I/O端的指令即输入（IN）指令和输出（OUT）指令时，硬件上会自动产生有效的 \overline{RD} 信号或者 \overline{WR} 信号，同时使 $\overline{M/IO}$ 信号处于低电平，通过外部逻辑电路的组合产生对I/O端口的读信号或者对I/O端口的写信号，以上是**I/O端口独立编址方式**的操作过程。

I/O端口和存储器也可以统一编址，用对存储器的访问指令来实现对I/O端口的读/写。存储器的读/写指令的寻址方式多，功能强，编制程序方便灵活。8086采用I/O端口独立的编址方式。

2-6 8086/8088的时序

一、时序的基本概念

时序。计算机中一条指令的执行，是通过将指令的功能分成若干个最基本的操作序列，顺序完成这些基本操作就实现了指令的功能。基本操作由具有命令性质的脉冲信号控制电路各部件完成的。命令信号的出现，必须有严格的时间先后顺序。**这种严格的时间上的先后顺序称为时序（Time Order）。**

时钟信号与T状态。确定时序必须有定时信号。微型计算机中的定时信号由时钟信号CLK产生。是CPU一切操作的计时标准和基本控制信号。通常时钟周期（Clock Cycle）是CPU计时的最小单位，称为一个T状态（T State）或T周期。时钟信号是各种命令脉冲信号和定时信号的脉冲源。时钟信号在时间上有先后次序，在空间上由不同的输出信号线输出。

指令周期。执行一条指令所需要的时间。8086中的指令的指令周期是**不等长的**。

总线周期。CPU通过系统总线对存储器或I/O端口进行一次存取操作所需的时间称为一个总线周期（Bus Cycle）。

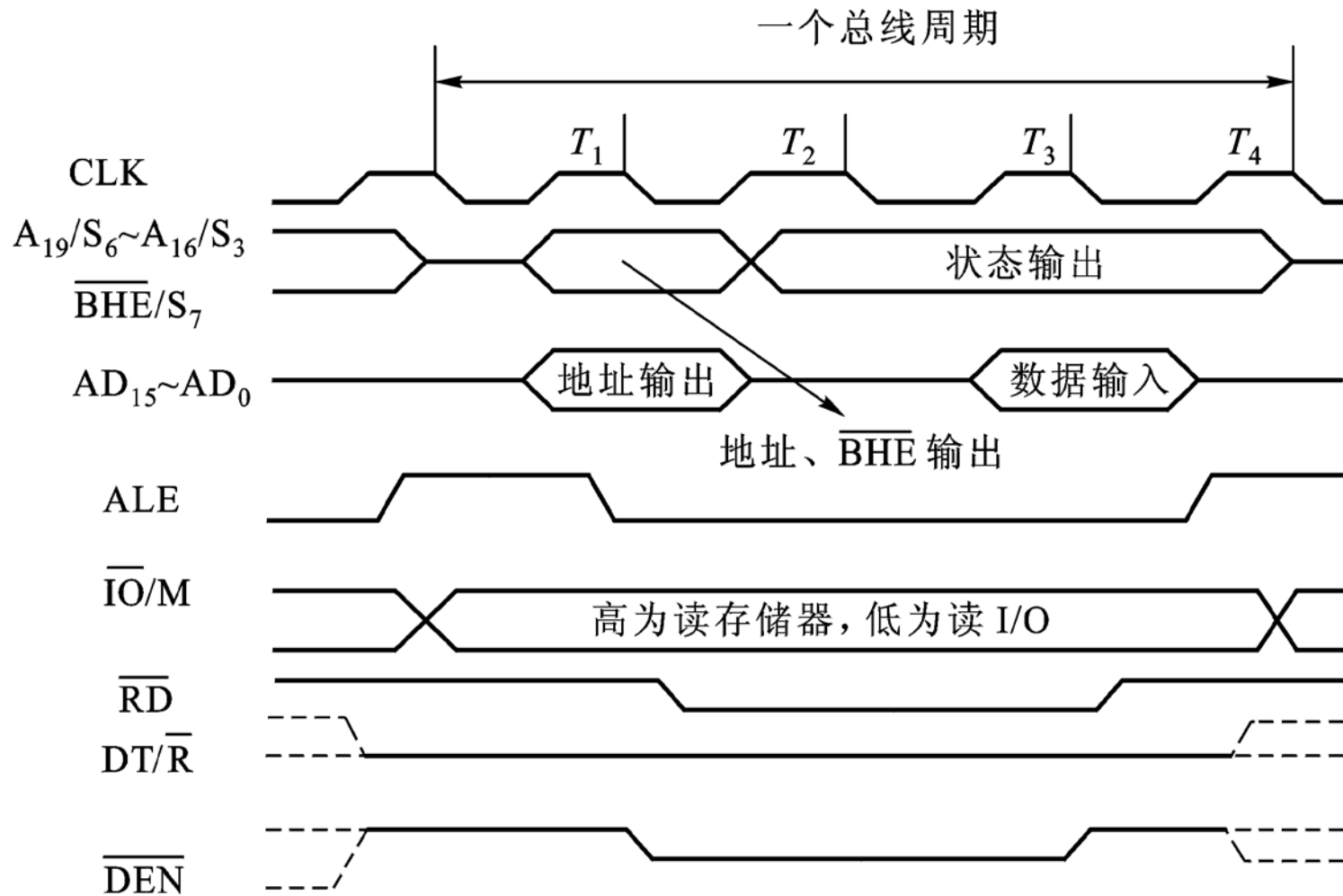
二、时序分析

分析步骤：首先搞清时序图对应的基本操作和操作过程，二是要熟悉时序图中出现的所有信号的含义。三是按照定时信号的先后顺序，抓住信号变化，按时间先后依次分析，确定各微操作的定时关系。

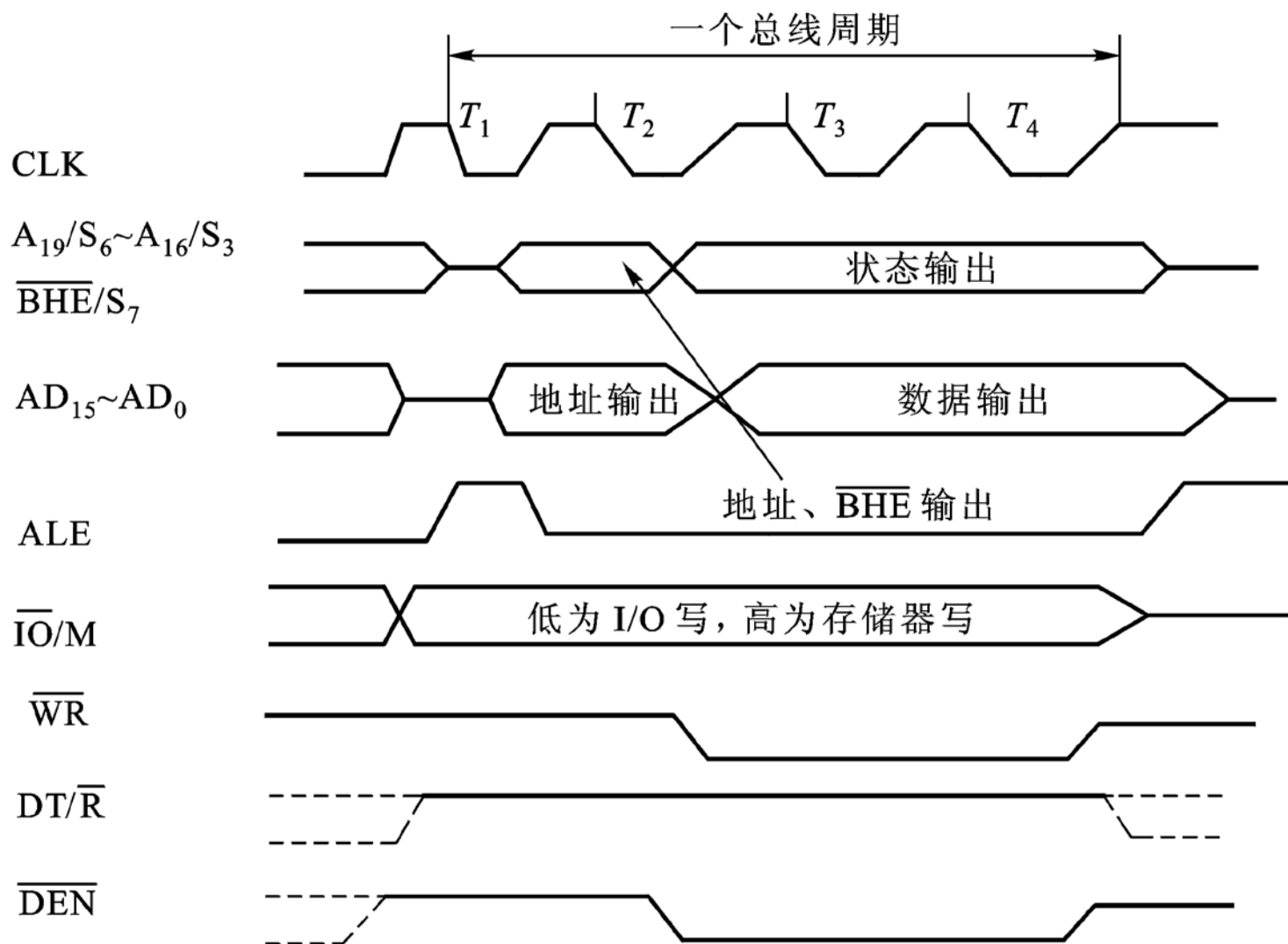
以8086为例讲述总线读操作和总线写操作的时关系。

1、最小模式下的总线读操作

按4个时钟周期分析。



2、最小模式下的总线写操作



最大模式中，其时序分析过程与最小模式的分析过程相同。



作业

P51 2-1

P52 2-21 2-23

