

## 接口部分复习整理

### 第五章复习

### 第六章复习

### 第七章复习

### 第八章复习

#### 中断程序分析

#### 键盘中断

### 第九章

#### 8250 编程

### 第十章

#### 8255初始化编程

### 第十一章

#### 8254初始化编程

## 接口部分复习整理

### 第五章复习

- 知识点1：
  - 三大总线：地址总线、控制总线、数据总线（填空题）
  - 总线周期：CPU通过总线与存储器、I/O交换一个数据所需要的时间
- 知识点2：M/ IO反、W/R反、D/C反
  - M：表示CPU与存储器交换信息
  - /IO：表示CPU与IO接口交换信息
  - W：表示CPU进行写操作
  - /R：表示CPU进行读操作
  - D：表述传输数据
  - /C：表示传输指令代码

### 第六章复习

- 知识点1：多存储器扩展

$$\text{总片数} = \frac{\text{总容量}}{\text{容量/片}} \quad (1)$$

- 例题1：

例：现有  $16K \times 1$  的静态 RAM 芯片，欲组成  $128K \times 8$  位的存储器，需要 64 片这样的 RAM 芯片，需组成 8 个芯片组，这属于 字位同时 扩展，用于片内地址选择的地址线需用 14 根，至少需用 3 根地址线进行译码来实现不同芯片组的选择。

• 例题2：

**例：存储器容量为  $8K \times 8$ ，若选用 2114 芯片 ( $1K \times 4$ )，则需要：**

$$\frac{8K \times 8}{1K \times 4} = 8 \times 2 = 16 \text{ 片}$$

## 第七章复习

- 知识点1：端口的概念以及分类：
  - 能够与CPU交换信息的寄存器成为I/O端口寄存器，简称"端口"，可分为数据端口、状态端口、控制端口
- 知识点2：端口的编址方式
  - 分为两种：统一编址和独立编址方式。PC系列机采用端口独立编址
- 知识点3：寻址方式
  - 直接寻址的输入输出指令（8位地址）
  - DX间接寻址的输入输出指令（8~16位地址）
- 知识点4：信息交换方式
  - 微机系统与IO端口的信息交换方式
    - 无条件传送
    - 查询方式
    - 中断方式
    - DMA方式（与CPU无关，存储器之间直接操作）
- 知识点5：CPU的输入输出指令：IN\OUT

## 第八章复习

- 中断和中断源
  - 中断是指CPU在执行程序的过程中，由于某种外部或内部事件作用使CPU停止当前正在执行的程序而转去为该事件服务，待事件服务结束后，又能自动返回到被终止了的程序中继续执行的过程
  - 中断源是能够引发中断事件即发出中断请求的来源
- 什么是中断向量？
  - 实模式下，中断服务子程序的入口地址
- 什么是中断向量表？
  - 实模式下，CPU把256种中断向量组成一张表，放在系统RAM最低端的1K单元 (0~3FFH)

- o n型中断向量放在  $4n \sim 4n+3$  单元内
- 中断向量的分类
  - o 内部中断
    - 软件中断
    - 异常
  - o 外部中断（硬件中断）
    - 可屏蔽中断：INTR 引脚输入
    - 非屏蔽中断：NMI 引脚输入
- 响应可屏蔽中断的条件
  - o 有可屏蔽中断请求，没有DMA请求，没有非屏蔽中断请求
  - o CPU一条指令执行完毕
  - o CPU处于开中断状态(I标=1)
- 响应非屏蔽中断的条件
  - o 有非屏蔽中断请求，没有DMA请求
  - o 一条指令执行完
- 执行级别：DMI > 非屏蔽 > 屏蔽
- 中断指令
  - o 开中断指令 STI 使I标志置为1，CPU处于开中断状态（开可屏蔽硬件中断）
  - o 关中断指令 CLI 使I标志置为0，CPU处于关中断状态 注：ST~均为置一，CL~均为置零，STI、CLI指令只对可屏蔽中断有效
  - o 软件中断指令 INT n 无条件转向n型中断服务程序
  - o 中断返回 IRET
    - 一次从栈顶弹出6个字节，IP，CS，F
    - RET指令弹出IP，CS
- 硬件中断源与中断类型表
  - o 其存储类型对应中断向量表的单元内存
  - o 中断优先级主IR0~IR1，从IR0~IR7，主IR3~IR7

对应端口	中断源	中断类型
(主) IR0	日时钟	08H
(主) IR3	辅串口	0BH
(主) IR4	主串口	0CH
(辅) IR0	实时时钟	70H
(辅) IR1	用户中断	71H —> 0AH

- 系统给主从8259的口地址
  - o 其存储类型对应实际接口号，奇地址对应中断屏蔽寄存器的口地址、偶地址对应接受中断结束命令的寄存器口地址

	奇地址	偶地址
主8259A	21H	20H
从8259A	A1H	A0H

- 软硬件中断相同点
  - 都会引起程序终止
  - CPU都会获得终端类型码，通过对应的入口地址获得该中断源的终端服务程序
- 软硬件中断的不同点
  - 引发方法不同：硬件终端是外设通过INTR和NIM引脚发起的，软件中断通过INT n指令执行
  - 获取方式不同，可屏蔽和非屏蔽分别由中断控制器8259A和自动产生；软件中断由指令提供
  - 响应条件不同，硬件终端存在可屏蔽，也存在非屏蔽。软件中断是不可屏蔽的
  - 结束方式不同。可屏蔽中断需要处理结束后发送中断结束命令+IRET，软件中断只需要IRET

## 中断程序分析

- 考试要求掌握
  - 中断向量的置换（读和写中断向量）
  - 硬件中断通路的开放和屏蔽，
  - 例题给出了类似303页的用户中断和键盘中断服务程序，具体代码见代码附录
  - 建议掌握 8.6 至8.7 所有代码
- 第一部分：写入中断向量
  - 方法1：直接向N型向量地址内写入

```

CLI
    PUSH    DS
    MOV     AX, 0000H
    MOV     DS, AX
    MOV     BX, 4*n
    MOV     AX, OFFSET SERVICE
    MOV     [BX], AX
    MOV     AX, SEG SERVICE
    MOV     [BX+2], AX
    POP     DS
    STI

```

- 方法2：调用21H功能写入中断向量

[ 35H 功能 ]  
 功能： 读出n型中断向量  
 入口： AL=中断类型码  
 出口： ES:BX=n型中断向量

[ 25H 功能 ]  
 功能： 写入n型中断向量  
 入口： DS:DX=段基址:偏移地址  
       AL=中断类型码  
 出口： 无

示例：

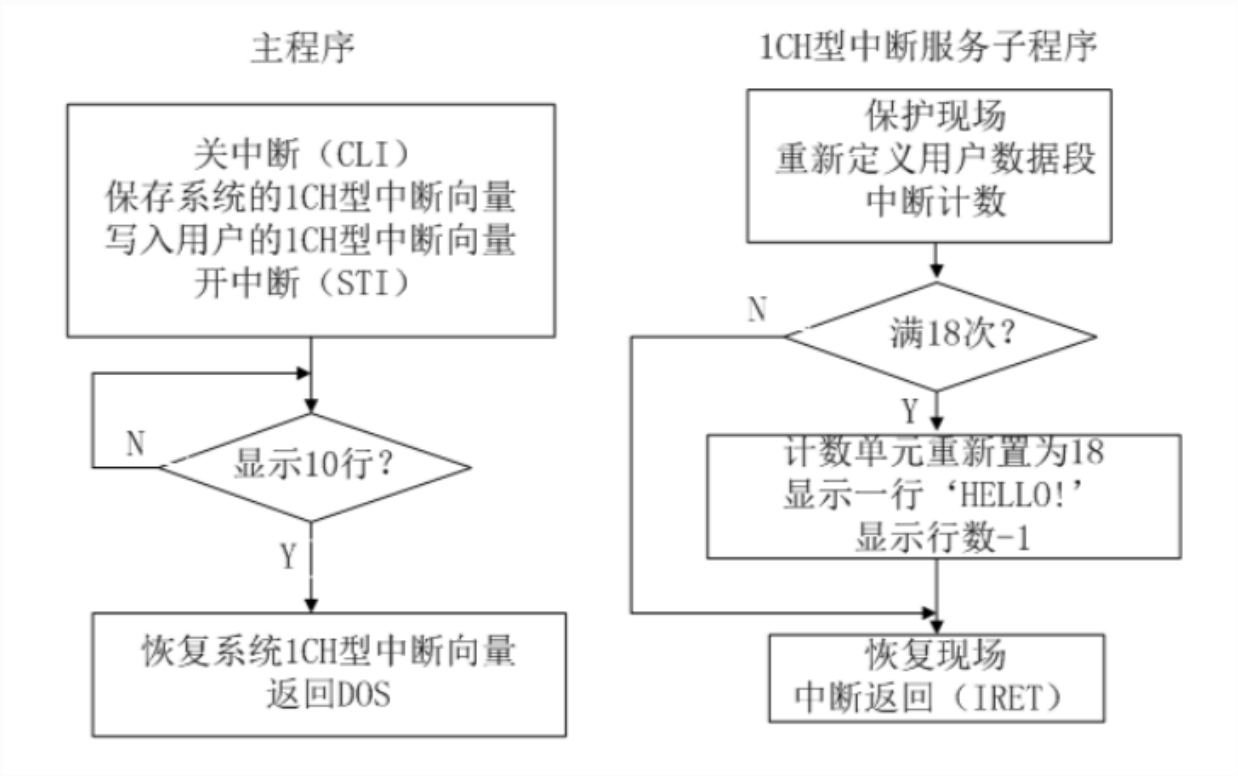
```

CLI
PUSH DS
PUSH A
MOV AX,SEG SERVICE
MOV DS,AX
MOV DX,OFFSET SERVICE
MOV AH,25H
MOV AL,n
INT 21H
POPA
POP DS
STI

```

- 中断服务程序&8259原理
- 第二部分：中断服务程序的处理过程
  - F 寄存器压入栈、段基址： 偏移地址压入栈
  - 执行中断服务程序
  - 恢复偏移地址，段基址， 栈
  - 中断返回
- 第三部分：8259A——中断控制
  - 结构：信号区->中断请求寄存器 -> 中断屏蔽寄存器 -> 中断服务寄存器 -> CPU
  - 将IR0-ir7的8个可能的中断信号，分别通过屏蔽、择优两个步骤进行控制
  - 筛选完成后，从INT端口送出中断请求信号，由/INTA 返回第一个中断信号时，将请求寄存器的中断信号清零（清除中断信号源），中断服务寄存器置1（正在服务状态）
  - /INTA 第二个信号返回，发送中断类型码，执行中断服务程序，在服务程序结束前写结束字（如果多个服务程序嵌套运行，那么最后一个执行语句的相应服务中需要写入结束字）
  - 结束后，继续执行中断判断
- 系统日时钟介绍
  - 系统中有一个每隔55ms的08H向量，称为日时钟
  - 08H向量中内嵌了一个1CH向量，用于日时钟开发
  - 我们可以通过修改08H或者1CH程序来达到日时钟开发的目的
- 复习提纲程序用户中断例题

要求利用PC系统机上的8254的0号定时计数器引发的日时钟中断，设计程序:每间隔1秒在PC终端屏幕上显示1行字符串“HELLO!”，显示10行后结束。【例题所给为修改1CH程序进行的日时钟开发】



● 程序解析

```
[主程序段]
BEG:      MOV AX,DATA
          MOV DS,AX
          CLI                                ;关中断，防止替换过程日时钟程序跑动
          CALL READ1C
          CALL WRITE1C
          STI                                ;开中断，日时钟程序正常工作
SCAN:     CMP COUNT,0                       ;判断是否执行10次
          JNZ SCAN                          ;是否已经显示10行，否转 CALL RESET
          CALL RESET                        ;还原日时钟
          MOV AH,4CH
          INT 21H                           ;结束程序
```

关键掌握上面的程序框架，之后的程序除了替换到日时钟程序中的service 就是读向量、写向量、还原写向量三个程序，重点熟练掌握25H、35H功能调用。

```
[自定义日时钟段]
SERVICE PROC
          PUSHA                            ;保护现场
          PUSH DS                          ;
          MOV AX,DATA                      ;数据段预设
          MOV DS,AX
          DEC ICOUNT
          JNZ EXIT
```



```

        INT 21H
        MOV WORD PTR OLD1C,BX
        MOV WORD PTR OLD1C+2,ES
        RET
READ1C ENDP
;-----
WRITE1C PROC                                ;写入用户1CH型中断向量
        PUSH DS
        MOV AX,CODE
        MOV DS,AX
        MOV DX,OFFSET SERVICE
        MOV AX,251CH
        INT 21H
        POP DS
        RET
WRITE1C ENDP
;-----
RESET PROC                                ;恢复系统1CH型中断向量
PTR OLD1C
        MOV DX,WORD PTR OLD1C
        MOV DS,WORD PTR OLD1C+2 ;不可互换位置，DS会影响代码执行位置
        MOV AX,251CH
        INT 21H
        RET
RESET ENDP
CODE ENDS
        END BEG

```

- 08H 代码解析

```

[主程序段]
CODE SEGMENT
OLD08 DD ?                                ;关键：代码段保存系统08H中断向量中的变量

BEG:    MOV AX,DATA
        MOV DS,AX
        CLI
        CALL READ08
        CALL WRITE08
        STI
SCAN:   CMP COUNT,0
        JNZ SCAN
        CALL RESET
        MOV AH,4CH
        INT 21H                            ;结束程序

```

关键掌握上面的程序框架，之后的程序除了替换到日时钟程序中的service 就是读向量、写向量、还原写向量三个程序，重点熟练掌握25H、35H功能调用。

```

[自定义日时钟段]
SERVICE PROC

```



```

        PUSHA                                ;保护现场
        PUSH DS                                ;
        MOV AX,DATA                          ;数据段预设
        MOV DS,AX
        DEC ICOUNT
        JNZ EXIT
        MOV ICOUNT,18                        ;ICOUNT*55ms即为下面代码执行一次相邻
的时间
        DEC COUNT                            ;重新给DS赋值 ;中断计数

        <替换程序的内容>

EXIT:    POP DS
        POPA
        JMP CS:ODL08                        ;跳转回原来的08H程序就不用再写结束字了

```

## 键盘中断

【例】假设微机系统外扩了如下的一个‘单脉冲发生器’，该‘单脉冲发生器’电路受一个自复开关K的控制，每按一次K，该电路输出一个正脉冲，输入到系统机从8259的IR1作为外部中断请求。

要求:每按一次K，屏幕上显示一行字符串“Welcome!”。主机键盘按任意键，程序结束，返回DOS。

- 编写开放8259和中断向量置换程序

```

;开放用户中断(记, 不变)
I8259A PROC
    IN     AL,21H
    AND    AL,11111011B        ; 从8259A IMR1置零
    OUT    21H,AL
    IN     AL,0A1H
    AND    AL,11111101B        ; 主8259A IMR2置零
    OUT    0A1H,AL
    RET
I8259A ENDP

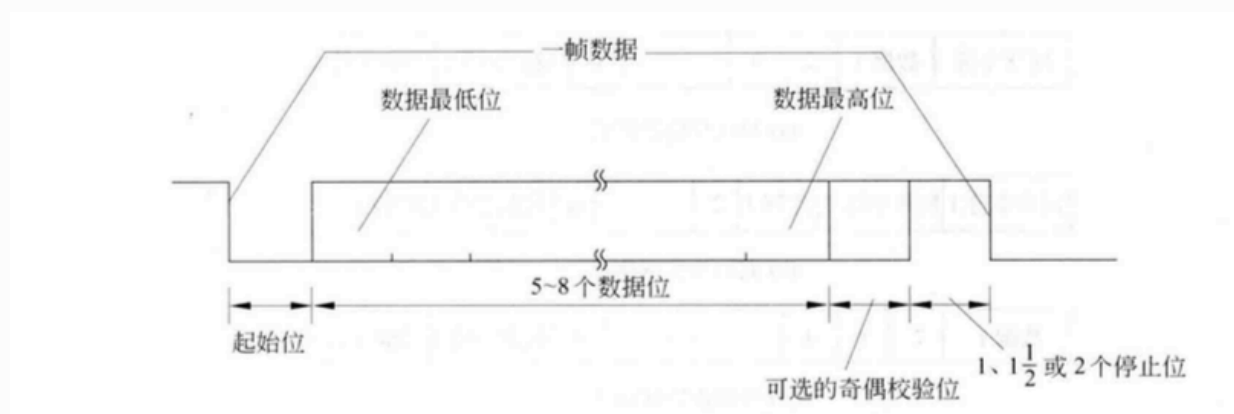
;-----
WRITE0A PROC
    PUSH   DS
    MOV    AX,CODE
    MOV    DS,AX
    MOV    DX,OFFSET SERVICE
    MOV    AX,250AH            ;或者可以将0AH 换成71H
    INT    21H
    POP    DS
    RET
WRITE0A ENDP

;-----

```

## 第九章

- 异步串行通信：



- 一帧数据格式

一帧数据包括起始位、数据位、奇偶校验位、停止位

- 起始位

无数据传输则为1，有数据传输为0，检测到0开始接受数据

- 数据位

数据位个数5~8位，低位先传送，读信号反向读取

- 校验位

双方约定为检验逻辑1的奇校验或者偶校验，然后通过该位进行检验，也可以选择无校验位

- 停止位

可以是1、1.5、2位的逻辑1信号

由此我们可以得到1帧数据的范围为7~12位

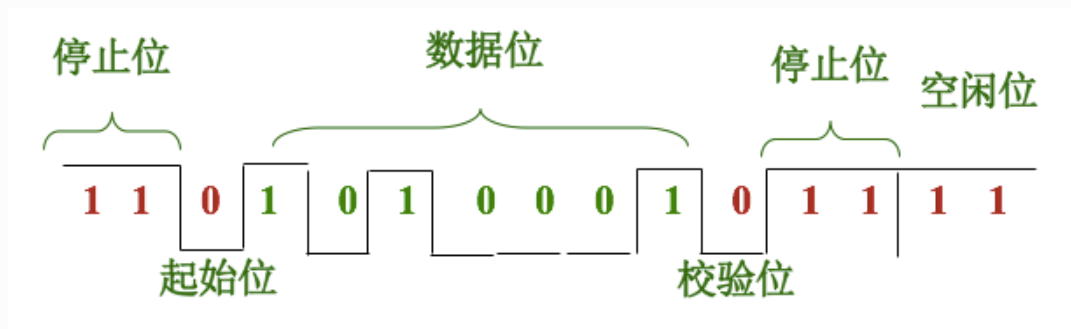
- 通信速率

- 数据信号速率：单位时间传送信号的个数，单位为波特（B/S），称波特率、信号速率

- 数据通信速率：单位时间传送二进制比特的个数，单位为比特/秒（bps, b/s），称比特率或传信速率

- 我们采用二进制信号传输时，波特率=比特率，一个信号为一位二进制

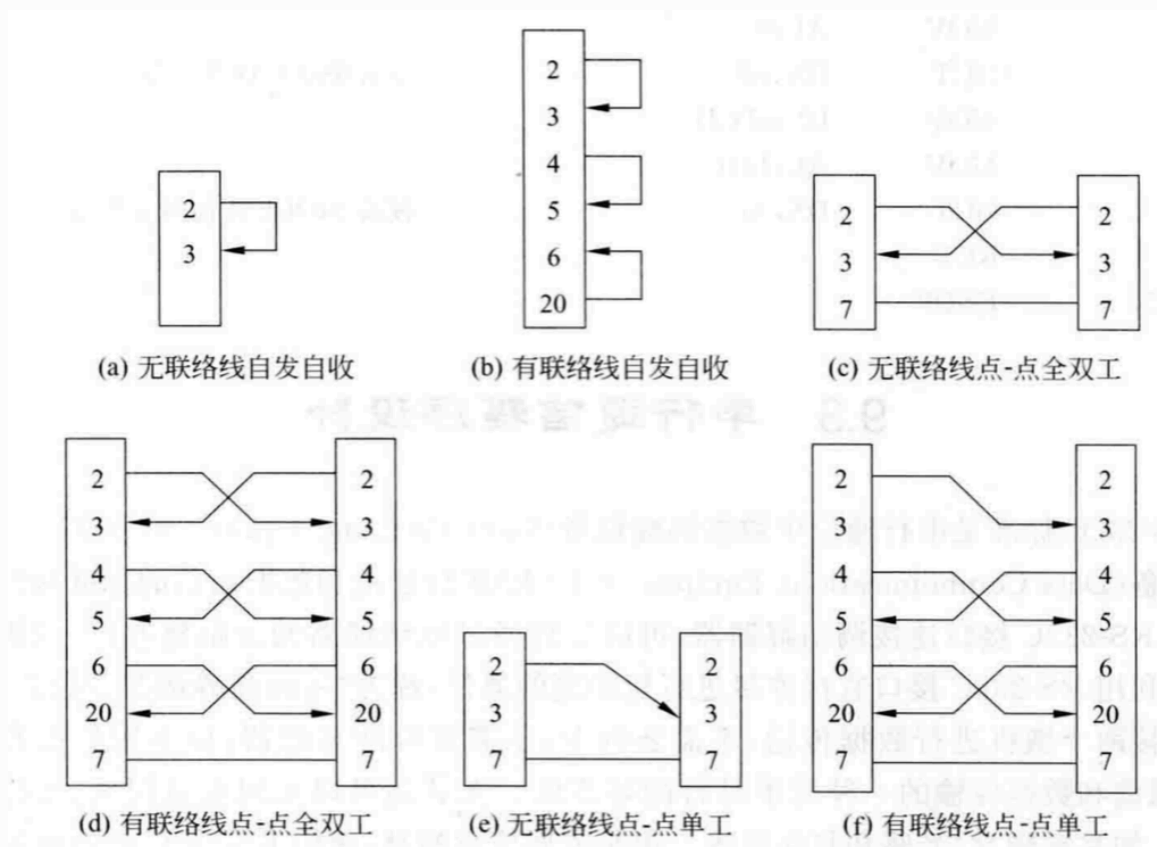
- 例如：传送一个字符“E”，（ASCII码为：1000101B =45H）7个数据位,奇校验, 2个停止位.



例：设异步通信一帧字符有 8 个数据位，无校验，1 个停止位，如果波特率为 9600，则每秒钟能传输多少帧字符？(960) 传输一帧字符需要多少时间？(1/960 秒)

RS-232C 串行接口标准名称记忆（看英文记）

- TXD：发送数据（transmit Data）
- RXD：接受数据（Receive Data）
- RTS反：请求发送（Request to send）
- CTS反：允许发送（Clear to Send）
- DSR反：数据设备准备好（Data set Ready）
- DTR反：数据终端准备好（Data Term Ready）
- RS-232采用负逻辑，逻辑1为-15~-3V，逻辑0为+3V~15V
- 学会绘制TTL电平信号以及负逻辑的232的电平信号
- 8250 内部有四级中断，从高到低依次排序
  - 接受数据错中断
  - 接受中断
  - 发送中断
  - Modem中断
- 串行通信的分类



## 8250 编程

编程分为三个步骤：初始化、查询方式\中断方式、输入输出

### 【附录内容】

#### 1. 通信线控制寄存器 (3FB/2FB)

D7	D6	D5 D4 D3	D2	D1 D0
寻址	中止	校验位	停止位	数据位

(1) 寻址位：D7=1，访问除数寄存器；D7=0，访问非除数寄存器

(2) 中止位：D6=0，正常通信；D6=1，输出长时间中止信号

(3) 校验位：D5~D3=XX0，无校验位；

D5~D3=001，奇校验；

D5~D3=011，偶校验；

D5~D3=101，1校验位；

D5~D3=111，零校验位

(4) 停止位：D2=0，1位停止位；

D2=1，对应5数据位，1.5位停止位；

D2=1，对应7、8位数据，2位停止位；

(5) 数据位：D1D0=00，5位数据；

D1D0=10，7位数据；

D1D0=11，8位数据；

#### 2. 通信线状态寄存器 (3FDH/2FDH)

D7=0

D6：发送移位寄存器忙闲标志位

D5：发送保持寄存器忙闲标志位

D4：线路中断标志位

D3：帧错标志位

D2:奇偶错标志位  
D1:溢出错标志位  
D0:接受数据准备标志位

### 3.MODEM控制寄存器(3FCH/2FCH)

D7~D5=000  
D4=0正常收发, D4=1,内环方式  
D3: 确定\OUT2引脚的输出电平  
D2: 确定\OUT1引脚的输出电平  
D1: 确定\RTS引脚的输出电平  
D0: 确定\DTR引脚的输出电平

### 4.中断允许寄存器(3F9H/2F9H)

D7~D4=0000  
D3: Modem状态中断允许位  
D2: 接受字符错中断允许位  
D1: 发送数据中断允许位  
D0: 接受数据中断允许位

### 5.除数寄存器高8位(3F9H/2F9H),低8位(3F8H/2F8H)

### 6.发送保持寄存器&接收缓冲寄存器(3F8H/2F8H)

- 8250含有发送保持寄存器、接收缓冲寄存器、通信线状态寄存器、中断允许寄存器、中断识别寄存器、除数寄存器、通信线控制寄存器、MODEM状态寄存器

- 编程题型: 对端口直接编程的初始化; 查询方式下接受和发送

- 查询方式下接受和发送程序

在发送数据前, 读通信线状态寄存器(状态口)获取发送保持或移位寄存器(数据口)是否空闲;在接收数据前, 读通信线状态寄存器(状态口)获取接收缓冲寄存器(数据口)是否已经收到1帧数据。

;EG:利用主串口查询方式发送一个“A”

SCANT:

```
MOV     DX, 3FDH
IN      AL, DX
TEST    AL, 20H; 00100000      ;查询发送保持位状态
JZ      SCANT; D5=1
MOV     DX, 3F8H
MOV     AL, 'A'
OUT     DX, AL
```

;EG:利用主串口查询方式接收一个字符

SCANR:

```
MOV     DX, 3FDH
IN      AL, DX
TEST    AL, 01H; 00000001      ;查询接受数据位状态
JZ      SCANR ; D0=1
MOV     DX, 3F8H
IN      AL, DX
```

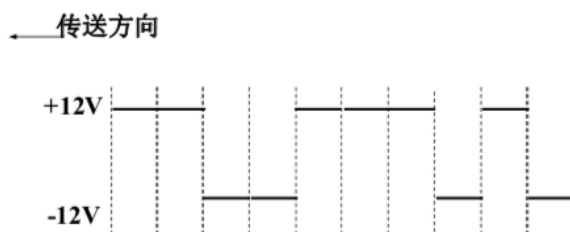
- 对端口直接编程的初始化

- A、B两台PC机利用主串口进行点-点单工通信(不用联络线)，发送采用查询方式，接收采用中断方式。一帧字符包含7个数据位，1个停止位，1个校验位，通信速率为4800波特(分频系数为0018H)。

(1) 知识点：异步通信的分类

(2)

(2) 下图是从PC机的RS-232C接口引脚观察到的波形, 所传送字符的16进制ASCII码是\_\_\_\_\_ ; 该帧数据采用的奇偶校验方式是\_\_\_\_\_ 校验; 传送该帧数据需要的时间是\_\_\_\_\_。



(3) 用对端口直接编程的方法为接收方编写8250初始化程序段。

## 初始化步骤

- 设置寻址位，使通信线控制器设置为除数状态（操作对象：线控制寄存器）
- 将除数传到到相应的端口（操作对象：除数寄存器高低八位）
- 设置中断允许寄存器，查询方式为0，中断方式，相应为1
- 设置MODEM控制寄存器

I8250 PROC

```
MOV      DX, 3FBH
MOV      AL, 80H
OUT      DX, AL          ;在控制器中设置除数状态
MOV      DX, 3F9H
MOV      AL, 00H
OUT      DX, AL          ;设置高八位除数
MOV      DX, 3F8H
MOV      AL, 18H          ;设置低八位除数
OUT      DX, AL
MOV      DX, 3FBH
MOV      AL, 0AH
OUT      DX, AL          ;设置一帧数据格式，更改为非除数操作
MOV      DX, 3F9H
MOV      AL, 01H
OUT      DX, AL          ;设置中断允许寄存器
MOV      DX, 3FCH
MOV      AL, 08H
OUT      DX, AL          ;设置MODEM寄存器
RET
50      ENDP
```

## 第十章

- 8255A有4个端口寄存器

- A数据口地址=60H
  - B数据口地址=61H
  - C数据口地址=62H
  - 控制口地址 = 63H
- 工作方式
  - A口：0、1、2方式
  - B口：0、1方式
  - C口：0方式
  - 方式0：基本输入输出型
  - 方式1：选通型输入输出
  - 方式2：双向数据输入输出

只有A口可以双向输入输出
- 方式特点
  - A口与C口高4位为一组；B口与C口低4位一组；
  - 方式0：直接采用IN、OUT进行读写
  - 方式1：输入前确定IBF=1，表示A、B口可读入数据；  
输出前确定OBF=1，表示A、B口可以写出数据；

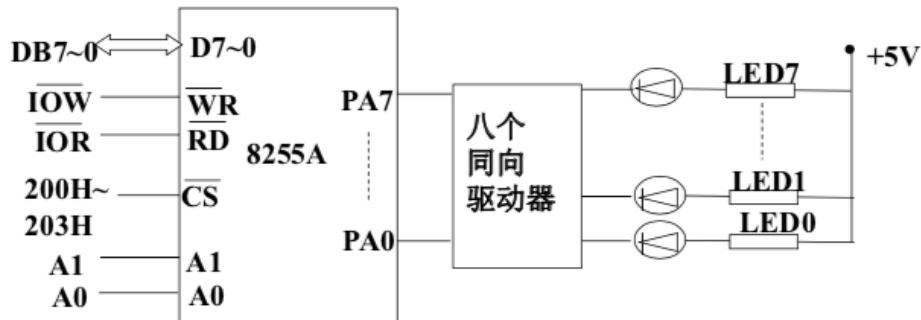
- 学会读图

例：8255A的B口初始化定义为选通型(方式1)输入，对8255A采用查询方式，必须先查询IBF(PC1)；若采用中断方式，必须先置PC /STB (PC2) 置为'1'，并且利用 PC0 (INTR) 作为中断请求信号线。

## 8255初始化编程

- 初始化编程步骤
  - 方式选择命令字→控制口，目的选择某一口的工作方式
  - 根据需要:把C口置0/置1命令字→控制口，目的是禁止/允许某一口提中断
- 例题

系统机外扩一片8255A及相应电路如下图所示，外扩8255A的端口地址为200H~203H,现利用系统机的日时钟外扩1CH型中断，实现每隔1s使八个发光二极管同时闪烁一次，主机键盘有按键按下时结束，返回DOS操作系统。根据要求完成相关内容。



- (1)从图可以分析出，A口工作在方式 0 的输 出。
- (2)假设8255A的A口工作在方式1的查询输出方式，编写8255A的 初始化子程序 I8255。

```

I8255A      PROC
                MOV DX,203H
                MOV AL,10100000B
                OUT DX,AL                ;写入工作方式字
                MOV AL,00001100B
                OUT DX,AL
                RET
I8255      ENDP

```

## 【附录】

### 1.8255A方式选择控制字

(1)D7=1：置方式标志

(2)A组控制字：

D6D5=00，端口A方式0

D6D5=01，端口A方式1

D6D5=1X，端口A方式2

D4=1，端口A输入；D4=0，端口A输出；

D3=1，端口C输入；D3=0，端口C输出；

(3)B组控制字：

D2=0，端口B方式0

D2=1，端口B方式1

D1=1，端口B输入；D1=0，端口B输出

D0=1，端口C输入；D0=0，端口C输出

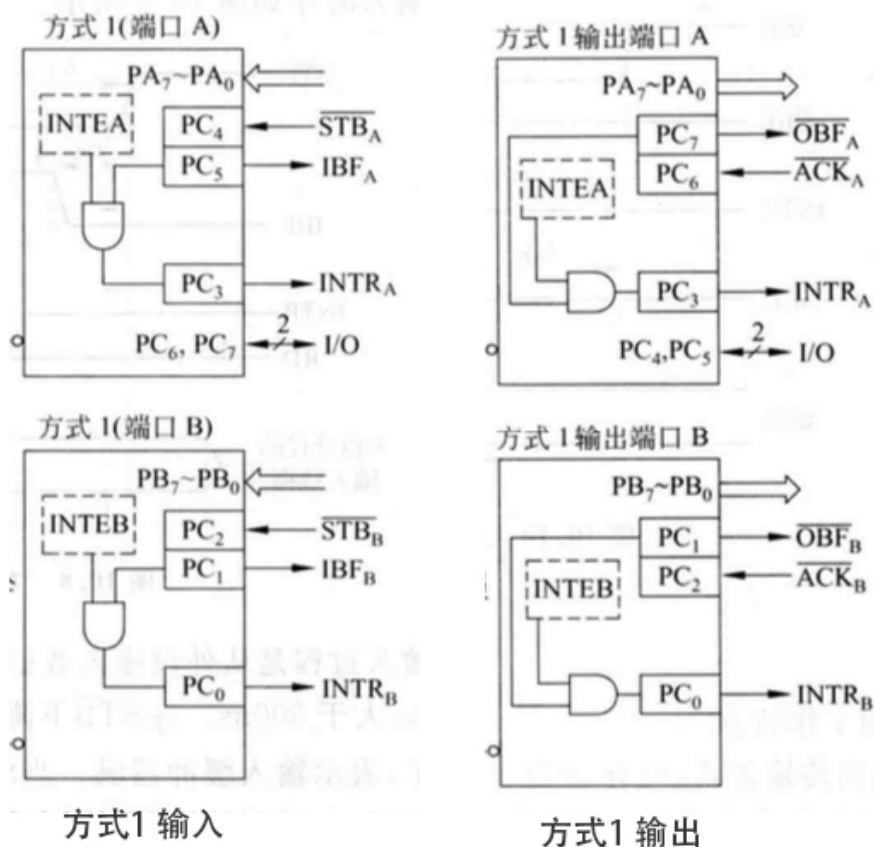
### 2.8255A C口按位置0/置1控制字的格式

D7~D4=0000 D0=0 PCn清零

D0=1 PCn置一

D3~D1：000~111 分别设置PC0~PC7



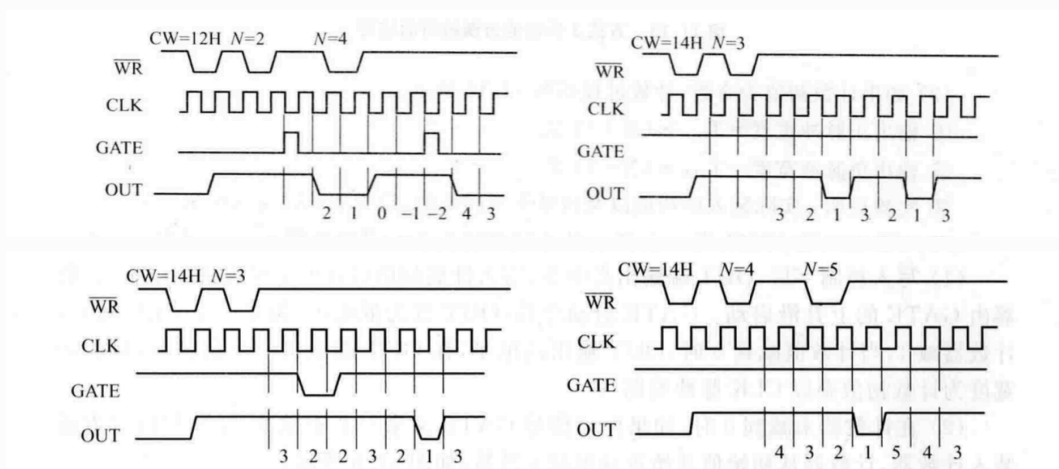


## 第十一章

- 8254内部集成了3个16位的计数器,每个计数器有6种工作方式,计数初值可设定为二进制或BCD码。最高工作频率10兆。
- 组成部分: 数据总线缓冲器(8位)、读写控制、控制字寄存器、0号~2号计数器
- 实现功能
  - CPU通过数据总线缓冲器向8254写入确定工作方式命令字
  - 向某一计数器写入计数初值
  - 读取当前计数器计数值
- GATE 信号触发方式:
  - 持续正信号正常计数
  - 上升沿触发初值计数
  - 每一种工作方式下都有两种触发方式的计数过程

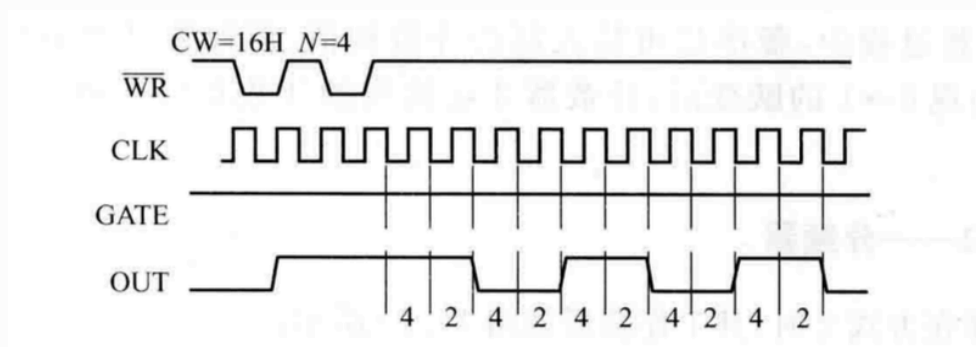
- 六大工作方式（重点2、3,建议看书 P376）

- 方式0：计数结束输出正信号（自动重装新初值）
- 方式1：单脉冲发生器（门控重装）
- 方式2：分频器（自动重装、门控重装、初始输出为正，计数至1为低电平）

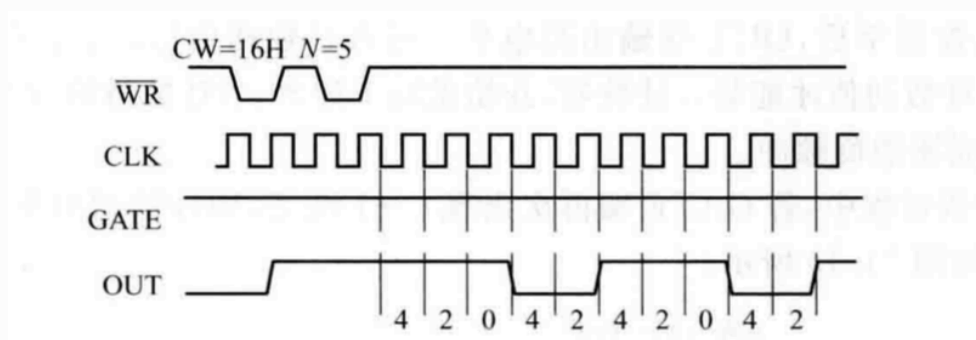


- 方式3：方波发生器

- 偶数初值：每一计数周期换一次输出电平，每次计数减2



- 奇数初值：高电平比低电平计数时长+1，低电平无零状态（计数4,2,0,4,2,4,2,0,...）



- 方式4：软件触发（无重装，计数到零输出一个低电平）
- 方式5：硬件触发（有新初值自动重装、门控重装、计数零输出一个低电平）

- 注意事项

- 当计数初值>9999时，计数初值只能选二进制
- 程序中表示十进制下的1234 应该如下表示 `MOV AX,1234H`

## 8254初始化编程

## 【附录内容】

### 8254控制字

D7	D6	D5	D4	D3	D2	D1	D0
计数器选择		读写方式选择		工作方式选择		数制选择	

#### (1) 计数器选择:

- D7D6=00, 0号计数器
- D7D6=01, 1号计数器
- D7D6=10, 2号计数器
- D7D6=11, 读出控制字的标志

#### (2) 读写方式选择:

- D5D4=00, 表示锁存器计数器的当前值
- D5D4=01, 读出、写入, 只对低八位操作
- D5D4=10, 读出、写入, 只对高八位操作
- D5D4=11, 先读写低八位, 后读写高八位

#### (3) 工作方式选择:

- D3D2D1=000, 工作方式0
- D3D2D1=001, 工作方式1
- D3D2D1=X10, 工作方式2
- D3D2D1=X11, 工作方式3
- D3D2D1=100, 工作方式4
- D3D2D1=101, 工作方式5

#### (4) 数值选择:

- D0=0, 二进制初值
- D0=1, 二-十进制初值

## ● 例题

设PC 系统机外扩了一片8254 及相应的实验电路。8254口地址为200H~203H, 设CLK0 已接至8MHz 时钟, 为了能从OUT0 输出4KHz 的方波, 编写了8254初始化程序, 其中假设0 号定时器工作在二进制方式。

### 程序步骤

- 向末地址写入控制字
- 向首地址写入分频数
- 得到的信号频率为:  $CP / \text{分频数} = F$
- 当分频数=0, 最大满分频

### I8254 PROC

```
MOV     DX, 203H
MOV     AL, 00110110B
OUT     DX, AL
MOV     DX, 200H
MOV     AX, 2000
OUT     DX, AL
MOV     AL, AH
OUT     DX, AL
RET
```

