

intel

微型机

器件应用集

科学技术文献出版社重庆分社

Intel 微型机器件应用集

李纪松 时文平 周保太
宋云生 李 萌
张 焕 文 校

科学技术文献出版社重庆分社

内 容 提 要

本书选编了5篇有关iAPX86/286、8274、80186等Intel微机器件在小型商业计算机、处理三维图形、同/异步通信等方面应用的文章。内容新颖而实用。除对有关器件的性能、用途作了简介外，特别介绍了它们的实际应用和实施方案，列举了大量的应用程序，是一本很有实用价值的参考书和应用指南。

本书不仅对广大Intel微机用户是必不可少的，而且适用于研究单位、大专院校、工矿企业、邮电通信、银行、仓库等系统和部门从事微机研究、设计、教学和应用的入員。

Intel 微型机器件应用集

李纪松等译

科学技术文献出版社重庆分社 出版
(重庆市市中区胜利路132号)

新华书店重庆发行所 发行
重庆通信学院印刷所 印刷

开本: 787×1092毫米1/16 印张: 16.5 字数: 400千字

1986年7月第一版

1986年7月第一次印刷

科技新书号: 125-192

印数: 3,800

书号: 15176-641

定价: 3.10元

前 言

随着微型计算机性能的不断改进，微机的应用和开发问题已日益为人们所关注。《Intel 微型机器件应用集》一书正是为了满足国内微机用户和广大读者对应用和开发微机的迫切需要而编译出版的。

本书选编了5篇有关iAPX 86/20、8274、80186等Intel微机器件的应用文章。内容新颖而实用。除对上述器件的性能、用途作了简介外，还对如何实现异步、同步通信，如何处理三维图形，商业部门如何合理经营，银行计算利息、年金、偿债基金，企业和仓库如何合理安排补充进货等实际应用问题提供了算法和具体实施方案，并列举了大量的应用程序。因此本书是一本很有实用价值的参考书和应用指南。

本书的特点是面向应用，不仅对广大Intel微机用户是必不可少的，而且也适用于研究单位，大专院校，工矿企业，邮电通信，银行，仓库等系统和部门从事微机研究、设计、教学和应用的人员。

本书由总参谋部第五十一研究所翻译。参加翻译的有李纪松（第一篇和第五篇附录）、时文平（第三篇）、周保太（第五篇）、李萌（第二篇）和宋云生（第四篇）。全书由张焕文校定。时文平、李纪松担任本书的绘图工作。

由于经验不足，水平有限，加之时间仓促，书中错误和不妥之处在所难免，请读者给予批评指正。

译 者

一九八五年六月

目 录

8274 多协议串行控制器在异步通信中的应用.....	(1)
前言.....	(1)
通信功能.....	(1)
系统接口.....	(1)
范围.....	(2)
串行异步数据链路.....	(2)
字符.....	(3)
帧格式.....	(3)
定时.....	(4)
奇偶校验.....	(4)
通信方式.....	(5)
中断状态.....	(5)
MPSC系统接口.....	(5)
硬件环境.....	(5)
操作接口.....	(6)
复位.....	(7)
外部/状态锁存.....	(7)
错误报告.....	(8)
发送器/接收器的预置.....	(8)
查询操作.....	(9)
中断驱动操作.....	(10)
中断结构.....	(10)
中断源/优先级.....	(11)
中断预置.....	(11)
中断服务例程.....	(12)
数据链路接口.....	(13)
串行数据接口.....	(13)
数据计时.....	(13)
调制解调器控制.....	(13)
附录A 异步通信的命令/状态说明.....	(14)
附录B MPSC查询式发送/接收字符例程.....	(22)
附录C 中断驱动式发送/接收软件.....	(24)

附录D 使用SDK-86的应用实例	(28)
硬件接口	(28)
软件接口	(29)
简短的8274应用程序	(33)
8274 多协议串行控制器在同步通信中的应用	(49)
引言	(49)
同步协议综述	(49)
面向比特的协议综述	(49)
字节同步通信	(50)
方框图	(51)
CPU 接口	(52)
串行接口	(52)
发送和接收数据通路	(53)
多协议串行控制器的中断结构	(54)
接收中断	(55)
发送中断	(57)
外部/状态中断	(57)
中断优先级的判定	(59)
多协议串行控制器的操作方式	(61)
中断驱动方式	(61)
DMA 方式	(64)
wait 方式	(65)
向量/非向量方式	(65)
应用实例	(67)
iSBC 88/45	(67)
软件	(68)
8274的预置	(68)
中断子程序	(70)
主程序	(73)
几种特殊应用	(74)
MPSC链路操作	(74)
双同步透明通信	(75)
自动允许方式	(76)
高速 DMA 操作	(76)
编程提示	(77)
异步操作	(77)
非向量方式	(77)
HDLC/SDLC 方式	(77)

状态寄存器RR2.....	(77)
预置程序.....	(77)
发送短缺/EOM锁存.....	(77)
同步字符装入禁止.....	(78)
EOI命令.....	(78)
DMA 方式的优先级.....	(78)
附录 A 应用实例: 软件表格.....	(78)
iAPX86/20 数值数据处理在小型商业计算机中的应用.....	(94)
引言.....	(94)
iAPX86/20硬件概述.....	(95)
例行程序的执行.....	(96)
利息.....	(98)
年金.....	(99)
偿债基金.....	(99)
补充进货算法.....	(100)
线性规划.....	(104)
结论.....	(108)
附录A.....	(109)
利率计算例行程序.....	(109)
年金计算例行程序.....	(110)
用Silver-Meal试探方式计算的例行程序.....	(113)
用Wagner-Whitin方式计算的例行程序.....	(117)
解线性规划问题的例行程序.....	(120)
iAPX86/20 数值数据处理在三维图形中的应用.....	(122)
引言.....	(122)
iAPX 86/20 硬件概述.....	(123)
三维图形的基本原理.....	(124)
执行过程.....	(128)
三维图形描述及用户接口.....	(131)
软件包的内部操作.....	(135)
现行转换.....	(133)
观察转换.....	(134)
Z-裁剪.....	(134)
投影.....	(135)
X-Y裁剪.....	(135)
窗口到观察口的转换.....	(136)
绘图机接口.....	(136)

性能测定.....	(136)
结论.....	(138)
附录A.....	(139)
附录B.....	(165)
80186 的应用介绍.....	(166)
前言.....	(166)
80186 综述.....	(167)
CPU.....	(167)
增强型80186 CPU.....	(168)
DMA 单元.....	(168)
定时器.....	(168)
中断控制器.....	(169)
时钟发生器.....	(169)
片选和就绪信号发生器.....	(169)
集成外围电路的访问.....	(169)
80186的使用.....	(170)
总线与80186的接口.....	(170)
存储器系统举例.....	(184)
HOLD/HLDA 接口.....	(190)
8086总线和80186总线的区别.....	(193)
DMA单元接口.....	(195)
DMA 特性.....	(195)
DMA 单元的程序设计.....	(196)
DMA 传送.....	(197)
DMA 请求.....	(198)
DMA 响应.....	(199)
内部产生的 DMA 请求.....	(200)
外部同步的 DMA 传送.....	(200)
DMA 暂停和 NMI.....	(202)
DMA 接口举例.....	(202)
定时器单元接口.....	(205)
定时操作.....	(205)
定时寄存器.....	(206)
定时器事件.....	(208)
定时器输入引脚操作.....	(209)
定时器输出引脚操作.....	(210)
80186 定时器应用实例.....	(210)
80186中断控制器接口.....	(211)

中断控制器模块.....	(211)
中断控制器操作.....	(211)
中断控制寄存器.....	(212)
中断源.....	(216)
中断响应.....	(218)
中断控制器的外部连接.....	(221)
8259A/级联方式接口举例.....	(224)
80130 iRMX 86方式接口举例.....	(224)
中断等待时间.....	(225)
时钟发生器	(225)
晶体振荡器.....	(226)
使用外部振荡器.....	(226)
时钟发生器.....	(226)
就绪信号的发生.....	(226)
复位.....	(226)
片选	(227)
存储器片选.....	(228)
外围片选.....	(229)
就绪信号的产生.....	(229)
片选用法举例.....	(230)
重叠片选区.....	(230)
80186 系统的软件	(231)
80186系统预置.....	(231)
iRMX 86系统预置.....	(231)
8086和80186执行指令的区别.....	(232)
结语	(233)
附录A 外围控制块	(233)
附录B 80186同步信息	(235)
附录C 80186 DMA 接口程序举例	(236)
附录D 80186 定时器接口程序举例	(239)
附录E 80186中断控制器接口程序举例	(243)
附录F 80186/8086 系统预置程序举例	(245)
附录G 80186等待状态特性	(246)
附录H 80186 的新指令	(250)
附录I 80186/80188 的区别	(253)

8274多协议串行控制器在 异步通信中的应用

前 言

8274多协议串行控制器(MPSC)是一种高级的双通道通信控制器,它用同步或异步协议来连接微处理器系统与高速串行数据链路(速度为每秒880K位)。8274很容易与大多数通用微处理器(如8048、8051、8085、8086和8088)、DMA控制器(如8237和8257)以及8089 I/O处理器接口。两条MPSC通信通道是完全独立的,能以全双工通信方式工作(同时进行数据发送和接收)。

通信功能

8274能执行许多面向通信的功能,包括:

- 把来自微处理器系统的数据字节转换为串行比特流,以便通过数据链路传送给接收系统。
- 接收串行比特流并把它重新转换为微处理器系统易于处理的并行数据字节。
- 在数据传送过程中进行错误检测。错误检测功能包括计算/传送错误代码(例如奇偶校验位或CRC字节)和用这些代码来检查接收数据的正确性。
- 进行不依赖于系统处理器的操作,以便在某种意义上减少与数据传送有关的系统总开销。

系统接口

MPSC的系统接口非常灵活,可支持下列数据传送方式:

1. 查询方式。系统处理器周期性地读(查询)8274状态寄存器,以便确定何时接收字符、何时传送字符和何时检测传送错误。
2. 中断方式。当接收到字符、需要传送字符和检测到传送错误时,MPSC就中断系统处理器。
3. DMA方式。MPSC通过每条串行通道的两个DMA请求信号,自动地为发送和接收功能请求来自系统存储器或I/O设备的数据传送。这些DMA请求信号可直接接到8237或8257 DMA控制器,或接到8089 I/O处理器。
4. 等待方式。MPSC 8274用就绪信号迫使处理器进入等待状态,并维持到8274为传送另一个数据字节准备就绪,因而使处理器的数据传送同步。这一性能使得8274可以通过一组用于高速数据链路的I/O指令直接与8086或8088处理器接口。

范围

本文介绍的是8274在异步通信方式中的应用。异步通信通常用来把数据传送到视频显示终端、调制解调器、打印机和其它低中速外围设备，或传送来自这些设备的数据。本文只介绍8274在中断驱动方式和查询系统方式这两种场合中的应用，而不涉及DMA方式和等待方式。因为这两种方式主要用于需要极高数据速率的同步通信系统。本文中的程序例子是用PL/M-86来编写的（附录B和附录C）。PL/M-86适于在iAPX-86和iAPX-88处理器系列中运行。PL/M-86与PL/M-80（适于在MCS-80和MCS-85处理器系列中运行）非常相似。此外，附录D描述了在iAPX-86/88设备中使用SDK-86的一个应用例子。

串行异步数据链路

串行异步接口是接收系统和发送系统不必同步的一种数据传送方法。在这种数据传送方法中，发送和接收系统使用本地产生的时钟脉冲（是数据传送速率的16、32或64倍），而不随数据一起传送时钟脉冲。发送系统发送一个字符信息时，该数据字符就由专用的START位和STOP位来构成帧（START位放在数据之前，STOP位放在数据之后）。这种帧格式信息使接收系统有可能暂时与数据传送同步（在阅读下文介绍的异步数据传输时请参考图1）。

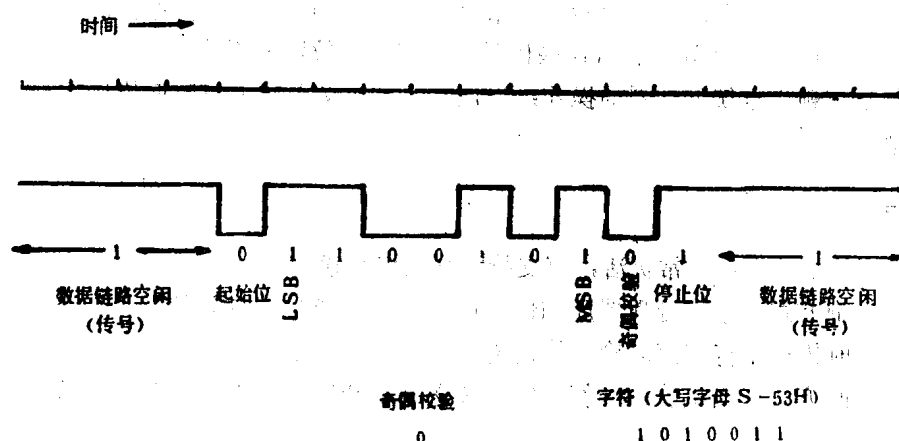


图1 带偶数奇偶校验位的7位ASCII字符的传送

通常，数据链路处于空闲或标记（marking）状态时，就连续地发送“传号”（二进制的1）。当要发送一个字符时，就立即在字符的数据位之前插入一个空号位（二进制的0，START位）。从发送“传号”转变为发送空号即通知接收系统，在起始位之后马上就会出现一个信息字符。图1所示是一个带偶数奇偶校验位的7位ASCII字符（大写字母S）的发送过程。请注意，字符是在起始位之后立即被发送的。字符中的各数据位

的发送是从最低位到最高位。在发送完字符的数据位之后立即发送奇偶校验位。STOP位（二进制的1）表示字符结束。

异步接口通常与人机接口设备如CRT/键盘等一起使用，这些设备的数据传送时间变化是极大的。

字符

在异步通信方式中，字符的长度可在5位到8位之间变化。字符长度取决于所使用的编码方法。例如，若发送博多码时，就使用5位字符；若发送ASCII数据时，就要求7位字符；而发送EBCDIC（扩充的二—十进制交换码）和二进制数据，则需要8位字符。传送由多个字符组成的报文信息时，各字符各自成帧并分别发送（图2）。

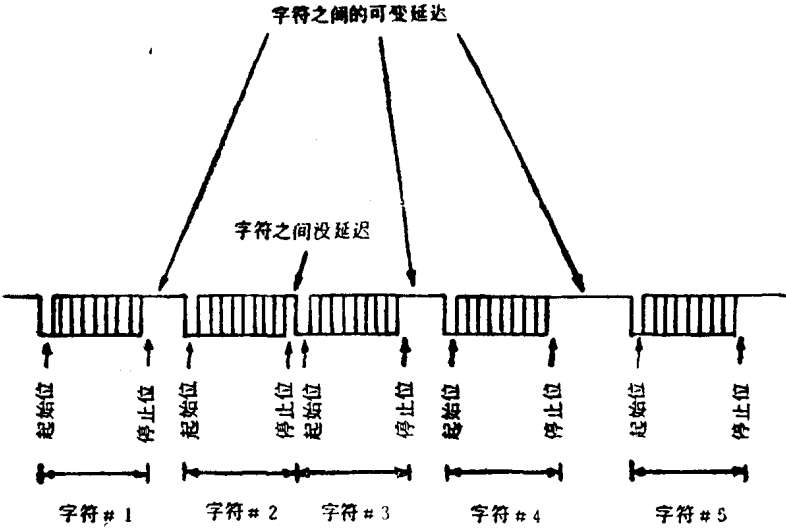


图2 多字符传送

这种帧格式方法保证了接收系统易于与各个字符的起始位和停止位同步，避免了接收器的同步错误。此外，这种同步方法可使发送系统和接收系统不受字符传送时可能存在的时间延迟的影响。

帧格式

字符由上述的START位和STOP位来构成帧。当检测到START位跳变（传号到空号）时，接收系统就假定后面将是一个数据字符。为了验证这种假定（并分离数据链路中的噪声脉冲），接收系统等待半个位的时间并再次抽样数据链路。如果链路已恢复到标记状态，那么就是噪声干扰，接收系统就等待另一START位跳变。

如果检测到一个有效的START位，接收器就对数据链路抽样后续字符的每个位。字符数据位和奇偶校验位（需要的话）都在其标称值的中心被抽样，直至接收到所有需要的字符为止。在数据位之后，接收器立即对数据链路抽样表示字符结束的STOP位。大多数系统都可用1个、1½个或2个停止位的规格。

定时

异步数据链路结构中的发送器和接收器是分别定时的。一般说来,各个时钟脉冲都是本地产生的,它们并不同步。事实上,各个时钟脉冲的频率是稍有不同的(实际上,这种频率差异不会超过百分之几。如果发送器和接收器的时钟速率相差很大的话,就会出现错误,因为会把数据位错误地认为是START或STOP帧格式位)。这些时钟脉冲是按以16、32或64倍的通信数据速率运行而设计的。时钟频率可使接收装置正确抽样输入的比特流。

串行接口数据速率以比特/秒来计算。“波特”这一术语用来表示每秒钟发送的信号电平可能改变状态的次数。通常,波特不等于比特率。只有当发送的信号具有两个状态(电平)时,波特率才与比特率相等。大多数点对点串行数据链路都使用RS-232-C、RS-422或RS-423电接口。这些接口要求两个电信号电平(波特等于比特率)。然而,调制解调器接口的比特率和波特率通常是不相等的。

虽然对使用于异步数据链路中的数据速率通常没什么限制,但为便于设备的相互连接,还是规定了有限的一组标准化速率,其范围从每秒75比特到每秒38400比特。表1列出了典型的异步数据速率及发送器和接收器所要求的有关时钟频率。

表 1 通信数据速率和相关的发送器/接收器的时钟频率

数据速率 (比特/秒)	时 钟 频 率 (KHz)		
	× 16	× 32	× 64
75	1.2	2.4	4.8
150	2.4	4.8	9.6
300	4.8	9.6	19.2
600	9.6	19.2	38.4
1200	19.2	38.4	76.8
2400	38.4	76.8	153.6
4800	76.8	153.6	307.2
9600	153.6	307.2	614.4
19200	307.2	614.4	—
38400	614.4	—	—

奇偶校验

为了检测传输错误,可在字符数据传过数据链路时给它加上一个奇偶校验位。奇偶校验位的置位或清除使得字符中的“1”的总位数变为偶数(偶数奇偶校验)或奇数(奇数奇偶校验)。例如,字母“A”用7位ASCII码1000001(41H)来表示,则带奇偶校验位时,所发送出的这一字符的数据码就有8位。采用偶数奇偶校验时是01000001(41H);采用奇数奇偶校验时则为11000001(0G1H)。值得注意的是,单个位的错误会改变所接收字符的奇偶性,因而容易检测出来。8274支持奇数和偶数奇偶校验以及奇

偶校验禁止方式，以支持二进制数据传送。

通信方式

两个装置之间的串行数据通信可用三种方式中的任一种来进行。在单工传送方式，数据链路只能单方向发送数据；在半双工方式，数据链路可双向发送数据，但不能同时发送；在全双工方式（最常用），数据链路可同时向两个方向发送数据。8274直接支持全双工方式。通过适当的软件控制，它也可与单工和半双工通信数据链路接口。

间断状态

异步数据链路通常包含有一个称为间断状态的专用序列。当发送装置迫使数据链路进入空号状态（二进制的0）并持续一定时间（一般为150毫秒）时，间断状态就开始了。许多终端都具有启动间断序列的键。在软件控制下，8274可在发送数据时启动间断序列，在接收数据时检测间断序列。

MPSC系统接口

硬件环境

MPSC通过一条8位数据总线来与系统处理器接口。每条串行I/O通道对应两个I/O地址或存储器地址，如表2所示。此外，MPSC还支持向量中断和链接式中断。

8274可构成存储器映象操作或I/O映象操作。

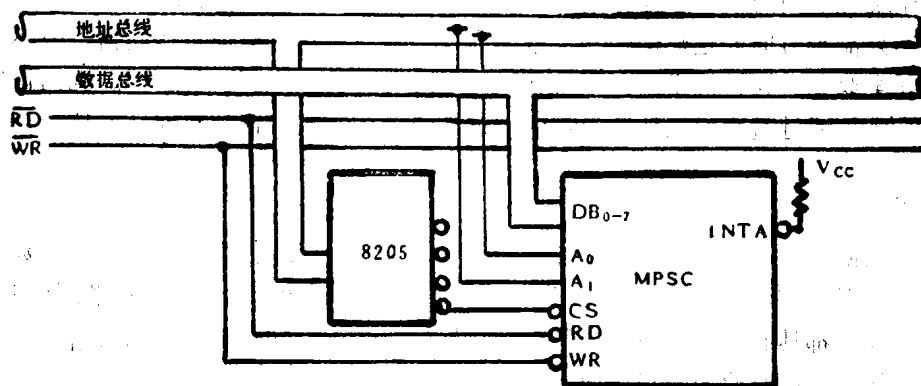
表2 8274的寻址

CS	A ₁	A ₀	读 操 作	写 操 作
0	0	0	读通道A数据	写通道A数据
0	1	0	读通道A状态	通道A命令/参数
0	0	1	读通道B数据	写通道B数据
0	1	1	读通道B状态	通道B命令/参数
1	X	X	高阻态	高阻态

8274可用极灵活的方法来构成与处理器的硬件接口。这主要取决于所选用的操作方式——查询方式、中断驱动方式、DMA方式或等待方式。图3示出了供8088微处理器在采用查询方式和中断驱动方式时使用的典型MPSC结构。

所有在异步串行I/O操作时所需要的串行到并行转换、并行到串行转换和奇偶校验都由MPSC自动完成。

a) 查询式结构



b) 链接式中断结构

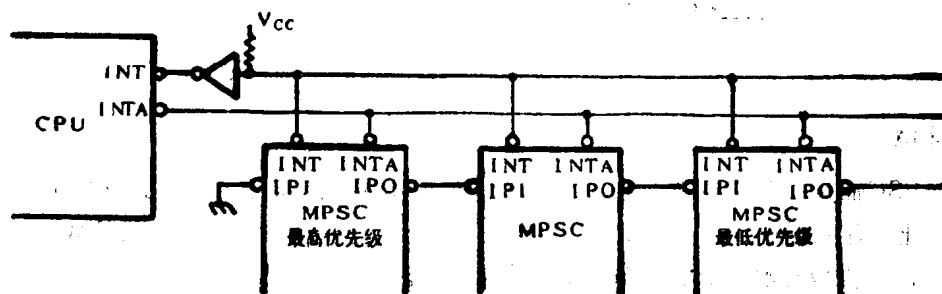


图3 查询方式和中断驱动方式环境中的8274硬件接口

操作接口

命令、参数和状态信息保存在 MPSC 中的 22 个寄存器内 (每条通道有 8 个可写寄存器和 3 个可读寄存器)。这些寄存器都通过每条通道的命令/状态端口进行访问。在访问 MPSC 的一条通道内的命令/状态寄存器时, 由内部的一个指针寄存器来选择对哪个命令或状态寄存器进行读或写。图 4 示出了每条串行通道的命令/状态寄存器的结构。在下文的讨论中, 可写寄存器称为 WR0—WR7, 可读寄存器称为 RR0—RR2。

每次写 WR0 时, WR0 的低 3 位自动装入指针寄存器。复位之后, 把 WR0 置为 0。因此, 对命令寄存器的第一次写操作就把数据装入 WR0 (从而把指针寄存器置位)。写 WR0 之后, 下一个读或写操作就访问由指针选定的寄存器。在完成读或写操作以后, 该指针就复位。用这种方法, 读或写任何一个 MPSC 通道寄存器都需要两次 I/O 访问。第一次访问总是一条写命令, 这条写命令用来置位指针寄存器; 第二次访问不是一条读命令就是一条写命令。指针寄存器 (先前已置位) 确保了读或写合适的内部寄存器。在第二次访问之后, 指针寄存器就自动复位。写 WR0 和读 RR0 时不需预置指针寄存器, 这一点请加以注意。

在预置阶段和正常的 MPSC 操作时，系统处理器读和/或写各个寄存器。有关这方面的操作，将在后面各有关章节中详细讨论。在异步通信方式中不使用 WR6 和 WR7，这一点也请注意。

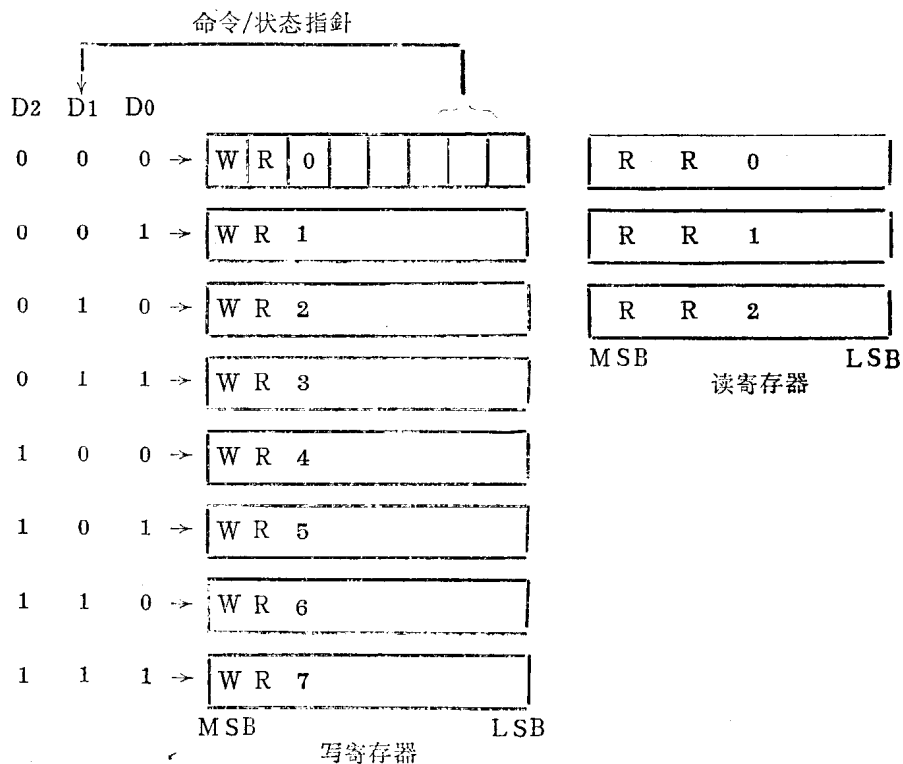


图 4 命令/状态寄存器结构(每条串行通道一组)

复位

一旦启动8274的RESET(复位)线，两条 MPSC 通道就进入空闲状态。迫使串行输出线进入标记状态(高电平)，从而也迫使调制解调器的接口信号 (RTS、DTR) 变为高电平。此外，指针寄存器置为 0。

外部/状态锁存

MPSC不断地监控下列四个外部/状态条件的状态：

1. CTS——清除发送输入脚。
2. CD——载波检测输入脚。
3. SYNDDET——同步检测输入脚，在异步通信方式中该脚可作为通用输入来使用。
4. BREAK——间断条件(接收器输入脚上的一系列空位)。

这些监控条件中任一个的状态的变化，都将引起 RR 0 中的关联状态位被锁存(和任意引起中断)。

错误报告

在异步方式的数据接收期间，可能碰到如下三种错误状态：

1. 奇偶校验错误。如果算上奇偶校验位并把这些位与每个字符一起发送，同时把MPSC设置为检验奇偶校验（WR 4 中的位 0 置位）的话，那么一旦该字符中“1”的位数（包括奇偶校验位）与奇偶校验标记的奇/偶设置（WR 4 中的位 1）不相符，就出现奇偶校验错误。

2. 帧格式错误。如果在奇偶校验位（若允许奇偶校验的话）之后或在最高有效数据位（若不允许奇偶校验的话）之后没有检测到停止位，就出现了帧格式错误。

3. 超限。如果收到一个输入字符但接收缓冲器已满（因为系统处理器没读出先前收到的字符），那就会产生超限。若出现超限错误，则刚接收到的输入字符将会改写先前的字符。

发送器/接收器的预置

为了实现异步通信方式，必须用下述信息对每条MPSC通道进行预置：

1. 时钟频率。这一参数由WR 4 的位 6 和位 7 来指定。时钟频率可设置成是数据链路比特率的16、32或64倍（参阅附录A的WR 4 说明）。

2. 停止位的位数。这一参数由WR 4 的位 2 和位 3 来指定。停止位的位数可设置成1、1½或2（参阅附录A的WR 4 说明）。

3. 奇偶校验选择。奇偶校验由WR 4 的位 0 和位 1 来设置成奇、偶或无校验（参阅附录A的WR 4 说明）。

4. 接收字符长度。这一参数把所接收的字符长度设置成5、6、7或8位。该参数由WR 3 的位 6 和位 7 来指定（参阅附录A的WR 3 说明）。

5. 接收器允许。可通过置位或清除WR 3 的位 0 来允许或禁止串行通道的接收器操作（参阅附录A的WR 3 说明）。

6. 发送字符长度。这一参数把所发送的字符长度设置成5、6、7或8位。该参数由WR 5 的位 5 和位 6 来指定（参阅附录A的WR 5 说明）。长度小于5位的字符，可通过把发送长度设置成5位来发送（将WR 5 的位 5 和位 6 置为1）。

MPSC 确定被发送的字符数据字节的实际位数。这些被发送的位必须在数据字节中右侧对齐，邻接的三个位必须为0，其余的所有位必须置为1。下表列出了1到5位数据的传送格式：

								发送的位数 (字符长度)
D7	D6	D5	D4	D3	D2	D1	D0	
1	1	1	1	0	0	0	C	1
1	1	1	0	0	0	C	C	2
1	1	0	0	0	C	C	C	3
1	0	0	0	C	C	C	C	4
0	0	0	C	C	C	C	C	5