

附录一 可编程串行通信接口 8251A

8251A 是美国 Inter 公司的产品，它是 8251 的改进型。具有同步、异步接收或发送（USART）功能，使用单一+5V 电源和单相时钟，双列直插 28 脚封装形式。

（一）8251A 的基本性能

1. 可以工作在同步或异步方式下，两种方式下的字符位数 5—8 个；
2. 同步方式时传输速率可达 0—64K，异步方式时传输速率可达 0—19.2K；
3. 异步传输时，可自动产生一个起始位，程控产生 1 个、1.5 个、2 个停止位；
4. 具有奇偶错、数据丢失和帧错误和检测能力；
5. 同步方式时，可自动检测，插入同步字符。

（二）8251A 内部结构

8251A 内部结构框图如图 9—22 所示。

它共由 7 个方框组成，即接收缓冲器、接收控制电路、发送缓冲器、发送控制电路、调制/解调制控制电路、读/写控制逻辑和数据总线缓冲器。

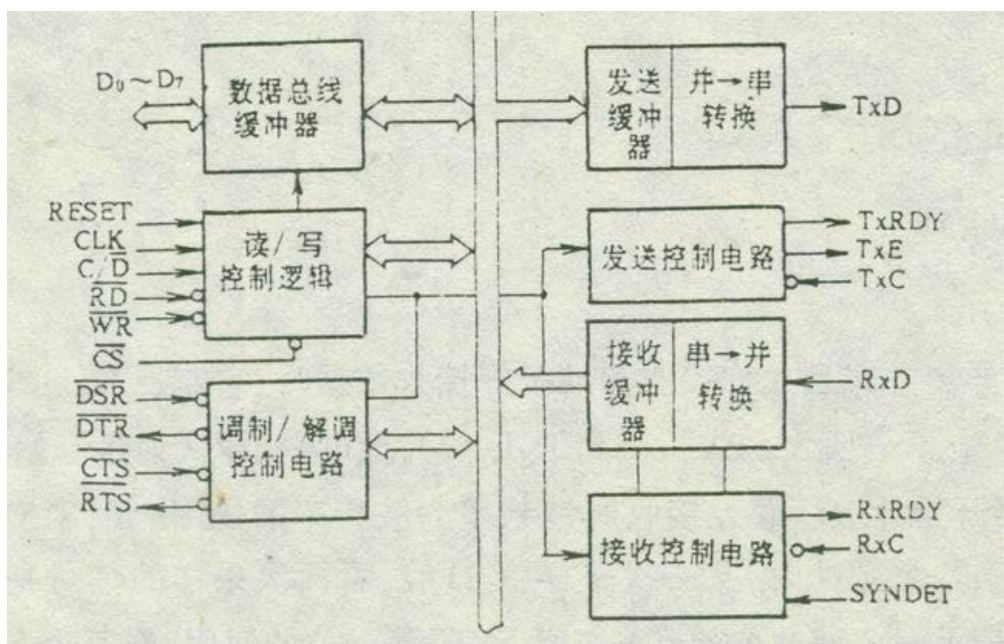


图 9—22 8251A 内部结构原理图

1. 数据总线缓冲器

这是 8251A 与 CPU 之间的一条数据通道，来自 CPU 的各种控制命令和待发送的字符信息经该通道到达 8251A 的内部，由 8251A 从发送设备处接收到的各种字符信息经该通道到达 CPU 的内部。

2. 读/写控制逻辑电路

它是接收来自 CPU 的各种控制信息，从而确定本次操作的方式。如果 $\overline{WR}=0$ ，表示 CPU 向 8251A 写入数据或控制字；如果 $\overline{RD}=0$ ，表示 CPU 读取 8251A 的数据信息或状态信息；而 C/\overline{D} 则表示读写对象是 8251A 内部的控制/数据寄存器， $C/\overline{D}=1$ 表示对控制寄存器操作， $C/\overline{D}=0$ 表示对数据寄存器操作；CLK 是时钟信号，提供给 8251A 作为内部定时器使用，RESTE 则是使 8251A 处于空闲状态。总之，读/写控制逻辑电路提供的各种信号的组合，构成了 8251A 操作命令。

3. 调制/解调控制电路

当计算机进行远程通信时，要用调制器将串行接口送来的数字信号变成模拟信号，再通过电话线发送出去。接收器则是将接受到的模拟信号经解调器变成数字信号，再由串行接口送入计算机。在全双工通信情况下，每个收发站都需连接调制/解调器。8251A 提供的调制/解调控制电路，就是提供一组通用的控制信号，使 8251A 可以直接与调制/解调器相连，以完成远程通信的任务。

4. 发送缓冲器

由数据发送缓冲器和并→串移位寄存器组成。当要发送数据时，按照发送的要求，将发送数据变成串行数据，经 TXD 引脚发送出去。

5. 发送控制电路

它是协调发送缓冲器工作的。同时也为同步或异步方式传送提供必需的识别控制位信息，如：起始位、同步字符等等。

6. 接收缓冲器

它与发送缓冲器的功能相反，将接收到的串行数据转换成与计算机处理的数据格式相同的并行数据。

7. 接收控制电路

协调接收缓冲器工作的。

（三）8251A 的工作原理

1. 异步接收方式

当 8251A 工作于异步方式且允许接收和准备好接收数据时，它监视 RXD 线。在没有字符信息时，RxD 为高电平。一旦 8251A 检测到 RxD 线为低电平，即认为是起始位（Space）到达，便启动内部计数器开始计数。假设接收时钟频率为波特率的 16 倍频，8251A 的内部寄存器计数接收始终的第 8 个脉冲时，又一次采样 RXD 线，看两次采样的信号是否一致。如果相同，即都为低，则表示一个起始位的到来。此后，每隔一位的时间，在每个数据中间的一个接收时钟 \overline{RxC} 的上升沿采样一次 RxD 线作为输入信号，送至串→并移位寄存器。在移位寄存器中数据被转换成并行，并且进行奇偶校验并去除停止位后，经 8251A 内部数据总线送至接收缓冲器，同时发出 RxRDY 信号，表示一个字符的接收和转换全部完成。

如果在第二次采样 RxD 线发现为高电平，则可能是一个干扰噪声。于是 8251A 将不予理会，重新进行下一次的采样。

2. 异步发送方式

异步发送时，首先必须由程序设置 TxEN (Transmitter Enable——发送允许) 和 \overline{CTS} (Clear To Send——由外设发来的对 CPU 请求发送信号的响应信号) 有效后，方可发送。发送时，发送器为每个字符自动地加上 1 位起始位，并按照程序的要求加上 1 位奇偶校验位，1、1.5、或 2 位停止位，在发送时钟 \overline{TxC} 的下降沿经发送移位寄存器从 \overline{TxD} 线发出。

3. 同步接收方式

常用的串行同步通信数据格式分为单、双、外同步和 SDLC/HDLC 四种格式。

单同步数据格式的串行同步通信方式，在内同步方式允许接收后，8251A 由编程命令进入搜索方式。它监视 RxD 线，每出现一个数据位就把它移一位，然后把接收寄存器与含有同步字符（由程序给定）的寄存器相比较，如果相同，表示接收和发送方已同步，接收方便使 SYNDET 信号输出为高。如果不同，则接收下一个数据并重新进行比较过程。

如果采用双同步数据格式传送，则在比较第一个同步字符相同后，进行第二个同步字符的比较，若相等，则表示已同步。如果不相等，则重新比较输入移位寄存器和第一个同步字符寄存器的内容，相等，已同步。否则重新进行下一个数据的比较过程。

对于外同步的情况，则有所区别。它是由外加同步信号使同步输入端 SYNDET 变为高电平实现的。在数据格式中没有同步字符。SYNDET 加上高电位以后，立即发送相应字符数据。

SDLC/HDLC 的情况与其它同步接收方式有所区别，但也是以同步字符（称为标志）作为数据同步的。其同步字符格式固定为 01111110。当接收方收到该标志时，进入同步，从而完成相应的 SDLC/HDLC 操作。

在实现同步以后，通信双方即进行数据的传输，8251A 利用接收时钟采样和移位 RxD 线上的数据位，且按规定的位数，把它送至接收数据缓冲器，并在 RxRDY 线上发出一个信号，告知 CPU 接收到一个有效的字符。

4. 同步发送方式

与一步发送方式一样，同步发送方式是在 TxEN 和 $\overline{\text{CTS}}$ 有效后开始的。首先发送的是用以同步的一个或两个字符，随后就是有效数据位，在数据中可能含有一个奇/偶校验位，也可能没有，由程序设定。对 SDLC/HDLC 在发完同步字符后，还要发送地址、控制两个场的规定信息，然后才是有效数据信息。

在传送过程中，可能会出现 CPU 来不及将新的字符数据输出给 8251A 的情况。为此，8251A 能自动地在 TxD 线上插入同步字符，从而使字符之间没有间隙存在。

（四）8251A 的引脚特性和外部连接

8251A 的作用是串行通信接口，连接 CPU 与外设（或调制解调器）。因此其引脚从功能上分成五组两大部分。

一部分是与 CPU 相连的数据总线组、控制总线组，另一部分是与外设或调制解调器相连的发送控制组、接收控制组以及外设/调制解调器控制组。每组信号的具体定义如图 9—23 所示。8251A 与 8086CPU 和外部设备的连接如图 9—24 所示。

1. 数据总线组

D7—D0：数据总线信号，双向三态。这是 8251A 与 CPU 之间的数据通道。传送各种数据控制命令和状态信息。

2. 控制总线组

该组由 8 个信号组成，均为输入。

①+5V 和 GND：电源和地。8251A 使用单一+5V 电源工作。

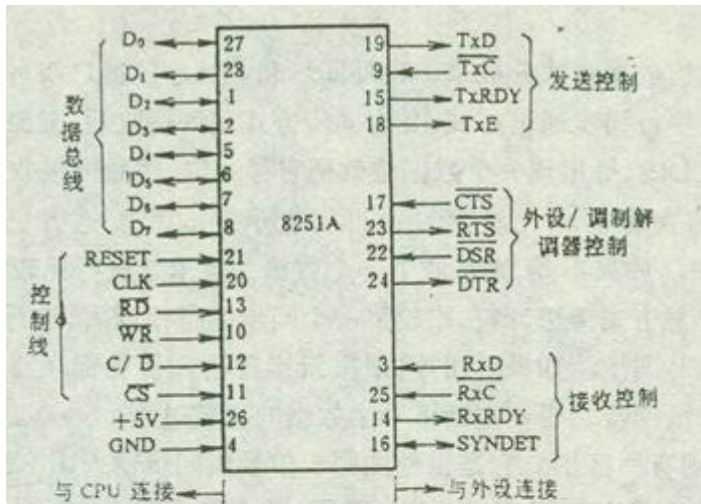


图 9-23 8251A 引脚特性图

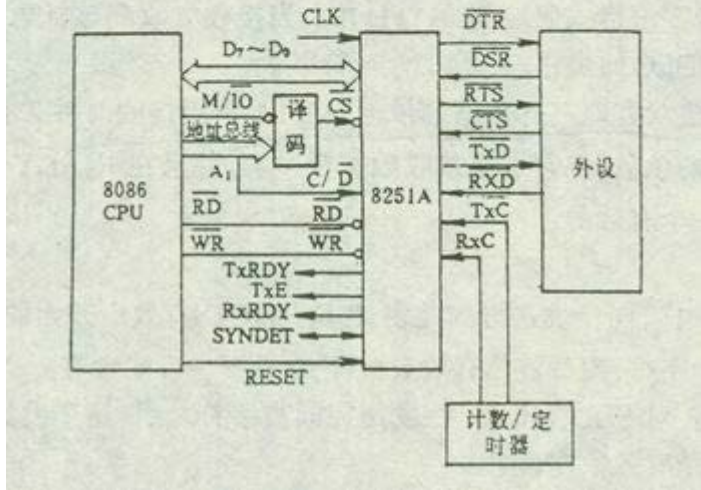


图 9-24 8251A 与 CPU 及外设的连接图

②RESET 信号：复位信号。当 RESET 有效，8251A 的所有功能复位，从头开始。

③CLK:系统时钟信号。CLK 为 8251A 内部提供定时信号。在同步方式时，CLK 的频率必须大于发送器输入时钟 $\overline{\text{TxC}}$ 和接收器输入时钟 $\overline{\text{RxC}}$ 的 30 倍；在异步方式时，必须大于发送和接收时钟的 4.5 倍。

④ $\overline{\text{RD}}$:CPU 读信号。当 $\overline{\text{RD}}=0$ 时，表示此时 CPU 正从 8251A 中读入数据或状态信息。

⑤ $\overline{\text{WR}}$: CPU 写信号。当 $\overline{\text{WR}}=0$ 时，表示此时 CPU 正向 8251A 的控制寄存器或数据寄存器中写入控制命令或数据。

⑥ $\overline{\text{C/D}}$: 控制/数据寄存器的选择信号。由于 8251A 内部具有完全独立的控制或数据寄存器，在对它们进行操作时，以不同的地址加以区别。当 $\overline{\text{WR}}=0$, $\overline{\text{C/D}}=0$ 时，表示本次操作选中数据寄存器；而当 $\overline{\text{WR}}=0$, $\overline{\text{C/D}}=1$ 时，表示选中控制寄存器。

⑦ $\overline{\text{CS}}$: 片选信号。与 8255A 可编程并行接口芯片相同，8251A 在工作时， $\overline{\text{CS}}$ 必须有效。通常 $\overline{\text{CS}}$ 是经过译码器与 CPU 的地址总线相连。这样，就构成了 8251A 的芯片地址（也称端口地址）。

欲使 8251A 工作，首先由 CPU 向地址总线送去相应的地址信息，使 $\overline{CS}=0$ ，然后， \overline{WR} 、 \overline{RD} 、 C/\overline{D} 上的信号才有效。 \overline{CS} 与 \overline{WR} 、 \overline{RD} 、 C/\overline{D} 之间构成的编码和对应操作，如表 9—5 所示。

表 9—5 8251A 的编码和对应的操作

\overline{CS}	C/\overline{D}	\overline{RD}	\overline{WR}	功能
0	0	0	1	CPU 从 8251A 读数据
0	1	0	1	CPU 从 8251A 读状态
0	0	1	0	CPU 写数据到 8251A
0	1	1	0	CPU 写命令到 8251A
0	X	1	1	8251A 数据总线三态
1	X	X	X	8251A 数据总线三态

3. 发送控制组

该组有四个信号，表明 8251A 工作于发送器时所处的状态和输出的数据。

①TxRDY (Transmitter Ready): 发送器准备好信号，高电平有效，输出 TxRDY 表示当前 8251A 的状态处于发送缓冲器空，且 Tx \overline{E} 和 \overline{CTS} 有效。实际使用时，常将 TxRDY 信号作为中断申请信号与 CPU 的 INT 端相连。当 8251A 发送缓冲器空时，TxRDY=1，向 CPU 发出中断申请，CPU 在执行中断服务程序时，向 8251A 送出一个字符数据。当该数据到达 8251A 之后，TxRDY=0。同样，CPU 也可以采用查询的方式获取 TxRDY 的状态。如果 TxRDY=1，表示发送缓冲器已空，CPU 发出一个字符数据给 8251A。否则，继续查询。

②Tx \overline{E} (Transmitter Empty): 发送器空信号，输出高电平有效。当 Tx \overline{E} 输出有效时，表明发送器中并→串转换器已空。Tx \overline{E} 与 TxRDY 是不同的。TxRDY 表示的是发送数据缓冲器的状态，而 Tx \overline{E} 表示的是发送数据缓冲器后的并→串转换器的状态。它们之间的关系是，TxRDY 较 Tx \overline{E} 之前有效。

TxE 有效，必定 TxRDY 有效，而 TxRDY 有效则不一定 Tx \overline{E} 有效。只要 CPU 向发送器输出一个数据，Tx \overline{E} 和 TxRDY 必然都无效。

在同步方式工作时，若 CPU 来不及输出一个字符，则 Tx \overline{E} 输出变高，同时在发送器的输出线上将自动插入同步字符，以填补传送空隙。在插入同步字符时，Tx \overline{E} 的输出仍为高，表示发送器此时在发送同步字符，而非数据字符。

③Tx \overline{D} (Transmitter Data): 发送器数据信号，输出。由 CPU 送给 8251A 的并行数据在 Tx \overline{D} 线上串行发出。

④Tx \overline{C} (Transmitter Clock): 发送器时钟，输入。发送器时钟控制发送字符的速度。在同步方式下，Tx \overline{C} 的频率等于字符传输的波特率，在异步方式下，Tx \overline{C} 的频率可以是字符传输波特率的 1 倍、16 倍、32 倍或 64 倍，具体由程序控制设定。

例如：

波特率=300Band，则
Tx \overline{C} =300Hz (1 倍频)
Tx \overline{C} =4800Hz (16 倍频)
Tx \overline{C} =19.2kHz (64 倍频)
波特率=2400Band，则
Tx \overline{C} =2400Hz (1 倍频)

$\overline{\text{TxC}} = 38.4\text{kHz}$ (16 倍频)

$\overline{\text{TxC}} = 153.6\text{kHz}$ (64 倍频)

8251A 要求 $\overline{\text{TxC}}$ 时钟频率在 1 倍频方式最大不超过 64kHz, 在 16 倍频方式小于等于 310kHz, 在 64 倍频时, 小于等于 615kHz。

8251A 在 $\overline{\text{TxC}}$ 的上升沿采样数据。如果发、收两方的波特率相同, 通常可以使用同一时钟发生器为 $\overline{\text{TxC}}$ 和 $\overline{\text{RxC}}$ 时钟信号。

4. 接收控制组

①RxRDY (Receiver Ready): 接收器准备好信号, 输出, 高电平有效。接收器准备好信号有效表示此时已经接收了一个字符数据。这个信号可以作为中断申请信号与 CPU 的 INT 信号相接, 也可以作为状态信号为 CPU 查询所用。当 CPU 得知 RxRDY=1 时, 将从接收数据缓冲器中读取数据, 一旦 CPU 读取一个字符之后, RxRDY 信号自动复位。

注意, 如果 RxRDY 信号和 TxRDY 信号同时接到 CPU 的 INT 端, 须经过相应的逻辑组合, 当 CPU 相应中断请求时, 能够区分是 RxRDY 信号有效还是 TxRDY 信号有效。

②SYNDET (Synchorous Detect): 同步检测信号, 双向, 高电平有效。该信号仅用于同步方式。其输入/输出由程序控制。

当 8251A 工作于内同步方式时, SYNDET 输出。如果 8251A 检测到所要求的同步字符时, 该信号输出有效高电平, 表示此刻接收、发送端同步。若为双同步字符格式数据, 此信号在第二个同步字符的最后一位的中间变高。当 CPU 执行一次读状态操作时, SYNDET 复位。

当 8251A 工作在外同步方式时, SYNDET 为输入。从这个输入端输入的一个正跳变沿使 8251A 在下一个接收时钟 $\overline{\text{RxC}}$ 的下降沿开始接收数据。SYNDET 输入的高电平至少应维持一个 $\overline{\text{RxC}}$ 周期, 直到 $\overline{\text{RxC}}$ 出现一个下降沿方可变低。对于 8251A 而言, 某一时刻由程序设定为内或外同步方式。

③RxD (Receiver Data): 接收器数据, 输入线。8251A 经过 RxD 线接收来自发送方的串行数据。

④ $\overline{\text{RxC}}$ (Receiver Clock): 接收器时钟, 输入。控制 8251A 接收数据的速度。其它与 $\overline{\text{TxC}}$ 相同。

5. 外设/调制解调器控制组

① $\overline{\text{DTR}}$ (Data Terminal Ready): 数据终端准备好。这是一个通用的输出信号, 低电平有效。该信号受 CPU 的控制, 当 CPU 通过指令使 $\overline{\text{DTR}} = 0$ 时, 告知外设, CPU 当前已经准备就绪。

② $\overline{\text{DSR}}$ (Data Set Ready): 数据装置准备好, 是一个通用输入信号, 低电平有效。该信号是外设通过 8251A 传送给 CPU 的状态信号, 当外设和 $\overline{\text{DSR}}$ 端相连时, CPU 可以由软件查询 8251A 状态寄存器的 $\overline{\text{DSR}}$ 位, 得到 $\overline{\text{DSR}}$ 的状态。 $\overline{\text{DSR}} = 0$ 表示外设已经准备好。

③ $\overline{\text{RTS}}$ (Request To Send): 发送请求, 由 8251A 送往外设的, 低电平有效。当 CPU 准备好发送数据时, 使 $\overline{\text{RTS}} = 0$, 通知外设, CPU 将发送数据。

④ $\overline{\text{CTS}}$ (Cleat to Send): 发送清除也成为发送允许, 输入, 低电平有效。这是外设对 8251A 的 $\overline{\text{RTS}}$ 信号的应答信号。当 CPU 发送请求信号有效后, 一旦外设发来 $\overline{\text{CTS}} = 0$, 则发送器开始发送。在发送过程中, 如果 $\overline{\text{CTS}}$ 无效, 发送器将在已经写入的数据全部发完之后, 停止发送。

(五) 8251A 的编程

8251A 是可编程串行接口, 在使用之前必须由程序对其工作状态进行设置 (称为初始化),

其中包括：同步方式还是异步方式、传输波特率、字符代码位数、校验方式、停止位位数等。如果是同步方式，还要设定是内同步还是外同步。

8251A 经过初始化编程后，即可执行通信操作。

在初始化编程时，向 8251A 发的控制字分为两类：方式控制字和命令控制字。

1. 方式控制字

由于同步和异步方式在操作上区别很大，所以方式控制字的基本格式为：异步方式控制字格式和同步方式控制字格式。CPU 向其设置时，用的端口地址是相同的，都是 $C/\bar{D}=1$ 。两种方式控制字的区别在最低两位：低两位为 00 是同步方式控制字；否则是异步方式控制字。

方式控制字的基本格式如图 9-25 所示。从图中可知：

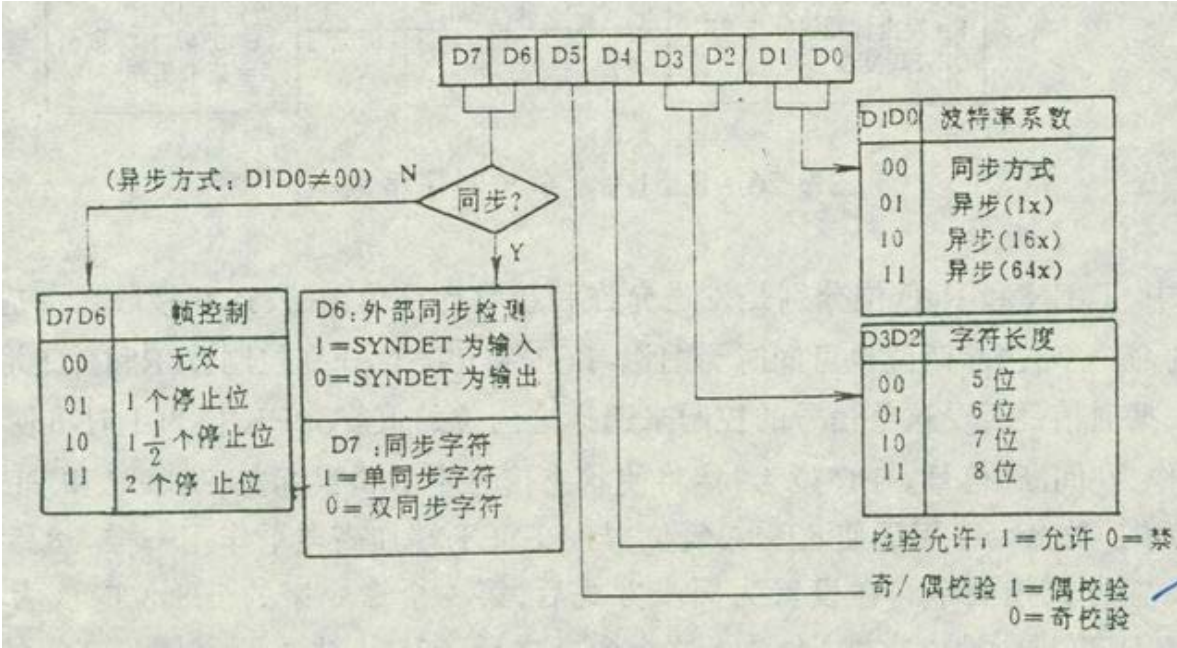


图 9-25 8251A 方式控制字格式

D0D1: 首先区分发送方式，其次是在异步条件下的输入时钟频率与波特率之间的系数。接收和发送的波特率可以不同，接收时钟和发送时钟的频率也可以不同，但是接收和发送的波特率系数只能是同一个。

D3D2: 确定每个字符的位数。字符长度值可以从 5 位到 8 位不等。当程序制定字符位数小于 8 位时，有效数据位右对齐，高位以 0 补充。

D4: 决定是否使用奇偶校验位，D5 表示校验的方式。注意，校验位仅仅是提供传送过程中是否有出错的判定，当正确接收到有效数据后，检验位的作用完成。因此，从 RxD 上接收的奇偶校验信号是不会进入 CPU 的。

D7D6: 与采用的传输方式有关。当 D1D0≠00 为异步方式时，表示停止位的个数，其中 D7D6=00 时无效。当 D1D0=00 为同步方式时，D6=1 为外同步，D6=0 为内同步。D7 表示同步字符的个数，D7=1 为单同步，D7=0 为双同步。如果为外同步方式，则 D7 无效。

2. 命令控制字

CPU 向 8251A 发命令控制字，控制 8251A 的实际操作。发命令控制字时用的端口地址与方式控制字的地址相同，它们的区别是靠发送的前后顺序。其格式如图 9-26 所示。

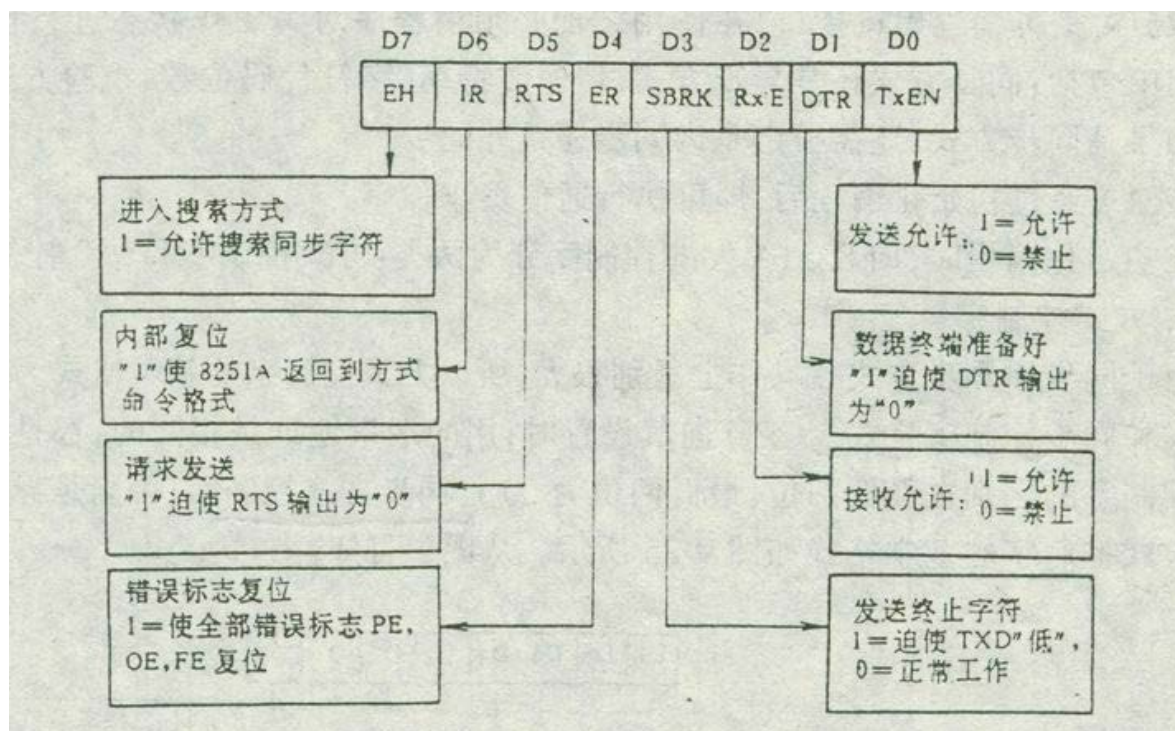


图 9-26 8251A 命令控制字格式

格式中，TxEN 和 Rx E 位分别是发送允许和接收允许位，在发送和接收之前应发相应位为“1”的命令字，当然两位也可同时为“1”。DTR 和 RTS 位分别控制 DTR 和 RTS 端的输出状态，要根据通信是否受这些信号的控制来选择这两位值。SBRK 位为 1 时将使 TXD 输出低电平作为“间断”信号。ER 位为“1”将使状态信息中的错误标志 PE、OE 清除为 0。IR 位为“1”的命令字用于使 8251A 复位，与 RESET 端加高电平作用一样。8251A 复位以后，等待设置方式控制字。在设置为同步方式后，第一个命令字的 EH 为应该为 1，称为 ENTER HUNT（进入搜索方式）命令。这个命令之后，8251A 进入测试同步字符的操作状态。

3. 状态字

8251A 内部设有状态寄存器，CPU 可由读指令 IN 获取状态寄存器的内容，判定 8251A 当前的工作状态。状态寄存器各位的定义如图 9-27 所示。

D1 (RxRDY)、D0 (TxRDY) 位可供 CPU 查询。状态位 TxRDY 和输出引脚 TxRDY 有所不同。状态位 TxRDY 并不受命令控制字中允许发送位 TxEN 和输入的允许发送引脚 CTS 的控制，它只反映发送命令/数据缓冲器的状态，只要数据缓冲器一空就置位；而输出引脚 TxRDY 却要受到上述内部和外部两个条件限制，它不只反映发送过程中数据缓冲器的状态。

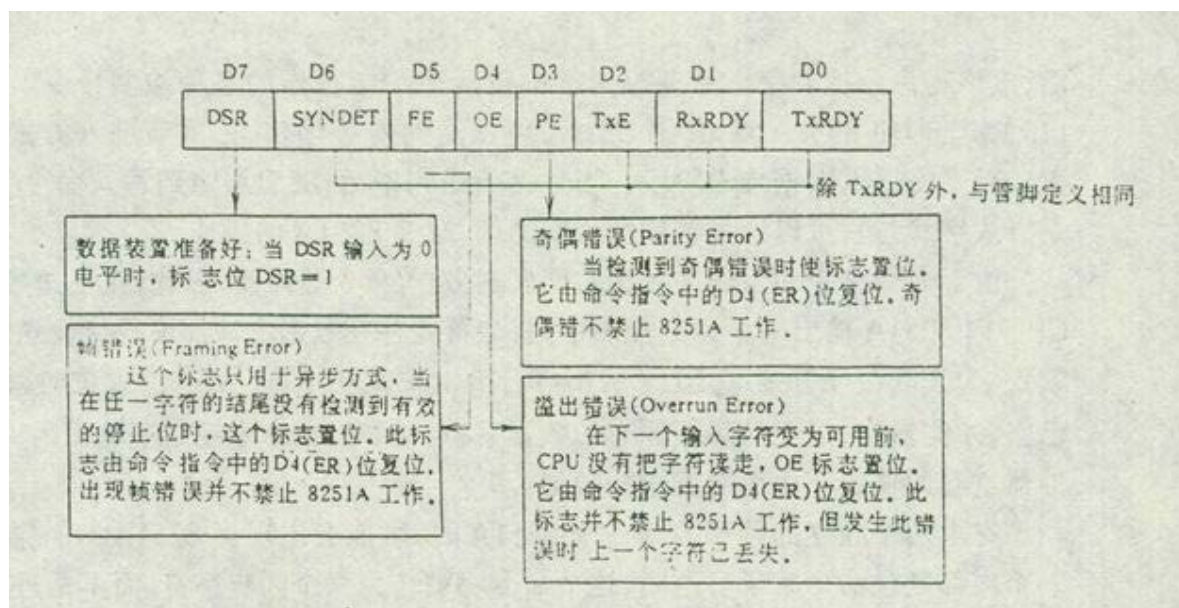


图 9-27 8251A 状态字格式

在发送前和发送后 TxRdy 的状态位和输出引脚的状态可能不一致，在发送过程中二者总是一致的。前者可供 CPU 查询，后者可作为向 CPU 发出的中断申请信号。

状态位的置位比状态的出现总是要滞后后，最坏情况下要延迟 28 个时钟脉冲 (CLK 端)。在读状态的操作过程中，状态位是不变的。

注意：各种控制字发送后，由于内部操作需要一定的时间，因此最好设置几条空操作指令，然后再设置其它指令。

4. 8251A 初始化编程方法

由上所知，8251A 要工作在规定的状态中，必须进行初始化。初始化的过程就是按照方式和命令控制字的格式，向方式寄存器和命令寄存器中写入控制字。由于方式或命令控制字均没有表示其标志的信息位，因此只能依靠不同的端口地址进行区别。但是有的端口地址含有一个不同内容的寄存器，如方式控制字端口地址中还有同步字符寄存器等。这时写入 8251A 的控制字的顺序是非常重要的。对 8251A 初始化流程图如图 9-28 所示。

8251A 初始化编程总是从设置方式控制字开始，随后是命令控制字。方式控制字必须紧跟在复位之后设置。

由图 9-28 可见，当硬件复位或者通过软件编程对 8251A 复位后，便向方式寄存器中写入方式控制字，设置 8251A 工作在同步或异步方式。如果是同步方式，则必须指出同步字符的个数，并随后将同步字符送入 8251A 的同步字符寄存器中。

无论是异步方式还是同步方式，在设置方式控制字之后，应该写命令控制字。命令控制字中包括 8251A 操作的各种控制命令。其中如果 D6 位 (IR) 为 1，即使 8251A 复位，则 8251A 将回到初始化状态，重新进入方式字、命令字的设置。否则将进入数据传送阶段。当数据传送完毕之后，8251A 回到写入命令字状态（注意，

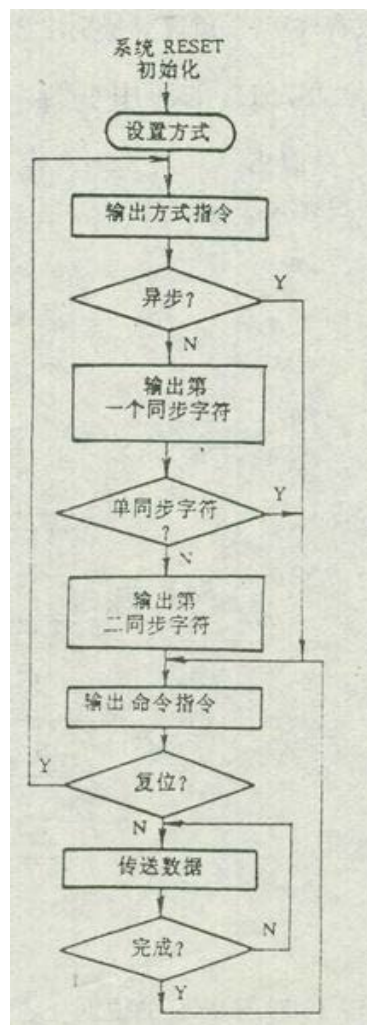


图 9-28 8251A 初始化流程图

不是方式字状态), 可以通过改变命令字的值, 改变 8251A 的操作。

由于命令指令和发送的数据共同发送数据/命令缓冲器, 因此, 在发送数据过程中, 如果 CPU 向 8251A 输出一个命令控制字, 将会覆盖存在数据缓冲器中等待发送的任何字符。这就要求 CPU 必须等到 TxRDY 输出上升沿或出现 TxRDY 状态位置位时, 即确保缓冲器中已空, 才能输出, 以免破坏了原有的数据字符。同时, 在命令控制字输出以后, 必须不等发送缓冲器空立即输出下一个要发送的数据。

在由两个独立的程序控制一个 8251A 时, 可能会出现当 8251A 等待装入同步字符时, 一个内部复位命令来了。这时, 这个命令将被视作一个同步字符而不是进行复位。解决的方法: 在发送复位命令前线发送三个全“0”的命令给 8251A, 使其避开这种可能性。

使用 8251A 时应该注意: 8251A 具有发送连续的 Space 电平的能力。因此 8251A 只能靠接收到一连串字符连续出现帧错误(无停止位)来识别终止符。如果在终止符之后接着接收有效字符, 就需要特别注意识别终止符的最后一个字符。

(六) 8251A 的应用举例

[例 9.3] 用 8251A 作为串行通信接口, 完成 8086CPU 对 CRT 的控制。电路如图 9-29 所示。

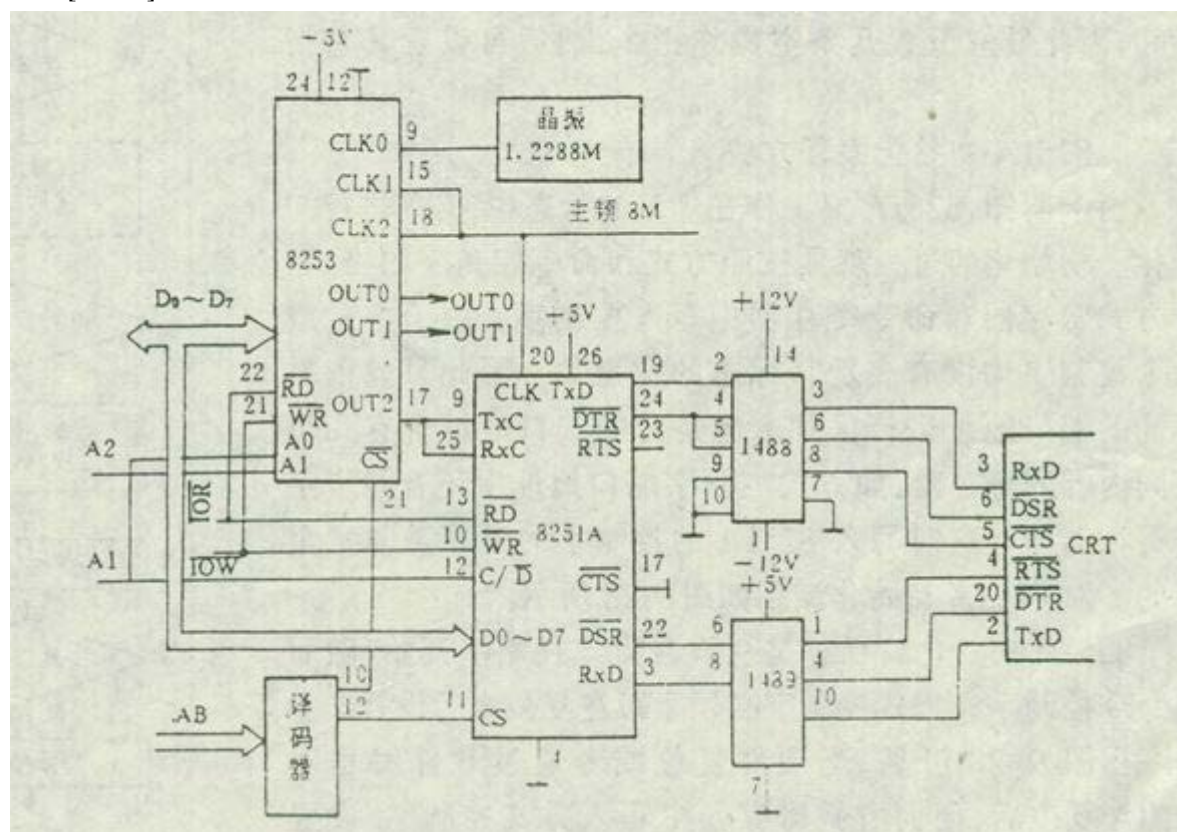


图 9-29 用 8251A 作为 CRT 接口

1. 对图 9-29 的说明

8251A 的主时钟 CLK 是系统时钟 8MHz, 8251A 的发送时钟 TxC 和 RxC 由可编程计数器/定时器 8253 的计数器 2 的输出供给。8251A 的片选信号 \overline{CS} 由译码器供给。读信号 \overline{RD} 和写信号 \overline{WR} 分别由控制总线上的 \overline{IOR} 和 \overline{IOW} 供给。8251A 的数据线 D0—D7 和 8086CPU 的 16 位数据总线的低 8 位 D0—D7 相连。

由于 8251A 的输入输出均为 TTL 电平, 而 CRT 的信号电平是 RS-232-C 的 EIA 电平。所以,

两者之间需有 1488 和 1489 作为电平转换。

2. 程序说明

因为 CRT 是一个输出设备，因此 8251A 是单方向的，即完成由 CPU 送来的数据传输到 CRT 显示。这种传输的方式是一步串行。波特率因子设为 16，8 位数据、1 位停止位。字符数据的传输采用状态查询方式，即每次从 8251A 的状态寄存器中读状态字，判断 TxRDY 状态位的值，如果 TxRDY=1，说明当前数据输出缓冲区为空，CPU 即可向 8251A 发送一个字符。否则，继续测试。

设 8251A 的控制端口地址为 DAH，数据端口地址为 D8H。

程序结构：首先初始化 8251A，其步骤按图 9—28 的顺序进行。当初始化完成后，8251A 的发送器和接收器启动，即可执行字符输出程序。

字符输出程序由状态查询和字符输出两步分组成，当查询到满足传输条件的 TxRDY 状态时，输出一个字符。

注意：在对 8251A 初始化时应首先用硬件或软件复位。这里采用软件复位方法。

3. 程序

①8251A 初始化程序段

```
INIT:  MOV    AX,AX           ;AX 清 0
        MOV    CX,0003
        MOV    DX,00DAH      ;往 8251A 的控制口送 3 个 00
OUT1:  CALL    KKK
        LOOP   OUT1
;
        MOV    AL,40H        ;往 8251A 的控制口送 40H，使它复位
        CALL    KKK
;
        MOV    AL,4EH        ;往 8251A 的控制口送方式字，异步方式，
        CALL    KKK          ;波特率因子为 16，8 位数据，1 位停止位
;
        MOV    AL,27H        ;往 8251A 的控制口送命令字启动发送器和
        CALL    KKK          ;接收器
;
        .....
;
KKK:    OUT     DX,AL          ;输出子程序，将 AL 中数据输出到 DX 制定的端口
        PUSH   CX
        MOV    CX,0002        ;等待输出操作完成
ABC:    LOOP   ABC
;
        POP    CX            ;恢复 CX 的内容，并返回
        RET
```

②CPU 查询 8251A 的状态字，并发送字符程序段。

```
CHAROUT:  MOV    DX,0DAH      ;从状态口 DAAH 读入状态字
STATE:    IN      AL,DX
          TEST    AL,01        ;测试状态位 TxRDY=1? 不是，再测
```

```

JZ      STATE
;

MOV     DX,0D8H      ;DX 指向数据口 0D8H
POP     AX            ;AX 中为要输出的字符
OUT     DX,AL         ;往端口中送出一个字符

```

该例子仅说明一种最基本的查询方式下的 8251A 的使用方法。程序中仅给出了核心部分。读者可以根据需要将该程序补充上必须的内容，形成一个完整的符合汇编程序格式要求的程序。

此外，读者还可以根据前面的介绍和该例的思想，设计出采用中断方式下的字符输出程序（电路稍加改动即可），和能够控制字符的输出个数的程序。

附录二 中断向量地址一览表

中断向量	类型	功能
一、8088 中断向量		
0—3	0	除以零
4—7	1	单步（用于 DEBUG）
8—B	2	非屏蔽中断
C—F	3	断点指令（用于 DEBUG）
10—13	4	溢出
14—17	5	打印屏幕
18—1F	6,7	保留
二、8259 中断向量		
20—23	8	定时器
24—27	9	键盘
28—2B	A	彩色/图形
2C—2F	B	异步通讯（Secondary）
30—33	C	异步通讯（Primary）
34—37	D	硬磁盘
38—3B	E	软磁盘
3C—3F	F	并行打印机
三、BIOS 中断		
40—43	10	屏幕显示
44—47	11	设备检验
48—4B	12	测定存储器容量
4C—4F	13	磁盘 I/O
50—53	14	串行通讯口 I/O
54—57	15	盒式磁带 I/O
58—5B	16	键盘输入
5C—5F	17	打印机输出
60—63	18	BASIC 入口代码
64—67	19	引导装入程序
68—6B	1A	日时钟
四、提供给用户的中断		
6C—6F	1B	Ctrl-Break 控制的软中断
70—73	1C	定时器控制的软中断
五、数据表指针		
74—77	1D	显示器参量表
78—7B	1E	软盘参量表
7C—7F	1F	图形表
六、DOS 中断		
80—83	20	程序结束

84—87	21	DOS 功能调用
88—8B	22	结束退出
8C—8F	23	Ctrl-Break 退出
90—93	24	严重错误处理
94—97	25	绝对磁盘读功能
98—9B	26	绝对磁盘写功能
9C—9F	27	驻留退出
A0—BB	28—2E	DOS 保留
BC—BF	2F	打印机
CO—FF	30—3F	DOS 保留
七、 BASIC 中断		
100—17F	40—5F	保留
180—19F	60—67	用户软中断
1A0—1FF	68—7F	保留
200—217	80—85	由 BASIC 保留
218—3C3	86—F0	BASIC 中断
3C4—3FF	F1—FF	保留

附录三 DOS 功能调用（INT 21H）

功能号	功能	入口参数	出口参数
00	程序终止 (同 INT 20H)	CS=程序段前缀	
01	键盘输入并回显		AL=输入字符
02	显示输出	DL= 输出字符	
03	异步通讯输入		AL=输入字符
04	异步通讯输出	DL=输出数据	
05	打印机输出	DL=输出字符	
06	直接控制台 I/O	DL=FF (输入) DL=字符 (输出)	AL=输入字符
07	键盘输入 (无回显)		AL=输入字符
08	键盘输入 (无回显) 检测 Ctrl-Break		AL=输入字符
09	显示字符串	DS: SS=串首地址 \$'结束字符串	
0A	键盘输入到缓冲区	DS: SS=缓冲区首地址 (DS: SS)=缓冲区最大字符数	DS:(DX+1)=实际输入的字符数
0B	检验键盘状态		AL=00 无键入 AL=FF 有键入
0C	清除输入缓冲区并请求指定的输入功能	AL=输入功能号 (1, 6, 7, 8, A)	
0D	磁盘复位		清除文件缓冲区
0E	指定当前缺省的磁盘驱动器	DL=驱动器号 0=A,1=B,...	AL=驱动器数
0F	打开文件	DS:DX=FCB 首地址	AL=00 文件找到 AL=FF 文件未找到
10	关闭文件	DS:DX=FCB 首地址	AL=00 目录修改成功 AL=FF 目录中未找到文件
11	查找第一个目录项	DS:DX=FCB 首地址	AL=00 找到 AL=FF 未找到
12	查找下一个目录项 (文件名中带*或?)	DS:DX=FCB 首地址	AL=00 找到 AL=FF 未找到
13	删除文件	DS:DX=FCB 首地址	AL=00 删除成功 AL=FF 未找到
14	顺序读	DS:DX=FCB 首地址	AL=00 读成功 =01 文件结束, 记录中无数据 =02 DTA 空间不够 =03 文件结束, 记录不完整
15	顺序写	DS:DX=FCB 首地址	AL=00 写成功

功能号	功能	入口参数	出口参数
			=01 盘满 =02 DTA 空间不够
16	建文件	DS:DX=FCB 首地址	AL=00 建立成功 =FF 无磁盘空间
17	文件改名	DS:DX=FCB 首地址 DS:(DX+1)=旧文件名 DS:(DX+17)=新文件名	AL=00 成功 =FF 未成功
19	取当前缺省 磁盘驱动器号		AL=缺少的驱动器号 0=A,1=B,2=C...
1A	置 DTA 地址	DS:DX=DTA 地址	
1B	取缺省驱动器 FAT 信息		AL=每簇的扇区数 DS:BX=FAT 标识字节 CX=物理扇区的大小 DX=缺少驱动器的簇数
1C	任取一驱动器 FAT 信息	DL=驱动器号	同上
21	随机读	DS:DX=FCB 首地址	AL=00 读成功 =01 文件结束 =02 缓冲区溢出 =03 缓冲区不满
22	随机写	DS:DX=FCB 首地址	AL=00 写成功 =01 盘满 =02 缓冲区溢出
23	测定文件大小	DS:DX=FCB 首地址	AL=00 成功 文件长度填入 FCB
24	设置随机记录号	DS:DX=FCB 首地址	
25	设置中断向量	DS:DX=中断向量	
26	建立程序段前缀	DX=新的程序段的段前缀	
27	随机分块读	DS:DX= FCB 首地址 CX=记录数	AL=00 读成功 =01 文件结束 =02 缓冲区太小，传输结束 =03 缓冲区不满 CX=读取的记录数
28	随机分块写	DS:DX=FCB 首地址 CX=记录数	AL=00 写成功 AL=01 盘数 =02 缓冲区溢出
29	分析文件名	ES:DI= FCB 首地址 DS:SI=ASCIIZ 串 AL=控制分析标志	AL=00 标准文件 =01 多义文件 =FF 非法盘符
2A	取日期		CX=年 DH:DL=月：日（二进制）

功能号	功能	入口参数	出口参数
2B	设置日期	CX:DH:DL=年：月：日	AL=00 成功 =FF 无效
2C	取时间		CH:CL=时：分 DH:DL=秒：1/100 秒
2D	设置时间	CH:CL=时：分 DH:DL=秒：1/100 秒	AL=00 成功 AL=FF 无效
2E	置磁盘自动	AL=00 关闭标志 AL=01 打开标志	
2F	取磁盘缓冲区的首地址		ES:BX=缓冲区首地址
30	取 DOS 版本号		AH=发型号，AL=版号
31	结束并驻留	AL=返回码 DX=驻留区大小	
33	Ctrl-Break 检测	AL=00 取状态 AL=01 置状态(DL) DL=00 关闭检测 =01 打开检测	DL=00 关闭 Ctrl-Break 检测 =01 打开 Ctrl-Break 检测
35	取中断向量	AL=中断类型	ES:BX=中断向量
36	取空闲磁盘空间	DL=驱动器号	成功：AX=每簇扇区数 BX=有效簇数 CX=每扇区字节数 DX=总簇数 失败：AX=FFFF
38	置/取国家信息	DS:DX=信息区首地址	BX=国家码（国际电话前缀码） AX=错误码
39	建立子目录(MKDIR)	DS:DX= ASCIIZ 串地址	AX=错误码
3A	删除子目录(RMDIR)	DS:DX= ASCIIZ 串地址	AX=错误码
3B	改变当前目录 (CHDIR)	DS:DX= ASCIIZ 串地址	AX=错误码
3C	建立文件	DS:DX= ASCIIZ 串地址 CX=文件属性	成功：AX=文件代码 失败：AX=错误码
3D	打开文件	DS:DX= ASCIIZ 串地址 AL=0 读 =2 写 =3 读/写	成功：AX=文件代码 失败：AX=错误码
3E	关闭文件	BX=文件号	失败：AX=错误码
3F	读文件或设备	DS:DX=数据缓冲区地址 BX=文件代号 CX=读取的字节数	读成功 AX=实际读入的字节数 AX=0 已到文件尾 读出错：AX=错误代码
40	写文件或设备	DS:DX=数据缓冲区地址 BX=文件代号 CX=写入的字节数	写成功： AX=实际写入的字节数 写出错：AX=错误代码

功能号	功能	入口参数	出口参数
41	删除文件	DS:DX= ASCIIZ 串地址	成功: AX=0 出错: AX=错误码 (2, 5)
42	移动文件指针	BX=文件代号 CX:DX=位移量 AL=移动方式 (0, 1, 2)	成功: DX:AX=新指针位置 出错: AX=错误码
43	置/取文件属性	DS:DX= ASCIIZ 串地址 AL=0 取文件属性 AL=1 置文件属性 CX=文件属性	成功: CX=文件属性 失败: AX=错误码
44	设备文件 I/O 控制	BX=文件代号 AL=0 取状态 =1 置状态 DX =2 读数据 =3 写数据 =6 取输入状态 =7 取输出状态	DX=设备信息
45	复制文件代号	BX=文件代号 1	成功: AX=文件代号 2 失败: AX=错误码
46	人工复制文件代号	BX=文件代号 1 CX=文件代号 2	失败: AX=错误码
47	取当前目录路径名	DL=驱动器号 DS:SI= ASCIIZ 串地址	(DS:SI)= ASCIIZ 串 失败: AX=错误码
48	分配内存空间	BX=申请内存容量	成功: AX=分配内存首地址 失败: BX=最大可用空间
49	释放内存空间	ES=内存起始段地址	失败: AX=错误码
4A	调整已分配的存储块	ES=原内存起始地址 BX=再申请的容量	失败: BX=最大可用空间 AX=错误码
4B	装配/执行程序	DS:DX= ASCIIZ 串地址 ES:BX=参数区首地址 AL=0 装入执行 AL=3 装入不执行	失败: AX=错误码
4C	带返回码结束	AL=返回码	
4D	取返回代码		AX=返回代码
4E	查找第一个区配文件	DS:DX= ASCIIZ 串地址 CX=属性	AX=出错代码 (02, 18)
4F	查找下一个区配文件	DS:DX= ASCIIZ 串地址 (文件名中带?或*)	AX=出错代码 (18)
54	取盘自动读写标志		AL=当前标志值
56	文件改名	DS:DX= ASCIIZ 串 (旧) ES:DI= ASCIIZ 串 (新)	AX=出错码 (03, 05, 17)
57	置/取文件日期时间	BX=文件代号 AL=0 读取	DX:CX=日期和时间

功能号	功能	入口参数	出口参数
58	置/取分配策略码	AL=1 设置(DX:CX) AL=0 取码 =1 置码(BX)	失败: AX=错误码 成功: AX=策略码 失败: AX=错误码
59	取扩充错误码		AX=扩充错误码 BH=错误类型 BL=建议的操作 CH=错误场所
5A	建立临时文件	CX=文件属性 DS:DX= ASCIIZ 串地址	成功: AX=文件代号 失败: AX=错误码
5B	建立新文件	CX=文件属性 DS:DX= ASCIIZ 串地址	成功: AX=文件代号 失败: AX=错误码
5C	控制文件存取	AL=00 封锁 =01 开启 BX=文件代号 CX:DX=文件位移 SI:DI=文件长度	失败: AX=错误码
62	取程序段前缀地址		BX=PSP 地址

* AH=1—2E 使用 DOS1.0 以上版本;
 AH=1—2E 使用 DOS1.0 以上版本;
 AH=1—2E 使用 DOS1.0 以上版本;
 ASCII 串表示文件路径名 (包括盘符)。
 ASCII 串表示存放文件路径名的缓冲区首地址。

附录四 BIOS 中断调用

INT	AH	功能	入口参数	出口参数
10	0	设置显示方式	AL=00 40×25 黑白方式 =01 40×25 彩色方式 =02 80×25 黑白方式 =03 80×25 彩色方式 =04 320×200 彩色图形方式 =05 320×200 黑白图形方式 =06 640×200 黑白图形方式 =07 80×25 单色文本方式 =08 160×200 16 色图形(PCjr) =09 320×200 16 色图形(PCjr) =0A 640×200 16 色图形(PCjr) =0B 保留(EGA) =0C 保留(EGA) =0D 320×200 彩色图形(EGA) =0E 640×200 彩色图形(EGA) =0F 640×350 黑白图形(EGA) =10 640×350 彩色图形(EGA) =11 640×480 单色图形(EGA) =12 640×480 16 色图形(EGA) =13 320×200 256 色图形(EGA) =40 80×30 彩色文本(CGE400) =41 80×50 彩色文本(CGE400) =42 640×400 彩色文本(CGE400)	
10	1	置光标类型	(CH)0-3=光标起始行 (CL)0-3=光标起始行	
10	2	置光标位置	BH=页号 DH,DL=行, 列	
10	3	读光标位置	BH=页号	CH=光标起始行 DH,DL=行, 列
10	4	读光笔位置		AH=0 光笔未触发 =1 光笔触发 CH=象素行 BX=象素列 DH=字符行 DL=字符列
10	5	置显示页	AL=页号	
10	6	屏幕初始化或上卷	AL=上卷行数 AL=0 整个窗口空白	

INT	AH	功能	入口参数	出口参数
			BH=卷入行属性 CH=左上角行号 CL=左上角列号 DH=右下角行号 DL=右下角列号	
10	7	屏幕初始化或下卷	AL=下卷行数 AL=0 整个窗口空白 BH=卷入行属性 CH=左上角行号 CL=左上角列号 DH=右下角行号 DL=右下角列号	
10	8	读光标位置的字符和属性	BH=显示页	AL=属性 AL=字符
10	9	在光标位置显示字符及其属性	BH=显示页 AL=字符 BL=属性 CX=字符重复次数	
10	A	在光标位置显示字符	BH=显示页 AL=字符 CX=字符重复次数	
10	B	置彩色调板 (320×200 图形)	BH=彩色调板 ID BL=和 ID 配套使用的颜色	
10	C	写像素	DX=行(0-199) CX=列(0-639)(640 个数) AL=像素值	
10	D	读像素	DX=行(0-199) CX=列(0-639)	
10	E	显示字符 (光标前移)	AL=字符 BL=前景色	
10	F	取当前显示方式		AH=字符列数 AL=显示方式
10	13	显示字符串 (适用 AT)	ES:BP=串地址 CX=串长度 DH,DL=起始行, 列 BH=页号 AL=0,BL=属性 串: char,char,... AL=1,BL=属性 串: char,char,... AL=2 串: char,attr,char,attr...	光标返回起始位置 光标跟随移动 光标返回起始位置

INT	AH	功能	入口参数	出口参数
			AL=3 串: char,attr,char,attr...	光标跟随移动
11		设备检验		AX=返回值 bit0=1,配有磁盘 bit0=1,80287 协处理器 bit4,5=01,40×25BW(彩色板) =10,80×25BW(彩色板) =11,80×25BW(黑白板) bit6,7=软盘驱动器号 bit9,10,11=RS-232 板号 bit12=游戏适配器 bit13=串行打印机 bit14,15=打印机号 AX=字节数(KB)
12		测定存储器容量		
13	0	软盘系统复位		
13	1	读软盘状态		AL=状态字节
13	2	读磁盘	AL=扇区数 CH,CL=磁道号, 扇区号 DH,DL=磁头号, 驱动器号 ES:BX=数据缓冲区地址	读成功: AH=0 AL=读取的扇区数 读失败: AH=出错代码
13	3	写磁盘	同上	写成功: AH=0 AL=写取的扇区数 写失败: AH=出错代码
13	4	检验磁盘扇区	同上(ES:BX 不设置)	成功: AH=0 AL=检验的扇区数 失败: AH=出错代码
13	5	格式化磁盘道	ES:BX=磁道地址	成功: AH=0 失败: AH=出错代码
14	0	初始化串行通信口	AL=初始化参数 DX=通讯口号(0,1)	AH=通讯口状态 AL=调制解调器状态
15	0	启动盒式磁带马达		
15	1	停止盒式磁带马达		
15	2	磁带分块读	ES:BX=数据传输区地址 CX=字节数	AH=状态字节 AH=00 读成功 =01 冗余校验错 =02 无数据传输 =04 无引导 =08 非法命令
15	3	磁带分块写	DS:BX=数据传输区地址 CX=字节数	AH=状态字节 (同上)

INT	AH	功能	入口参数	出口参数
16	0	从键盘读字符		AL=字符码 AH=扫描码
16	1	读键盘缓冲区字符		ZF=0 AL=字符码 AH=扫描码 ZF=1 缓冲区空
16	2	取键盘状态字节		AL=键盘状态字节
17	0	打印字符， 回送状态字节	AL=字符 DX=打印机号	AH=打印机状态字节
17	1	初始化打印机回送 状态字节	DX=打印机号	AH=打印机状态字节
17	2	取状态字节	DX=打印机号	AH=打印机状态字节
1A	0	读时钟		CH:CL=时：分 DH:DL=秒：1/100 秒
1A	1	置时钟	CH:CL=时：分 DH:DL=秒：1/100 秒	CH:CL=时：分(BCD) DH:DL=秒：1/100 秒 (BCD)
1A	2	读实时钟 (适用 AT)		DH:DL=秒：1/100 秒 (BCD)
1A	6	置报警时间 (适用 AT)	CH:CL=时：分(BCD) DH:DL=秒：1/100 秒(BCD)	
1A	7	清除报警 (适用 AT)		

附录六 IBM PC ASCII 码字符表

低 四 位 B	高 四 位 H	B								H							
		0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000	0	(null)	▶	◀	0	@	P	'	p	Ç	É	á				∞	≡
0001	1	☺	◀	!	1	A	Q	a	q	ü	æ	í				β	±
0010	2	☹	↑	"	2	B	R	b	r	é	Æ	ó				Γ	≥
0011	3	♥	!!	#	3	C	S	c	s	â	ô	ú				π	≤
0100	4	♦	¶	\$	4	D	T	d	t	ä	ö	ñ				Σ	∫
0101	5	♣	§	%	5	E	U	e	u	à	ò	Ñ				σ	∫
0110	6	♠	■	&	6	F	V	f	v	å	û	ä				ρ	÷
0111	7	•	↓	'	7	G	W	g	w	ç	ù	ó				τ	≈
1000	8	☐	↑	(8	H	X	h	x	ê	ÿ	ï				ø	°
1001	9	○	↓)	9	I	Y	i	y	ë	Ö	ü				θ	•
1010	A	●	→	*	:	J	Z	j	z	è	Ü	ü				Ω	•
1011	B	♂	←	+	;	K	[k	{	ï	ç	½				δ	√
1100	C	♀	└	,	<	L	\	l		î	£	¼				∞	n
1101	D	♪	↔	—	=	M]	m	}	ì	¥	ì				φ	²
1110	E	♫	▲	.	>	N	^	n	~	Ä	Ŕ	«				€	■
1111	F	☼	▼	/	?	O	_	o	Δ	Å	ƒ	»				∩	...