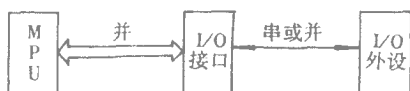


## 第九章 并行接口与串行接口

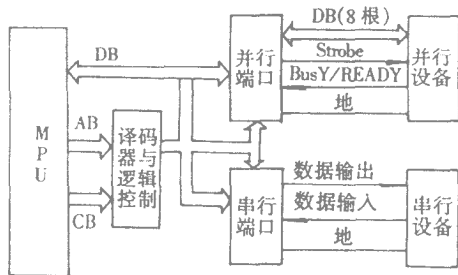
### § 9.1 并行接口与串行接口概述

如前所述,微型计算机与 I/O 设备的接口按照数据传送方式的不同,可分为并行接口和串行接口两种。前者使传送数据的各位同时在总线上传输,后者则使数据一位一位地传输。并行传输又有字并行和字节并行之分,并行接口实现的一般是字节并行传输。

其实所谓的并行和串行,仅是指 I/O 接口与 I/O 设备之间的通信方式一个并行,一个串行。就接口与 MPU 之间的数据交换来说,无论并行接口还是串行接口都是一样的,均以并行方式传送数据,如图 9.1(a)所示。从该图不难想像,两者在结构和功能上的主要差别在于:串行接口需要实行并行和串行之间的相互变换,而并行接口则无需实现这种变换。至于在所用特定的 I/O 端口地址及其使用的地址译码,以及控制逻辑方面,对串行和并行两种接口来说都是类似的。当然,两种接口在接收到一次数据后的工作情况是大不相同的。



(a) 并行和串行的概念



(b) 并行接口和串行接口的结构、功能示意图

图 9.1 并行接口和串行接口的异同

并行接口和串行接口在结构、功能上的异同可从图 9.1(b) 中大体看出。其中并行接口有 8 根数据线通向并行设备,还有两根应答控制线,一根地线,总共 11 根基本连线。串行接口将来自 MPU 的并行数据转换成位数据流,利用一根数据线,每次向串行设备输出一位数据;如果是输入设备,则接口可用类似方法,通过一根数据输入线一位一位地接收来自串行设备的数据,然后并行传给 MPU;公共地线是第 3 根连线。

显然,这两种接口比较起来各有所长。串行接口的优点是所需连线少,无论数据传输宽度是多少,一般只需一根 3 芯电缆,因此实现起来成本低。并行接口的优点则是传输速度快,一次即可同时传送 8 位,以至更多的位。究竟哪种接口更好呢?这不能笼统而言,而要根据计算机与外设之间距离的长短和对信息传输速度要求的不同而决定。通常在速

度要求不高或传输距离较远的场合，例如外部设备是 CRT 显示器、磁带机或远距离数据采集、控制装置时，常选用简单、经济的串行接口。反之，则常选用并行接口。

## § 9.2 可编程并行接口芯片 8255A - 5

并行接口电路有不可编程和可编程两种。前者由数据锁存器和（或）I/O 三态缓冲器组成，电路简单，使用方便。但是，它们在硬件连接好了以后，功能就很难改变，因此，使用不灵活是其主要缺点。而可编程并行接口电路的最大特点恰恰就在于使用灵活，可以在不改变硬件的情况下通过软件编程来改变芯片的功能，因而其功能很强。

8255A ~ 5 是目前应用最广的可编程并行接口芯片之一。它是 8255 和 8255A 的改进型产品，除在某些时间参数上有所缩短，使之能适应更高的工作频率外，其它参数基本相同，芯片功能和使用方法没有差别，相互间可完全兼容。所以，下面讲述时将它们统称为 8255。

### § 9.2.1 内部结构与引脚功能

8255 是一种使用单一的 + 5V 电源、40 引脚双列直插式的大规模集成电路芯片，其内部结构与外部引脚分别如图 9.2 和图 9.3 所示。

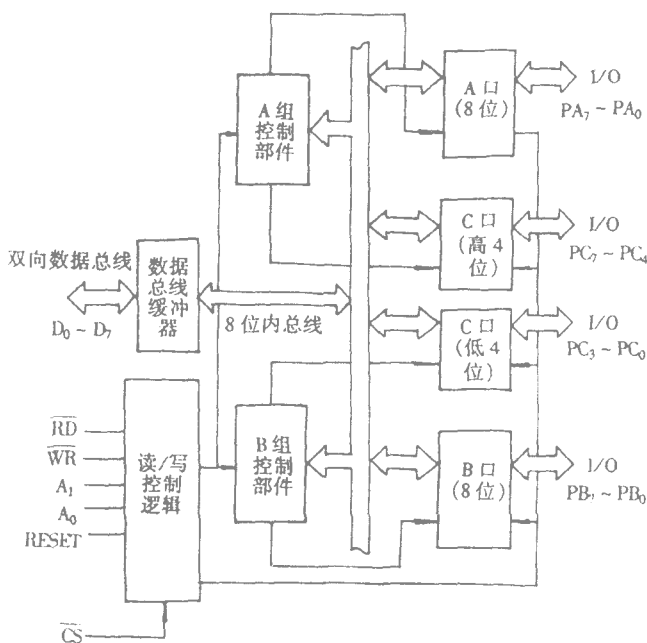


图 9.2 8255 内部结构

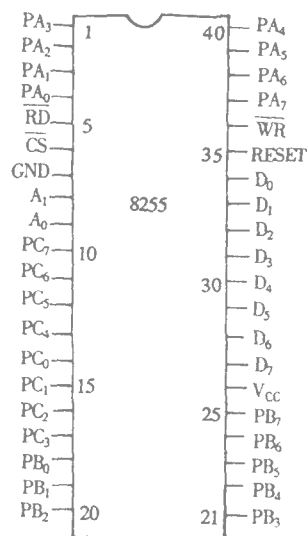


图 9.3 8255 引脚图

从图 9.2 可看出，8255 由三部分组成。

## 1. 与外设接口部分

这部分有 A、B、C 三个 8 位端口寄存器。通过 24 根端口线  $PA_7 \sim PA_0$ 、 $PB_7 \sim PB_0$ 、 $PC_7 \sim PC_0$  与外部设备相连。其中 C 口又分为上半部（高半字节  $PC_7 \sim PC_4$ ）和下半部（低半字节  $PC_3 \sim PC_0$ ）。这 24 根端口线全部为双向三态。B 口和 C 口的引脚具有达林顿复合晶体管驱动能力，在 1.5V 时输出 1mA，所以通常最好选择 B 口或 C 口作输出端口。A、B、C 三个端口的工作方式和 A 口、B 口、C 口上半部、C 口下半部的输入/输出状态可通过程序来选择。8255 有 3 种基本的工作方式，定义如图 9.4 所示。

### (1) 方式 0——基本输入/输出方式

这种方式下，3 个端口分为 A 组（8 位 A 口和 C 口高 4 位）和 B 组（8 位 B 口和 C 口低 4 位），每个端口都可通过编程作为输入口或输出口，作为输入口时都有三态缓冲器功能，作为输出口时都有数据锁存器功能。

### (2) 方式 1——应答式输入/输出方式

这种方式下，A 口和 B 口作为 8 位输入或输出端口，C 口主要为 A、B 两口提供输入/输出的应答信号。这时的 A 口和 B 口，无论输入或输出都有数据锁存功能。

### (3) 方式 2——应答式双向数据传送方式

这种方式下，A 口作为双向数据传送端口，C 口中的 5 位作为相应的应答控制信号，B 口和 C 口中余下的 3 位可工作于方式 0 或方式 1。

在上述 A、B、C 三个端口中，C 口还有位操作功能，即 C 口各位可以分别单独置“1”或置“0”，这就为形成所需的应答控制信号提供了很大的方便。

## 2. 与微处理机接口部分

这部分主要是保证微处理机对 8255 的编程、监视和提供数据通道。

(1) 数据总线缓冲器 这是一个 8 位双向三态缓冲器。所有数据的输入输出，以及对 8255 发的控制字和从 8255 读的状态信息，都是通过这个缓冲器传送的。

(2) 读写控制逻辑 有 6 根控制线，控制 8255 内部各种操作。RESET 用于 8255 内部复位， $\overline{CS}$  和地址线  $A_1$ 、 $A_0$  分别用于选片和片内选端口， $\overline{RD}$  和  $\overline{WR}$  用于控制 8255 数据的读/写。RESET 为高电平有效信号。芯片复位时，片内各寄存器都被清“0”，且 A、B、C 三个端口都被置为输入方式，24 条 I/O 端口线均为“高阻”态。RESET 信号通常与系统复位信号连在一起。 $\overline{CS}$ 、 $A_1$ 、 $A_0$ 、 $\overline{RD}$ 、 $\overline{WR}$  等信号对 8255 内部各寄存器及其操作的选择控制作用如表

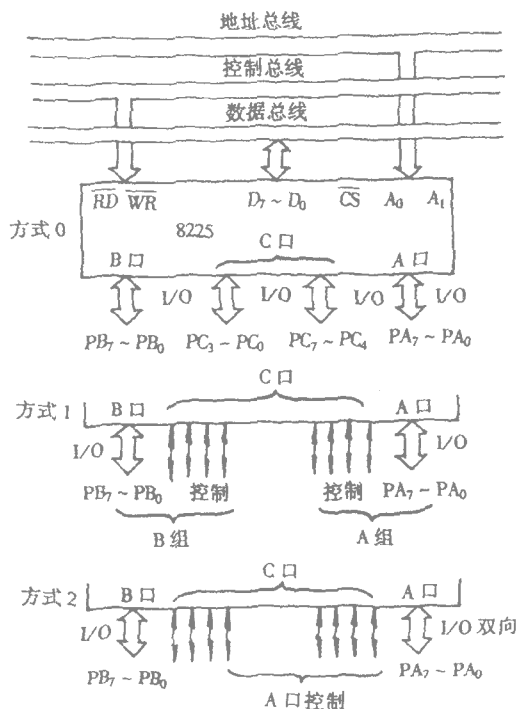


图 9.4 8255 三种基本工作方式

9.1 所示, 8255 与微处理机总线的接口方法如图 9.5 所示。需要说明如下三点:

对于采用专用 I/O 编址方式的 MPU 将  $\overline{IOR}$  和  $\overline{IOW}$  分别接到 8255 的  $\overline{RD}$  和  $\overline{WR}$  而采用存储器映像式 I/O 编址方式的 MPU 则将  $\overline{MEMR}$  和  $\overline{MEMW}$  分别接到  $\overline{RD}$  和  $\overline{WR}$ 。

②地址总线中的  $i$  代表 I/O 端口地址线的根数, 一般将 AB 的最低两位  $A_1$ 、 $A_0$  直接连到芯片的  $A_1$ 、 $A_0$  作为端口选择线, 而将其它高位地址线通过译码形成片选信号。

因为 8255 的 RESET 信号是高电平有效, 所以对于复位信号是低电平有效的 MPU, 如 Z80/Z8000 系列 MPU 和 MC6800/MC68000 系列 MPU 等, RESET 引脚必须经一反相器才能与 8255 的 RESET 引脚相连。

表 9.1 8255 的端口与操作选择表

| $A_1$ | $A_0$ | $\overline{RD}$ | $\overline{WR}$ | $\overline{CS}$ | 操 作                      |    |
|-------|-------|-----------------|-----------------|-----------------|--------------------------|----|
| 0     | 0     | 0               | 1               | 0               | A 口 $\rightarrow$ 数据总线   | 输入 |
| 0     | 1     | 0               | 1               | 0               | B 口 $\rightarrow$ 数据总线   |    |
| 1     | 0     | 0               | 1               | 0               | C 口 $\rightarrow$ 数据总线   |    |
| 0     | 0     | 1               | 0               | 0               | 数据总线 $\rightarrow$ A 口   | 输出 |
| 0     | 1     | 1               | 0               | 0               | 数据总线 $\rightarrow$ B 口   |    |
| 1     | 0     | 1               | 0               | 0               | 数据总线 $\rightarrow$ C 口   |    |
| 1     | 1     | 1               | 0               | 0               | 数据总线 $\rightarrow$ 控制寄存器 |    |
| x     | x     | x               | x               | 1               | 端口输出为“高阻”                | 禁止 |
| 1     | 1     | 0               | 1               | 0               | 非法                       |    |
| x     | x     | 1               | 1               | 0               | 端口输出为“高阻”                |    |

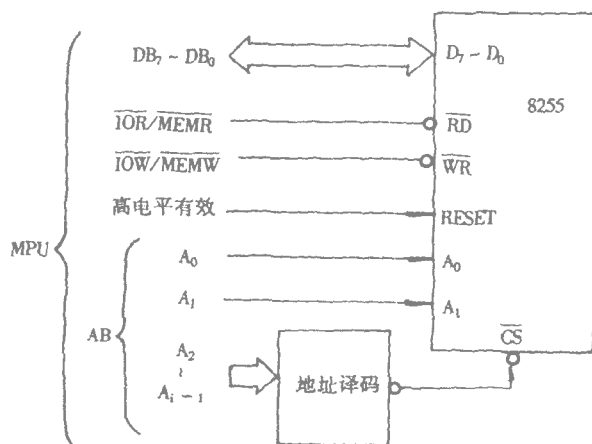


图 9.5 8255 与 MPU 总线的接口方法

### 3. 内部控制部分 (A 组和 B 组控制部件)

这是两组根据 CPU 送来的控制字控制 8255 工作方式和输入/输出状态的控制部件。每组控制部件从读/写控制逻辑接收各种命令, 从内部数据总线接收控制字并发出适当的命令到各自相应的 I/O 端口。它也可以根据 CPU 的命令字对 C 口的每一位实现按位置

“1”或置“0”控制。

A 组控制部件控制端口 A 和 C 口的上半部, B 组控制部件控制端口 B 和 C 口的下半部。实际上, A 组、B 组控制部件就是同一个 8 位的控制寄存器。有些位被分配控制 A 组, 有些位被分配控制 B 组。CPU 用一条输出指令写一个控制字到该控制寄存器, 即可选择和控制 A、B、C 各端口的工作方式。

### § 9.2.2 控制字的确定和初始化编程

8255 的工作方式和接口功能是 CPU 通过把控制字写入控制寄存器来实现的, 通常把这个过程称为初始化编程, 简称为初始化。初始化的基础是根据应用需要正确确定控制字, 因此必须了解 8255 的控制字格式。

#### 1. 工作方式控制字

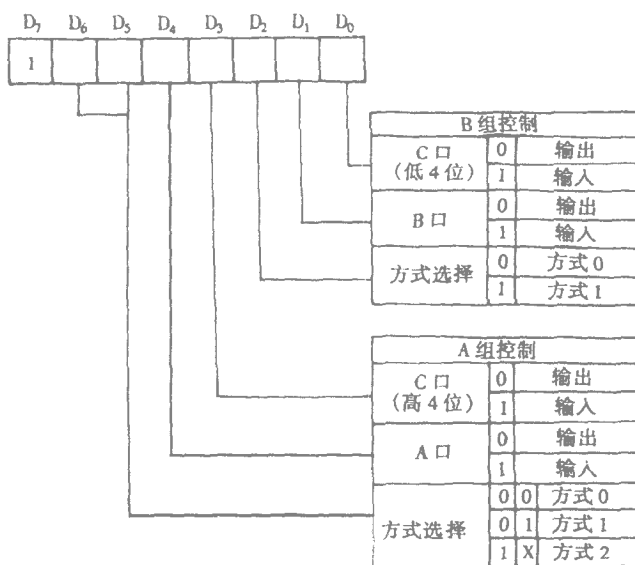


图 9.6 8255 工作方式控制字格式

工作方式控制字的格式如图 9.6 所示, 它规定了控制字各位的含义。  $D_7$  为方式控制字标志位, “1”为有效。  $D_6 \sim D_3$  为 A 组控制位。其中  $D_6 D_5$  用于设置 A 口的工作方式 (A 口有 3 种工作方式),  $D_4$  用于设置 A 口的输入/输出状态,  $D_3$  用于设置 C 口高 4 位的输入/输出状态。  $D_2 \sim D_0$  为 B 组控制位。其中  $D_2$  用于选择 B 口的工作方式 (B 口只有两种工作方式),  $D_1$  用于选择 B 口的输入/输出状态,  $D_0$  用于选择 C 口低 4 位的输入/输出状态。工作方式控制字的这种格式表明:

端口 A 和端口 B 要分别作为一个整体确定工作方式, 而端口 C 则是分成高 4 位、低 4 位两部分分别确定工作方式。端口 A 和端口 B 的工作方式可以不同, 端口 C 的上半部和下半部的工作方式也可以不同。8255 这四部分的工作方式可以任意组合, 这就使 8255 的 I/O 结构有很大的灵活性, 几乎能适应任何一种外部设备的连接需要, 还能满足

同时连接几种不同 I/O 设备的需要。

②虽然 8255 的 I/O 有上述 4 个部分，每部分的工作方式又可以不同，但是所有各个部分的工作方式却是 CPU 用一条输出指令，通过一个控制字写入一个控制寄存器而确定的。这对于简化初始化编程是十分有利的。

## 2. C 口按位置位 / 复位控制字

这是专门用于对 C 口 8 位中任何一位实现置“1”或置“0”的控制字。该控制字的格式如图 9.7 所示，它只使用了 5 位有效位。 $D_7$  为按位置位 / 复位控制字标志位，“0”表示是本控制字。 $D_3D_2D_1$  用于选择 C 口中要置“1”/置“0”的位，000 选  $PC_0$ ，001 选  $PC_1$ ， $\dots$ ，111 选  $PC_7$ 。 $D_0$  用于确定所选  $PC_i$  是置“1”还是置“0”， $D_0 = 1$ ，置“1”； $D_0 = 0$ ，置“0”。 $D_6D_5D_4$  三位未使用，原则上可任意设置，但一般取 000。

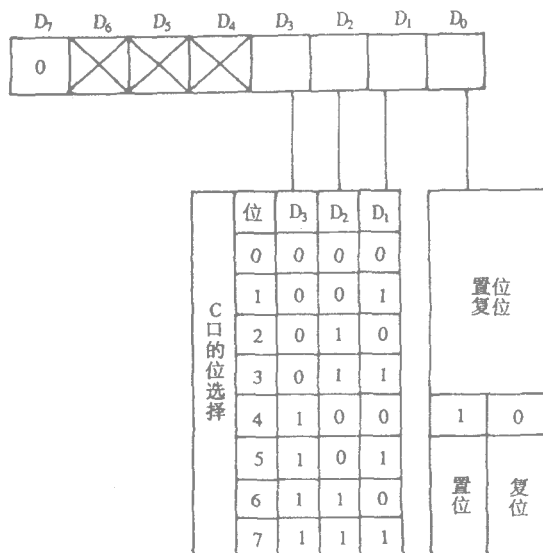


图 9.7 8255 C 口按位置位 / 复位控制字格式

C 口按位置位 / 复位的功能主要用于对外设的控制。利用这一功能，可使 C 口某一位输出一个开关量或一个脉冲，作为外设的启动或停止信号。当端口 A 或 B 工作在方式 1 和方式 2 时，利用这一功能，也可使作为应答控制线的 C 口有关位产生所需的联络信号（脉冲或电平）。这样，显然提高了应答线使用的灵活性，当然在程序上也增加了一些额外的控制步骤。在使用 C 口按位置位 / 复位功能时要注意如下三点：

这一功能可使 C 口的任一位置“1”或置“0”，但一次（一条输出指令，一个控制字）只能使 C 口的一位置“1”或置“0”。如果使 C 口几位都要置“1”或置“0”，必须用几条输出指令，写入几个不同控制字。表 9.2 给出了按位置位 / 复位控制字中的  $D_6D_5D_4$  取 000 时，C 口各位的置位、复位控制字。

②如果要在某位上输出一个开关量信号，对该位的置 1/置 0 操作必须成对使用。比

如要使  $PC_5$  输出一个负跳变信号, 可用下列 80X86 指令:

```
MOV AL, OBH          ;PC5 置“1”控制字送 AL
MOV DX, x x x x x 11B ;控制寄存器口地址送 DX
OUT DX, AL           ;置“1”控制字送控制寄存器
MOV AL, OAH          ;PC5 置“0”控制字送 AL
OUT DX, AL           ;置“0”控制字送控制寄存器
```

表 9.2 8255 C 口按位置位 / 复位控制字

| C 口位   | 置位控制字     |      | 复位控制字     |     |
|--------|-----------|------|-----------|-----|
|        | 二进制       | 十六进制 | 二进制       | 十六制 |
| $PC_0$ | 00000001B | 01H  | 00000000B | 00H |
| $PC_1$ | 00000011B | 03H  | 00000010B | 02H |
| $PC_2$ | 00000101B | 05H  | 00000100B | 04H |
| $PC_3$ | 00000111B | 07H  | 00000110B | 06H |
| $PC_4$ | 00001001B | 09H  | 00001000B | 08H |
| $PC_5$ | 00001011B | 0BH  | 00001010B | 0AH |
| $PC_6$ | 00001101B | 0DH  | 00001100B | 0CH |
| $PC_7$ | 00001111B | 0FH  | 00001110B | 0EH |

如果要在  $PC_5$  输出一个负脉冲, 则在上列程序后面还要增加两条使  $PC_5$  再置“1”的指令 (因为各口初始化之前的状态是置“0”的):

```
MOV AL, OBH
OUT DX, AL
```

③C 口按位置位 / 复位控制字不是送到 C 口地址, 而是送到控制寄存器地址。对此千万不要混淆。

综上所述可以看出, 8255 只有一个控制寄存器地址, 但需要写入该地址的控制字却有两种: 工作方式控制字和 C 口按位置位 / 复位控制字。这两种控制字是以最高位  $D_7$  作为区分的标志,  $D_7 = 1$  为方式控制字,  $D_7 = 0$  为按位置位 / 复位控制字。所以, 控制字值大于、等于 80H 的是方式控制字; 小于 80H 的是按位置位 / 复位控制字, 而其中奇数值是置位控制字, 偶数值是复位控制字。

### § 9.2.3 三种工作方式和接口方法

#### 1. 方式 0 的功能及接口方法

这是一种基本输入输出工作方式, 通常不用应答信号, 或不使用固定的应答线, 不使用中断。在这种工作方式下, 三个端口都可以由程序选定作为输入或输出。

##### (1) 简单的输入输出操作

简单的输入输出适合于: 外设始终有数据可以给微处理器, 以及始终能接受微处理器送来的数据。这种情况下可以进行无条件传送。典型的例子是以开关作为输入, 以指示灯显示作为输出, 如图 9.8 所示。

微处理器通过 8255 的 B 口从外部开关取得数据, 进行处理后, 再通过 8255 的 A 口用发光二极管将处理结果显示。此时的 8255 方式控制字为 10000010(82H)。实现此功能的流程图如图 9.9 所示。

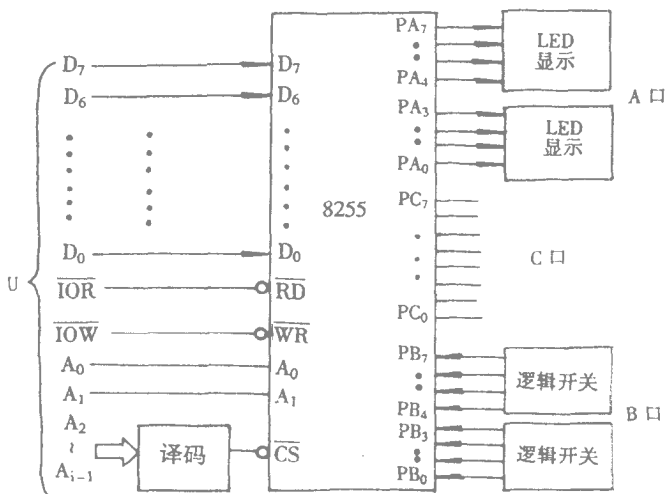


图 9.8 开关输入和显示输出接口

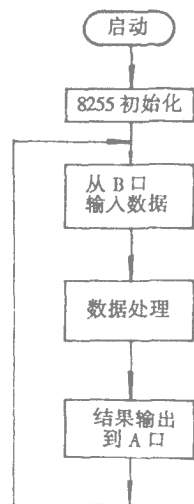


图 9.9 无条件数据传送的流程图

方式 0 输入输出的工作特点是：只要  $\overline{RD}$ （通过输入指令产生）或  $\overline{WR}$ （通过输出指令产生）信号一变低，就进行数据传送，不管此时数据是否稳定。

## (2) C 口的位操作

在方式 0 中，可利用 C 口的按位置位 / 复位功能来产生脉冲，用作门控、选通和复位等。例如用 8255 组成一个脉冲频率计数器，在一秒钟内对输入脉冲计数并把计数值显示出来如图 9.10 所示。用两块 SN7493 四位二进制计数器组成一个 8 位计数器。用 8255 的 C 口第 0 位  $PC_0$  控制计数器复位，第 7 位  $PC_7$  控制计数器的启动和停止，用端口 B 输入计数值，由端口 A 输出显示码。

8255 工作于方式 0 端口 A 为输出，端口 B 为输入，端口 C 为输出。此时方式控制字为 10000010(82H)。而 C 口的  $PC_0$  的置位字为 00000001(01H)， $PC_0$  的复位字为 00H； $PC_7$  的置位字为 0FH，复位字为 0EH。这个频率计数器控制程序如下：

|        |               |                   |
|--------|---------------|-------------------|
| MODE   | EQU 82H       | 方式控制字             |
| PRORTA | EQU 80H       | ; A 口地址           |
| PRORTB | EQU 81H       | ; B 口地址           |
| CNTRL  | EQU 83H       | ; 控制字寄存器地址        |
| START: | MOV AL, MODE  |                   |
|        | OUT CNTRL, AL | ; 设置方式控制字         |
| LOOP:  | MOV AL, 01    |                   |
|        | OUT CNTRL, AL | ; $PC_0$ 置位，计数器复位 |

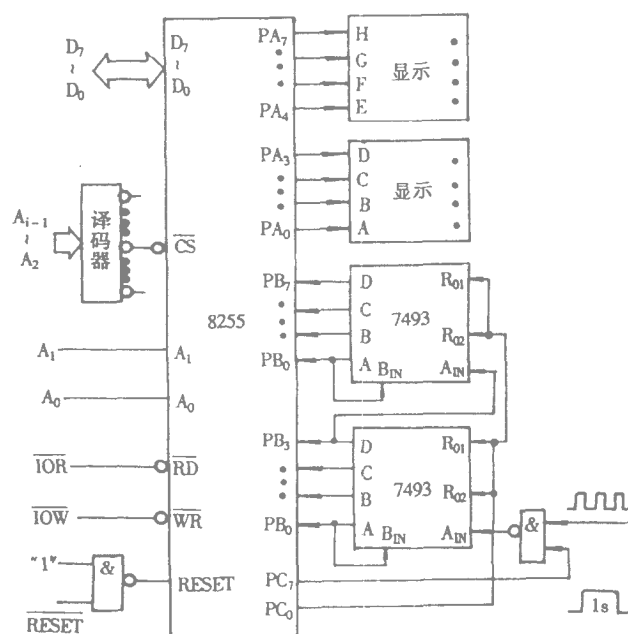


图 9.10 8255 组成的脉冲频率计数器

|      |           |                              |
|------|-----------|------------------------------|
| DEC  | AL        |                              |
| OUT  | CNTRL, AL | ; PC <sub>0</sub> 复位 允许计数    |
| MOV  | AL, 0FH   |                              |
| OUT  | CNTRL, AL | ; PC <sub>7</sub> 置位, 开始输入脉冲 |
| CALL | DELAY     | 延时 1 秒                       |
| DEC  | AL        |                              |
| OUT  | CNTRL, AL | ; PC <sub>7</sub> 复位, 停止输入脉冲 |
| IN   | AL, PORTB | 输入计数值                        |
| OUT  | PORTA, AL | 输出显示                         |
| JMP  | LOOP      |                              |

### (3) 方式 0 的应答式工作

在方式 0 工作时, 也可以用应答查询的方式来同外设交换信息。一般用端口 A 和 B 作为输入或输出的数据通道, 而端口 C 作为应答的控制和状态信息通道, 如 C 口的 PC<sub>4</sub> ~ PC<sub>7</sub> 之一作为选通 (Strobe) STB 线 (输出)、C 口的 PC<sub>0</sub> ~ PC<sub>3</sub> 之一作为外设准备就绪 (Ready) RDY 线 (输入)。这与方式 1 中的固定应答线不同, 可以由程序任意设定。

### 2. 方式 1 的功能及接口方法

方式 0 的应答要占用微处理器的时间, 方式 1 可以用硬件中断方式来进行数据传送, 其效率要比方式 0 的软件应答方式高得多。

#### (1) 方式 1 的输入

在输入时, A 口和 B 口都有 3 条控制线  $\overline{\text{STB}}$ 、IBF 和 INTR。芯片内还有 1 条内部控制线

INTE 如图 9.11 所示。

$\overline{STB}$ 是由外设给 8255 的选通信号,表示外设的数据已准备好,当  $\overline{STB}$ 变为低时,数据锁存入 A 口或 B 口。

IBF 是 8255 给外设的回答信号,表示输入缓冲器满 (Input Buffer Full),即数据已锁存好。它在  $\overline{STB}$ 变低后约 300 毫微秒变为高电平。

INTR 是中断请求。数据锁存后,若此时  $INTE = 1$  即允许中断,则 8255 向微处理器发中断请求。

INTE 是 8255 中断允许,它的位置由  $PC_4$  对 A 口或  $PC_2$  对 B 口的位操作来实现。在方式 1 中,  $PC_4$  或  $PC_2$  的位操作只影响内部 INTE 状态而不影响  $PC_4$  或  $PC_2$  的输出端状态。

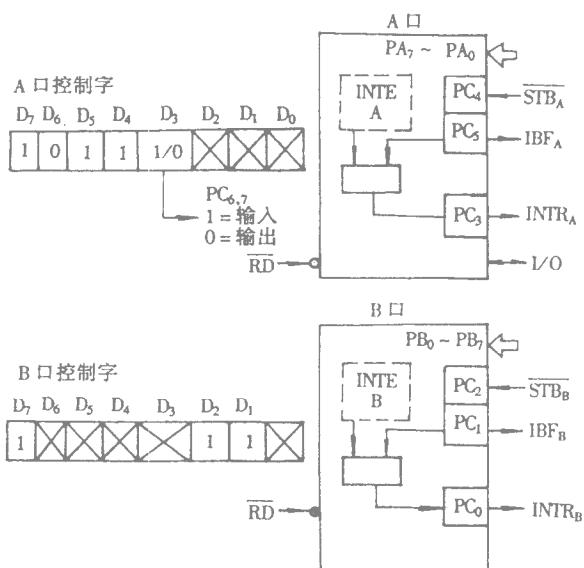


图 9.11 8255 的方式 1 输入

方式 1 输入的时序图如图 9.12 所示。方式 1 的数据输入是从  $\overline{STB}$ 变低开始,它将外设数据锁存入 8255 并使 IBF 变为高送给外设,使外设开始下一数据的准备。此时数据已存入 8255 但尚未被微处理器读入。若  $INTE = 1$ ,则可向微处理器发 INTR 中断请求。微处理器接受中断请求后,转到相应的中断服务程序。在中断服务程序中,执行到输入指令时产生  $\overline{RD}$ 信号将 8255 锁存的数据读入微处理器,同时  $\overline{RD}$ 的下降沿使 INTR 复位,上升沿使 IBF 复位。至此,方式 1 的一字节数据输入操作完成。

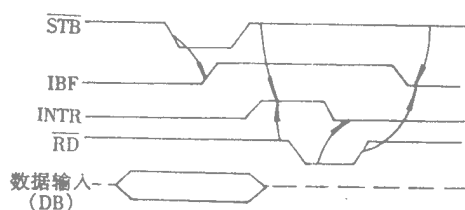


图 9.12 8255 方式 1 输入时序图

## (2) 方式 1 的输出

在输出时, A 口和 B 口也都有 3 条控制线 OBF、 $\overline{ACK}$  和 INTR, 芯片内仍有一条中断允许控制线 INTE, 如图 9.13 所示。

OBF 是输出缓冲器满 (Output Buffer Full)。8255 发出的数据选通信号, 其低电平表示微处理器已将数据送至 8255, 并锁存在相应的端口上。

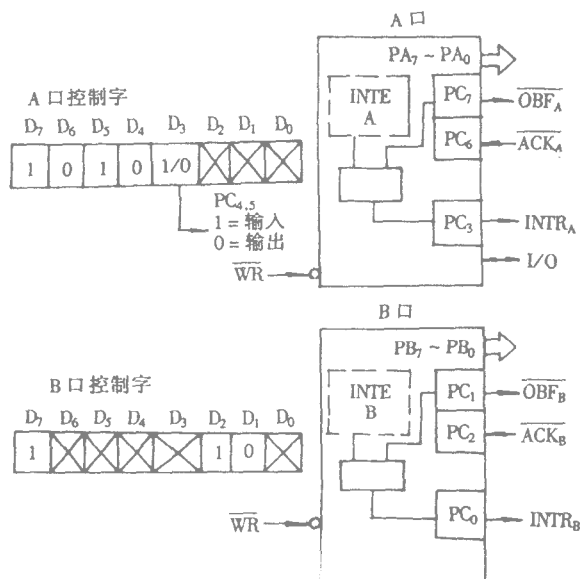


图 9.13 8255 的方式 1 输出

ACK是响应输入 (Acknowledge Input)。这是外设对 8255 发出的 OBF 的响应信号,  $\overline{\text{ACK}}$  为低电平, 表明外设已从 8255 的端口接收到微处理器输出的数据。

INTR 是中断请求。当外设接收到数据后, 若该端口允许中断, 即  $\overline{\text{ACK}}$ 、 $\overline{\text{OBF}}$ 、INTE 均为高电平时, 发出中断请求 INTR 信号。

INTE 是 8255 允许中断, 它由  $\text{PC}_6$  (对 A 口) 或  $\text{PC}_2$  (对 B 口) 的位操作来控制。  $\text{PC}_6$  和  $\text{PC}_2$  的位操作对  $\text{PC}_6$  和  $\text{PC}_2$  引脚状态不产生影响, 只影响  $\text{INTE}_A$  和  $\text{INTE}_B$  的状态。

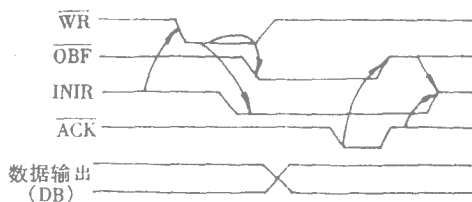


图 9.14 8255 方式 1 输出时序

方式 1 输出的时序图如图 9.14 所示。方式 1 的输出过程是由微处理器响应中断开始的。在中断服务中, 当执行到输出指令时, 产生  $\overline{\text{WR}}$  信号,  $\overline{\text{WR}}$  的下降沿将微处理器数据送到输出数据锁存器, 上升沿将  $\text{INTR}$  置为“0”即撤消中断请求。然后, 产生  $\overline{\text{OBF}}$  信号给外设, 作为输出数据的选通信号。当外设接收到数据后, 便发  $\overline{\text{ACK}}$  响应信号给 8255。当  $\overline{\text{OBF}}$  和  $\text{ACK}$  都为无效时, 即  $\overline{\text{OBF}}$  和  $\text{ACK}$  均为高电平时, 若此时  $\text{INTE} = 1$  则产生  $\text{INTR}$  中断请求信号, 通过中断进行新的数据输出。

(3) 方式 1 的状态字

8255 不能直接提供向量式中断, 因此, 当 8255 申请中断时, 微处理器可以读状态字来确定是哪个端口发出的中断。状态字可以由读 C 口得到 如图 9.15 所示。D<sub>3</sub> 和 D<sub>0</sub> 位分

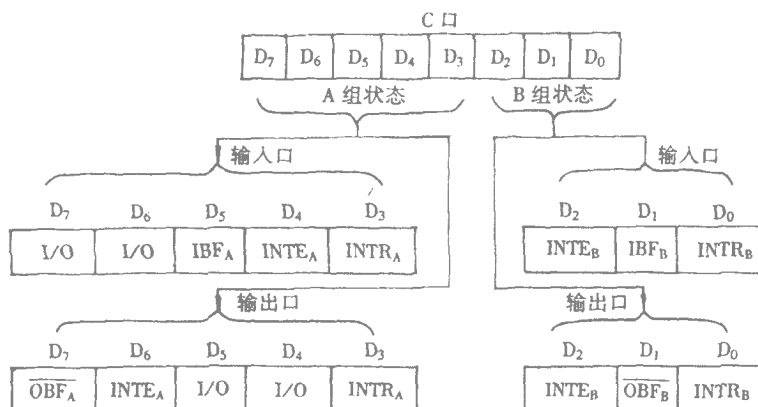


图 9.15 方式 1 的状态字

别表示 A 口和 B 口中断请求  $INTR_A$ 、 $INTR_B$  的状态, 这样可以查询这两位的状态来确定中断源。但要注意, C 口的状态字与 C 口各位引脚上的状态不一样, 如在输入时,  $PC_4$  和  $PC_2$  反映的是  $INTE_A$  和  $INTE_B$  的状态, 而不是  $\overline{STB}$  的状态; 输出时,  $PC_6$  和  $PC_2$  反映的也是  $INTE_A$  和  $INTE_B$  的状态, 而不是  $\overline{ACK}$  的状态。

(4) 方式 1 的接口方法

首先根据应用的具体要求确定 A 口和 B 口是输入还是输出, 然后把 C 口的应答线与外设的控制状态线相连。

方式 1 的中断处理, 可把两个中断请求直接输入, 如图 9.16 所示。图中,  $INTR_A$  接  $IRQ_1$ ,  $INTR_B$  接  $IRQ_3$ , 所以 A 口的输入在  $IRQ_1$  中断服务程序中完成, B 口的输出在  $IRQ_3$  中断服务程序中完成。

若把 A 口和 B 口的中断请求通过或门接到同一中断输入  $INT$ , 那么, 可以在中断服务程序中查询 C 口的状态字来确定产生中断的中断源是 A 口还是 B 口 如图 9.17 所示。

方式 1 的工作方式也可以不用中断处理方式, 而用软件查询应答来进行。要注意, 此时 C 口可供查询的应答信号是输入方式的  $IBF$  和输出方式的  $\overline{OBF}$ ,  $\overline{STB}$  和  $\overline{ACK}$  不能从 C 口读取 (见图 9.15 的 C 口状态字)。图 9.18 是 8255 与控制台打字机的接口图和查询式程序流程图。

### 3. 方式 2 的功能及接口方法

#### (1) 方式 2 的功能

图 9.19 是 8255 工作于方式 2 时 A 口和 C 口的功能, 在方式 2 中, A 口为双向输入输出, C 口的 5 条线 ( $PC_3$ ~ $PC_7$ ) 作为应答控制线。B 口可作为方式 0 或方式 1 工作。

方式 2 输入时的步骤如下: ① 外设发  $\overline{STB}_A$  信号将数据锁存入 A 口; ② 数据锁存后,

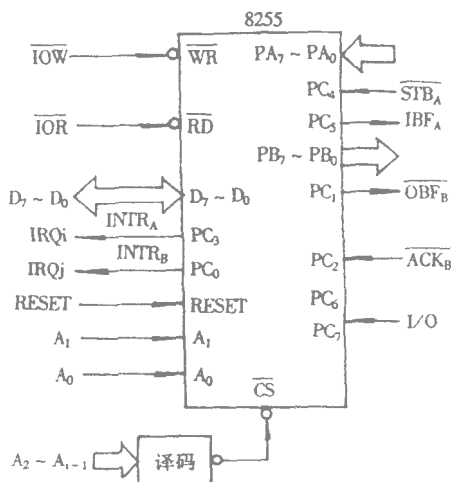


图 9.16 8255 方式 1 中断

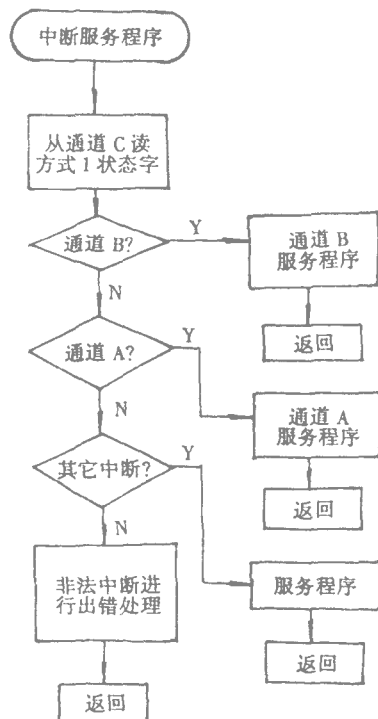


图 9.17 8255 方式 1 中断查询流程图

8255 发  $\overline{\text{IBF}}_A$  给外设；③若  $\text{INTE}_A = "1"$  则 8255 发出  $\text{INTR}_A$  中断请求信号给 CPU；④CPU 通过中断服务程序来读取数据。

方式 2 输出时的步骤如下：①CPU 把数据送至 8255A 口锁存后发  $\overline{\text{OBF}}_A$  至外设；②外设发  $\overline{\text{ACK}}_A$  回答信号给 8255 并将 A 口数据读入外设；③当  $\overline{\text{OBF}}_A$  和  $\overline{\text{ACK}}_A$  变为无效时，若  $\text{INTE}_A = "1"$  则 8255 向 CPU 发  $\text{INTR}_A$  中断请求；④CPU 通过中断服务程序再向 8255 输出数据。

因此，方式 2 的双向工作方式实际上是方式 1 的输入与输出的组合。在输入和输出时都可以产生中断：

$$\text{INTR}_A = \text{IBF} \cdot \text{INTE}_2 \cdot \text{STB} \cdot \text{RD} + \text{OBF} \cdot \text{INTE}_1 \cdot \text{ACK} \cdot \text{WR}$$

### (2) 方式 2 的状态字

在方式 2 工作时，仍可象方式 1 那样采用中断驱动式同步工作方式，通过读取状态字来确定是输入产生中断还是输出产生中断。当然，也可以通过查询  $\overline{\text{OBF}}_A$  和  $\overline{\text{IBF}}_A$  的状态，采用查询式工作方式来实现同步。8255 方式 2 的状态字如图 9.20 所示。

### (3) 方式 2 接口方法

图 9.21 示出了按 8255 方式 2 实现的一种实用主—从机接口方案。8255 在主、从机之间起着数据缓冲器的作用。从机也可以看作是主机的一个智能外设。

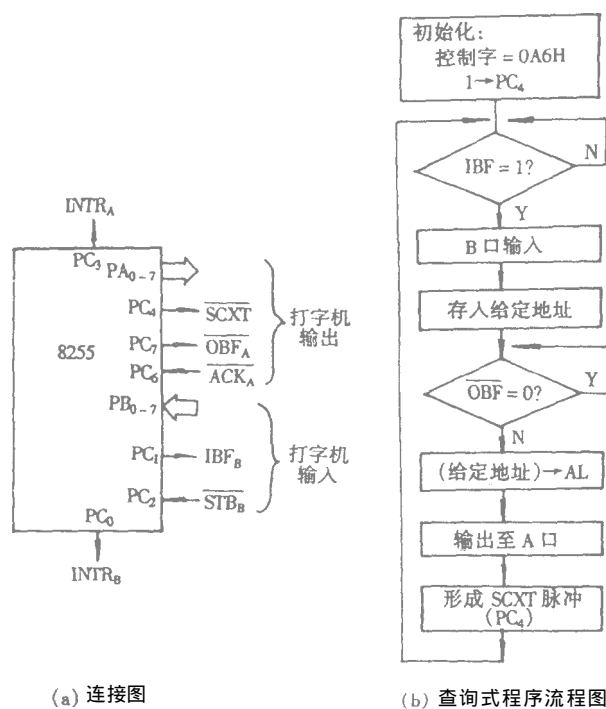


图 9.18 8255 与控制台打字机的查询式接口

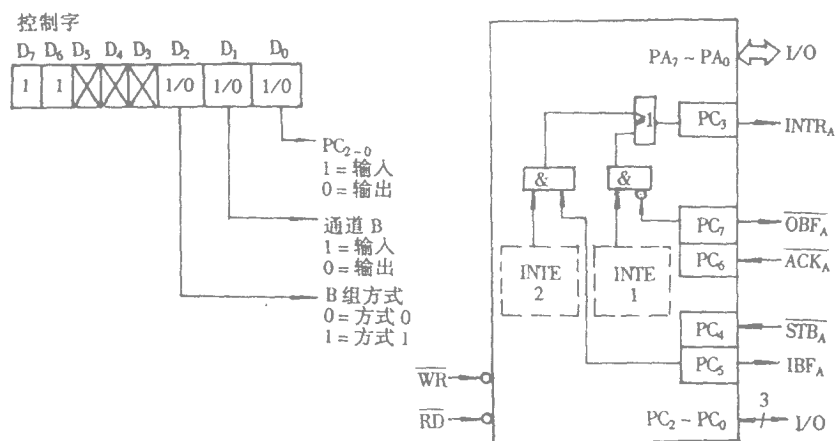


图 9.19 8255 方式 2 的 A、C 口功能

从机通过对 80H 口的 IN 操作来读取 IBF<sub>A</sub> 和  $\overline{\text{OBF}}_A$  状态, 从而判定能否对 20H 口发 OUT 命令来将新的数据送入 A 口 或对 40H 口发 IN 命令来将主机输出至 A 口的数据取出来。就是说, 从机是通过查询的方式来实现与 8255 之间数据传送的同步的。其数据读

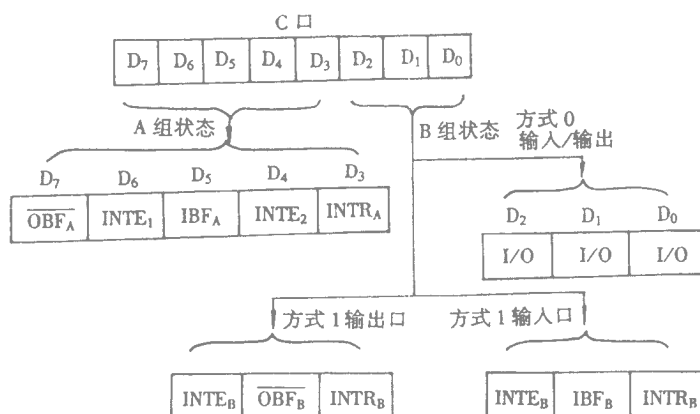


图 9.20 8255 方式 2 状态字

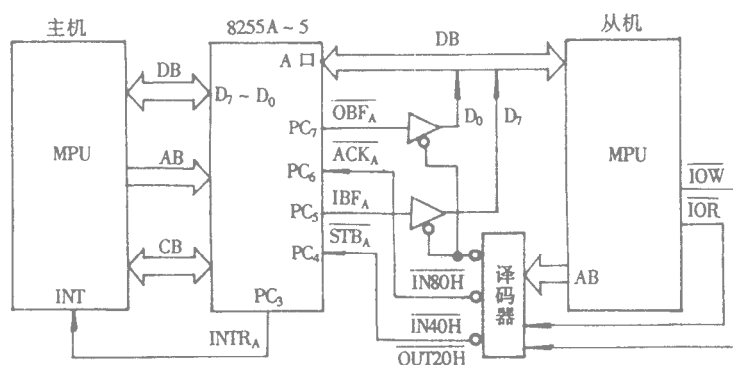


图 9.21 8255 方式 2 实现的主-从机接口

写流程如图 9.22 所示。

主机通过中断方式来同 8255 交换数据。在中断服务程序中，读取 C 口状态字，通过查询 IBF 和 OBF 的状态来决定执行输入操作还是输出操作。为了控制主、从机间数据交换的字节数和待传送字节的地址，可事先在主机存储空间设置一个由若干字节组成的控制区，分别存放这两个参数，如图 9.23 所示。图中设置的控制块为 IOBLOK 开始的连续 3 字节地址单元，其中待传送字节数占 1 个字节，待传送字节的地址占 2 个字节；DATADD 为待传送数据块的首地址。

主机读写数据的程序流程如图 9.24 所示。它包括主程序和中断服务程序两部分：主程序主要完成 8255 初始化和控制块设置，并在此基础上开中断；中断服务程序则是在查询、识别具体中断源的基础上完成数据读入或写出的任务。按图中的流程，读入比写出的优先权高，当然也可反过来，使写出比读入的优先权高。

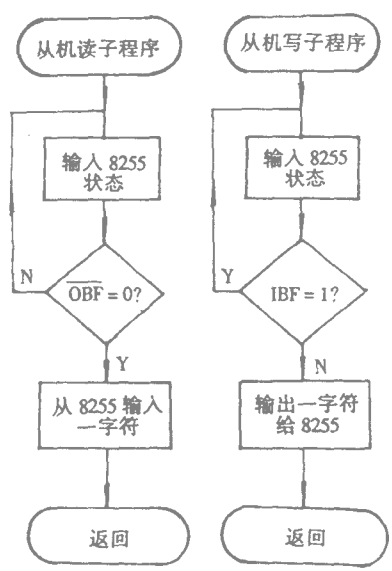


图 9.22 从机读 / 写流程图

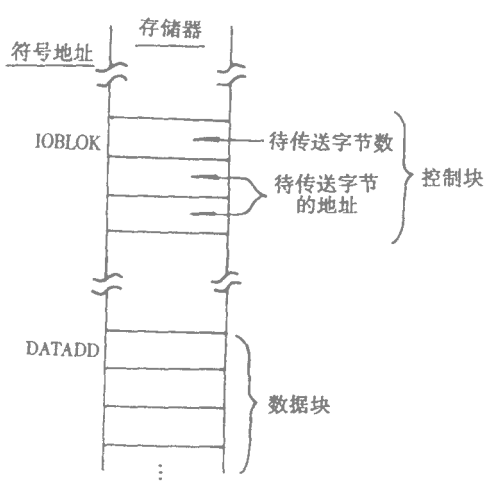


图 9.23 控制区示意图

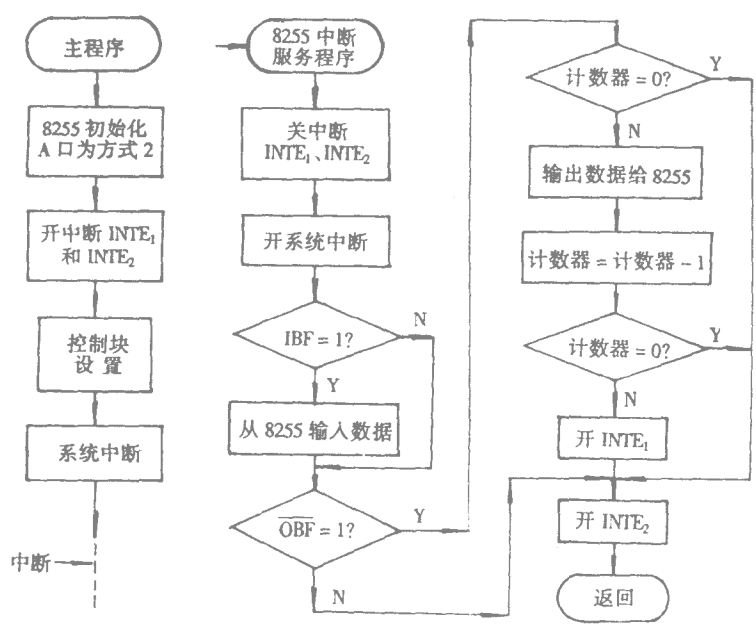


图 9.24 主机读写数据流程图

## § 9.3 高档 PC 系列微机中的并行标准接口

### § 9.3.1 IDE/EIDE 标准接口

IDE 标准又称 ATA(AT - Attachment, AT 嵌入式)接口标准,它最初是作为硬盘控制器的特定接口标准而由 Compaq 公司和 Western Digital 公司合作推出的。它的主要特点是将原来老式 ST - 506 硬盘接口的控制器部分(包括数据编码/译码电路、温盘状态缓冲器、磁头定位控制逻辑等)设置到驱动器中去了,其结果不仅使硬盘适配器逻辑大大简化了,更主要的是有利于驱动器采用不同的磁记录编码方式和采用音圈电机伺服定位方式,以大幅度提高磁道密度和位密度,使磁盘存储空间大大增加,并易于实现更高的数据传输速率,减少数据传输出错率。

IDE 标准接口采用 40 线转插口,信号定义基本上对应于 AT 总线的信号,这也正是将它称作 ATA 接口的原因所在。表 9.3 列出了 IDE 接口标准的转插器引脚信号定义,并与 ISA 总线信号作了比较。不难看出,IDE 信号只不过是 ISA 总线信号的一个子集。其中两个芯片选择信号  $CS_1$ 、 $CS_0$  和三个设备选择信号  $DA_2$ 、 $DA_1$ 、 $DA_0$  由 ISA 总线的地址信号经译码后产生和驱动,用于选择硬盘机的 I/O 端口。一般是 0 号硬盘机的地址为 1FXH,1 号硬盘机的地址为 3FXH,使用硬件可屏蔽中断 IRQ14。IDE 标准中 DMA 方式是一种可选方式,一般不使用而采用 PIO 方式传送数据。支持的 PIO 方式有 PIO0、PIO1、PIO2 三种,它们的最小总线周期分别为 600ns、383ns、240ns。限定每个硬盘机的最大容量为 528MB。

为了适应硬盘的速度不断提高、容量不断增大的需要,硬盘驱动器工业组织以 SFF (Small Form Factor)的名义,于 1993 年在 ATA 标准的基础上推出了一个增强型 ATA 接口标准,即 ATA - 2 标准或称 EIDE 标准,并于其后相继推出了 ATA - 3、Ultra ATA 和 ATA - 4 新标准。

ATA - 2(EIDE)标准新增的功能、性能主要表现在以下几方面:

支持每台硬盘机的最大容量可达 8GB。

② 适配器卡又增加了辅助的第二个 40 脚的插座,以扁平电缆菊花链方式又可接两台硬盘机,端口地址是 17XH 和 37XH,使用 IRQ15 硬件中断。这样,EIDE 的适配器卡最多可连接 4 台硬盘机。

PIO 方式,除原有的 PIO0、PIO1、PIO2 方式外,又增加了快速的 PIO 方式:PIO3 和 PIO4。

④ DMA 方式在 ATA/IDE 标准中是一种可选数据传输方式,新标准将其规范化并分为单字 DMA 方式和多字 DMA 方式。单字 DMA 方式是每次 DMA 请求获得响应后只传送一个 16 位的字,多字 DMA 方式是只要 DMA 请求信号保持有效,就将不断传送 16 位的字,直到终止计数。多字 DMA 方式是一种突发式传送方式。