


# TBM AC/XT



## 集成电路手册



苏州电子计算机厂

---

IBM PC/XT

# 集成电路手册

苏州电子计算机厂

## 前 言

当前 IBM-PC/XT 及其兼容机、国产 0520 微型计算机，已经大量使用。但是有关这些计算机的硬件资料，针对性强的尚不多见。为了学习这类微型机的工作原理，我们收集了 IBM-PC/XT、长城 0520、苏州 0520 等微型机基本系统中差不多全部的通用集成电路，翻译编辑了这本资料。其中包括大规模集成电路的技术规范、中小规模集成电路的引脚安排和逻辑功能说明。为了便于其它 8086 系列微型计算机用户参考，也收集了 8086 系列中主要的大规模集成电路。

本书编译过程中，我们着眼于简明、全面、实用，力求使其成为 IBM-PC/XT 及其兼容机、0520 机，和其它 8086 系列微型计算机的一本较理想的学习、使用、维修和设计扩展部件的参考资料。

全书由本厂 0520 组集体编译整理，郑筑鸣同志对全部文稿作了审校，从而保证了本书技术术语和文字叙述上的统一。

由于我们水平有限，失误之处在所难免，望专家及同志们批评指正。

0520 组 1986.2

# 目 录

## I 大规模集成电路

- 一、8086 16 位 HMOS 微处理器 ..... ( 1 )
- 二、8088 8 位 HMOS 微处理器 ..... ( 33 )
- 三、8087 数值数据处理器 ..... ( 61 )
- 四、8089 8/16位HMOS I/O微处理器 ..... ( 84 )
- 五、8284A/8284A-1 8086 8088 微处理器的时钟发生器与驱动器 ..... (102)
- 六、8288 8086 8088 微处理器用的总线控制器 ..... (110)
- 七、8289/8289-1总线仲裁器 ..... (119)
- 八、8237A/8237A-4/8237A-5 高性能可编程直接存储器存取  
(DMA)控制器 ..... (132)
- 九、8259A/8259A-2/8259A-8可编程中断控制器 ..... (154)
- 十、8253/8253-5可编程定时器 ..... (179)
- 十一、8255A/8255A-5可编程外围接口 ..... (190)
- 十二、8272A单/双密度软磁盘控制器 ..... (214)
- 十三、8273/8273-4可编程 HDLC/SDLC规程控制器 ..... (246)
- 十四、INS8250-B异步通讯接口器件 ..... (277)
- 十五、MC6845 CRT控制器(CRTC) ..... (297)
- 十六、MM58167微处理器适用的实时钟 ..... (316)
- 十七、2114A 1024×4位静态RAM ..... (323)
- 十八、2118 16384×1位动态RAM ..... (327)
- 十九、2164 65536×1位动态RAM ..... (339)
- 二十、4416 16384×4位动态RAM ..... (349)
- 二十一、50256 262144×1位动态RAM ..... (364)
- 二十二、2716 16K(2K×8)紫外光可擦PROM ..... (373)
- 二十三、2732A 32K(4K×8)紫外光可擦 PROM ..... (380)
- 二十四、2764 64K(8K×8)紫外光可擦 PROM ..... (388)
- 二十五、27128 128K(16K×8)紫外光擦除的 PROM ..... (395)

## I 中小规模集成电路

— 8086

# 16 位 HMOS 微处理器

8086/8086-2/8086-1

- 直接寻址能力到 1 兆字节
- 体系结构设计成适合于强功能的汇编语言和高效率的高级语言
- 具有对称操作的 14 个 16 位寄存器组
- 24 种操作数寻址方式
- 位, 字节, 字和块操作能力
- 带符号和不带符号的、8 位和 16 位的二进制或十进制数的算术运算, 包括乘、除法,
- 时钟频率: 8086 为 5MHz, 8086-2 为 8 MHz, 8086-1 为 10MHz,
- 多总线系统兼容的接口

Intel 8086 高性能 16 位 CPU 有三种时钟速率: 5M、8M、和 10MHz。该 CPU 采用 N 沟道耗尽型硅栅工艺 (HMOS), 用 40 条引线双列直插式封装。8086 工作于两种配置: 单处理器和多处理器, 性能优良。

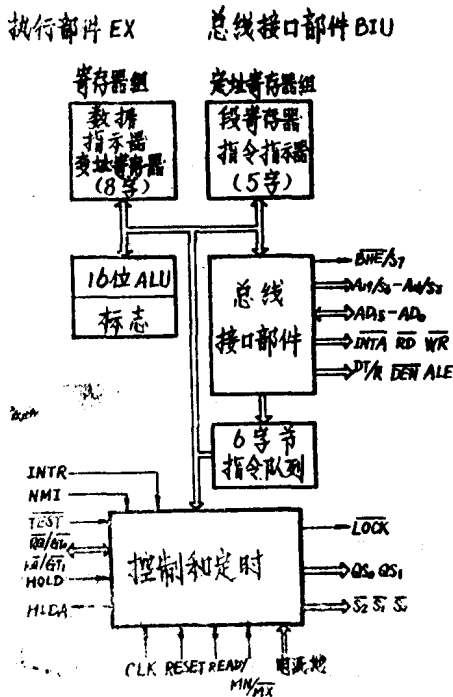


图 1 8086 CPU 方框图

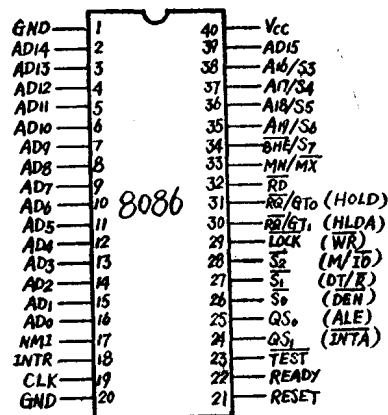


图 2 8086 引脚图

## 引 脚 说 明

下面的引脚功能说明，既适用于8086系统的最小模式，又适用于最大模式，说明中的“局部总线”是指直接到8086的多路复用总线(没有考虑附加总线缓冲器)。

### 地址数据总线 $AD_{15}-AD_0$ (输入/输出，三态；引脚 2-16, 39)

这些引线构成定时多路复用存储器/IO地址总线 ( $T_1$ ) 和数据总线 ( $T_2, T_3, T_4, T_5$ )。 $A_0$ 与BHE功能相似，它用来选通数据总线低位字节引线端  $D_7-D_0$ 。在存储器或I/O操作中，当一个字节在低8位总线上传送时， $T_1$ 期间 $A_0$ 为低电平，一个联接到低8位总线的8位设备，通常使用 $A_0$ 作为片选条件。这些引线是高电平有效，在中断响应和局部总线“保持响应”期间，浮动于高阻态。

### 地址/状态线 $A_{19}/S_6, A_{18}/S_5, A_{17}/S_4, A_{16}/S_3$ (输出，三态；引脚35-38)

对于存储器操作来说， $T_1$ 期间是有效的高4位地址线。在I/O操作期间，这些线是低电平。对存储器和I/O操作，在 $T_2, T_3, T_4$ 和 $T_5$ 期间，这些线上具有可以利用的状态信息。中断允许标志位( $S_5$ )的状态在每个时钟周期开始时被修改。 $A_{17}/S_4$ 和 $A_{16}/S_3$ 编码如下：

A17/S4	A16/S3	功 能
0(低)	0	替换数据
0	1	堆 栈
1(高)	0	代码或无内容
1	1	数 据

$S_6$  是 0(低)

这些信息指明，哪一个寄存器正在被用于数据存取。

局部总线“保持响应”期间，这些线浮动于高阻态。

### 总线高位允许/状态线 $\overline{BHE}/S_7$ (输出，三态；引脚34)

$T_1$ 期间，总线高位允许信号 ( $\overline{BHE}$ )用来选通高8位数据总线(引线 $D_{15}-D_8$ )的数据。联到高8位数据总线上的8位设备通常用 $\overline{BHE}$ 作为片选条件。当在总线高8位传送一个字节时，对于读、写和中断响应，周期的 $T_1$ 期间 $\overline{BHE}$ 是低电平。在 $T_2, T_3$ 和 $T_4$ 期间， $S_7$ 状态信息有效。这一信号是低电平有效，在“保持”时浮动成高阻态。对于第一个中断响应，周期的 $T_1$ 期间 $\overline{BHE}$ 是低电平。

### 读信号 $\overline{RD}$ (输出, 三态; 引脚 32)

读选通脉冲指明, 根据S2引线端的状态, 处理器正在执行一个存储器读周期或是 I/O 读周期。这一信号用来读8086局部总线上的设备。任一读周期的 $T_2$ ,  $T_3$ 和 $T_w$ 期间,  $\overline{RD}$ 是有效的低电平。在8086局部总线浮动前, 应保证使它在 $T_2$ 里保持高电平。

在“保持响应”时, 该信号浮动成高阻态。

### 准备好 READY(输入; 引脚 22)

READY是来自被寻址存储器或I/O设备的回答信号, 表示它已经为数据传送做好准备。来自存储器/I/O的准备信号, 由8284时钟信号发生器同步而形成READY。该信号为高电平有效。8086的READY输入是非同步的。如果建立时间和保持时间不能满足, 就不能保证操作正确。

### 中断请求 INTR(输入; 引脚 18)

中断请求是一个电平触发的输入, 在每条指令的最后时钟周期内被采样, 以决定处理器是否要执行一个中断响应操作。中断服务子程序通过设置在系统存储器中的中断向量检索表来引导。用软件复位中断允许位, 可在内部屏蔽它。INTR在内部同步。该信号为高电平有效。

### 测试TEST(输入; 引脚23)

用于“Wait”指令的外部输入信号。 $\overline{TEST}$ 是低电平时程序继续执行, 否则处理器处于“空闲”状态中等待。这个信号在每个时钟周期的CLK前沿被内同步。

### 非屏蔽中断NMI(输入; 引脚17)

非屏蔽中断是一种边沿触发中断。它产生类型2中断, 经过系统存储器中设置的中断向量检索表, 引导一段服务程序。NMI不能由软件从内部进行屏蔽。在现行指令结束时的上升沿启动中断。NMI输入在内部被同步。

### 复位RESET(输入; 引脚21)

RESET使处理器立即结束它的当前活动。这个信号必须至少维持四个时钟周期的有效(高电平)时间。当RESET返回低电平时, 如同指令系统部分中所述, 它重新启动程序执行。RESET是内部同步的。

### 时钟CLK(输入; 引脚19)

时钟为处理器和总线控制器提供基本的定时。为了提供最佳的内部定时, 它是一个具有33%占空度的非对称矩形波。

### 电源 $V_{CC}$ (引脚40)

$V_{CC}$ 是+5V电源引线端。

### 地 GND(引脚1, 20)

GND是接地端。

### 最小模式/最大模式 $\overline{MN}/\overline{MX}$ (输入; 引脚33)

表示处理器工作在何种模式。两种工作模式在下面有关章节中说明。

下面的引脚功能说明用于最大模式的8086/8288系统(即 $\overline{MN}/\overline{MX} = V_{SS}$ )。这里仅对最大模式专用的引脚功能加以说明。所有其它引脚功能在上面已经说明了。

### 状态信号 $S_2$ , $S_1$ , $S_0$ (输出, 三态; 引脚26-28)

这些状态编码如下:

$\overline{S_2}$	$\overline{S_1}$	$\overline{S_0}$	功 能
0(低)	0	0	中断响应
0	0	1	读I/O口
0	1	0	写I/O口
0	1	1	暂 停
1(高)	0	0	代码存取
1	0	1	读存储器
1	1	0	写存储器
1	1	1	通过状态

在 $T_4$ 、 $T_1$ 和 $T_2$ 期间，这些状态信息有效，且当READY是高电平时，在 $T_3$ 或 $T_4$ 期间返回到通过状态(1,1,1)。8288总线控制器用这些状态产生所有的访问存储器和I/O的控制信号。在 $T_4$ 期间， $\overline{S_2}$ 、 $\overline{S_1}$ 或 $\overline{S_0}$ 的任何变化都用以指示一个总线周期的开始；而在 $T_3$ 或 $T_4$ 时返回到通过状态，以指出一个总线周期结束。

在“保持响应”时，这些信号浮动为高阻态。

**请求/应答  $\overline{RQ}/\overline{GT_0}$ 、 $\overline{RQ}/\overline{GT_1}$ (输入/输出，引脚30,31)**

两条请求/应答引线，可以被另外的局部总线主设备用来强使处理器在它的现行总线周期结束时释放局部总线。每条引线都是双向的，而 $\overline{RQ}/\overline{GT_0}$ 比 $\overline{RQ}/\overline{GT_1}$ 具有更高的优先权。 $\overline{RQ}/\overline{GT}$ 有一个内部正偏电阻，因此这些引线允许悬空。请求/应答时序如下(见波形图)：

1. 从另一个局部总线主设备发来一个CLK宽度的脉冲(脉冲1)，表示向8086提出一个局部总线请求(“保持”)。

2. 在 $T_4$ 或 $T_1$ 期间，从8086向请求的主设备发出一个CLK宽度的脉冲(脉冲2)，表明8086已允许局部总线浮动，和在下一个CLK周期它将进入“保持响应”状态。在“保持响应”期间，CPU的总线接口在逻辑上从局部总线脱开。

3. 请求的主设备发来一个CLK宽度脉冲(脉冲3)通知8086：“保持”请求将要结束，在下一个CLK时8086可以收回局部总线。

局部总线的每次主设备转换，都有按序的3个脉冲。每次总线转换之后，必须有一个停滞CLK周期。这些脉冲是低电平有效。

如果请求是在CPU执行存储器周期时出现，当下列条件全部满足时，在周期的 $T_4$ 中CPU释放局部总线：

1. 请求在 $T_2$ 或 $T_2$ 之前出现。
2. 现行周期不是字的低字节(在奇地址上)。
3. 现行周期不是中断响应过程的第一个响应周期。
4. 封锁指令当前没有执行。



如果请求出现时局部总线为空闲状态, 则有两种可能:

1. 局部总线在下一个时钟释放。
2. 在3个时钟周期里开始一个存储器周期, 则当前为存储器周期的四条规则使用到总是满足的条件1。

#### 封锁LOCK(输出, 三态, 引脚29)

LOCK (封锁)输出表明, 在 $\overline{\text{LOCK}}$  是有效的低电平时, 其它系统总线主设备不能获得系统总线的控制权。LOCK 信号是由“LOCK”前缀指令启动的, 到下一条指令完成之前保持有效。该信号为低电平有效, 在“保持响应”时浮动为高阻态。

#### 队列状态 $QS_1, QS_0$ (输出, 引脚 24,25)

在队列操作执行后的CLK周期内队列状态有效。

$QS_1$ 和  $QS_0$  提供了对 8086 内部指令队列状态进行外部跟踪的状态信息。

$QS_1$	$QS_0$	功 能
0(低)	0	无操作
0	1	从队列送出操作码的第一字节
1(高)	0	队列空
1	1	从队列送出后面的字节

下面的引脚功能说明适用于8086最小模式(即 $\overline{\text{MN}}/\overline{\text{MX}} = V_{CC}$ )。这里仅对最小模式专用引脚功能加以说明; 其它的引脚功能如前所述。

#### 状态线 $\overline{\text{M}}/\overline{\text{IO}}$ (输出, 三态, 引脚28)

这一状态在逻辑上与最大模式的 $S_2$ 是等同的。用它来区别是一个存储器存取还是 I/O 存取。在前一个总线周期的 $T_4$ ,  $\overline{\text{M}}/\overline{\text{IO}}$ 成为有效( $\overline{\text{M}} =$ 高电平,  $\overline{\text{IO}} =$ 低电平), 并一直保持到本总线周期的 $T_4$ 为止。在局总部线“保持响应”时,  $\overline{\text{M}}/\overline{\text{IO}}$ 浮动为高阻态。

#### 写信号 $\overline{\text{WR}}$ (输出, 三态, 引脚29)

写选通脉冲, 根据 $\overline{\text{M}}/\overline{\text{IO}}$ 信号的状态, 表示处理器正在执行存储器写 或是 I/O 写周期。任何写周期的 $T_2, T_3$ 和 $T_w$ ,  $\overline{\text{WR}}$  为有效。它是低电平有效, 在局部总线“保持响应”时浮动为高阻态。

#### 中断响应 $\overline{\text{INTA}}$ (输出, 引脚24)

在中断响应周期,  $\overline{\text{INTA}}$  用作一个读选通脉冲。每个中断响应周期的 $T_2, T_3$ 和 $T_w$ 期间, 它是低电平有效。

#### 地址锁存允许ALE(输出, 引脚25)

由处理器提供的地址锁存允许信号, 把地址锁存到 8282/8283 地址锁存器中。任何总线周期的 $T_1$ 期间, 它是一个有效的高电平脉冲。注意, ALE 不会浮动为高阻抗。

#### 数据发送/接收 $\overline{\text{DT}}/\overline{\text{R}}$ (输出, 三态, 引脚 27)

在采用8286/8287数据总线收发器的最小模式系统中,需要数据发送/接收信号来控制通过收发器的数据方向。在逻辑上,DT/R与最大模式中的S<sub>1</sub>是等同的,而定时与M/IO相同(T=高电平,R=低电平)。在局部总线“保持响应”时,这个信号浮动为高阻态。数据允许DEN(输出,三态;引脚26)

在用收发器的最小模式系统中,处理器提供的数据允许信号作为8286/8287的输出允许。在每次存储器与I/O存取期间以及INTA周期里,DEN是有效的低电平。对于读或INTA周期,从T<sub>2</sub>的中间时刻到T<sub>4</sub>的中间时刻,它是有效的;而对于一个写周期,从T<sub>2</sub>开始直到T<sub>4</sub>的中间时刻它是有效的。在局部总线“保持响应”时,DEN浮动为高阻态。

**保持HOLD(输入;引脚31), HLDA(输出;引脚30)**

HOLD表示另一个主设备正在请求局部总线“保持”。为了得到响应,HOLD必须有效(高电平)。在T<sub>1</sub>或T<sub>4</sub>中,接收“保持”请求的处理器,发出HLDA(高电平)作为一个响应信号。与此同时处理器就浮动其局部总线和控制线为高阻态。处理器检测HOLD为低电平之后,就把HLDA拉至低电平,和当处理器需要运行另一个周期时,它再驱动局部总线和控制线。

当局部总线释放时,依据对RQ/GT应用所考虑的同样规则。

HOLD不是异步输入信号。如果系统不能用别的方法保证建立时间,就应该进行外部同步。

## 功能说明

### 一般操作

8086处理器的内部功能在逻辑上分为两个处理部件。如图1方框图所示,第一个是总线接口部件(BIU),第二个是执行部件(EU)。

这两个部件可以直接相互作用,但大多数情况是各自作为异步操作处理器工作。总线接口部件提供有关取指令和排队、操作数读取及存储、地址定位等功能。由这个部件提供的指令予取重迭操作,通过改进总线带宽,增强了处理器的性能。在等待指令译码和执行时,指令流最多可排队6个字节。

指令流的排队机构(队列)使总线接口部件非常有效地使用存储器。无论何时,只要在队列中至少有两个字节的空间,总线接口部件就试图执行读取一个字的存储器周期。这样就大大缩短了存储器总线上的“停滞时间”。这个队列可以起到一个先进先出(FIFO)缓冲器的作用,执行部件从中提取所需要的指令字节。如果队列是空的(例如,在一个转移指令之后)进入队列的第一个字节,立即由执行部件利用。

执行部件从总线接口部件队列中接受予取的指令,并给总线接口部件提供未定位的操作数地址。存储器中的操作数,通过总线接口部件送至执行部件处理,并将结果送到总线接口部件以便存储。详见指令系统寄存器组和体系结构说明。

### 存储器组织

处理器向存储器提供一个20位的地址,用来定位被访问的字节。存储器组织成一兆字节的线性阵列,编址为00000(H)到FFFFFF(H)。存储器在逻辑上分成代码段、数据段、附加数据段和堆栈段,每段最多可达64K字节,且每段落在16字节的边界上(见图3a)。

全部存储器访问都是相对于基地址进行的,这个基地址包含在高速段寄存器里。按照

程序的寻址需要选择段的类型。段寄存器的选择是按下表规则自动进行的。在一种类型的段内全部信息具有同样的逻辑属性(即为代码或数据)。用建立相同特性的存储器区域结构和自动选择段寄存器的方法,程序可写得较短,执行较快,其结构更好。

字(16位)操作数可放置于偶数或奇数地址边界上,而不必像许多16位计算机中那样一定要放置于偶数地址边界。对于地址和数据操作数,字的低字节存放于较低的地址单

存储器引用	所用段寄存器	段选择规则
指令	代码段CS	所有取指令操作自动使用代码段。
堆栈	堆栈段SS	所有堆栈压入与弹出。除了数据引用之外的相对于基址寄存器BP的存储器引用。
局部数据	数据段DS	数据引用。出现在:相对于堆栈、串操作目标或明确的段代用等时候。
外部(全局)数据	附加段ES	串操作目标、明确选择的段代用。

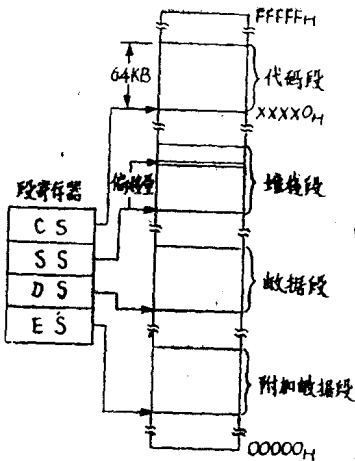


图 3a 存储器组织

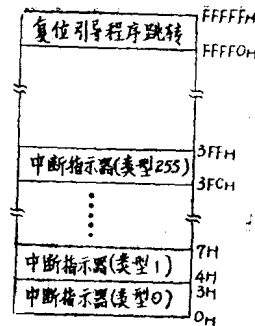


图3b保留的存储器单元

元,而高字节则存放于邻接的较高地址单元。总线接口部件自动地执行适当次数的存储器访问,如果字操作数在偶数字节边界上,则存取一次;如果它在奇数字节边界上,则存取二次。除了多一次存取的性能代价之外,这种双次存取对软件来说是透明的。取指令操作时不会出现这种性能代价,而只有取字操作数才会出现。

实际上,存储器由处理器的地址线 $A_{19}-A_1$ 并行寻址的两个512K的8位字节存储体,一个高位存储体( $D_{15}-D_8$ ),和一个低位存储体( $D_7-D_0$ )组成。具有偶数地址的字节数

据在总线的  $D_7-D_0$  上传送, 而奇数地址的字节数据 ( $A_0 =$  高电平) 在总线的  $D_{15}-D_8$  上传送。处理器提供两个允许信号  $\overline{BHE}$  和  $A_0$ , 用来有选择地从奇数字节地址或偶数字节地址, 或两者同时读出或写入。指令流是按字从存储器读取的, 由处理器内部寻址需要的字节。

在引用字数据时, 根据字的起始字节是偶数地址还是奇数地址, 总线接口部件分别需要一个或两个存储周期。因此, 在引用字操作数时, 通过把数据定位在偶数地址边界上, 可以使操作性能最佳。对于使用堆栈来说, 这是一种特别有用的技巧, 因为对于中断处理或多任务操作来说, 对堆栈的奇数地址引用可能给上下转换时间带来不利影响。

为了进行特定的CPU操作, 要保留一些存储器单元(见图3b)。从  $FFFF0(H)$  到  $FFFFF(H)$  单元是一个保留区, 该区包括一个转移到初始化程序的装载例行程序。在RESET后, CPU一开始总是执行  $FFFF0H$  单元内的转移指令。从  $00000H \sim 003FFH$  单元为中断操作保留。256种可能的中断类型的每一种, 都有一个指向服务程序的4字节指示器向量, 它由一个16位的段地址和一个16位的偏移地址组成。在中断出现以前, 指示器向量已经预先存放到存储器保留区中的相应位置上。

### 最大与最小模式:

为了支持最小的和最大的8086系统, 对CPU的要求显然是不同的, 这不能用唯一定义的40个引脚来实现。因此, 8086设置了一个搭接引线 ( $MN/\overline{MX}$ ), 由它来确定系统的配置。有关引线功能的定义取决于搭线端的状态。当  $MN/\overline{MX}$  引线接地(GND), 8086以最大模式处理24到31引脚。8288总线控制器对  $\overline{S_0}$ 、 $\overline{S_1}$ 、 $\overline{S_2}$  的状态信息译码以产生与多总线体系结构兼容的总线定时和控制信号。当  $MN/\overline{MX}$  引线接  $V_{cc}$  时, 8086本身在引脚24到31上产生总线控制信号, 如图2括号中所示的那样。在图4中示出了最小模式系统和最大模式系统的例子。

### 总线操作

8086有一组通常称为分时多路复用的复合地址和数据总线。这一技巧将最有效地使用处理器上的引线, 允许使用标准的40引脚的封装。在存储器和I/O模块提供地址锁存情的况下, “局部总线”可以直接缓冲, 和用在整个系统中。另外, 如果系统要求一种标准的非多路复用总线, 只要在处理器上用一组地址锁存器就能使总线不多路复用。

处理器每一个总线周期至少由四个CLK周期组成。称为  $T_1$ 、 $T_2$ 、 $T_3$  和  $T_4$  (见图5)。在  $T_1$  期间, 处理器发出地址。在  $T_3$  和  $T_4$  期间, 在总线上进行传递数据,  $T_2$  主要用于在读操作期间改变总线的方向。倘若由被寻址设备给出一个“未准备好”信号时, 则在  $T_3$  和  $T_4$  之间插入“等待”状态 ( $T_w$ )。每个插入的“等待”状态持续时间为一个时钟周期。在8086总线周期之间可能出现另一些周期, 这些周期称为“空闲”状态 ( $T_i$ )。处理器把这些周期用于内务处理。

在任何总线周期的  $T_1$  期间, 都发出(地址锁存允许)ALE信号(根据  $MN/\overline{MX}$  的搭接情况, 由处理器或由8288总线控制器发出)。在这个脉冲的后沿, 有效地址和周期的某些状态信息就被锁存。

在最大模式中, 状态位  $\overline{S_0}$ 、 $\overline{S_1}$  和  $\overline{S_2}$ , 由总线控制器按下表来识别总线处理的类型:

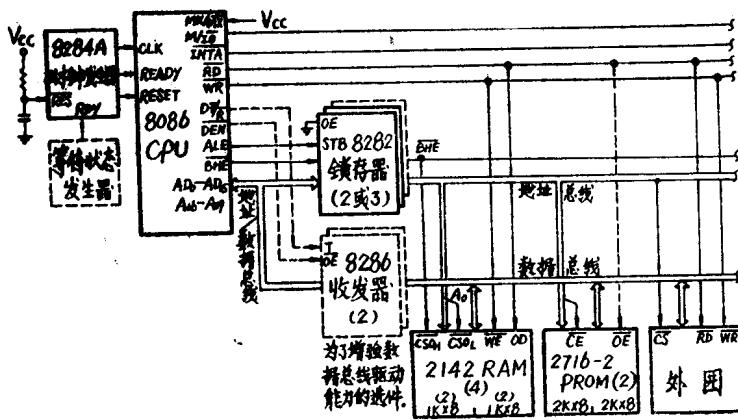


图 4a 8086最小模式典型系统结构

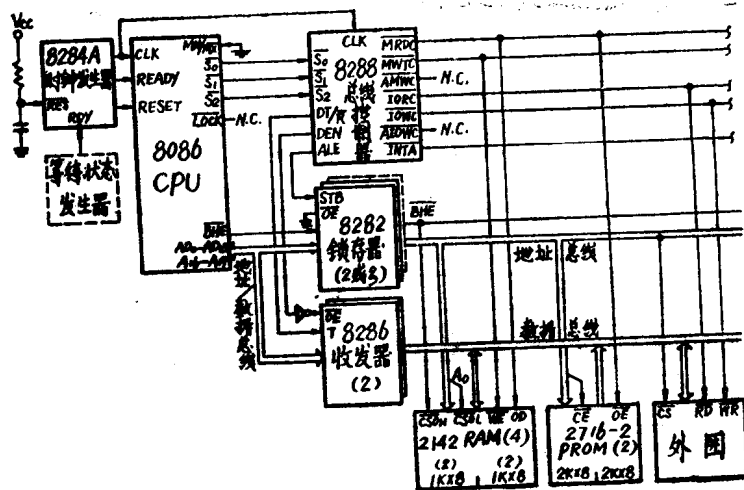


图 4b 8086最大模式典型系统结构

$\overline{S}_2$	$\overline{S}_1$	$\overline{S}_0$	功 能
0	0	0	中断响应
0	0	1	读I/O
0	1	0	写I/O
0	1	1	暂停
1	0	0	取指令
1	0	1	从存储器读数据
1	1	0	向存储器写数据
1	1	1	通过状态(无总线周期)

状态 $S_3$ 至 $S_7$ 与高位地址和BHE信号多路复用, 这些状态位信息在 $T_2$ 到 $T_4$ 期间是有效的。 $S_5$ 和 $S_4$ 指示在该总线周期内用哪一个段寄存器来形成地址(见指令系统说明):

$S_4$	$S_3$	段寄存器
0	0	附加数据(附加段)
0	1	堆 栈
1	0	代码或无效
1	1	数 据

$S_5$ 是PSW中断允许位的反映。 $S_6 = 0$ ,  $S_7$ 为一个备用状态位。

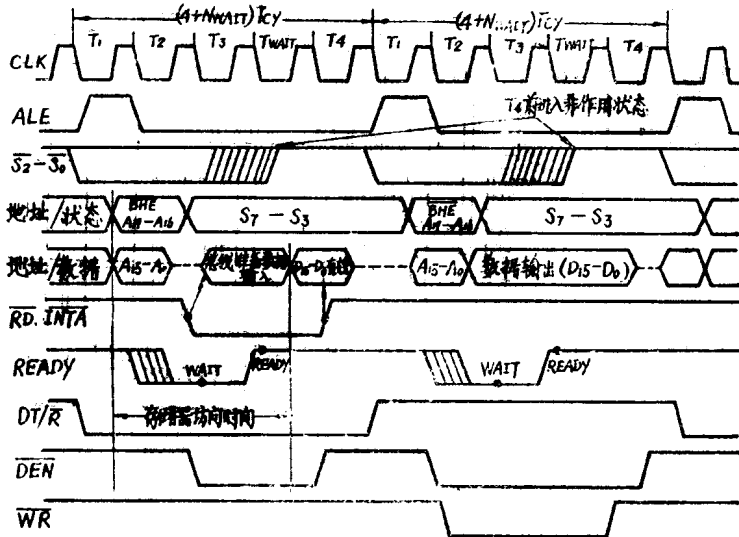


图 5 基本系统定时

### 输入/输出寻址

在8086中I/O操作最大寻址能力为64K I/O字节寄存器或32KI/O字寄存器。I/O地址以与存储器地址相同的形式出现在总线的 $A_{15}-A_0$ 线上。在I/O操作中, 地址线 $A_{19}-A_{16}$ 为零。用寄存器DX作为指示器的I/O指令具有全寻址能力, 而直接I/O指令对在I/O指令空间第零页中的256个I/O字节中的一个或两个单元直接寻址。

I/O端口以与存储器单元同样的方式寻址。偶数编址字节在总线的 $D_7-D_0$ 上传送, 奇数编址字节在 $D_{15}-D_8$ 上传送。必须注意, 应保证置于总线低八位上的一个8位外围设备中的每个寄存器都按偶数编址。

## 外部接口

### 处理器复位和初始化

处理器的初始化或启动由 RESET 引线的激活(高电平)来实现。8086 RESET 高电平的持续时间应大于四个时钟周期。8086 将在 RESET 的上升沿时刻终止操作,并且只要 RESET 为高电平,它就保持在休止状态。RESET 的下降沿触发一个大约10时钟周期的内部复位时序。在这个时序之后,8086 执行绝对地址 FFFF0H 单元中的指令,开始正常操作(见图3b)。这一操作的细节见 MCS-86 用户手册指令系统的说明。RESET 输入在其内部与处理器的时钟同步。在初始化时,RESET 下降沿的出现必须不早于加电后 50 $\mu$ s,以保证完成 8086 的初始化过程。在 RESET 结束后的第二个时钟周期之前,不允许出现不可屏蔽的中断(NMI)。

### 中断操作

中断操作分为两类:软件启动的和硬件启动的。软件启动的中断和硬件中断的软件化见指令系统说明。硬件中断又可分为不可屏蔽的中断(NMI)和可屏蔽的中断(INTR)。

中断导致控制转移到一个新的程序。包含指向中断服务程序地址指示器的 256 类中断向量表放在绝对地址 0 到 3FFH(见图3b),这些单元是为此用途而保留的。表中每个向量四个字节,对应着一种中断“类型”。在中断响应时序期间,中断设备提供 8 位“中断类型”码,用相应的向量指向新的中断服务程序入口地址。

### 不可屏蔽中断(NMI)

处理器提供一个不可屏蔽中断引线(NMI),它比可屏蔽中断请求引线(INTR)有更高的优先权。它的典型应用是启动电源故障服务程序。NMI 是由上升沿触发的。启动这一引线端产生类型 2 中断(见指令系统说明)。

NMI 高电平维持时间要大于两个时钟周期,但不要求它与时钟同步。NMI 的任何一个上升沿都在芯片上锁存,并在现行指令结束或在数据块传送指令中间进行中断服务。响应 NMI 最慢的情况发生在乘法、除法和变量移位指令执行时。对 NMI 下降沿的出现条件没有规定,它可以发生在 NMI 中断服务之前,之中或之后。如果在 NMI 中断服务过程开始之后又出现了一个上升沿,它将触发另一次响应。因此,通常要求 NMI 信号没有尖峰,也没有下降沿的反冲,以避免触发多余的响应。

### 可屏蔽中断(INTR)

8086 提供一个中断请求输入(INTR),通过软件将中断允许标志状态位复位可以对它进行内部屏蔽。这一中断请求信号是由电平触发的,每个时钟周期内,它在时钟的上升沿进行内部同步。为了使中断得到响应,INTR 必须在现行指令或数据块传送指令全部动作结束之前的时钟周期内出现(高电平)。在中断响应时序期间,禁止下一个中断。虽然被自动推入堆栈的标志(FLAGS)寄存器反映了中断之前处理器的状态,中断允许位在任一中断响应(INTR, NMI, 软件中断或单步中断)的过程中被复位。直到老的 FLAGS 寄存器内容恢复之前,中断允许位总为零。除非专门用一条指令置位。

在中断响应时序期间(图 6),处理器执行两个连续的中断响应周期。从第一个总线周期的  $T_2$  到第二个总线周期  $T_2$ ,8086 发出封锁(LOCK)信号。直到第二个总线周期结束之前,局部总线“保持”请求将不被理睬。在第二个总线周期中,从外部中断系统(即 8259A PIC)取一个字节,用它来识别中断源的类型。将这个字节乘以 4,作为进入中断向量检索

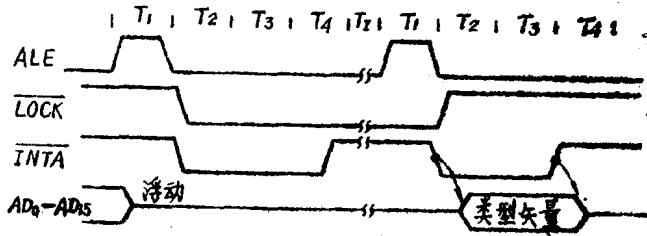


图 6 中断响应时序

表的指示字。在中断允许位和采样周期所限定的条件下，保持高电平的 INTR 信号将被响应。中断返回指令包括从堆栈中弹出标志操作，当标志被恢复时，就恢复了原来的中断允许位状态。

### 暂停

当执行软件“HALT”(暂停)指令时，处理器根据搭接模式，以两种方法之一指示出它正在进入“HALT”状态，在最小模式中，处理器发出一个非限定总线控制信号的ALE。在最大模式中，处理器在 $\overline{S_2}$ 、 $\overline{S_1}$ 、 $\overline{S_0}$ 上发出相应的“暂停”状态，而8288总线控制器发出一个ALE信号。在8086处于“暂停”状态下，局部总线进入“保持”(HLDA)时，8086不脱离“暂停”状态。在这种情况下，处理器将重新发出HALT指示。中断请求或RESET能迫使8086脱离“暂停”状态。

### 通过 LOCK 进行的读/修改/写(信号灯)操作

在执行一条指令期间，当需要直接连续的总线周期时，由处理器提供 LOCK 状态信息。这给处理器提供对存储器进行读/修改/写操作(例如，在执行寄存器与存储器互换指令过程中)而不会出现另一个系统总线主设备插入存储周期。这对于多处理器系统实现“test and set lock”(检测与置位封锁)操作是很有用的。在软件“LOCK”前缀指令被EU译码后的一个时钟周期，激活 LOCK 信号(低电平)。在“LOCK”前缀指令后的指令的最后总线周期结束时，LOCK无效。在 LOCK有效时，屏蔽所有的中断，并记录于 RQ/GT 引线端的请求信号，而在 LOCK 结束时处理。

### 经过 TEST 实现外同步

作为中断和通用 I/O 功能的另一选择，8086 提供一个软件可检测的输入，它称为 TEST 信号。在任何时间，程序都可执行一条 WAIT 指令。如果此刻 TEST 信号无效(高电平)，程序执行暂停，处理器等待 TEST 信号变为有效。TEST 保持有效的时间必须至少 5CLK 周期。在此以前，CPU 反复地重复执行 WAIT 指令。但这个活动并不占用总线周期，在等待时处理器处于空闲状态。如果总线进入“保持”，则所有的8086驱动器都进入高阻态。如果允许中断，在处理器处于等待时中断可以出现。当中断出现时，处理器进入中断，一旦从中断返回，再重新读取和执行 WAIT 指令。

### 基本系统定时

工作在最小模式和最大模式的处理器的典型系统配置，分别如图4a和4b所示。在最小模式中 $\overline{MN}/\overline{MX}$ 引线端接 $V_{cc}$ ，处理器发出总线控制信号。在最大模式中， $\overline{MN}/\overline{MX}$ 引线端接 $V_{ss}$ ，处理器发送编码的状态信息，8288总线控制器用它产生与多总线相容的总线控制信号。图5示出了信号定时关系。



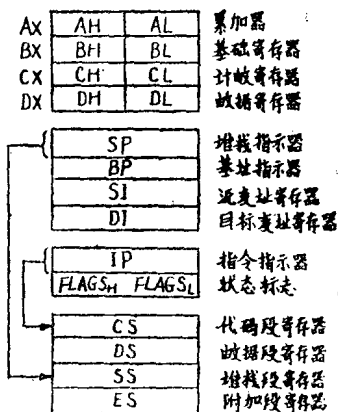


图 7 8086寄存器模块

### 系统定时——最小系统

读周期在 $T_1$ 开始同时发出地址锁存允许(ALE)信号。这个信号的后沿(下降沿)用来锁存地址信息,即把此刻其局部总线上的有效信息选通到8282/8283锁存器。用 $\overline{BHE}$ 和 $A_0$ 信号寻址低位字节、高位字节或高低位字节。从 $T_1$ 到 $T_4$ , $M/\overline{IO}$ 信号指明是存储器或是I/O操作。在 $T_2$ ,地址从局部总线上消失,同时总线变为高阻态。在 $T_2$ 还发出读控制信号。读信号(RD)导致被寻址器件把它的 $\overline{data}$ 总线驱动器选通到局部总线。尔后局部总线上的有效数据可被利用,及被寻址器件将READY线驱动为高电平。当处理器使读信号返回到高电平时,被寻址器件将再次使其 $\overline{data}$ 总线驱动器成为三态。如果需要收发器(8286/8287)缓冲8086局部总线,8086提供 $\overline{DT/R}$ 和DEN信号。

写周期也与发出ALE和发送地址同时开始。还是用 $M/\overline{IO}$ 信号来指明是存储器或I/O操作。在紧接着地址后的 $T_2$ ,处理器发出写到寻址单元的数据。直到 $T_4$ 的中间时刻,这一数据一直保持有效。在 $T_2$ 、 $T_3$ 和 $T_4$ 期间,处理器发出写控制信号。与读信号不同的是,写(WR)信号在 $T_2$ 一开始就变为有效,而读信号在进入 $T_2$ 后延迟一点时间才变为有效,以便为总线浮动提供时间。

用 $\overline{BHE}$ 和 $A_0$ 信号选择存储器/I/O字的适当字节进行读或写,按照下表来选择:

BHE	$A_0$	功 能
0	0	全字
0	1	读或写奇数地址相应的高位字节
1	0	读或写偶数地址相应的低位字节
1	1	无