实验 A 基于 IA-16 的定时调度机制研究

一．实验目的

（1）回顾熟悉 TPC 微机实验装置及综合设计任务分析
（2）掌握基于时间的任务分配方法；
（3）掌握基于 8253 定时中断的软件定时器构建
（4）掌握基于 8253 定时中断的任务调度与通信
（5）提供 IA-16 多任务系统调度运行的感性认识

二．实验环境

1． 硬件环境

微型计算机（Intel x86 系列 CPU）一台，清华科教仪器厂 TPC-2003A 微机接口实验装置一台（支持一路外部中断）；

数字记忆示波器一台.

2． 软件环境

(1)Windows XP 操作系统，编辑、汇编、链接和调试程序；运行环境为 DOS-裸机模式

(2)PC2003A 集成开发环境软件一套及实验装置电子版资料

三．基本样例实验

1．基本样例实验内容和要求

（1）软件框架 由于系统只支持一路外部中断，所以 IA-16 多任务调度程序主要基于时间调度的思想。这种调度方式下各个任务的执行顺利都是固定的，即各个任务的执行周期固定。因此，编写程序时需要根据具体任务对时间的依赖性，预定为其分配执行顺序。

时间调度由中断来实现。中断来到前正常执行主程序，一旦中断来到转而执行相应的中断服务子程序。在中断服务子程序中进行任务的调度与切换。以分支结构判断执行某个任务的条件是否满足，若满足则调用相应任务的子程序；否则继续判断执行下一个任务的条件是否满足。单个任务的程序段应在 1ms 内执行完毕，然后返回到中断程序部分，通知中断控制器中断处理结束，最后返回到主程序，等待下一次中断的来临。

设计程序中共有四个相对独立的任务（可方便地扩展到 N 个任务），分别控制处理不同接口/对象：基本时间中断（T0=1ms）利用实验装置上 8253 计数器 0 产生。因此时间粒度=T0。主程序中任务不间断地执行，完成基本管理（结束条件）；在中断情况下任务 taska、taskb、taskc 作为三个独立的子程序在满足条件时被调度而执行。设计：A 任务 2ms 执行一次，B 任务 4ms 执行一次，C 任务 4ms 执行一次。

主程序中的任务：相对较为简单，读 8255 的 C 口，判断 PC0 的值，若连接到开关 K0 的 PC0 读到为 1，即 K0 拨到“1”位置，则结束程序，准备返回 DOS。

任务 A：在屏幕固定位置输出字符串，并显示其执行的次数，计数到 0FFFFH 然后计数归零。

任务 B：控制 DAC0832 实现正弦波发生器。先把组成一个完整的正弦波的 32 个数值列

成表，每次执行任务 B 时读表输出一个值；任务 B 执行 32 次后即可输出一个完整的正弦波。

任务 C：通过 8255 的 C 口读入拨动开关 K0-K7 值,A 口输出到 LED0-LED7 亮灯显示。

(2) 定时器 8253 地址 Port+280H,Gate0 接+5V; Out0 接 IRQ

基本 I/O 实验电路 1 参考图 1，8255C 口接逻辑电平开关 K0～K7，编程 A 口接 LED 指示灯显示电路 L0～L7；C 口输入数据，再从 A 口输出.；

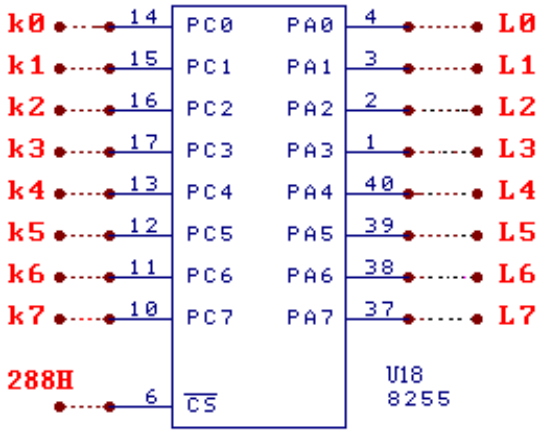

图 1 任务 0/1： 8255 简单输入输出

基本 I/O 电路 2 参考图 2：DAC0832 采用单缓冲方式，具有单双极性输入端，编程产生正弦波输出；

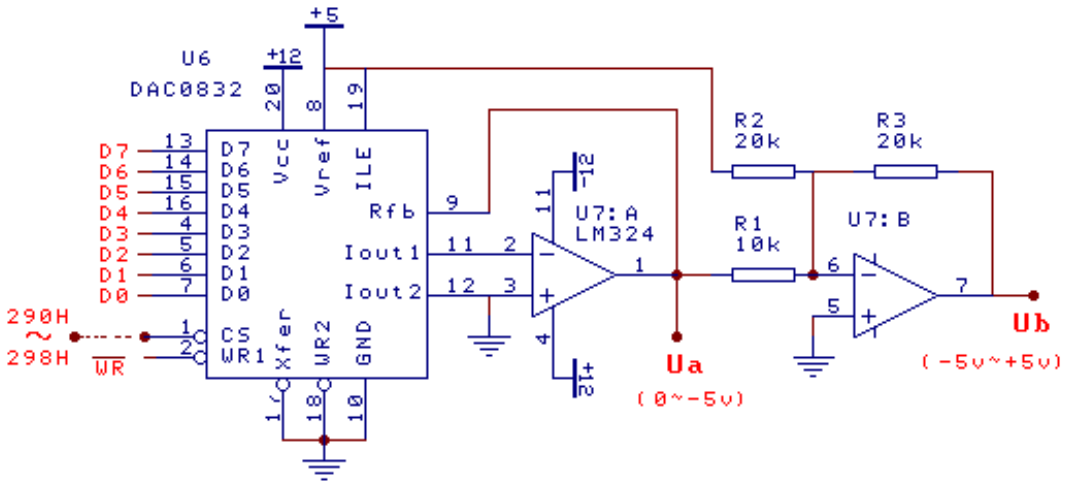

图 2 任务 2：DAC0832 正弦波产生输

（3）思考：

如果要求在主程序中显示任务 A 的执行次数，并把次数写入 PC 显示存储器，则两个任务之间应该如何通信？请修改程序并调试运行。

2．样例编程提示

(1) 该程序中任务 B 的实现需要特别注意。一般波形发生器实现时，查表依次输出表中

这个数值即可，相邻两个数据之间的时间间隔通过延时子程序来实现。而在该任务调度程序中，因为定时中断 1ms 一次，每个任务都需要在 1ms 内执行完返回到主程序。所以，延时子程序不可以采用。取而代之的是控制任务 B 中的某些变量，每次任务执行后作相应调整，将波形发生器的数据分若干次输出，其输出相邻两个数据的时间间隔由任务 B 的执行周期决定。

(2) 参考流程图(见图 3 和 4)

(附注：该程序需要在纯 DOS 环境下执行)

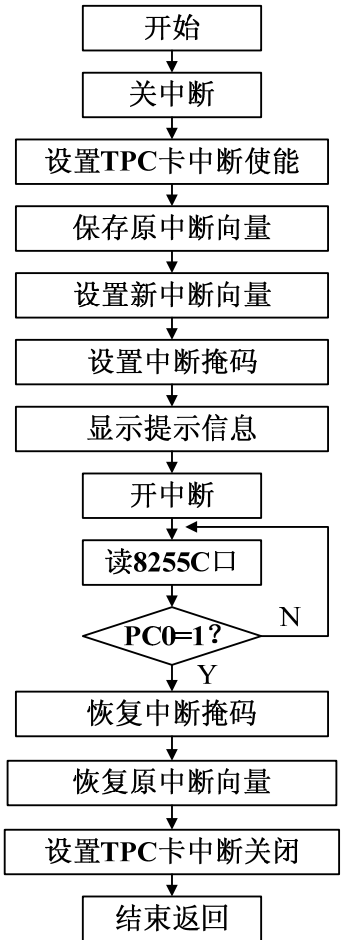

图 3　初始化与主任务程序流程图

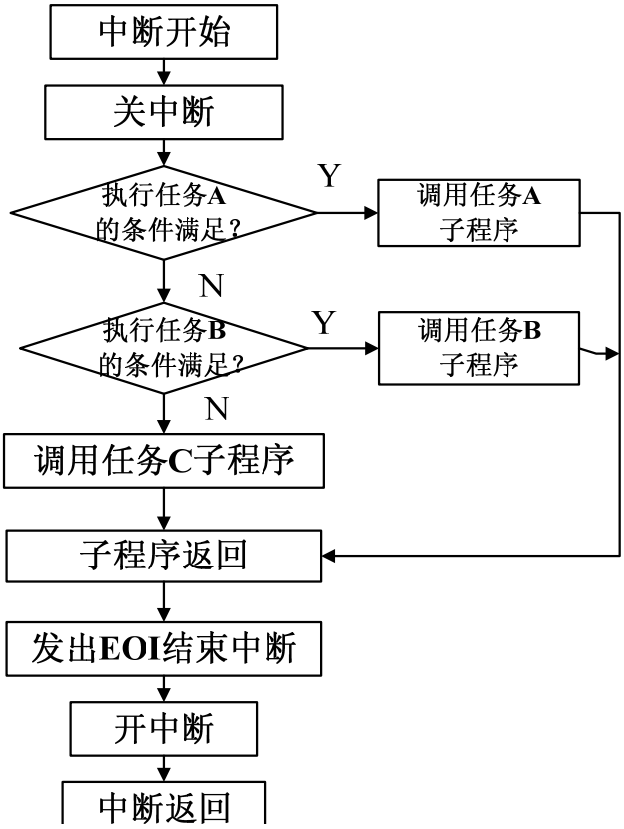

图 3　定时中断调度任务程序流程图

```
；IA16task.asm
;中断来到时，通过判断中断计数值来确定执行任务
;三个任务，A 任务 2MS 执行一次，B 任务 4MS 执行一次，C 任务 4MS 执行一次
;A 任务：在屏幕上显示其运行次数，B 任务：用 DAC0832 产生正弦波；C 任务：8255 实现 C 口读入，A 口输出
;开关 K0 打到 1 时返回 DOS
;---------------------------------------------
data segment
messstr      db 'the schedule task',0dh,0ah,'$'
messend      db 'set the k0 the "1" position to exit',0dh,0ah,'$'
msg1          db 'TPC PCI CARD INTERRUPT',0dh,0ah,'$'
```

;针对不同的中断号，只需要改动四个地方：int_vect；irq_mask_2_7，irq_mask_9_15（两个中断掩码，相应为 0 时清屏蔽）
;ioport_cent（portr 程序读取）

```
  int_vect      EQU      071H          ; 中 断  0-7  的 向 量 为 :08h-0fh, 中 断  8-15  的 向 量 为:70h-77h。这里用的是 IRQ9
 irq_mask_2_7     equ  011111011b   ;中断掩码,中断 0-7 时从低至高相应位为零,中断 8-15 时第 2 位为零
 irq_mask_9_15   equ  011111101b   ;中断 0-7 时全一,中断 8-15 时从低至高相应位为零
 ioport_cent   equ  0c000h          ;tpc 卡中 9054 芯片的 io 地址
 csreg          dw  ?
 ipreg          dw  ?          ;旧中断向量保存空间
 irq_times      dw  00h        ;中断计数

dacpy                db          00h

pianyi   db   00h            ;行保存
colma    db   01h             ;列保存

;TASKA 部分

tames1               db 'TaskA is running. ',0dh,0ah,'$'
tames1len=$-tames1
ta_run_times       dw 00h                        ;taska 运行次数
numbuff             db 30H,30H,30H,30H        ；次数 HEX 对应 ACSCII 码
numbufflen=$-numbuff          ；字符数

;TASKB:   8253 部分端口定义

ioport      equ   0c400h-280h
io8253k     equ   ioport+283h
io82530     equ   ioport+280h
io82531     equ   ioport+281h
tbmes1              db 'Greeting from TaskB. ',0dh,0ah,'$'

;TASKC:   8255 部分端口定义

io8255a             equ ioport+288h
io8255c             equ ioport+28ah
io8255k             equ ioport+28bh
tcmes1              db   'Hello from TaskC .',0dh,0ah,'$'
```

```
buff          DB    44h,54h,54h,7fh,54h,0dch,44h,24h

;双色 LED 显示
proth      equ            ioport+290h
protlr     equ            ioport+298h

SINBUF       db       80h,96h,0aeh,0c5h,0d8h,0e9h,0f5h,0fdh
             db       0ffh,0fdh,0f5h,0e9h,0d8h,0c5h,0aeh,96h
             db       80h,66h,4eh,38h,25h,15h,09h,04h
             db       00h,04h,09h,15h,25h,38h,4eh,66h

cnt1         db    01h
cnt2         db    02h
cnt3         db    00h

data ends


stacks      segment
            db 100 dup (?)
stacks ends

code segment
assume cs:code,ds:data
start:
.386
          cli
            mov ax,data
            mov ds,ax
            mov es,ax
            mov ax,stacks
            mov ss,ax

            mov dx,ioport_cent+68h    ;设置  tpc  卡中 9054 芯片 io 口,使能中断
            in    ax,dx
            or    ax,0900h
            out dx,ax

            mov al,int_vect                ;保存原中断向量
       mov ah,35h
            int 21h
       mov ax,es
       mov csreg,ax
       mov ipreg,bx
```

```
        mov ax,cs                       ;设置新中断向量
    mov ds,ax
    mov dx,offset int_proc
    mov al,int_vect
    mov ah,25h
        int 21h

        in      al, 21h             ;设置中断掩码
        and     al, irq_mask_2_7
        out     21h, al
        in      al, 0a1h
        and     al, irq_mask_9_15
        out     0a1h, al

      mov ax,data
      mov ds,ax

      lea dx,messstr
      mov ah,09h
      int 21h

      lea dx,messend
      mov ah,09h
      int 21h
;sti                            ;sti 的位置有待确定

    mov dx,io8253k          ;给计数器 0 写 8253 控制字，方式 3，先写低字节，再写高字节
    mov al,36h
    out dx,al

    mov dx,io82530          ;给计数器 0 送初值 1000
    mov ax,1000               ;8253 的 CLK0 接 1MHZ,1000 分频后从 OUT0 输出的是 1KHZ
的方波
    out dx,al               ;将 OUT0 接到 IRQ 后，1ms 产生一次中断
    mov al,ah
    out dx,al
 sti
loop1: nop
loop2:
      mov dx,io8255c                ;从 C 口读数据，K0 为 1 时返回 DOS
      in    al,dx
      test al,01h
      jnz    exit
```

```
        jmp loop1

exit:                                   ;结束运行
        cli
            mov         bl, irq_mask_2_7            ;恢复中断掩码
        not   bl
        in    al, 21h
        or    al, bl
        out   21h, al
        mov bl, irq_mask_9_15
        not   bl
        in    al, 0a1h
        or    al, bl
        out   0a1h, al

            mov dx,ipreg                            ;恢复原中断向量
        mov ax,csreg
        mov ds,ax
        mov ah,25h
        mov al,int_vect
            int 21h

            mov dx,ioport_cent+68h                  ;设置 tpc 卡中 9054 芯片 io 口,关闭中断
        in    ax,dx
            and ax,0f7ffh
        out dx,ax

         mov ah,4ch
         int 21h
;定时中断服务: 根据时间(软件定时器)调度
int_proc proc far
        cli
            pusha
            push ds
            push es

            inc cnt1
            inc cnt2
            inc cnt3
chka: cmp    cnt1,02h
          jnz    chkb
          call taska       ;激活运行 A 任务
          mov    cnt1,00h
          jmp    intend
```

```
chkb: cmp cnt2,04h
        jnz    chkc     ;
        call taskb    ; ;  激活运行 B 任务
        mov    cnt2,00h

chkc:; cmp    cnt3,04h
       ; jnz      chkFinal
        call    taskc    ; ;  激活运行 C 任务
        ;CALL TASKD
        mov      cnt3,00h

chkFinal:nop
intend:

            mov al,20h                    ;Send EOI
            out 0a0h,al
            out 20h,al
            pop es
            pop ds
;            pop dx
            popa
            sti
            iret
int_proc endp

taska proc near

   push ax
   push bx
   push cx
   push dx
   push ds
   push es
   push bp
   push si
   push di
        MOV    AX,data
        mov    ds,ax
        MOV    ES,AX
        LEA    BP,tames1
        MOV    DX,1700H      ;Row: Collum No.
        MOV    BH,0
        MOV    CX,tames1len;  串长.
```

```
        MOV    AL,0
        MOV    BL,0D2H    ;属性字
        MOV    AH,13H
        INT    10H          ；显示，可用直接写 VRAM 代替：B8000H

   inc    ta_run_times
   cmp ta_run_times,0ffffh
   jnz tdisp

zerot: mov ta_run_times,01h    ；计数复位

tdisp:    mov ax,ta_run_times    ；计数次数
            call disp                    ；转换成 ASCII 码

        LEA    BP,numbuff    ; ES:BP=被显示串在内存储中的地址
        MOV    DX,1740H    ;DH:DL=起始行号:列号
        MOV    BH,0
        MOV    CX,numbufflen    ；串的字符数(4 个)
        MOV    AL,0                ;选择,光标返回到 DH,DL 指定位置
        MOV    BL,0D2H            ;属性
        MOV    AH,13H
        INT    10H          ;BIOS 调用 13H:,CX

   pop di
   pop si
   pop bp
   pop es
   pop ds
   pop dx
   pop cx
   pop bx
   pop ax
      ret
taska endp

；次数 HEX 转换成 ASCII 码串存入 numbuff 子程序
;入口: AX=运行次数
DISP PROC NEAR
     push dx
     push cx
     push bx
     push si

     mov    cx,4
```

```
    mov    bx,16
     lea si,numbuff
dloop1:     push ax
    push cx
    sub    bx,4
    mov    cx,bx
    shr    ax,cl
    and    al,0fh        ;首先取低四位
    mov    dl,al

    cmp dl,9             ;30h-39h:0-9;A-F:41-46H;a-f:61-66h
    jle num1
    ADD dl,7
num1:
    ADD dl,30h

    mov [si],dl
    inc si

  ; mov ah,02h
  ; int 21h
    pop cx
    pop ax
    loop dloop1
    pop    si
    pop    bx
    pop    cx
    pop    dx

    ret                  ;子程序返回
DISP ENDP

taskb proc near
  ; push ax
  ; push dx
pusha
  ; lea dx,tbmes1
  ; mov ah,09h
  ; int 21h
    lll:
                mov        dx,0c420h              ;DAC0832 接 2A0H
                lea        bx,SINBUF
                mov        al,dacpy
                xlat
```

```
                out         dx,al
            ;   call        WaveOutDelay
                mov     ah,dacpy
                inc     ah
                cmp     ah,32
                je      rsetdacpy
                jmp         setdacpy
rsetdacpy:      mov     ah,00h
setdacpy:       mov     dacpy,ah

 popa
 ret
taskb endp

taskc proc near
    push ax
    push dx

 ;   lea dx,tcmes1
;    mov ah,09h
 ;   int 21h

aa1:        mov dx,io8255k              ;设 8255 为 C 口输入,A 口输出
        mov al,8bh
        out dx,al
inout:      mov dx,io8255c              ;从 C 口输入一数据
        in al,dx
        mov dx,io8255a              ;从 A 口输出刚才自 C 口
        out dx,al                   ;所输入的数据
         ;    jmp aa1
    pop   dx
    pop   ax
    ret
taskc endp

code ends
end   start
```

四. 选作探索与应用设计

(1)样例中将 Taska 改为直接写 PC 的显示存储器(地址:B800:0~3FFFH), 每秒刷新一次;
(2)扩展任务 D: 定时修改日时钟，获得（00-59）秒，（00-23 时，00-59 分）
(3)扩展任务 E：在 8*8 点阵上显示运行基本时间秒（00-59）
（4）根据具体应用构建应用系统和程序框架

五．实验预习与报告

1．实验前阅读基本实验指示书和装置介绍。

2．样例实验仅提供 IA-16 多任务系统调度运性的感性认识，为研讨系统报告提供依据。具体研究报告按课程要求撰写，可讨论相关问题。

3. 选做实验方案及内容讨论,说明 D-E 任务通信关系。

六．参考资料

设备实验指示书 预备实验 1： PCI 设备查询和配置空间的读取

实验三 可编程定时器 / 计数器（8253）

实验四 可编程并行接口（一）（8255 方式 0）

实验九 中断

实验十一 数/模转换器