

IBM PC/XT

硬件技术与维修技术

实用手册

(微机配套资料)

北京信通电脑技术公司

一九八六年四月

内 容 简 介

本手册根据国内广大IBM PC机用户的实际需要,在IBM PC繁多的资料中提取最实用的部分编成。内容包括PC硬件系统的介绍,其中有系统板、盘驱动器、各种接口板和各种外设的工作原理和逻辑图。在此基础上,由浅入深,系统地介绍了PC的维修技术,其中结合我们维修的经验和体会,加以整理总结。本书语言通俗易懂,插图较多,便于理解掌握,可供国内IBM PC及其兼容机使用人员作为工作查阅手册,也可作为学习班的教材或自学之用。

《微机实用手册》编辑部

目 录

第一部分 IBM PC/XT硬件技术

序 言

第一章	8088 CPU	(1)
第二章	总线操作	(6)
第三章	配置方式	(9)
第四章	存储器访问	(13)
第五章	输入/输出	(16)
第六章	中断系统	(17)
第七章	寻址方式	(23)
第八章	8086 /8088指令系统.....	(28)
第九章	8087数值数据处理器 (NDP)	(52)
第十章	8284时钟产生/驱动器.....	(95)
第十一章	8288总线控制器	(99)
第十二章	8259A可编程序中断控 制器 (PIC)	(105)
第十三章	8253可编程序定时器 (PIT)	(123)
第十四章	系统板	(132)
第十五章	音频系统	(156)
第十六章	电源系统	(157)
第十七章	键盘	(160)
第十八章	外围设备	(164)

第二部分 IBM PC/XT维修技术

序 言

第十九章	系统板及初始上电的故障	(249)
第二十章	存储器故障	(258)
第二十一章	键盘故障	(265)

第二十二章	单色显示器故障·····	(267)
第二十三章	彩色/图形显示器故障·····	(274)
第二十四章	软盘驱动器故障·····	(282)
第二十五章	并行打印机卡故障·····	(292)
第二十六章	RS232异步通讯卡故障·····	(294)
第二十七章	游戏控制卡故障·····	(296)
第二十八章	打印机故障·····	(298)
第二十九章	其它部件的故障·····	(309)

3. SSO (输入)

最大配置时接高电平。

4. RD (输出, 三态)

$\overline{RD}$ 信号在读周期中变为低电平时, 表示CPU正在进行读存储器或I/O操作。

5. READY (输入)

READY输入端在被寻址的存储器或I/O没有完成数据传送之前处于低电平, 使CPU自动插入一个或几个等待状态, 以便与低速存储器或I/O设备同存。

6. INTR (输入)

中断请求信号INTR是依靠电平触发的, CPU在每条指令的最后一个时钟周期对它进行取样、然后决定是否进入中断响应周期。在中断响应周期中CPU从中断源取得中断矢量、并根据这个中断矢量可以在存储器的中断指针表中查到相应的中断服务子程序的入口地址。

7. $\overline{TEST}$ (输入)

CPU在执行WAIT指令时, 每隔5个时钟周期对 $\overline{TEST}$ 输入进行一次测试, 如果 $\overline{TEST}$ 是高电平, CPU就进入踏步状态, 隔5个时钟周期再对 $\overline{TEST}$ 进行取样; 当 $\overline{TEST}$ 是低电平时, CPU就执行下一条指令。

8. NMI (输入)

NMI是非屏蔽中断输入信号, 用上升沿触发, 不能用软件禁止。NMI信号引起的中断使CPU自动地到中断指针表中找出类别2的中断入口地址。

9. RESET (输入)

RESET是复位信号, 它至少应当保持四个时钟周期的高电平, 使CPU停止正在进行的操作, 并使内部的标志寄存器, 段寄存器和指令队列复位到起始状态。

10. $\overline{S_2}, \overline{S_1}, \overline{S_0}$ (输出, 三态)

这三条状态输出线表示出CPU总线周期的操作类型。8288总线控制器利用这三个状态信息产生出访问存储器和I/O的控制信号。

11. $\overline{RQ/GT_0}, \overline{RQ/GT_1}$ (输入/输出)

这是两条双向的信号线, 是总线上其它主设备向CPU请求使用总线和CPU响应的信号线。两条信号线可以同时联系两个主设备, $\overline{RQ/GT_0}$ 的优先权高于 $\overline{RQ/GT_1}$ 。

12. $\overline{LOCK}$ (输出三态)

CPU发出 $\overline{LOCK}$ (低电平有效) 信号向总线上的其它主设备表示不允许他们占用总线。 $\overline{LOCK}$ 信号是用软件设置的, 即在一条指令的前面加上'LOCK'前缀, 它在该指令执行完后就失去作用。

13. QS_1, QS_0 (输出)

QS_1, QS_0 用于表示当前CPU中指令队列的状态。

14. $M/\overline{IO}$ (输出三态)

这个信号用来区别访问存储器还是访问I/O。

15. $\overline{WR}$ (输出三态)

这个信号表示CPU在进行写操作, 低电平有效。

16. $\overline{INTA}$ (输出)

$\overline{INTA}$ 信号在中断响应周期中相当于一个读选通信号。

17. ALE (输出)

这是CPU把分时复用的地址信息锁存到地址锁存器8282/8283时所用的触发信号。

18. $\overline{DT/R}$ (输出, 三态)

这个信号用于控制双向驱动器8286/8287的数据传输方向。

19. $\overline{DEN}$ (输出, 三态)

这是双向驱动器8286/8287的选通信号。

20. HOLD (输入), HLDA (输出)

HOLD是其它总线主设备向CPU请求使用总线信号, 当CPU让出总线时, 就发出HLDA信号作为响应。

21. CLK (输入)

时钟信号, 占空度33%为最佳状态。

22. V_{cc}

电源电压采用 $+5V \pm 10\%$ 。

23. GND

地线。

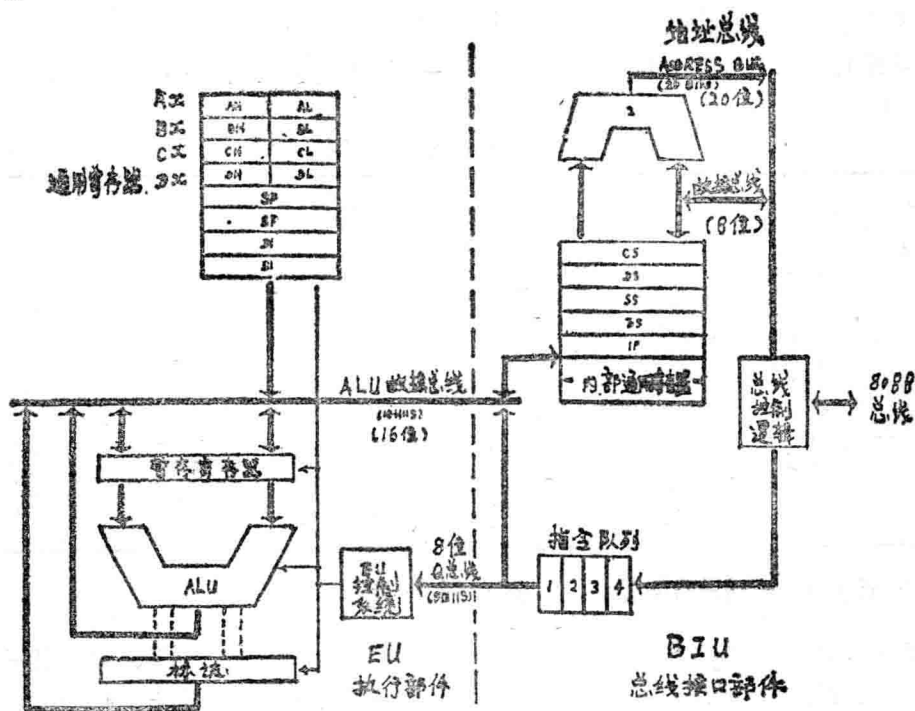


图 1-1-2 8088的基本结构方块图

注: ADDRESS BUS地址总线

(三) 8088 CPU的结构

8088 CPU由执行部件(EU)和总线接口部件(BIU)两部分组成(见图1-1-2)。执行部件EU负责全部指令的执行, 向总线接口部件BIU提供数据和地址, 并对通用

寄存器和标志寄存器进行管理。除了少数几条控制管腿之外，EU和外界是隔绝的。BIU由段（SEGMENT）寄存器和通信寄存器、指令计数器和指令目标代码的队列组成，负责执行所有的外部总线周期。BIU中的加法器把段寄存器中的数值和位移量寄存器的数值相加，得出20位的地址；把取来的指令装进指令队列中，然后再被EU取走。算术逻辑部件（ALU）的数据总线用于EU和BIU之间传送数据。

当EU准备好执行指令时，它就从BIU的指令队列中取出一字节指令目标代码，然后加以执行。如果EU到指令队列中取指令时指令队列是空的，EU就处于等待取指令的状态。在指令执行过程中，如果需要访问存储器或I/O端口，EU就请求BIU进入访问存储器或I/O端口的总线周期。

EU和BIU部件的操作在CPU中是互相独立的。如果BIU正在取指令的总线周期中EU发出访问总线的请求，BIU必须在完成它的取指令周期后才能响应EU的请求。

（四）通用寄存器

8088的EU中有八个16位的通用寄存器。这八个寄存器又可分为两组，每组四个。一组叫数据寄存器，另一组叫指针和变址寄存器。数据寄存器的特点是高八位和低八位可以分别寻址，所以每一个数据寄存器既可以用作八位寄存器，也可以用作16位寄存器。指针和变址寄存器则只能作为16位寄存器使用。这八个通用寄存器都可以象累加器那样参加算术和逻辑操作，而且在某些指令中具有隐含的特性（即在指令中不需要写出该寄存器的名称），如下表所示：

表 1-1-2 通用寄存器的隐含特性

寄存器	隐 含 操 作	寄存器	隐 含 操 作
AX	字的乘法，字的除法，字的I/O	CL	变量移位和循环
AL	字节的乘法，字节的除法，字节的I/O， 翻译指令，十进制算术运算	DX	字的乘法，字的除法，间接I/O
AH	字节乘法，字节除法	SP	栈操作
BX	翻译指令	SI	字符串操作
CX	字符串操作，循环指令	DI	字符串操作

（五）段（SEGMENT）寄存器

8088的BIU中有四个16位的段寄存器。代码段寄存器CS指向当前程序的代码段，取指令靠它进行寻址。堆栈段寄存器SS指向当前的堆栈，栈操作时靠它进行寻址。数据段寄存器DS指向当前的数据段，数据段通常用于存放程序中的变量。附加段寄存器ES指向当前的附加段，附加段也是用于存储数据。以上所述的四种段寄存器的内容是可以指令进行设置或修改的。

（六）指令指针寄存器

16位的指令指针寄存器IP的作用与8080中的程序计数器相似，它的内容是CS寄存器

的一个位移量。当BIU到存储器中取指令时，实际的地址就是以CS寄存器的内容为基址，以IP的内容为位移量，即：

$$\text{实际地址} = \text{基址} \times 16 + \text{位移量}$$

（七）标志寄存器

CPU的EU中有一个标志寄存器。标志寄存器的内容是一些标志位，用于反映算术或逻辑操作结果的特征，如右图所示：

这些标志位的说明如下：

(1) AF置“1”表示16位寄存器的低8位向高8位进位或借位。这个标志位可用于十进制算术指令。

(2) CF置“1”表示16位或8位多字节加减运算时最高位发生进位或借位。CF也可以用于循环移位指令移入的内容来决定。

(3) OF置“1”表示算术运算的结果发生溢出。有一条中断指令（INTO）能够在OF置“1”时，产生硬件中断，使程序自动转向溢出中断处理服务子程序。

(4) SF表示补码运算时，运算结果的正负号，SF=1时为负值，SF=0时为正值。

(5) PF表示操作结果“1”位的奇偶数，PF=1时表示偶数，PF=0表示奇数。这个标志位可用于数据传输时的奇偶校验。

(6) ZF置“1”表示操作结果为零。

(7) DF标志可用程序进行设置，当DF=1，表示字符串指令执行时，地址从高到低自动减1或从右向左；当DF=0，表示地址从低到高或从左向右。

(8) IF置“1”表示准许CPU响应外部可屏蔽的中断请求，IF=0表示禁止外部可屏蔽的中断。IF对不可屏蔽的中断和内部中断没有影响。IF标志可用中断指令STI和INT进行置位。

(9) TF置“1”表示让CPU执行单步方式，以便调试程序。在这种方式中，CPU每执行完一条指令后就自动产生一次内部中断。TF标志可用程序进行置位。

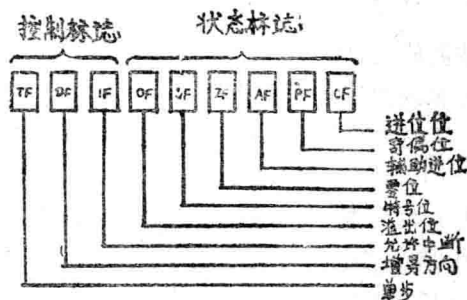


图 1-1-3 标志寄存器

第二章 总线操作

为了说明总线分时复用的操作方式,让我们先看看8088 BIU的总线周期。在总线周期中,总线上先出现I/O外围芯片或存储单元的地址,然后出现读或写的控制信号和相应的数据。被寻址的存储单元或I/O芯片在写周期中从数据总线上接受数据;而在读周期中却把数据放到总线上去。当总线周期结束时,被寻址I/O设备或存储单元就把写入的数据锁存起来或再读进新的数据。

所有的总线周期至少由四个时钟周期(或T状态)组成,即 T_1 、 T_2 、 T_3 和 T_4 。在状态 T_1 中,CPU把存储单元或I/O芯片的地址放到总线上。如果是在写的总线周期中,CPU就从状态 T_2 起到状态 T_4 之间把数据送到总线上;在读的总线周期中,CPU就在状态 T_3 和 T_4 中从总线上接收数据,并且在状态 T_2 中使分时复用的地址/数据总线处于高阻状态,允许CPU有时间从写方式(输出地址)变为读方式(输入数据)。

必须指出:只有在BIU为了填满指令队列或者当EU在执行指令,需要申请一个总线周期的时候,BIU才执行一个总线周期。所以,在两个总线周期之间的间隙可能存在一些没有BIU活动的时钟周期。这些没有BIU活动的时钟周期被称为空闲状态 T_1 。空闲状态产生的条件可能是各种各样的,例如在总线主设备之间交换总线控制权的时候,或者在执行

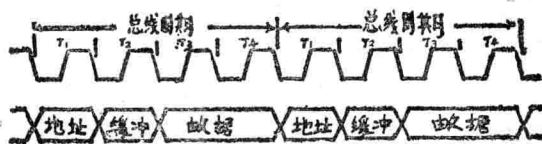


图 1-2-1 典型的BIU总线周期

一条占周期数很多的‘长’指令,比如说一条8位寄存器的乘法指令(MUL)时,8088执行这样一条指令要用70~77个时钟周期。除了上述的空闲状态之外,考虑到存储器或I/O的数据传送速率可能不够快,8088CPU还具备一种插入附加状态的功能。这些附加状态称为等待状态 T_w 。

需要的时候, T_w 可以插在 T_3 和 T_4 之间。在 T_w 状态中,总线上的数据保持不变。等到存储器或I/O设备准备好数据或接收完数据时,再用信号通知CPU让它离开等待状态,进入状态 T_4 。下面用定时波形图表示出8086CPU的读和写总线周期。这里所表示的是最小方式的定时波形,最大方式的定时波形将在下面谈到。从读、写总线周期时序图上,可以看出,8088CPU在状态 T_1 把20位的地址送上分时复用的地址/数据总线,在整个总线周期中,地址线 $A_{15} \sim A_8$ 上始终保持着地址信息。在状态 T_2 中,8088CPU把地址/数据线($AD_7 \sim AD_0$)上的地址信息撤除,如果是在读周期中就让这些线($AD_7 \sim AD_0$)处于高阻状态;如果是在写周期中,就把写入的数据送上这些线。地址/状态线上保持着总线周期状态信息。在状态 T_3 中总线周期状态仍然保持在地址/状态线上,写入或读出的数据

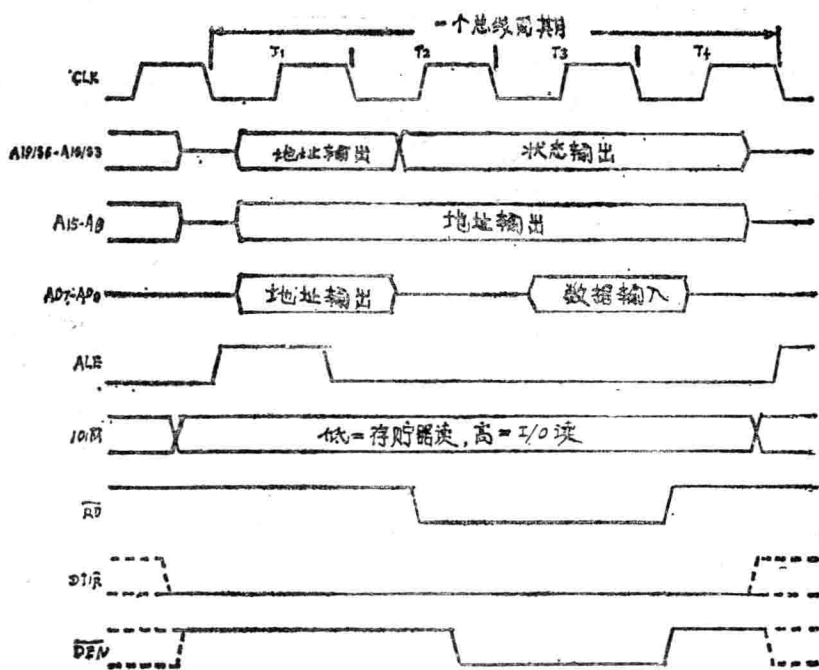


图 1-2-2 8088的读总线周期

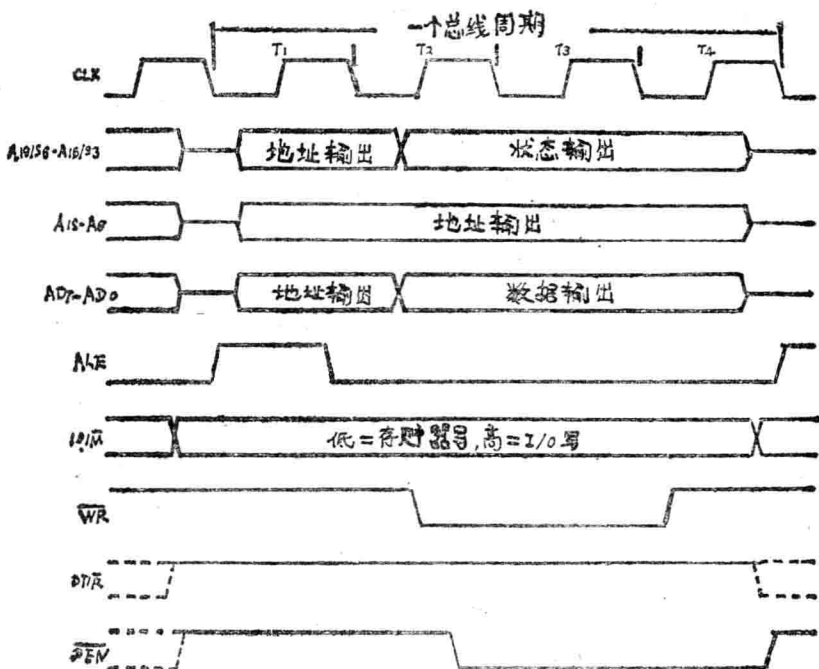


图 1-2-3 8088的写总线周期

可以通过地址/数据总线进行传送。总线周期在状态 T_4 中结束，同时，控制线上的控制信号也结束，被寻址的存储器或I/O芯片不再受地址线的选择。

多数存储器和外围芯片都要求在整个总线周期中保持稳定的地址信息，因此在每一个总线周期的 T_1 状态中，用允许地址锁存（ADDRESS LATCH ENABLE）信号ALE的后沿把地址信息锁存起来。在最小方式时，ALE信号是直接从CPU发出的，在最大方式时，ALE信号是间接地通过总线控制器8228发出的。

第三章 配置方式

8088CPU有最小和最大两种系统结构方式。用户定义这两种方式可以通过管腿MN/ $\overline{\text{MX}}$ 。当这条管腿为高电平时，CPU选择最小方式；低电平时它选择最大方式。

表 1-3-1 最小/最大方式的管腿功能的差别

8088		
管腿编号	最 小 方 式	最 大 方 式
31	HOLD	$\overline{\text{RQ/GT0}}$
30	HLDA	$\overline{\text{RQ/GT1}}$
29	$\overline{\text{WR}}$	LOCK
28	$\text{IO}/\overline{\text{M}}$	$\overline{\text{S2}}$
27	$\text{DT}/\overline{\text{R}}$	$\overline{\text{S1}}$
26	$\overline{\text{DEN}}$	$\overline{\text{S0}}$
25	ALE	QSO
24	$\overline{\text{INTA}}$	QSI
34	SSO	高阻状态

在最小方式中，管腿MN/ $\overline{\text{MX}}$ 接高电平（+5V）。在这种方式中，CPU用于构成单微处理机系统，并且不需要多总线结构。CPU本身产生全部的总线控制信号（ $\text{DT}/\overline{\text{R}}$ ， $\overline{\text{DEN}}$ ，ALE和M/ IO 或 $\text{IO}/\overline{\text{M}}$ ）和读写控制信号（ $\overline{\text{RD}}$ ， $\overline{\text{WR}}$ 或 $\overline{\text{INTA}}$ ），而且有允许其它总线主控者（8237或8257DMA控制器）请求使用总线的信号输入端和响应这种请求的信号输出端（HOLD和HLDA）。当某一总线主控者请求使用总线时，它把CPU的HOLD输入端置成高电平，当CPU响应它的请求时，就把HLDA输出端置成高电平，并且使自己的系统总线和控制线处于浮动状态。由于这种总线请求信号是异步产生的，所以CPU在每个时钟信号CLK的上升沿对HOLD输入信号进行采样，并且在空闲时钟周期的下降沿或现行总线周期的末了（如果在总线周期中遇到总线请求）发出响应信号HLDA。保持状态（HOLD）可以继续到总线主控者把HOLD信号置成低电平为止，然后CPU又重新得到总线的控制权。应注意的是在保持状态中CPU可以继续执行指令，一直到需要一个总线周期时才停止下来。IBM PC/XT是个多微处理器系统，故用最大方式，管腿MN/ $\overline{\text{MX}}$ 接地。许多总线控制信号是通过总线控制器产生的，而不是CPU直接产生的。

8288总线控制器利用CPU送给它的状态位 $\overline{\text{S0}}$ ， $\overline{\text{S1}}$ 和 $\overline{\text{S2}}$ 产生出总线周期中所需要的全部控制信号。根据这三个状态位译码产生的控制信号，如下表所示：

下面是最大方式中某些管腿的说明。

- (1) $\overline{\text{RQ/GT1}}$ ， $\overline{\text{RQ/GT0}}$

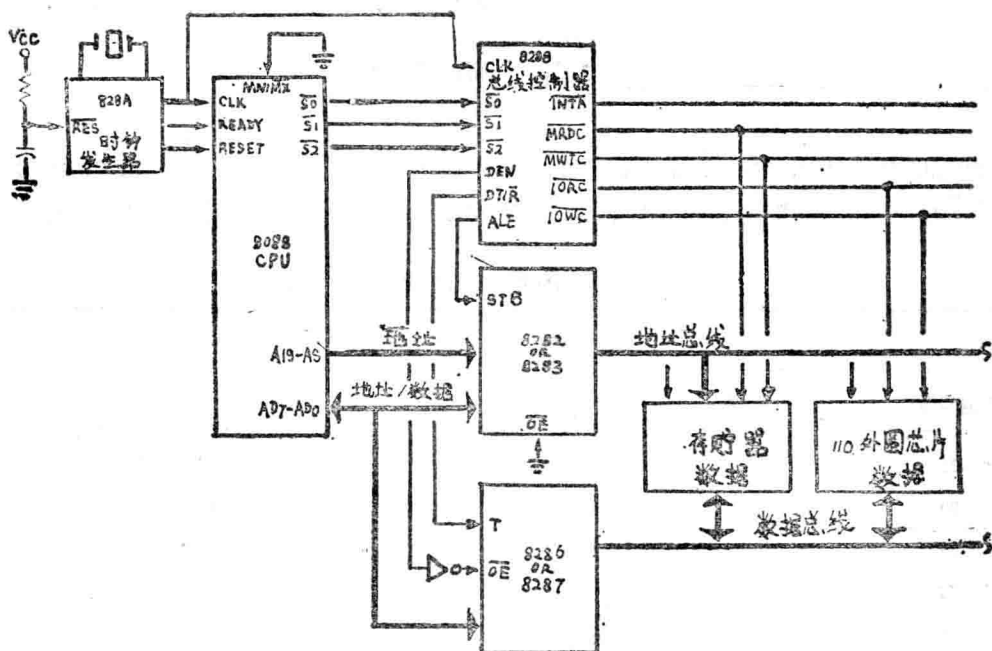


图 1-3-1 基本的最大方式系统

表 1-3-2 状态位的译码

状 态 输 入			CPU的周期状态	8288命令
S ₂	S ₁	S ₀		
0	0	0	中断响应	$\overline{\text{INTA}}$
0	0	1	读I/O端口	$\overline{\text{IORC}}$
0	1	0	写I/O端口	$\overline{\text{IOWC}}, \overline{\text{AIOWC}}$
0	1	1	暂 停	无
1	0	0	取指令	$\overline{\text{MRDC}}$
1	0	1	读存储器	$\overline{\text{MRDC}}$
1	1	0	写存储器	$\overline{\text{MWTC}}, \overline{\text{AMWC}}$
1	1	1	被动状态	无

在最大方式中总线的请求授予信号线是 $\overline{\text{RQ}}/\text{GT1}$ 和 $\overline{\text{RQ}}/\text{GT0}$ ，这两个信号的功能和最小方式中的 HOLD/HLDA 是一样的，所不同的是这两条信号线是双向的，而 HOLD 和 HLDA 是单向的。这两条信号线是专为利用8089I/O处理机（在本地方式）来构成多处理机系统而设计的。总线请求和授予的过程可以分为三个阶段，即：请求、授予和释放。系统总线上的其它处理机可以利用两条 $\overline{\text{RQ}}/\overline{\text{GT}}$ 线之一发出一个脉冲，表示请求使用总线，而CPU就在现行总线周期的末了或空闲的时钟周期的下降沿，在同一条信号线上发

出一个脉冲作为授予信号，向请求使用总线的处理机表示它已经让系统总线处于浮动状态。这时候CPU就进入保持状态。应当注意，在保持状态中CPU可以继续执行指令，一直到需要访问总线或指令队列被取空为止。第三个阶段是请求总线的处理机再次在 $\overline{RQ}/\overline{GT}$ 线上输出一个脉冲。这个脉冲向CPU表示它已准备好释放所用的总线，于是CPU在下一个时钟周期又获得访问总线的权利。在 $\overline{RQ}/\overline{GT}$ 线上的这些脉冲的来往传送都是和时钟信

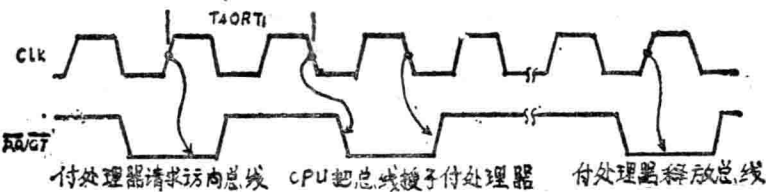


图 1-3-2 请求/授予的定时波形

号同步的，当然，CPU和请求总线的处理机必须使用相同的时钟信号，信号线 $\overline{RQ}/\overline{GT0}$ 比 $\overline{RQ}/\overline{GT1}$ 的优先权级别高。如果这两条信号线同时请求使用总线，那么在 $\overline{RQ}/\overline{GT0}$ 线发信号的处理机应首先得到响应。在 $\overline{RQ}/\overline{GT1}$ 线发信号的处理机，只有在前一个处理机释放总线大约一至二个时钟周期的通道传输延迟之后才能得到响应。这两条 $\overline{RQ}/\overline{GT}$ 线与最小方式中的HOLD线一样，具有比矢量中断更高的优先权级别。CPU从接收到请求总线的脉冲到发出授予脉冲所花的时间，在不同的条件下是不相同的，如下表1-3-3所示。

这种延迟时间在正常执行指令的情况下（即正在执行一条没有访问存贮器的指令）如果 $\overline{LOCK}$ 不起作用，可能只有三个时钟周期，至多不超过十个时钟周期。如果正在执行一条有 $\overline{LOCK}$ 前缀的指令，或正处于中断响应周期，延迟的时间就增长。当执行一条有 $\overline{LOCK}$ 的XCHG指令时，最大的时间延迟可以达到39个时钟周期。执行一条有 $\overline{LOCK}$ 的长指令，延迟时间可能更长，当然这种情况最好设法避免。

表 1-3-3 请求/响应的时间延迟

操 作 条 件	请求/响应延迟 (时钟周期)	
	8086	8088
正常的指令操作 ($\overline{LOCK}$ 不起作用)	3~6 (10*)	3~10
INTA周期 ($\overline{LOCK}$ 起作用)	15	15
加 $\overline{LOCK}$ 的XCHG指令 ($\overline{LOCK}$ 起作用)	24~31 (39*)	24~39

*括号中的数值表示指令正在访问奇数地址开始的16位数据所需的时钟周期数

在处理机释放总线之后，8088在收到释放脉冲的两个时钟周期之后（或在最小方式中HOLD变为无效两个时钟周期之后）才能重新得到总线的控制权。

如果在RESET信号转向无效时出现HOLD请求信号，这个HOLD请求应能得到立即的响应。这对于把程序调入内存，尤其是在CPU开始工作之前设置好存贮单元的地址FFFF0H是很必要的。在最大方式中，当RESET信号变成无效的一至二个时钟周期之后，通过 $\overline{RQ}/\overline{GT}$ 线发出一个请求总线的脉冲，也可以得到响应。

(2) $\overline{LOCK}$

当用总线裁决器8289控制总线时，利用CPU的输出信号 $\overline{LOCK}$ ，可以在执行一条指

令的时间内保持系统总线的使用权。这个 $\overline{\text{LOCK}}$ 信号是用软件控制的，即在一条指令之前加上 $\overline{\text{LOCK}}$ 前缀。这个前缀经过EU的译码后，通过BIU使 $\overline{\text{LOCK}}$ 在下一个时钟周期变为有效。这个 $\overline{\text{LOCK}}$ 信号一直保持有效，直到一条指令执行完后一个时钟周期为止。

(3) QS1, QS0

输出信号QS1和Q0表示CPU内部指令队列的状态，Intel的仿真器在跟踪操作方式

表 1-3-4 队列状态位的编码

QSI	QSO	队 列 状 态
0	0	无操作、队列中的指令没有被取走
0	1	指令的第一字节，从队列中取出
1	0	队列已空，在执行一条传送指令时，队列被重新初始化
1	1	从队列中取出的字节不是指令的第一字节，而是随后的字节

时将用到这些信号。这些信号的编码所代表的意思如上表所示。

第四章 存储器访问

(一) 存储器结构

8088CPU的地址总线有20位，可以寻址1,048,576字节的存储空间。这一兆字节的存储空间可以分为两个512K字节的存储体，一个用于存放奇数地址的字节（高字节）；另一个用于存放偶数地址的字节（低字节）。偶数地址存储体的数据线与数据总线的低八位连接（D7~D0），奇数地址存储体的数据线与数据总线的高八位连接（D15~D8）。地址线A19~A1可以同时对这两个存储体的存储单元寻址。A₀用于存储体的选择，当A₀=0时选择偶数地址的存储体；A₀=1时不能选择偶数地址的存储体。选择奇数地址存储体采用另一单独的信号BHE，利用A₀和BHE这两个控制信号可以同时对这两个存储体进行读或写操作（即16位数据的操作），也可以单独对其中一个存储体进行读或写操作。因为8088只有8位数据线，所以存储器地址空间在逻辑上可以看作1兆字节的线性数组，一个单一的存储体。地址位A₀也用于寻址，但没有使用BHE信号。8088进行16位数据操作时，不论低字节放在偶数地址或奇数地址，都需要用两个总线周期，且都是8088自动完成的。

(二) 存储器的段（SEGMENT）结构

8088把1兆字节的存储器空间分为若干段来使用，每一段代表64K字节长的连续存储

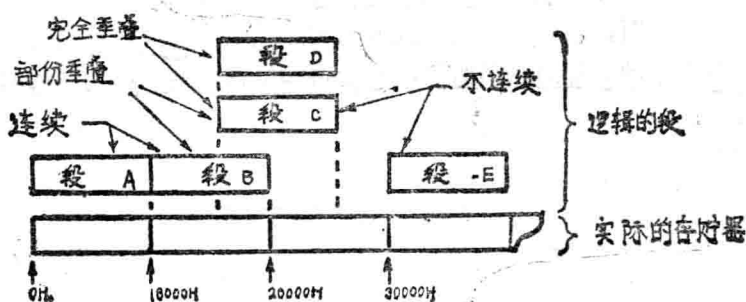


图 1-4-1 存储器的段结构

单元。每个段的起始地址也叫基址，是用软件设置的。段和段之间可以是连续的，分开的，部分重叠的，也可以是完全重叠的。

段的基址放在BIU的段寄存器中。BIU中有四个段寄存器（CS，DS，SS，ES），所以CPU可以通过这四个段寄存器访问四个段。如果要访问其它的段，则必需用软件修改段寄存器的内容，把基址指向所要访问的存储段。8088存储器的段结构对程序的模块化设计很方便。这四个段可支持64K字节的代码段，64K字节的堆栈段和128K字节的数据