

'84 INTEL

微型计算机系统器件手册

(四)

上海交通大学 微机研究所
科技交流室

1986.12

'84 INTEL 微型计算机系统器件手册(四)

目 录

二、外围设备支持器件	(6-247)
应用篇	
用8256设计	(6-247)
数据图表	
8231A 算术处理部件	(6-320)
8253/8253-5可编程计数定时器.....	(6-333)
8254可编程计数定时器	(6-346)
8255A/8255A-5可编程序的外围接口	(6-364)
8256AH 多功能通用异步接收发送器(MUART).....	(6-387)
8279/8279-5可编程序键盘显示器接口.....	(6-414)
82285 I/O协处理器时钟发生器和READY 接口.....	(6-431)
三、磁盘控制器	(6-438)
应用篇	
一个采用 8272 的智能数据库系统.....	(6-438)
磁盘子系统的软件设计和运行	(6-476)
数据图表	
8271/8271-6可编程序软盘控制器.....	(6-523)
8272A 单/双密度软盘控制器.....	(6-559)
四、硬盘控制器	(6-586)
数据图表	
82062温彻斯特磁盘控制器.....	(6-586)
微处理器外设——通用外设接口用户手册	
引言	(6-614)
功能描述	(6-619)
指令系统	(6-640)
单步、编程和电源跌落模式	(6-668)
系统操作	(6-674)

二、外围设备支持器件

应用篇

用8256设计

引言

英特尔 8256MUART 是多功能通用异步接收发送器,它是为用于串行异步通讯而设计的,同时也为并行输入/输出、计时、计数和中断控制提供硬件支持。它的多用设计允许它直接连结到MCS-85, iAPX-86, iAPX-88, iAPX-186 和 iAPX-188微计算机系统,以及单片微型计算机 MCS-48 和 MCS-51 系列。

MUART 中含有的四个通常使用的外部功能是:

- 1)、具有芯片上的波特率产生器,全双工、双缓冲器的串行异步接收发送器;
- 2)、两个八位并行输入/输出端口;
- 3)、五个八位计数/计时器;
- 4)、八级优先级中断控制器;

这本手册可分成两部分。第一部分详细地描述 MUART,包括它的功能,寄存器和引脚。这一部分也描述在MUART和英特尔CPU之间的接口,加上有关程序设计的讨论。第二部分提出了一个应用实例:一个基于MUART的行打印机多路传输器。附录包含关于行打印机的多路传输器的软件清单和一些有用的参考信息。

MUART 的说明

MUART 的逻辑上可以划分为七个部分:微处理器总线接口、命令和状态寄存器、时钟电路、异步串行通讯、并行输入/输出、计时器/事件计数器和中断控制器。这可从图1所示的8256MUART的方框图看到。MUART 的引脚配置如图2所示。

微处理器总线接口

微处理器总线接口是MUART的硬件部分,它允许一个微处理器与 MUART 通讯。它由三态双向数据总线缓冲器,一个地址锁存器,一片选(CS)锁存和总线控制器逻辑组成。为了在40只双列直插式引脚中提供全部的MUART 功能,同时保持直接寄存器寻址,使用一个多路传输地址/数据总线。

地址/数据总线

MUART 包含十六个内部可直接寻址的读/写寄存器。八根地址/数据线中的四根用来产生地址。当使用8位微处理器时,例如MCS-85, MCS-48 和 MCS-51, AD0-AD3 是用来对十六个内部寄存器寻址,而第四根地址数据线(AD4)不用来寻址。对于十六位系统,AD1-AD4是用来为内部数据寄存器产生地址,而AD0 在作为第二个有效低电平芯片选择。

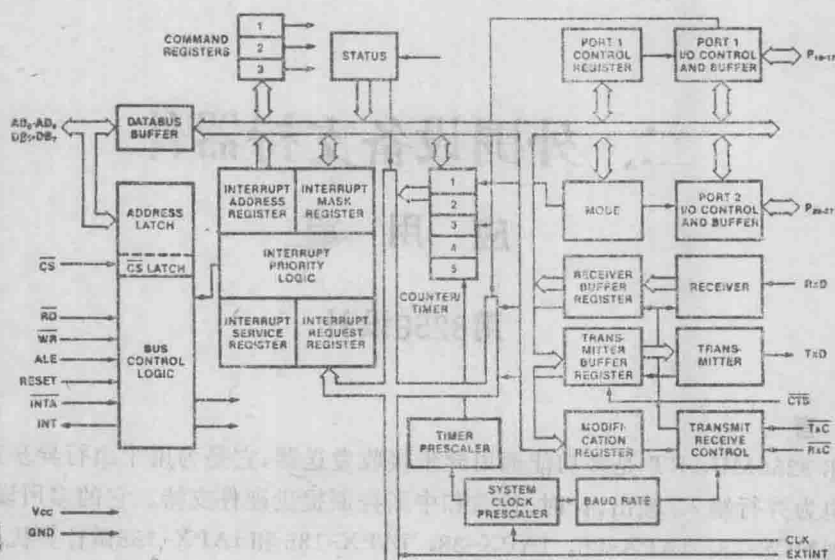


图 1 8256MUART的方框图

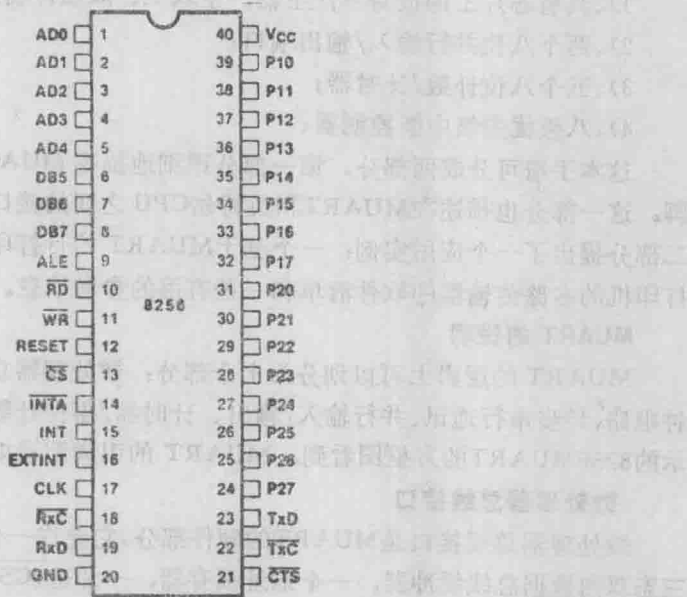


图 2 MUART引脚配置

来自 RD, WR, CS

8256总线接口使用标准总线控制信号，这些控制信号与所有的外部设备和微处理器兼容。其典型的情况是来自一个地址译码器的片选信号(CS)，在ALE的下降沿和地址一起锁存。因此，片选信号在整个总线周期不必保持低电平。然而，数据总线缓冲器将保持三态，除非当片选在低电平已经被锁存时，RD或WR信号变为有效。

INT, INTA

INT 和 INTA 信号用来中断 CPU 并接受 CPU 对中断请求的应答。MUART 根据中断源可指引 CPU 到一合适的服务程序。

RESET

当一高电平出现在 RESET 引脚时, MUART 被置为一已知的初始状态。这个初始态在“硬件复位”过程中描述。

命令和状态寄存器

如图 1 所示, 有三个命令寄存器和一个状态寄存器。三个命令寄存器是读/写寄存器, 而状态寄存器是一个只读寄存器。命令寄存器使 MUART 形成它的操作环境(即, 8 或 16 位 CPU, 系统时钟频率)。其次, 它们控制它的更高级功能, 例如控制 UART, 选择中断控制器的工作方式和为计时器选择基本频率。第三个命令寄存器是 MUART 中唯一的寄存器, 这是一个置位/复位寄存器, 允许程序设计者简单地实现对任何位置位或复位的写入。

状态寄存器提供关于 UART 发送器和接收器状态以及中断引脚状态的全部信息。状态寄存器是 MUART 中只读的唯一寄存器。

时钟电路

五个计时器的时钟和波特率产生器是来自系统时钟。引脚 17 (CLK) 的系统时钟送入一个系统时钟的预定标器, 该预定标器依次送到五个计时器和波特率产生器。MUART 的系统时钟可能与微处理器的时钟异步。

系统时钟预定标器

系统时钟预定标器是一个程序可控的分频器, 它把计时器和波特率产生器的内部时钟频率标准化为 1.024MHz。它用 1, 2, 3 或 5 划分系统时钟 (CLK), 提供 1.024MHz, 2.048MHz, 3.072MHz 或 5.12MHz 的时钟频率。(8085 通常使用的 6.144MHz 晶体频率形成来自 8080 的 CLK 引脚的 3.072MHz 频率。)如果系统时钟不是上述四个频率中的一个, 那末, 波特率产生器和计时器的频率将是不标准的; 然而, 只要系统时钟满足数据图表 tcy 规格, MUART 将仍然运行。

计时器的预定标器

计时器的预定标器允许用户选择所有 MUART 计时器的两种基本计时频率中的一个, 或 1KHz 或 16KHz。经过命令寄存器 0 决定频率选择。

异步串行接口

MUART 的异步串行接口是一个双工双缓冲发送器和接收器, 并具有独立的控制寄存器。使用如图三所示的标准异步格式。MUART 的 UART 部分的操作非常类似 8251A 的 USART 操作。

UART 的接收器部分

串行异步接收器部分包含一个串行移位寄存器, 一个接收缓冲寄存器和接收器控制逻辑。按规定的波特率, 对从 RxD 引脚进入接收移位寄存器的串行输入数据记时。采样实际发生在 RxC 的上升沿, 假定一个外部时钟, 或者内部波特时钟的上升沿。当接收器启动, 但无效时, 接受逻辑将以 32 或 64 次的位速率对 RxD 采样, 并寻找从标志(高电平)到空格(低电平)状态的变化。这个通常作为起动位检索方式来访问。该状态变化出现时, 接

收逻辑等待一位半时间, 然后对 RxD 再采样。如 RxD 仍在空格状态, 则接收逻辑开始计时以便在下一位周期开始接收数据。如 RxD 已返回到标志状态(即, 假起始位), 接收逻辑将返回到起动位检索方式。

通常在每一位的中心, 对已接收的数据采样, 然而, 调节采样位的位置是可能的。可通过改变寄存器来控制这个特性。

串行接受数据的位速率来自内部的波特率产生器或一个外部时钟。使用一个外部时钟时, 程序员有三个采样速率可选: 1X, 32X 或 64X, 如使用内部波特率产生器, 采样速率都是 64X, 除 32X 是 19.2Kbps 外。

当串行移位寄存器计算停止位时间时, 产生一个内部传送脉冲, 它把移位寄存器的内

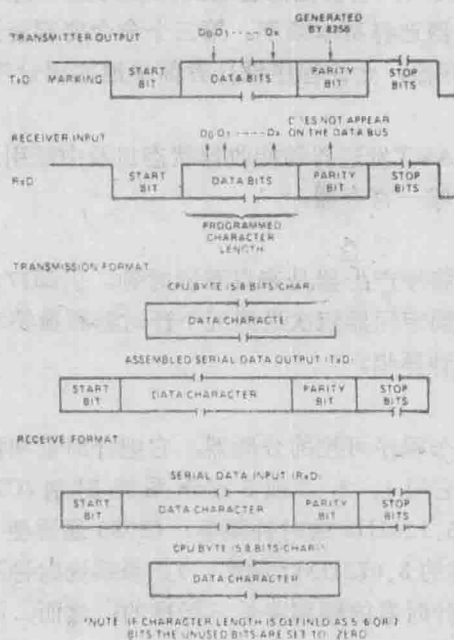


图 3 异步格式

容传送进接收缓冲器。在第一个停止位前半部期间, 发生传送。这个传送脉也触发关联接收部分的其他信号, 包括接收缓冲器满(RBF)、奇偶错(PE)、超界错(OE)和画面错(FE)。当接受缓冲器已经锁存时, 第一个停止位的中间之后, 可修改这个状态位。锁存这四个状态位的每一个。它们在对状态寄存器寻址的第一个读脉冲(RD)的上升沿复位。在“寄存器描述”部分, 给出了状态寄存器的完整说明。

当串行接收器没启用时, (借助于第三个命令寄存器的第六位)禁止传输传送脉冲。结果是不能用移位寄存器的内容装入接收缓冲器, 并且RBF, PE, OE和FE这些在状态寄存器中的位不能修改。即使接收器不能启用, 串行移位寄存器将仍然计算从 RxD 来的数据时间。这意味着接收器将仍然与起始和停止位同步。例如, 如果经过第三个命令寄存器, 在接收一个串行字符的过程中, 启用接收器, 则字符将仍然正确组装。接收器不能启用时, 接收的最后一字符将继续在接收缓冲器内。电源上升时, 接收缓冲器内的值不能确

定。

每当少于8位的一字符长度被编入程序,一个已接收字符的大多数有效位将总是读为零。还有,接收器将只检查一字符的第一个停止位,而不管许多停止位是怎样被编程到设备中。

接收截断的检测(Receive Break Detect)

当RxD保持在间隙状态一个字符时间,包括奇偶位(即使有)和第一个停止位时,则接收截断出现。当MUART接收一截断时,它将把截断状态检测位(BD)置位。在第一个停止位中间之后,对截断状态检测位置位。如MUART检测到一截断,它将禁止接收缓冲器传送脉冲,于是,接收缓冲器将不把零字符装入,并且四个状态位(PE,OE,PE和RBF)将不修改。最后接收的字符保留在接收器中。一个截断状态检测有如不用接收器一样的相同作用——它们都禁止传送脉冲——因此,人们可以把截断状态作为不允许接收器工作来看待。

截断状态检测位被锁存。由对状态寄存器寻址的读脉冲上升沿清除它。如一个截断出现,那末,读出状态寄存器之前,RxD数据线返回到标志状态,在读出BD状态位前,它将保持置位。如RxD在BD状态位真的已读出后返回到标志状态,则不读状态寄存器,BD状态位将自动复位。

MUART的接收截断检测逻辑不取决于接收器是启用还是没有启用。因此,即使接收器没有启用,MUART也会识别截断。在一截断后,RxD线返回到标志状态时,8256将处于起动位搜索方式中。

如果接收器中断级启用,不管接收器是否被启用,打断会产生一截断请求。在RxD引脚返回到标志状态前,另一个接收中断不会产生。

UART的发送器部分

MUART的串行异步发送器部分由一个发送缓冲器、一发送(移位)寄存器和有关的控制逻辑组成。在状态寄存器中有两位表示发送缓冲器的状态和发送寄存器的状态:TBE(发送缓冲器空)和TRE(发送寄存器空)。

为了发送一个字符,把一个字节写入发送缓冲器。当TBE=1时,发送缓冲器只应该写入。当发送寄存器是空着和CTS=0时,字符将自动地从发送缓冲器传送到发送寄存器。在起动位的传输期间,发生从发送缓冲器到发送寄存器的数据传输。这个传输发生后,有时,在第一个数据位传输的开始,把TBE置为1。

当发送器空转时,TBE和TRE二者都将置为1。一字符写到发送缓冲器后,则TBE=0和TRE=1。该状态将保持一短周期的时间。然后,字符将传送到发送寄存器,并将读状态位,TBE=1和TRE=0。在TBE=0和TRE=0后,在这一时刻,第二个字符可以写入发送缓冲器。发送寄存器变为空和用发送缓冲器中的字节重新装入前,TBE将不再置为1。

唯一的方法——使用CTS引脚,使发送器不能工作。当CTS=0时,发送器被启用,而当CTS=1时,发送器不工作。如果发送器空转和CTS从0变为1,使发送器不工作,TBE和TRE将继续置位到1。因为TBE=1,一字符可以写入发送缓冲器。字符将存在发送缓冲器,但是,在CTS变为低电平前,它将不被传送到发送寄存器。

在一个字符传送期间,如CTS从低变为高电平,将完成传送中的字符,并TxD将返

回到标志状态。如果发送器满, (即, TBE 和 TRE = 0), 传送移位寄存器将为空, 而传送缓冲器将非空; 所以, TBE = 0 和 TRE = 1。

发送器截断(Break)特性

MUART 有三个发送截断特性: 截断检测、发送截断 (TBRK), 和单字符的截断 (SBRK)。

截断检测——当 MUART 正在发送串行信息和由接收部分强迫传送线到间隙状态时, 截断条件出现。截断通常使用半双工传送, 以便接收器对发送器能发信号。为了检测一个截断, 端口16必须在外部的连接到发送线。如果传送电压比TTL使用的电压拉低得更多, 那末必须提供适当的缓冲, 以便MUART中的端口16将接收正确的极性和电平。

当截断检测启用后, 在一个传送字符的最后一位或唯一的停止位期间, 在内部查询端口16。如果在传送停止位期间, 这个引脚是低电平, 截断状态检测位(BD)将被置位。对截断检测和接收截断检测进行逻辑或运算, 以便BD状态位置位(二者中的任一个都能使这一位置位)。通过中断控制能引起区别。如果发送和接收中断被启用, 一截断(Break in)将产生一个第五级中断即发送中断。而打断(Break)将产生一个四级中断即接收中断。如RxC和TxC用于串行位速率, 则不能检查截断(Break-in)。

发送间断——为了象第三个命令寄存器中的TBRK位置位一样长。它使得TxD引脚被强迫为低电平, 发送间断有效的同时, 从发送缓冲器到发送寄存器的数据传送将被禁止。

如发送缓冲器和发送寄存器两者都装满, 并且发出发送间断命令(第三个命令寄存器, TBRK = 1), 则在发送寄存器中的全部字符被送出, 包括停止位。驱使TxD为低电平, 并且, 在发送间断不启用前(第三个命令寄存器, TBRK = 0), 发送缓冲器中的字符仍然存在。在这时, TxD将在一位时间内为高电平, 然后送出发送缓冲器中的字符。

单字符间断——对于包括起动位、数据位、奇偶位和停止位的一个字符, 这个引起TxD被置为低电平。用这一特性, 用户可发送特定数目的间断字符。

如发送缓冲器和发送寄存器两者都装满, 并发送发送的间断命令(第三个命令寄存器, SBRK = 1), 在发送寄存器中的全部字符, 包括停止位被发送。在发送缓冲器中的字符被发送后, 由两位时间的高电平跟随的一个完整字符时间内, 驱使TxD为低电平。

修改寄存器

为了改变接收器的两个标准功能(起动位检查和采样时间)和起用半双工操作的特殊显示器(发送器状态), 使用寄存器修改。如果RxD已经返回到在起动位中心的标志状态, 不启用起动位检查意味着接收器将不返回到起动位搜索方式。它将简单地处理, 以便装配来自RxD引脚的字符, 不管它接收一个假的起动位或不是启动位。修改寄存器也允许用户定义, 在接收数据位期间内, MUART将进行采样。

并行 I/O

MUART包含已分为两个8位端口的16位并行I/O引脚。这两个并行I/O端口(端口1和端口2)可用于基本的数字I/O, 例如把一位置为高电平或低电平, 或用于二线同步交换的字节传送。对于输入或输出端口1是位可编程序的, 这样, 在端口1中的八位任意组合是可选为输入或输出。端口2是半字节可编程序的, 这意味着较高的或较低的半字节中的四位必定选作输入或输出。对用二线同步交换的字节传送, 当端口1中的两位是用于同步交换

信号时，端口 2 可以输入或输出字节。

端口1中的所有位有变化的功能，不同于I/O端口。象上面已提到的，当用于字节同步交换方式时，端口1中的两位被用于同步交换信号。因此，这两位不能用于通用的I/O。在端口1中的另外六位，如果他们不用作I/O端口，则也有变化的功能。表 1 列出了端口 1 中每一位和它的相应的变化功能。

表 1 端口 1 的控制信号

引脚符号	引脚号码	控 制 功 能	条 件
P10	39	ACK 端口2的控制信号	寄存器方式 P2C2—P2C0=101
P11	38	OBF 8位同步交换输出	
P10	39	STB 端口2的控制信号	寄存器方式 P2C2—P2C0=100
P11	38	IBF 8位同步交换输入	
P12	37	第二个事件计数器 时钟输入	寄存器方式 CT2=1
P13	36	第三个事件计数器 时钟输入	寄存器方式 CT3=1
P14	35	内部波特率产生器 时钟输出	方式字 P2C0—P2C2=111 端口1控制字P14=1 第二个命令寄存器 B3—B0≥3H
P15	34	计时器5的触发输入	寄存器方式 T5C=1
P16	33	截断检测输入	第一个命令寄存器 BRK1=1
P17	32	外部边沿变化敏感的中断输入	第一个命令寄存器 BIT1=1

端口1控制寄存器的若干位选择端口1中的引脚是否为输入或输出。通过关联控制信号的其他可编程序寄存器，在端口1中的引脚被选作控制引脚。把端口1中的一位配置为一控制功能，不考虑在端口1控制寄存器中它的规定如何。如果端口1中的引脚重新规定为控制信号引脚，不论引脚是端口1控制寄存器中的输入或输出，这一规定保持不变的。如端口1中的引脚转换为I/O引脚，它们呈现在端口1控制寄存器已规定的状态。

每一并行I/O端口有一个锁存器和驱动器。当端口在输出方式时，写到端口的数据已锁存，并被驱送在引脚上。在端口再写入之前，已锁存在I/O端口的数据保持不变。读端口，不论这个端口是一输入或输出端口，使在引脚上的状态通向数据总线上，往一输入端口写在引脚上将没有作用，不过如在引脚的方向改变以后，数据存贮在锁存器中，并输

出。向端口1的控制引脚写入有如向一输入引脚写入同样的作用。如端口1中的引脚2, 3, 5和6被用于控制信号, 将读出各自的输出锁存器的内容, 而不是控制信号的状态。如端口1中的引脚0, 1和7用于控制信号, 将读出控制信号的内容。如端口1中的引脚4用作内部波特率的测试输出, 这个时钟信号将通过输出锁存器输出、于是在输出锁存器中的信息将失掉。

双线字节同步交换

为实现双线字节同步交换的输入或输出, 经过方式寄存器, 可对8256编制程序。当为字节同步交换对方式寄存器编制程序时, 端口2用作发送或接受字节, 而引脚P10和 P11是用于两个同步交换控制信号。图4和5表明了双线同步交换输入和输出的框图和计时。

为了建立使用中断的双线同步交换输出, 人们必须首先对方式寄存器编制程序, 然后经过中断屏蔽寄存器允许中断。在双线同步交换中断启用后, 中断将不是马上出现。由 $\overline{\text{ACK}}$ 的上升沿触发中断。有两种方法产生第一个中断。在一中断出现前, 第一个字节必须写入端口2并已全部传送完, 或者, 当 $\overline{\text{ACK}}$ 是低电平时, 启用双线同步交换中断, 然后 $\overline{\text{ACK}}$ 变为高电平。

事件计数器/计时器

MUART 的五个8位可编程序计数器/计时器是二进制的可预置减计数器。时钟信号源决定计时器和计数器之间的差别。一计时器测量绝对时间间隔, 而它的时钟频率是来自MUART的系统时钟。计数器的输入时钟频率来自施加到外部引脚的一脉冲。在这一脉冲钟的上升沿递减计数器。

当计数器/计时器形成为计时器时, 它们的时钟信号源通过两个分频器: 系统时钟预定标器和计时器预定标器。正如以前提到的, 系统时钟预定标器把内部系统时钟标准化到1.024MHz。计时器预定标器接收这一标准化的系统时钟, 并把它分频为1KHz或16KHz, 这要取决于第一个命令寄存器是怎样编制程序的。如果需要更高的计时分辨率, 则时钟频率可通过I/O端口从外部输入。

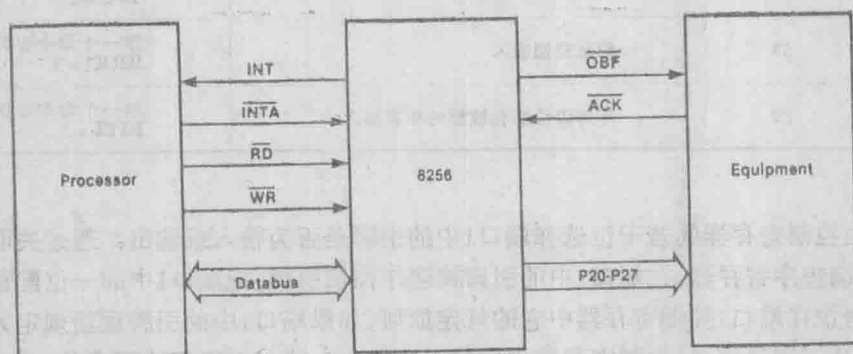


图4 同步交换输出的框图

根据可编程序的方式寄存器, 可级联四个8位计数器/计时器, 形成两个16位计数器。第三和五计数器/计时器可以级联在一起, 第二和四计数器/计时器可级联在一起。第二和三计数器/计时器是低字节, 而第四和五计数器和计时器是级联方式中的较高字节。

每一计数器可用任意的内部值装入。第五计时器是唯一具有特别存储寄存器的计数

器, 这个寄存器保持它的内部值。每当第五计时器用一初始值装入时, 特别存储寄存器也装入这个值。第五计时器可以用端口P15的一个高到低转变重新装入它的初始值。

装入初始值后, 在时钟的第一上升沿递减计数器的内容。当使用一外部时钟时, 装入计数器的形成时间在数据表中规定。使用内部时钟时, 用户无法知道时钟与写脉冲的相位的关系。因此, 计时的精确度是一个时钟周期。

计数器正在不停地计数, 在单个计数器或一对级联计数器达到零的任何时间, 发出一个中断请求。如果该计数器正在被中断使用, 那末程序员应该首先用初始值装入计时器, 然后启动中断。如果程序员首先启用中断, 就可能, 在初始值装入前, 中断将出现。当从任一计时器来的中断出现时, 中断屏蔽寄存器中的相应位自动复位, 防止更多的中断请求出现。

事件计数器/计时器能用于以下操作方式:

第一计时器

——用作一个 8 位计时器

第二事件计数器和计时器

——用作一个 8 位计时器或事件计数器, 或与第四计时器级联, 作为一个 16 位计时器或计数器。

第三计数器/计时器

——用作 8 位计时器或事件计数器, 或第五计时器级联, 作为一 16 位计时器或事件计数器, 与为第五计时器可选择操作的其他方式级联。

第四计时器

——用作 8 位计时器, 或与第二事件计数器/计时器级联, 作为 16 位计时器或事件计数器。

第五计时器

1) 不可重新触发的 8 位计时器。

2) 可重新触发的 8 位计时器, 它的内部值从一个存储寄存器装入, 这个存储寄存器跟在一外部信号的负跳变之后启动。在计数已启动后, 这个信号的以后转变重新装入初始值, 并重新启动计数。

3) 与事件计数器/计时器 3 级联, 是不可重新触发的 16 位计数器, 由两个写操作把一个初始值装入其中。

4) 与事件计数器/计时器 3 级联, 是不可重新触发的 16 位计时器, 由两个写操作把一个初始值装入其中。

5) 与事件计数器/计时器 3 级联, 是可重新触发的 16 位计数器。用来自存储寄存器的初始值将装入最有效的字节(计时器 5), 而最后的有效字节(事件计数器/计时器 3)将自动地置为 FFH, 装入, 启动和重新触发操作象 2) 中的相同模式跟着出现。

6) 与事件计数器/计时器 3 级联, 是可重新触发的 16 位计数器。用来自存储寄存器的初始值将装入最有效的字节(计时器 5), 而最后的有效字(事件计数器/计时器 3)将自动置为 0FFH。装入, 启动和重新触发操作象 2) 中的相同模式跟着出现。

① 8256 在最后的字符已经被接收和输出锁存, 已经再次为空转时发生 INT 信号。

② 于是, 微处理器把下面的数据传送到 8256。

③WR 的上升沿把数据锁存进端口 2(P20,P27)并且“输出缓冲器满”(OBF)置位, 这表明一个新的数据已获得。

④设备用 $\overline{\text{ACK}}$ 的下降沿识别, $\overline{\text{ACK}}$ 的下降沿识别 $\overline{\text{OBF}}$ 。

⑤于是, 8256释放 $\overline{\text{OBF}}$ 。

⑥设备用 $\overline{\text{ACK}}$ 的上升沿识别数据传送, $\overline{\text{ACK}}$ 的上升沿使8256置位 INT。

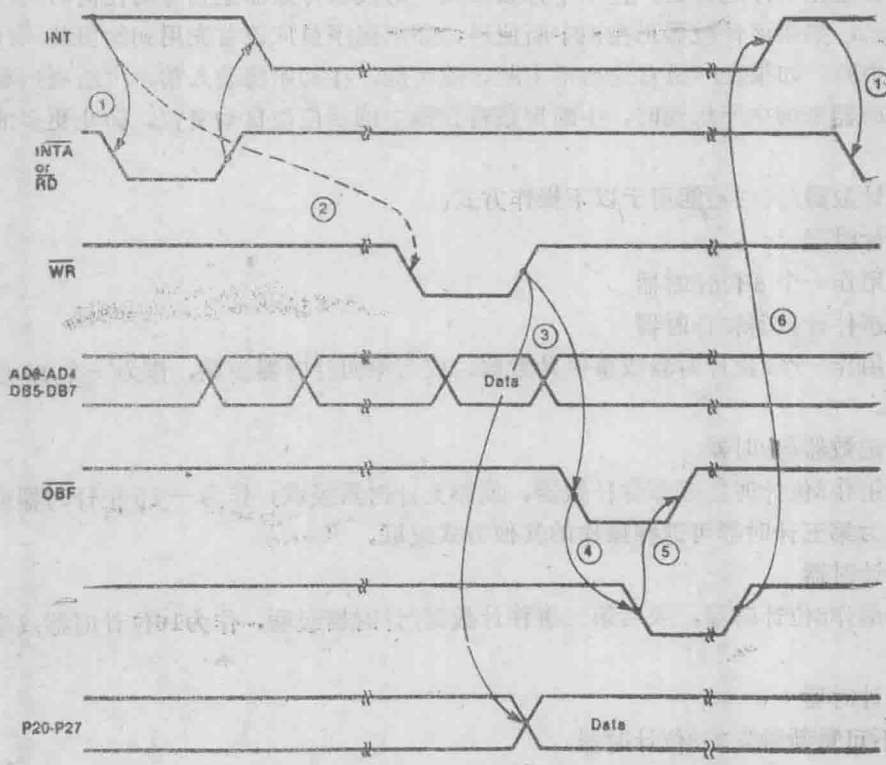


图 4a 同步交换输出的计时

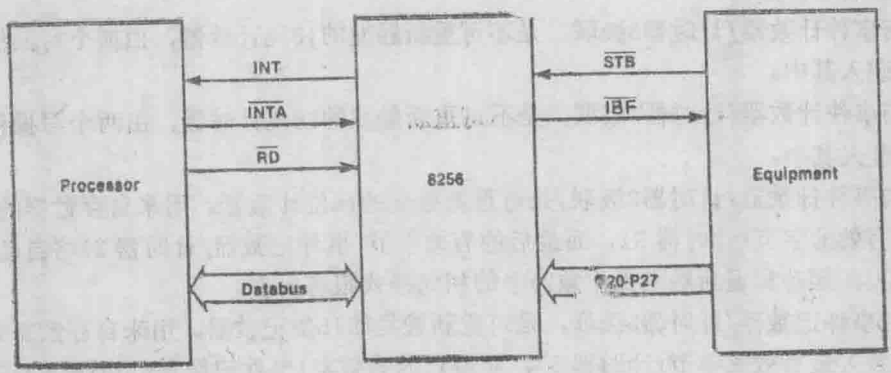


图 5 同步交换输入的框图

中断控制器

微型计算机系统中, 有几种CPU 识别某一外部设备需要服务的方法。两个最普通的方法

法是查询法和中断服务法。

在查询法中，CPU读出每一外部设备的状态，以便确定，是否它需要服务。如外部设备不需要服务，浪费了CPU用来查询的时间。因此，这个内务操作引起增加执行时间。某些系统必须满足一个对特殊响应时间的请求，例如一个实时信号。在这一情况下，程序员必须保证，用一定的频率查询外部设备，当CPU必须执行一主程序以及子程序时，这个查询频率不可能总是容易满足。每一外部设备通常有它自己响应请求的时间；因此，用户必须建立一优先级方案。

中断法对查询法来说，有某些优点。当一外部设备需要服务时，它异步地通过硬件向CPU发信号，于是减少了查询不需要服务设备的内务操作。典型情况是，CPU可能结束它正在执行的指令，贮存重要的寄存器，识别外部设备的中断请求。在响应期间，CPU读出一个向量，这个向量使CPU访问相应中断服务程序的起始单元。如果几个中断请求同时出现，特殊逻辑能对这些请求进行优先级化，以便CPU响应中断时，最高优先级请求被CPU响应。

一驱使系统的中断要求附加的硬件以便控制中断请求信号，优先级和向量。8256把这些附加的硬件集成在这一芯片上。MUART 内的中断控制器直接与 MCS-85，iAPX-86，APX-88，iAPX-186，iAPX-188 微型计算机系统系列相兼容，并且也能用于其他微处理器。它包含八个优先级，然而，共有12个可中断源：10个内部的和2个外部的。因为有八个优先级，所以，在一个时间内只有八中断可使用。由命令寄存器1和方式寄存器选择分配中断的工作任务。MUART的中断源有一个固定的优先级。表2表示十二个中断源怎样

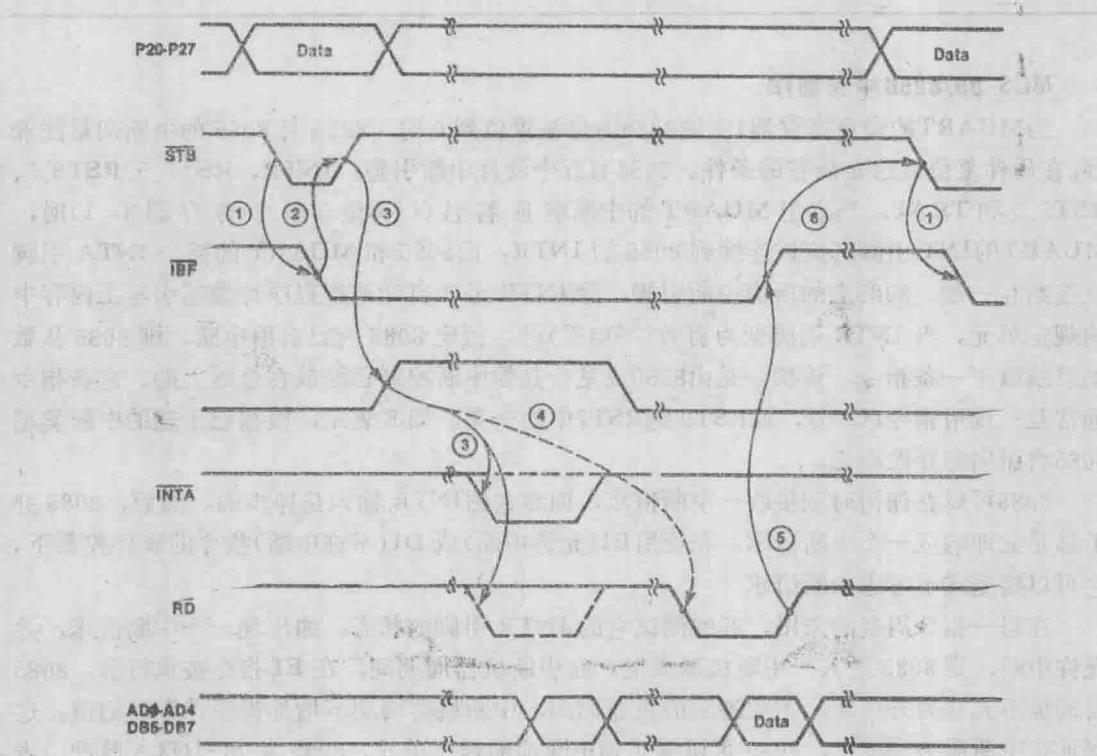


图 5a 同步交换输入/输出

分布在 8 个优先级中。

①设备表示使用 STB (Strobe) 的下降沿, 在端口 2 得到一个新字符。8256 通过触发 IBF (输入缓冲器满) 来识别。

②于是, 设备释放 STB, 而 8256 锁存字符。

③8256 通过 INT 通知微处理器, 一新字符已准备好传送。

④微处理器读出字符。

⑤信号 RD 的上升沿把信号 IBF 复位

⑥这个动作对设备发出 8256 的输入锁存器是空的和下一字可以传送的信号。

表 2 中断源对优先级的分配

优 先 级	源
最高 L0	第一计时器
L1	第二计时器或端口 1 中断
L2	外部中断 (EXTINT)
L3	第三计时器或第三计时器与第 5 个计时器,
L4	接收器中断
L5	发送器中断
L6	第四计时器或第二计时器与第四计时器
最低 L7	第五计时器或端口 2 的同步交换

MCS-85/8256 中断操作

当 MUART 的命令寄存器 1 中的 8086 的位被置位到 0 时, 8256 与 8085 的中断向量法兼容。在硬件复位后这是缺省的条件。8085 有五个硬件中断引脚: INTR, RST7.5, RST6.5, RST5.5 和 TRAP。当启用 MUART 的中断响应特性 (IAE 位 5 命令寄存器 3 = 1) 时, MUART 的 INT 引脚 15 应该连续到 8085 的 INTR, 而 8085 和 MUART 的两个 INTA 引脚应连结在一起。8085 上的所有中断引脚, 除 INTR 外, 自动地把程序计数器引导至内存中的规定单元。当 INTR 引脚变为有效 (高电平) 时, 假定 8085 有已启用中断, 则 8085 从数据总线取下一条指令, 该指令是由 8256 或某一其他中断控制已经放在总线上的。这条指令通常是一调用指令 (Call), 或 RST0 到 RST7 中的一条。图 6 表示, 根据已出现的中断类型 8085 将引向的存贮单元。

8085 可以在任何时刻接收一中断请求, 因为它的 INTR 输入是异步的。然而, 8085 并不总是立即响应一个中断请求。在使用 EI (允许中断) 或 DI (不许中断) 指令的软件控制下, 它可以接受或不考虑中断请求。

在每一指令周期的末尾, 8085 测试它的 INTR 引脚的状态。如出现一个中断请求, 并允许中断, 则 8085 进入一中断机器周期。在中断机器周期间, 在 EI 指令被执行前, 8085 自动地不允许另外中断。不象通常的机器周期, 中断机器周期不增加程序计数器的值。这保证在中断服务完成后, 8085 可以返回到中断前的程序单元。8085 发出 -INTA 脉冲, 表示正在接收请求, 并准备好处理中断。

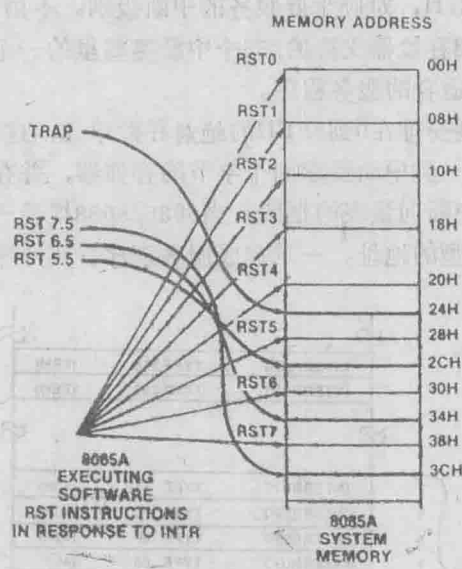


图 6 8085A硬件和软件RST转移单元

8256现在可以把程序的执行引向相应的服务程序。在第一个也是唯一的 $\overline{\text{INTA}}$ 脉冲期间，这个已完成。当接收 $\overline{\text{INTA}}$ 脉冲时，8256把操作码 RSTn 放置在数据总线上，根据中断要求的级别，这里的 n 等于 0 到 7。 RSTn 指令使程序计数器的内容被推入堆栈，然后把控制变换到指令，这条指令的地址是 $8 \times n$ ，如图6所示。

注意，在中断响应后的期间，因为中断是不允许的，在另外的中断能够处理前，在服务程序或主程序中，必须执行 EI 指令。

对于有关 8085 的中断操作的其他信息和 RSTn 指令可参考 MCS-85 用户手册。

iAPX-86/88-8256 的中断操作

当在命令寄存器1中的8086位被置为1时，MUART与8086/8088的中断引导方法兼容。MUART的INT引脚连接到8086/88的INTR引脚，而它的INTA引脚连接到8086/88的INTA引脚。象8085一样，8086/8088的INTR引脚也是异步的，以便中断请求能在任何时间出现。在软件控制指令下，8086/8088可以接受或不考虑INTR引脚上的请求。这些指令对允许中断标志IF置位或清除。8086/8088 加电或复位时，清除IF标志，禁止在 INTR 上的外部中断。

虽然，存在若干基本的类似点，8086/8088 中断的实际处理不同于 8085。中断请求出现和允许中断时，8086/8088 进入它的中断响应机器周期。中断响应机器周期把标志寄存器推入堆栈(象在 PUSHF 指令中的一样)。然后它清除 IF 标志，禁止中断。最后，代码段寄存器和指令指示器两者的内容被推入堆栈。于是，堆栈保留中断前的标志状态和程序单元，该单元是用来从服务程序返回的。然后8086/8088 发出两个 INTA 脉冲的第一个，第一个 INTA 脉冲向8256发出，8088 已承认它的中断请求。

现在，8256已准备好把程序的执行引向适合的服务程序。不象8085，第一个 INTA 脉冲是用来把一件指令放置在数据总线上，而第一个来自8086/8088 的 INTA 脉冲只用来发出承认8256的请求。第二个 INTA 脉冲使 8256 把单个中断向量字节放置在数据总线上。8256

设置中断向量字节40H到47H，对应于要服务的中断级别。不用作一个直接地址，这个中断向量字节属于由8086/8088存储器支持的256个中断类型里的一个。由已指定的中断类型的内容，把程序的执行引向适合的服务程序。

全部的256个中断类型安排在0到3FFH的绝对存储单元，这组成8086/8088的中断向量表。在中断向量表中的每一类中断要求四个字节的存储器，并存储一个代码段地址和一个指示器地址。图7表示出中断向量表的框图。当8086/8088接受一个中断向量字节时，它用4乘它的值，取得中断类型的地址。一旦完成服务程序，用一条IRET(中断返回)指令可

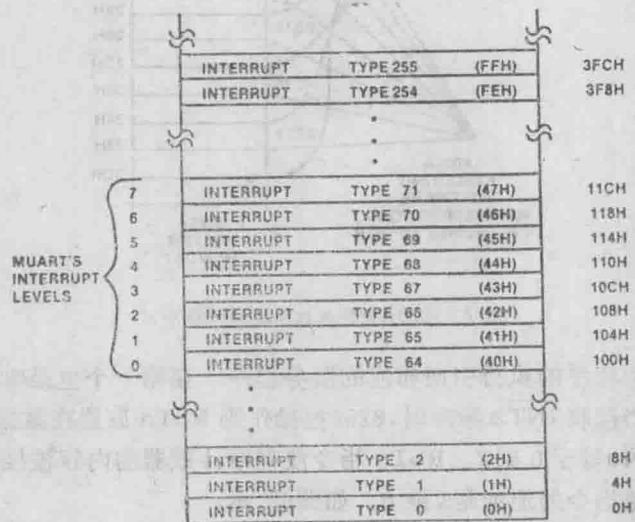


图 7 8086/8088中断向量表

以重新进入主程序。IRET指令将把中断前的指令指示器内容，代码段内容和标志弹出堆栈。于是，主程序将取回它中断地方的相同标志，而不管在服务程序中的变化。特别注意，这个包括IF标志的状态；从服务程序返回时，于是，中断自动地重新允许进行。对于更多的信息，请参考iAP×86，88用户手册。

没有INTA使用8256的中断控制器

没有把INTA信号连接到8256上的地方有几种配置。几个实例是8256与一个8051或8048一起使用时或者把8256上的INT引脚连接到8085的RST7.5，RST6.5或RST5.5输入端时。在这些配置中，命令寄存器3中的IAE位置为0，8256上的INTA引脚接到高电平。当中断出现时，CPU应该转移到一个服务程序，这个服务程序读出中断地址寄存器，以便确定中断请求级出现。中断地址寄存器包含4乘中断级。读出中断地址寄存器实际上等于INTA信号；它清除INT引脚，并对MUART表明中断请求已经响应。CPU读出中断地址寄存器中的值后，它可以把一个偏移量加到这个值中，并转移到中断向量表，该表包含向合适的中断服务程序跳转的指令。在图8中绘出一个示范这个子程序的8085程序。表3总结了若干优先级和8256送回CPU的中断向量。注意，当使用计时器1时，在8085方式中的RST0和硬件复位之间，有一个冲突出现，因为两条要求的指令在地址0H启动。然而，有一种在两个中间辨别的方法。在硬件复位后，全部控制寄存器复位到0H的值，因此，当使用计时器1时，用读出8256的一个控制寄存器的方法，可以识别复位和RST0，这个

寄存器还没有用 0H 值设计程序。如果 RST0 出现，这些控制寄存器将包含以前编制程序的值。

INTA, IN	INTADD	，读中断地址寄存器
MOV	L, A	，把中断地址放入HL
XRA	A	
MOV	H, A	
LXI	B, TABLE	，中断表偏移量送入BE
DAD	B	，把偏移量加到中断地址
PCHL		，跳向中断向量表

图 8 软件中断响应程序

中断寄存器

8256的中断控制器有几个寄存器与它有关：中断屏蔽寄存器，中断地址寄存器，中断请求寄存器，中断服务寄存器和一个优先级控制器。只有中断屏蔽寄存器和中断地址寄存器可以由用户存取。

表 3 中断源的中断级分配

中断级	重启动命令 8085方式	中断向量 8086方式	中断地址	触发方式	源(在任意时刻只能分 配一个源)	选 择
最高优先级 0	RST0	40H	0H	脉冲沿	第一计数器	—
1	RST1	41H	4H	脉冲沿	事件计数器/计时器2或在端口1 的 P17上的外部中断请求	命令字1的 BIT1(位2)
2	RST2	42H	8H	电 平	输入EXTIT	—
3	RST3	43H	CH	脉冲沿	事件计数器/计时器3或级联事 件计数器/计时器3和5	方式字T35 (位7)
4	RST4	44H	10H	脉冲沿	串行接收器	—
5	RST5	45H	14H	脉冲沿	串行发送器	—
6	RST6	46H	28H	脉冲沿	计时器4或级联事件计数器/ 计时器2和4	方式字T24 (位6)
7 最低优先级	RST7	47H	1CH	脉冲沿	计时器5或字符交换中 断请求	方式字P2C2- P2C0(位 2...0)

注意：如果没有中断请求挂起和 INTA周期出现，中断级2将是引导CPU 的缺省值。