

## 第8章 定时计数控制接口

定时控制在微机系统中具有极为重要的作用。例如，微机控制系统中常需要定时中断、定时检测、定时扫描等；实时操作系统和多任务操作系统中要定时进行进程调度。IBM PC系列机的日时钟计时、DRAM刷新定时和扬声器音调控制都采用了定时控制技术。

微机系统实现定时功能，主要有三种方法。

1) 软件延时——利用微处理器执行一个延时程序段实现。因为微处理器执行每条指令都需要一定时间，所以程序员通过正确地挑选指令和安排循环次数，很容易编写软件延时程序段；微处理器执行这个程序段就产生一个延时时间。这种软件定时方法在实际中经常使用，尤其是在专用系统上进行软件开发以及延时时间较短而重复次数又有限的时候。软件定时具有不用硬件的特点，但却占用了大量CPU时间；另外，定时精度不高，在不同系统时钟频率下，同一个软件延时程序的定时时间也会相去甚远，这是我们必须注意的。

2) 不可编程的硬件定时——可以采用数字电路中的分频器将系统时钟进行适当分频产生需要的定时信号；也可以采用单稳电路或简易定时电路（如常用的555定时器）由外接RC电阻、电容电路控制定时时间。这样的定时电路较简单，利用分频不同或改变电阻阻值、电容容值，还可以使定时时间在一定范围内改变。

3) 可编程的硬件定时——在微机系统中，常采用软硬件相结合的方法，用可编程定时器芯片构成一个方便灵活的定时电路。这种电路不仅定时值和定时范围可用程序确定和改变，而且具有多种工作方式，可以输出多种控制信号，具备较强的功能。本章学习IBM PC系列机使用的Intel公司的8253和8254可编程定时器。

定时器由数字电路中的计数电路构成，通过记录高精度晶振脉冲信号的个数，输出准确的时间间隔。计数电路如果记录外设提供的具有一定随机性的脉冲信号时，主要反映脉冲的个数（进而获知外设的某种状态），又称为计数器。例如，微机控制系统中往往使用计数器对外部事件计数。因此，人们就统称它们为定时计数器。

### 8.1 8253/8254定时计数器

Intel 8253是可编程间隔定时器（Programmable Interval Timer），同样也可以用作事件计数器（Event Counter）。每个8253芯片有3个独立的16位计数器通道，每个计数器有6种工作方式，都可以按二进制或十进制（BCD码）计数。

Intel 8254是8253的改进型，内部工作方式和外部引脚与8253完全相同，只是增加了一个读回命令和状态字，时钟输入频率8253支持2MHz、8254支持10MHz。所以，后面论述的8253同样适用于8254。

#### 8.1.1 8253/8254的内部结构和引脚

8253的内部结构如图8-1所示，外部采用24引脚双列直插式封装，图8-2是引脚图。

##### 1. 计数器

8253有3个相互独立的计数器通道，每个通道的结构完全相同。每一个计数器通道有一个

16位减法计数器，还有对应的16位预置寄存器和输出锁存器，如图8-1所示。计数开始前写入的计数初值存于预置寄存器；在计数过程中，减法计数器的值不断递减，而预置寄存器中的预置不变。输出锁存器则用于写入锁存命令时锁定当前计数值。

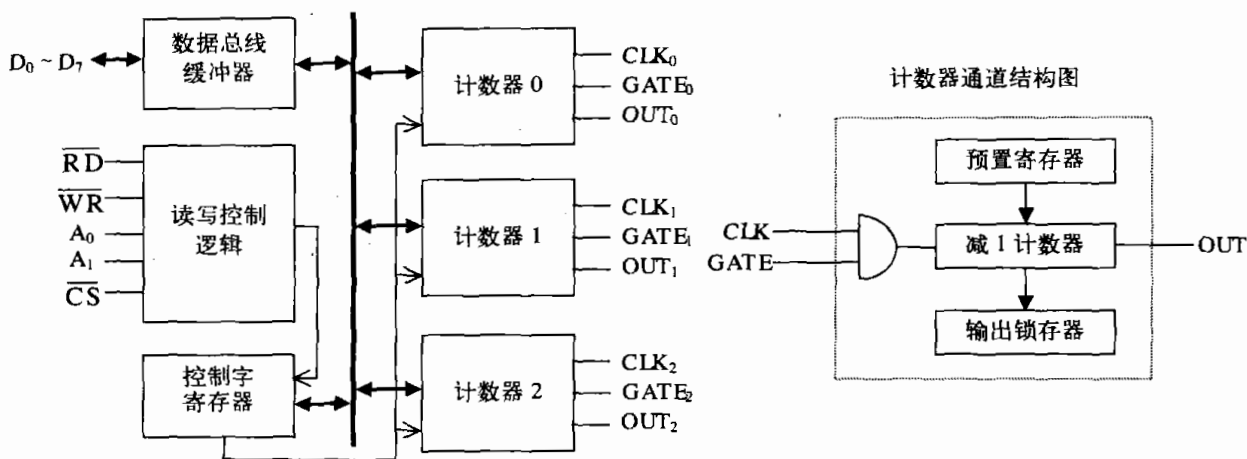


图8-1 8253的内部结构图

8253的每个计数器通道都有3个信号与外界接口。

- CLK时钟输入信号——在计数过程中，此引脚上每输入一个时钟信号（下降沿），计数器的计数值减1。由于该信号通过“与门”才到达减1计数器，所以计数工作受到门控信号GATE的控制。
- GATE门控输入信号——这是控制计数器工作的一个外部输入信号。在不同工作方式下，其作用不同，可分成电平控制和上升沿控制两种类型。
- OUT计数器输出信号——当一次计数过程结束（计数值减为0），OUT引脚上将产生一个输出信号，其波形取决于工作方式。

## 2. 与处理器接口

这部分与8259A相同部分作用一样。数据总线缓冲器用于将8253与系统数据总线相连，接收处理器的控制字和计数值以及发送计数器的当前计数值和工作状态。

读/写逻辑接受来自系统总线的读写控制信号，控制整个芯片的工作。这部分信号有数据线D<sub>0</sub>~D<sub>7</sub>、读信号RD、写信号WR、地址信号A<sub>0</sub>~A<sub>1</sub>和片选CS，其功能见表8-1。

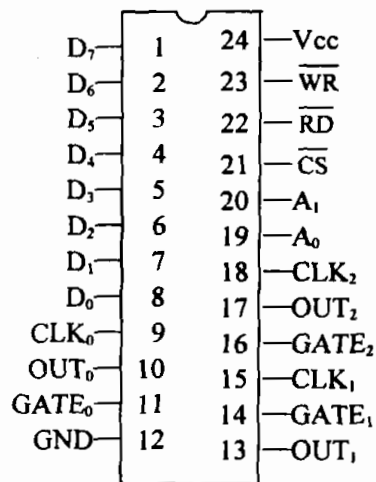


图8-2 8253的引脚图

表8-1 8253的端口选择

CS A <sub>1</sub> A <sub>0</sub>	I/O地址 <sup>①</sup>	读操作(RD)	写操作(WR)
0 0 0	40H	读计数器0	写计数器0
0 0 1	41H	读计数器1	写计数器1
0 1 0	42H	读计数器2	写计数器2
0 1 1	43H	无操作	写控制字

① I/O地址是指PC系列机上8253/8254的I/O地址。

另外，芯片中的控制字寄存器用于保存处理器写入的方式控制字。

### 8.1.2 8253/8254的工作方式

8253有6种工作方式，由方式控制字确定。每种工作方式大致相同，如下所述：

- 1) 微处理器写入方式控制字，设定工作方式。
- 2) 微处理器写入预置寄存器，设定计数初值。
- 3) 对方式1和方式5，需要硬件启动，即GATE端出现一个上升沿信号；对其他方式，不需要这个过程，直接进入下一步，即设定计数值后软件启动。
- 4) CLK端的下一个下降沿，将预置寄存器的计数初值送入减1计数器。
- 5) 计数开始，CLK端每出现一个下降沿（GATE为高电平时），减1计数器就将计数值减1。计数过程要受到GATE信号的控制，GATE为低电平时，不进行计数。
- 6) 当计数值减至0，一次计数过程结束。通常OUT端在计数值减至0时发生改变，以指示一次计数结束。

对方式0、1和4、5，如果不重新设定计数初值或提供硬件启动信号，计数器就此停止计数过程；对方式2和方式3，计数值减至0后，自动将预置寄存器的计数初值送入减1计数器，同时重复下一次的计数过程，直到写入新的方式控制字才停止。

需要注意：处理器写入8253的计数初值只是写入了预置寄存器，之后到来的第一个CLK输入脉冲（需先由低电平变高，再由高变回低）才将预置寄存器的初值送到减1计数器。从第二个CLK信号的下降沿，计数器才真正开始减1计数。因此，若设置计数初值为N，则从输出指令写完计数初值到计数结束，CLK信号的下降沿有N+1个；但从第一个下降沿到最后一个下降沿之间正好又是N个完整的CLK信号；请参见各种工作方式的波形图。

#### 1. 方式0：计数结束中断

当某一个计数通道设置为方式0后，其输出OUT信号随即变为低电平。在计数初值经预置寄存器装入减1计数器后，计数器开始计数，OUT输出仍为低电平。以后CLK引脚上每输入一个时钟信号（下降沿），计数器的计数值减1。当计数值减为0即计数结束时，OUT端变为高电平，并且一直保持到该通道重新装入计数值或重新设置工作方式为止。由于计数结束，OUT端输出一个从低到高的信号，该信号可作为中断请求信号使用，所以方式0被称为“计数结束中断”方式。图8-3为方式0时CLK、GATE和OUT三者的对应关系，图中写信号 $\overline{WR}$ 的波形仅是示意（下同）。

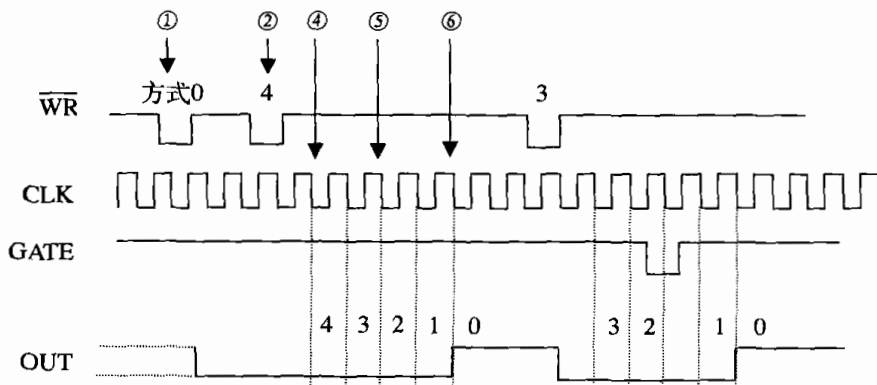


图8-3 工作方式0的波形

GATE输入信号可控制计数过程。当高电平时，允许计数；低电平时，暂停计数。当GATE重新为高电平时，接着当前的计数值继续计数。计数期间给计数器装入新值，则会在写

入新计数值后重新开始计数过程。

### 2. 方式1：可编程单稳脉冲

当CPU写入方式1的控制字之后( $\overline{WR}$ 的上升沿), OUT将为高(若原为低, 则由低变高; 若已经为高, 则不变)。当CPU写完计数值后, 等待外部门控脉冲GATE启动。硬件启动后的CLK下降沿开始计数, 同时输出OUT变低。在整个计数过程中, OUT都维持为低, 直到计数到0, 输出才变为高。因此, OUT端输出一个单稳脉冲。若外部再次触发启动, 则可以再产生一个单稳脉冲, 如图8-4所示。由此可见, 方式1的特点是由GATE触发后, OUT产生一个宽度等于计数值乘时钟周期的单稳负脉冲。

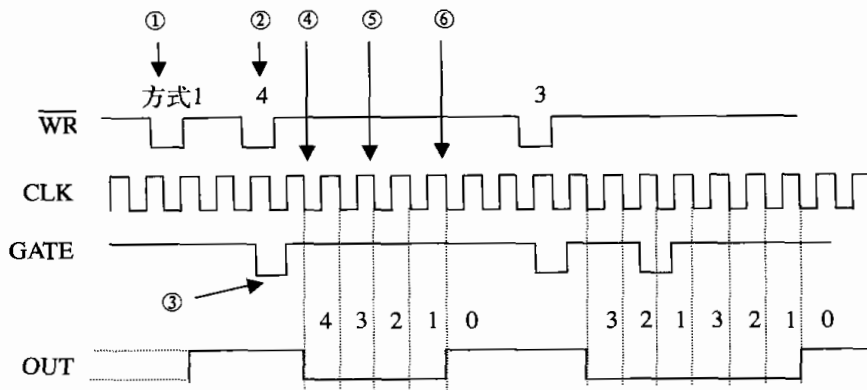


图8-4 工作方式1的波形

计数过程中写入新计数值, 将不影响当前计数; 但若再次由GATE触发启动, 则按新值开始计数。计数过程结束前, GATE再次触发, 则计数器重新装入计数值, 从头开始计数。

### 3. 方式2：频率发生器 (分频器)

当CPU输出方式2的控制字后, OUT将为高。写入计数初值后, 计数器开始对输入时钟CLK计数。在计数过程中, OUT始终保持为高, 直到计数器减为1时, OUT变低。经一个CLK周期, OUT恢复为高, 且计数器开始重新计数, 如图8-5所示。方式2的一个特点是能够连续工作。如果计数值为N, 则每输入N个CLK脉冲, OUT输出一个负脉冲。因此, 这种方式颇似一个频率发生器或分频器。

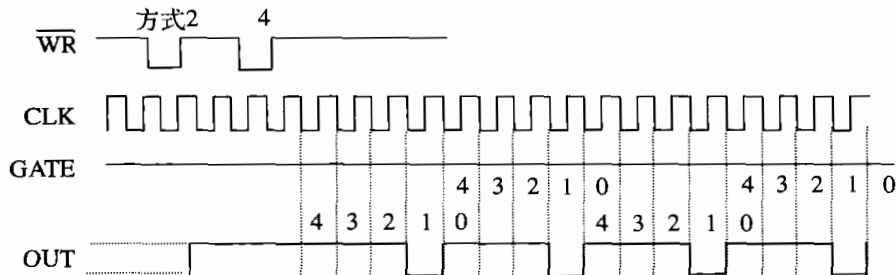


图8-5 工作方式2的波形

计数过程中装入新值, 将不影响现行计数; 但从下个周期开始按新计数值计数。GATE为低电平, 将禁止计数, 并使输出为高。GATE变高电平, 计数器将重新装入预置计数值, 开始计数。这样, GATE能用硬件对计数器进行同步。

### 4. 方式3：方波发生器

方式3和方式2的输出都是周期性的。其主要区别是: 方式3在计数过程中输出的OUT有一半时间为高电平, 另一半时间为低电平。所以, 方式3的OUT输出一个方波。

在这种方式下，当CPU设置控制字后，输出为高；在写完计数值后就自动开始计数，输出仍为高电平；当计数到一半计数值时，输出变为低；直到计数为0，输出又变高，并重新开始计数，如图8-6所示。计数值为偶数时，前半输出为高电平，后半输出为低电平。如果计数值为奇数，前半多一个时钟脉冲的时间输出为高，随后输出为低。

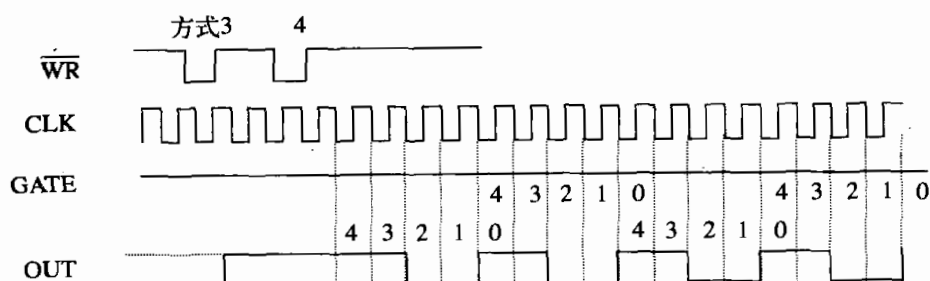


图8-6 工作方式3的波形

#### 5. 方式4: 软件触发选通信号

当处理器写入方式4的控制字后，OUT为高；写入计数值后开始计数（软件启动），当计数值减为0时，OUT变低；经过一个CLK时钟周期，OUT又变高；计数器停止计数。这种方式计数是一次性的，只有在输入新的计数值后，才能开始新的计数，如图8-7所示。

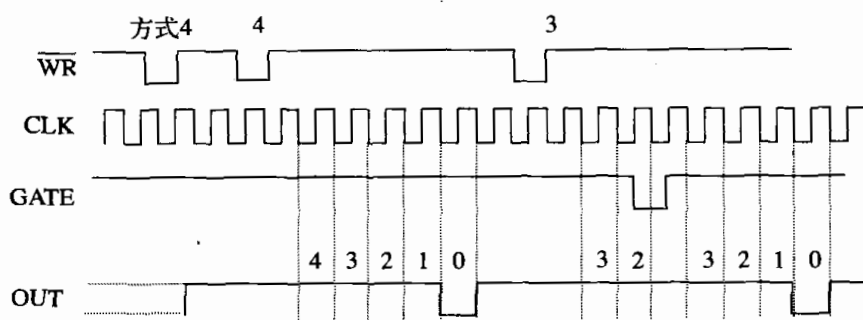


图8-7 工作方式4的波形

计数过程中重新装入新值，将不影响当前计数。GATE为低禁止计数，变为高则计数器重新装入计数初值，开始计数。

#### 6. 方式5: 硬件触发选通信号

当写入方式5的控制字后，OUT为高；写入计数初值后，由GATE的上升沿启动计数过程（硬件启动）。当计数到0时，OUT变低，经过一个CLK脉冲，OUT恢复为高，停止计数，如图8-8所示。

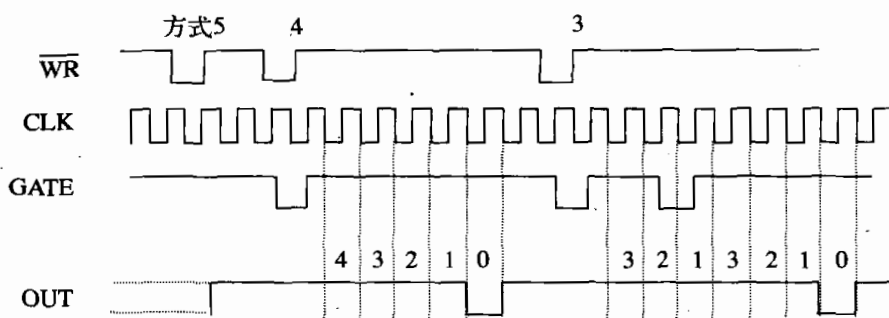


图8-8 工作方式5的波形



计数过程中重新装入新值，将不影响当前计数。GATE又有触发信号，则计数器重新装入计数初值，从头开始计数。

7. 工作方式小结

8253有6种工作方式，它们具有不同的特点。下面做个简单的总结，以便对比。

每种工作方式写入计数值N开始计数后，OUT输出信号都不尽相同，在计数过程中写入新计数值，也将引起输出波形的改变，如表8-2所示。

表8-2 计数值N与输出波形

方 式	N与输出波形的关系	改变计数值
0	写入计数值N后，经N+1个CLK脉冲输出变高	立即有效（写入后下一个CLK脉冲）
1	单稳脉冲的宽度为N个CLK脉冲	外部触发后有效
2	每N个CLK脉冲，输出一个宽度为CLK周期的脉冲	计数到1后有效
3	前一半高电平，后一半低电平的方波	外部触发有效或计数到0后有效
4	写入N后过N+1个CLK，输出宽度为1个CLK的脉冲	计数到0后有效
5	门控触发后过N+1个CLK，输出宽度为1个CLK的脉冲	外部触发后有效

总的来说，GATE信号为低禁止计数、为高允许计数、上升沿启动计数，参见表8-3。

表8-3 门控信号的作用

方 式	GATE		
	低或变为低	上升沿	高
0	禁止计数	—	允许计数
1	—	启动计数，下一个CLK脉冲使输出为低	—
2	禁止计数，即使输出为高	重新装入计数值，启动计数	允许计数
3	禁止计数，即使输出为高	重新装入计数值，启动计数	允许计数
4	禁止计数	—	允许计数
5	—	启动计数	—

8.1.3 8253/8254的编程

8253没有复位信号，加电后的工作方式不确定。为了使8253正常工作，微处理器必须对其初始化编程，写入控制字和计数初值。计数过程中，还可以读取计数值。

1. 写入方式控制字

虽然8253的每个计数器都需要方式控制字，但控制字格式相同，如图8-9所示。而且写入控制字的I/O地址也相同，要求A<sub>1</sub>A<sub>0</sub>=11（控制字地址）。

(1) 计数器选择 (D<sub>7</sub>D<sub>6</sub>)

因为共用一个控制字地址，所以需要两位决定当前控制字是哪一个通道的控制字。

在8253中，D<sub>7</sub>D<sub>6</sub>=11的编码是非法，而8254利用它作为读回命令，详见后面介绍。

(2) 读/写格式 (D<sub>5</sub>D<sub>4</sub>)

8253的数据线为8位，一次只能进行一个字节的交换，但计数器是16位的，所以8253设计了几种不同的读写计数值的格式。

D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>
计数器		读写格式		工作方式			数制
00	计数器 0	00	计数器锁存命令				0 二进制
01	计数器 1	01	只读写低字节				1 十进制
10	计数器 2	10	只读写高字节				000 方式 0
11	非法	11	先读写低字节 后读写高字节				001 方式 1
							×10 方式 2
				×11 方式 3			
				100 方式 4			
				101 方式 5			

图8-9 8253的方式控制字

如果只需要1~256之间的计数值,则用8位计数器即可,这时可以令 $D_5D_4=01$ 只读写低8位,而高8位自动置0。若是16位计数,但低8位为0,则可令 $D_5D_4=10$ ,只读写高8位,低8位自动为0。在令 $D_5D_4=11$ 时,就必须先读写低8位,后读写高8位。

$D_5D_4=00$ 的编码是锁存命令,用于把当前计数值锁存进“输出锁存器”,供以后读取。

### (3) 工作方式 ( $D_3D_2D_1$ )

8253的每个通道可以有6种不同的工作方式,由这三位决定。

### (4) 数制选择 ( $D_0$ )

8253的每个通道都有两种计数制:二进制和十进制(BCD码)。采用二进制计数,读写的计数值都是二进制数形式,例如,64H表示计数值为100。在直接将计数值进行输入或输出时,使用十进制较方便,读写的计数值采用BCD编码,例如,64H表示计数值为64。

例如,已知某个8253的计数器0、1、2端口和控制端口的地址依次是40H~43H。要求设置其中计数器0为方式0,采用二进制计数,先低后高写入计数值。初始化程序段如下:

```
mov al,30h          ;方式控制字:30H=00 11 000 0B
out 43h,al          ;写入控制端口:43H
```

## 2. 写入计数值

每个计数器通道都有对应的计数器I/O地址用于读写计数值。读写计数值时,还必须按方式控制字规定的读写格式进行。

因为计数器是先减1,再判断是否为0,所以,写入0实际上代表最大计数值。选择二进制时,计数值范围为0000H~FFFFH,其中,0000H是最大值,代表65536。选择十进制(BCD码)时,计数值范围为0000~9999,其中0000代表最大值10000。

在上例中,要求计数器0写入计数初值1024 (=400H),初始化程序段接着是:

```
mov ax,1024          ;计数初值:1024 (=400H),写入计数器0地址:40H
out 40h,al           ;写入低字节计数初值
mov al,ah            ;高字节已在AH中
out 40h,al           ;写入高字节计数初值
```

## 3. 读取计数值

利用计数器I/O地址,可以读取计数器的当前计数值。但对8位数据线的8253来说,读取16位计数值需要分两次。由于计数在不断进行,在前后两次执行输入指令的过程中,计数值可能已经变化。所以,如果计数过程可以暂停,可在读取计数值时使GATE信号为低;否则应该将当前计数值先行锁存,然后读取,过程如下:

先向8253写入锁存命令(使方式控制字 $D_5D_4=00$ ,用 $D_7D_6$ 确定锁存的计数器,其他位没有用),将计数器的当前计数值锁存(计数器可继续计数)进输出锁存器。然后,CPU读取锁存的计数值。读取计数值后、或对计数器重新编程,将自动解除锁存状态。读取计数值,要注意设置的读写格式和计数数制。

## 4. 8254的读回命令

8254的改进主要就是多了一个读回命令,这个命令可以令3个通道的计数值和状态锁存,向CPU返回一个状态字,如图8-10所示。

读回命令写入控制端口,状态字和计数值都通过计数器端口读取。如果使读回命令的 $D_5$ 和 $D_4$ 位都为0,即计数值和状态字都要读回。则读取的顺序是:第一次输入指令读取状态字;接着的一条或两条输入指令读取计数值。

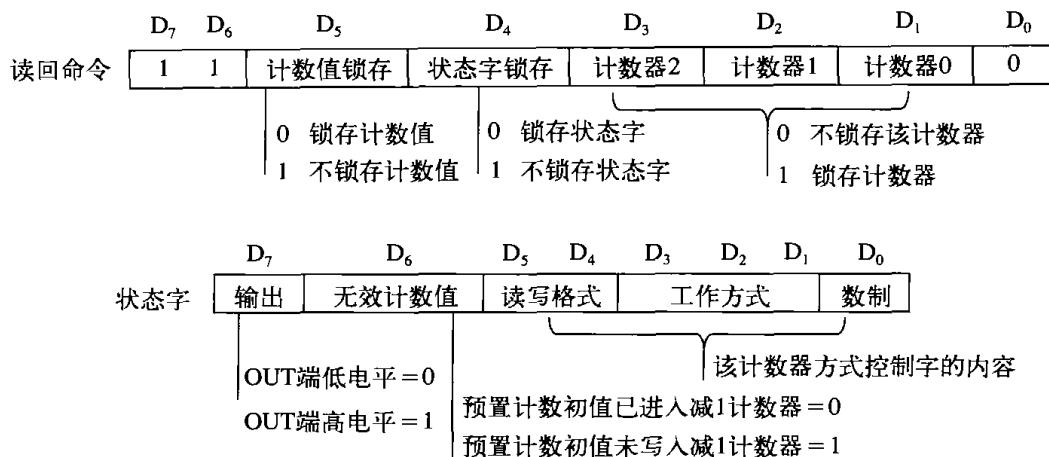


图8-10 8254的读回命令和状态字

## 8.2 8253/8254在IBM PC系列机上的应用

PC系列机使用一片8253/8254。其3个计数通道分别用于日时钟计时、DRAM刷新定时和控制扬声器发声声调，如图8-11所示为其连接图。

根据主机板的I/O地址译码电路可知，当 $A_9A_8A_7A_6A_5=00010$ 时，定时器片选信号有效，所以8253/8254的I/O地址范围为040~05FH。再由片上 $A_1A_0$ 连接方法可知，计数器0、计数器1和计数器2的计数器地址分别为40H、41H和42H，而控制字的端口地址为43H。其他端口地址为重叠地址，一般不使用它们。

3个计数器的时钟输入CLK均连接到频率为1.19318MHz的时钟信号上。

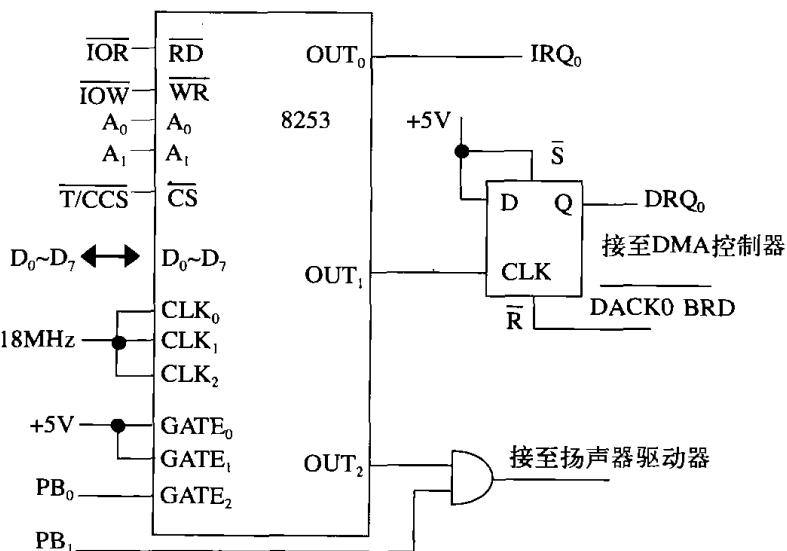


图8-11 PC机上的8253/8254

### 8.2.1 定时中断和定时刷新

首先，让我们阅读一下系统ROM-BIOS对计数器0的初始化编程：

```

mov al, 36h          ; 计数器0为方式3, 采用二进制计数, 先低后高写入计数值
out 43h, al          ; 写入方式控制字
mov al, 0             ; 计数值为0
out 40h, al          ; 写入低字节计数值
out 40h, al          ; 写入高字节计数值

```

由此可知，计数器0采用工作方式3，计数值写入0产生最大计数初值65536，因而OUT输出频率为 $1.19318\text{MHz} \div 65536 = 18.206\text{Hz}$ 的方波信号。结合硬件连接，门控GATE<sub>0</sub>接+5V为常启状态，这个方波信号将周而复始不断产生。OUT<sub>0</sub>端接8259A的IRQ<sub>0</sub>，用作中断请求信号，



即每秒产生18.206次中断请求，或说每隔55ms (54.925493ms) 申请一次中断。

DOS系统利用计数器0这个特点，通过08号中断服务程序（参见习题7.13）实现了日时钟计时功能，即记录18次中断就是时间经过了1秒。

其次，让我们利用计数器1实现DRAM定时刷新请求。

当门控GATE<sub>1</sub>接+5V，为常启状态时，才能重复不断地提出刷新请求；同时应该配合工作方式2或方式3进行重复计数。输出OUT1从低变高使D型触发器置1，Q端输出一正电位信号，作为主存刷新的请求信号；一次刷新结束，响应信号将触发器复位。

PC/XT机要求在2ms内进行128次刷新操作，PC/AT机要求在4ms内进行256次刷新操作。由此可算出每隔 $2\text{ms} \div 128 = 15.6\mu\text{s}$ 必须进行一次刷新操作。这样，将计数器1置成方式2，计数初值为18，每隔 $18 \times 0.838\mu\text{s} = 15.084\mu\text{s}$ 产生一次刷新请求，满足刷新要求，初始化程序如下：

```
mov al,54h      ;计数器1为方式2,采用二进制计数,只写低8位计数值
out 43h,al      ;写入方式控制字
mov al,18       ;计数初值为18
out 41h,al      ;写入计数值
```

## 8.2.2 扬声器控制

PC系列机利用计数器2的输出，控制扬声器的发声音调，作为机器的报警信号或伴音信号。计数器2的OUT输出端接扬声器，只要输出一定频率的方波，经滤波后得到近似的正弦波，就可以推动扬声器发声。

```
                ;发音频率设置子程序,入口参数:AX=1.19318×106÷发音频率
speaker        proc
    push ax      ;暂存入口参数以免被破坏
    mov al,0b6h  ;定时器2为方式3,先低后高写入16位计数值
    out 43h,al
    pop ax       ;恢复入口参数
    out 42h,al   ;写入低8位计数值
    mov al,ah
    out 42h,al   ;写入高8位计数值
    ret
speaker        endp
```

即使完成了计数器2的初始化编程，计数器是否工作仍受控于它的门控信号。GATE<sub>2</sub>接并行接口PB<sub>0</sub>位，即I/O端口地址61H的D<sub>0</sub>位（在XT机中是并行接口电路8255的PB<sub>0</sub>位）。同时，输出OUT<sub>2</sub>经过一个与门，这个与门受PB<sub>1</sub>位控制。PB<sub>1</sub>是I/O端口地址61H的D<sub>1</sub>位（XT机中是8255的PB<sub>1</sub>位）。所以，必须使PB<sub>0</sub>和PB<sub>1</sub>同时为高电平，扬声器才能发出预先设定频率的声音。

```
speakon        proc                ;扬声器开子程序
    push ax
    in al,61h      ;读取61H端口的原控制信息
    or al,03h      ;D1D0=PB1PB0=11B,其他位不变
    out 61h,al     ;直接控制发声
    pop ax
    ret
speakon        endp
speakoff       proc                ;扬声器关子程序
    push ax
    in al,61h
```

```

        and al,0fch          ;D1D0=PB1PB0=00B,其他位不变
        out 61h,al          ;直接控制闭音
        pop ax
        ret
speakoff endp

```

### 例8.1 扬声器声音的控制。

为了方便调用,将频率设置、扬声器“响”与“不响”分别编写成子程序。主程序设置好音调后,让声音出现,用户在键盘上按任何键后声音停止。

```

        ;数据段
freq    dw 1193180/600      ;给一个600Hz的频率
        ;代码段
        mov ax,freq
        call speaker        ;设置扬声器的音调
        call speakon       ;打开扬声器声音
        mov ah,1            ;等待按键
        int 21h
        call speakoff      ;关闭扬声器声音
        .....              ;子程序

```

### 8.2.3 可编程硬件延时

PC机利用主板8253/8254计数器0的定时中断请求,实现日时钟计时。DOS系统有3个有关日时钟的中断。

- 1) 中断向量号08H的日时钟中断服务程序,负责日时钟的跟踪,参见习题7.13。
- 2) 中断向量号1AH的日时钟调用程序,供应用程序读取或设置时间。
- 3) 中断向量号1CH的报时中断,这是在08H号中断服务程序中调用的中断。DOS并没有利用它,只是初始化1CH号中断向量指向一条中断返回指令,参见例7.3。

由于日时钟的计时单位是54.925493毫秒,因此,每秒、每分、每小时的换算关系为:

1秒为18个计时单位:  $54.925493\text{ms} \times 18 = 0.98865887\text{s} \approx 1\text{秒}$

1分为1092个计时单位:  $54.925493\text{ms} \times 1092 = 58.978638\text{s} \approx 60\text{秒}$

1小时为65543个计时单位:  $54.925493\text{ms} \times 65543 = 3598.9815\text{s} \approx 3600\text{秒}$

利用日时钟每隔55ms中断一次不变的特点,可以编写一段不随系统时钟频率变化的固定延时程序。方法是:通过读取日时钟计时变量,把要延时的时间(如5秒)以计时单位形式与当前计时变量相加(如 $5 \times 18 = 90$ )。然后,程序不断读取计时变量,并与期望值相比较,当两值相等时,表示延时时间到,可继续执行操作。

#### 日时钟功能调用

- 中断调用指令: INT 1AH, 子功能号: AH=00H。
- 出口参数: CX. DX=计时变量值(以55ms为单位, CX为高字, DX为低字); AL=0, 未超过24小时。

```

        ;延时开始
        mov ah,0                ;读取日时钟功能调用
        int 1ah
        add dx,90                ;加5秒 (5 × 18 = 90)
        mov bx,dx                ;期望值送bx

```

```

repeat:  int lah                      ;再读日时钟
         cmp bx,dx                    ;与期望值比较
         jne repeat                  ;不等,则循环
         .....                      ;相等,则延时结束

```

看起来,这段延时程序很像软件延时,但实际上利用了硬件定时器,所以它的定时比较准确。

最后,简单说明一下PC/AT以后微机中的实时时钟。实时时钟电路以RT/CMOS RAM为核心,系统断电后由后备电池提供电源,继续计时。CMOS RAM中存放着年月日、时分秒等时间,还有系统的配置信息等内容。当微机系统加电后,系统ROM-BIOS程序读取实时时间,转换成日时钟计时单位,将日时钟与实时时钟同步。1AH号日时钟调用中断的2~7号功能,用于读写实时时间。

实时时钟中断引向从片8259A的IRQ<sub>9</sub>,向量号为70H。实时时钟电路有两个中断源,一个是每隔976.5625 $\mu$ s (1/1024Hz)的周期中断,另一个是报警中断(报警中断的服务程序是4AH号)。但是,微机启动后,周期中断和报警中断都是被禁止的。

由于日时钟中断的时间单位是55ms,所以无法实现更短时间的延时。这时只有利用实时时钟中断。系统ROM-BIOS的15H号中断调用(INT 15H)的86H号子功能为用户实现较短时间的延时提供了方便。

可编程硬件延时中断调用:

- 中断调用指令: INT 15H, 子功能号: AH=86H。
- 入口参数: CX.DX=延时时间(以微秒 $\mu$ s为单位, CX为高字、DX为低字)。
- 出口参数: 标志CF=0表示调用正确、执行了延时; CF=1, 则调用不正确、未执行延时。

虽然以微秒为单位,但实际上该功能调用的实际延时总是976 $\mu$ s的整数倍,因为实时时钟的最小时间单位是976.5625 $\mu$ s (1/1024Hz)。例如,实现约2ms延时的程序段如下:

```

mov cx,0
mov dx,1952          ;延时1.952ms = 2 × 976 $\mu$ s
mov ah,86h
int 15h              ;功能调用返回时,定时时间到

```

### 8.3 扩充定时计数器的应用

由于系统将8253/8254的3个计数器都占用了,所以有时需要扩充定时器芯片。

**例8.2** 利用扩充定时计数器实现对外部事件的计数。

通过PC系列机的系统总线在外部扩充一个8253芯片,利用其计数器0记录外部事件的发生次数,每输入一个高脉冲表示事件发生一次。当事件发生100次后就向CPU提出中断请求(边沿触发)。假设此片8253片选信号的I/O地址范围为200H~207H,则3个计数器和控制I/O地址依次为200H(或204H)、201H(或205H)、202H(或206H)、203H(或207H),如图8-12所示。

```

;8253初始化程序段
mov dx,203h          ;设置方式控制字
mov al,10h           ;设定为工作方式0,二进制计数,只写低字节计数值
out dx,al
mov dx,200h          ;设置计数初值
mov al,64h           ;计数初值为100
out dx,al
.....
;程序其他部分

```

### 例8.3 为A/D转换电路提供可编程的采样信号。

使用一片8253为A/D转换电路提供采样信号，不但可以设置采样频率，还可以决定采样信号的持续宽度，如图8-13所示。假设此片8253片选信号的I/O地址范围为200H~207H，因为将微机系统的地址 $A_2$ 和 $A_1$ 对应与8253的地址线 $A_1$ 和 $A_0$ 连接，所以3个计数器和控制I/O地址依次为200H（或201H），202H（或203H），204H（或205H），206H（或207H）。

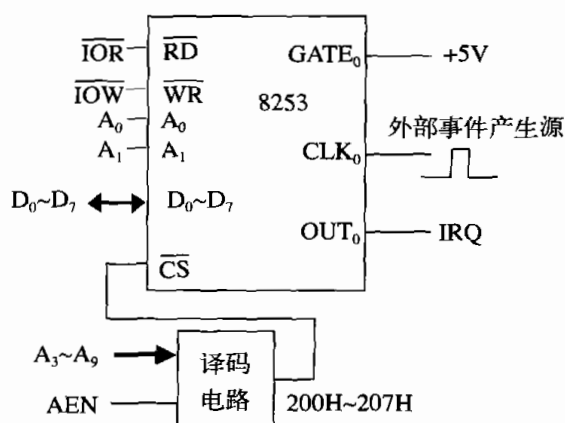


图8-12 扩充8253的连接示意图

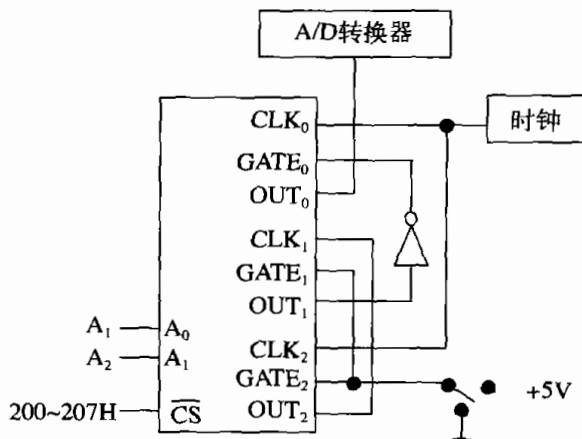


图8-13 8253提供采样信号

让计数器0工作在方式2，计数器1工作在方式1，计数器2工作在方式3。3个计数器的计数初值分别为cnt0、cnt1、cnt2，且都小于256。设时钟频率为F。从图中可以看出，由于将OUT<sub>2</sub>输出作为CLK<sub>1</sub>时钟，所以CLK<sub>1</sub>的频率为 $F/\text{cnt}_2$ ；输出OUT<sub>1</sub>的脉冲周期为 $(\text{cnt}_1 \times \text{cnt}_2)/F$ ；输出OUT<sub>0</sub>的脉冲频率为 $F/\text{cnt}_0$ ，门控信号GATE<sub>0</sub>又受OUT<sub>1</sub>的控制。

当对3个计数器设置好之后，将手动开关（或继电器）从低电平接到+5V，计数器开始工作，输出OUT<sub>0</sub>送A/D转换电路。A/D转换器便按 $F/\text{cnt}_0$ 的采样频率进行工作，每次采样的持续时间为 $(\text{cnt}_1 \times \text{cnt}_2)/F$ 。

；计数器0的初始化编程

```
mov al,14h
mov dx,206h
out dx,al
mov al,cnt0
mov dx,200h
out dx,al
```

；计数器1的初始化编程

```
mov al,52h
mov dx,206h
out dx,al
mov al,cnt1
mov dx,202h
out dx,al
```

；计数器2的初始化编程

```
mov al,96h
mov dx,206h
out dx,al
mov al,cnt2
mov dx,204h
out dx,al
```

## 习题

- 8.1 微机中实现定时控制的主要方法是什么？
- 8.2 8253每个计数通道与外设接口有哪些信号线，每个信号的用途是什么？
- 8.3 8253每个通道有\_\_\_\_\_种工作方式可供选择。若设定某通道为方式0后，其输出引脚为\_\_\_\_\_电平；当\_\_\_\_\_后通道开始计数，\_\_\_\_\_信号端每来一个脉冲\_\_\_\_\_就减1；当\_\_\_\_\_，则输出引脚输出\_\_\_\_\_电平，表示计数结束。8253的CLK<sub>0</sub>接1.5MHz的时钟，欲使OUT<sub>0</sub>产生频率为300KHz的方波信号，则8253的计数值应为\_\_\_\_\_，应选用的工作方式是\_\_\_\_\_。
- 8.4 试按如下要求分别编写8253的初始化程序，已知8253的计数器0~2和控制字I/O的地址依次为204H~207H。
- (1) 使计数器1工作在方式0，仅用8位二进制计数，计数初值为128。
  - (2) 使计数器0工作在方式1，按BCD码计数，计数值为3000。
  - (3) 使计数器2工作在方式2，计数值为02F0H。
- 8.5 设8253计数器0~2和控制字的I/O地址依次为F8H~FBH，说明如下程序的作用。
- ```
mov al,33h
out 0fbh,al
mov al,80h
out 0f8h,al
mov al,50h
out 0f8h,al
```
- 8.6 PC机是如何应用8253每个通道的？
- 8.7 例8.2中CLK<sub>0</sub>端实际输入多少个下降沿后产生中断？按照要求，还可以采用8253的什么工作方式完成同样的功能？如果利用外部信号启动计数，则GATE<sub>0</sub>应怎样使用，应选用什么工作方式？写出初始化程序。
- 8.8 某系统中8253芯片的计数器0~2和控制字端口地址分别是FFF0H~FFF3H。定义计数器0工作在方式2，CLK<sub>0</sub>=5MHz，要求输出OUT<sub>0</sub>=1KHz频率波。定义通道1工作在方式4，用OUT<sub>0</sub>作计数脉冲，计数值为1000，计数器计到0，向CPU发中断请求信号，接于PC系列机IRQ<sub>4</sub>。编写8253两个计数器通道的初始化程序及中断服务程序入口地址、中断屏蔽位设置程序，并画出两个计数器通道的连接图。
- 8.9 利用扬声器控制原理，编写一个简易乐器程序：
- 当按下1~8数字键时，分别发出连续的中音1~7和高音i（对应频率依次为524Hz、588Hz、660Hz、698Hz、784Hz、880Hz、988Hz和1048Hz）。
  - 当按下其他键时暂停发音。
  - 当按下ESC键（ASCII码为1BH），程序返回操作系统。
- 8.10 计数器的定时长度和精度受脉冲输入信号频率和计数值影响。对于频率为f的脉冲输入，计数器输出的最小定时时间为\_\_\_\_\_；此时计数初值应为\_\_\_\_\_。16位计数器输出的最大定时时间是\_\_\_\_\_。当需要加大定时时间时，或者利用硬件方法进行多个计数器的级联；或者利用软件辅助方法，使计数单元扩大计数值。