



# 第6章 单片机的中断与定时系统

---

6.1 MCS-51单片机中断系统

6.2 MCS-51单片机的定时器/计数器

6.3 外部中断源的扩展

6.4 定时/计数器与中断的应用举例：  
时钟计时

# 6.1 MCS-51单片机中断系统

## 一、中断技术概述

**中断技术：**一个CPU面对多项任务，采用中断技术可解决资源共享问题，使CPU分时操作。

**中断技术实现的功能：**CPU与外设速度配合，实时控制，故障检测与处理，人机联系，多机系统等。

# 中断处理流程

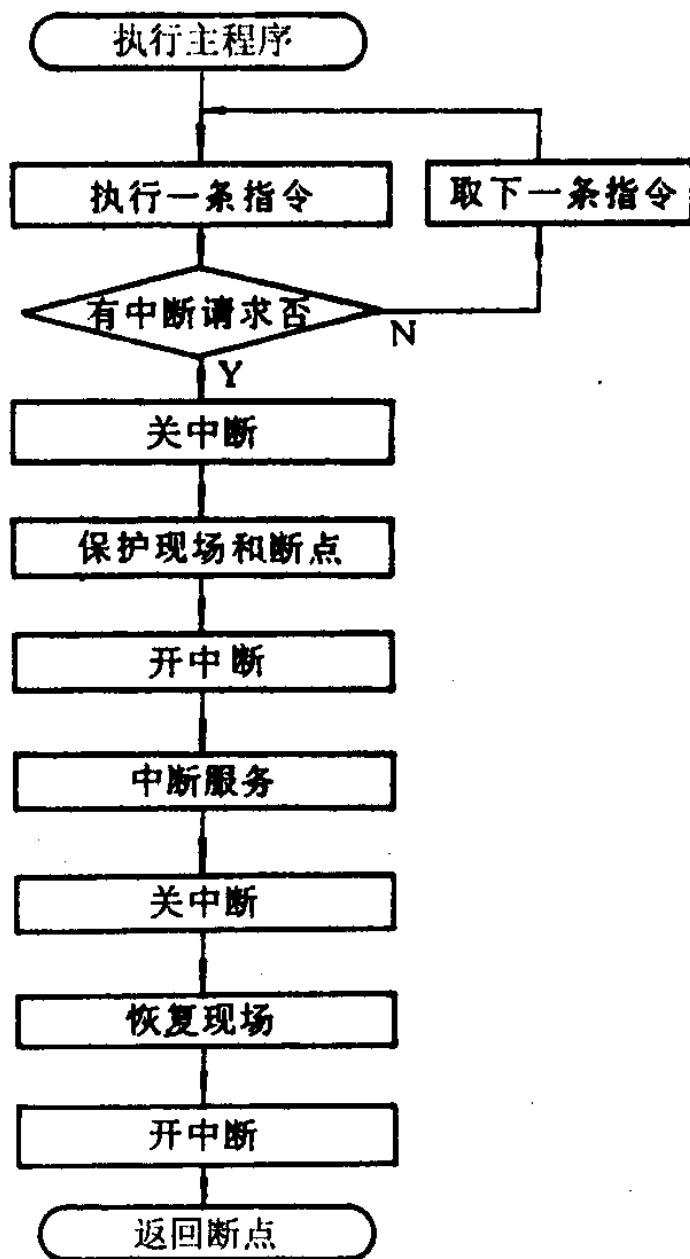


图 6.3 中断服务流程图



# 8051共5个中断源

**外部中断0:** /INT0 (P3.2), 电平方式（低电平有效）或脉冲方式（负跳变有效）；入口0003H

**外部中断1:** /INT1 (P3.3), 电平方式（低电平有效）或脉冲方式（负跳变有效）；入口0013H

**定时中断0:** 技术溢出时发出中断；入口000BH

**定时中断1:** 技术溢出时发出中断；入口001BH

**串行中断:** 串行口接收或发送完一组数据时发出中断。入口0023H

# 8051共5个中断源（图）

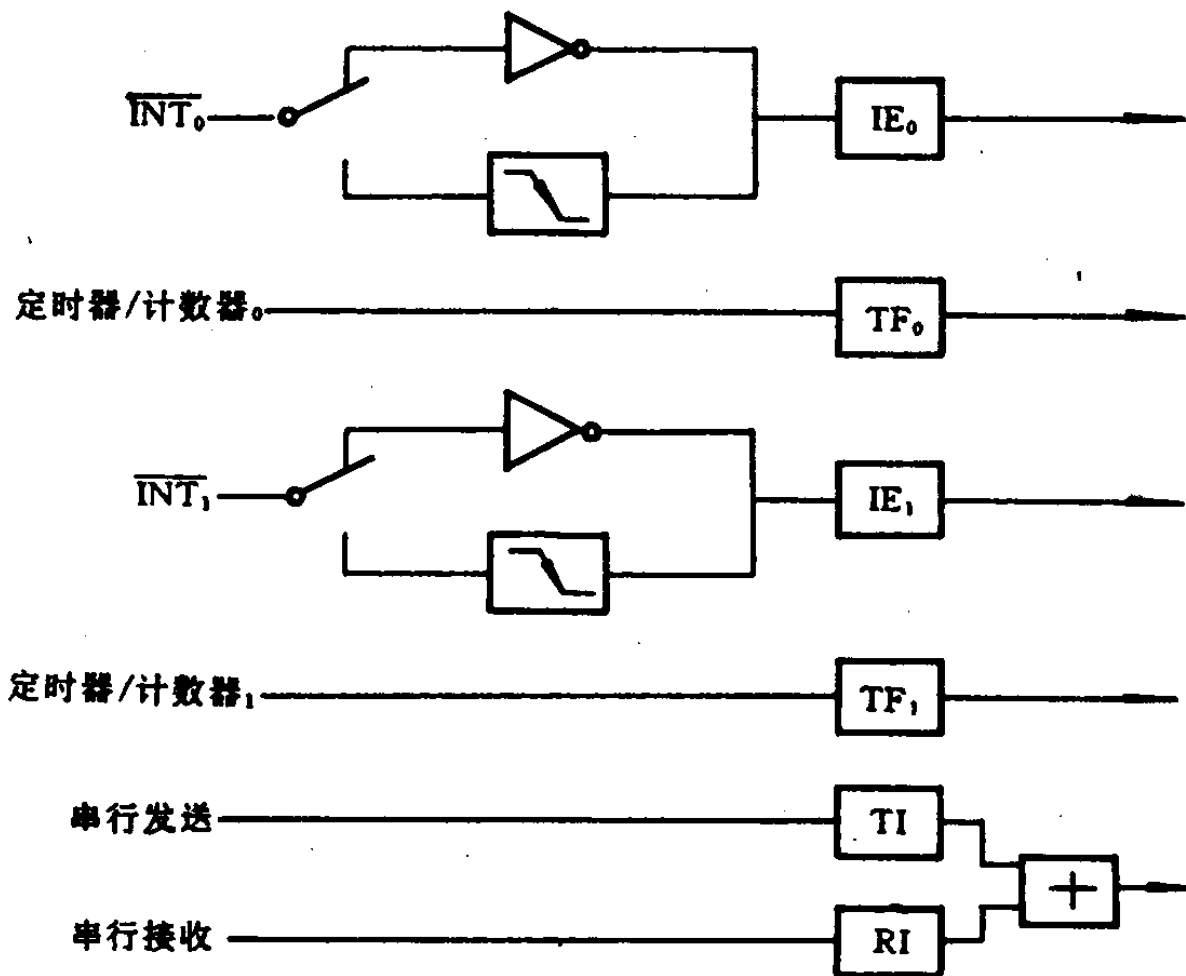


图 6.1 MCS-51 的中断汇集



# MCS-51的中断控制

与中断控制有关的寄存器有**4**个：

定时控制寄存器（**TCON**）；

中断允许控制寄存器（**IE**）；

中断优先控制寄存器（**IP**）；

串行口控制寄存器（**SCON**）。

# 定时控制寄存器 (TCON)

位地址	8FH	8EH	8DH	8CH	8BH	8AH	89H	88H
位符号	TF <sub>1</sub>	TR <sub>1</sub>	TF <sub>0</sub>	TR <sub>0</sub>	IE <sub>1</sub>	IT <sub>1</sub>	IE <sub>0</sub>	IT <sub>0</sub>

1 溢出标志位

查询方式时软件清 0，中断方式时硬件清 0

1 运行控制位

1：工作，0：停止，软件设置

0 溢出标志位

查询方式时软件清 0，中断方式时硬件清

1 运行控制位

1：工作，0：停止，软件设置

外中断 1 请求标志位

电平方式 0，脉冲方式 1  
外中断 1 方式控制位

外中断 0 请求标志位

电平方式 0，脉冲方式 1  
外中断 0 方式控制位

# 串行口控制寄存器 (SCON)

位地址	9FH	9EH	9DH	9CH	9BH	9AH	99H	98H
位符号	SM <sub>0</sub>	SM <sub>1</sub>	SM <sub>2</sub>	REN	TB <sub>8</sub>	RB <sub>8</sub>	TI	RI

工作方式选择

多机通信控制位

允许接收位

发送数据位 8

接收数据位 8

串行口发送中断请求标志位

串行口接收中断请求标志位



# 中断允许控制寄存器 (IE)

位地址	0AFH	0AEH	0ADH	0ACH	0ABH	0AAH	0A9H	0A8H
位符号	EA	/	/	ES	ET <sub>1</sub>	EX <sub>1</sub>	ET <sub>0</sub>	EX <sub>0</sub>

总允许1，总禁止0  
中断允许总控制位

串行中断允许位  
0 允许1，禁止0  
1 中断允许

外中断允许位  
1 允许1，禁止0  
0 中断1，禁止0

外中断允许位  
1 允许1，禁止0  
0 中断允许

外中断允许位  
1 允许1，禁止0  
0 中断允许

# 中断优先级控制寄存器 (IP)

位地址	0BFH	0BEH	0BDH	0BCH	0BBH	0BAH	0B9H	0B8H
位符号	/	/	/	PS	PT <sub>1</sub>	PX <sub>1</sub>	PT <sub>0</sub>	PX <sub>0</sub>

**0:** 优先级低

**1:** 优先级高

串行中断优先级设定

**1** 中断优先级设定

外中断 **1** 优先级设定

**0** 中断优先级设定

外中断 **0** 优先级设定



# 中断优先级控制原则

- 1、低级不能打断高级，但高级可以打断低级；
- 2、一个中断已被响应，则同级中断响应将被禁止；
- 3、多个同级中断同时出现，则按**CPU**查询次序：**INT0→T0→INT1→T1→S**。

# 中断响应过程

外部中断  
请求采样

中断  
查询

中断  
响应

(CPU在每个机器周期的S5P2对/INT0和/INT1进行采样, 将识别后的外部中断请求信号锁定在TCON中)

(CPU在每个机器周期的S6按优先级顺序对中断请求进行查询, 当标志位为1时, 从下个机器周期的S1开始响应)

(由硬件生成一条LCALL指令, 转向中断相应的入口地址)

共3~8个机器周期



# 中断请求的撤除

中断响应后，应将**TCON**、**SCON**中的请求标志位清除。

**定时中断：**硬件自动撤除；

**串行中断：**不自动清零，中断响应后，要判断接收还是发送，然后进行软件撤除；

**外部中断：**

脉冲请求方式：自动撤除；电平请求方式：中断标志自动撤除 + 中断请求信号的强制撤除。

## 中断请求信号的强制撤除

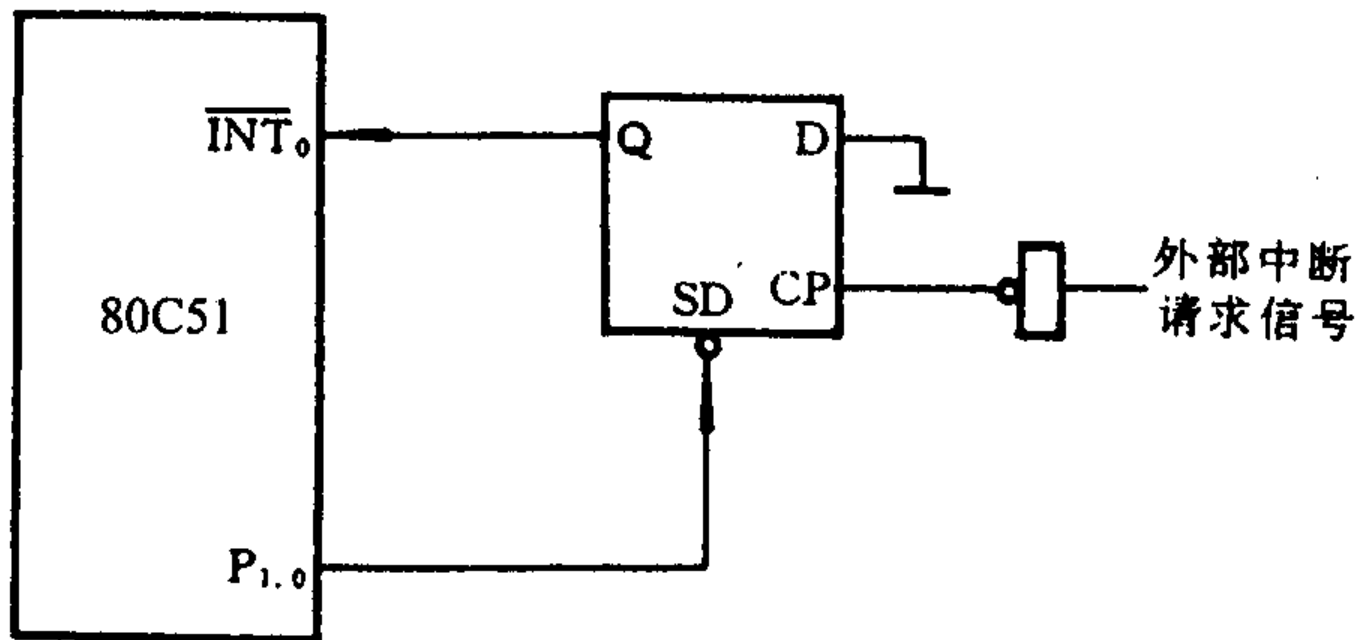


图 6.2 电平方式外部中断请求的撤销电路

## 6.2 MCS-51单片机的定时器/计数器

一、**定时方法**：软件定时，硬件定时，可编程定时（对系统时钟脉冲进行计数）

二、**定时/计数器的定时功能和计数功能**

计数：T0(P3.4)、T1(P3.5)可以对外来脉冲进行计数，负跳变有效，计数器加1；

定时：每个机器周期（12个振荡周期）产生一个计数脉冲，计数器加1。

三、**定时/计数器的控制寄存器**：有3个控制寄存器有关

# TCON (定时控制寄存器)

位地址	8FH	8EH	8DH	8CH	8BH	8AH	89H	88H
位符号	TF <sub>1</sub>	TR <sub>1</sub>	TF <sub>0</sub>	TR <sub>0</sub>	IE <sub>1</sub>	IT <sub>1</sub>	IE <sub>0</sub>	IT <sub>0</sub>

1 溢出标志位

查询方式时软件清0，中断方式时硬件清0

1 运行控制位

1：工作，0：停止，软件设置

0 溢出标志位

查询方式时软件清0，中断方式时硬件清

1 运行控制位

1：工作，0：停止，软件设置

外中断1请求标志位

电平方式0，脉冲方式1  
外中断1方式控制位

外中断0请求标志位

电平方式0，脉冲方式1  
外中断0方式控制位



# TMOD (工作方式控制寄存器)

位 序	B <sub>7</sub>	B <sub>6</sub>	B <sub>5</sub>	B <sub>4</sub>	B <sub>3</sub>	B <sub>2</sub>	B <sub>1</sub>	B <sub>0</sub>
位 符 号	GATE	C/ $\bar{T}$	M <sub>1</sub>	M <sub>0</sub>	GATE	C/ $\bar{T}$	M <sub>1</sub>	M <sub>0</sub>

定时/计数器 1

定时/计数器 0

**1** : 门控位。**0** : 以外中断启动。

定时/计数选择位。**0** : 定时。**1** : 计数

**10**、**11** 方式 **0** ~ **3** 对应于 **00**、**01**、

# IE（中断允许控制寄存器）

位地址	0AFH	0AEH	0ADH	0ACH	0ABH	0AAH	0A9H	0A8H
位符号	EA	/	/	ES	ET <sub>1</sub>	EX <sub>1</sub>	ET <sub>0</sub>	EX <sub>0</sub>

总允许1，总禁止0  
中断允许总控制位

串行中断允许位  
0 允许1，禁止0  
1 中断允许

外中断允许位  
0 允许1，禁止0  
1 中断允许

外中断允许位  
0 允许1，禁止0  
1 中断允许

外中断允许位  
0 允许1，禁止0  
1 中断允许

## 定时工作方式0（13位加法计数器）

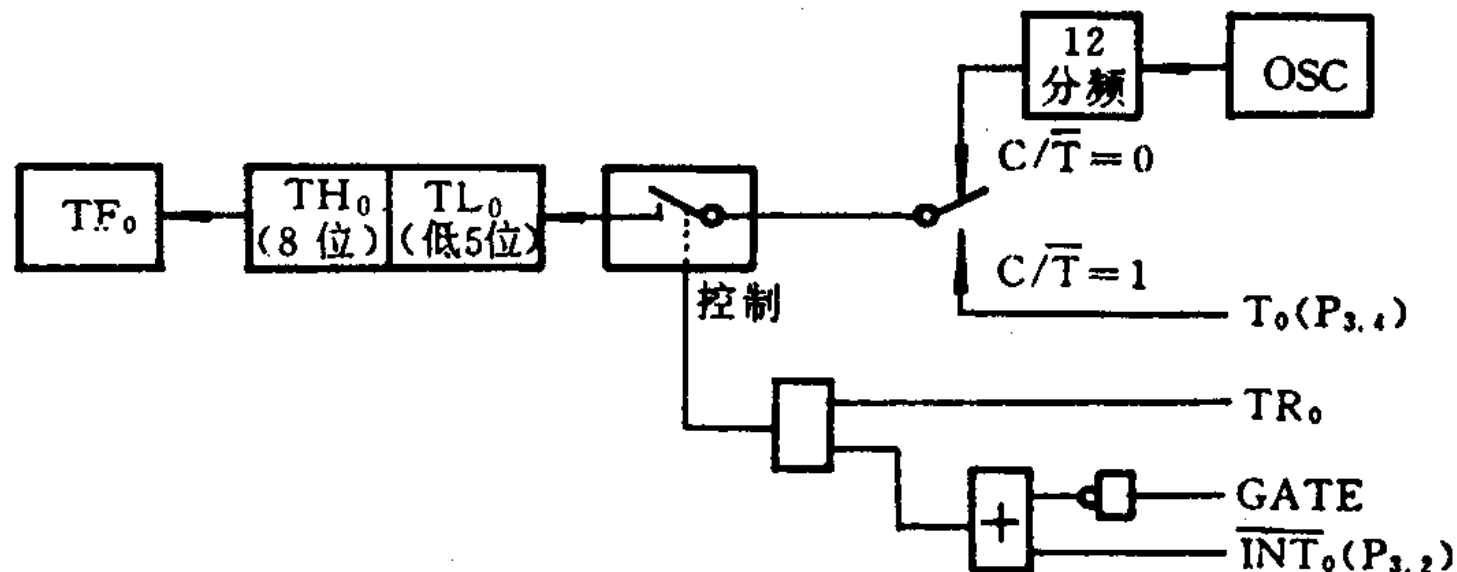


图 6.4 定时器/计数器 0 的工作方式 0 逻辑结构

计数： **$1 \sim 2^{13}$  (8192)**

定时： **$(2^{13} - \text{计数初值}) \times \text{晶振周期} \times 12$**

**[例 6.1]** 设单片机晶振频率为 6 MHz,使用定时器 1 以方式 0 产生周期为 500  $\mu$ s 的等宽正方波连续脉冲,并由 P<sub>1.0</sub> 输出。以查询方式完成。

## 程序设计

	MOV	TMOD, #00H	;设置 T <sub>1</sub> 为工作方式 0
	MOV	TH <sub>1</sub> , #0FCH	;设置计数初值
	MOV	TL <sub>1</sub> , #03H	
	MOV	IE, #00H	;禁止中断
LOOP:	SETB	TR <sub>1</sub>	;启动定时
	JBC	TF <sub>1</sub> , LOOP1	;查询计数溢出
	AJMP	LOOP	
LOOP1:	MOV	TH <sub>1</sub> , #0FCH	;重新设置计数初值
	MOV	TL <sub>1</sub> , #03H	
	CLR	TF <sub>1</sub>	;计数溢出标志位清“0”
	CPL	P <sub>1.0</sub>	;输出取反
	AJMP	LOOP	;重复循环

# 定时工作方式1（16位加法计数器）

结构同方式0

计数： $1 \sim 2^{16}$ （65536）

定时： $(2^{16} - \text{计数初值}) \times \text{晶振周期} \times 12$

【例 6.2】 题目同【例 6.1】，但以中断方式完成。即单片机晶振频率为 6 MHz，使用定时器 1 以工作方式 1 产生周期为 500  $\mu\text{s}$  的等宽连续正方波脉冲，并在 P<sub>1.0</sub> 端输出。

1. 计算计数初值

$$\text{TH}_1 = 0\text{FFH} \quad \text{TL}_1 = 83\text{H}$$

2. TMOD 寄存器初始化

$$\text{TMOD} = 10\text{H}$$

3. 程序设计

### 3. 程序设计

#### 主程序:

MOV	TMOD, #10H	;定时器 1 工作方式 1
MOV	TH <sub>1</sub> , #0FFH	;设置计数初值
MOV	TL <sub>1</sub> , #83H	
SETB	EA	;开中断
SETB	ET <sub>1</sub>	;定时器 1 允许中断
LOOP: SETB	TR <sub>1</sub>	;定时开始
HERE: SJMP	\$	;等待中断

#### 中断服务程序:

MOV	TH <sub>1</sub> , #0FFH	;重新设置计数初值
MOV	TL <sub>1</sub> , #83H	
CPL	P <sub>1.0</sub>	;输出取反
RETI		;中断返回

## 定时工作方式2：8位自动加载的加法计数器

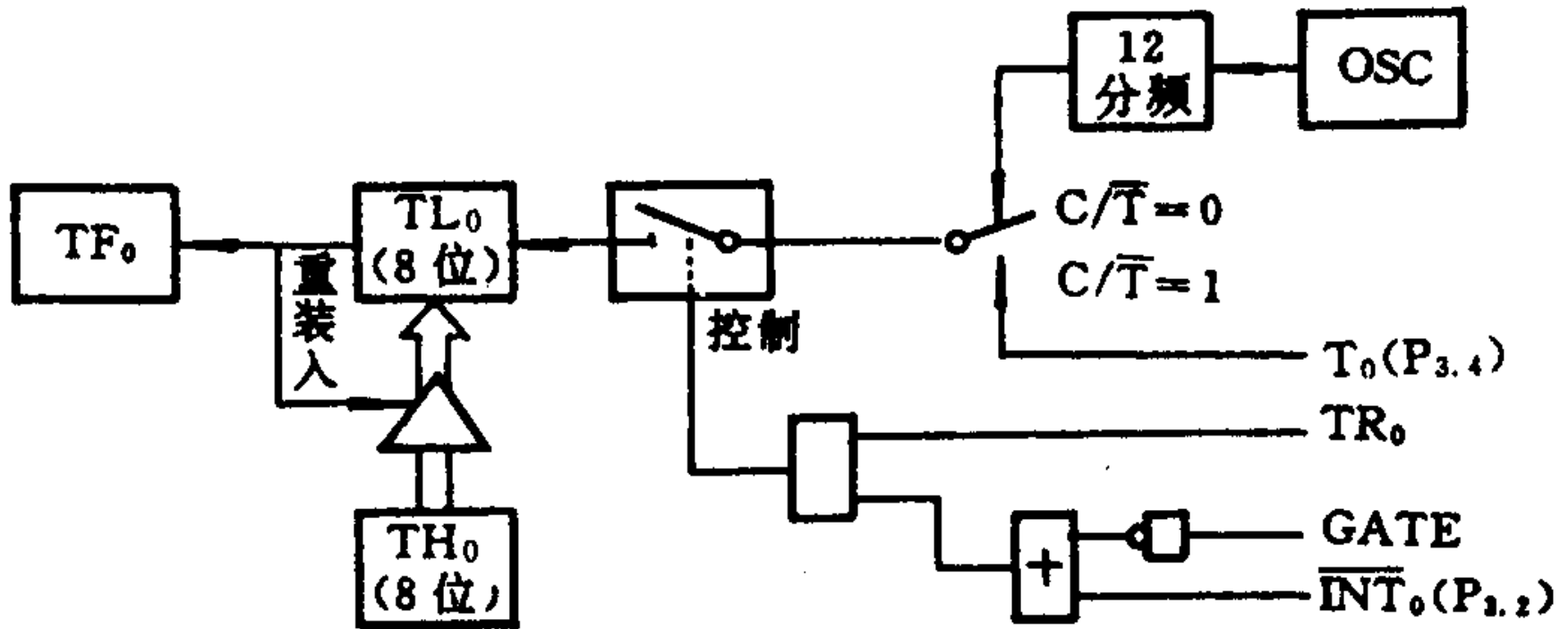



图 6.5 定时器/计数器 0 的工作方式 2 逻辑结构

可以作为串行通信的波特率发生器用。



[例 6.3] 使用定时器 0 以工作方式 2 产生  $100\ \mu\text{s}$  定时, 在  $P_{1.0}$  输出周期为  $200\ \mu\text{s}$  的连续正方波脉冲。已知晶振频率  $f_{\text{osc}}=6\ \text{MHz}$ 。

### 程序设计(查询方式)

MOV	IE, #00H	;禁止中断
MOV	TMOD, #02H	;设置定时器 0 为工作方式 2
MOV	TH <sub>0</sub> , #0CEH	;保存计数初值
MOV	TL <sub>0</sub> , #0CEH	;设置计数初值
SETB	TR <sub>0</sub>	;启动定时
LOOP: JBC	TF <sub>0</sub> , LOOP1	;查询计数溢出
AJMP	LOOP	
LOOP1: CPL	P <sub>1.0</sub>	;输出方波
AJMP	LOOP	;重复循环



## 程序设计(中断方式)

### 主程序:

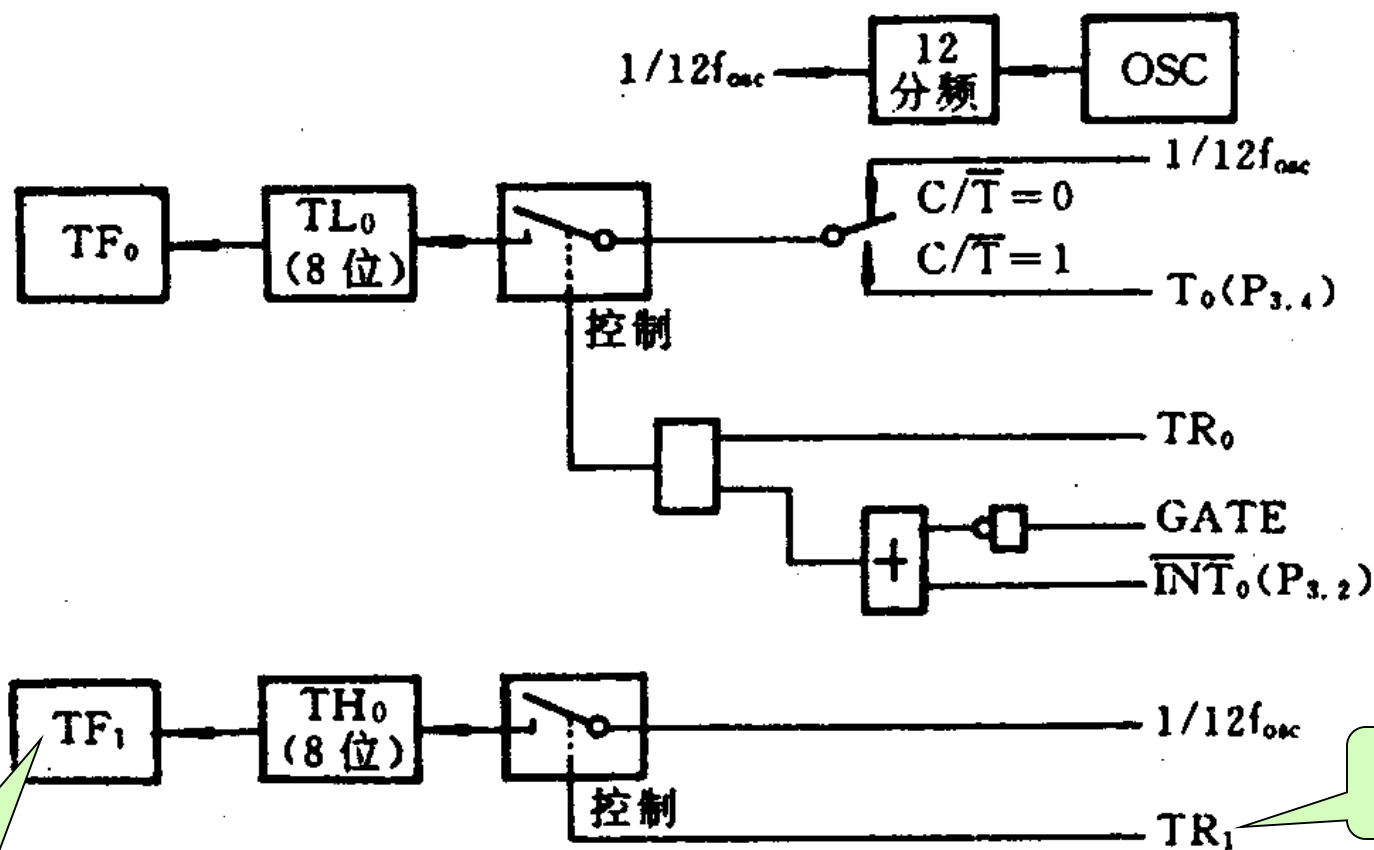
MOV	TMOD, #02H	;设置定时器0为工作方式2
MOV	TH <sub>0</sub> , #0CEH	;保存计数初值
MOV	TL <sub>0</sub> , #0CEH	;设置计数初值
SETB	EA	;开中断
SETB	ET <sub>0</sub>	;定时器0允许中断
LOOP: SETB	TR <sub>0</sub>	;开始定时
HERE: SJMP	\$	;等待中断

### 中断服务中断:

CPL	P <sub>1.0</sub>	;输出方波
RETI		;中断返回

# 定时工作方式3

**T0**：拆成**2**个独立的**8**位计数器**TL0**（可以计数或定时）和**TH0**（只能定时，借用**T1**的**TR1**和**TF1**）。

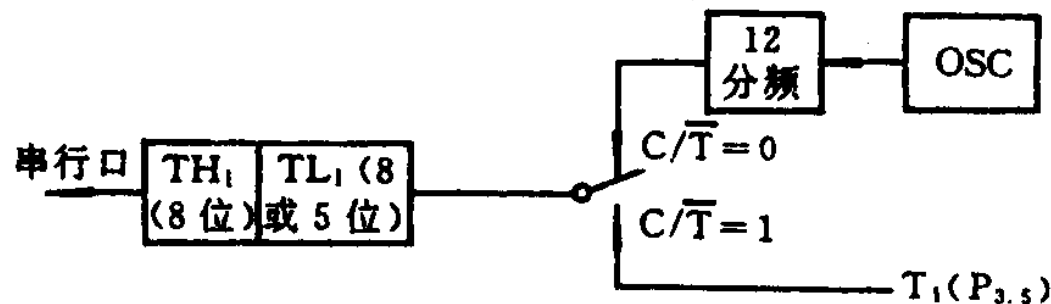


借用

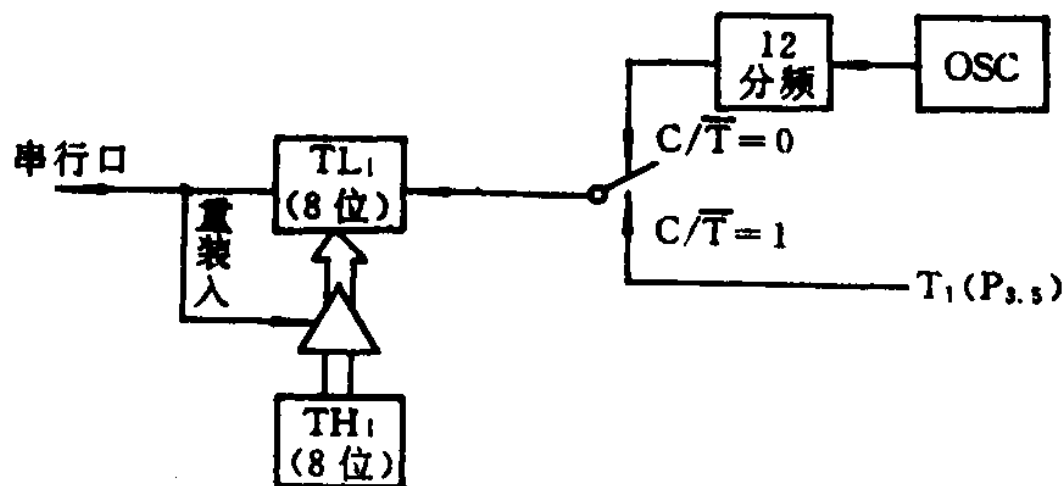
借用

图 6.6 定时器/计数器 0 工作方式 3 逻辑结构

**T1: 只能工作在方式0、1、2，计数溢出送给串行口。这时没有 $TR_1$ 和 $TF_1$ 。**



(a) T1 方式 1(或 0)



(b) T1 方式 2

# 定时/计数器与中断的应用举例

设计一个以时分秒为单位的计时程序

## 基本方法

### 1. 计数初值计算:

1秒 = 8 × 125毫秒,

$$(2^{16}-X) \times 2 = 125 \times 10^3, X = 3036D = 0CDCH$$

软件计数实现    定时方式1实现

设晶振为  
6MHz

2. 采样中断方式, 每次溢出中断计数1次, 8次为1秒;

3. 秒→分, 分→时由软件累加、数值比较实现;

4. 设置显示缓冲区: 时分秒共设6个单元缓冲区:

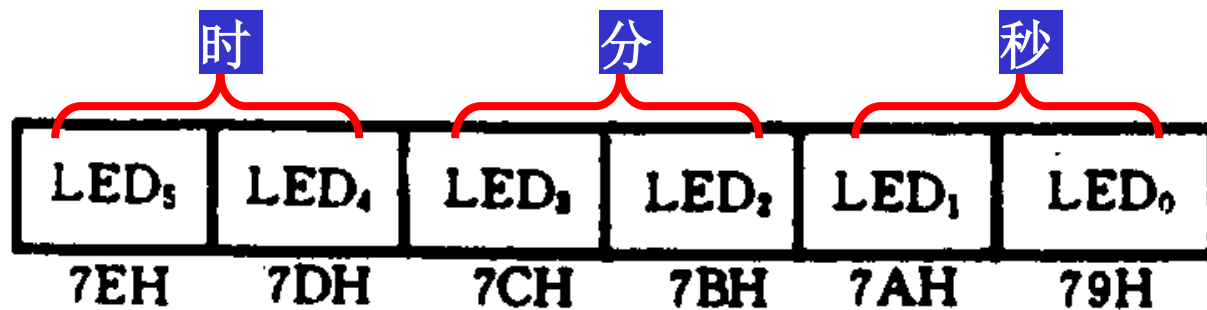


图 6.10 LED 显示器的缓冲单元

# 主程序流程

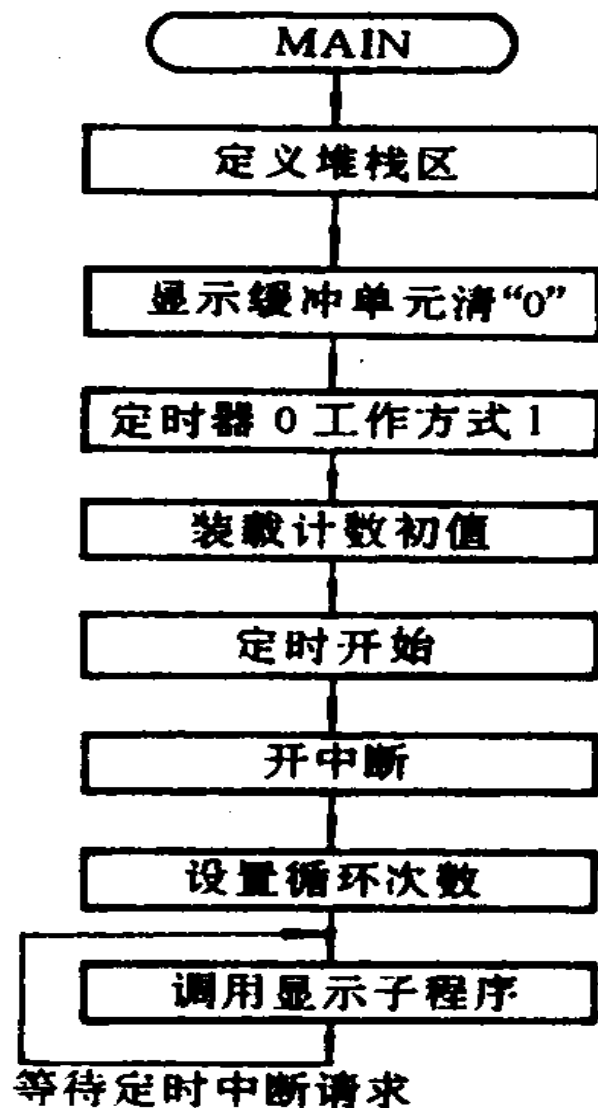


图 6.11 时钟计时主程序流程

## 中断服务程序流程

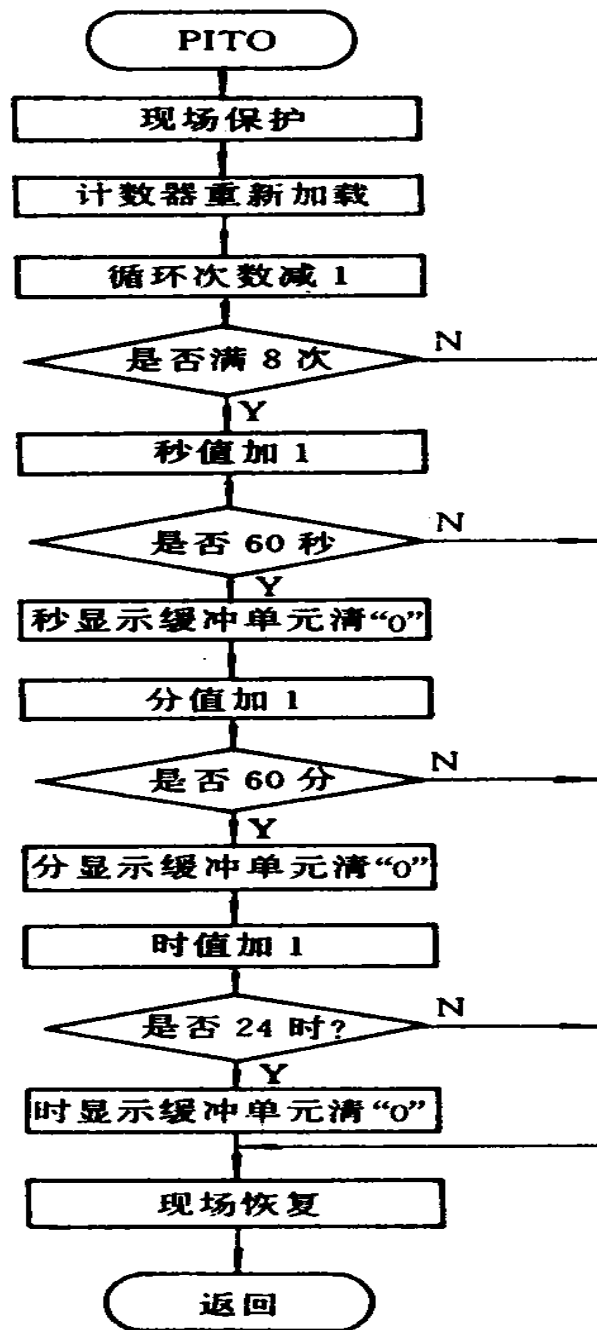


图 6.12 中断服务程序流程

## 加1子程序流程

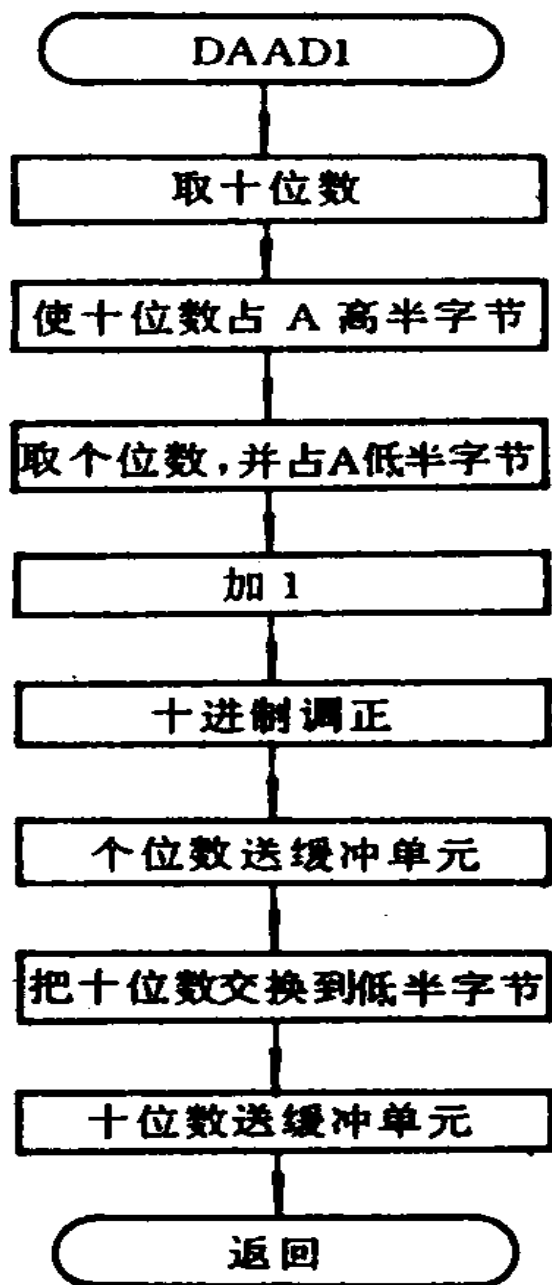


图 6.13 加 1 子程序流程

```

        ORG      8000H
START: AJMP     MAIN

```

```

        ORG      800BH
        AJMP     PITO

```

```

        ORG      8100H
MAIN:   MOV      SP, #60H           ;确立堆栈区
        MOV      R0, #79H         ;显示缓冲区首地址
        MOV      R7, #06H         ;显示位数
ML1:    MOV      @R0, #00H         ;显示缓冲单元清“0”
        INC      R0
        DJNZ     R7, ML1
        MOV      TMOD, #01H       ;设置定时器 0 为工作方式 1
        MOV      TL0, #0DCH       ;装计数器初值
        MOV      TH0, #0CH
        SETB     8CH              ;TR0 置“1”,定时开始
        SETB     AFH              ;EA 置“1”,中断总允许
        SETB     A9H              ;ET0 置“1”,定时器 0 中断允许
        MOV      30H, #08H        ;要求的计数溢出次数,即循环次数
ML0:    LCALL    SMXS              ;调用显示子程序
        SJMP     ML0
PITO:   PUSH     PSW               ;中断服务程序,现场保护
        PUSH     ACC
        SETB     PSW.3             ;RS1 RS0 = 01,选 1 组通用寄存器
        MOV      TL0, #0DCH       ;计数器重新加载
        MOV      TH0, #0CH
        MOV      A, 30H            ;循环次数减 1

```

## 程序清单



MOV	A, 30H	;循环次数减 1
DEC	A	
MOV	30H, A	
JNZ	RET0	;不满 8 次,转 RET0 返回
MOV	30H, #08H	;满 8 次,开始计时操作
MOV	R <sub>0</sub> , #7AH	;秒显示缓冲单元地址
ACALL	DAAD1	;秒加 1
MOV	A, R <sub>2</sub>	;加 1 后秒值在 R <sub>2</sub> 中
XRL	A, #60H	;判是否到 60 s
JNZ	RET0	;不到,转 RET0 返回
ACALL	CLR0	;到 60 s 显示缓冲单元清“0”
MOV	R <sub>0</sub> , #7CH	;分显示缓冲单元地址
ACALL	DAAD1	;分加 1
MOV	A, R <sub>2</sub>	
XRL	A, #60H	;判是否到 60 in
JNZ	RET0	
ACALL	CLR0	;到 60 in 显示缓冲单元清“0”
MOV	R <sub>0</sub> , #7EH	;时显示缓冲单元地址
ACALL	DAAD1	;时加 1
MOV	A, R <sub>2</sub>	
XRL	A, #24H	;判是否到 24 h
JNZ	RET0	
ACALL	CLR0	;到 24 h,时显示缓冲单元清“0”

RET0:	POP	ACC	;现场恢复
	POP	PSW	
	RETI		;中断返回
	DAAD1: MOV	A, @R <sub>0</sub>	;加1子程序,十位数送 A
	DEC	R <sub>0</sub>	
	SWAP	A	;十位数占高4位
	ORL	A, @R <sub>0</sub>	;个位数占低4位
	ADD	A, #01H	;加1
	DA	A	;十进制调整
	MOV	R <sub>2</sub> , A	;全值暂存 R <sub>2</sub> 中
CLR0:	ANL	A, #0FH	;屏蔽十位数,取出个位数
	MOV	@R <sub>0</sub> , A	;个位值送显示缓冲单元
	MOV	A, R <sub>2</sub>	
	INC	R <sub>0</sub>	
	ANL	A, #0F0H	;屏蔽个位数取出十位数
	SWAP	A	;使十位数占低4位
	MOV	@R <sub>0</sub> , A	;十位数送显示缓冲单元
	RET		;返回
	CLR	A	;清缓冲单元子程序
	MOV	@R <sub>0</sub> , A	;十位数缓冲单元清“0”
	DEC	R <sub>0</sub>	
	MOV	@R <sub>0</sub> , A	;个位数缓冲单元清“0”
	RET		;返回