

# IBM PC微机的实时时钟及应用

白 雨 虹

**摘要：**本文论述了IBM PC微机的实时时钟的原理，以及它在高分辨率图像滚动显示和上机机时管理中的应用。

## 一、概 述

实时时钟是计算机系统中一个必不可少的部分，它不受CPU控制，独立运行。

实时时钟由时钟电路和定时计数电路两部分组成。时钟电路产生周期脉冲，定时计数电路将这些脉冲数进行累计。当累计到程序设定的值时，向中断控制器（8259）发一个脉冲，申请中断，执行操作系统的时钟中断处理程序。由时钟中断处理程序将累计结果存贮在内存的固定单元，并转换为与通用时间相一致的操作系统时间。实时时钟使在操作系统下运行的软件具有定时和计时的工作能力。

本文将论述IBM PC微机的实时时钟原理及其应用。

## 二、工 作 原 理

### 1. 时钟电路的工作原理

时钟电路原理框图如图1所示。

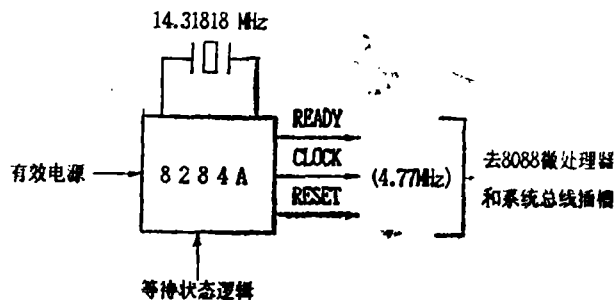


图1 时钟电路原理框图

晶体振荡器产生14.31818MHz的振荡频率，该频率经8284A时钟芯片三次分频得到供8088微处理器用的4.77MHz的信号，此信号还连到系统总线插槽。

### 2. 定时计数器的工作原理

定时计数器原理框图如图2所示：

具有三个16位定时计数器的8253—5定时计数芯片被连接到系统的总线上。来自8284A的4.77MHz信号四分频后变成1.19MHz，作为定时计数器三个通道的时钟输入驱动信号。8253定时器计算系统时钟数，将若干个时钟周期转换成一个脉冲，这些脉冲用来计时。

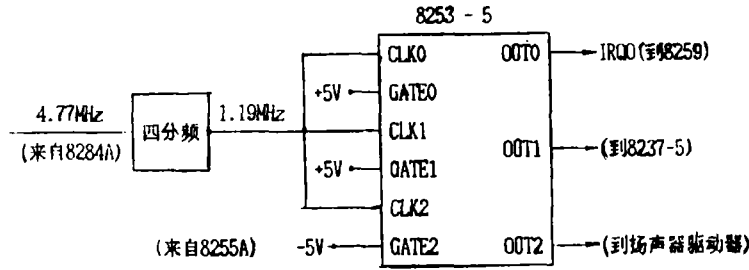


图2 定时计数器原理框图

8253芯片独立于CPU运用，CPU对8253芯片编程后，就可去做其它工作。

通道0用作通用的系统定时器。它的选通信号任何时候都接高电平。由BIOS设置这个通道，使得每54.936ms (18.2次/秒)产生一个0级 (IRQ0) 的中断请求到8259A。BIOS例行程序对这些中断请求计数，以产生一个能读或能写的日历计数。每次中断请求例行程序更新日历计数，修正时间-日期时钟，并在退出之前执行INT1CH中断指令。这是该例行程序的一个重要功能，利用这个功能可以进行定时操作，诸如外设的定时驱动或采样等。

通道1用来刷新系统动态存贮器。通道2用于驱动系统扬声器。

### 3. 日历计数的更新及设置

日历计数更新的工作过程如图3所示：

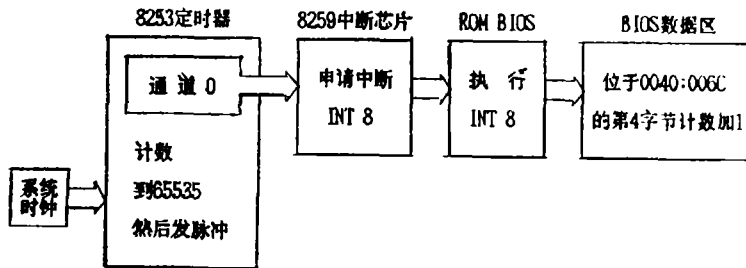


图3 更新BIOS日历计数框图

计算机采用BIOS数据区的一变量，记录定时器通道0发出的脉冲数，这一脉冲计数通常称为“日历计数”。通道0每秒输出18.2个脉冲，申请18.2次硬件中断（称“定时器中断”）。硬件中断只简单中断CPU的工作，增加日历计数，修正时间-日期时钟。如果日历计数0代表午夜12.00，那么，当计数值达到与24小时对应的值时应该回零。这样，一天内的任何时刻就可简单地用计数值除以18.2得出。目前，日历计数已广泛应用于定时操作。

IBM PC机启动时，DOS提请用户输入当前日期和当前时间。当前时间输入值放入存贮日历计数的四字节内存中。放入内存前，当前时间值必须首先转换成日历计数，即必须将时间折合成午夜以来所经过的十八分之一秒的个数。日历计数由定时器中断每秒修正18.2次，以后任意时间都可由那一时刻的日历计数算出，并转换成时-分-秒的格式。如果启动时没有输入当前时间值，则日历计数置0。

IBM PC机时间记录在BIOS数据区，日期放在command.com中的一个变量里，其格式为日、月、年，存放在三个顺序的字节中。年份从0开始算起，0代表1980。与日历计

数不同，日期的存贮位置随DOS版本和command.com在存贮空间的位置而变化。

当BIOS日历计数在24小时后翻转时，DOS会相应地更正日期。

### 三、延时、定时与实时操作

延时操作是指利用计算机的有关指令，如循环、转移、增加、减少等指令，使计算机“原地踏步”，利用计算机执行这些指令所消耗的时间，达到延时的目的。延时操作根据延时时间以及每条指令的时钟周期数决定采用单重循环或多重循环。为了得到精确的延时，必须准确地掌握所用指令的执行周期和准确计算循环次数。

这种延时操作执行时，CPU工作在延时程序，占用了大量CPU时间。为了把CPU解脱出来，提高系统效率，除了必须，一般采用定时操作。

定时操作使得机器和人一样可以准确地确定操作时间。程序可以读取日历计数的初始值并与随后的读数相比较。在每次读取日历计数判断时间以前，可以加入其它指令，这样就可以在定时时间到达以后转去执行其它操作；否则，继续现行操作。

有一点需特别注意，当执行延时或定时程序时，不允许在执行中间产生中断，因为这样会延长定时时间。为此，可在定时程序前加上关中断指令，在执行完定时程序后，执行一条开中断指令即可。

实时操作具有实时性以及分时性的特点。实时性是要求程序在特定时间发出指令，而分时性是指定时器中断（每秒18.2次）修正日历计数后，调用时间判断例程。在调用返回之后到再次调用之前这段时间内，或者说调用时间判断例程以外的时间，用户可以执行任何操作，就如同用户程序分一点时间给时间判断例程。由于这个时间很短，所以并不能感觉到很明显的执行速度降低。

在IBM PC中，时间判断例程由BIOS所包含的特殊伪中断（INT1CH）调用，即可执行实时操作。

### 四、应用举例

#### 1. 在高分辨率CRT显示控制上的应用

通过读取“日历计数”，计算两次读取“日历计数”的差值，除以18.2，可以获得两次读取“日历计数”经过了多少时间。在实际应用中，可以用来测定延时操作的准确延时时间。即在延时操作的开始前，结束后各取一次“日历计数”，其差值除以18.2就是延时操作的精确延时时间。在“SAR光学处理器CCD图像显示系统”<sup>[2]</sup>中，为了使CRT屏幕上的图像稳定地滚动显示，需精确测定在高分辨率监视器上显示 $1024 \times 1$ 行或 $1024 \times n$ 行所需的时间，以便插入对应的新行数据，推出老行数据。为此，就可以在第一个象素显示之前和最后一个象素显示之后分别读“日历计数”，将两次日历计数之差除以18.2就获得了显示 $1024 \times 1$ 行或 $1024 \times n$ 行所需的时间（秒值）。这就为线阵CCD与微机以及微机与高分辨率图形板之间的有效连接，提供了准确可靠的实际时间数据，使图像可以在程序控制下，按快慢不同的几个档次，稳定可靠地进行由下至上的滚动显示。

#### 2. 在微机工作机时管理上的应用

在一定的场合下，微机的工作机时管理，是个必须解决的问题。什么人上机，上机时间

多长，规定上机时间用完后如何通知用户等，如均由人工管理，将十分繁锁，有时是不可能的。本实例原则上可以实现上机人员登记，控制上机时间。这样，什么时间，什么人在上机均可记录在案，可以区分有权用户与无权用户，无权用户（上机人员名录中无名的用户）不准许上机等。本文介绍的例子只是实现定时时间控制：由管理员输入上机时间，也可以程序设定，到达定时时间后，演奏一段音乐，通知用户，上机时间到，提示用户下机，机器死锁，只有在管理人员控制下，才可继续上机。

该例的原理可以应用到工业自动控制，如数控机床、精炼炉控制、温度采集等，尤其适用于对时间要求苛刻的场所。在中断例程中，完成采样、运算、向外设接口发布控制命令等。下面介绍具体实现原理：

(1) 基本原理

INT1CH伪中断所调用的子程序由于只含有一个中断返回语句 (IRET)，所以在用户给它提供子程序之前不起任何作用。它的特别之处在于，它是在BIOS 定时器中断修正日历计数之后，由该中断调用的。也就是说，这是每秒18.2次自动发生的硬件中断。我们可以改变这一中断向量，使之指向我们的中断例程。这样，中断例程每秒被调用 18.2 次。每次调用，中断例程读出刚刚刷新的日历计数，与到定时时间的日历计数比较。相等时，进行相应的实时操作，不等时，例程只简单返回，CPU 可以节省时间去做其它工作。

本例中，判断定时时间的方法是，计算在定时时间内中断例程被调用的次数，定时计数。如定时时间为 1 小时，则中断例程被调用次数为  $3600 \text{秒} \times 18.2 \text{次/秒} = 65520 \text{次}$ ，将其放在内存中一固定存贮空间。例程被调用一次，该值减 1，直到为零，则定时时间到。

(2) 程序流程图见图 4、图 5

(3) 具体实现

时间的输入及转换

FFFFH 对应的十进制数是 65535，而一小时的定时计数值为 65520。由此可见，超过 1 小时的定时要涉及到复杂的 32 位操作，可以采用定时时间的小时数与分钟数分开存贮的办法。使用 DOS 调用 INT21H 的 φAH 功能，将小时数与分钟数做为字符串输入，各自放在不同的存贮单元。由于存贮单元放的是 ASCII 码，所以要将其转换为十进制数，以进行定时计数的计算。

定时计数的计算

中断 INT1CH 每秒发生 18.2 次，将定时时间的秒值乘以 18.2 可以获得定时计数值，即 INT1CH 执行的次数，也就是中断例程被调用的次数。

中断向量的读取与设置

中断向量表共可存放 256 个中断向量。计算机通过中断号，计算出该中断的中断向量在中断向量表中的存放地址，取出中断向量赋给寄存器 CS 和 IP，使计算机转去执行中断服务例程。我们可以修改中断向量表中对应此中断的中断向量，使其指向我们的中断例程，使它每秒被调用 18.2 次，据此进行定时操作。因此，要设置中断向量值。在此以前，必须将原中断向量值读取出来保留。当中断例程结束后，必须恢复原来的中断向量，否则，每秒 18.2 次

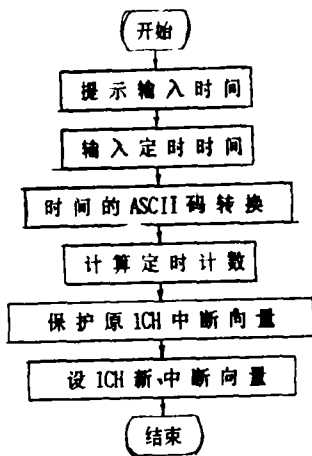


图 4 主程序流程图

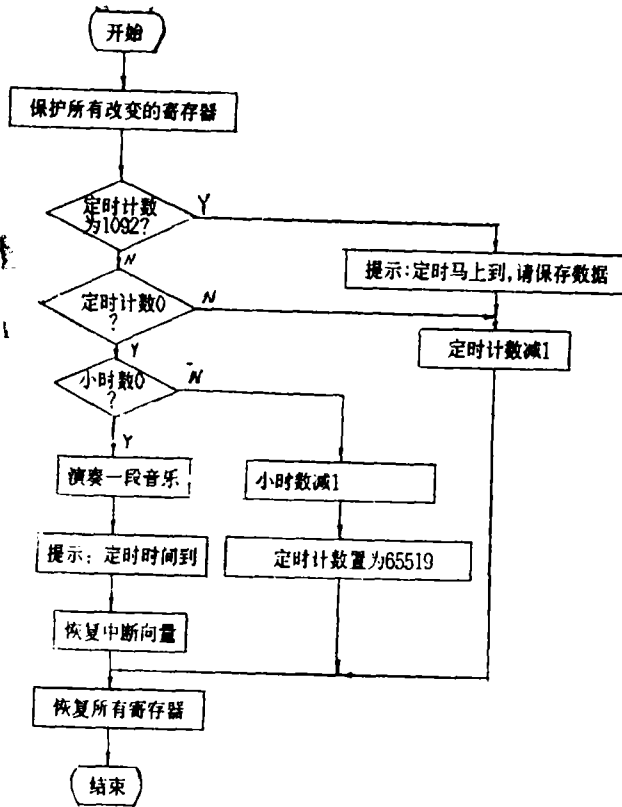


图5 中断例程流程图

的中断仍将跳转到中断例程在存储器中的原驻留处，由于定时时间已到，一些指针已不起作用，甚至原分配存储器可能已被撤消，程序已不在该驻留处。如果不恢复原中断向量，会出现错误的结果。

INT 21H的功能调用35H能读出当前中断向量值，将中断向量段值存入寄存器 ES，偏置值存入DX 中。在进入定时中断例程前，由主程序调用此功能，得到当前中断向量值，保存此值，并在定时中断的定时时间到时，奏响音乐通知用户。在中断例程退出前，使用功能 25H恢复此值。

中断向量的设置使用INT 21H的功能25H。该功能可以将中断向量设置到特定的地址，地址长为两个字，高位字为段值(CS)，低位字为偏置值(IP)。只需将中断例程段值存入DS，偏置值存入DX，接着将中断号放入AL，执行 INT 21H 调用中断，就可使中断向量指向定时中断例程。当功能25H改变中断向量时，将自动禁止硬件中断。设置完毕后，定时中断例程每秒被调用18.2次。具体程序如下：

，设置中断向量

PUSH DS，保护DS

LEA DX，ROUTINE，中断例程 ROUTINE 的偏置放入DX

MOV AX，SEG ROUTINE，将中断例程段

MOV DS，AX，放入DS中

MOV AX，251CH，功能调用号25H

```
INT 21H, 中断向量号1CH
POP DS, 恢复DS
```

, 中断例程

```
ROUTINE PROC FAR
:
IRET
FINISH EQU $
ROUTINE ENDP
```

程序驻留

为完成实时操作, 必须使中断例程驻留内存。由于 .exe 文件很难使程序驻留, 所以采用 .com 文件。使用 INT 27H 指令, 将开始偏移地址 (100H) 到终止偏移地址 (DX) 之间的程序驻留在内存, 做为 DOS 的一部分。终止偏移地址的获得可以在中断例程结束语句 IRET 之后, 使用下述语句完成: “FINISH EQU \$”。最后, 在执行 INT 27H 语句之前, 执行 “LEA DX, FINISH。” 将驻留程序终止偏移地址送入 DX, 完成驻留。

定时时间的判定

由于采用定时时间的小时数与分钟数分开存贮, 所以可以先求出与定时时间分钟数对应的定时计数, 当其随中断例程被调用递减为零后, 再判定定时时间的小时数是否为零, 如不为零, 则求出对应 1 小时的定时计数, 同时将定时时间小时数减 1, 退出中断例程, 该例程每秒被继续调用 18.2 次; 如定时时间的小时数为零, 则定时时间到, 演奏一段音乐, 并输出提示: “定时时间到” 通知用户。

对应 1 分钟的定时计数为  $1 \times 60 \times 18.2 = 1092$  次, 所以, 定时计数为 1092 时, 距定时时间到还有一分钟, 此时输出提示: “还有一分钟到定时时间, 请马上保存数据!”, 然后, 退出例程。

本例存在的不足

由于程序驻留内存, 被作为 DOS 的一部分保护起来, 这样, 当定时时间到达后, 除非释放该程序所占内存, 否则, 该内存区不会被其它程序使用。而例子并没有释放内存, 尽管占内存很少, 但仍造成资源浪费, 如多次运行这个程序, 将会在原占用内存之外继续占用内存来驻留。如果程序驻留内存部分为 1KB, 运行 10 次该程序, 将损失 10KB 内存空间, 对于 640KB 内存, 这当然不算很大, 但对于较大驻留内存的程序, 将造成更大的资源浪费。

解决办法: (a) 将主程序也驻留内存, 使用几个组合键 (如 ALT—CTRL—ESC) 随时启动主程序给定时时间及其计数赋值, 这样可以随时定时, 也可以不必释放内存, 不必多次运行该主程序, 造成资源浪费。

(b) 定时时间到达以后, 释放内存。

对比之下, a 方案比较优越。

对于没有联网的计算机, 如用户关机之后再开机, 则定时无法继续进行, 除非开机后重新装入程序, 这可以在 C:\AUTOEXEC.BAT 文件中写入该文件名, 如用 C 盘启动, 可以重新开始计时, 但对于使用用户自带软盘驱动, 则另当别论。对于联网微机, 修改该程序可以更好地完成上机管理。

## 五、结 束 语

实时时钟是计算机不可缺少的部分，任何操作系统都必须建立在特定的实时时钟之上，而软件又必须建立在特定操作系统之上。因此，研究IBM PC机实时时钟原理及应用，对于了解硬件结构，作好接口设计，更好地用微机进行实时控制，以及了解操作系统，编制实时应用软件，都具有一定的价值。本文在实际工作的基础上探讨了实时时钟的硬软件构成，并给出了两个实用例子。

感谢黄廉卿，于前洋老师在本文写作过程中给予的指导和帮助。

### 参 考 文 献

- [1] 郝忠孝,《微型机实时软件设计》机械工业出版社, 1987
- [2] 黄廉卿等,《航空遥感实用系统与应用》能源出版社, 1989

## A Real-time Clock in a IBM PC and It's Applications

Bai Yuhong

### Abstract

In this article, the principle of a real-time clock in a IBM PC microcomputer and it's applications in the rolling display of images with high resolution and the management of microcomputer working-time are described.