

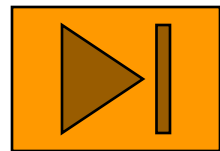
第十七章 定时及音响程序设计

17.1 概述

17.2 8253/8254可编程定时/计数器

17.3 时钟程序设计

17.4 音响程序设计



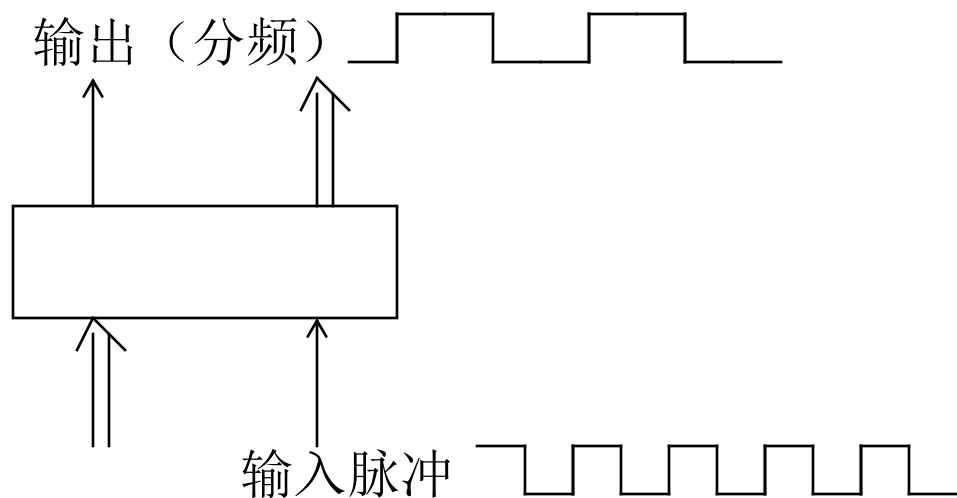
17.1 概述

CPU通过接口电路产生时间符合要求的信号的过程，称为**定时**。该接口电路称为**定时器**。

CPU通过接口电路对外部事件的数量进行统计的过程称为**计数**。实现计数的接口电路称为**计数器**。

在微型机里，把定时器和计数器做成一片大规模集成电路，称为**定时器/计数器**。

17.1 概述



声音的产生:

对输出方波整形 

变成正弦波，经放大处理
接到扬声器上，产生不同
声音的波形。

若选择标准输入脉冲，
例1KHz，则当输入1000个
标准输入脉冲，需要1s，
则当计数值为1000时，输
出端每隔1s输出一个脉冲，
实现了定时。

17.1 概述

实现定时/计数有三种不同的方法：

(1) 软件定时：执行一个具有固定延迟时间的循环程序。

优点：不需外加硬件，灵活，定时较准确。

缺点：在定时过程中CPU不能做任何其它工作。

总结：适用于定时时间短的场合。

(2) 硬件定时：采用中规模TTL或CMOS芯片外加阻容件来实现的。不同的时间间隔主要是通过配接不同的阻容值达到的。

优点：不占用CPU时间。

缺点：变换定时较难。

总结：适用于定时时间间隔固定的场合。

(3) 可编程的定时：综合了上述两种方法的优点，采用固定的硬件，通过编程（写控制字）实现不同的要求。既不占用CPU的时间，又有灵活性。



17.2 8253/8254可编程定时/计数器

17.2.1 8253/8254的基本功能和内部结构

17.2.2 8253的工作方式

17.2.3 8253/8254在IBM PC系列机中的应用

17. 2. 1 8253/8254的基本功能和内部结构

(1) 8253的基本功能

8253有3个独立的16位计数器。每个计数器的主要功能有：

- 1) 按照二进制或二—十进制计数；
- 2) 计数速率可高达2MHz；
- 3) 可以由程序确定按照6种不同的方式工作；
- 4) 所有的输入输出都与TTL兼容。

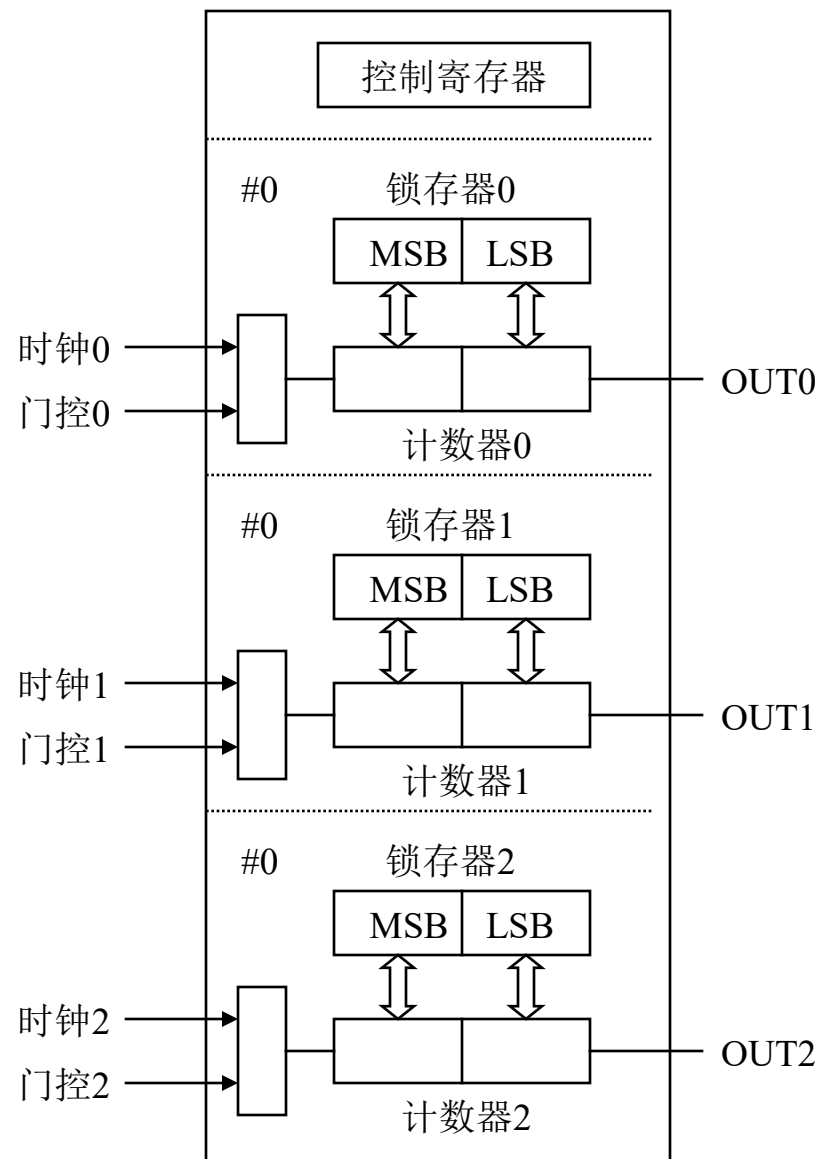
17.2.1 8253/8254的基本功能和内部结构

(2) 8253的内部结构

8253主要由一个控制寄存器和三个计数器及相应的控制电路所组成。

控制寄存器负责对三个计数器设定工作方式。三个计数器的操作是完全独立的，每个计数器都可以对其时钟输入端的始终脉冲按照二进制或二—十进制从预置的初值开始减1计数。当预置的初值减到0时，从OUT输出端输出一个脉冲信号。在计数开始和计数的过程中，计数器可以受门控输入端输入的门控信号的控制。

计数器的初始值必须在开始计数之前，由CPU用输出指令预置。在计数过程中，CPU随时可以用输入指令将任意一个计数器的当前数值读出，而不必中断计数器的时钟输入，也不会影响计数器的继续计数。



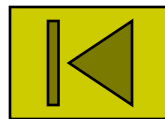
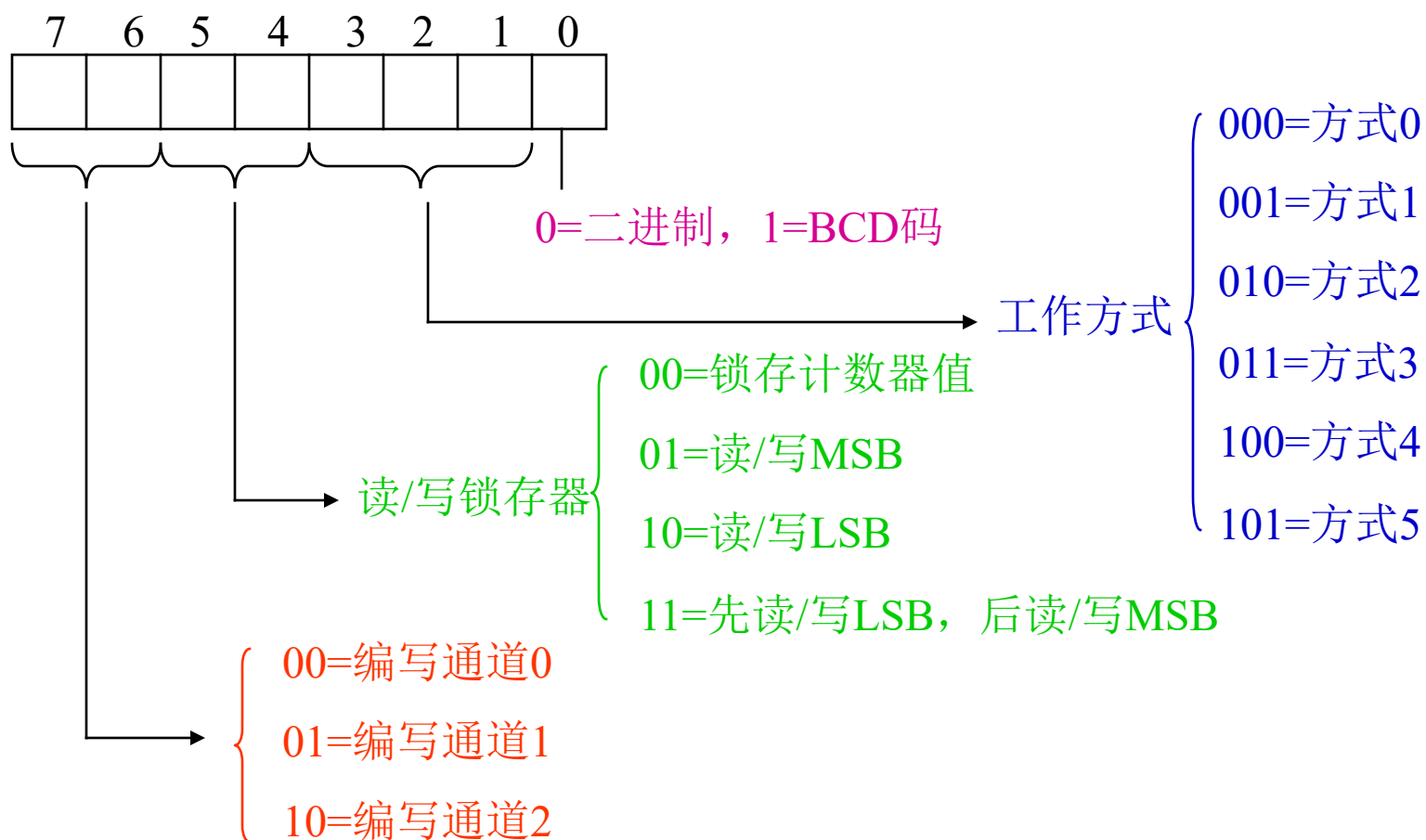
锁存器用来保存写入到计数器的初始值。

计数器是16位的，然而，系统只给其分配一个地址，因此只能按字节进行存取，写入计数值时，要分两次写。

在我们对计数器进行读写时，实际上是读写锁存器的值。因此，当计数到某一时刻时，若想要了解计数器的值，则要先将计数器的值送到锁存器中，然后再读写锁存器的值。

17.2.1 8253/8254的基本功能和内部结构

(3) 8253的控制字



17.2.2 8253的工作方式

8253有6种工作方式：方式0~方式5。对于每一种工作方式，由时钟输入信号CLK确定计数器递减的速率。门控信号GATE用于允许或禁止计数器计数。计数结束时在输出线OUT上产生一个信号。

无论采用哪一种工作方式，都会遵循下面几条原则：

(1) 控制字写入计数器时，所有的控制逻辑电路立即复位，输出端OUT进入初始态（高电平或低电平）。

(2) 计数初值写入后，要经过一个时钟上升沿和一个下降沿，计数执行部件才开始计数。

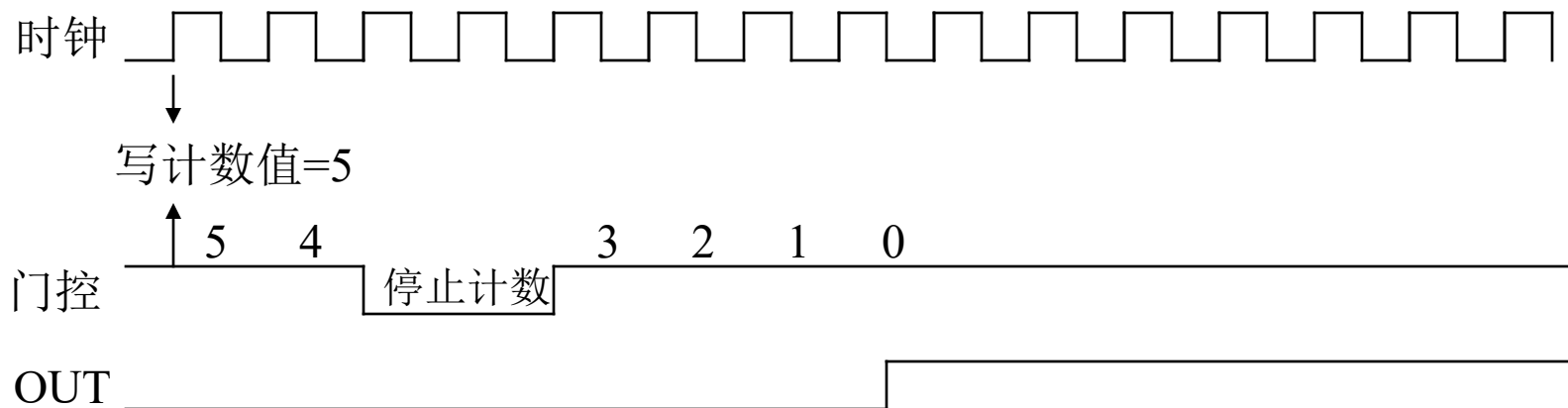
(3) 在时钟脉冲CLK的上升沿时，门控信号被采样。

(4) 在时钟脉冲CLK的下降沿时，计数器作减1计数，输出端OUT的波形也都是在时钟周期的下降沿时产生电平的变化。

17.2.2 8253的工作方式

(1) 方式0—计数结束时中断

当控制字写入控制寄存器后，计数器的输出OUT立即变低，即使未给计数器赋予初值，也未开始计数，只要方式0—确定，输出就为低。计数值送入计数器且门控信号为高时，计数器开始计数。在计数器开始计数和整个计数过程中，OUT都保持为低，直到计数到0，OUT才变高。



17.2.2 8253的工作方式

方式0的工作特点是：

（1）计数器只计一遍。当计数到0时，计数器并不恢复初始值重新开始计数（输出保持为高），直到CPU又写入一个新的计数值，输出立即变低，计数器按照新的数值开始计数。

（2）方式0的计数过程可由门控信号控制暂停。当门控信号为低时，计数器暂停计数。门控信号变高后，计数器继续计数。在计数过程中，GATE信号的变化不影响OUT输出的状态。

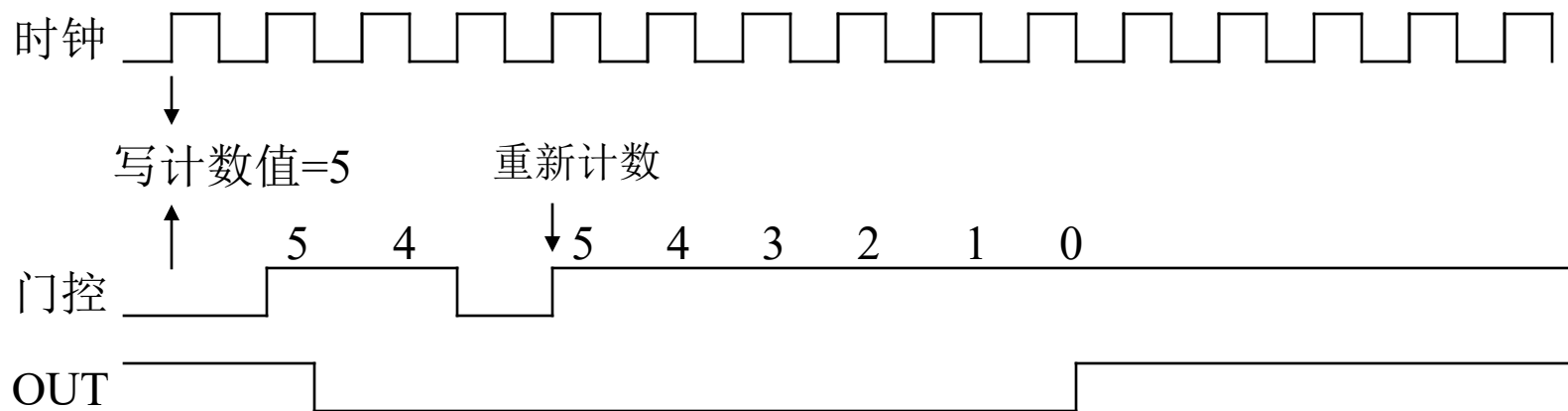
（3）在计数的过程中，可重写或改变计数值，写入新的计数值后，计数器将按新的计数值重新开始计数，改变计数值立即有效。

17.2.2 8253的工作方式

(2) 方式1—可编程的单拍脉冲

这种方式是由外部门控脉冲（硬件）触发启动定时或计数，使输出变低，单拍脉冲开始，在经过固定的时间间隔或计数之后，输出变高，单拍脉冲结束，硬件再次触发，输出可再发一个同样的单拍脉冲。

门控信号变低，可停止计数，门控信号变高后，重新开始计数。



17.2.2 8253的工作方式

方式1的工作特点是：

(1) 写入控制字后，输出端OUT立即变低，知道外部门控信号GATE的上升沿启动之后的CLK下降沿开始计数，OUT变低，并在计数到达0之前一直维持低电平。当计数到0时，OUT变为高电平，并在下一次门控信号触发启动后的第一个时钟脉冲到来之前一直保持高电平。若外部再次触发启动，则可在OUT上再次产生一个单拍脉冲。

(2) 当计数到0后，可再次由外部门控信号GATE的上升沿触发启动。

(3) 即使在计数的过程中，也可用门控脉冲进行再触发，在再触发脉冲上升沿之后的一个CLK脉冲的下降沿，计数器将重新开始计数，而不管原来计数到什么地方。

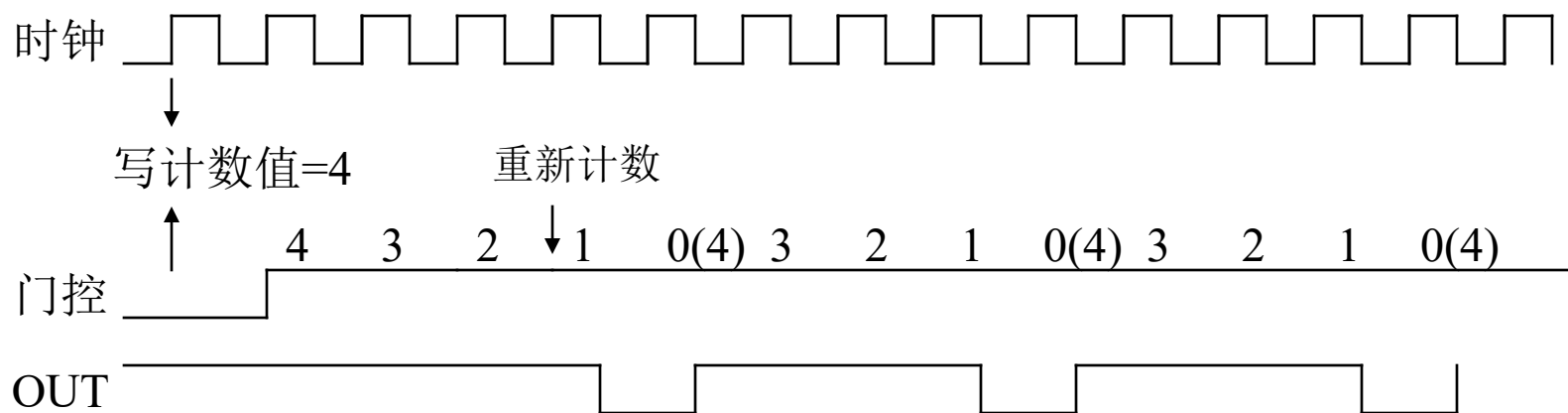
在再触发时，输出的状态将保持不变。在任何一次再触发之后，输出OUT都将继续保持低电平。

(4) 在计数的过程中，CPU可改变计数值，这时候计数过程和输出都不受影响，计数到0后，输出为高。若再次触发计数器，计数器将按新输入的计数值计数。

17.2.2 8253的工作方式

(3) 方式2—速率发生器

这种方式的功能如同一个N分频计数器，输出是输入时钟按照N计数值分频后的一个连续脉冲。如果计数值为N，结果是输入N个脉冲，输出一个脉冲。



17.2.2 8253的工作方式

方式2的工作特点是：

(1) 当CPU输出控制字后，OUT将为高。在写入计数值后的下一个CLK开始减1计数。在计数过程中，OUT保持为高。直到计数值减至1时，OUT变为低，保持一个CLK周期，又恢复为高，重新开始计数。

(2) 不用重新设置计数值，通道能够连续工作，输出端送出固定频率的脉冲。

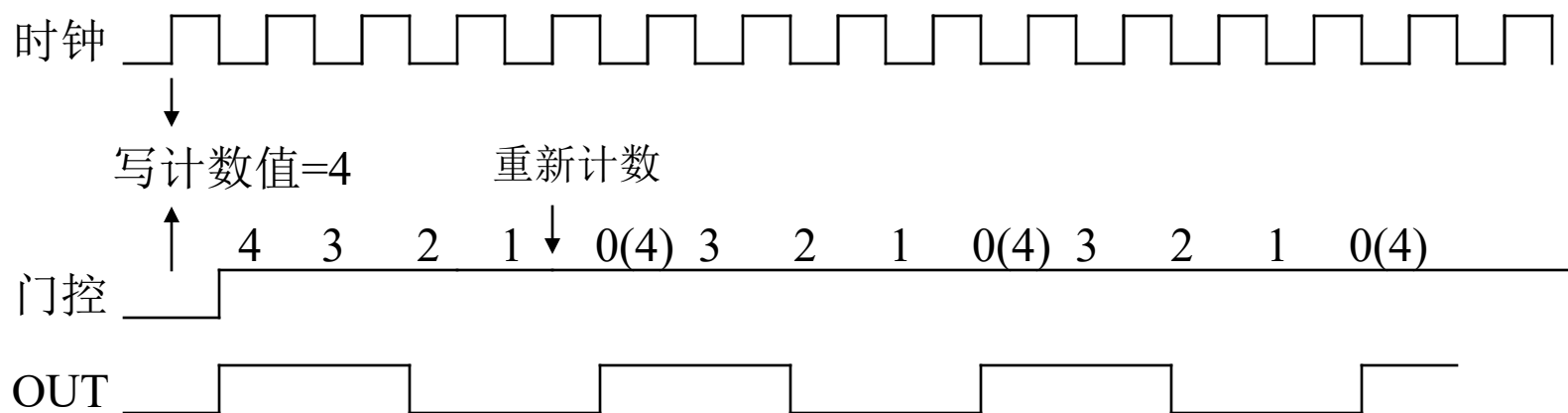
(3) 计数过程可以由门控脉冲GATE控制。当GATE变低时就暂停计数，OUT仍保持高电平，在GATE变高后的下一个时钟脉冲，使计数器恢复初值，重新计数。这种利用门控信号使计数器开始工作（即同步）的方法成为硬件同步。

(4) 在计数过程中可以改变计数值，这对计数过程没有影响，但在计数变到1时，输出变低，过一个CLK周期输出又变高，计数器将按新的计数值开始计数，可见改变计数值是在下一次有效。

17.2.2 8253的工作方式

(4) 方式3—方波速率发生器

方式3和方式2都是周期性的，但方式3输出的是对称的方波。输入时钟脉冲与输出脉冲频率之比等于加载的计数值。



17.2.2 8253的工作方式

方式3的工作特点是：

(1) 设置控制字后，OUT将为高。在写入计数值后的下一个CLK开始减1计数。当计数到一半计数值时，OUT变为低，直到计数值变为0，输出又为高，再重新开始计数。

(2) 当计数值N为偶数时，输出端的高低电平持续时间相等，为完全对称的方波。当计数值N为奇数时，则输出端的高电平持续时间比低电平持续时间多一个时钟周期，即高电平持续 $(N+1)/2$ ，而低电平持续 $(N-1)/2$ ，输出为矩形波，整个输出周期仍为N个时钟脉冲周期。

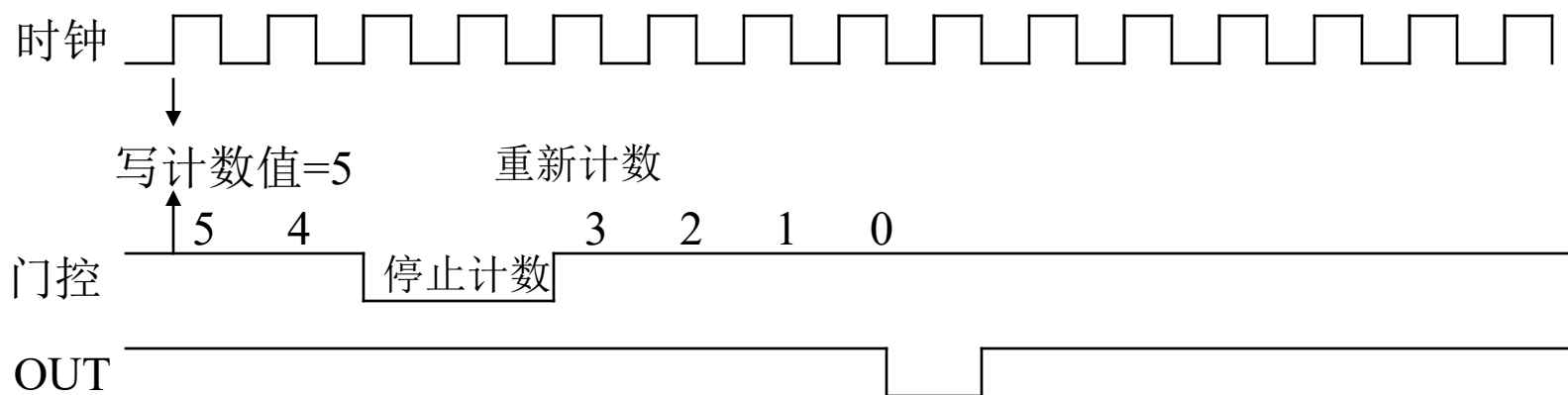
(3) GATE信号能使计数过程重新开始，GATE=1时，计数进行；GATE=0时，计数停止。如果在OUT为低期间，GATE变低，则OUT会立即变为高电平，此时计数停止。在GATE又变为高电平后，下一个时钟脉冲来到时，重新开始计数。

(4) 若在计数其间写入一个新的计数值，并不影响现行的计数过程。但是如果在输入新的计数值后，又受到门控信号上升沿的触发，就会结束当前输出的周期，而在下一个时钟周期时，计数的执行部件按新值开始计数。

17.2.2 8253的工作方式

(5) 方式4—软件触发选通

当控制字写入控制寄存器后输出开始为高，当设置完计数值后,计数器立即开始计数。当计数到0后，输出变低，持续一个输入时钟周期，然后又恢复为高，计数器停止计数。这种计数也是“一次性有效”，每次计数都要靠这种重新设置计数值进行“软件触发”。



17.2.2 8253的工作方式

方式4的工作特点是：

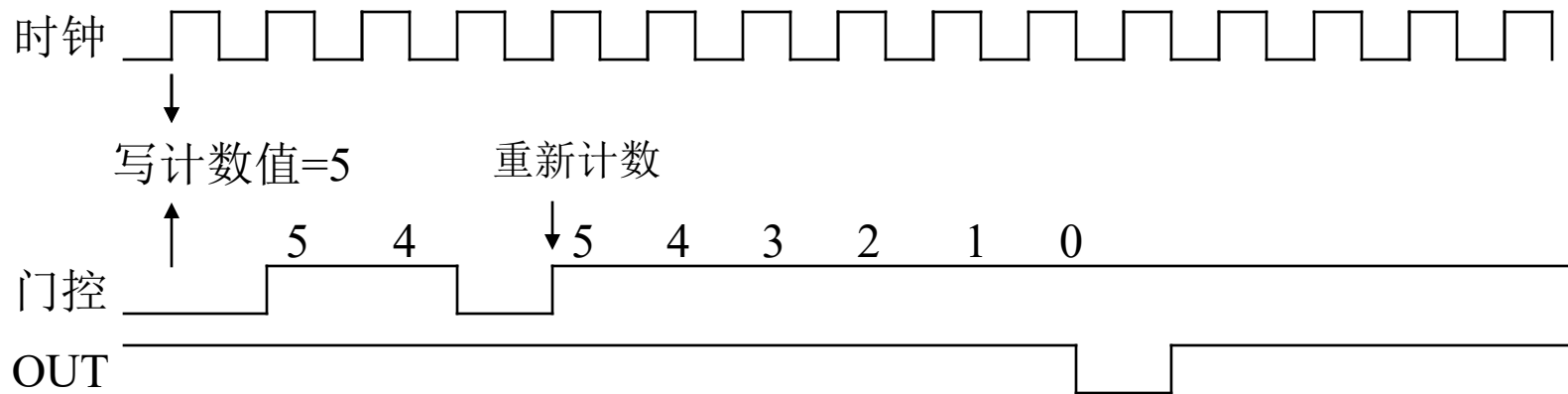
(1) 门控信号变低，可暂停计数，门控信号变高后，继续计数。

(2) 若在计数过程中改变计数值，新值写入后的下一个CLK周期开始从新的计数值开始重新计数。通过写入新的计数值使计数器从头工作，叫软件再触发。

17.2.2 8253的工作方式

(6) 方式5—硬件触发选通

方式5输出初始状态为高。在设置计数值之后，计数器并不立即开始计数，而是要等到门控脉冲的上升沿出现才开始计数，即靠硬件触发选通计数。。当计数到0时，输出变低，持续一个输入时钟周期，然后又恢复为高，计数器停止计数。下次门控脉冲触发后才能再重新开始计数。



17.2.2 8253的工作方式

方式5的工作特点是：

（1）若在计数过程中使用门控脉冲，则使计数器重新开始计数，但对输出状态没有影响。

（2）若在计数过程中改变计数值，只要没有门控信号的触发，则当前输出周期不受影响。当计数到0后，若有新的门控信号出发，则按新的计数值开始计数。如果在写入了新的计数值后，在未计到0之前，又有新的门控信号触发，则立即按新的计数值开始计数。

17.2.2 8253的工作方式

(7) 8253各种工作方式小结

① 几种方式比较

方式2, 4, 5输出的波形是一样的, 都是宽度为一个CLK时钟周期的低有效脉冲。主要区别是: 方式2连续不停地计数和输出; 方式4每次都要由程序设置计数值才能计数; 方式5不用每次设置, 但每次都要由门控脉冲触发启动才能计数。即: 计到0后, 方式2不用触发, 方式4要软件触发, 方式5要硬件触发, 才能继续计数。

方式1与方式5中的门控脉冲的作用和计数中改变计数值的结果都是相同的, 只是输出波形不同。方式1的输出信号从开始计数的整个过程都为低, 而方式5的输出从开始计数的整个过程都为高; 方式1输出的低有效脉冲宽度为N个输入时钟脉冲的宽度, 而方式5输出的低有效脉冲宽度为一个输入时钟脉冲的宽度。

② 输出端OUT的初始状态

只有方式0在置入控制字后, OUT立即变低, 其它5种方式OUT初始态均为高电平。

17.2.2 8253的工作方式

③ 启动计数与重复计数的条件

计数值写入后，并不立即开始工作，需要一定的条件才能开始工作。有些方式计数到0后，则停止工作，要重复计数同样也需要某些条件。

8253启动计数和重复计数的条件

方式	功能	启动条件	重复条件
0	计完最后一个数中断	写计数值	
1	硬件再触发大案拍脉冲	外部触发	外部触发
2	速率发生器		
3	方波速率发生器		
4	软件触发选通		写计数值
5	硬件触发选通	外部触发	外部触发

方式2和方式3是连续工作，输出是周期性的，不用重复启动再次触发，这两种方式一经写入计数值，计数器就开始不停顿地工作，直到CPU重新写入控制字为止。

17.2.2 8253的工作方式

④ 门控信号的作用.

8253门控信号GATE的作用

方式	GATE			改变计数值
	低或变为低	上升沿	高	
0(电平触发)	禁止计数		进行计数	立即有效
1(脉冲)		启动计数 下一个CLK脉冲 使输出变低		外部触发后有效
2 (电平、脉冲)	禁止计数 立即使OUT为高	重新装入计数值 启动计数	进行计数	计数到1后有效
3 (电平、脉冲)	计数停止 立即使OUT为高	启动计数	进行计数	a.外部触发后有效 b.计数到0后有效
4 (电平)	禁止计数			立即有效
5 (脉冲)		启动计数	进行计数	外部触发后有效

17.2.2 8253的工作方式

⑤ 在计数过程中改变计数值

新的计数值何时起作用因方式不同而有差别。

在8253的某一计数通道的计数过程中，要读取它的瞬间计数值，可以利用门控信号GATE使其计数过程暂停，然后分两次将16的当前计数值读入CPU；也可以给该通道送去一个锁存命令，将当前计数值在输出锁存器中锁存，而此时计数照常进行，这时分两次读取计数值送往CPU，输入指令过后，锁存器解除锁存。

17.2.3 8253/8254在IBM PC系列机中的应用

在IBM PC和PC/XT中，使用的是8253-5。在IBM PC/AT中使用的是8254。8253-5，8254在内部结构及功能上与8253完全一样，其编程方法也完全一样，其差别只是最高计数速率不同。8253的最高计数速率是2MHz，8253-5是5MHz，8254为10MHz。

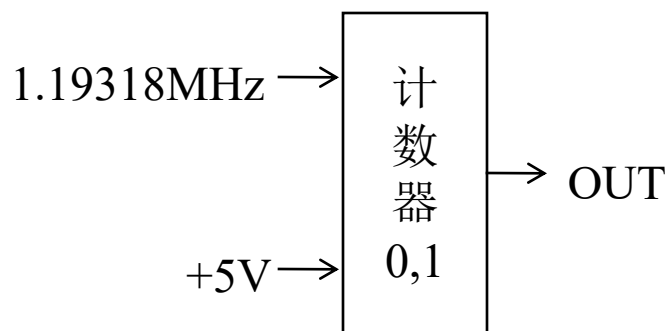
在PC系列机上，系统分配给8253/8254的地址是40H~43H，其中43H是控制寄存器的地址，40H、41H、42H分别为计数器0~2的锁存器地址。8253/8254的三个计数器各有其责，计数器0用于维持系统时钟；计数器1用于周期性地想DMA发送数据请求信号，供存储器刷新用；计数器2接到扬声器用来产生声音。

各计数器的输入时钟脉冲频率均为1.19318MHz。

计数器0和计数器1的门控信号输入端接至高电平，因此永远开启。计数器2的门控信号输入端则由8255A的PB口的第0位控制。

系统将计数器1设置为方式2，计数值为18，因此每15 μ s向DMA发一次请求信号，使DMA进行存储器刷新。此计数器绝对不可修改，否则将危及存储器的安全。

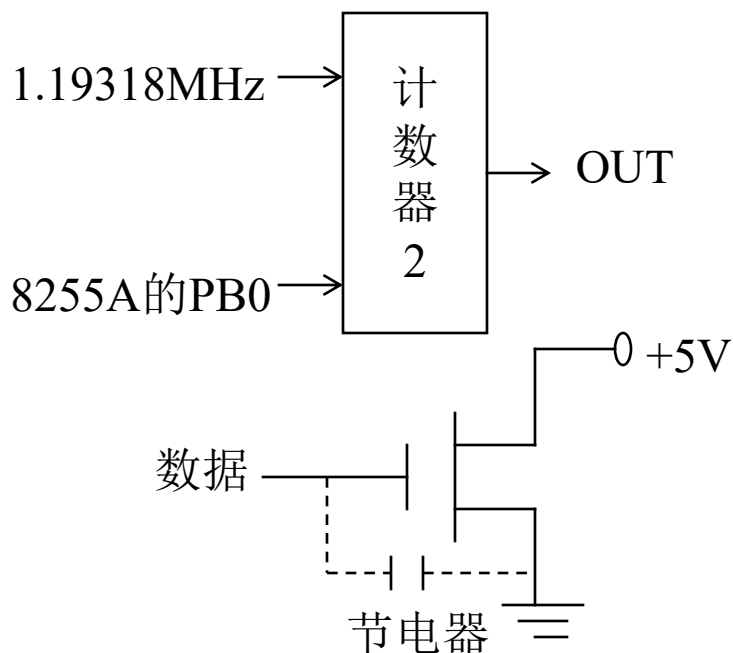
17.2.3 8253/8254在IBM PC系列机中的应用



计数器0接至8259A的IR0，工作于方式3，每隔固定时间产生一次中断，用来维持系统日历时钟，初始计数值为0（65536），因此输出脉冲频率为

$$1.19318\text{MHz}/65536=18.2$$

每秒产生18.2次中断，即每55ms产生一次中断。



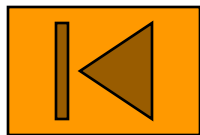
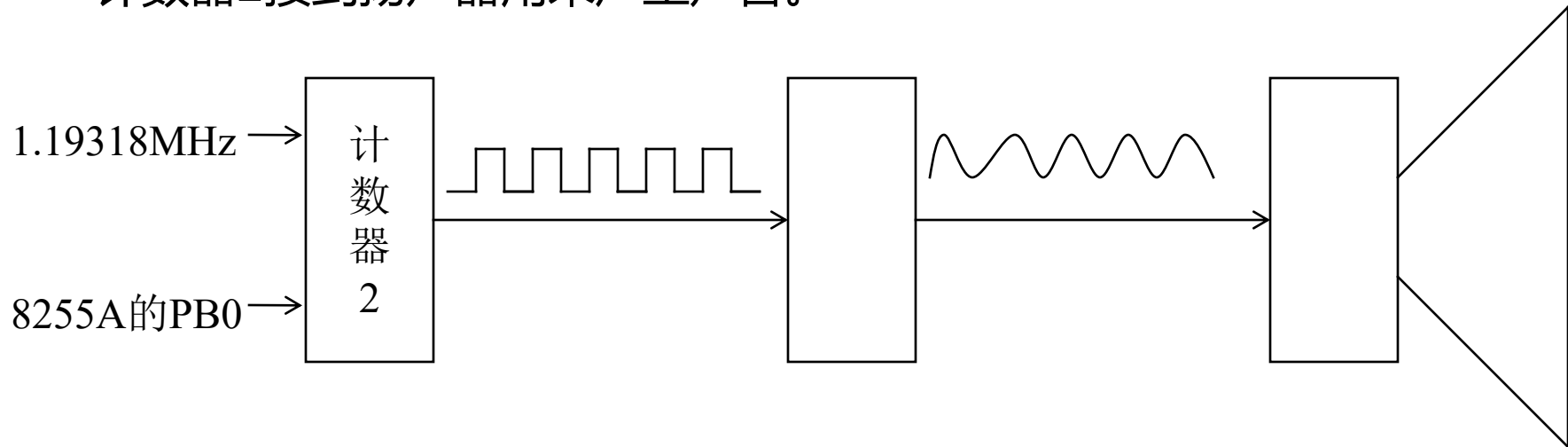
计数器1接到DMA上，工作于方式2，计数值为18，因此每15 μ s向DMA发一次请求信号，使DMA进行存储器刷新。

场效应管，记录1位数据

来存储数据，过一段时间要漏电，每隔一段时间要恢复其状态。

17.2.3 8253/8254在IBM PC系列机中的应用

计数器2接到扬声器用来产生声音。



17.3 时钟程序设计

17.3.1 使用8253/8254时钟

1. 8253/8254时钟中断的使用

在系统初始化时，已将8253/8254计数器0设置为方式3，计数值为0000H（相当于65536），所以计数器0的输出频率为 $1.19318\text{MHz}/65536=18.2\text{Hz}$ 。计数器0的输出又接至8259A的IR0上，因此系统每秒钟产生18.2次类型为8的中断（成为时钟中断）。系统就是通过该中断修改其时间计数值，以维持其日历时钟的。另外，该始终中断还负责在软盘操作完成之后关掉软盘驱动器马达的电源。因此，该时钟中断通常情况下是不能修改的，否则将影响日历时钟的运行和使系统在完成软盘操作后不能关掉磁盘驱动器的马达。

为了使拥护能够使用时钟中断，系统中还设置了另外一个中断INT 1CH，该中断被INT 8中断服务调用，因此它与INT 8保持同步，也是每秒执行18.2次，并且在使用INT 1CH时，在中断服务程序结束时也不需要向8259A输出EOI命令，因为这项工作INT 8中完成了。通常情况下，系统中INT 1CH中断服务程序只有一条指令IRET，即什么也不做。因此，可以修改INT 1CH的中断服务程序的入口地址指向我们自编的程序来完成所需要的工作。

系统中每1/18.2s的计数值被INT 8的中断服务程序保存在内存的4个字节中，需要时可通过BIOS中断INT 1AH的0号功能读出，读出的计数值是一个32位的数值，表示的是自从0时0分0秒以来的计数值，使用时可通过计算转换成时分秒的形式。

17.3 时钟程序设计

17.3.1 使用8253/8254时钟

例17.1 利用INT 1CH中断编写一个驻留程序，完成每隔大约一秒钟在屏幕的右下角显示一个数字的功能。要求数字在0~9之间变化，每隔大约一秒钟变化一个。

编写一个中断服务程序取代原INT 1CH的服务程序，这样该程序可被每隔大约1/18.2s启动执行一次，在该服务程序中需要对中断次数计数，计数达到18次，即时间过去大约1s时显示一个数字，并将当前数字加1，以备下次显示用。在显示数字之前，要先将当前的光标位置取出保存起来，以备在右上角显示完数字之后，恢复原光标的位置。

另外要解决的一个问题是要将该程序驻留内存，且不允许被其它程序执行时覆盖，这可采用DOS中断INT 27H完成。执行INT 27H之前要求将驻留程序的长度放入DX，程序段前缀的段基址放在CS中。由于编写.EXE型的驻留程序较麻烦，故通常都将驻留程序编写成.COM型的。编写.COM型程序通常需要注意如下几个问题：

- (1) 程序从100H单元开始，并且其开始必须为程序的入口；
- (2) 程序中只允许有一个段；
- (3) 不允许设置堆栈；
- (4) 不允许有存取段基址的指令存在；
- (5) 程序的长度（包括数据）不能超过64K字节。

```

CSEG      SEGMENT
          ORG      100H
          ASSUME   CS:CSEG,DS:NOTHING
BEGIN:    JMP      MAIN
COUNT    DB       18
CHAR      DB       '0'
INT1C     PROC
          DEC      COUNT
          JNZ      EXIT
          MOV      COUNT,18
          PUSH     AX
          PUSH     BX
          PUSH     CX
          PUSH     DX
          MOV      AH,03H
          MOV      BH,0
          INT      10H
          PUSH     DX
          MOV      AH,02H
          MOV      DX,004FH
          INT      10H

```

```

          MOV      AL,CHAR
          MOV      AH,0AH
          MOV      CX,1
          INT      10H
          POP      DX
          MOV      AH,02H
          INT      10H
          INC      CHAR
          CMP      CHAR,'9'
          JBE      OK
          MOV      CHAR,'0'
OK:        POP      DX
          POP      CX
          POP      BX
          POP      AX
EXIT:      IRET
INT1C     ENDP
MAIN:     MOV      DX,OFFSET INT1C
          MOV      AX,251CH
          INT      21H
CSEG      ENDS
          END      BEGIN

```


17.3 时钟程序设计

17.3.1 使用8253/8254时钟

2. 随机数的产生

生成随机数的一个最方便的办法就是利用时钟。前面已经提过每秒钟产生18.2次的时钟中断，对中断次数计数（时间计数），该计数值是一个32位的数据，保存在内存的4个字节中，可以通过INT 1AH的0号功能读出。读出之后，CX存放的是其高16位，DX是其低16位。这个计数值是持续变化的，特别是其低16位可以被认为是伪随机数。由于计数值总是增加而不是随机改变，所以不能提供一个真正的随机数。但是，因为计数值以每秒18.2次的速率变化，它的随机程度足够满足大部分应用了。

17.3 时钟程序设计

17.3.1 使用8253/8254时钟

例17.2 编写一个产生0~99范围内随机数的子程序。

时间计数值的低16位基本上可以作为一个16位的随机数使用。要产生0~99范围内的随机数，可将该计数值除以100，取其余数即可。做除法时还应考虑产生溢出的问题。为了保证不产生溢出，可将16位数据的高2位清0，然后再做除法就安全了。

子程序说明文件如下：

- (1) 子程序名：RAND99；
- (2) 子程序功能：产生0~99范围内的随机数；
- (3) 入口条件：无；
- (4) 出口条件：AL为产生的随机数；
- (5) 影响的寄存器：AH，AL，F。

```
RAND99  PROC
        PUSH    CX
        PUSH    DX
        XOR     AH,AH
        INT     1AH
        MOV     AX,DX
        AND     AX,3FFFH
        MOV     DL,100
        DIV     DL
        MOV     AL,AH
        POP     DX
        POP     CX
        RET
RAND99  ENDP
```

17.3 时钟程序设计

17.3.2 实时时钟的使用

在IBM PC/AT计算机上扩充了一个实时时钟，该时钟在系统断电后仍能保持运行。BIOS的INT 15H和INT 1AH中都扩充了对实时时钟的管理。

1. 实现延时

实现延时，我们可以编一段什么也不做的小程序，让它反复执行，反复执行的次数就决定了延时的长短。但是这样编写出来的程序依赖计算机的运行速度（主频、CPU类型），其通用性差。为了克服这个缺点，可利用BIOS提供的INT 15H的86H号功能来实现。使用该功能时，将延时的时间（微秒数）放入CX：DX联合组成的32位寄存器中，其中CX中为高16位。虽然该功能要求放入的时间单位是微秒，但替延时精度只能达到毫秒级。

例17.3 编写一个程序，在屏幕上每隔一秒显示一个数字，从0显示到9之后结束。

CSEG	SEGMENT	
	ORG	100H
	ASSUME	CS:CSEG
DELAY:	MOV	BL,'0'
NEXT:	MOV	DX,4240H
	MOV	CX,000FH
	MOV	AH,86H
	INT	15H
	MOV	DL,BL
	MOV	AH,2
	INT	21H
	INC	BL
	CMP	BL,'9'
	JBE	NEXT
	INT	20H
CSEG	ENDS	
	END	DELAY

17.3 时钟程序设计

17.3.2 实时时钟的使用

在IBM PC/AT计算机上扩充了一个实时时钟，该时钟在系统断电后仍能保持运行。BIOS的INT 15H和INT 1AH中都扩充了对实时时钟的管理。

2. 实现闹钟

INT 1AH的6号和7号功能允许我们将AT计算机变为一台多用途的闹钟，此时需做如下工作：

- (1) 准备一个中断服务程序，它完成在达到响铃时间之后所做的事情。
- (2) 设置中断服务程序的入口地址。中断类型为4AH。
- (3) 把响铃的时间放入CH(时)、CL(分)及DL(秒)中。时间应为BCD码的形式。
- (4) 使用INT 1AH的6号功能来设置响铃。

例17.4 编写一个闹钟程序，在下午3点03分30秒响铃。响铃后，如果用户按下[Esc]键则停止响铃。

```

CSEG      SEGMENT

            ORG      100H

            ASSUME    CS:CSEG

BEGIN:     JMP      MAIN

ALARM:     PUSH     AX

            PUSH     BX

AGAIN:     MOV      AX,0E07H

            INT      10H

            MOV      AH,1

            INT      16H

            JZ       AGAIN

            MOV      AH,0

            INT      16H

            CMP      AL,16H

            JNE      AGAIN

            MOV      AH,7

            INT      1AH

```

```

            POP      BX

            POP      AX

            IRET

MAIN:      MOV      DX,OFFSET ALARM

            MOV      AX,254AH

            INT      21H

            MOV      CX,1503H

            MOV      DH,30H

            MOV      AH,6

            INT      1AH

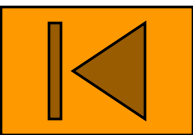
            MOV      DX,OFFSET MAIN

            INT      27H

CSEG      ENDS

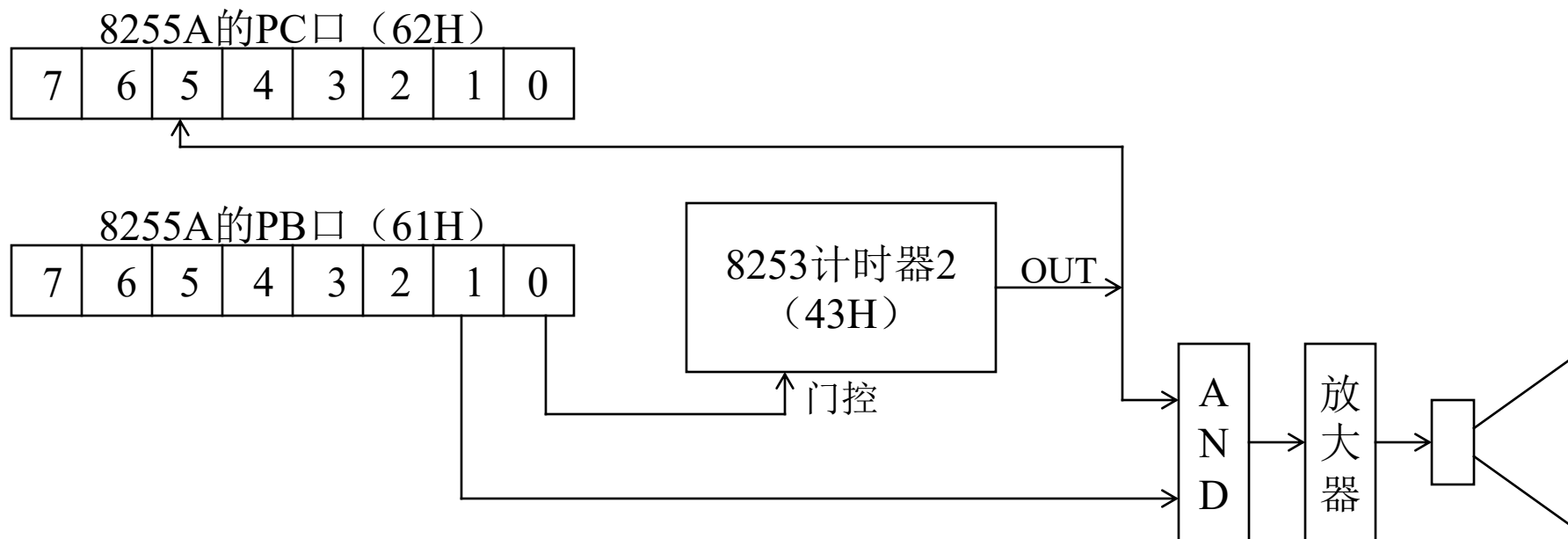
            END      BEGIN

```



17.4 音响程序设计

17.4.1 扬声器与系统的连接



由图可见，在IBM PC系列机上有两个可以驱动扬声器的信号。一路是8255A的PB口的第1位的输出；另一路是8253/8254计数器2的输出，这可以通过对8253/8254编程使其产生某一频率的方波加以控制。两路输出经过一个与门之后，经驱动器驱动，推动扬声器发出声音。由于两个信号是与的关系，因此，要想使其中一个正常输出，另一个的输出必须保持为1状态。

17.4 音响程序设计

17.4.1 扬声器与系统的连接

8253/8254计数器2的输出在IBM PC，PC/XT和PC/AT机上有不同。在PC和PC/XT机上，计数器2的输出分为两路，一路接至扬声器产生声音，另一路接至8255A的PC口第5位，可通过软件检测；在PC/AT机上，计数器2的输出仅接到扬声器。

8255A是一个可编程的通用外设接口，由一个控制寄存器（端口地址为63H）和三个输入输出数据寄存器组成。输入输出寄存器又成为输入/输出端口，名字分别为A口，B口和C口。在IBM PC系列机上，这三个端口的地址分别为60H，61H和62H。在系统初始化时，已将A口和C口设为输入方式，B口设为输出方式。

17.4 音响程序设计

17.4.2 音响程序的设计

1. 软件控制产生声音

可以通过将8255A的PB口的第1位交替置位和复位产生一个方波信号，推动扬声器发出声音。PB口交替置位和复位的频率就是方波的频率，也就是所产生的声音的频率。

例17.5 编写一个使扬声器发出600Hz声音，并持续2s的子程序。

延迟的时间可用重复执行LOOP指令实现。假设计算机的主频为4.77MHz，那么其每个时钟周期为 $1/4.77\mu\text{s}$ 。600Hz的声音其半周期为 $1/1200\text{s}$ ，8086/8088CPU执行一条LOOP指令需要17个时钟周期（即 $17/4.77\mu\text{s}$ ），所需要的次数为：

$$\begin{aligned}\text{循环次数} &= (1/1200) / (17/4.77 \times 10^{-6}) \\ &= (4.77 \times 10^{-6}) / (17 \times 1200) \\ &\approx 233.8\end{aligned}$$

由于每个方波的周期为 $1/600\text{s}$ ，要持续2s，只需要控制产生1200个方波即可。

```

BEEP    PROC

        MOV     DX, 1200

AGAIN:  IN      AL, 61H

        AND     AL, 0FEH

        OR      AL, 02H

        OUT     61H, AL

        MOV     CX, 233

        LOOP    $

        AND     AL,0FDH

        OUT     61H,AL

        MOV     CX, 233

        LOOP    $

        DEC     DX

        JNZ     AGAIN

        RET

BEEP    ENDP

```

17.4 音响程序设计

17.4.2 音响程序的设计

2. 对8253/8254编程产生声音

例17.6 编写一个子程序，用8253产生指定频率的声音并持续指定的时间。

要产生指定频率的声音，可将8253的计数器2设置为方式3，根据声音的频率计算出8253所需的计数值并送入8253计数器2即可。使声音持续一定时间可采用通过8255A的PC5检测8253计数器2的输出，并对其方波的周期进行计数来实现，延时的时间与输出方波周期（频率的倒数）和周期个数之间的关系如下：

持续时间=方波的周期× 输出方波的个数

通过8255A的PB1，我们还可以控制8253的计数器2，使其只输出方波，但不使扬声器发声，以达到仅延时的目的。

17.4 音响程序设计

17.4.2 音响程序的设计

2. 对8253/8254编程产生声音

例17.6 编写一个子程序，用8253产生指定频率的声音并持续指定的时间。

子程序说明文件：

- (1) 子程序名：SOUND；
- (2) 子程序功能：启动8253计数器2产生指定频率的声音
并持续指定的时间；
- (3) 入口条件：DX=产生方波的频率；
AX=持续（延时）的时间（ms）；
- (4) 出口条件：无；
- (5) 受影响的寄存器：AX，DX，F。

SOUND PROC		IN	AL,61H
	PUSH CX	AND	AL,11111100B
	PUSH SI	OR	AL,BL
	MOV SI,DX	OUT	61H,AL
	MUL DX	ZERO: IN	AL,62H
	MOV CX,1000	TEST	AL,00100000B
	DIV CX	JZ	ZERO
	MOV AX,CX	ONE: IN	AL,62H
	MOV AL,10110110B	TEST	AL,00100000B
	OUT 43H,AL	JNZ	ONE
	MOV DX,0012H	LOOP	ZERO
	MOV AX,34DCH	POP	SI
	DIV SI	POP	CX
	OUT 42H,AL	RET	
	MOV AL,AH	SOUND ENDP	
	OUT 42H,AL		

17.4 音响程序设计

17.4.2 音响程序的设计

2. 对8253/8254编程产生声音

例17.7 利用例17.6的子程序编制模拟电话振铃声音的程序。

电话铃声一般由两个频率的声音组成，一个为600Hz，另一个为1500Hz，两个声音交替产生，各持续0.03s，交替30次之后，停顿2s，然后再重复。重复数次之后停止。

CSEG	SEGMENT		MOV	DX,100
	ASSUME CS:CSEG		MOV	BL,1
RING	PROC FAR		MOV	AX,2000
	PUSH DS		CALL	SOUND
	XOR AX,AX		DEC	BP
	PUSH AX		JNZ	AGAIN
	MOV BP,8		RET	
AGAIN:	MOV CX,30	RING	ENDP	
NEXT:	MOV DX,600	CSEG	ENDS	
	MOV AX,30		END	RING
	MOV BL,3			
	CALL SOUND			
	MOV DX,1500			
	MOV AX,30			
	CALL SOUND			
	LOOP NEXT			

17.4 音响程序设计

17.4.2 音响程序的设计

3. 音乐的产生

例17.7 利用例17.6的子程序编制模拟电话振铃声音的程序。

电话铃声一般由两个频率的声音组成，一个为600Hz，另一个为1500Hz，两个声音交替产生，各持续0.03s，交替30次之后，停顿2s，然后再重复。重复数次之后停止。