

8086微计算机系列 用户手册

(中 册)

《舰船导航》编辑部

一九八三年六月

附 录 A

应 用 说 明

本附录是与Intel公司8086微处理器系列有关的应用说明,它包括:

AP—67	8086系统设计	(2)
AP—61	8086的多任务处理	(72)
AP—50	8089系统的调试对策及研究	(89)
AP—51	使用8289总线仲裁器的8086、8088、8089 多处理器系统的设计	(119)
AP—59	8259A可编程中断控制器的应用	(143)
AP—28A	Intel多总线接口	(181)
AP—43	执行8086程序码的iSBC—957执行工具的使用	(211)

AP—67

8086 系 统 设 计

目 录

1. 引言
2. 8086 概述和基本系统概念
 - A 总线周期定义
 - B 地址和数据总线概念
 - C 系统数据总线概念
 - D 多处理器环境
3. 8086 系统细节
 - A 工作模式
 - B 时钟信号的产生
 - C 复位
 - D “准备好”实现和定时
 - E 中断结构
 - F 8086 总线定时图解释
 - G 总线控制传送
4. 与I/O的接口
5. 与存储器的接口
6. AP—67附录

1. 引言

INTEL公司的微处理器和系统部件的新系列8086, 为设计者提供了一种先进的系统结构, 以满足广泛应用。在该系统内能进行各种不同速度、配置和部件选择, 使一个特定设计方案的成本和性能目标达到最佳。但是更重要的是, 8086系列概念使设计者在单一设计中构造一系列具有多级增强功能的系统, 并提供未来设计的途径。

本应用说明用以进行系统硬件的实施, 并介绍用该系列的8086CPU 部件构成的系统实例, 给出应用方法和定时分析, 以帮助设计者理解系统规格, 优点和局限。读者可以参阅另外的 Intel 资料, 如8086用户手册(9800722A), 8086汇编语言参考指南(9800749A), AP—28A多总线接口(98005876B), Intel 多总线规范(9800683), AP—45 8202动态RAM控制器使用(9800809A), AP—51用8289总线仲裁器设计, 8086、8088、8089多处理器系统和AP—59可编程中断控制器8259A的使用。有关 Intel 其它参考资料, 在本说明中随处给以说明。

2. 8086概述和基本系统概念

2A. 8086总线周期定义

8086是一个真正的16位微处理器,具有16位内部和外部数据通路,1兆字节存储器地址空间(2^{20})和一个独立的64K字节(2^{16})输入/输出地址空间。CPU通过一个20位多路复用地址、状态和数据总线和一个命令总线与它的外部环境通讯。为传送数据或取出指令,CPU要执行一个总线周期(图2A1)。最小总线周期由四个叫做T状态的时钟周期组成。在第一个T状态(T₁),CPU在20位多路复用地址/数据/状态总线上发出地址。在第二个T状态(T₂),CPU从总线上撤消地址,或者使总线低十六位输出成为三态,以便为读周期做准备,或者发出写数据。数据总线收发器在T₁或者在T₂被使能,这取决于8086系统结构和传送方向(进入CPU或由CPU发出)。读、写和中断应答命令,总是在T₂被使能。8086的最大结构模式(后面讨论)在T₃提供一个写命令使能,以保证数据建立时间先于命令作用。

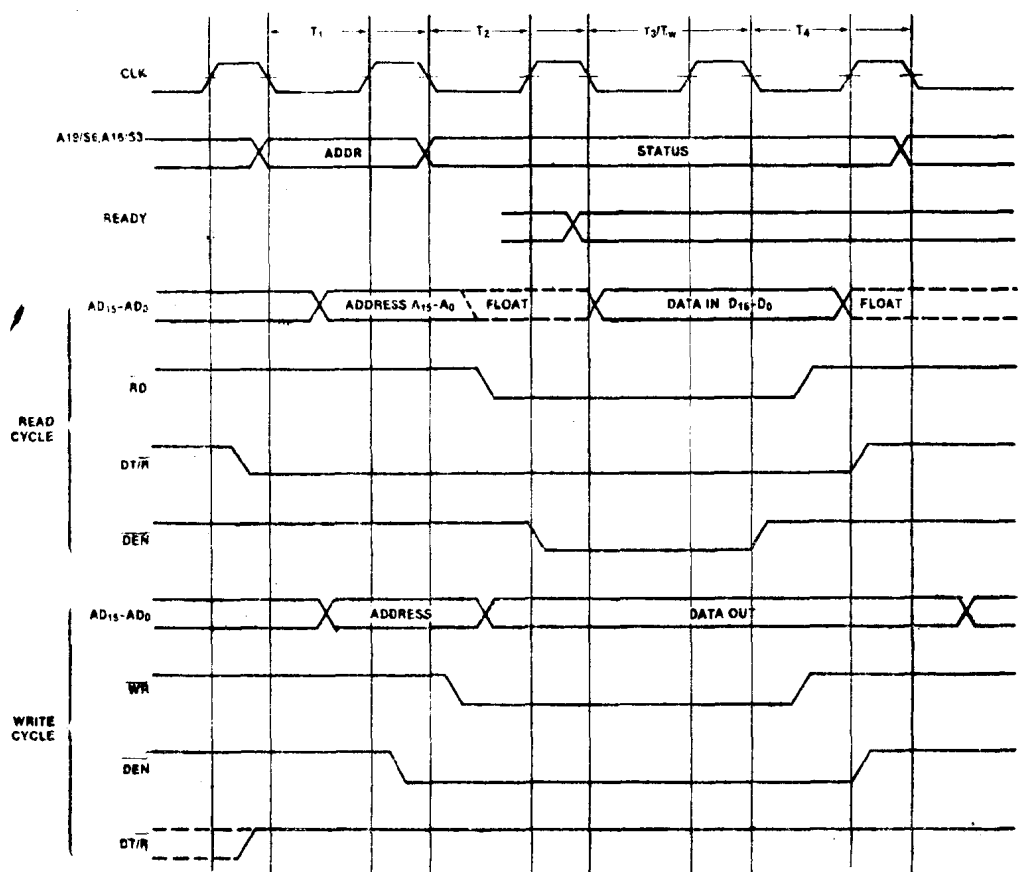


图2A1 基本 8086 总线周期

在T₂期间,多路复用总线的高四位地址(A19~A16),转换为总线周期状态(S6、S5、S4、S3)。状态信息(表2A1)主要用于诊断监视。但是S3和S4的译码能用以选择四个存储体中的一个,这些存储体各自对应指定的段寄存器。这种技术允许存储器按段划分,从而使存储器寻址范围超过1兆字节,它还能使对某一段的错误写操作不致跳进另一段,而破坏那里的

信息，从而提供了一定程度的保护。

在T3期间里，CPU在总线高四位上继续提供状态信息，在总线低十六位上，或者继续发出写数据，或者采样读数据。如果选定的存储器或I/O设备不能以CPU最大传送速率传送数据，该设备必须通知CPU“未准备好”，并迫使CPU在T3之后引入附加时钟周期（等待状态TW）。“未准备好”指示必须在T3起动前送到CPU。TW期间的总线活动与T3期间一样。当选定的设备有足够时间完成传送时，它就发出“准备好”信号，并使CPU脱离TW状态继续工作。在最后一个等待状态期间，或者当不要求等待状态时，在T3期间CPU锁存总线上的数据。总线周期在T4结束（命令线禁止，选定的外部设备从总线脱开）。总线周期在系统设备中是一种异步事件，它包括选择设备的地址，随后的读选通或是数据和写选通。选定设备在写周期期间接收总线数据，在读周期期间驱动所需数据到达总线。在命令结束时，该设备锁存写数据或使其总线驱动器禁止。在总线周期期间，设备发出的唯一控制就是等待周期的插入。

必须对存储器或I/O设备传送（进入或送出）指令或操作数时，CPU才执行总线周期。不执行总线周期时，总线接口执行空闲周期（T1）。在空闲周期期间，CPU在高位地址总线上继续驱动来自前一个总线周期的状态信息。如果前一个总线周期是写，则CPU继续在多路复用总线上驱动数据，直到下一个总线周期开始。如果CPU执行空闲周期是在读周期之后，则CPU在要求下一个总线周期之前不驱动低十六位总线。

由于PCU可予取多达六个字节的指令流，存放在内部指令排队机构中以备执行，所以取指令和传送操作数的时间关系可以交错，并由附加的取指总线周期分离进行。一般说来，如果一条指令取到8086的内部指令排队机构中，则在该指令从排队机构中移出并执行之前，可以取几条附加指令。如果从排队机构中取来的执行指令是转移指令或其它控制传送指令，则保留在排队机构中的任何指令就被丢弃而不执行，这不影响CPU操作。在执行一条特定指令时所观测到的总线活动，与上一条指令有关，但在特定顺序中总是确定的。

表 2A1

S3	S4	
0	0	交替(对应附加段ES)
1	0	堆栈(对应堆栈段SS)
0	1	代码/无(对应代码段CS或非法零)
1	1	数据(对应数据DS)
S5 = IF		(中断使能标志)
S6 = 0		(指示8086在总线上)

2B. 8086地址和数据总线概念

由于大多数系统存储器和外围设备在总线周期期间要求稳定的地址，故在T1期间，多路复用地址/数据总线上的地址应被锁存，并用锁存地址选择所要求的外围设备和存储器单元。由于8086具有16位数据总线，所以8085系列的多路复用总线元件不能用于8086（接在8—15位地址/数据总线的设备不能接收0—7位的字节选择地址）。为了分离总线（图2B1a），8086系统提供一个地址锁存使能信号（ALE），以便把地址锁存在8282或8283八位双稳存器中（时序图2B1）。锁存器或者是反向的（8283），或者是非反相的（8282），它们具有三态缓冲器驱动

的输出,驱动能力为32mA。带300PF容性负载的转换时间为22ns(反相)和30ns(非反相)。当ALE为高电平时,它们将地址信号直接传送到输出,而在ALE的下降沿将其锁存起来。这仅使地址存取和片选译码受到锁存延迟。通过低有效 $\overline{OE}$ 输入选通锁存器输出。多路复用地址/数据总线的分离(锁存由多路复用总线来的地址),可在系统的适当部位,或在CPU部位进行,得到分离的地址总线在整个系统内传送地址(图2B1b)。为了得到系统的最佳性能和多处理器与多总线结构的兼容性,后一种方法远较前者为好。在本说明的其余部分将假定总线是在CPU部位分离的

在程序设计者看来8086存储器地址空间是1兆字节序列,任何一个字节可包含一个八位数据元,而任意两个相邻的字节可包含一个十六位数据元。对字节或字地址(边界)没有限制。地址空间实际上分成两个高达512K字节的体,在物理上与一个十六位数据总线相连(图2B2)。一个体与十六位数据总线的低八位(D7—0)相连,并包含偶数编址字节($A_0 = 0$)。另一个体与数据总线的高八位(D15—8)相连,并包含奇数编址字节($A_0 = 1$)。在每个体中地址线A19—A1选定一个特定的字节。为了实现向偶数地址的字节传送(图2B3a),信息通过数据总线的低八位(D7—0)传送。 A_0 (低电平有效)用以选通与数据总线低八位相连的体,使其参与传送。8086提供的另一个信号,总线高位使能($\overline{BHE}$)用以禁止与数据总线高八位相连的体,使其不参与传送,为了防止向低八位体的写操作不破坏高位体的数据,这是必要的。由于 $\overline{BHE}$ 是与地址线A19—A16时序相同的多路复用信号,它也应由ALE锁存,以便在总线周期中提供稳定信号。在T₂到T₄期间 $\overline{BHE}$ 输出与状态线S₇多路复用,它们的状态相同。为了实现向奇数地址的字节传送(图2B3b),信息通过数据总线的高八位(D15—D8)传送, $\overline{BHE}$ (低电平有效)选通高位体,而 A_0 使低位体禁止。8086提供使能信号 $\overline{BHE}$ 和 A_0 引导数据传送到数据总线的适当部位,这对程序设计者是透明的。做

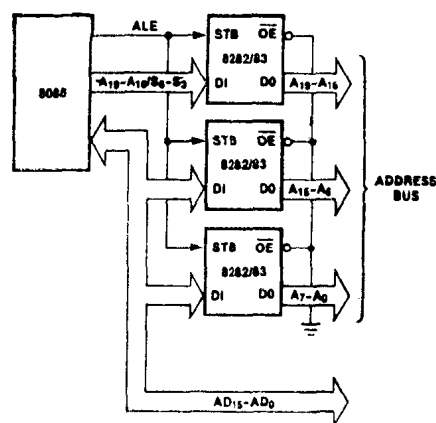


图2B1a 8086总线分离

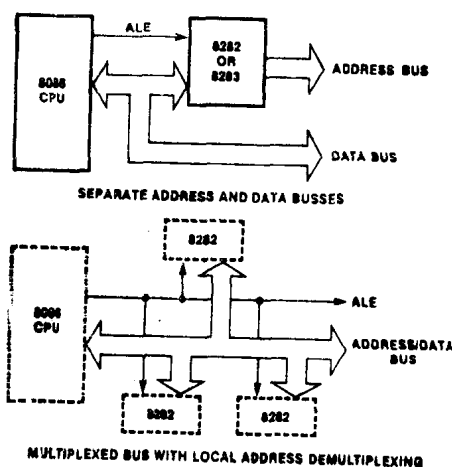
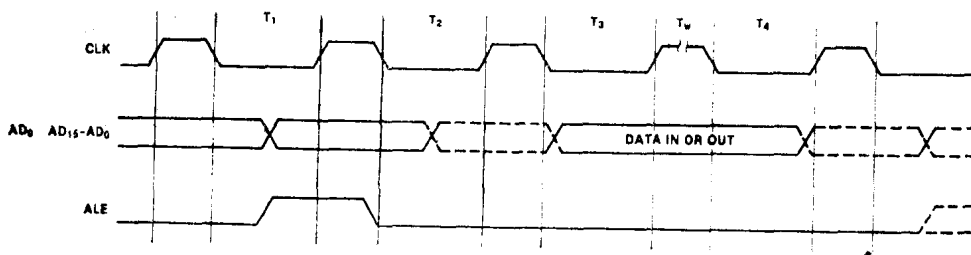


图2B1b



时序图2B1 ALE定时

为一个例子,假定由奇数编址存储器单元(定位在16位数据总线的高位)向CL寄存器(CX寄

存器的低半部)传送字节。数据通过数据总线的高八位向8086传送,并自动重新指向8086内部十六位数据通道的低位,存入CL寄存器。用AL寄存器进行字节输入/输出(I/O)传送时,这种能力可使其既能指向与十六位数据总线高位相连接的,也能指向与它的低位相连接的输入/输出(I/O)设备。

为了存取偶数编址的十六位字(最低有效字节在偶数字节地址的两个相邻字节),A19-A1在每个体中选择适当的字节,而A0和BHE(低电平有效)同时使能两个体(图2B3c)。为了存取奇数编址的十六位字(图2B3d),最低有效字节(由A19-A1确定地址)首先通过总线的高八位(奇数编址字节,高位体, BHE 低电平有效及 A0 = 1)传送。最高有效字节的存取通过对地址(A19-A0)增量来实现,使A19-A1对下一个物理字的单元寻址(记住, A0曾等于1,表明以奇数字节界面定位的字)。然后执行第二个总线周期,完成用低位体(这时 A0 是低电平有效及 BHE 为高电平)进行的最高有效字节传送。对奇数地址进行字传送时,这一过程由8086自动执行。8086内部十六位寄存器的高位和低位字节对数据总线适当部位的指向也是由8086自动进行的,这对程序设计者来说是透明的。

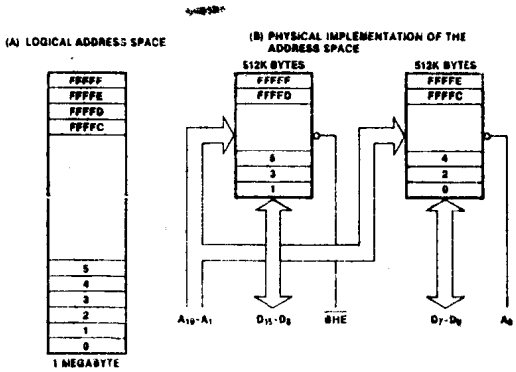


图2B2 8086存储器

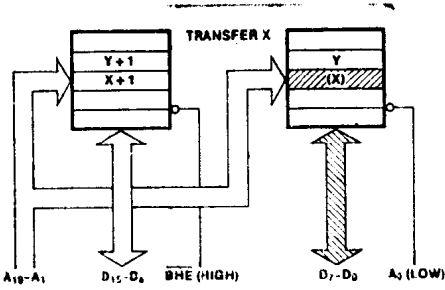


图2B3a 偶数编址字节传送

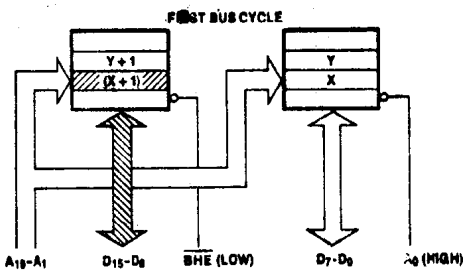


图2B3b 奇数编址字节传送

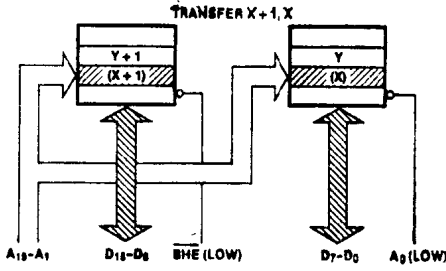


图2B3c 偶数编址字节传送

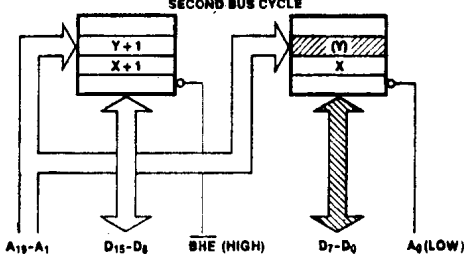
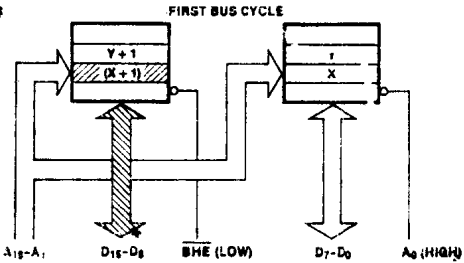


图2B3d 奇数编址字传送

在字节读过程中,即使数据只要求通过数据总线的高八位线或低八位线,CPU使整个十六位数据总线浮动。后面将说明,这样处理能简化对只读器件(ROM,EPROM)的片选译码要求。在字节写操作期间,8086驱动整个十六位数据总线。数据总线上不传送数据的八位线的信息是不确定的。这些概念也应用于输入/输出(I/O)地址空间。输入/输出(I/O)和存储器接口的具体例子将在相应的章节说明。

2C. 系统数据总线概念

对待系统数据总线,必须考虑两个实施方案,即多路复用地址/数据总线(图2C1a)和用收发器从多路复用总线得到的缓冲数据总线(图2C1b)

如果存储器或输入/输出(I/O)器件与多路复用总线直接连接,则设计者必须保证,在T1期间总线上的地址不受器件的干扰。

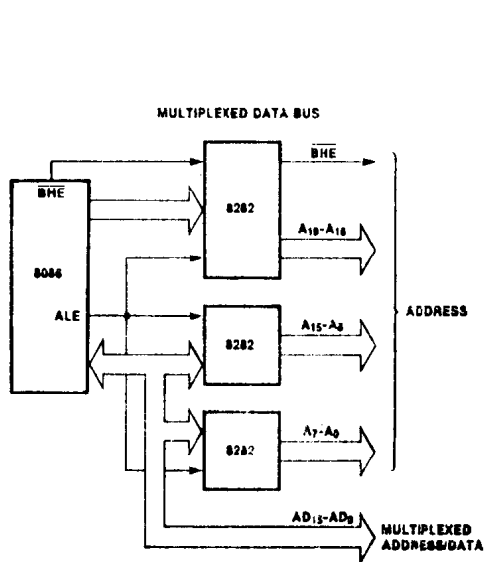


图2C1a 多路复用数据总线

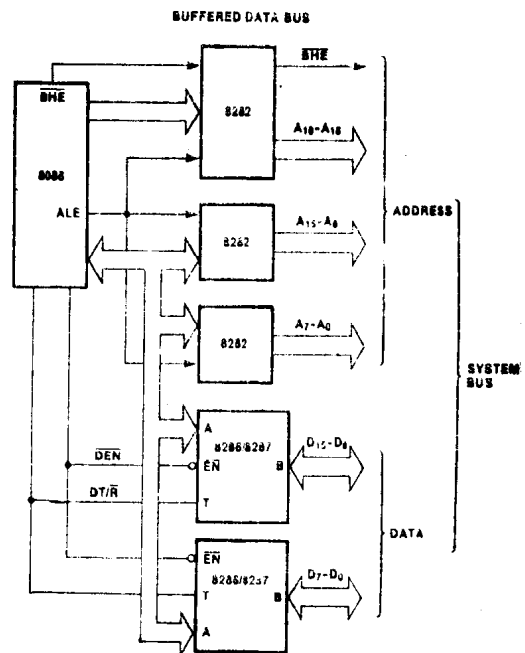
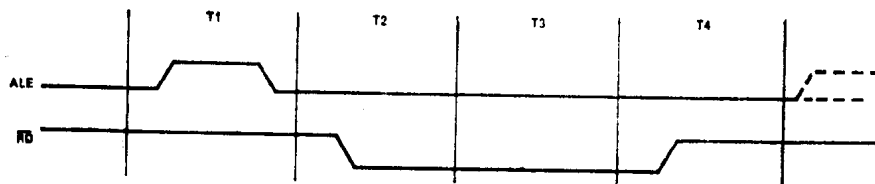


图2C1b 缓冲数据总线



时序图2C1 ALE与READ的关系

为了避免这一点,器件的输出驱动器不应被器件的片选使能,而应有一个被系统读信号控制的输出使能(图2C2)。8086时序保证读信号在地址被锁存之后才有效(图2C1)。所有Intel外围设备,用于微处理器的EPROM和RAM产品,都有输出使能或读输入以便与多路复用总线接通。

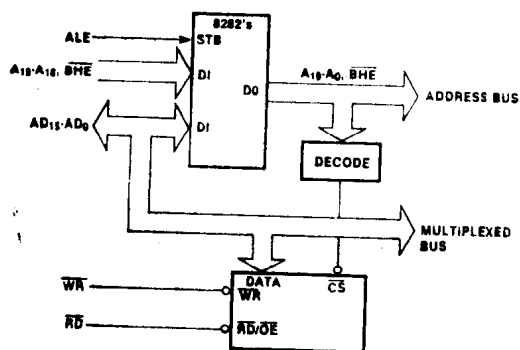


图2C2 在多路复用总线上
具有输出使能的设备

多路复用总线其次一个限制是8086的驱动能力为2.0mA，而容性负载能力为100pF，这才能保证规定的交流特性。假设输入/输出(I/O)设备的容性负载为20pF，地址锁存的为12pF，而存储器设备的为5—12pF，则由三个外围设备和两到四个存储器设备组成的系统，就接近负载极限。

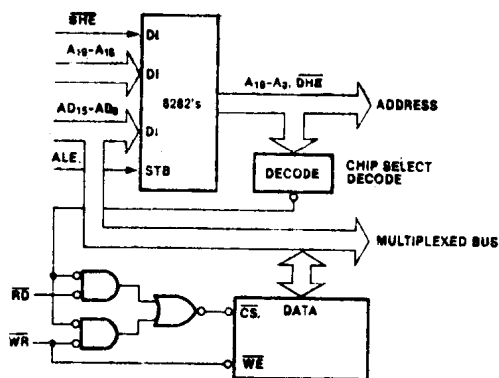
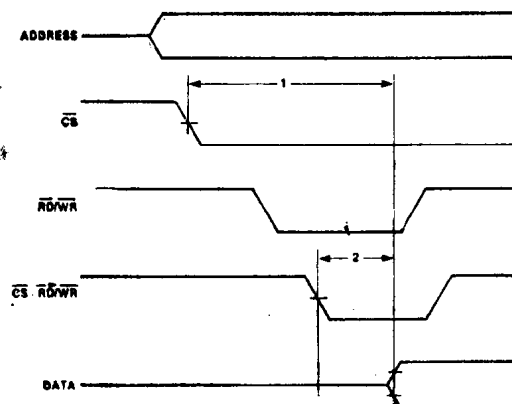


图2C3 在多路复用总线上不具有
输出使能的器件

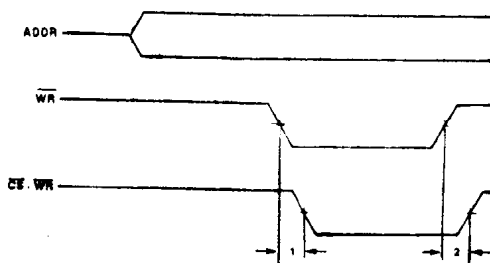
为了满足更大系统容性负载和驱动要求，数据总线必须缓冲。8286(非反相)和8287(反相)八位收发器作为8086系列的一个部件提供，用以满足这一要求。它们有三态输出缓冲器，在总线接口能驱动32mA，在CPU接口能驱动10mA，还能以22ns(8287)或30ns(8286)的开关时间，在总线接口带动300pF的容性负载，在CPU接口带动100pF的容性负载。为了使能和控制收发器的方向，8086系统提供数据使能信号(DEN)和数据发送/接收(DT/R)信号(图2C1b)。这些信号提供适当的时序，以保证多路复用总线



1. 由地址译码产生片选的选取时间

2. 用RD/WR选通片选的选取时间

时序图2C2 用RD/WR选通CS的选取时间



1. CS在写之前不确立而在一或两个门
延迟之后成为有效

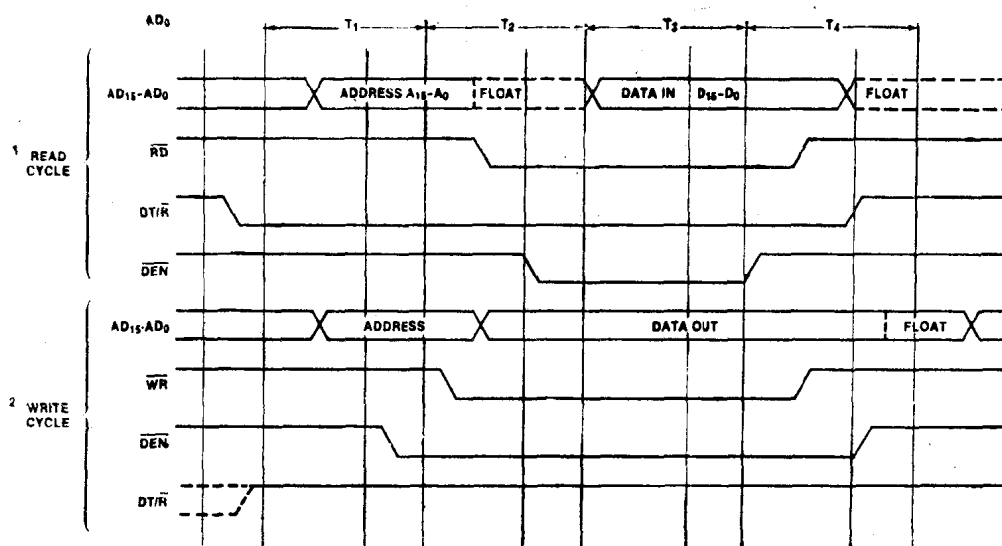
2. CS在写之后一或两个门延迟保持有
效

时序图2C3 CS对WR建立和保持

与系统在T₁期间内隔离，並消除CPU在读和写期间的总线争用（时序图2c4）。即使存储器
和外围器件与CPU隔离（图2c4），如果器件除片选外没有输出使能控制，总线争用问题仍
然存在。例如，在由一个片选到另一个片选过渡期间会出现总线争用（新选定的器件在先前的
器件完成其驱动器的禁止之前开始驱动总线）。在写周期期间存在着另一个更严重的情
况。如果一个器件的输出只由片选控制，在片选到写有效期间，它将与由CPU通过收发器驱
动的写数据同时驱动总线（时序图2c5）。在这里解决这些多路复用总线问题可采用同样的方
法，它们也受到相同的限制。

总线实施的最后一个扩充是第二级缓冲，它能减小器件对系统总线的总负载（图2c5）。
这对多板系统和隔离存储器陈列是典型的实施方法。这种结构所关心的问题是存取的附加延
迟，但更重要的是与系统总线接口器件有联系的第二个收发器的控制。图2c6示
出几种控制收发器的方法。第一种方法（图2c6a）就是直接把DEN和DT/R信号分布在整
个系统中。DT/R反相以便对第二级收发器提供正确的方向控制。第二个方法（图2c6b）是
用输出使能对器件提供控制信号。RD信号用于引导由系统总线通向外围设备的数据。每当
局部总线上的器件被片选时，就要选定缓冲器。在读取过程中，在器件局部总线上可能会出
现总线争用，这是因为读操作同时使能器件输出，並改变收发器的方向。在读操作结束时也
会出现总线争用问题。

如果对器件片选以读和写为条件，则对于没有输出使能的器件可应用同样的方法。用读/
写控制片选能防止器件在命令接收之前驱动收发器。采用这种方法的限制是选取被限定到读/
写时间，而片选被限定到写建立和保持时间。



1. 在8086浮动多路复用总线后，使能DEN
2. DEN在该周期早期使能收发器，但DT/R保证收发器处于发送方式而不是接收方
式，不驱动CPU

时序图2c4 总线收发器控制

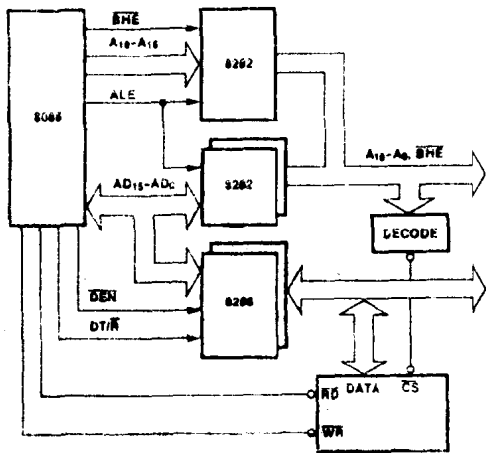
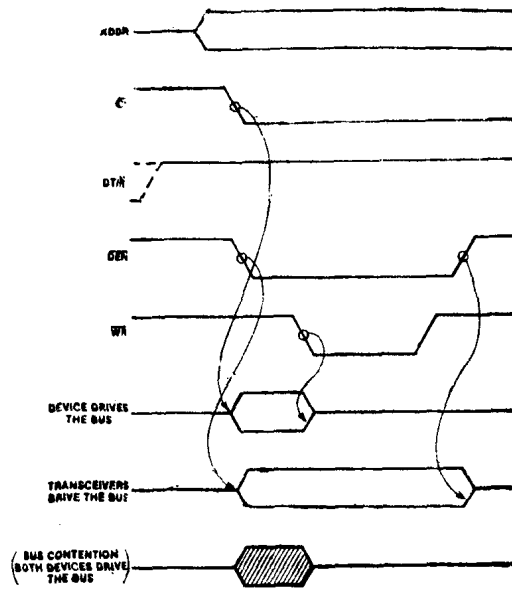


图2c4 在系统总线上具有输出使能的器件



时序图2c5

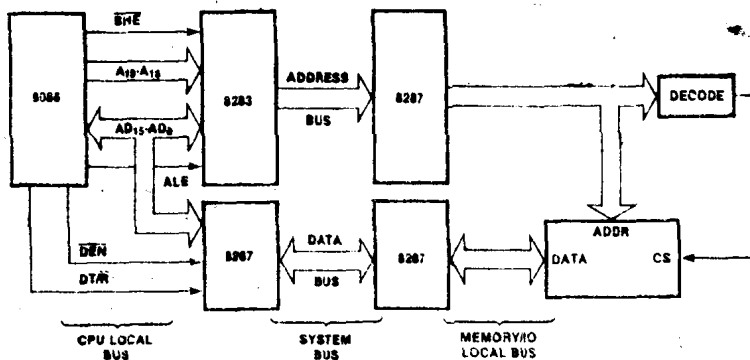


图2c5 全缓冲系统

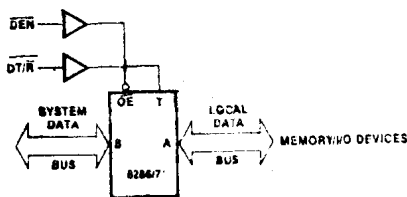


图2c6a 用DEN和DT/R控制系统收发器

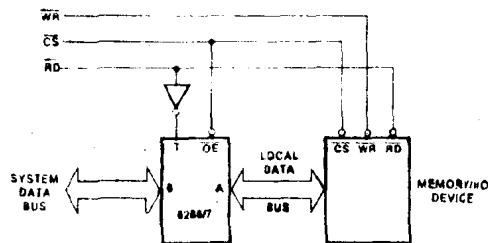


图2c6b 用OE/RD缓冲器件

对于具有和不具有输出使能的器件，图2c6d示出了另一种可采用的方法。 $\overline{RD}$ 还用以控制收发器的方向，但是直到命令和片选有效之前，它不被使能。总线争用的可能性仍然存在，但被减小到使能相对收发器方向改变时间的变化。可以利用由片选得到的完全的选取时间，但是数据在写之前未确定，而只是在写之后并经过收发器禁止延迟之后才保持有效。

最后一种方法是具有分离的输入和输出器件。它采用分离的总线接收器和驱动器，而不

采用单个收发器。接收器总是处于选通状态，而总线驱动器则由 $\overline{RD}$ 和片选控制。在这种系统中只有当局部总线的每根线上的多重器件，在片选改变期间使能和禁止时，在可能出现总线争用现象。

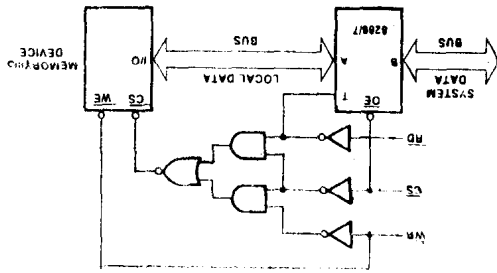


图2c6c 不用 $\overline{OE}/\overline{RD}$ 而用共同的或分离的输入/输出缓冲器件

在本说明中多路复用总线看作是局部CPU总线，而分离地址和缓冲数据总线看作是系统总线。关于总线争用以及与此有关的其它系统问题，可参见附录1。

2D. 多处理器环境

8086体系构造以共用系统总线的概念支持多处理器系统（图2D1）。系统中所有的CPU通过系统总线相互通信，并共享资源。系统总线可以是Intel多总线（Multibus™）系统总线，也可以是上节定义的系统总线的扩充。对分离系统总线所要求的主要补充是控制对系统总线使用的仲裁逻辑。因为每个CPU对共用总线的使用要求是异步的，仲裁逻辑解决优先权问题，并承认有最高优先级的CPU对总线的使用。在获得总线使用权之后，CPU进行传送，而后让出总线，或者等待被迫让出总线。关于多总线仲裁技术，可参考AP—28A，Intel多总线（Multibus™）接口。

为了支持8086系列多总线系统总线的多主接口8289总线仲裁器也是该系列的一个部件。8289与8086的局部总线兼容，并与8288总线控制器一起，实现对总线仲裁的多总线约定。8289提供各种仲裁技术和优先化技术，以便使总线利用率、吞吐量和共用资源的使用达到最优。附加特性（通过搭接选择）扩充了结构选择，它不仅包括为使用共用资源的对多主系统总线的单纯CPU接口，还包括对于专用资源的局部CPU总线的支持。关于8289的具体结构和其它资料参见应用说明AP—51。

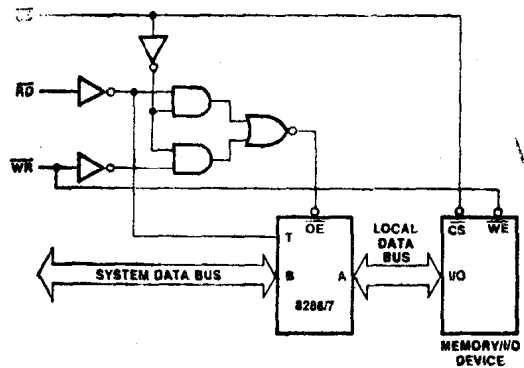


图2c6d 不用 $\overline{OE}/\overline{RD}$ 而用公共的或分离的输入/输出缓冲器件

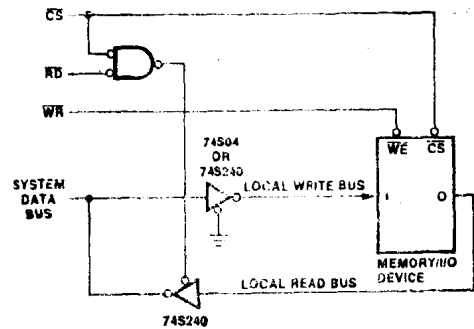


图2c6e 不用 $\overline{OE}/\overline{RD}$ 而用分离的输入/输出缓冲器件

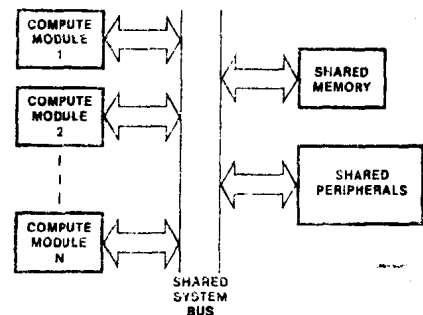


图2D1 8086系列多处理器系统

3. 8086系统细节

3A. 工作模式

8086无与伦比的特点是它选择最适合应用的基本机器配置的能力。8086MN/ $\overline{MX}$ 输入搭接选择允许设计者在8086输出的两种功能定义之间进行选择。

最小模式

8086的最小模式(图3A1)是一个最优化的小到中型(一到两块板)的单CPU系统。它的系统体系结构用以满足高性能十六位应用的低到中档要求。CPU保持了1兆字节存储器空间,64K字节的I/O空间和十六位数据通道。CPU直接提供全部总线控制(DT/ $\overline{R}$ 、 $\overline{DEN}$ 、ALE、M/ $\overline{IO}$),命令($\overline{RD}$ 、 $\overline{WR}$ 、 $\overline{INTA}$)和一个简单的CPU先占机构(HOLD、HLDA),它现有的DMA控制器兼容。

最大模式

最大模式(图3A2)扩充了系统体系结构以支持多处理器配置和局部指令集合扩充处理器(协处理器)。通过增加8288双极型总线控制器,8086在最小模式中对总线控制和命令的输出被重新规定,以实现这些扩充并增强系统性能。具体地说,(1)处理器先占机构($\overline{RQ}/\overline{GT0}$ 、 $\overline{RQ}/\overline{GT1}$)的两个优先权级别允许多个处理器驻留在8086局部总线上,并共享对系统总线的接口,(2)排队状态(QS0、QS1)能使外部设备如ICE^M-86或专用指令集合扩充协处理器,跟踪CPU指令的执行,(3)在多处理器系统中,共享设备的使用由硬件总线锁定机构支持,(4)系统命令和配置选择借助辅助设备如8288总线控制器和8289总线仲裁器得以扩充。

排队机构指明是什么信息正从内部排队机构移走和什么时候排队机构由于控制转移(表3A1)而复位。通过监视用于进入8086指令的状态线S0、S1、S2(1.0.0指明代码存取,A0和BHE指明字或字节)和用于离开8086内部排队机构的QS0、QS1,能够跟踪指令执行。由于指令是从8086的排队机构被执行,所以排队状态在每个CPU时钟周期都存在,并与总线周期活动无关。这个机构一方面使协处理器检测换码(ESCAPE)指令的执行,该指令引导协处理器完成一个特定任务,另一方面允许ICE-86捕获特定存储器单元的执行。图3A3示出使用的线路例子,第一个可逆计数器跟踪排队机构的深度,而第二个计数器则根据匹配条件将排队深度归零。第二个计数器在再次从排队机构中取出指令时减量,直到排队机构填满或计数器变为零,这表明匹配地址的执行。第一个计数器从排队机构中取出指令时减量(QS0=1),而在代码选入排队机构时增量。注意到正常的代码取出将两个字节传送到排队机构中去,这样,除非有单个字节通过总线的高位(A0—P是高)装入,一般给计数器两个时钟增量(T201和T301)。由于执行部件(EU)与总线接口部件(BIU)不同步,则从排队机构中取出并向排队机构中传送能同时进行。第一个计数器的异或门驱动ENP输入,使这两种同时操作互相抵消,但不改动排队机构深度。

表3A1 排队状态

QS ₁	QS ₀	
0(低)	0	无操作
0	1	从排队机构中送出操作码第一个字节
1(高)	0	排队机构已空
1	1	从排队机构中送出后面的字节

排队状态在 CLK周期 确立，然后完成排队操作。

为了处理共享资源的使用控制问题，8086最大模式提供了一个硬件 LOCK 输出。通过指令流由执行 LOCK 前缀指令启动 LOCK 输出。LOCK 输出在前缀指令执行后的第一个 CPU 时钟周期变为有效，并一直保持到跟在 LOCK 前缀指令后的指令的最后一个时钟周期的结束。为了在多处理器系统中提供总线使用控制，LOCK 信号应该结合到驻留在 CPU 中的系统总线仲裁逻辑中。

在正常的多处理器系统操作过程中，共用系统总线优先权由仲裁逻辑在一个时钟周期内确定。当每个 CPU 要求通过系统总线传送时，通过它的驻留总线仲裁逻辑

请求对总线的使用。当 CPU 得到优先权时（由系统总线仲裁电路和一些有关的逻辑确定）开始控制总线，执行其总线周期，然后保留总线控制直到自动地释放总线，或者由于失去优先权而被迫脱离总线为止。封锁机构能防止 CPU 失掉总线控制（或者自动地或者被迫地），并保证 CPU 在不受另一个 CPU 干予和可能破坏数据的情况下执行多总线周期的能力（在执行被封锁指令时）。这种机构的一种典型使用是“TEST 和 SET 信号灯”，在此期间 CPU 必须从共享存储器单元读取数据，然后将数据送回该单元，不允许另一个 CPU 在 TEST 操作（读）和 SET 操作（写）之间访问同一个单元。这在 8086 中是用一个被封锁的交换指令来完成的。

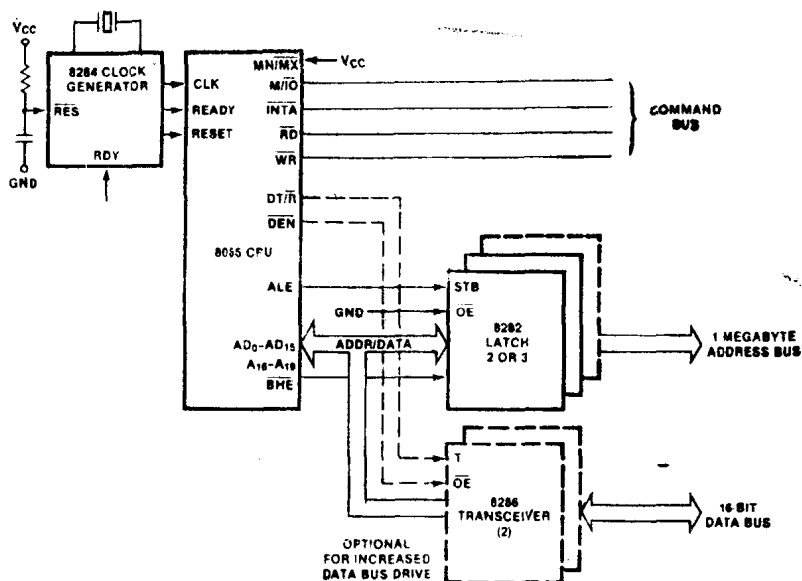


图3A1 最小模式8086

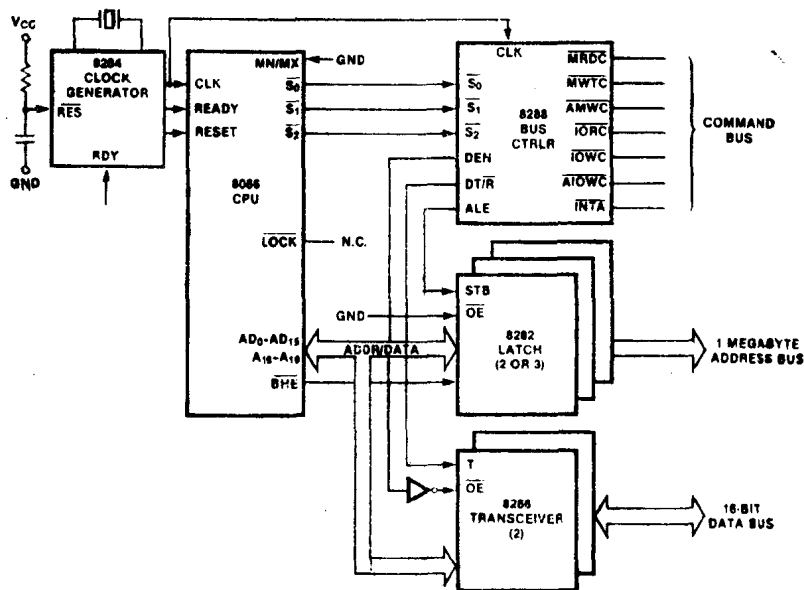


图3A2 最大模式8086

; address of the semaphore

LOCK输出作用于时序图3A1中。多处理器系统LOCK的另一个用处是分区封锁传送, 它可以实现从一个CPU信息缓冲器向另一个快速传送。

在封锁指令期间,对处理器先占($\overline{\text{RQ}}/\overline{\text{GT}}$)的请求进行记录,但在封锁指令完成之前不响应。 LOCK 对中断没有直接影响,例如,封锁的 HALT 指令会使 HOLD (或 RQ/GT)请求被忽略,但是在中断时将使CPU退出 HALT 状态。一般说来,前缀字节可以认为是它后面指令的扩充。因此,直到前缀后面的指令完成之后(除了在执行像 HALT 、 WAIT 和重复的串数据这些允许服务中断的指令期间),执行前缀时出现的中断才被响应。要注意,在指令前可能是多重前缀字节。作为另外一个例子,假定一个“串数据”处在前缀(REP)之后,在每次执行“串数据”之后,它是可以中断的。这一点即使在 REP 前缀与 LOCK 前缀结合时也是成立的,以防止中断在数据块传送或其它重复的串操作时被封闭。只要操作未被中断, LOCK 保持有效。关于具有多重前缀的中断串操作的更多情况,将在与8086中断结构有关的部分中提供。

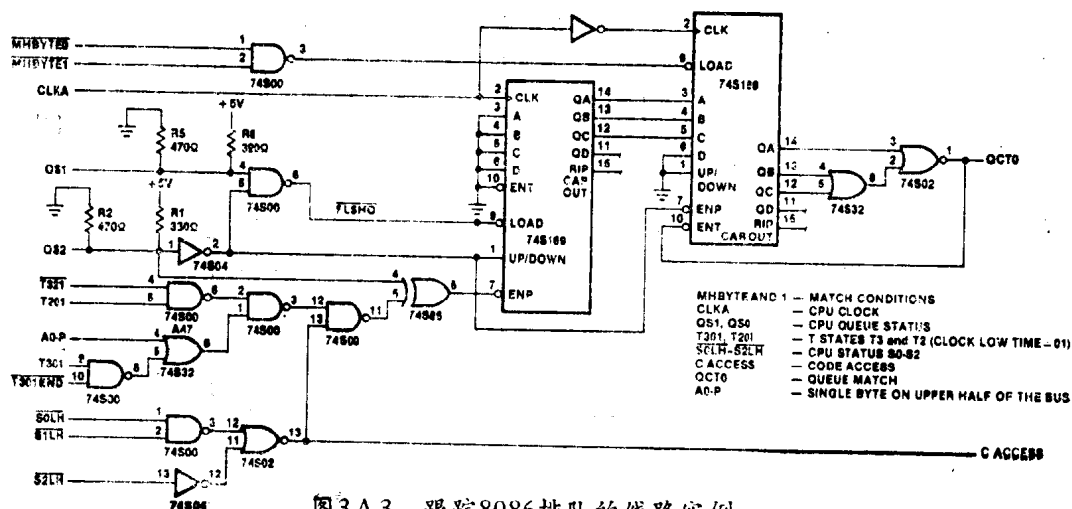


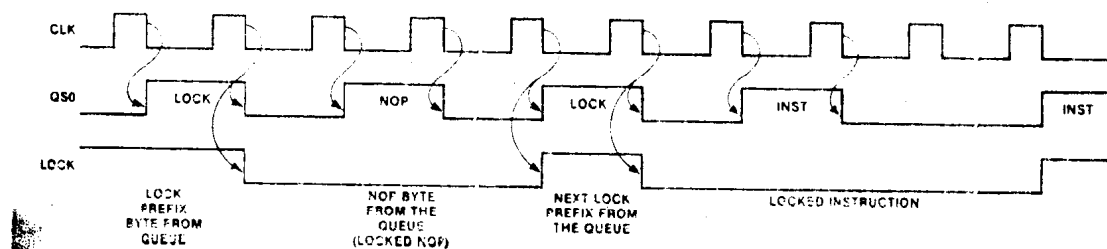
图3A3 跟踪8086排队的线路实例

规定三根附加状态线 ($\overline{S0}$ 、 $\overline{S1}$ 、 $\overline{S2}$) 提供与8288和8289的通讯。状态线告诉8288何时开始总线周期,产生何种形式的命令和何时结束总线周期。8288在每个CPU时钟信号(CLK)的开始对状态线采样。为了开始总线周期,CPU将状态线从无效状态 ($\overline{S0}$ 、 $\overline{S1}$ 、 $\overline{S2} = 1$) 激励成七个可能的命令编码(表3A2)中的一个。这一动作发生在前一个总线周期的T4或T1(空闲周期,无现行总线活动)信号的上升边。8288在每个时钟周期由高向低过渡时对状态线采样,藉此检测状态变化。8288在状态变化检测后立即出现的时钟周期(T1)内,产生ALE和适当的缓冲器方向控制,藉此开始一个总线周期。总线收发器和选定的命令在下一个时钟周期(T2)(正常的写命令为T3)被使能。如时序图3A2所示,当状态回到无效状态时,8288将结束命令。由于CPU直到接收到“准备好”指示时才使状态回到无效状态,8288将在任意数目的等待周期内,保持有效命令和总线控制。状态线还可被8086局部总线上的其它处理器使用,得到局部总线的控制时,可以监视总线活动和控制8288。

表 3A2

$\overline{S_2}$	$\overline{S_1}$	$\overline{S_0}$	
0 (低)	0	0	中断响应
0	0	1	读I/O口通道
0	1	0	写I/O口通道
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	无作用

8288提供取自CPU的总线控制 ($\overline{DEN}$ 、 $\overline{DT/R}$ 、 $\overline{ALE}$) 和命令 ($\overline{INTA}$ 、 $\overline{MRDC}$ 、 $\overline{IORC}$ 、 $\overline{MWTC}$ 、 $\overline{AMWC}$ 、 $\overline{IOWC}$ 、 $\overline{AIOWC}$)。命令结构有对存储器和I/O设备分离的读和写命令,以便能与多总线命令结构相容。



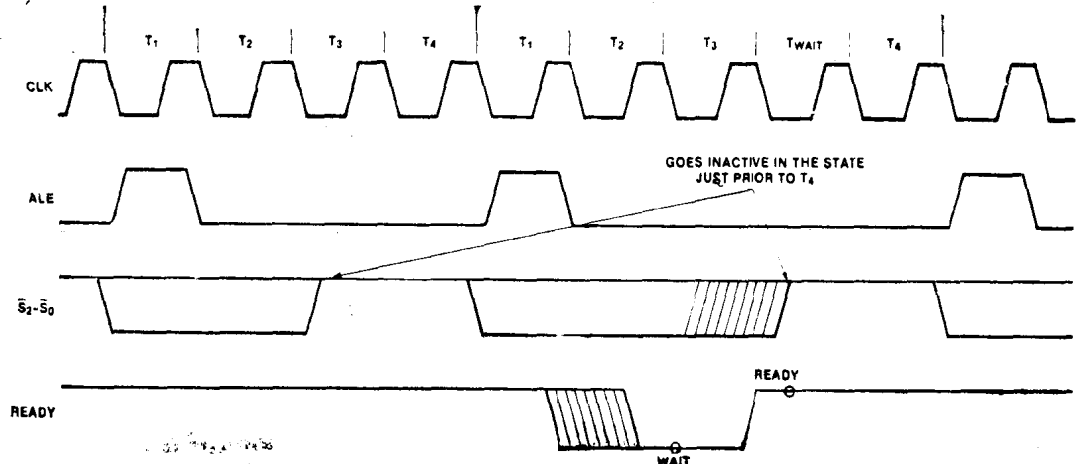
1. 排队状态指明排队操作码的第一个字节。
2. $\overline{LOCK}$ 输出在分离的封锁指令间变为无效。
3. 封锁前缀的译码和 $\overline{LOCK}$ 信号的作用要求两个时钟。
4. 由于排队状态反映前一个时钟周期中排队操作, $\overline{LOCK}$ 输出实际上与下一个指令开始同时成为有效, 并在此指令后一个时钟周期保持有效。
5. 如果跟在封锁前缀后的指令不在排队机构中, 则 $\overline{LOCK}$ 输出仍如图所示, 在取指时变为有效。
6. 在执行被封锁指令期间BIU仍完成指令取周期。在封锁指令期间, $\overline{LOCK}$ 仅在CPU完成的那些总线周期, 封锁到这一CPU的总线。

时序图3A1 8086封锁活动

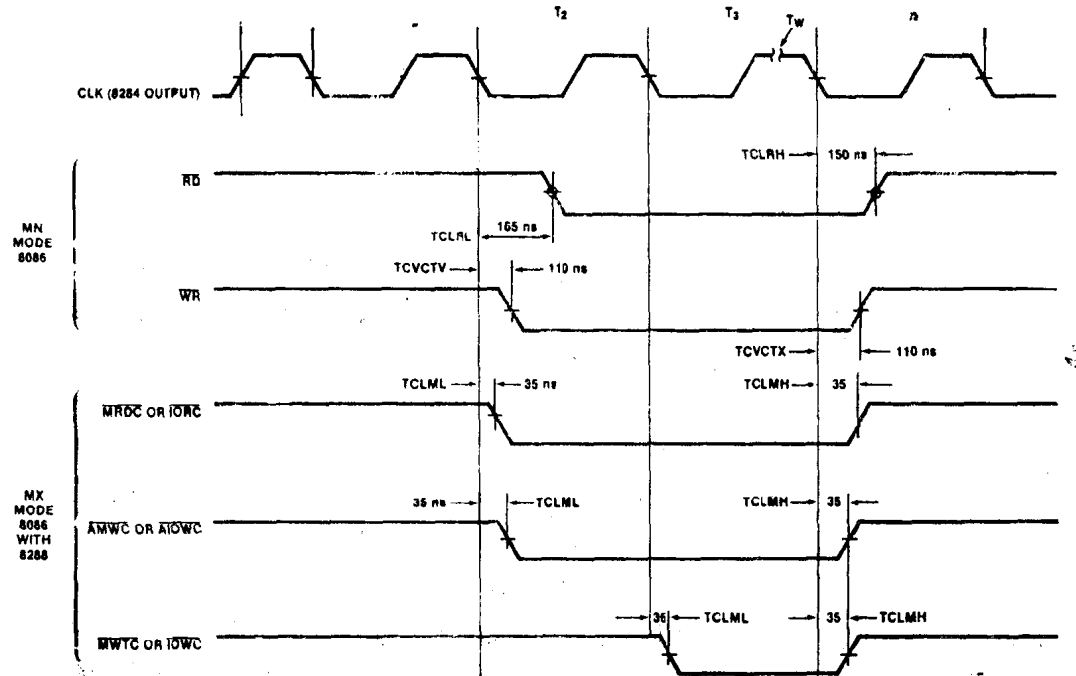
前置写命令比正常写操作早一个时钟周期被使能, 以适应外围设备和静态RAM要求的更宽的写脉冲宽度。正常的写操作是在该操作之前确立数据, 以适应动态RAM存储器和I/O设备的要求, 它们用写信号的前沿选通数据。前置写命令不保证在命令的前沿之前数据是真实的。为了按照所允许的 $\overline{DEN}$ 同其它信号的逻辑连接来扩充收发器控制, 与最小模式的差别是在最大模式中 $\overline{DEN}$ 信号反相。虽然在基本最大模式结构中看起来没有多大好处, 中断控制的引入和各种系统结构将表明改进 $\overline{DEN}$ 的用处。时序图3A3比较了最小模式和最大模式总线传送命令的时序信号。虽然最大模式结构是为多处理器环境设计的, 但是单CPU的大系统设计方案 (或多总线系统或多于两个PC插件板的系统), 也应当使用最大模式。由于

8288是一个双极性专用控制器件，因此它产生的命令输出驱动（32mA）和AC特性容限（定时参数和更坏情况的延迟），比最小模式的8086提供更好的大系统特性。

除了承担由CPU分来的功能外，8288 还提供附加的搭接选择和控制，以支持多处理器结构和CPU局部总线上的外围设备。这些能力允许按共用的（用于多总线系统总线）或专用的（只被该CPU存取）来分配资源（存储器或I/O），以减少对多总线系统总线使用的争用，並改进多CPU系统性能。具体配置方案在AP—51中讨论。



时序图3A2 状态线的激活和终止



时序图3A3 8086最小和最大模式命令时序图

3B. 时钟信号的产生

8086要求时钟信号在低电平和电平之间具有快速的上升和下降时间（最大10ns），低电平为-0.5伏到+0.6伏，高电平为3.9伏到VCC+1.0伏。8086的最大时钟频率是5MHz、对于8086-2是8MHz。由于8086的设计中包含了动态单元，所以为保持机器的状态，要求最小时钟频率为2MHz。由于最小频率要求，CPU的单步和周期操作就可以不用禁止时钟