201ASP
中档系列 PIC
外设配置和汇编编程

Microchip Technology Inc.
RTC
Regional Training Centers

Microchip Technology Inc.
RTC
Regional Training Centers

目标

- 学习完本课程后，将能够：
  - 理解基本PIC外设及相关的寄存器
  - 具有初始化中档外设的*“实践”*经验
  - 能实现本课程中未涉及到的外设
  - 理解中断和查询
  - 从零开始编写自己的应用程序代码

要从本课程获得最大收益

- *理想情况* 应熟悉以下内容：
  - 汇编器编程
  - 基本中档系列指令集
  - 数据和程序存储器构成
  - MPLAB集成开发环境
  - Microchip ICD2调试器

201ASP日程安排

- *简要地*回顾中档架构、指令集和MCHP工具
- 中档PIC器件中的中断
  - 基本中断实验
- 外设讨论：
  - 输入/输出端口
  - 定时器
    - Timer0
    - Timer1
      - Timer1实验
    - Timer2
      - Timer2实验

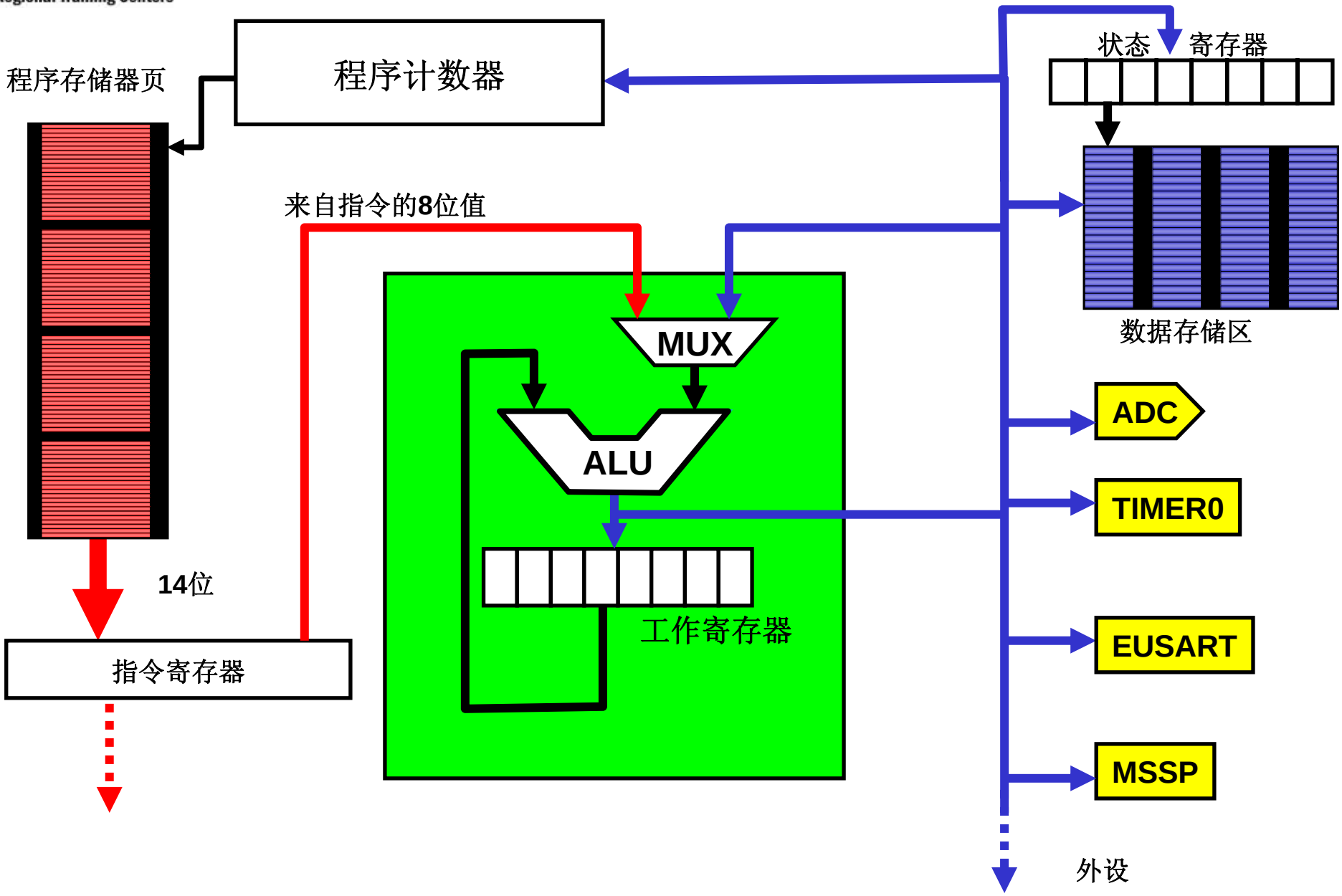
201ASP日程安排

- 增强型捕捉 /比较/ PWM 模块（ECCP）
  - PWM 和输出比较实验
- 模拟比较器
- 模数转换器（ADC）
  - ADC实验
- 增强型通用异步/同步接收器/发送器（EUSART）
- I^2C和SPI （主同步串行端口）
  - 多重中断实验
- 总结及其他问题

中档系列
基本架构和
开发工具

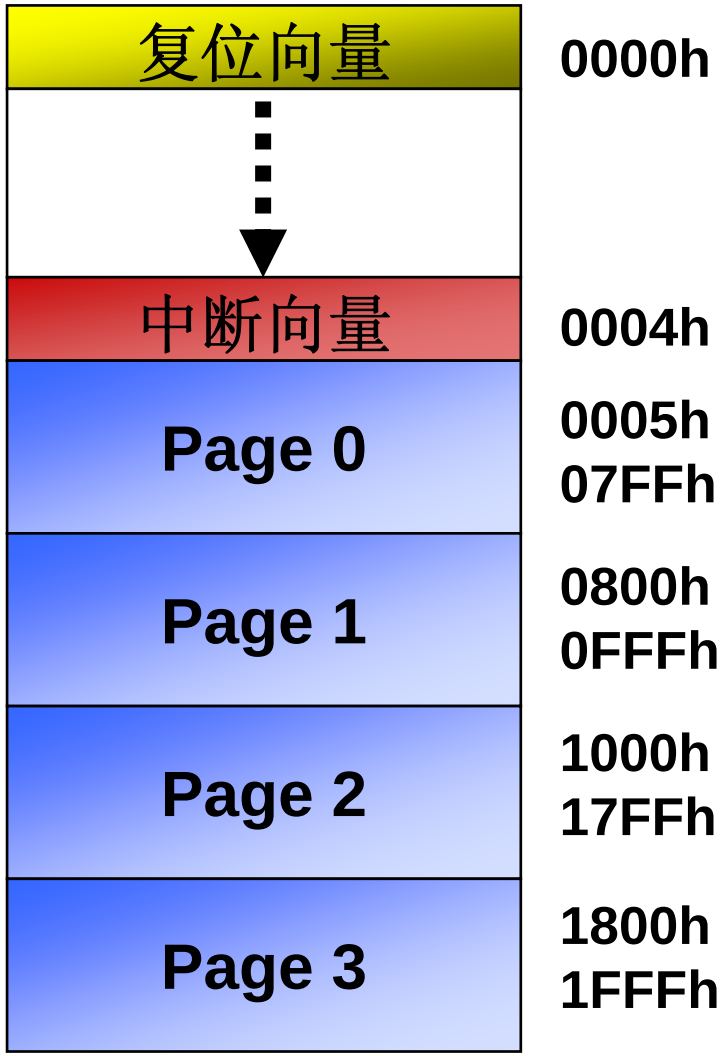
Microchip Technology Inc.
RTC
Regional Training Centers

中档PIC框图



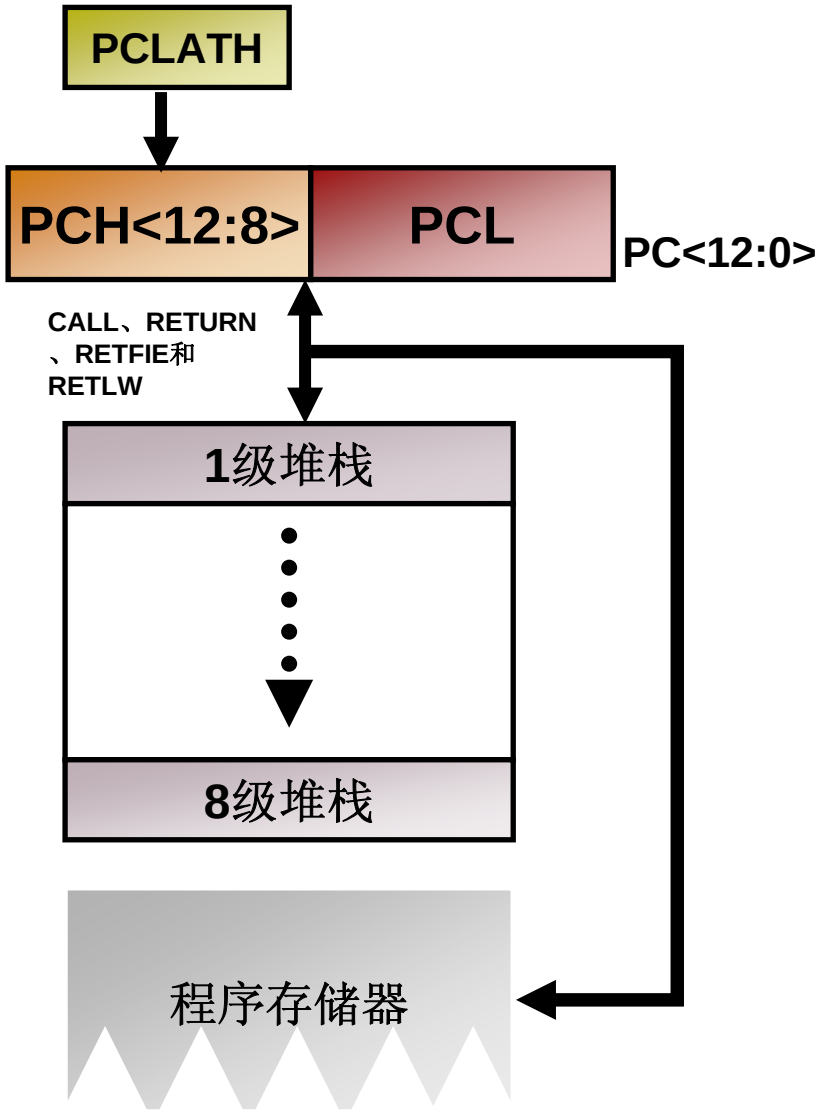
程序存储器

- 最大8K字：
  - (8K x 14位/字)/1字节 = 14 KB的存储空间
- 复位向量位于0000h单元
  - 发生任何复位时，PC 将跳转到此地址
- 中断向量位于0004h单元
  - 发生任何中断时，PC 将跳转到此地址

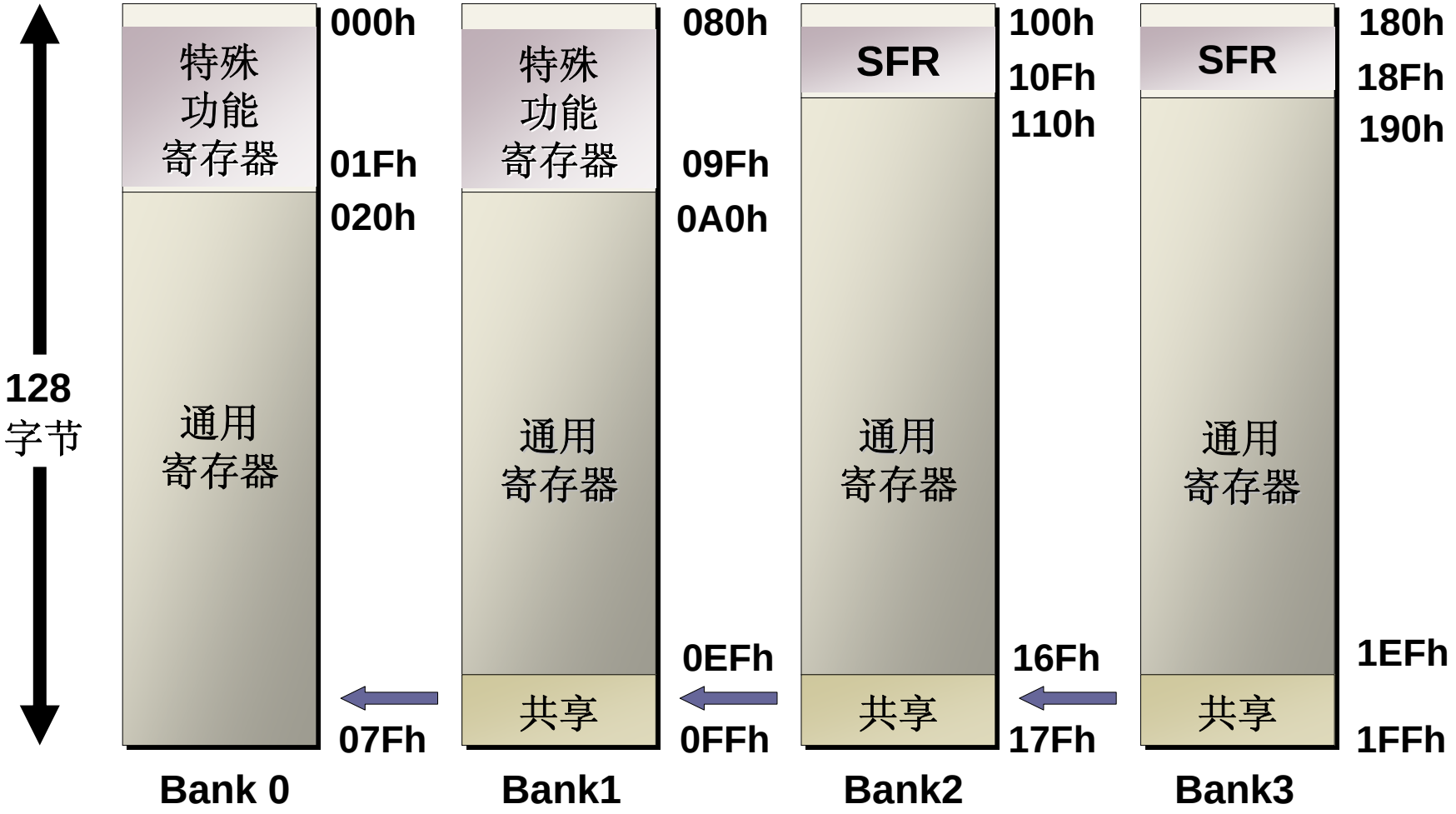


程序计数器（PC）和堆栈

- 13位PC
  - PCL→ ALU结果（8位）或 操作码（11位）
  - PCH→ 页选择位
    - 通过PCLATH更新
    - 指定程序存储器中的页
- 8 级深堆栈
  - 存储PC的内容
    - PUSH
      - 调用/中断
    - POP
      - RETURN、RETFIE和RETLW

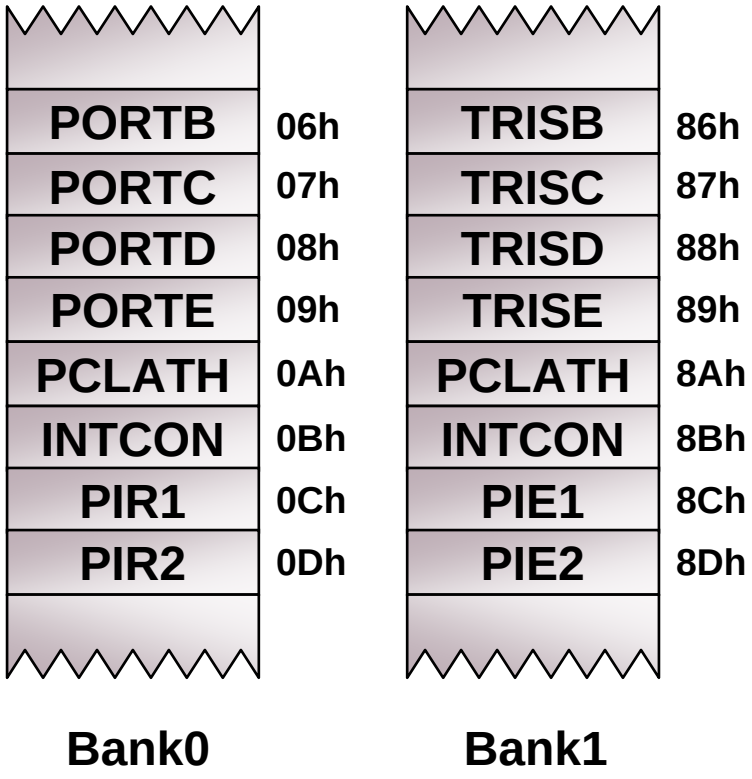


数据存储器映射

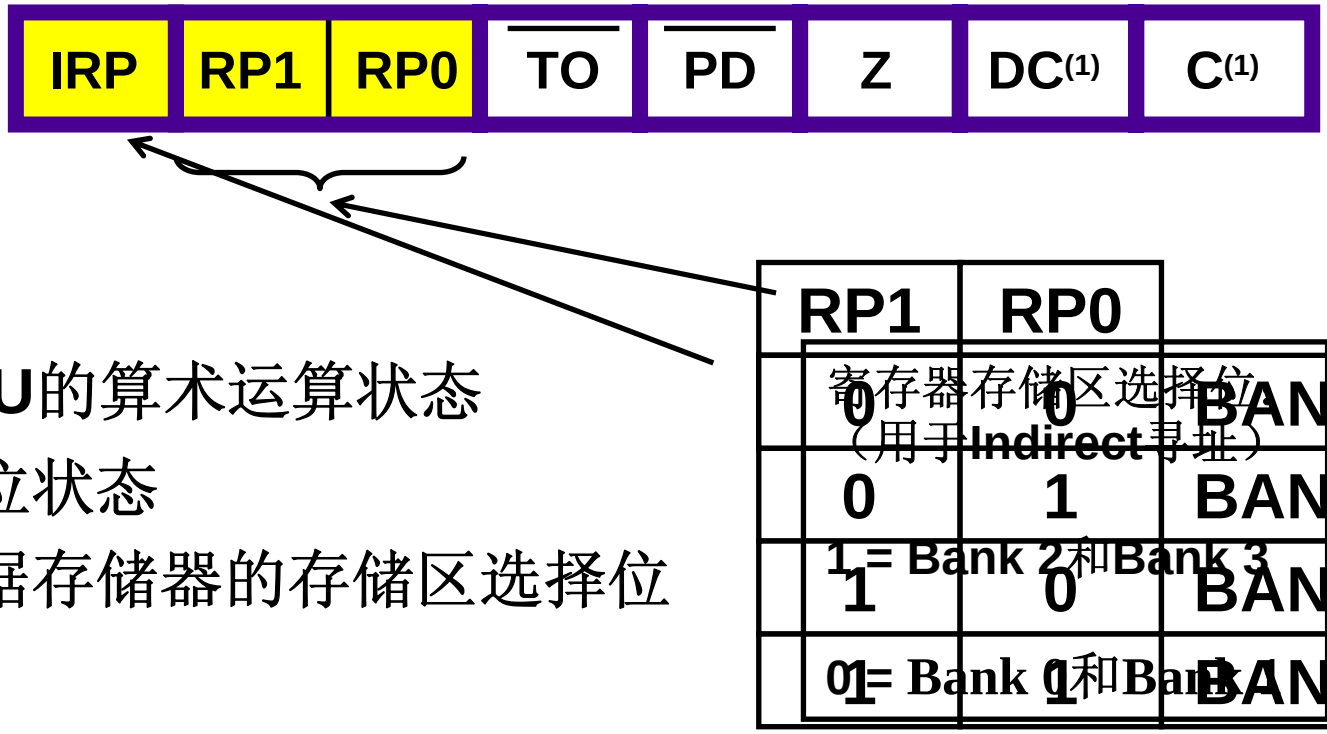


特殊功能寄存器（SFR）

- 寄存器文件概念
- 访问方式与访问其他寄存器一样
- 某些寄存器可跨越所有存储区（**PCLATH**和**INTCON**等）



状态寄存器



- 包含：
  - ALU的算术运算状态
  - 复位状态
  - 数据存储器的存储区选择位

Microchip Technology Inc.
RTC
Regional Training Centers

指令集概述

- **35** 条单字指令
- 除程序转移指令外，所有指令都是单周期指令
- **3**类操作：

字节操作类指令

ADDWF	f, d	Add W and f
ANDWF	f, d	AND W with f
CLRF	f	Clear f
CLRW	–	Clear W
COMF	f, d	Complement f
DECF	f, d	Decrement f
DECFSZ	f, d	Decrement f, Skip if 0
INCF	f, d	Increment f
INCFSZ	f, d	Increment f, Skip if 0
IORWF	f, d	Inclusive OR W with f
MOVF	f, d	Move f
MOVWF	f	Move W to f
NOP	–	No Operation
RLF	f, d	Rotate Left f through Carry
RRF	f, d	Rotate Right f through Carry
SUBWF	f, d	Subtract W from f
SWAPF	f, d	Swap nibbles in f
XORWF	f, d	Exclusive OR W with f

位操作类指令

BCF	f, b	Bit Clear f
BSF	f, b	Bit Set f
BTFSC	f, b	Bit Test f, Skip if Clear
BTFSS	f, b	Bit Test f, Skip if Set

立即数和控制操作类指令

ADDLW	k	Add literal and w
ANDLW	k	AND literal with w
CALL	k	Call Subroutine
CLRWDT	–	Clear Watchdog Timer
GOTO	k	Go to address
IORLW	k	Inclusive OR literal with w
MOVLW	k	Move literal to w
RETFIE	–	Return from interrupt
RETLW	k	Return with literal in w
RETURN	–	Return from Subroutine
SLEEP	–	Go into Standby mode
SUBLW	k	Subtract w from literal
XORLW	k	Exclusive OR literal with w

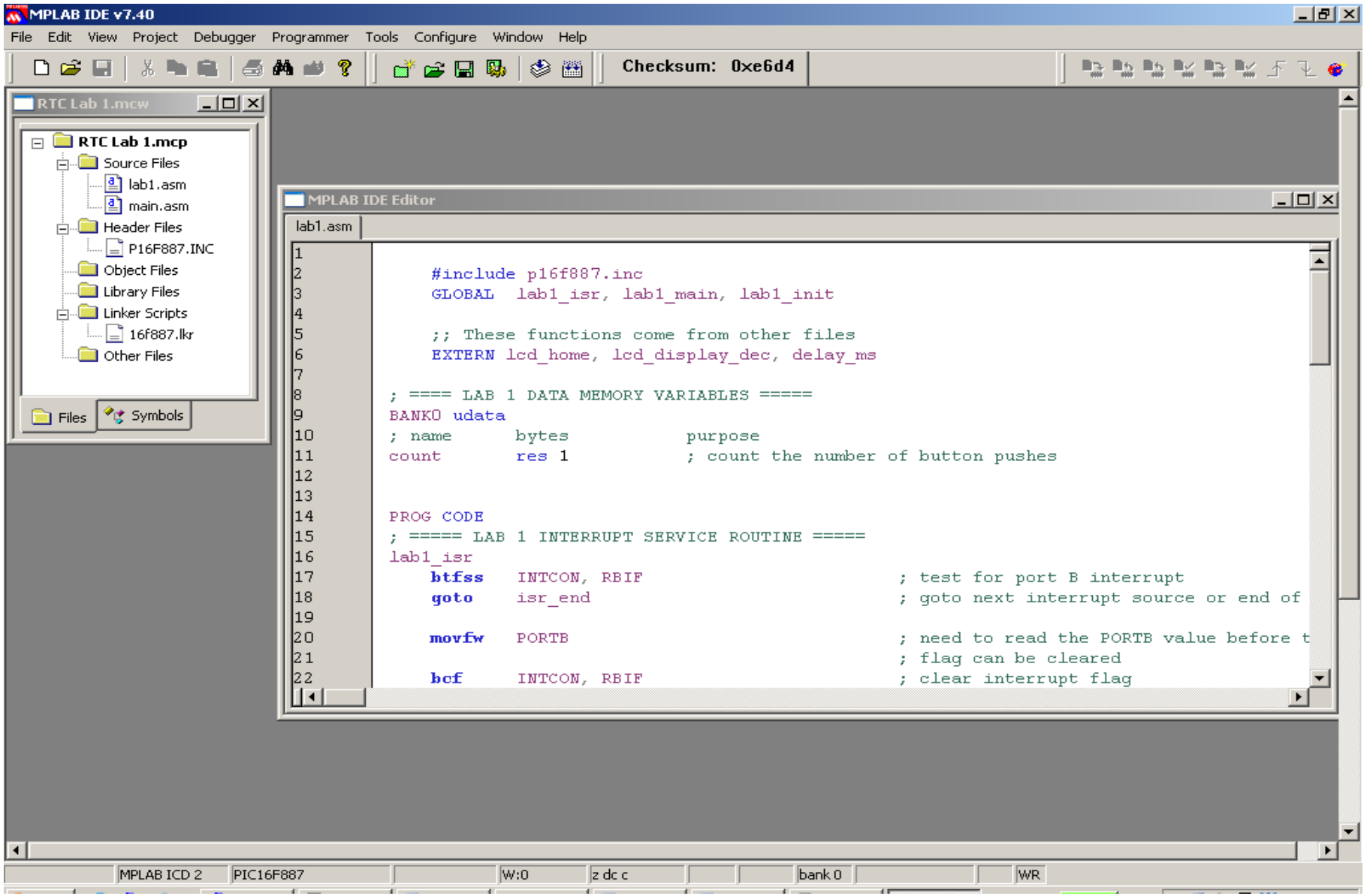
全部在一张幻灯片中!!!

PIC开发工具

Microchip Technology Inc.
RTC
Regional Training Centers

Microchip Technology Inc.
RTC
Regional Training Centers

PIC®：开发工具MPLAB® IDE



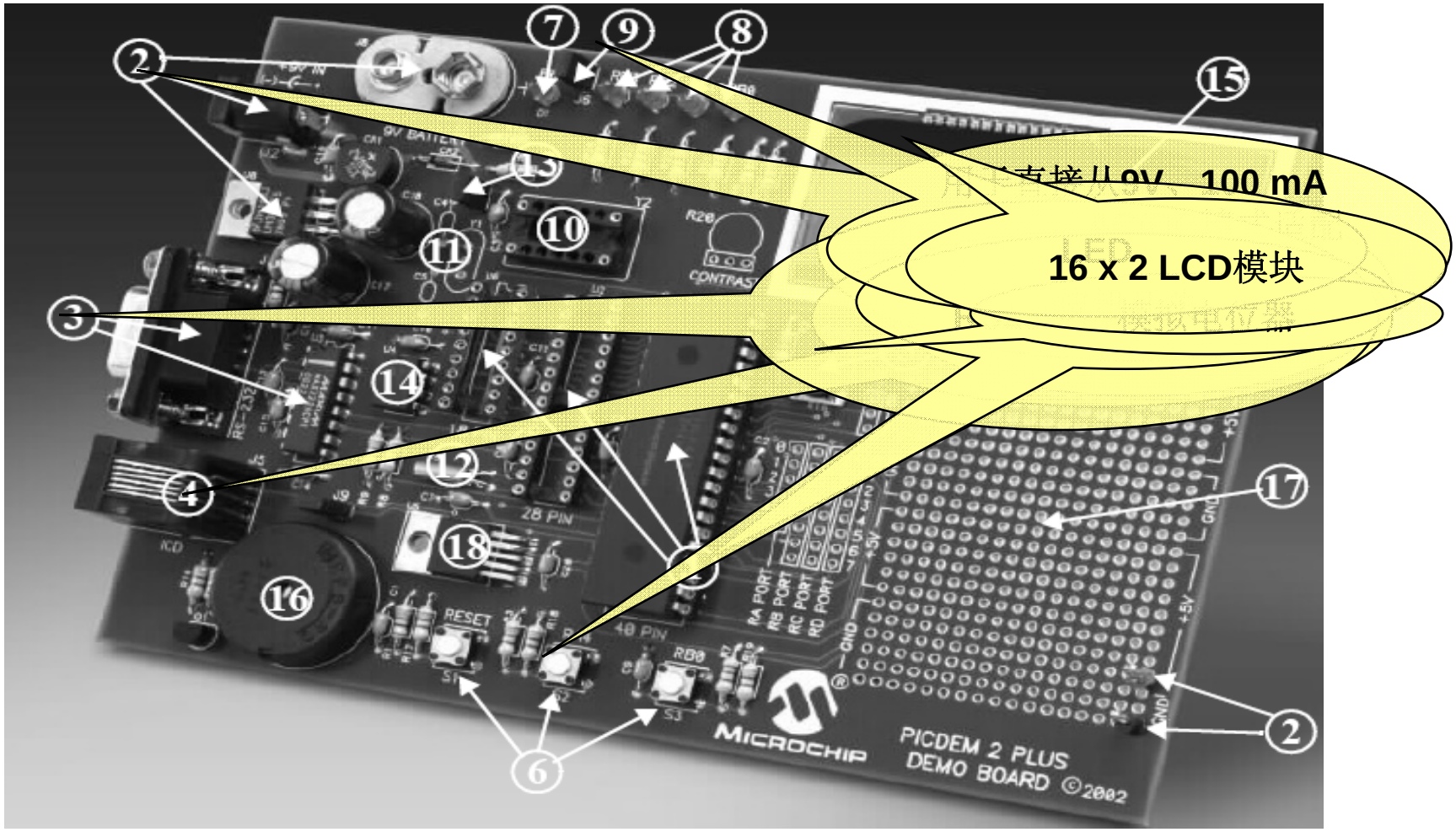
ICD 2 （在线调试器）

- **MPLAB® ICD 2** 是一款低成本的实时调试器和编程器。提供以下功能：
  - 实时后台调试
  - 读/写目标单片机的存储空间和**EEDATA**区域
  - 编程配置位
  - 擦除带校验的程序存储空间

Microchip Technology Inc.
RTC
Regional Training Centers

PICDEM® 2 Plus板

PICDEM® 2 Plus 板硬件详细信息



中断

Microchip Technology Inc.
RTC
Regional Training Centers

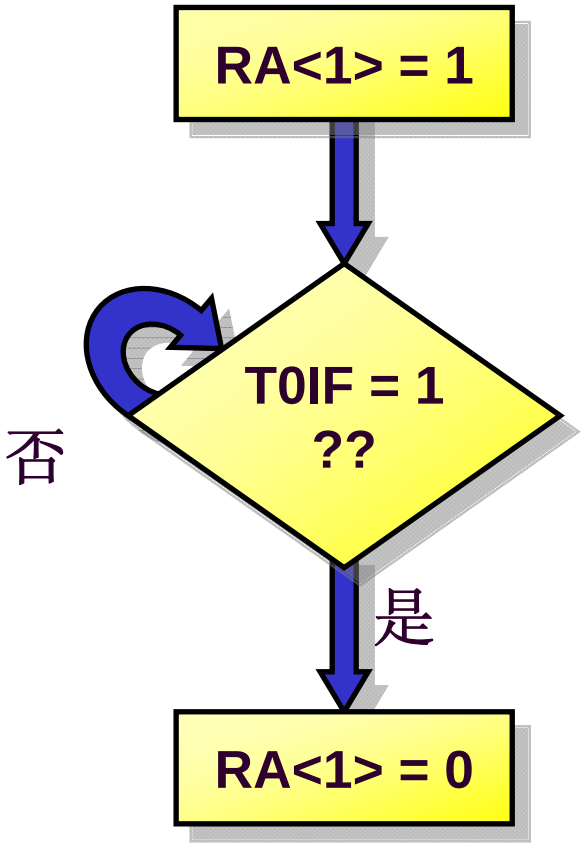
Microchip Technology Inc.
RTC
Regional Training Centers

查询和中断

- 发生特殊事件时，我们总是希望处理器能执行任务
- 有两种方法用于检查事件是否发生：
  - 查询：
    - 在代码执行的各个时间点连续对事件进行检查
  - 中断：
    - 仅在事件发生时“中断”正常的代码执行

查询

```
bsf     PORTA,1         ;Set bit 1 of
                        ;PORTA

btfss   INTCON, T0IF    ;Check Timer0
                        ;interrupt flag in
                        ;INTCON and
                        ;skip the next
                        ;instruction if it
                        ;is set

goto    $-1             ;Go back to
                        ;previous
                        ;instruction

bcf     PORTA,1         ;Clear bit 0 of
                        ;PORTA
```

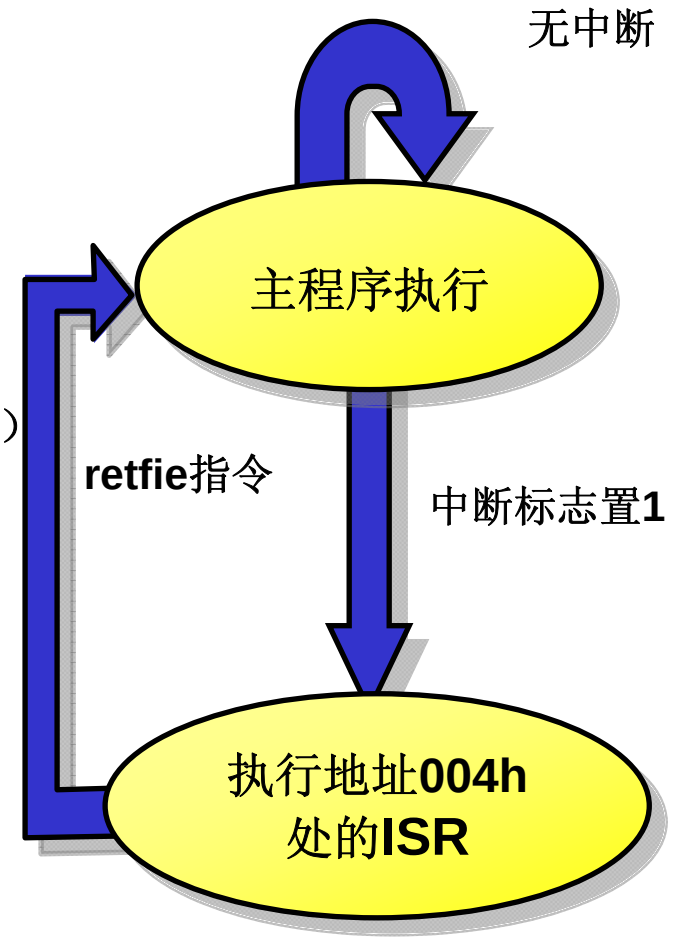


中断

```
Reset           code    000h
                goto    Start

;=========================
int_vector      code    004h
                .
                .
                .
retfie          ;return from
                ;interrupt
;=========================

main_prog       code

Start   ;start label for main code
                .
                .
                .
end
```

中断服务程序（ISR）

主程序代码



Microchip Technology Inc.
RTC
Regional Training Centers

允许中断

- 必须通知服务器代码中将使用中断
  - 所使用的带有允许位的寄存器：
    - 中断控制寄存器（**INTCON**）
    - 外设中断允许寄存器**1** （**PIE1**）
    - 外设中断允许寄存器**2** （**PIE2**）

INTCON （内核中断）

允许位

GIE：全局中断允许位

*必须置1以使用PIC器件中的任何中断

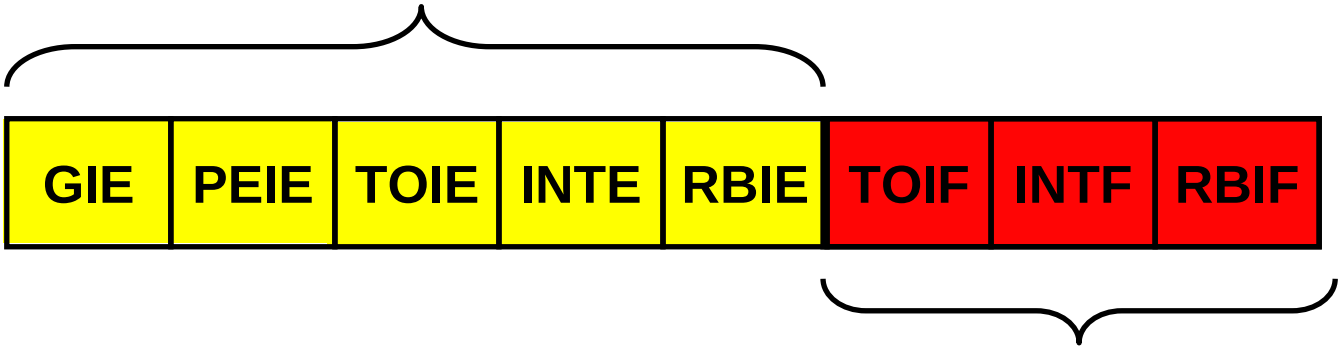
PEIE：外设中断允许位

*必须置1以使用外设中断

TOIE：Timer0溢出中断允许位

INTE：RB0/INT外部中断允许位

RBIE：PORTB电平变化中断允许位



*即使未允许中断，
也应置1的标志位！

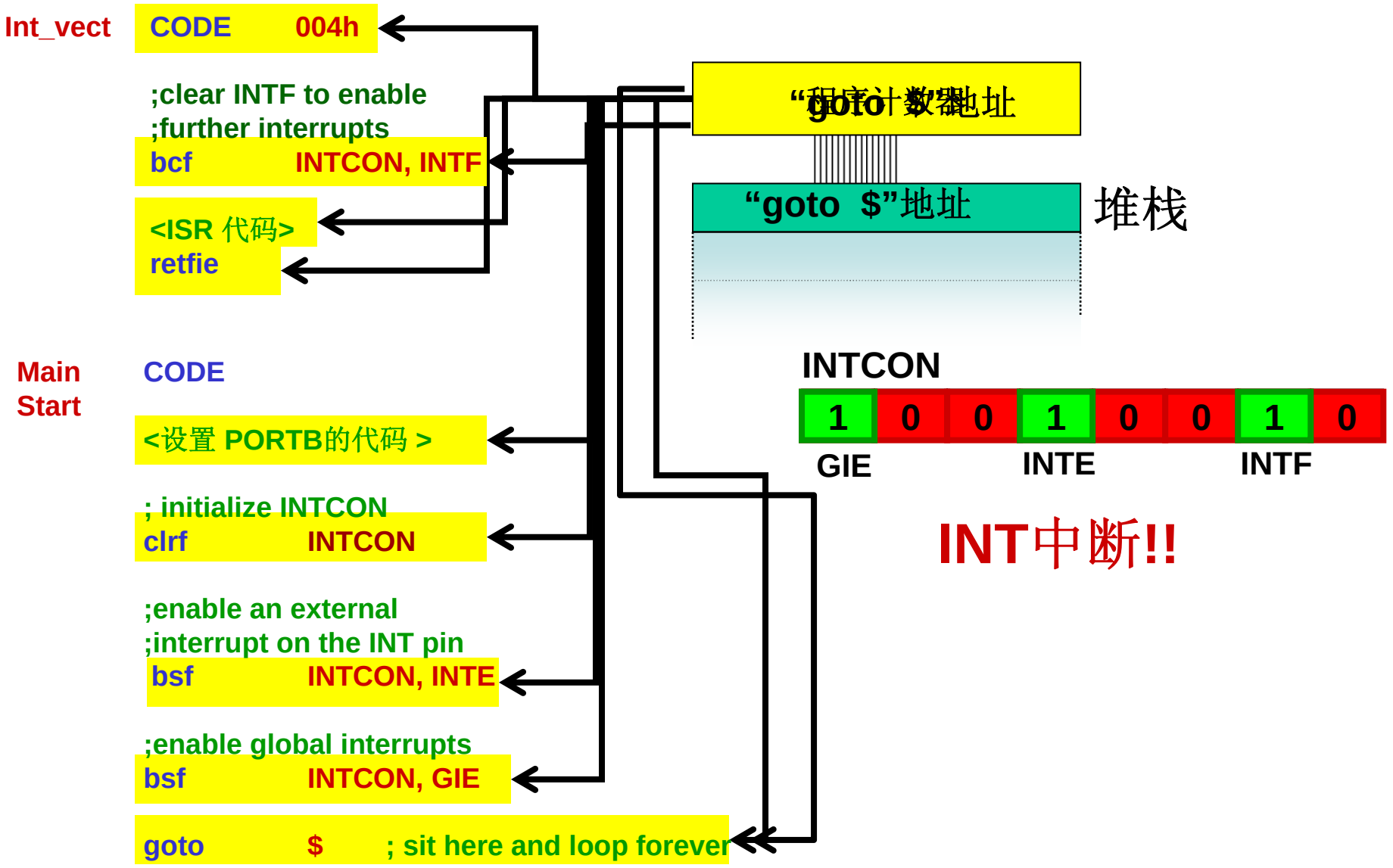
标志位

TOIF：Timer0溢出中断标志位

INTF：RB0/INT外部中断标志位

RBIF：PORTB电平变化中断标志位

允许内核中断



Microchip Technology Inc.
RTC
Regional Training Centers

外设中断

- 两个允许特定外设中断的寄存器：
  - 外设中断允许寄存器1 （PIE1）
  - 外设中断允许寄存器2 （PIE2）
- 两个包含特定外设中断标志位的寄存器：
  - 外设中断请求寄存器1 （PIR1）
    - PIE1中允许的外设中断的标志位
  - 外设中断请求寄存器2 （PIR2）
    - PIE2中允许的外设中断的标志位

*即使未允许中断，
也应置1标志位

Microchip Technology Inc.
RTC
Regional Training Centers

PIE1和PIR1寄存器*

PIE1寄存器（中断允许位）

	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE

PIR1寄存器（中断标志位）

	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF

允许位	标志位	条件
ADIE	ADIF	ADC转换完成
RCIE	RCIF	EUSART接收缓冲器满
TXIE	TXIF	EUSART发送缓冲器满
SSPIE	SSPIF	I^2C或SPI中断
CCP1IE	CCP1IF	Timer1寄存器捕捉或比较匹配
TMR2IE	TMR2IF	Timer2值和PR2周期值匹配
TMR1IE	TMR1IF	Timer1寄存器溢出

*检查各数据手册

PIE2和PIR2寄存器

PIE2寄存器（中断允许位）

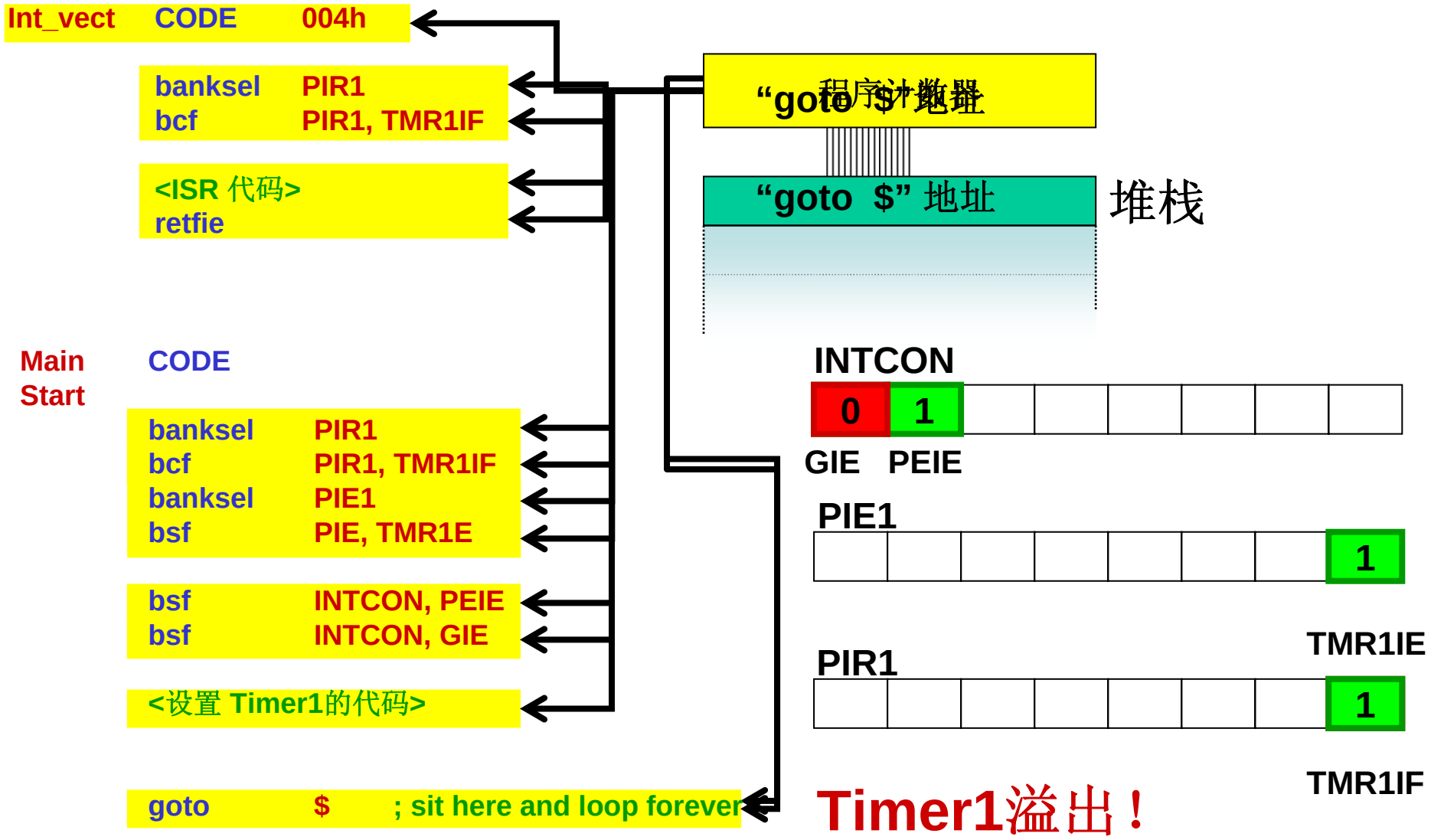
OSCFIE	C2IE	C1IE	EEIE	BCLIE	ULPWUIE		CCP2IE

PIR寄存器（中断标志位）

OSCFIF	C2IF	C1IF	EEIF	BCLIF	ULPWUIF		CCP2IF

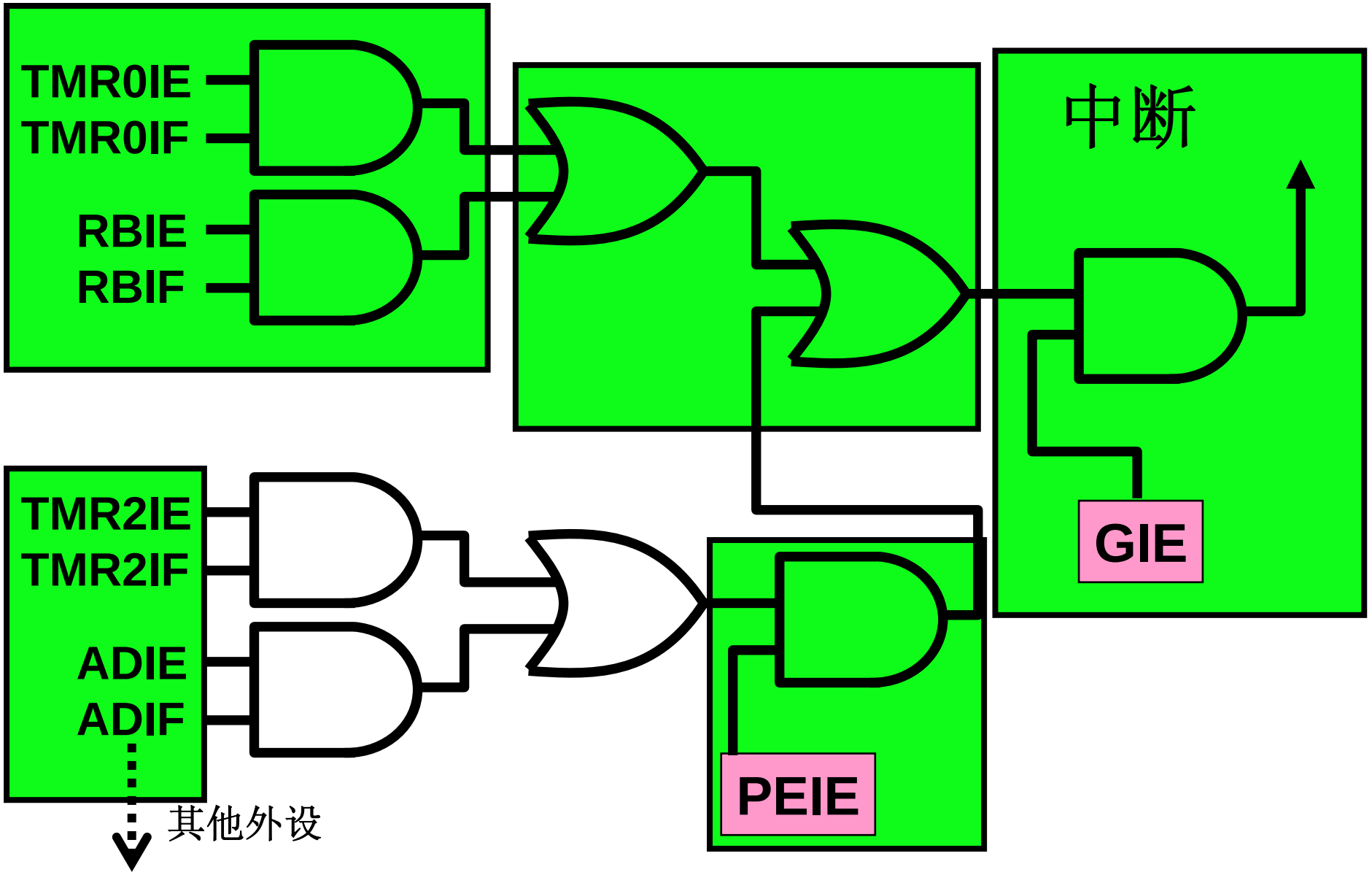
允许位	标志位	条件
OSCFIE	OSCFIF	系统振荡器发生故障
C2IE	C2IF	比较器 2输出发生变化
C1IE	C1IF	比较器 1输出发生变化
EEIE	EEIF	写操作已完成
BCLIE	BCLIF	在MSSP I^2C模式下发生了总线冲突
ULPWUIE	ULPWUIF	出现了ultralowpowerwakup唤醒条件
CCP2IE	CCP2IF	发生了timer1捕捉或比较匹配

允许外设中断



Microchip Technology Inc.
RTC
Regional Training Centers

中断逻辑



Microchip Technology Inc.
RTC
Regional Training Centers

中断响应延时

- 中断响应延时：
  - 从中断事件发生到地址**0004h**处的指令开始执行之间的时间
  - 同步中断（通常为内部中断）
    - 响应延时为**3Tcy**
  - 异步中断（通常为外部中断）
    - 响应延时为**3** 到 **3.75Tcy**

Microchip Technology Inc.
RTC
Regional Training Centers

现场保护

- 在中断期间：
  - 仅保存**PC** 值（压入堆栈）
  - 在**ISR**中发生变化的寄存器将被永久地改变
- 用户想保存的关键寄存器为：
  - 工作寄存器
  - 状态寄存器
  - **PCLath**寄存器
  - 用户定义的寄存器

Microchip Technology Inc.
RTC
Regional Training Centers

中断优先级

- 中档**PIC**单片机认为所有中断具有相同的优先级

- 用户必须执行以下操作：
  - 确定中断源
  - 确定响应中断的顺序

Microchip Technology Inc.
RTC
Regional Training Centers

中断优先级示例

```
INT_VECTOR      CODE  0x004           ; interrupt vector location

                movwf       temp_w      ; save WREG
                movf        STATUS,w
                movwf       temp_status ; save STATUS register

                btfsc       INTCON,RBIF ;  PORTB change?
                call        PORTB_ISR
                btfsc       PIR1,TMR2IF ; Timer 2 interrupt ?
                call        Timer2_ISR
                btfsc       PIR2,TMR1IF ; Timer 1 interrupt ?
                call        Timer2 ISR
Restore_context:
                movf        temp_status,w
                movwf       STATUS      ; restore STATUS reg.
                movf        temp_w,w    ; restore WREG
                retfie                  ; return from interrupt
```

基本中断
动手实验

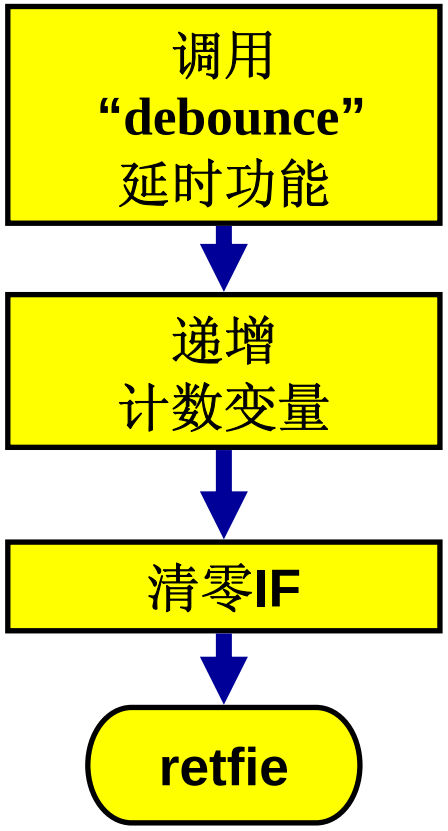
Microchip Technology Inc.
RTC
Regional Training Centers

基本中断

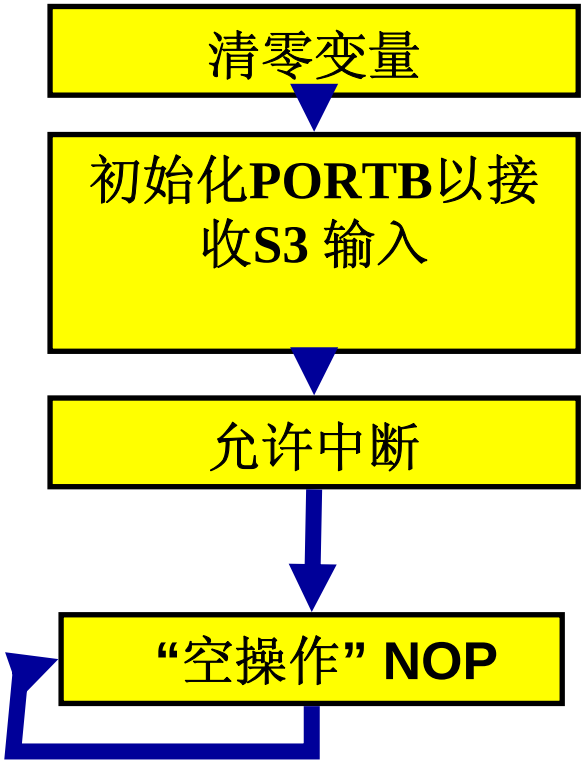
- 本实验的目标是：
  - 学习如何设置和允许中档PIC的内部中断
  - 更加熟悉MPLAB IDE、PICdem2 Plus和ICD2
    - 编译项目
    - 使用 ICD设置断点

基本中断实验概述

中断向量



主程序



实验细节（续）

- 代码位于C:\RTC\201_ASP\Lab1-INT中
- 使用MPLAB和ICD在代码中设置“断点”以查看变量push_count的变化值

Microchip Technology Inc.
RTC
Regional Training Centers

需要了解的内容

- INTCON寄存器位的功能
- 必须按顺序移除跳线器“J6”，以使INT0引脚工作
- 给出了一个名为 “**debounce**”的子程序，该子程序延迟处理并阻止S3弹出从而产生多个中断（稍后的实验中将详细介绍）
- 如何在MPLAB中设置断点和 “*Watch 窗口*”

基本中断实验解决方案

```
    bsf             STATUS,RP0      ; go to bank1
;*
;************** add line of code to set PORTB pin 0 as input
;*
    bsf             TRISB,0         ; Set RB0 ( bit 0 of PORTB ) as input
;*
;*************  add one line of code to set enable the INT 0 interrupt
;*
    bsf             INTCON,INTE
;*
;************** Add line of code to enable Global Interrupts
;*
    bsf             INTCON,GIE
    bcf             STATUS,RP0      ; return to bank0
```

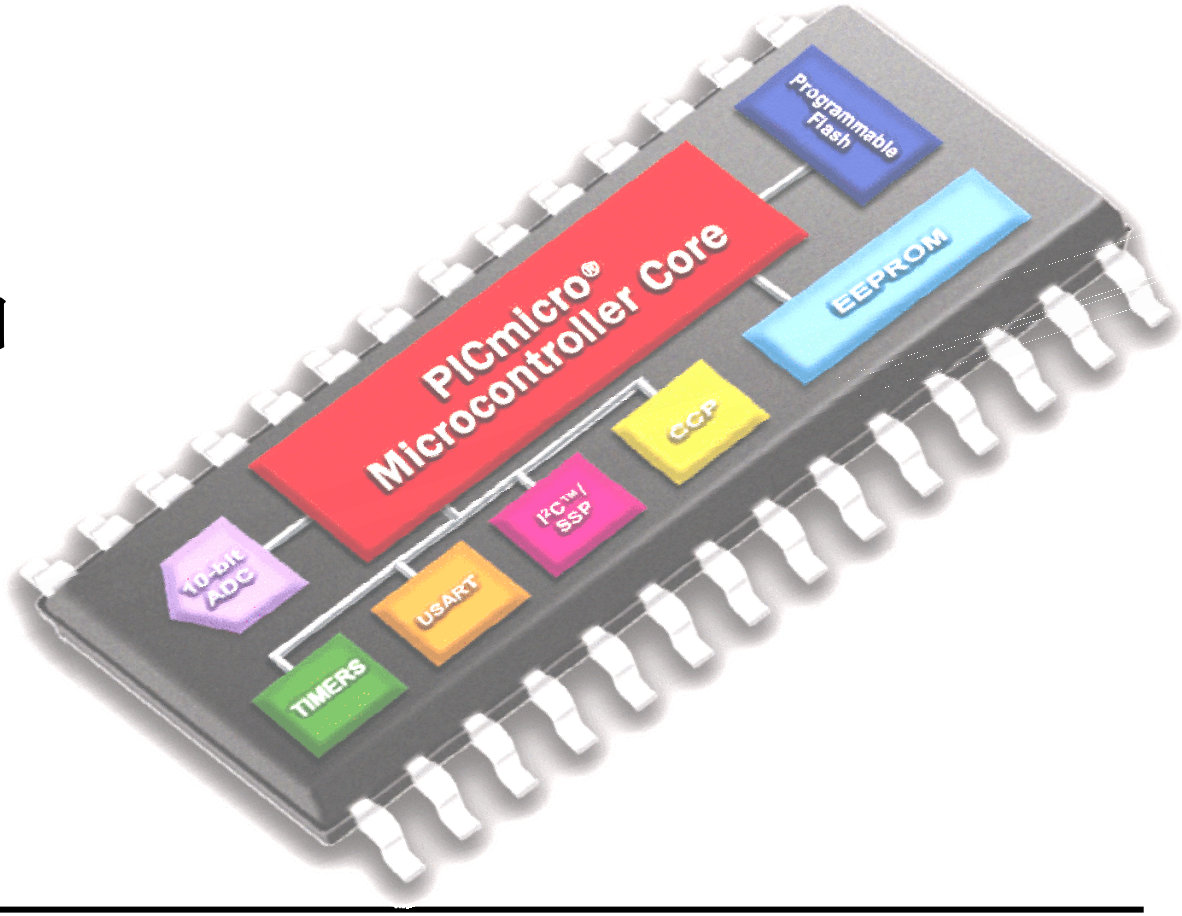
外设

Microchip Technology Inc.
RTC
Regional Training Centers

Microchip Technology Inc.
RTC
Regional Training Centers

中档系列外设

- 数字I/O端口
- 定时器（0、1或2）
- 增强型捕捉/比较/PWM
- 比较器
- 模数转换器
- EUSART
- I²C和SPI串行接口



Microchip Technology Inc.
RTC
Regional Training Centers

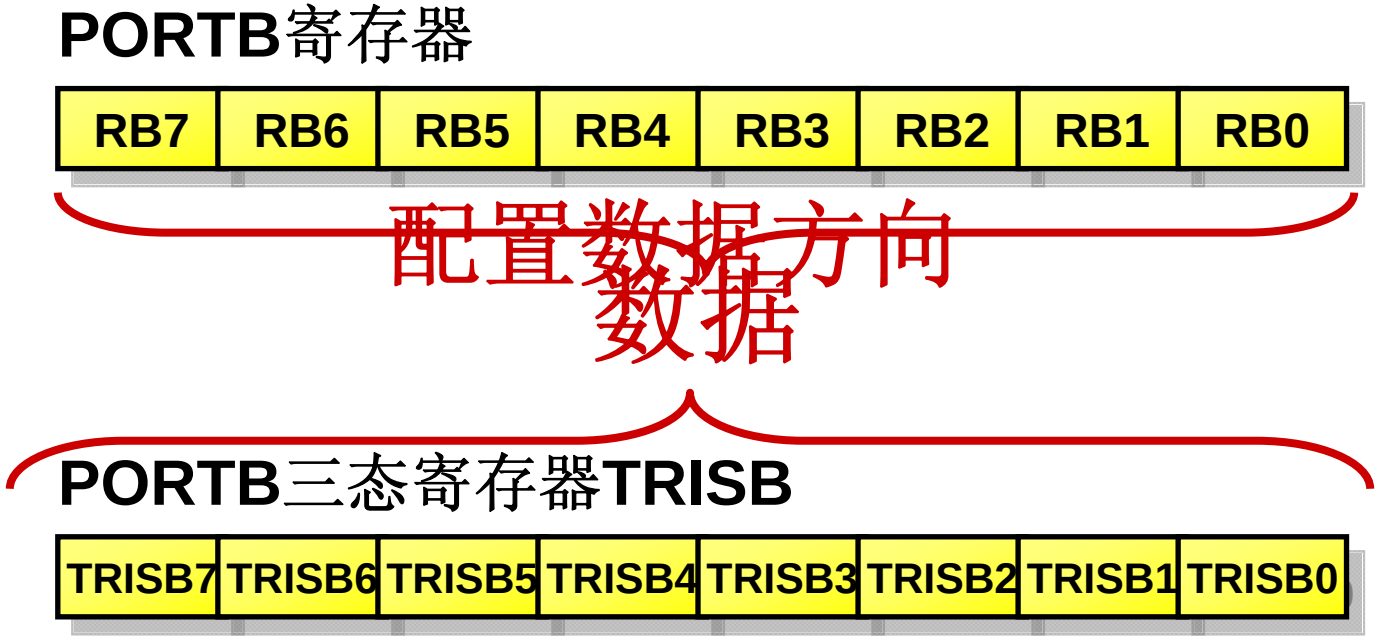
数字I/O概述

- 最多35个双向I/O口，其中一些与外设功能复用
- 高驱动能力
- 直接单周期位操作
- 多数I/O具有ESD
- 最初复用的I/O引脚被默认为模拟输入引脚（高阻态）

Microchip Technology Inc.
RTC
Regional Training Centers

PORTx和TRISx寄存器

- 每个端口（A、B、C、D或E）都具有相应的数据方向寄存器TRISx



1 = 相应的PORTB 引脚为输入
0 = 相应的PORTB 引脚为输出

ANSEL和ANSELH寄存器

- 两个用来将I/O配置为数字引脚的寄存器
  - 最初与模拟外设复用的I/O 被默认为模拟输入

模拟选择寄存器（ANSEL）

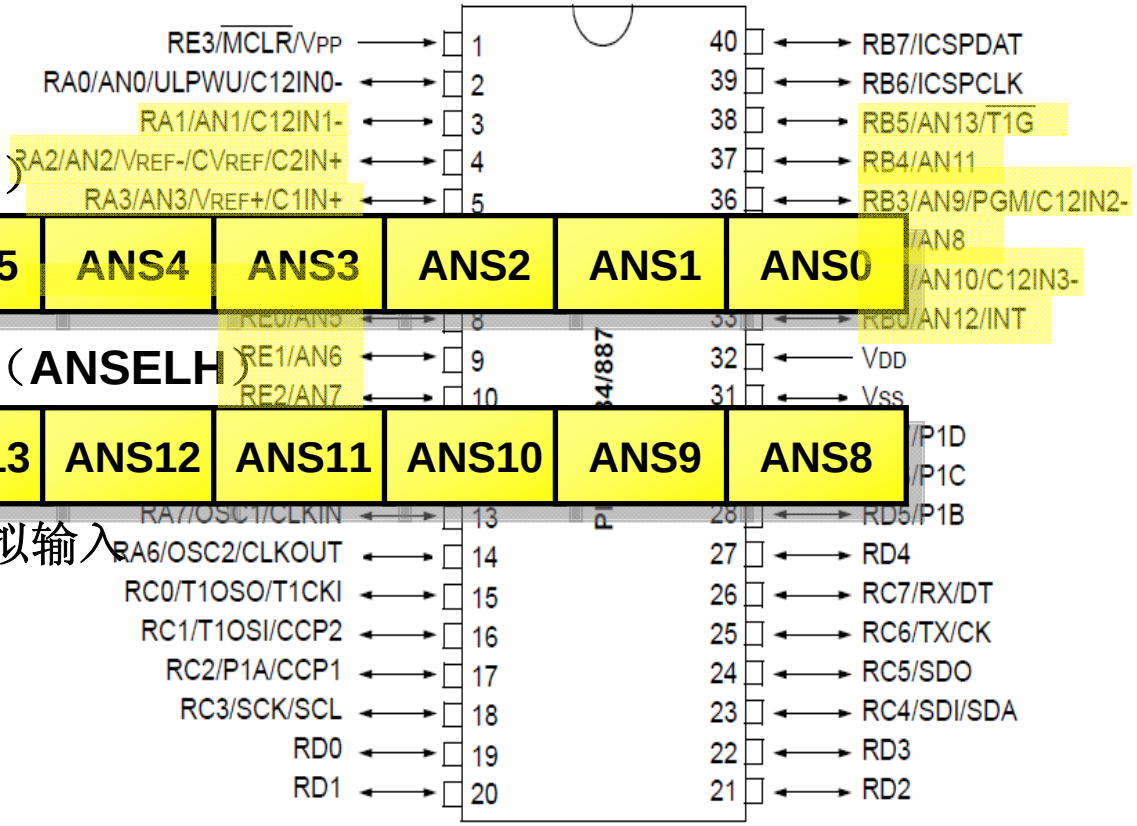
ANS7	ANS6	ANS5	ANS4	ANS3	ANS2	ANS1	ANS0

模拟选择寄存器的高字节（ANSELH）

		ANS13	ANS12	ANS11	ANS10	ANS9	ANS8

1 = 引脚被配置为模拟输入

0 = 数字I/O



初始化数字I/O

- 示例：
  - 初始化**PORTB**，设置**RB<7:4>**为输入，**RB<3:0>**为输出

```
;-----------configure PORTB for digital ------------------------------------
banksel     PORTB                   ;Go to bank containing PORTB
                                    ;register
clrf        PORTB                   ;Clear PORTB
banksel     ANSELH                  ;Go to bank containing ANSELH
                                    ;register
clrf        ANSELH                  ;Set as all digital
clrf        ANSEL
;-----------Set up direction of each PORTB pin-----------------------------
banksel     TRISB                   ;Go to bank containing TRISB
                                    ;register
movlw       b'11110000'             ;Value to set TRISB<7:4> high
                                    ;and TRISB<3:0> low move into
                                    ;W register
movwf       TRISB                   ;Move value in W into TRISB
```

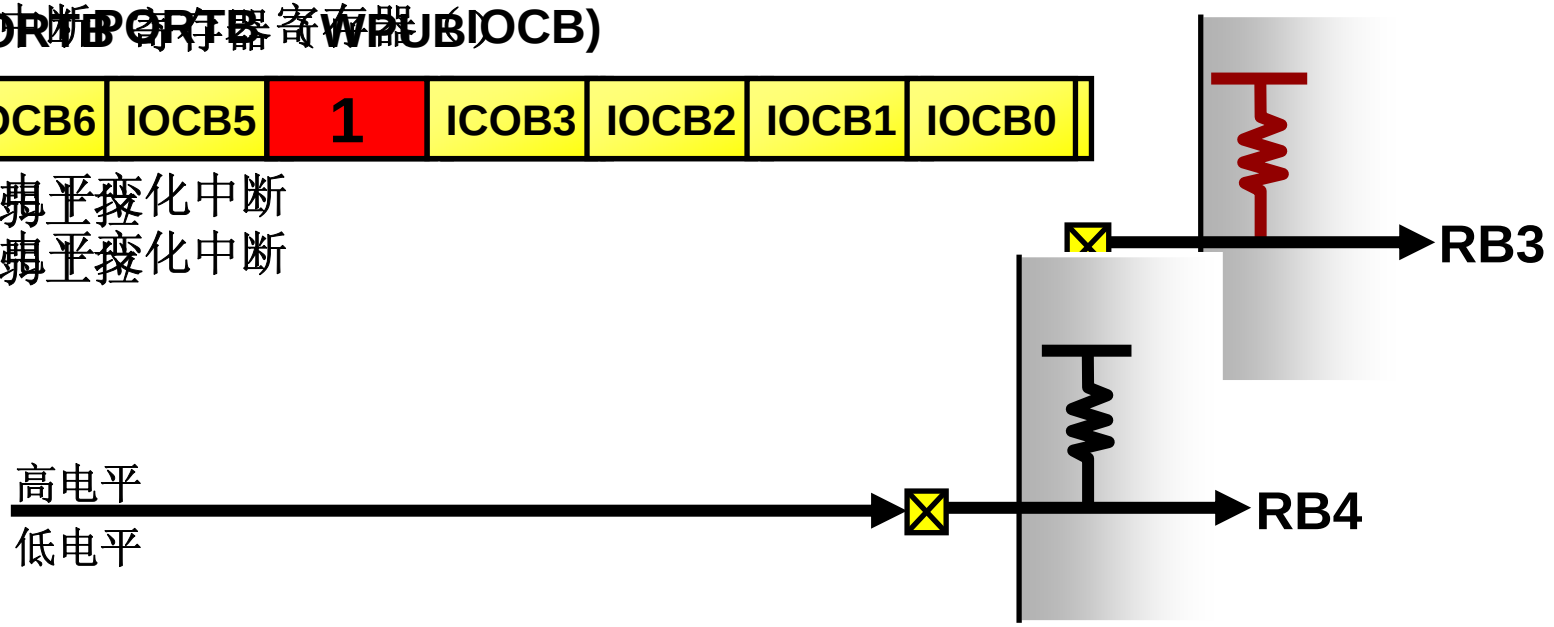
PORTB电平变化中断

- 在PIC16F877中所有PORTB引脚都具有电平变化中断和弱上拉选项

电平变化中断PORTB寄存器（IOCB）

IOCB7	IOCB6	IOCB5	1	ICOB3	IOCB2	IOCB1	IOCB0

1 = 允许电平变化中断
0 = 禁止电平变化中断



中断控制寄存器（INTCON）

GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF

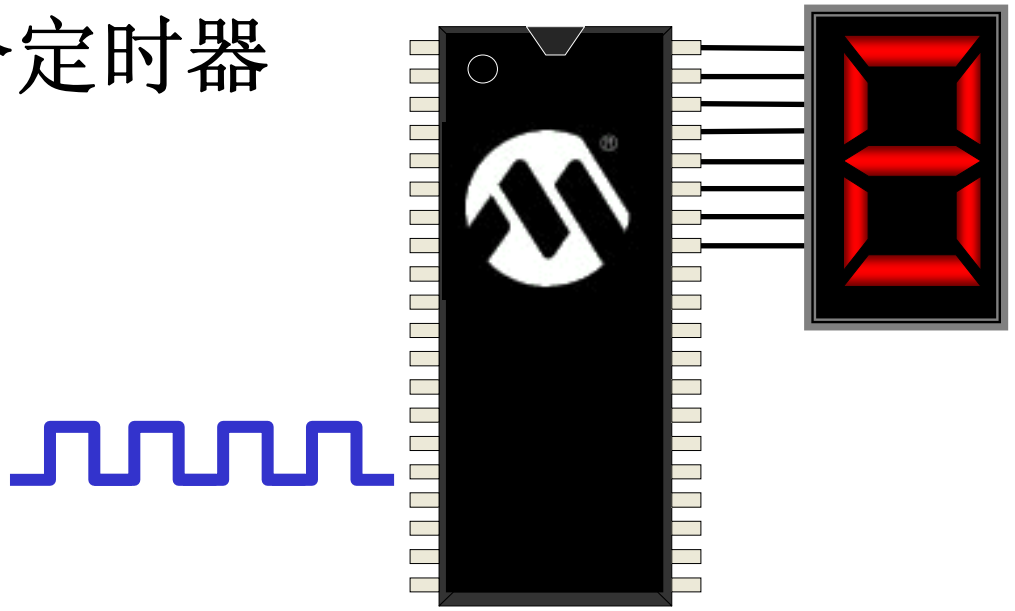
*必须先读/写PORTB，然后才能用软件清零RBIF

定时器

Microchip Technology Inc.
RTC
Regional Training Centers

定时器

- 定时器可用于多种功能：
  - 时序参考以产生中断
  - 计算事件的数量
  - 波形产生等
- **PIC16F887有3个定时器**
  - **Timer0**
  - **Timer1**
  - **Timer2**

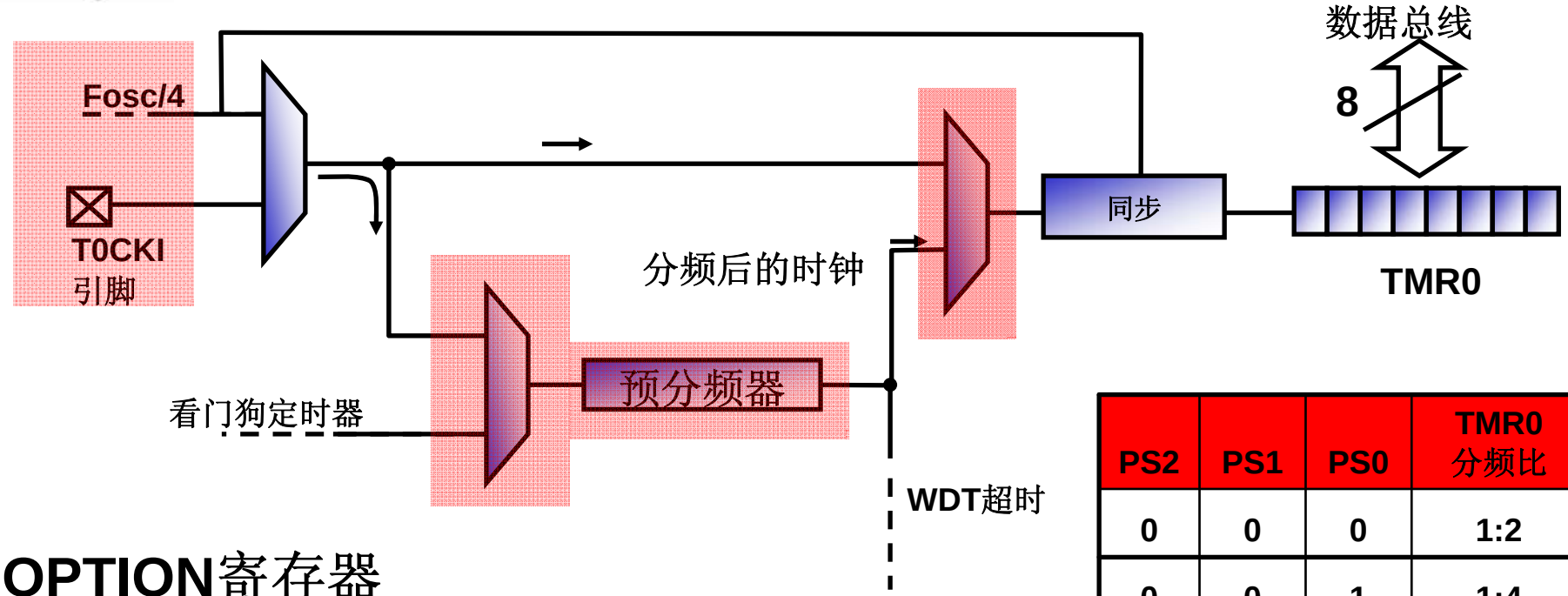


Microchip Technology Inc.
RTC
Regional Training Centers

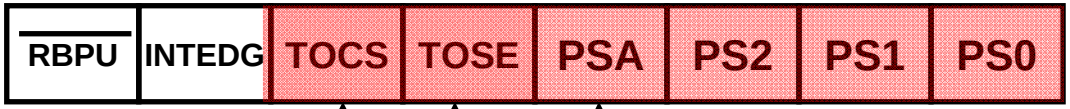
PIC16F887 定时器比较

	TIMER0	TIMER1	TIMER2
寄存器大小	8位（TMR0）	16位（TMR1H:TMR1L）	8位（TMR2）
时钟源（内部）	Fosc/4	Fosc/4	Fosc/4
时钟源（外部）	T0CKI引脚	T1CKI引脚或Timer 1振荡器（T1OSC）	无
可用的时钟分频比（分辨率）	8位预分频器（1:2→1:256）	3位预分频器（÷1、÷2、÷4或÷8）	预分频器（1:1、1:4或1:8） 后分频器（1:1→1:16）
中断事件和标志位位置	溢出时FFh→00h（T0IF在 INTCON中）	溢出时FFFFh→0000h（TMR1F在PIR1中）	TMR2与PR2匹配（TMR2F在PIR2中）
将PIC器件从休眠中唤醒	否	是	否

Timer 0框图



OPTION寄存器



预分频比选择位

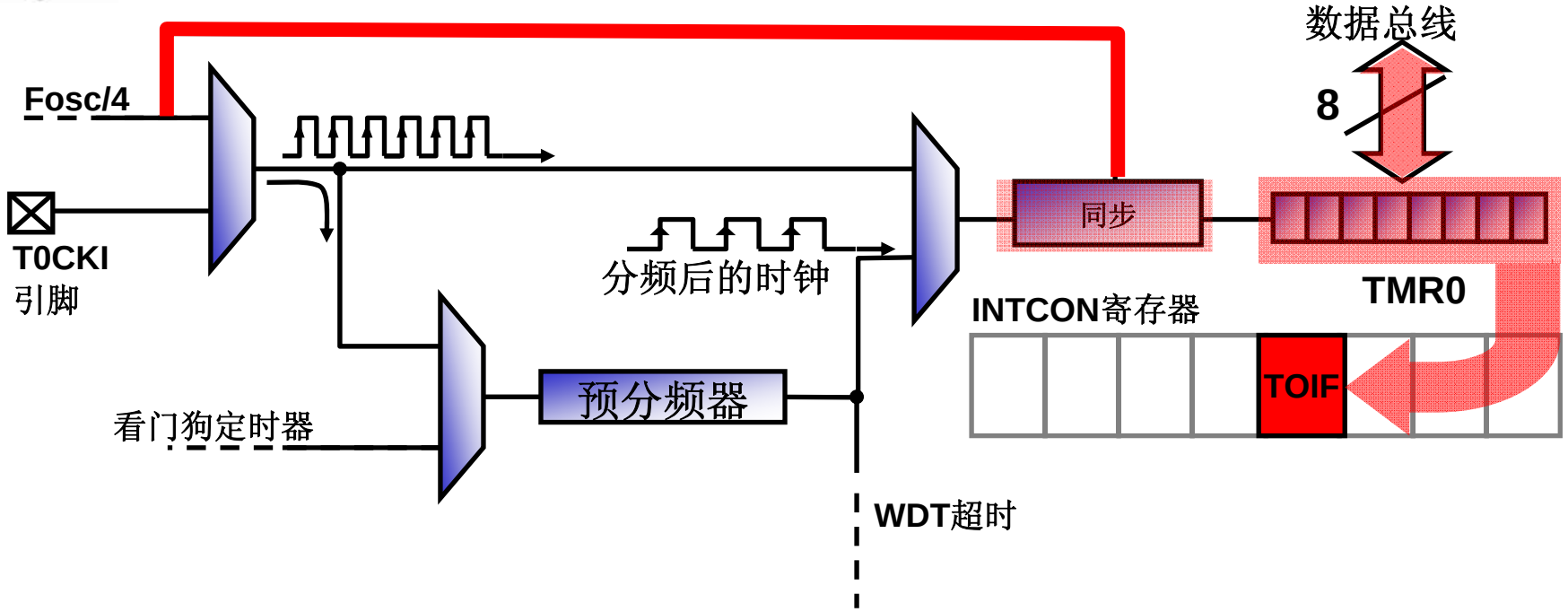
预分频器分配位
1 = 预分频器分配给WDT
0 = 预分频器分配给Timer 0

时钟源边沿选择位
1 = 在T0CKI引脚上的下降沿递增TMR0
0 = 在T0CKI引脚上的上升沿递增TMR0

TMR0时钟源选择位
1 = 在T0CKI引脚上的跳变递增TMR0
0 = 内部指令周期时钟(Fosc/4)

PS2	PS1	PS0	TMR0 分频比
0	0	0	1:2
0	0	1	1:4
0	1	0	1:8
0	1	1	1:16
1	0	0	1:32
1	0	1	1:64
1	1	0	1:128
1	1	1	1:256

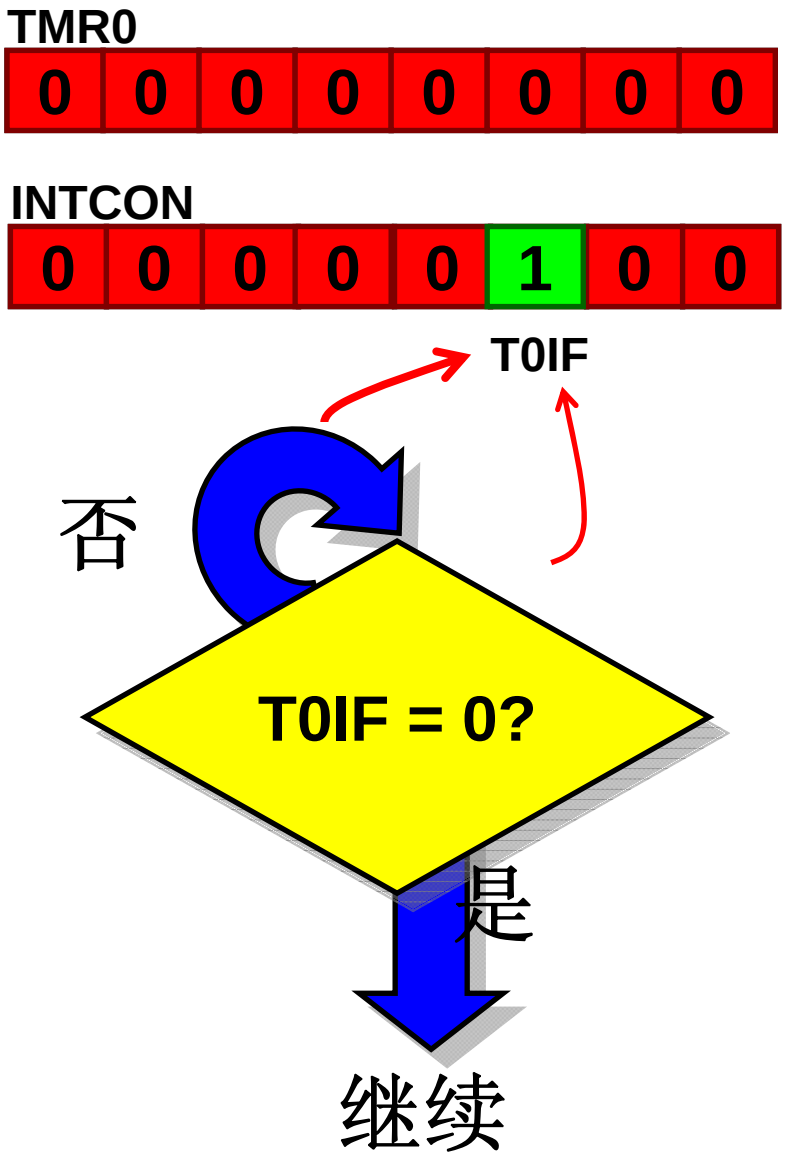
Timer 0框图



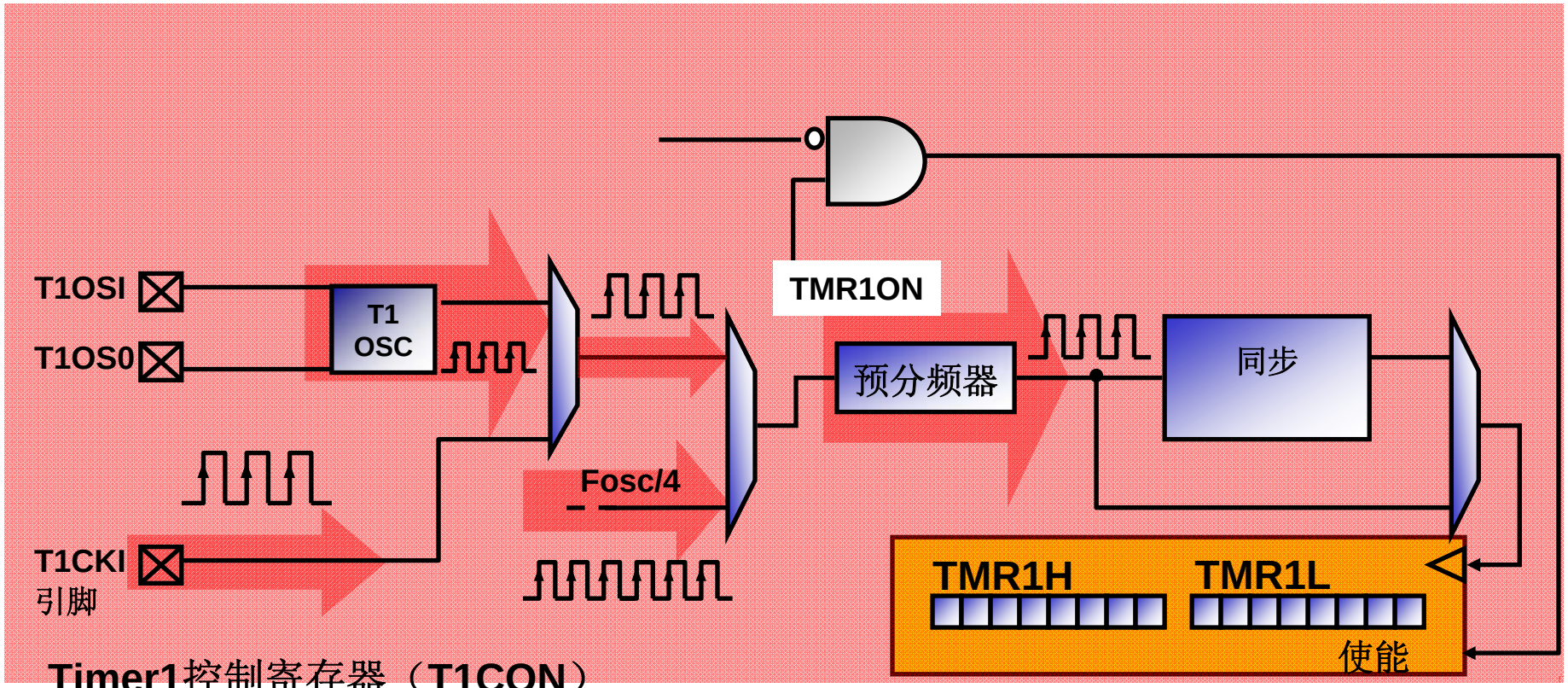
- 可通过数据总线读写Timer 0
  - 写操作将禁止定时器递增（时间为2个T_{CY}）
- INTCON寄存器中的T0IF由Timer 0计满回零置1
  - FFh → 00h

Timer0初始化（内部时钟源）

```
;Make sure the TMR0 register is clear
banksel         TMR0
clrf            TMR0

; Clear T0IF
bcf             INTCON,T0IF

;Setup the following in the OPTION_REG
;Timer0 increment from internal clock
;with a prescaler of 1:16
banksel         OPTION_REG
movlw           b'00000011'
movwf           OPTION_REG

;The TMR0 interrupt is disabled, do polling
;on the T0IF overflow bit

btfss           INTCON, T0IF
goto            $-1

<继续>
```



Timer1框图



Timer1控制寄存器（T1CON）

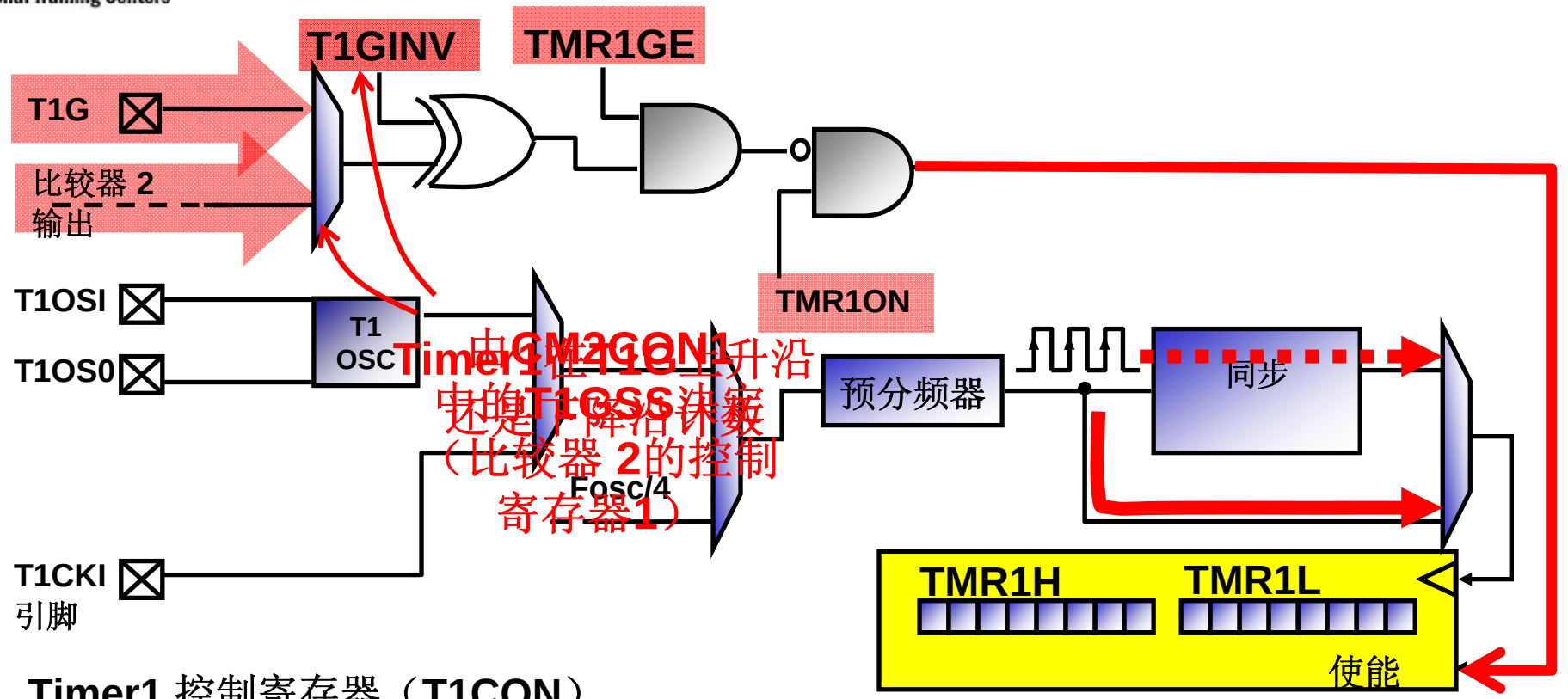
T1GINV	TMR1GE	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR1ON

LP 振荡器使能位
1 = 选择T1OSC
0 = 可以使用T1CKI

T1CKPS1	T1CKPS0	分频比
1	1	1:8
1	0	1:4
0	1	1:2
0	0	1:1

Timer1使能位
1 = 使能Timer1

Timer1框图



Timer1在T1G上升沿中的T1OSS决定还是下降沿计数（比较器 2的控制寄存器1）

Timer1 控制寄存器（T1CON）

T1GINV	TMR1GE	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR1ON

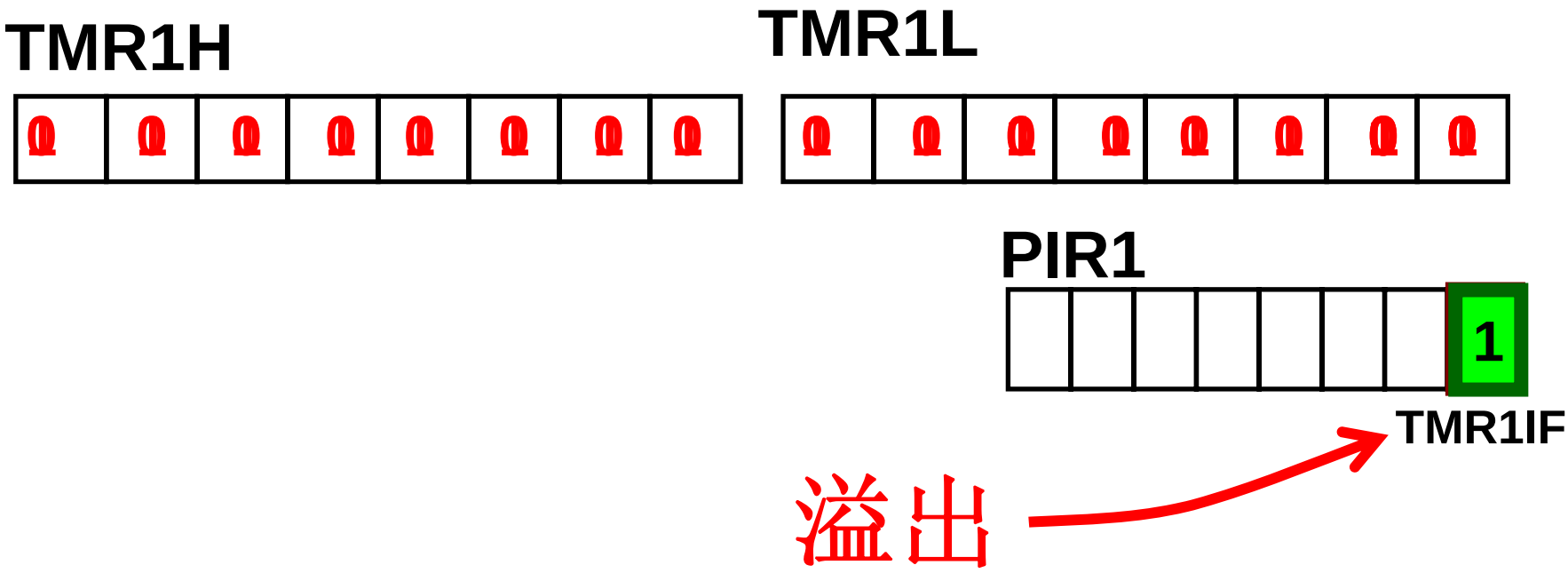
Timer1门控翻转位
1 =门
0 =门

Timer1门控使能位
1 = 若门控信号无效，则使能Timer1
0 = 使能Timer1

Timer1外部时钟输入同步位
1 = 不与外部时钟输入同步
0 = 与外部时钟输入同步

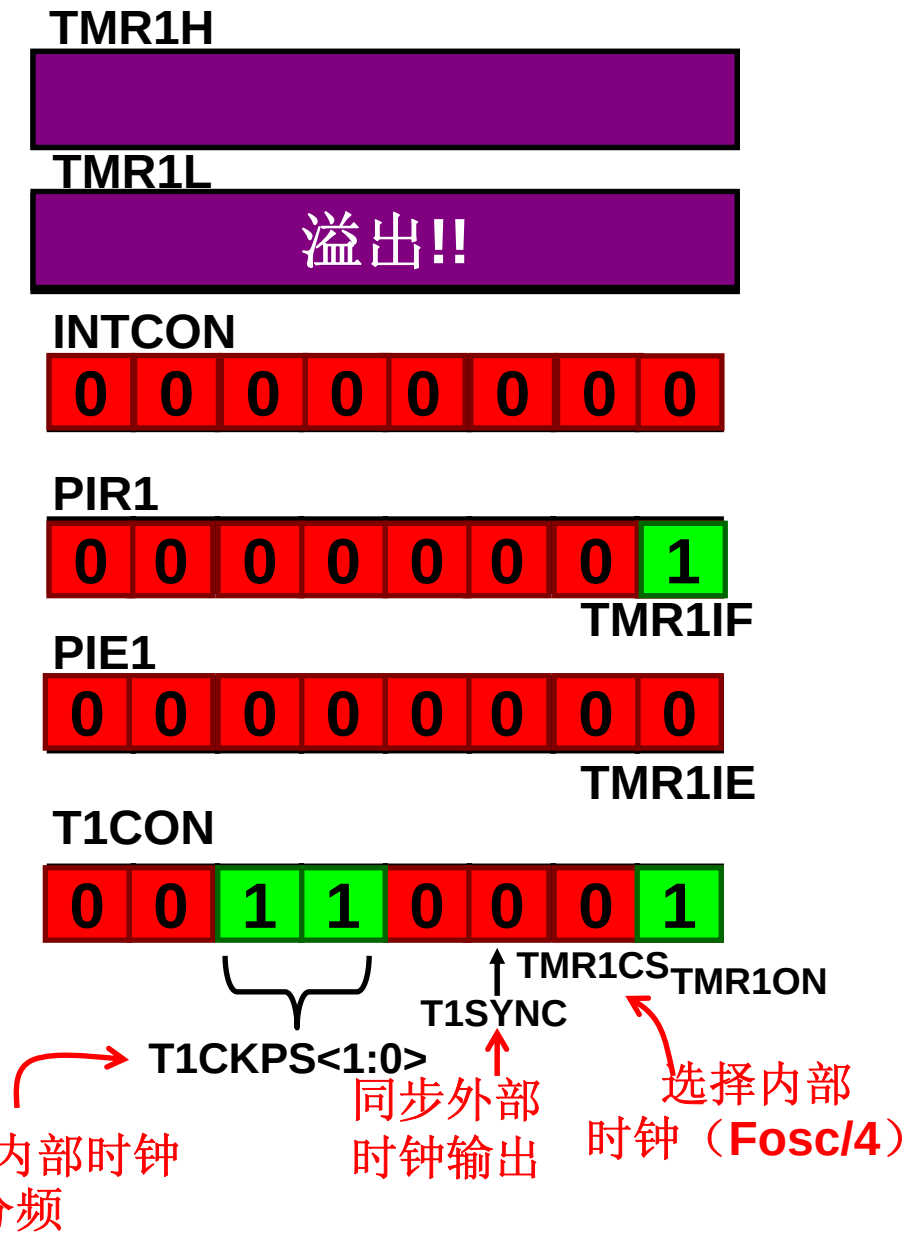
若TMR1ON为0，则忽略TMR1GE位

Timer1中断



Timer1初始化（内部时钟源）

```
;Make sure the TMR1 registers are clear
banksel         TMR1H
clrf            TMR1H
clrf            TMR1L

;Make sure the TMR1IF flag in PIR1
; is cleared
banksel         PIR1
bcf             PIR1,TMRIF


;Setup T1CON register for internal clock with
;1:8 prescaler, Timer1 is stopped and T1 osc
;is disabled
movlw           b'00110000'
movwf           T1CON

;Start Timer1 incrementing
bsf             T1CON, TMR1ON

;The TMR1 interrupt is disabled, do polling
;on the TMR1IF overflow bit
btfss           PIR1, TMR1IF
goto            $-1
```



Timer 1实验

Microchip Technology Inc.
RTC
Regional Training Centers

Timer 1实验

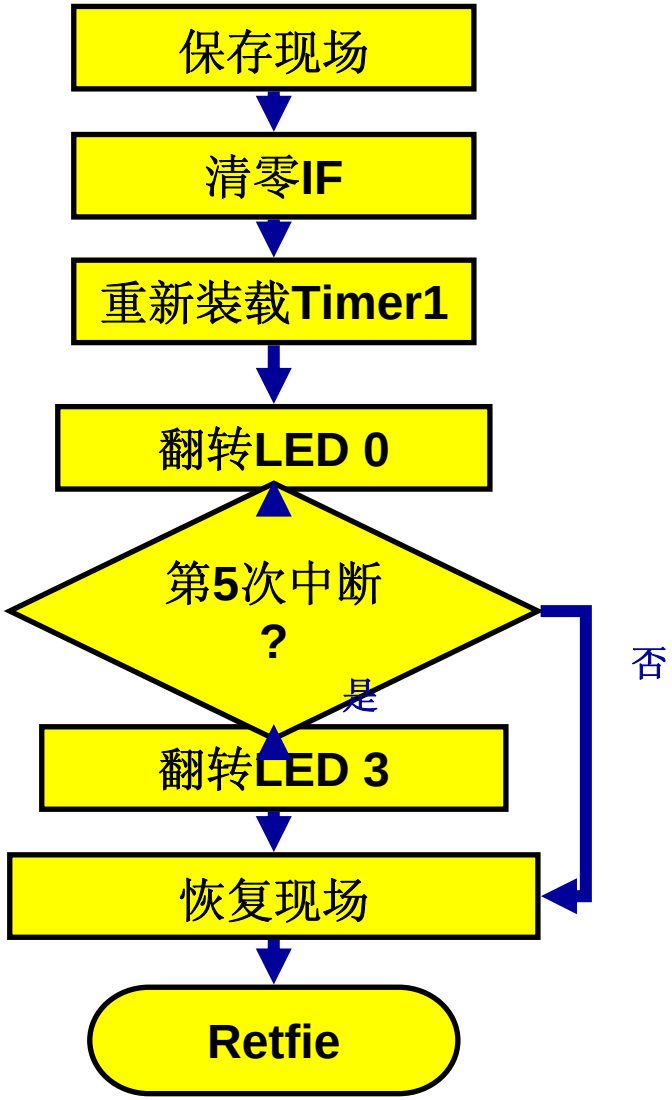
- 本实验目标是熟悉Timer1的工作原理

以及

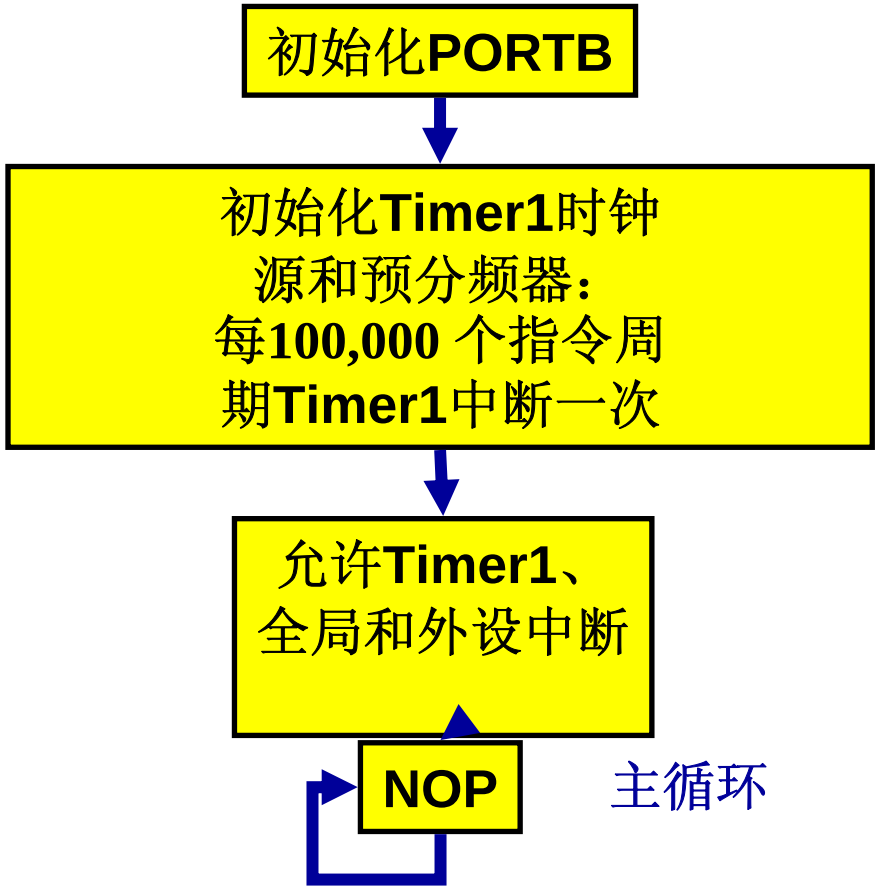
- 获取允许外设中断的经验

实验概述

中断向量



主程序



实验细节

- 本实验的代码位于

  C:\RTC\201_ASP\Lab2-TMR1

- 在lab2.asm中完成以下任务
  - 将Timer 1时钟源设置为Fosc/4
  - 将Timer 1预分频比设置为2
  - Timer 1装入0x3CB0 （**65,356 – 50,000**）
  - 启动Timer 1
  - 允许Timer 1、全局和外设中断

Microchip Technology Inc.
RTC
Regional Training Centers

需要了解的内容

- INTCON、T1CON、TMR1H、TMR1L和PIE1的寄存器操作
- 使用值0x3CB0和预分频比2，Timer1将每100,000个指令周期溢出一次
- 提供了翻转LED的中断向量代码

Microchip Technology Inc.
RTC
Regional Training Centers

Timer 1解决方案

```
;
;  ***************************************************************************
;   Set code to Select clock source, Set pre-scaler to 2, load hex 3CB0 into Timer1
;   and turn on Timer1
;  ***************************************************************************
;
        movlw           0x3C                    ; initialize TMR1L and TMR1H
        movwf           TMR1H
        movlw           0xB0
        movwf           TMR1L
        bsf             T1CON,T1CKPS0           ; set pre-scaler to 2
        bcf             T1CON,TMR1CS            ; set Clock source to Fosc/4
        bsf             T1CON,TMR1ON            ; turn Timer1 on
;
;***************************************************************************
; Enable Timer1 interrupts, Peripheral Interrupts and Global Interrupts
;***************************************************************************
;
;
        bsf             STATUS,RP0              ; go to bank1
        bsf             PIE1,TMR1IE
        bsf             INTCON,GIE
        bsf             INTCON,PEIE
        bcf             STATUS,RP0              ; return to bank0
```

实验问答

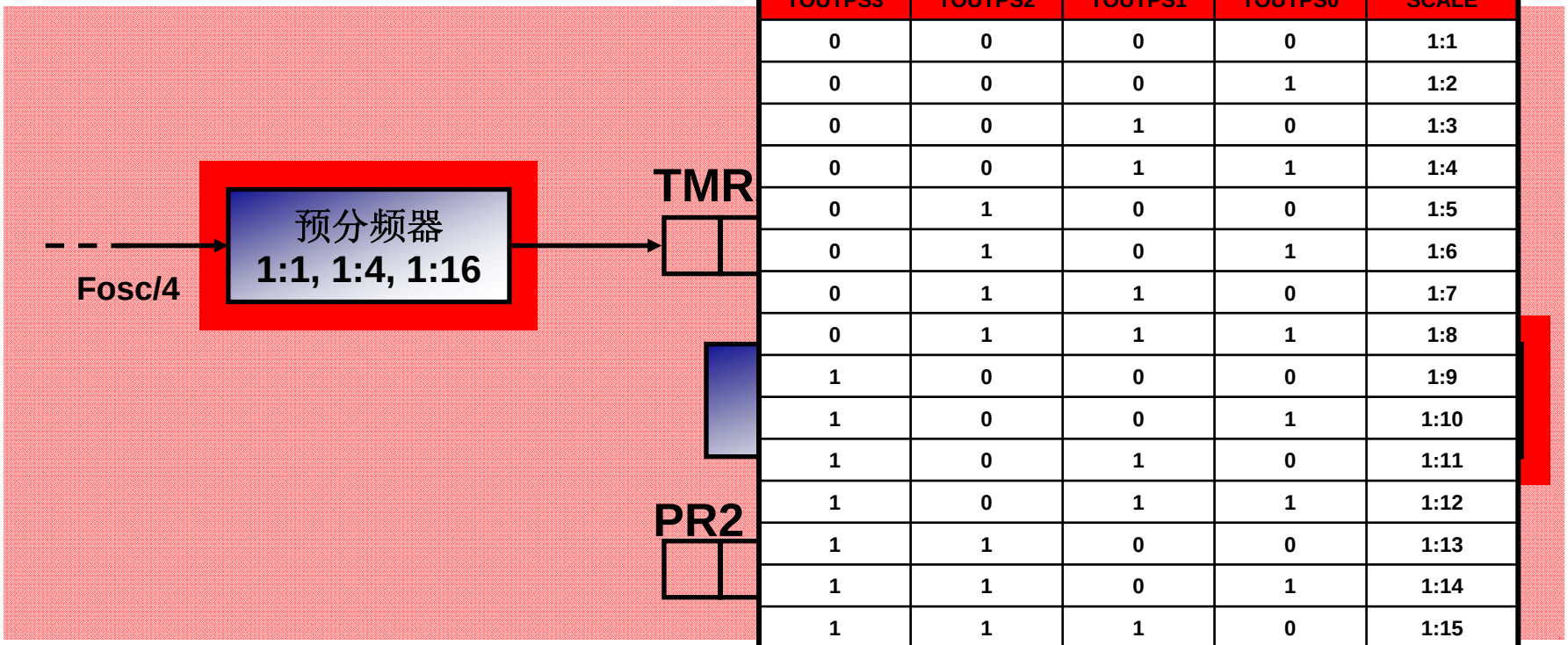
问： Timer 1在响应中断这段时间内是否继续工作？

答： 是

问： 这将对要被重新装载到TMR1L和TMR1H的值产生什么影响？

答：影响很大 – 要实现高精度，应当考虑重新装载Timer1的延时

Timer2框图



TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	SCALE
0	0	0	0	1:1
0	0	0	1	1:2
0	0	1	0	1:3
0	0	1	1	1:4
0	1	0	0	1:5
0	1	0	1	1:6
0	1	1	0	1:7
0	1	1	1	1:8
1	0	0	0	1:9
1	0	0	1	1:10
1	0	1	0	1:11
1	0	1	1	1:12
1	1	0	0	1:13
1	1	0	1	1:14
1	1	1	0	1:15
1	1	1	1	1:16

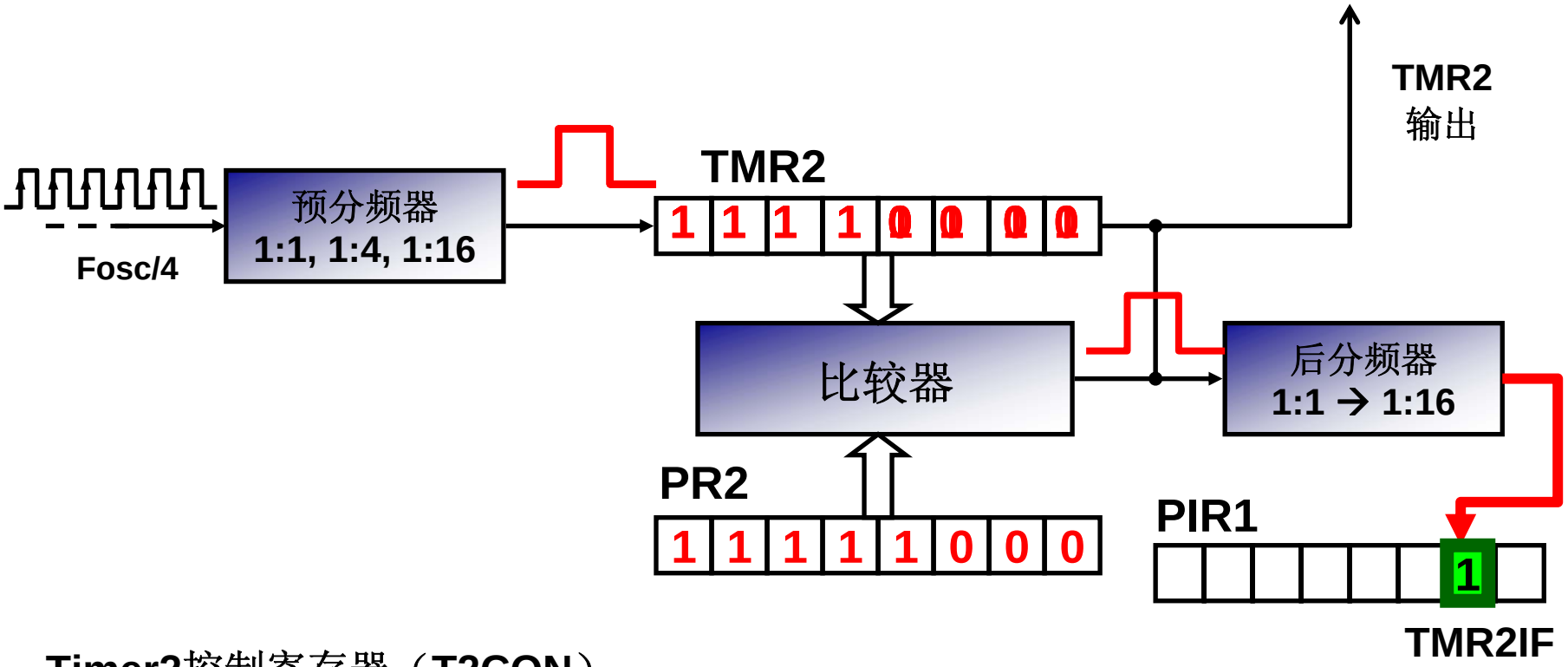
Timer2控制寄存器（T2CON）

	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0

T2CKPS1	T2CKPS2	分频比
0	0	1:1
0	1	1:4
1	X	1:16

Timer2使能位
1 =使能Timer2

Timer2框图

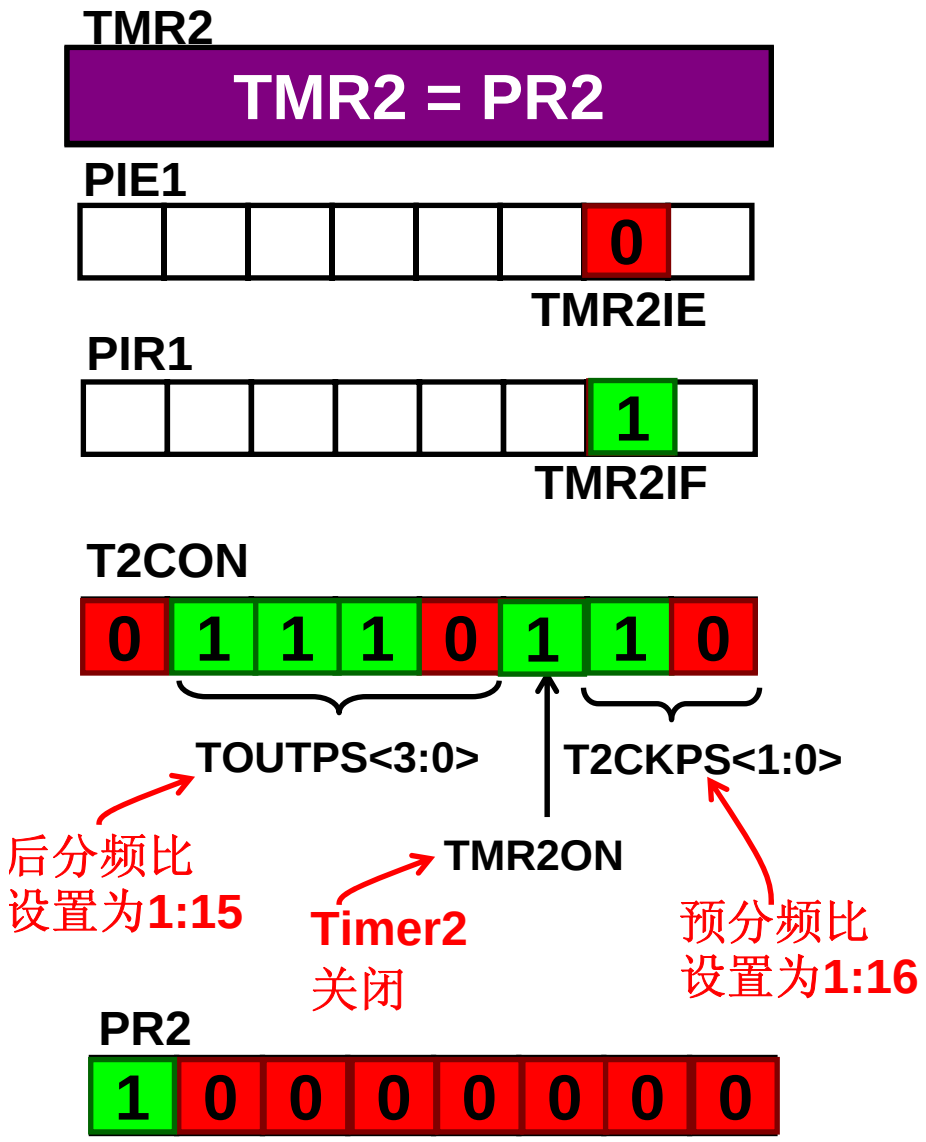


Timer2控制寄存器（T2CON）

	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0

Timer2初始化

T2CKPS1	T2CKPS2	分频比
0	0	1:1
0	1	1:4
1	X	1:16



Timer 2实验

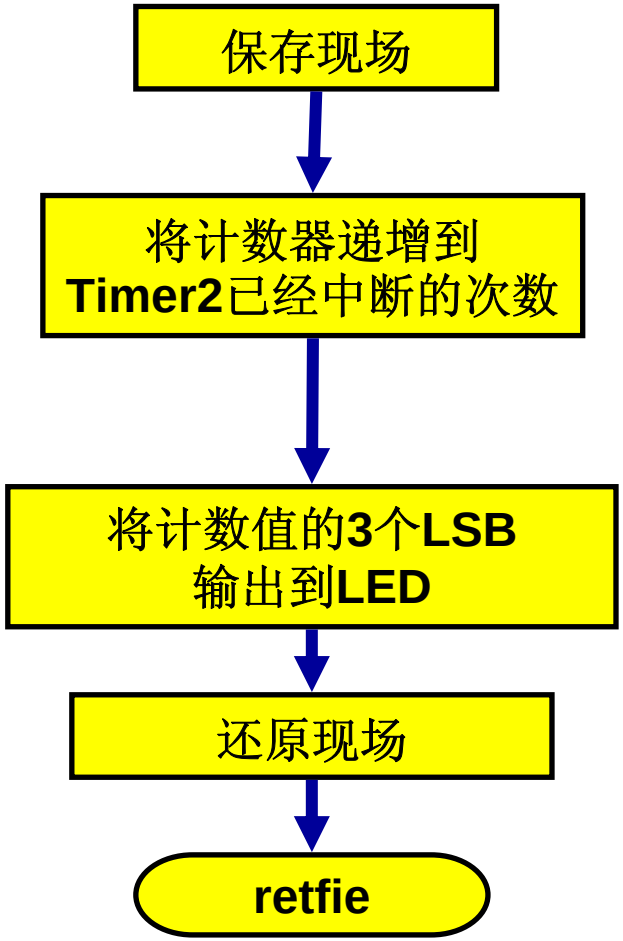
Microchip Technology Inc.
RTC
Regional Training Centers

Timer 2实验

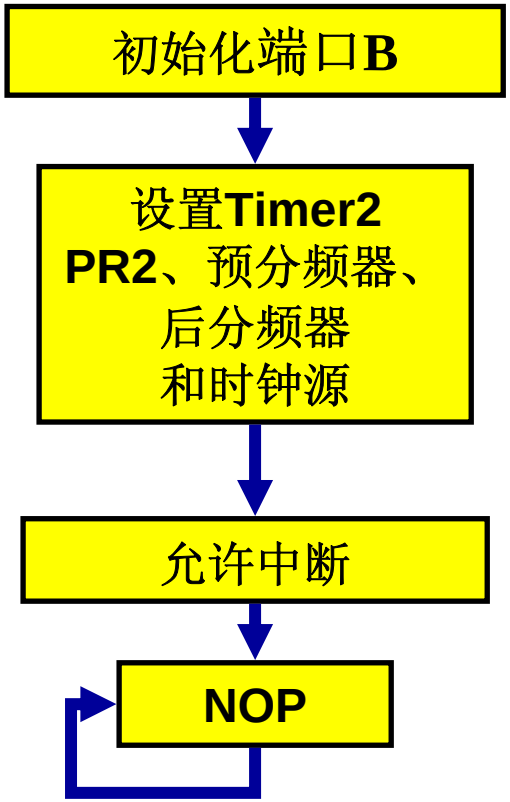
- 实验3的目标是熟悉以下知识：
  - Timer2时钟源
  - 预分频值
  - 后分频值
  - 使能Timer2
  - 将要使Timer2成功产生中断所需的所有允许位置1

实验概述

中断向量



主程序



实验细节

- 本实验代码位于
  C:\RTC\201_ASP\Lab3-TMR2

- 完成代码的以下部分
  - 将Timer 2预分频比设置为4
  - 将Timer 2 后分频比设置为13
  - 使能Timer 2
  - 配置GIE、PEIE和Timer 2中断允许位

Microchip Technology Inc.
RTC
Regional Training Centers

需要了解的内容

- 本实验需要的特殊功能寄存器（SFR） 为：INTCON、PIE1、PR2和T2CON

- 将PR2设置为250、预分频比设置为4，并将后分频比设置为13，使用4Mhz振荡器时（ Fosc/4 = 1Mhz ），Timer 2每13 ms中断一次 （大约1/80s）

Timer 2实验解决方案

```
;
;*****************************************************************************
;   configure Timer2   Prescaler of 4, PR2 of 250 and a postscaler of 13
;              and turn timer2 on.
;*****************************************************************************
;
;
      banksel         0                       ; set bank to 0
      movlw           0x68                    ; set 13 as postscaler
      movwf           T2CON
      bsf             T2CON,T2CKPS0           ; set prescaler to 4
      bsf             T2CON,TMR2ON            ; turn on Timer2


;*****************************************************************************
; Enable Timer2 interrupts, Peripheral Interrupts and Global Interrupts
;*****************************************************************************
;
;
      bsf             STATUS,RP0              ; go to bank1
      bsf             PIE1,TMR2IE
      bsf             INTCON,GIE
      bsf             INTCON,PEIE
      bcf             STATUS,RP0              ; return to bank0
```

Microchip Technology Inc.
RTC
Regional Training Centers

实验问答

问： Timer2是否与Timer1一样在中断响应延时期间继续运行？

答： 是的！

问：要确保精确的中断周期，用户是否必须考虑自由运行的Timer2 ？

答： 不用，因为中断是在匹配而非溢出时产生的

增强型
捕捉/比较/PWM模块

Microchip Technology Inc.
RTC
Regional Training Centers

ECCP概述

- 捕捉功能
  - 计算事件持续时间
  - 比较

ECCP模式	定时器资源
捕捉	Timer 1
比较	Timer 1
PWM	Timer 2

  - 为各种模式连接提供增强型功能

输

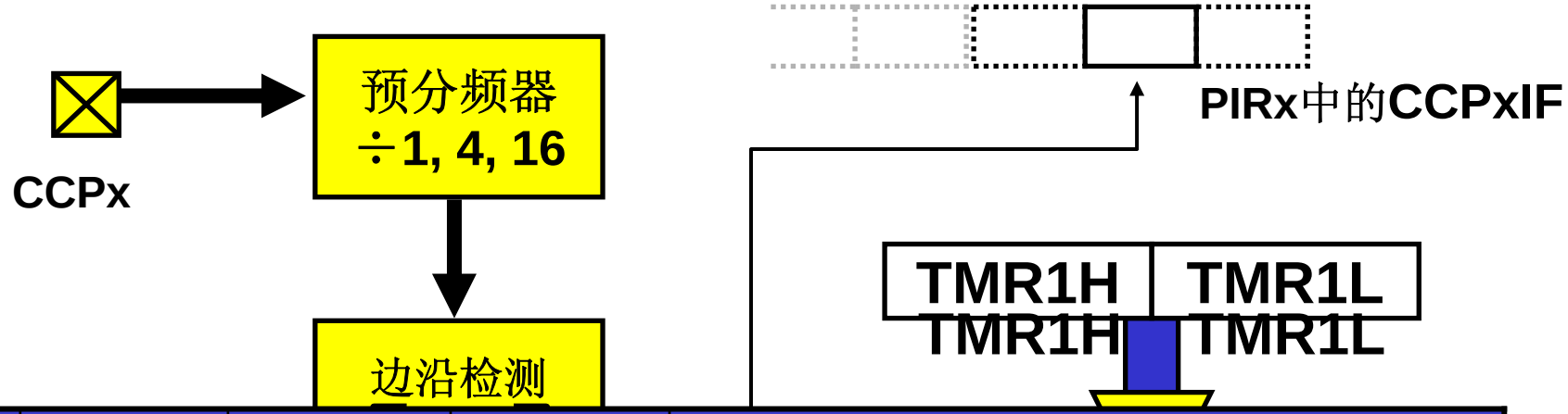
*** 模块与Timer 1和Timer 2交互**

Microchip Technology Inc.
RTC
Regional Training Centers

ECCP 控制寄存器

CCPxM3	CCPxM2	CCPxM1	CCPxM0	选择EECP模式
0	0	0	0	捕捉/比较/PWM 关闭（复位ECCP模块）
0	0	0	1	未用（保留）
0	0	1	0	比较模式，匹配时翻转输出
0	0	1	1	未用（保留）
0	1	0	0	捕捉模式，每个下降沿捕捉一次
0	1	0	1	捕捉模式，每个上升沿捕捉一次
0	1	1	0	捕捉模式，每4个上升沿捕捉一次
0	1	1	1	捕捉模式，每16个上升沿捕捉一次
1	0	0	0	比较模式，匹配时置1输出
1	0	0	1	比较模式，匹配时清零输出
1	0	1	0	比较模式，匹配时产生软件中断
1	0	1	1	比较模式，触发特殊事件
1	1	0	0	PWM模式；P1A和P1C高电平有效；P1B和P1D 高电平有效
1	1	0	1	PWM模式；P1A和P1C高电平有效；P1B和P1D低电平有效
1	1	1	0	PWM模式；P1A和P1C低电平有效；P1B和P1D高电平有效
1	1	1	1	PWM模式；P1A和P1C低电平有效； P1B和P1D低电平有效

捕捉模式



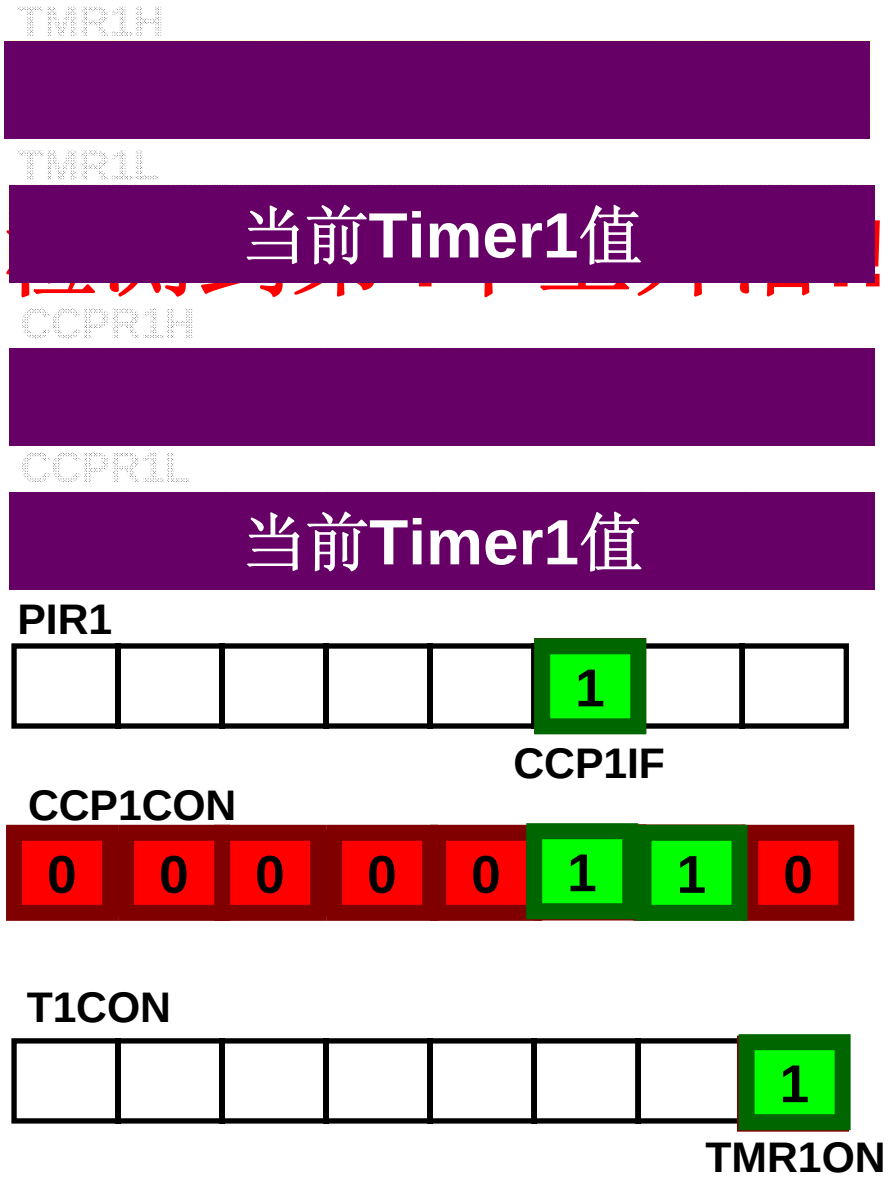
CCPxM3	CCPxM2	CCPxM1	CCPxM0	模式
0	1	0	0	每个下降沿捕捉一次
0	1	0	1	每个上升沿捕捉一次
0	1	1	0	每4个上升沿捕捉一次
0	1	1	1	每16个上升沿捕捉一次

P1M1	P1M0	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0

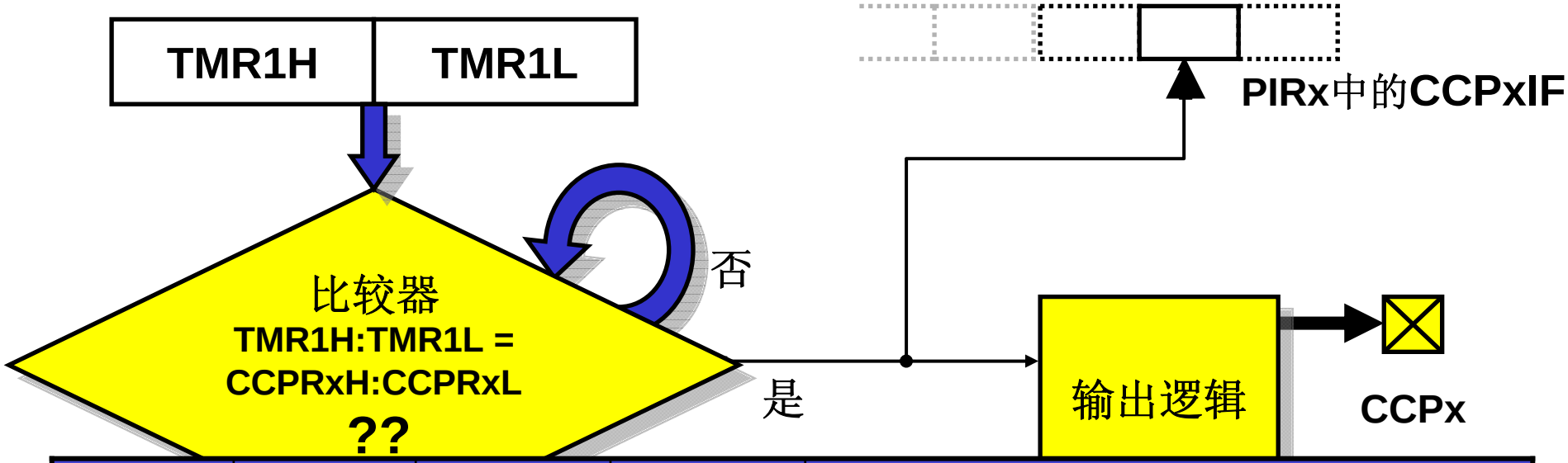
CCPxCON

捕捉模式的初始化

```
;Turn off CCP module
banksel         CCP1CON
clrf            CCP1CON
;Make sure Timer1 is off
bcf             T1CON, TMR1ON
;Clear Timer1 result registers
clrf            TMR1H
clrf            TMR1L
;Disable all interrupts for CCP
bcf             PIR1, CCP1IF
banksel         PIE1
bcf             PIE1, CCP1IE
;Set CCP1 pin for input
bsf             TRISC, 2
;Initialize Capture for every 4th rising edge
banksel         CCP1CON
movlw           b'00000110'
movwf           CCP1CON
;Start Timer1 incrementing
bsf             T1CON, TMR1ON
;Test the CCP1IF flag for capture
btfss           PIR1, CCP1IF
goto            $-1
```



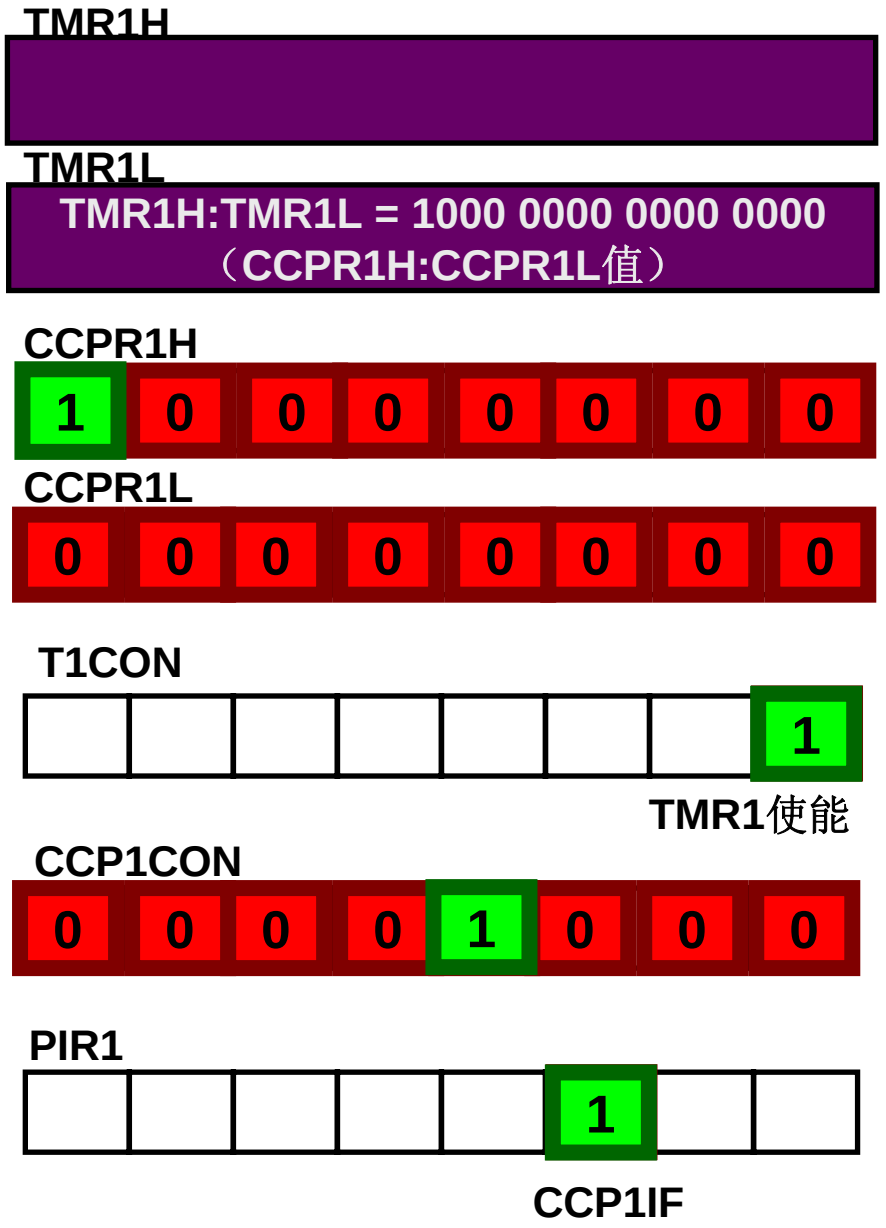
比较模式



CCPxM3	CCPxM2	CCPxM1	CCPxM0	模式
1	0	0	0	匹配时置1输出（CCPxIF置1）
1	0	0	1	匹配时清零输出（CCPxIF置1）
1	0	1	0	匹配时产生软件中断（CCPxIF置1，CCP1 引脚不受影响）
1	0	1	1	触发特殊事件（CCPxIF置1，CCP1复位TMR1或TMR2，如果使能ADC还将启动A/D 转换）

比较模式初始化

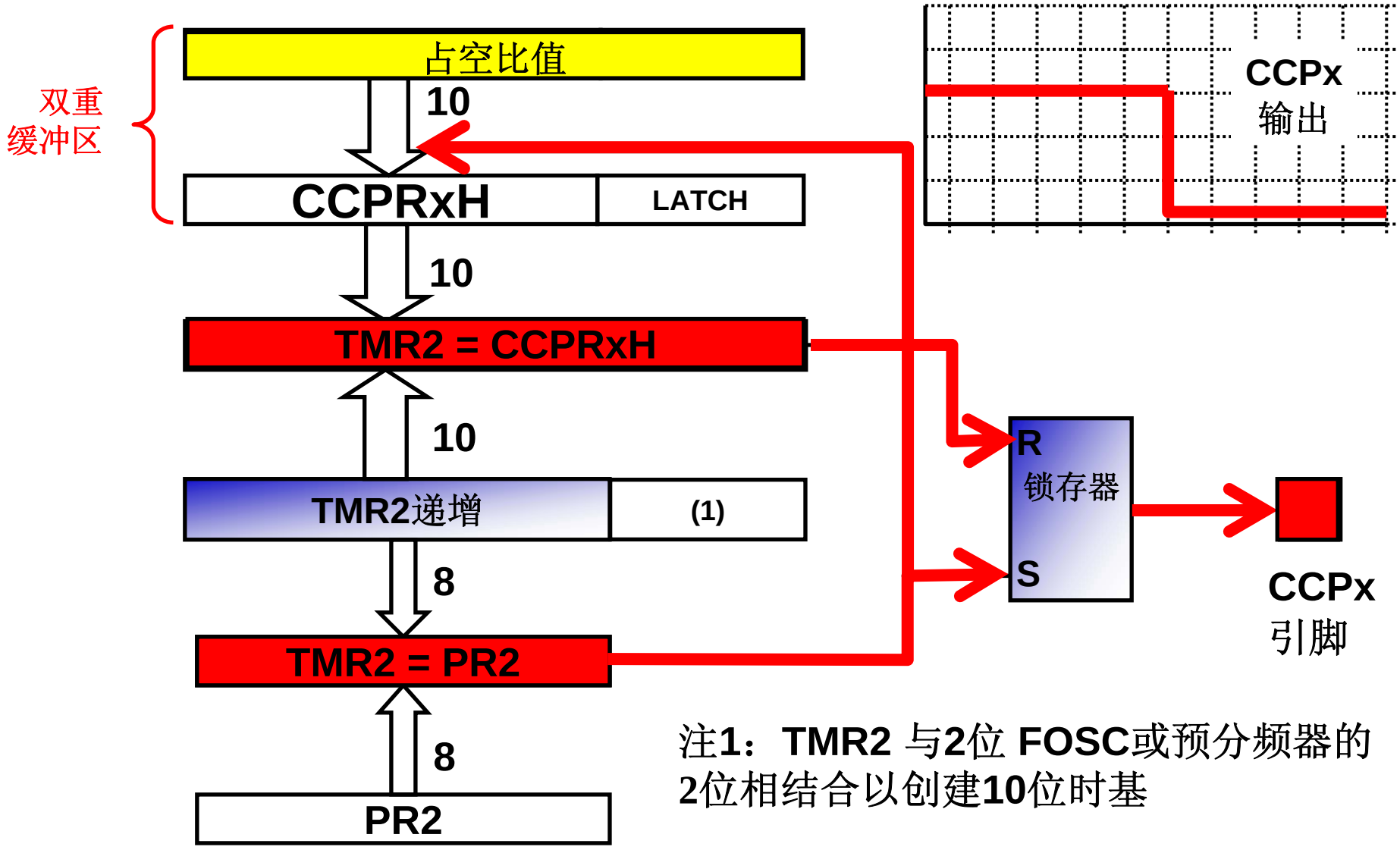
```
;Turn off the CCP module
banksel         CCP1CON
clrf            CCP1CON
;Turn off Timer1
bcf             T1CON, TMR1ON
;Clear Timer1 result registers
clrf            TMR1H
clrf            TMR1L
;Disable CCP1 interrupt and make sure
;its flag is clear
banksel         PIE1
bcf             PIE1, CCP1IE
banksel         PIR1
bcf             PIR1, CCP1IF
;Make CCP1 pin output
banksel         TRISC
bcf             TRISC, 2
;Initialize Compare to set output on match
banksel         CCP1CON
movlw           b'00001000'
movwf           CCP1CON
;Load half of Timer1 full scale value into
;CCPR1H:CCPR1L
banksel         CCPR1H
movlw           b'10000000'
movwf           CCPR1H
clrf            CCPR1L
;Start Timer1 incrementing
bsf             T1CON, TMR1ON
;Test CCP1IF for Timer1 match with CCPRx
btfss           PIR1, CCP1IF
goto            $-1
```



PWM模式

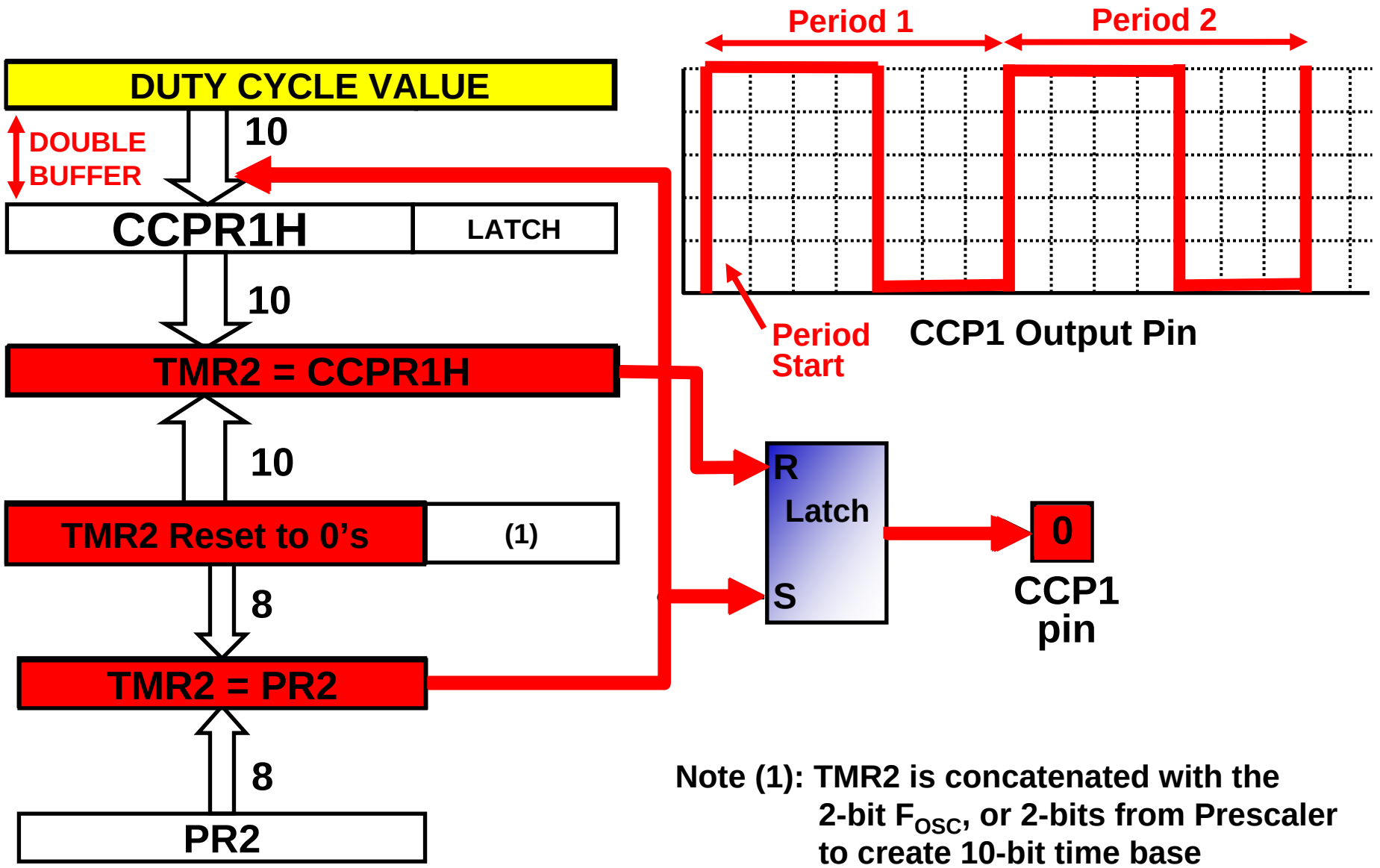
- 在CCPx 引脚上产生脉宽调制（ PWM ）信号
- 由以下寄存器决定占空比、周期和分辨率
  - PR2
  - T2CON
  - CCPRxL
  - CCPxCON

PWM框图



注1：TMR2 与2位 FOSC或预分频器的2位相结合以创建10位时基

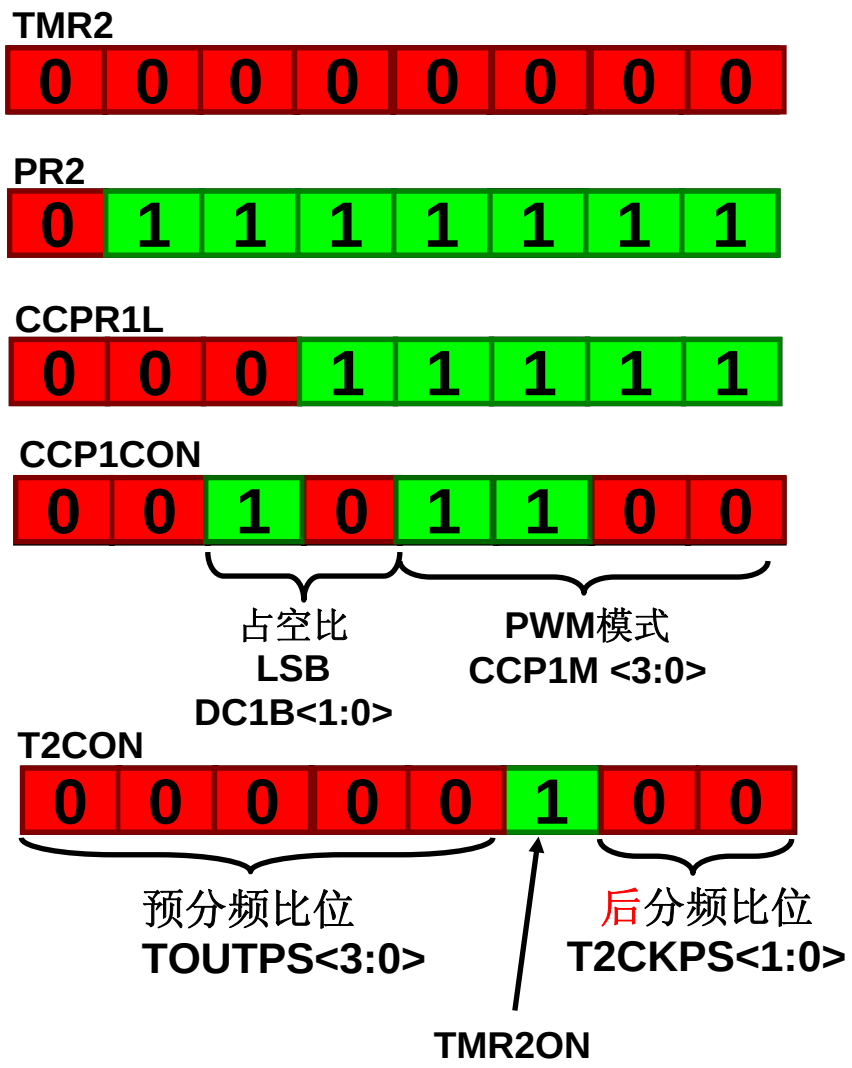
PWM Block Diagram



Note (1): TMR2 is concatenated with the 2-bit F_{OSC}, or 2-bits from Prescaler to create 10-bit time base

PWM操作的设置

```
;Turn off CCPx pin by setting TRISC bit HIGH
banksel     TRISC
bsf         TRISC, 2

;Clear Timer2
banksel     TMR2
clrf        TMR2

;Set up Period and Duty Cycle using an 8MHz oscillator
movlw       b'01111111'         ;
movwf       PR2                 ;Load a 64uS Period Value
movlw       b'00011111'         ;
movwf       CCPR1L              ;Load Duty Cycle Value
                                ; (25%) of PWM period

;Configure ECCP module for single PWM
; with P1A active HIGH and
;LSB's of Duty Cycle are '10'
movlw       b'00101100'         ;
movwf       CCP1CON             ;ECCP module is configured
                                ;for PWM and Duty Cycle
                                ;LSB's loaded
;Turn CCPx pin back on
banksel     TRISC
bcf         TRISC, 2            ;Make CCP1 output

;Timer2 starts when TMR2ON is set beginning PWM
movlw       b'00000100'         ;incrementing
movwf       T2CON               ;Prescaler and Postscaler
                                ;are both 1:1
```



脉宽调制（PWM）实验

Microchip Technology Inc.
RTC
Regional Training Centers

PWM实验

- 本实验的目标是熟悉ECCP模块的配置和PWM模式下的操作

以及

- 了解有关Timer2配置的更多信息

PWM实验概述

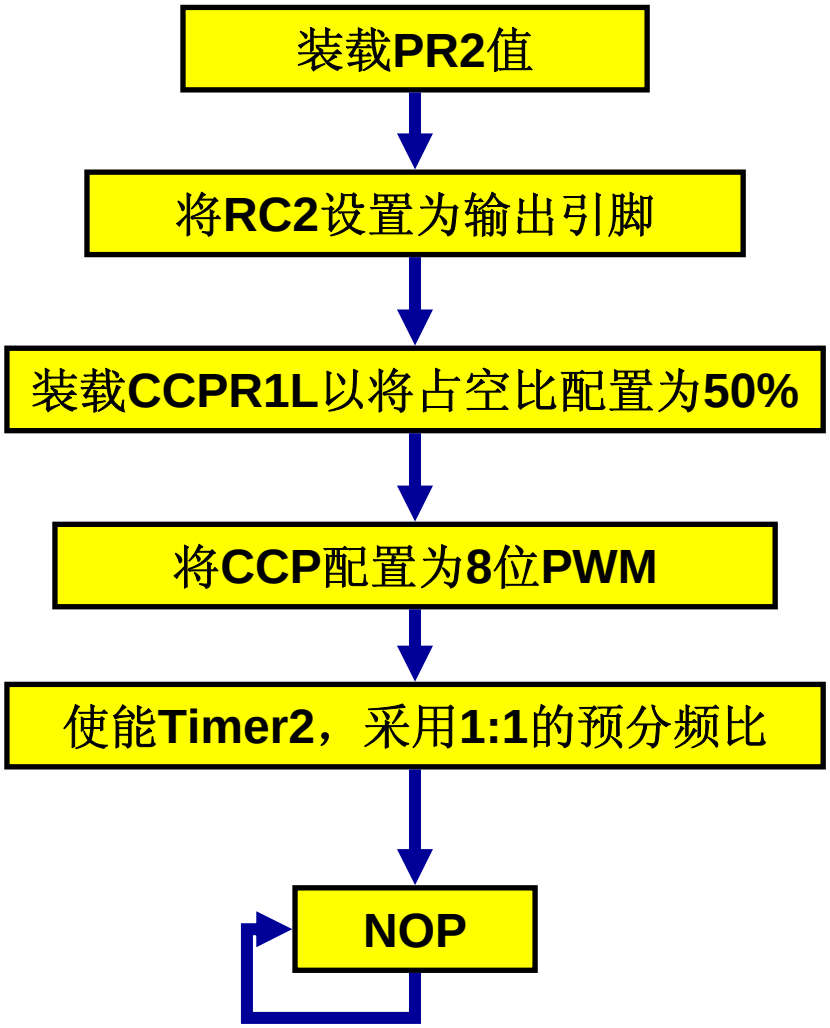
- PWM波形从CCP1引脚（RC2）输出， 使得PICdem2 plus板上的蜂鸣器发出一个声音。

- 完成实验后，周期为256/（Fosc/4）占空比为50%的信号将驱动此蜂鸣器

Microchip Technology Inc.
RTC
Regional Training Centers

PWM实验概述

主代码



实验细节

- 本实验代码位于
  C:\RTC\201_ASP\Lab4-PWM

- 完成以下部分
  - 将PORTC的引脚2 （CCP1）配置为输出引脚
  - 设置CCP工作在PWM模式
  - 清零DCB1和DCB0（8位PWM）
  - 将Timer2的预分频比值设置为1:1

需要了解的内容

- 提供了装载PR2 （Timer2）和设置50%占空比的代码。可在该代码中找到具体的值
- 在PIC16F877中，CCP1引脚为RC2（ PORTC 的引脚2）
- 完成本实验所需的寄存器为：TRISC、T2CON和CCP1CON

Microchip Technology Inc.
RTC
Regional Training Centers

实验解决方案

```
;*****************************************************************
;       Set CCPx  as an output
;*****************************************************************
        bcf             TRISC,2         ; set CCP1 pin as output pin
;
;       set duty cycle for 50%
;
        bcf             STATUS,RP0      ; go to bank 0
        movlw           0x80
        movwf           CCPR1L          ; set duty cycle
;*****************************************************************
;               Put CCP1 module in PWM mode.
;               Configure CCP to clear DCB1 and DCB0 ( 8-bit PWM)
;*****************************************************************
        movlw           0x0C
        movwf           CCP1CON

;*****************************************************************
;   Configure Timer2   Pre and post scale of 1:1
;   and turn Timer2 on
;*****************************************************************
        bsf             T2CON,TMR2ON    ; turn on Timer2
```

实验问答

问： 为什么我们不必允许中断来使PWM工作呢？

答：PWM可与PIC MCU同时运行，而不会降低处理器的速度

输出比较实验

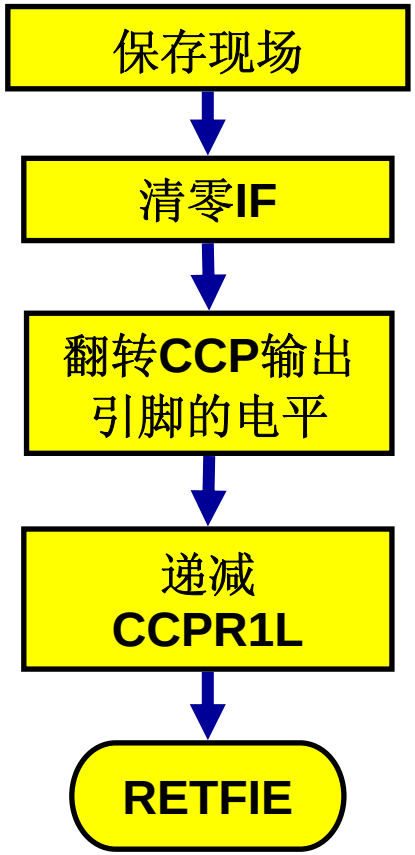
Microchip Technology Inc.
RTC
Regional Training Centers

输出比较实验

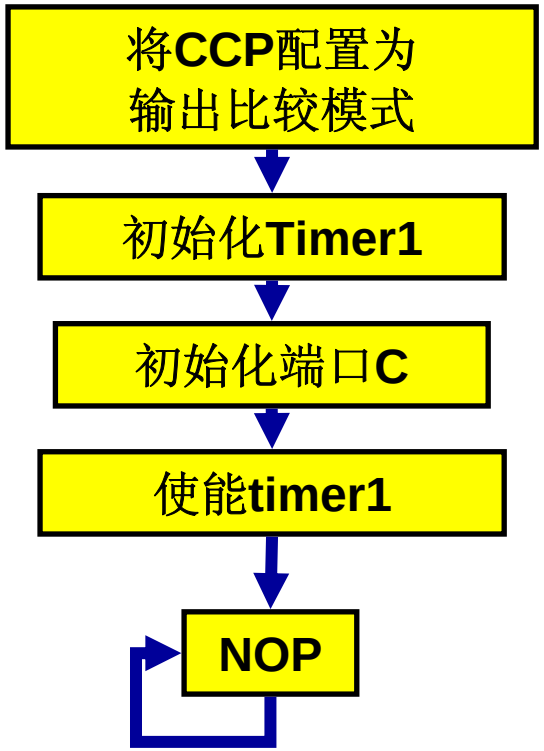
- 本实验的目标是获取以下经验：
  - 设置ECCP工作在输出比较模式
  - 配置特殊事件标志以复位Timer1
  - 配置ECCP 以在Timer1溢出时产生中断
  - 使用中断向量来修改中断间的时间间隔

实验概述

中断向量



主程序



CCP实验细节

- 本实验代码位于
  C:\RTC\201_ASP\Lab5-CCP

- 完成以下部分：
  - 将CCP配置为输出比较模式，比较匹配时将置1特殊事件标志位和CCP1IF
  - 配置Timer1，将其时钟源设置为Fosc/4，预分频比为1:8
  - 使能允许CCP中断发生所需的所有特殊功能寄存器

Microchip Technology Inc.
RTC
Regional Training Centers

需要了解的内容

- 完成本实验需要的寄存器为：INTCON、T1CON和CCP1CON。
- 提供了中断向量
- 值CCPR1L 将从0计满到0xFF，然后递减

Microchip Technology Inc.
RTC
Regional Training Centers

CCP实验解决方案

```
;
;       Set CCP1COM to Output Compare mode with Special Event Trigger
;       to clear the Timer 1 register pair on a match
;*********************************************************************
        movlw           0x0B
        movwf           CCP1CON             ; set value in CCP1CON
;
;       Configure Timer 1 for Fosc/4 operation.  8:1 Pre-scaler
;
;*********************************************************************
        movlw           0x30
        movwf           T1CON
;
;
;
;
; Enable Timer 1 interrupts, Peripheral Interrupts and Global Interrupts
;
;*********************************************************************
        bsf             PIE1,CCP1IE
        bsf             INTCON,GIE
        bsf             INTCON,PEIE
```

实验问答

问：PWM工作不需要中断。那么在输出比较模式下是否需要中断？

答： 不必

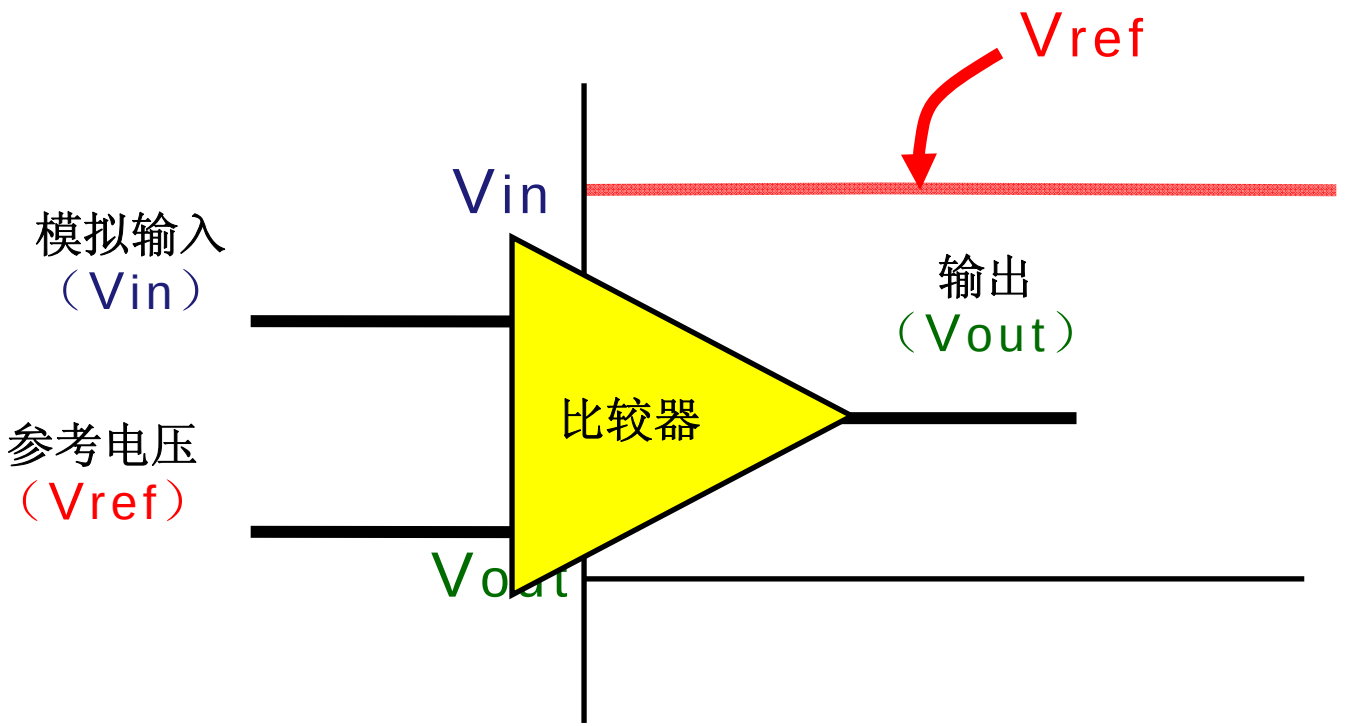
- 我们在本实验中确实使用了中断，那是因为我们在输出比较模式下的任务不在该问题所讨论的范围内

比较器

Microchip Technology Inc.
RTC
Regional Training Centers

比较器概述

- 比较器模块：
  - 将模拟输入电压与参考电压做比较，并输出数字电平结果
  - **PIC16F887有2个比较器 （C1和C2）**

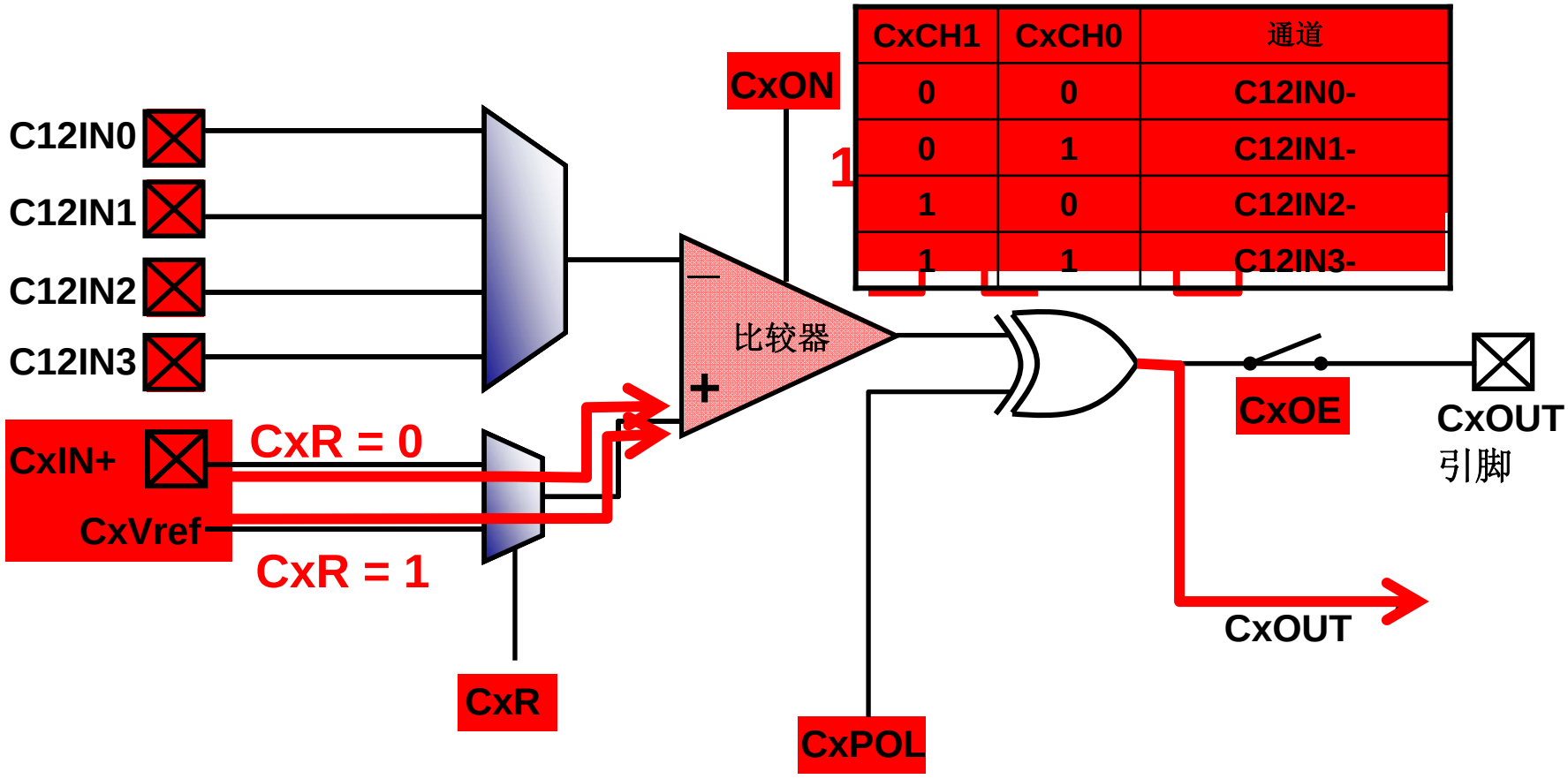


Microchip Technology Inc.
RTC
Regional Training Centers

比较器模块寄存器

- 各比较器（**C1**和**C2**）都有自己的控制寄存器
  - **CM1CON0**和**CM2CON0**
  - 比较器 **2** 还有一个**CM2CON1**，用于与**Timer1**交互
  - **CMxCON0** 寄存器控制以下操作
    - 使能
    - 输入选择
    - 参考电压选择
    - 输出选择
    - 输出极性

比较器简化框图



CxON	CxOUT	CxOE	CxPOL	---	CxR	CxCH1	CxCH2

Microchip Technology Inc.
RTC
Regional Training Centers

比较器模块寄存器CM2CON1

MC1OUT	MC2OUT	C1RSEL	C2RSEL	---	---	T1GSS	C2SYNC

位	功能
MC1OUT	C1OUT位的镜像拷贝
MC2OUT	C2OUT位的镜像拷贝
C1RSEL	1 = CVREF 连接到比较器C1的C1VREF输入端 0 = 0.6V绝对参考电压连接到C1VREF
C2RSEL	1 = CVREF连接到比较器C2的C2VREF输入端 0 = 0.6V绝对参考电压连接到C2VREF
T1GSS	0 = 在比较器输出时，Timer1 递增， 1 = 定时器门控信号随着外部引脚上的时钟源递增
C2SYNC	1 = 比较器 2输出与Timer1时钟的下降沿同步 0 = 比较器输出异步

同时读两个比较器

比较器中断

- 比较器中断标志（**PIR2**寄存器中的**C1IF/C2IF**）
  - 由相关比较器的任何变化置**1**
  - **PIE2**中的**C1IE/C2IE**以及**INTCON**中的**PEIE**和**GIE**必须置**1**
  - 中断标志必须用软件清零，以允许其他比较器中断
    - 必须读比较器以在清零**C1IF**或**C2IF** 之前清零此中断标志。

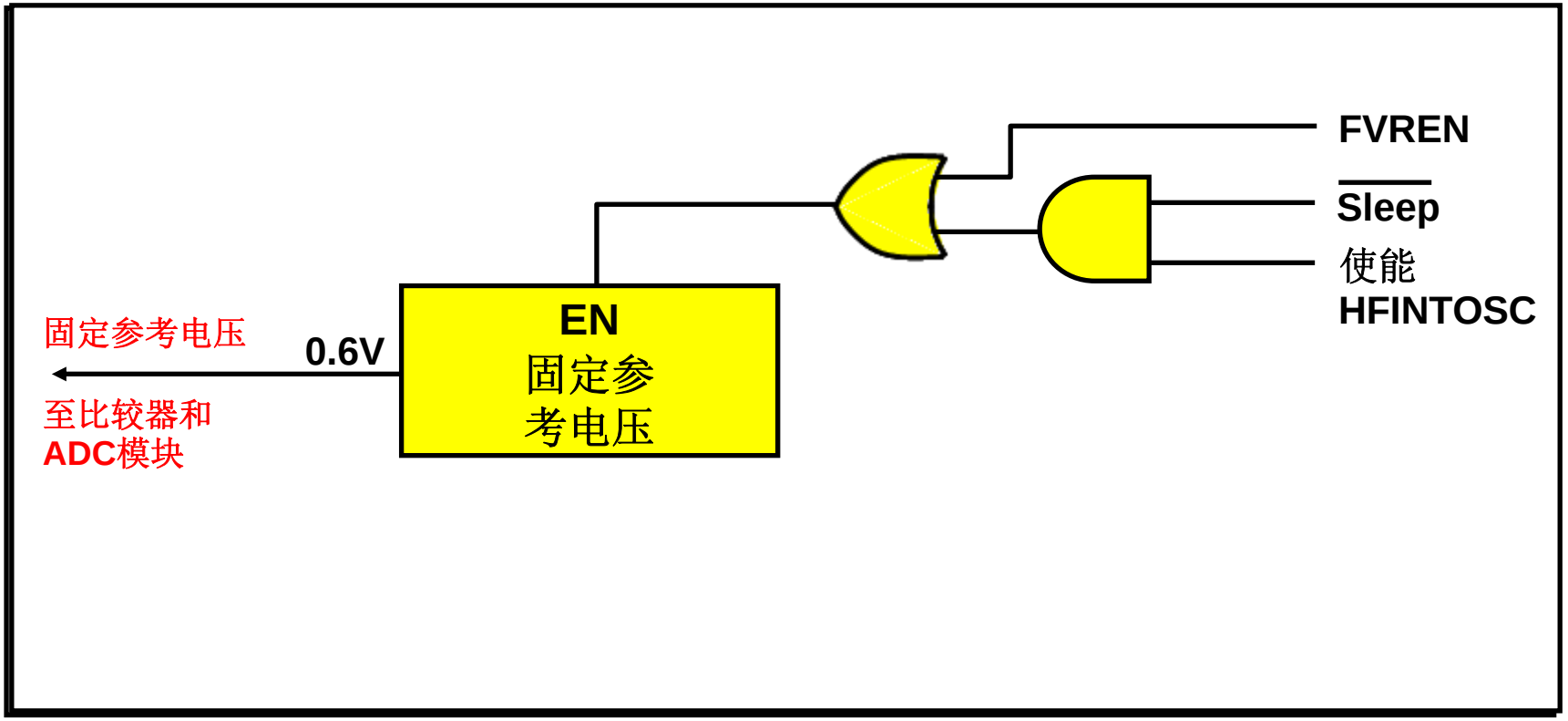
Microchip Technology Inc.
RTC
Regional Training Centers

比较器和休眠模式

- 如果在执行**SLEEP**指令之前使能了比较器，比较器将在休眠模式下继续工作
- 比较器输出电平的变化将唤醒**PIC**单片机
- **C1IE/C2IE（PIE2）和PEIE（INTCON）必须置1**
  - **SLEEP**指令后的下一条指令在唤醒时或**ISR**（允许中断）中执行
  - 将**GIE （INTCON）位置1转而执行ISR**
- 任何复位将关闭比较器，使得所有寄存器返回到默认状态

比较器参考电压

- 参考电压模块：



参考电压控制寄存器（VRCON）

VREN	VROE	VRR	VRSS	VR3	VR2	VR1	VR0

若VRR = 1或选择了低电压范围：

CVref = (VR<3:0>/24) x Vdd

或

若VRR = 0 或选择了高电压范围：

CVref = Vdd/4 + (VR<3:0>/32) x Vdd

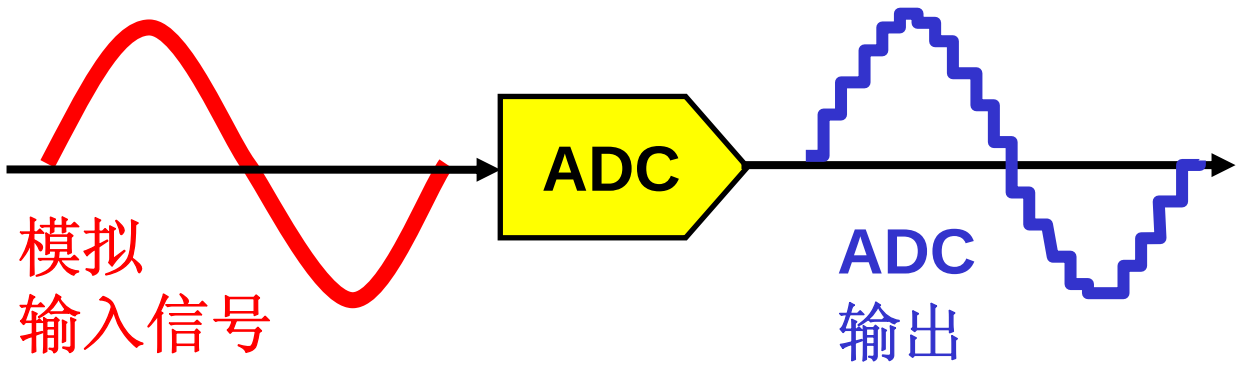
模数转换器（ADC）

Microchip Technology Inc.
RTC
Regional Training Centers

Microchip Technology Inc.
RTC
Regional Training Centers

ADC概述

- 模数转换器模块
  - 将模拟输入信号转换为8或10位二进制值
  - 可选的内部或外部参考电压
  - 转换完成后可以产生中断
  - 中断可用于将PIC单片机从休眠模式唤醒



Microchip Technology Inc.
RTC
Regional Training Centers

ADC寄存器

- **ADC实现两个控制寄存器**
  - **ADCON0和ADCON1**

ADCON0

ADCS1	ADCS2	CHS3	CHS2	CHS1	CHS0	GO/$\overline{\text{DONE}}$	ADON

位	功能
ADCSx位	A/D转换时钟选择位 00 = Fosc/2，01 = Fosc/8，10 = Fosc/32， 11 = FRC （内部振荡器）
CHSx位	模拟通道选择位
GO/$\overline{\text{DONE}}$	1 = A/D转换正在进行 0 = A/D转换完成
ADON	使能ADC模块

ADC寄存器

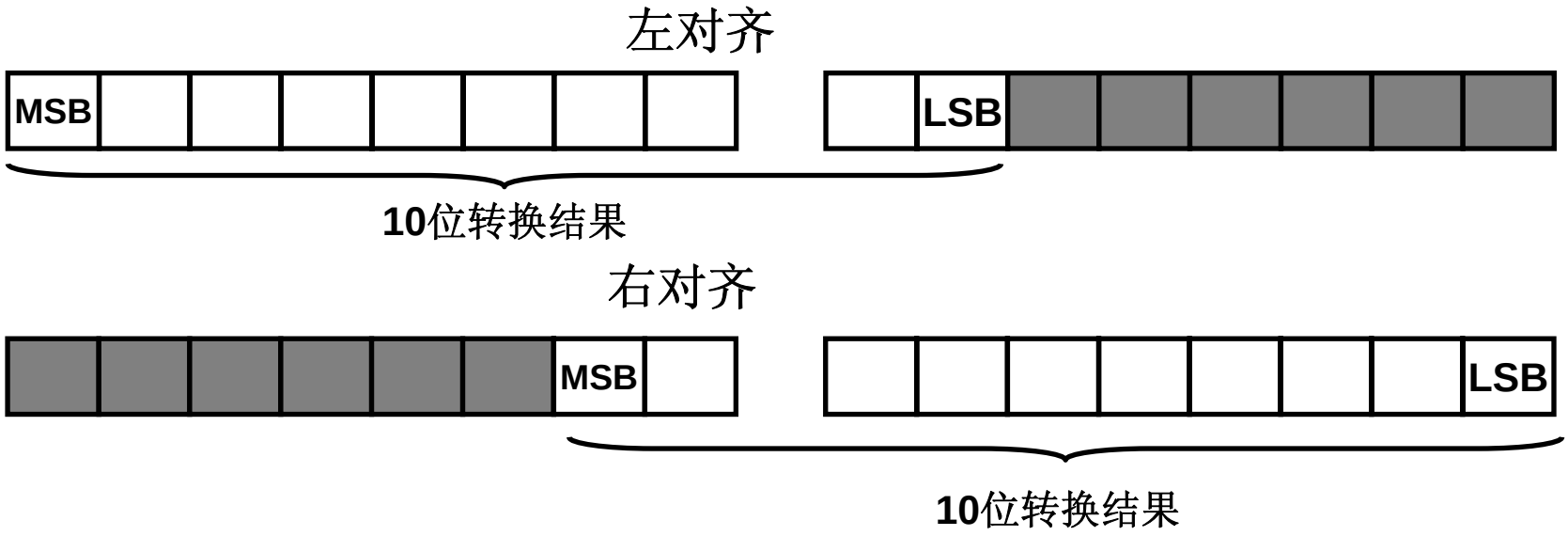
ADCON1

ADFM	---	VCFG1	VCFG0	---	---	---	---

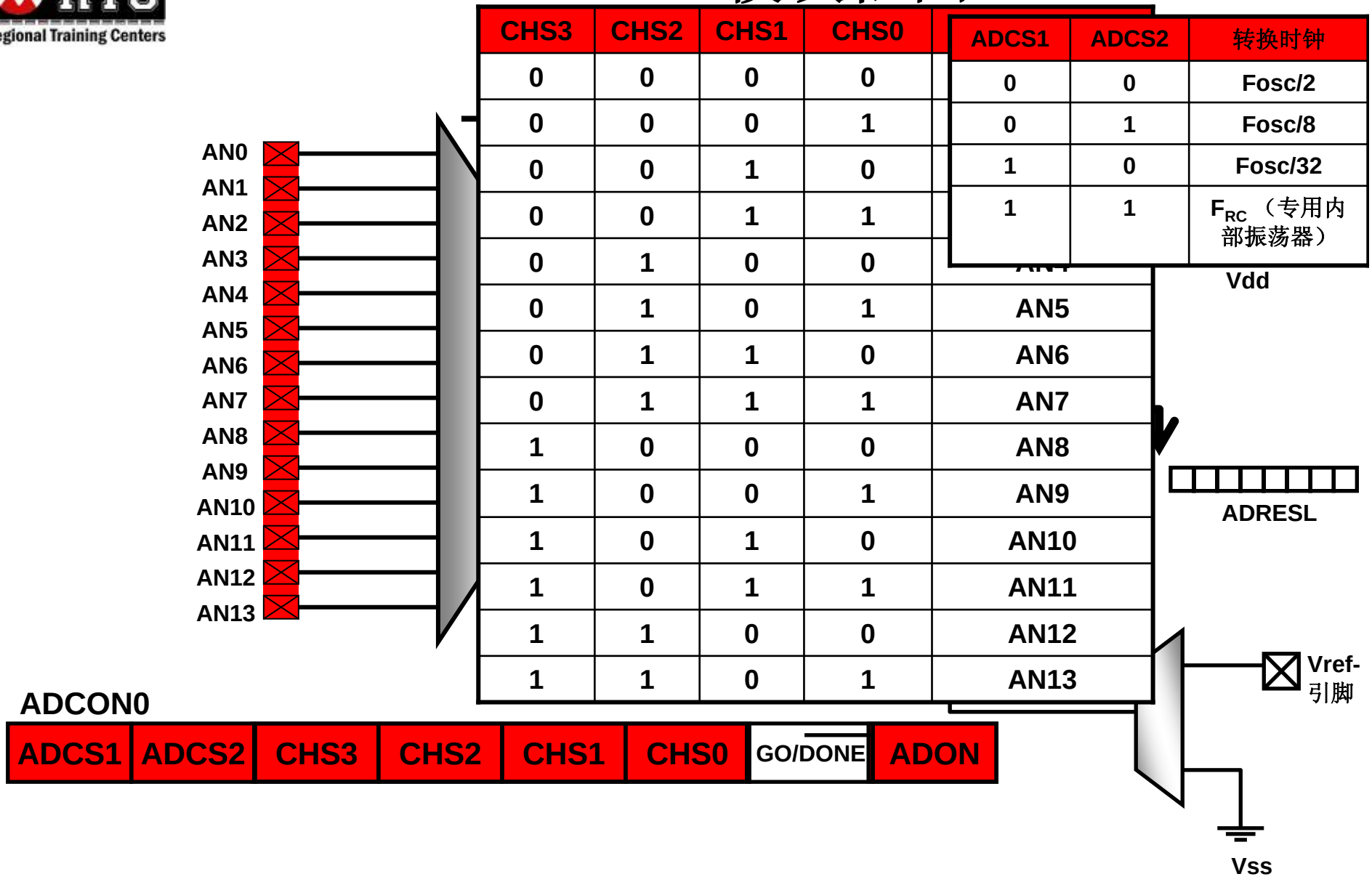
位	功能
ADFM	结果寄存器对齐方式位 **1 =** 右对齐， **0 =** 左对齐
VCFG1	负参考电压 **1 =** 来自**Vref-**引脚的外部电压源， **0 = Vss**
VCFG0	正参考电压 **1 =** 来自**Vref+**引脚的外部电压源， **0 = Vdd**

ADC寄存器

- 转换完成后，**ADC**转换结果被放到两个结果寄存器 **ADRESH**和**ADRESL**中
- **10**位**ADC**转换结果可以左对齐也可右对齐



A/D模块框图



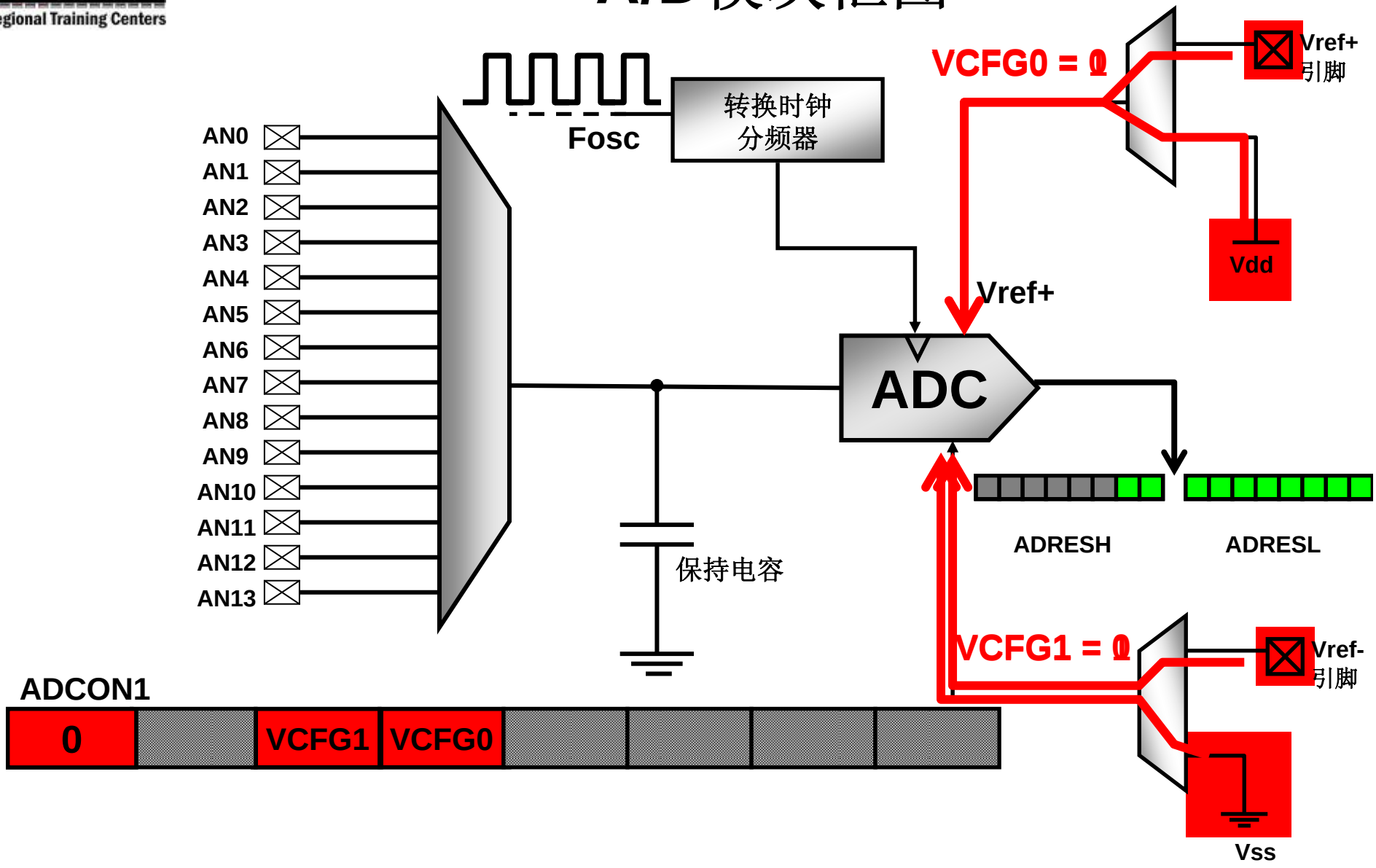
CHS3	CHS2	CHS1	CHS0	
0	0	0	0	
0	0	0	1	
0	0	1	0	
0	0	1	1	
0	1	0	0	
0	1	0	1	AN5
0	1	1	0	AN6
0	1	1	1	AN7
1	0	0	0	AN8
1	0	0	1	AN9
1	0	1	0	AN10
1	0	1	1	AN11
1	1	0	0	AN12
1	1	0	1	AN13

ADCS1	ADCS2	转换时钟
0	0	Fosc/2
0	1	Fosc/8
1	0	Fosc/32
1	1	F_{RC}（专用内部振荡器）

ADCON0

ADCS1	ADCS2	CHS3	CHS2	CHS1	CHS0	GO/DONE	ADON

A/D模块框图



Microchip Technology Inc.
RTC
Regional Training Centers

ADC时序注意事项

- 选择了一个模数转换通道时，必须花费一定的时间使保持电容充电
- 所有**10**位转换的完成需要**11**个周期
- 用户必须根据系统时钟频率选择适当的**ADC**时钟

模数转换实验

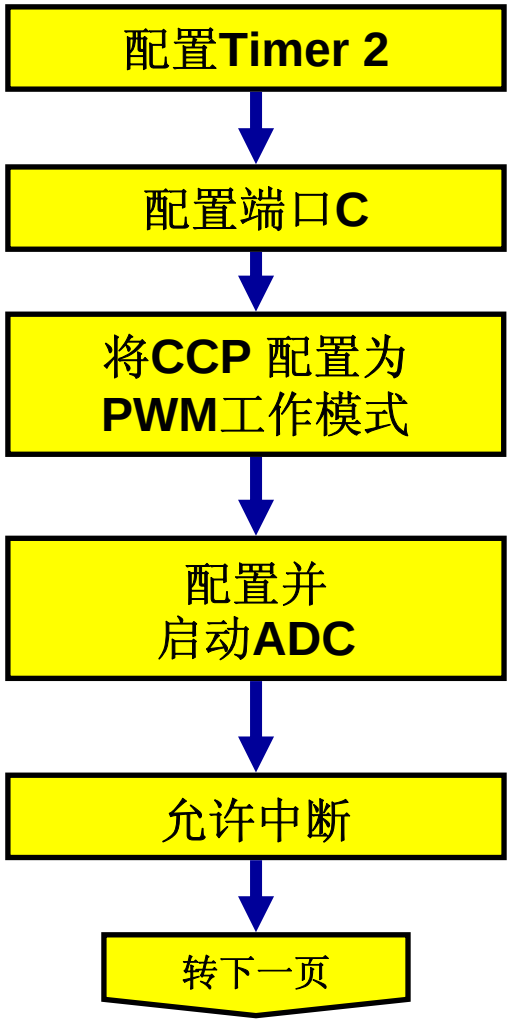
Microchip Technology Inc.
RTC
Regional Training Centers

模数转换器实验

- 本实验将使您熟悉以下操作：
  - 设置ADC模块
  - 从“主”程序而非中断向量对外设进行操作
  - 使用从一个外设（ ADC ）读到的值来驱动另一个外设（ PWM 模式下的ECCP ）

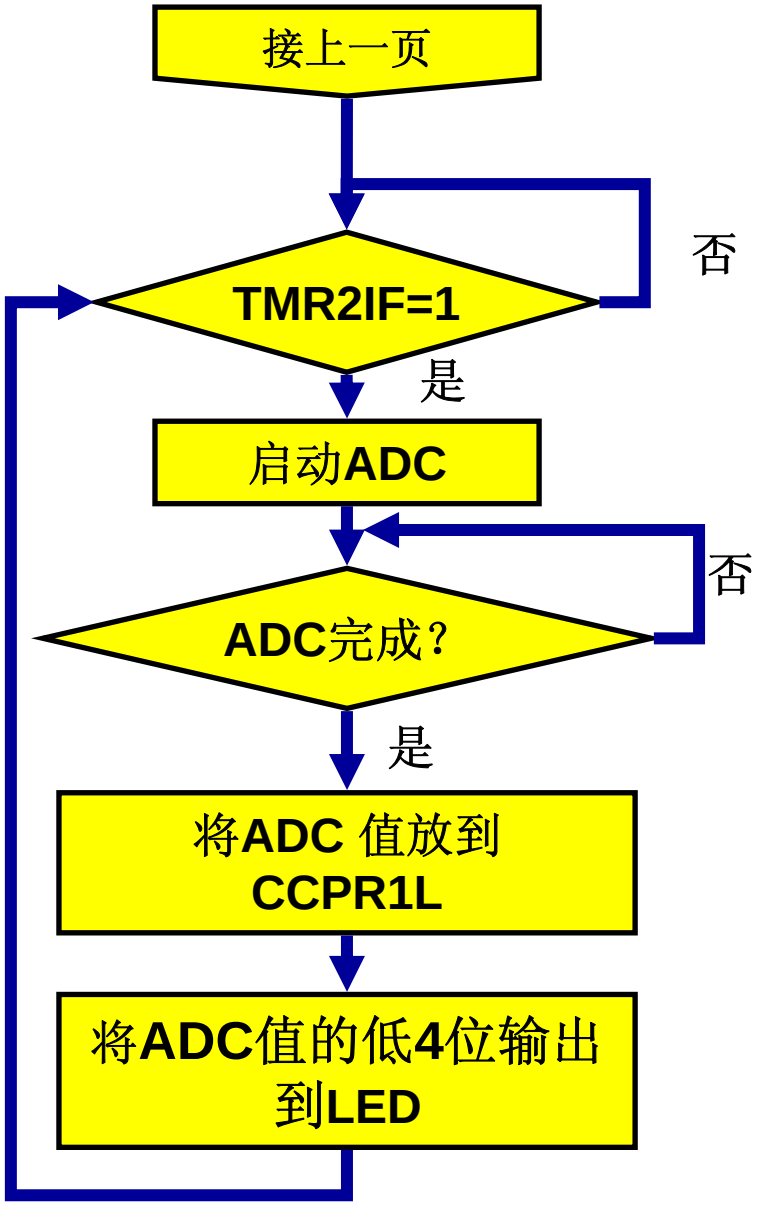
ADC实验概述

主程序



实验概述 （续）

主循环



实验细节

- 完成项目C:\RTC\201_ASP\Lab6-ADC中代码的以下部分
  - 配置ADC以将左对齐的值返回到 ADRESH
  - 将Tad设置为Tosc*8
  - 启动ADC单元
  - 完成此代码以启动ADC，并在主控制循环中等待转换结束

Microchip Technology Inc.
RTC
Regional Training Centers

需要了解的内容

- 本实验不在中断子程序中执行ADC转换，而是使用查询方法。
- 将ADC转换的结果值写入CCPR1L将更改蜂鸣器的占空比
- 使用ADCON1和ADCON0 特殊功能寄存器完成本实验

Microchip Technology Inc.
RTC
Regional Training Centers

ADC解决方案

```
;*******************************************************************************
;       Configure ADC ,  Channel 0, left justified, Tad=8 * Tosc, turn on ADC
;*******************************************************************************
        clrf    ADCON0          ; esnure default Channel is set to channel 0
        bsf     ADCON0,ADCS1    ; set Tad = 8 Tosc
        bsf     ADCON0,ADON     ; turn on ADC unit
        bsf     STATUS,RP0      ; go to bank1
        movlw   0x0E            ; Left Justify and set configuration
        movwf   ADCON1
;
; Enable Timer 2 interrupts, Peripheral Interrupts and Global Interrupts
;

        bsf     PIE1,TMR2IE
        bsf     INTCON,GIE
        bsf     INTCON,PEIE
        bcf     STATUS,RP0      ; return to bank 0
loop
;
;*******************************************************************************
; add three lines of code to start the ADC conversion and wait for the conversion
; to complete
;*******************************************************************************
        bsf     ADCON0,GO       ; start A-to-D conversion on channel 0
        btfsc   ADCON0,GO       ; Is the conversion done?
        goto    $-1             ;   No: Check again
```

ADC实验问答

问： 不在主程序中等待TMR2IF置1，能否从中断子程序中启动ADC转换？

答：可以

增强型通用同步异步收发器（EUSART）

Microchip Technology Inc.
RTC
Regional Training Centers

Microchip Technology Inc.
RTC
Regional Training Centers

EUSART概述

- 串行I/O通信外设
  - 有时也称为串行通信接口或SCI
- 主要功能：
  - 可以同步或异步
  - 可以接收或发送
    - 全双工异步发送和接收
    - 半双工同步主模式和从模式
- 最常使用
  - RS-232与PC串行端口通信
    - 需要用于RS-232电平转换器的驱动器
- 增强型特性允许与局域互联网（ LIN ）总线系统接口

EUSART寄存器

- 有几个与**EUSART**结合使用的寄存器：
  - 波特率发生器寄存器：
    - **SPBRG**和**SPBRGH**
  - 发送状态和控制寄存器（**TXSTA**）
  - 接收状态和控制寄存器（**RCSTA**）
  - 接收和发送数据寄存器
    - 发送数据寄存器（**TXREG**）
    - 接收数据寄存器（**RCREG**）

Microchip Technology Inc.
RTC
Regional Training Centers

TXSTA寄存器

CSRC	TX9	TXEN	SYNC	SENB	BRGH	TRMT	TX9D

位	功能
CSRC	时钟源选择位 **1 =** 主模式（由内部 **BRG**产生时钟信号） **0 =** 从模式（由外部时钟源产生时钟信号）
TX9	第**9**位发送使能位
TXEN	**1 =** 发送使能
SYNC	**EUSART**模式， **1 =** 同步模式， **0 =** 异步模式
SENB	**1 =** 发送同步间隔字符位 **0 =** 同步间隔字符发送完成
BRGH	波特率选择位，**1 =** 高速， **0 =** 低速
TRMT	**1 =** 发送移位寄存器（**TSR**）为空， **0 = TSR**满 指示最后一位
TX9D	发送数据的第**9**位

Microchip Technology Inc.
RTC
Regional Training Centers

RCSTA寄存器

SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D

位	功能
SPEN	串行端口使能位 **1 =**使能串行端口（将**RX/DT**和**TX/CK**引脚配置为串行端口引脚） **0 =** 禁止串行端口（保持在复位状态）
RX9	**1 =** 使能**9**位数据接收， **0 = 8**位数据
SREN	同步模式 – 主模式， **1 =** 使能， **0 =** 禁止单字节接收
CREN	连续接收使能位
ADDEN	**1 =** 使能地址位检测 （ 允许中断并在**RSR<9>**置**1**时装载接收缓冲器）
FERR	**1 =** 发生帧错误（未检测到停止位）
OERR	**1 =** 发生溢出错误（装载其他数据时，**FIFO** 仍然为满）
RX9D	接收到数据的第**9**位

波特率控制寄存器（BAUDCTL）

ABDOVF	RCIDL	----	SCKP	BRG16	----	WUE	ABDEN

位	功能
ABDOVF	自动波特率检测溢出位（仅用于异步模式） **1 =** 自动波特率定时器溢出
RCIDL	接收器空闲标志位，**1 =** 接收器空闲， **0 =** 接收到了起始位，接收器正在接收
SCKP	同步时钟极性位 异步模式： **1 =** 将取反后的数据发送到**RB7/TX/CK**引脚 同步模式：**1 =** 在时钟的上升沿传输数据 **0 =** 在时钟的下降沿传输数据
BRG16	**16**位波特率发生器位 **1 =** 选择**16**位**BRG**，**0 =** 选择**8**位**BRG**
WUE	唤醒使能位（仅用于异步模式）
ABDEN	自动波特率检测使能位，**1 =** 使能 在休眠模式下，当第**9**位置**1**时，进行检测

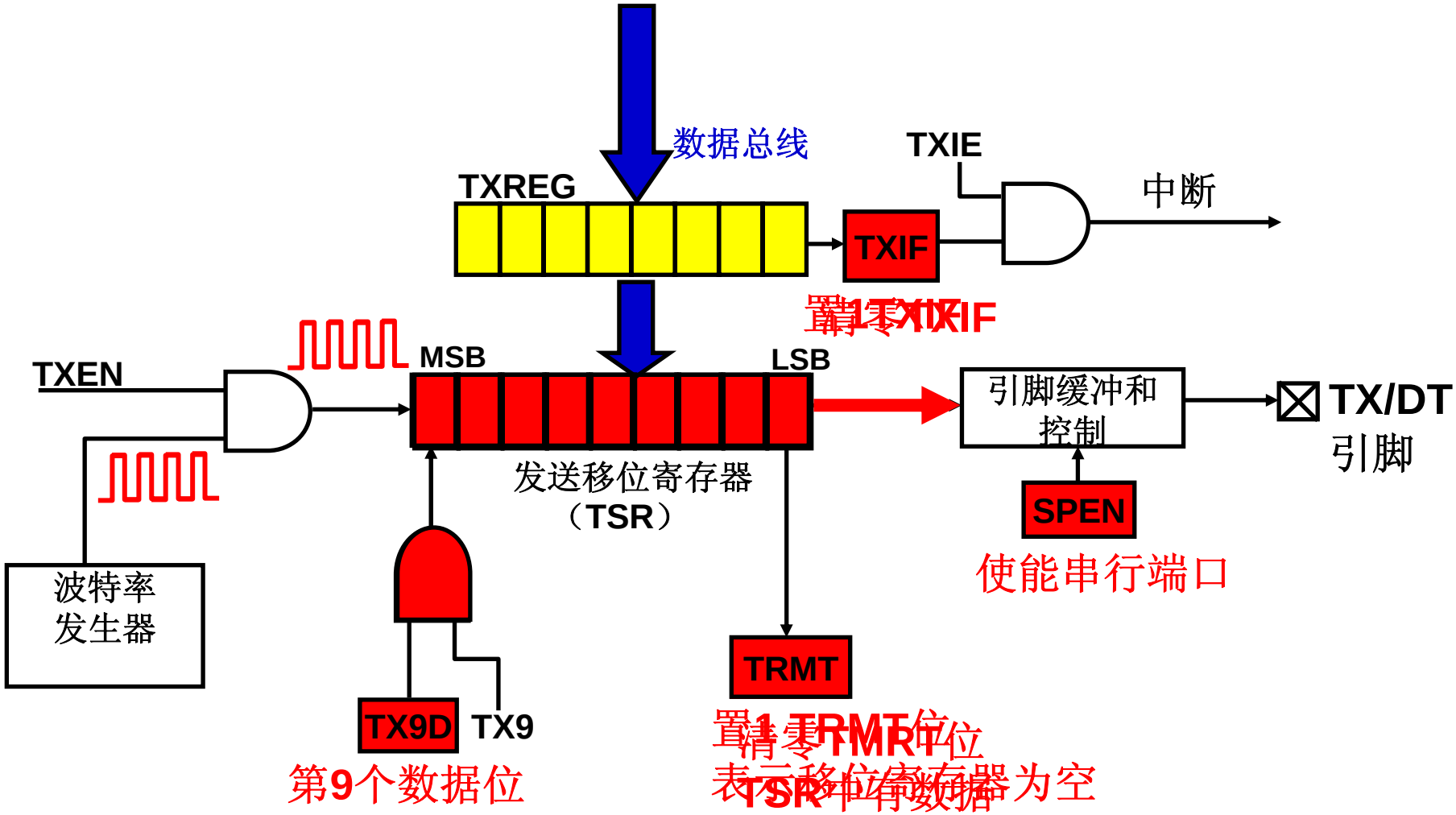
Microchip Technology Inc.
RTC
Regional Training Centers

波特率公式

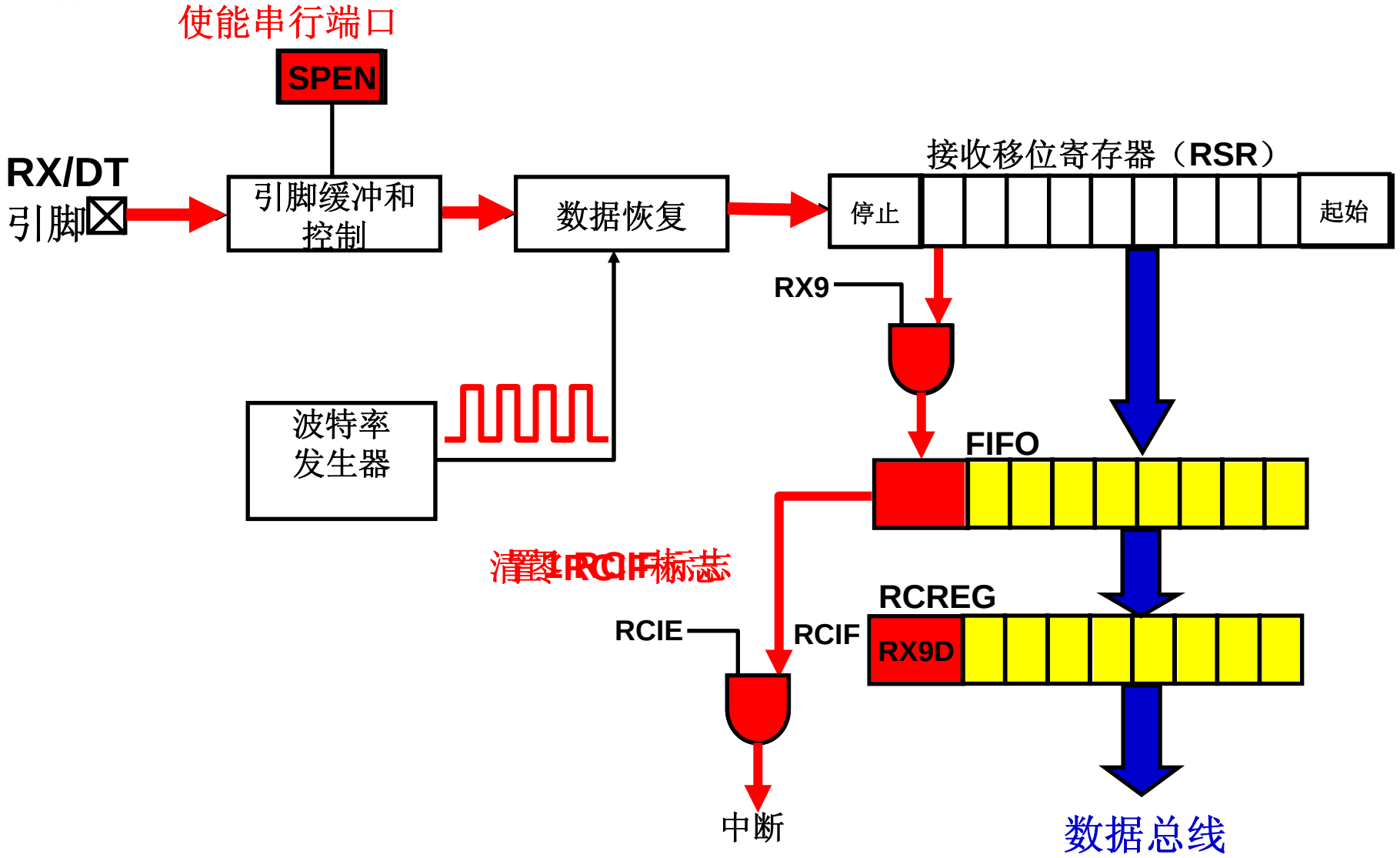
配置位			BRG/ EUSART 模式	波特率公式
SYNC（TXSTA）	BRG16（BAUDCTL）	BRGH（TXSTA）		
0	0	0	8位/异步	Fosc/[64 (n+1)]
0	0	1	8位/异步	Fosc/[16 (n+1)]
0	1	0	16位/异步	
0	1	1	16位/同步	Fosc/[4 (n+1)]
1	0	X	8位/同步	
1	1	X	16位/同步	

***n = SPBRGH:SPBRG寄存器对的值**

发送框图



接收框图



主同步串行端口（MSSP）模块

Microchip Technology Inc.
RTC
Regional Training Centers

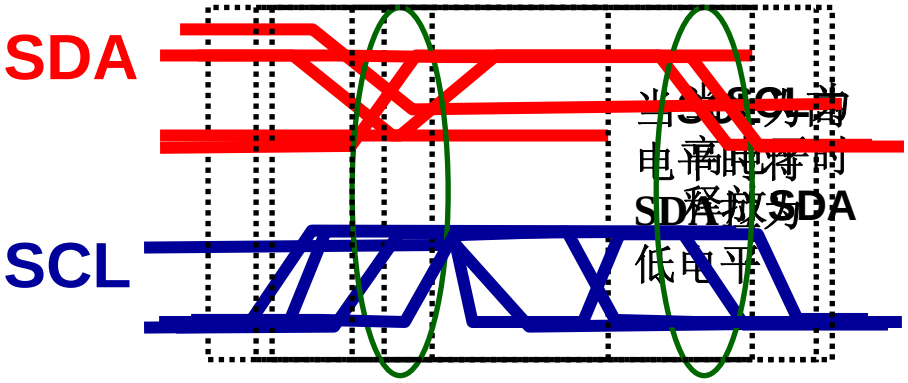
Microchip Technology Inc.
RTC
Regional Training Centers

概述

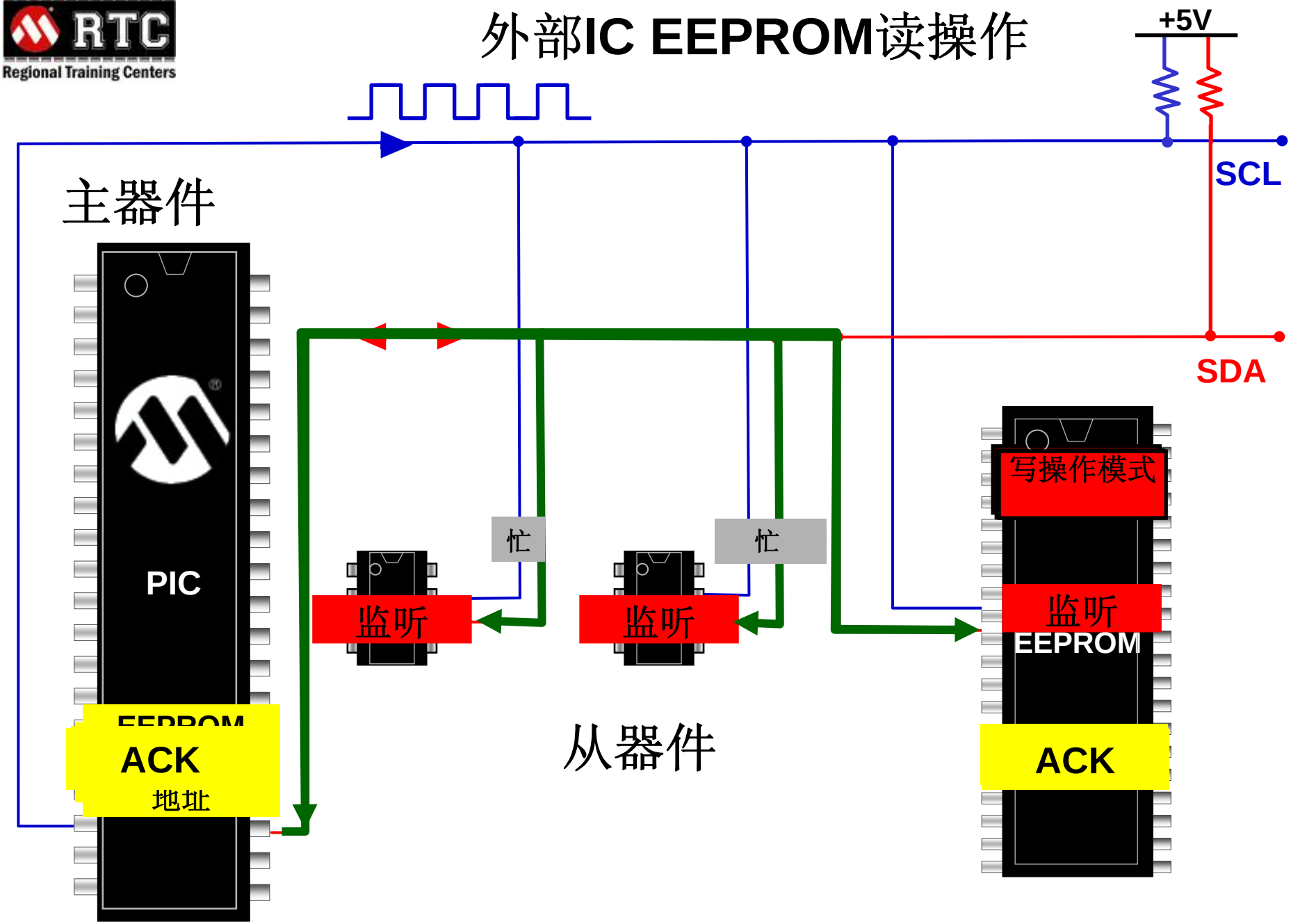
- MSSP模块有以下两种工作模式：
  - 串行外设接口（SPI）
  - I²C™
    - 完全主模式
    - 从模式（带有广播地址呼叫）
- I²C接口通过硬件支持以下模式：
  - 主模式
  - 多主机模式
  - 从模式

I²C条件

- 条件：
  - 启动（S）
  - 停止（P）
  - 应答（A）
  - 重复启动（R）
  - 否定或无应答（N）

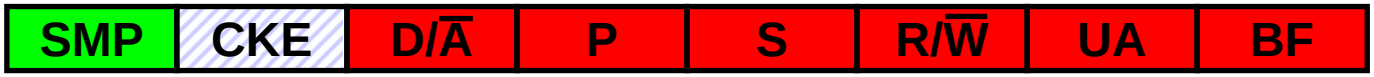


外部IC EEPROM读操作



MSSP控制寄存器

- 有3 个相关的控制寄存器
  1. **MSSP 状态寄存器（SSPSTAT）**



控制位　　检测位（标志）

Microchip Technology Inc.
RTC
Regional Training Centers

MSSP控制寄存器

SSPM3	SSPM2	SSPM1	SSPM0	模式
0	0	0	0	SPI主模式，时钟 = FOSC/4
0	0	0	1	SPI主模式，时钟 = FOSC/16
0	0	1	0	SPI主模式，时钟 = FOSC/64
0	0	1	1	SPI主模式，时钟 = TMR2输出/2
0	1	0	0	SPI从模式，时钟 = SCK引脚，使能SS引脚控制
0	1	0	1	SPI从模式，时钟 = SCK引脚，禁止SS引脚控制，SS可用作 I/O 引脚
0	1	1	0	I2C从模式，7位地址
0	1	1	1	I2C从模数， 10位地址
1	0	0	0	I2C主模式，时钟 = FOSC / (4 * (SSPADD+1))
1	0	0	1	保留
1	0	1	0	保留
1	0	1	1	I2C固件控制的主模式（从器件空闲）
1	1	0	0	保留
1	1	0	1	保留
1	1	1	0	I2C从模式，7位地址，并允许启动和停止位中断
1	1	1	1	I2C 从模式，10位地址，并允许启动和停止位中断

MSSP 控制寄存器

3. MSSP 控制寄存器2 （SSPCON2）

控制位　　检测位（标志）

Microchip Technology Inc.
RTC
Regional Training Centers

与I²C相关的寄存器

4. **MSSP接收/发送缓冲器（SSPBUF）**
   - 保存要发送的数据或MSSP模块接收到的数据
   - 缓冲器满时， SSPSTAT寄存器中的BF （缓冲器满）位置1
   - 在发送/接收数据期间，忽略任何写SSPBUF寄存器的操作，并且SSPCON寄存器中的写冲突检测位WCOL置1

Microchip Technology Inc.
RTC
Regional Training Centers

与I²C相关的寄存器

4. 串行端口地址（SSPADD）：

- 从模式：
  - 包含PIC器件的从地址
  - 与接收到的值进行比较
- 主模式：
  - 用于计算I²C系统的时钟速率（波特率）

$$\text{波特率} = \frac{Fosc}{4\times(SPADD+1)}$$

*注： Fosc是振荡器频率，而非指令时钟T_{CY}

Microchip Technology Inc.
RTC
Regional Training Centers

MSSP中断

- 发生以下事件时，**PIR1**寄存器中的**SSPIF**中断标志位会置**1**
  - 启动条件
  - 停止条件
  - 数据传输字节已发送/已接收
  - 应答发送
  - 重复启动条件

仅当使能了**PIE1** 寄存器中的**SSPIE** 位以及**INTCON** 中的**GIE**和**PEIE**位时，才会产生**SSP**中断。

多重中断实验

Microchip Technology Inc.
RTC
Regional Training Centers

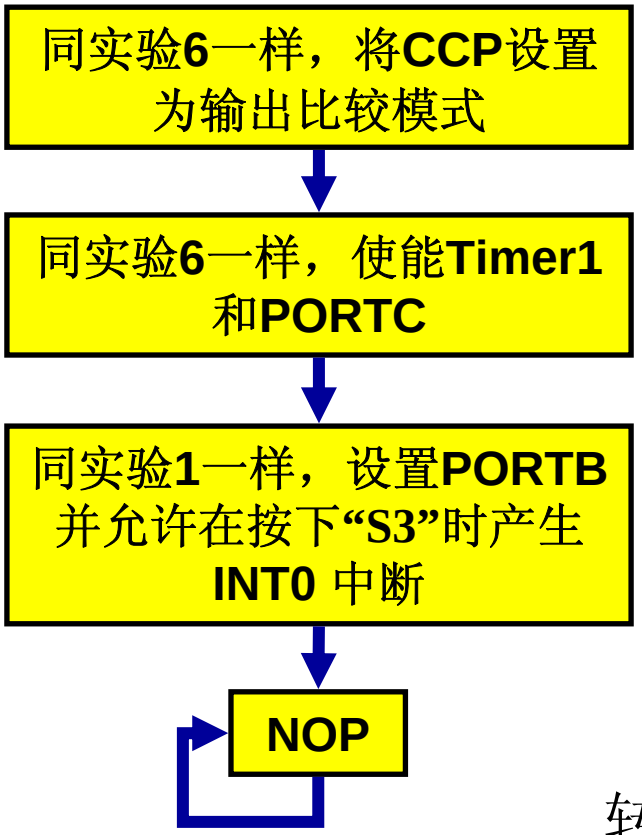
Microchip Technology Inc.
RTC
Regional Training Centers

多重中断实验

- 本实验包括：
  - 处理2个（或多个） 同时发生的中断
  - 确定中断源
  - 决定将要首先响应的中断请求

实验概述

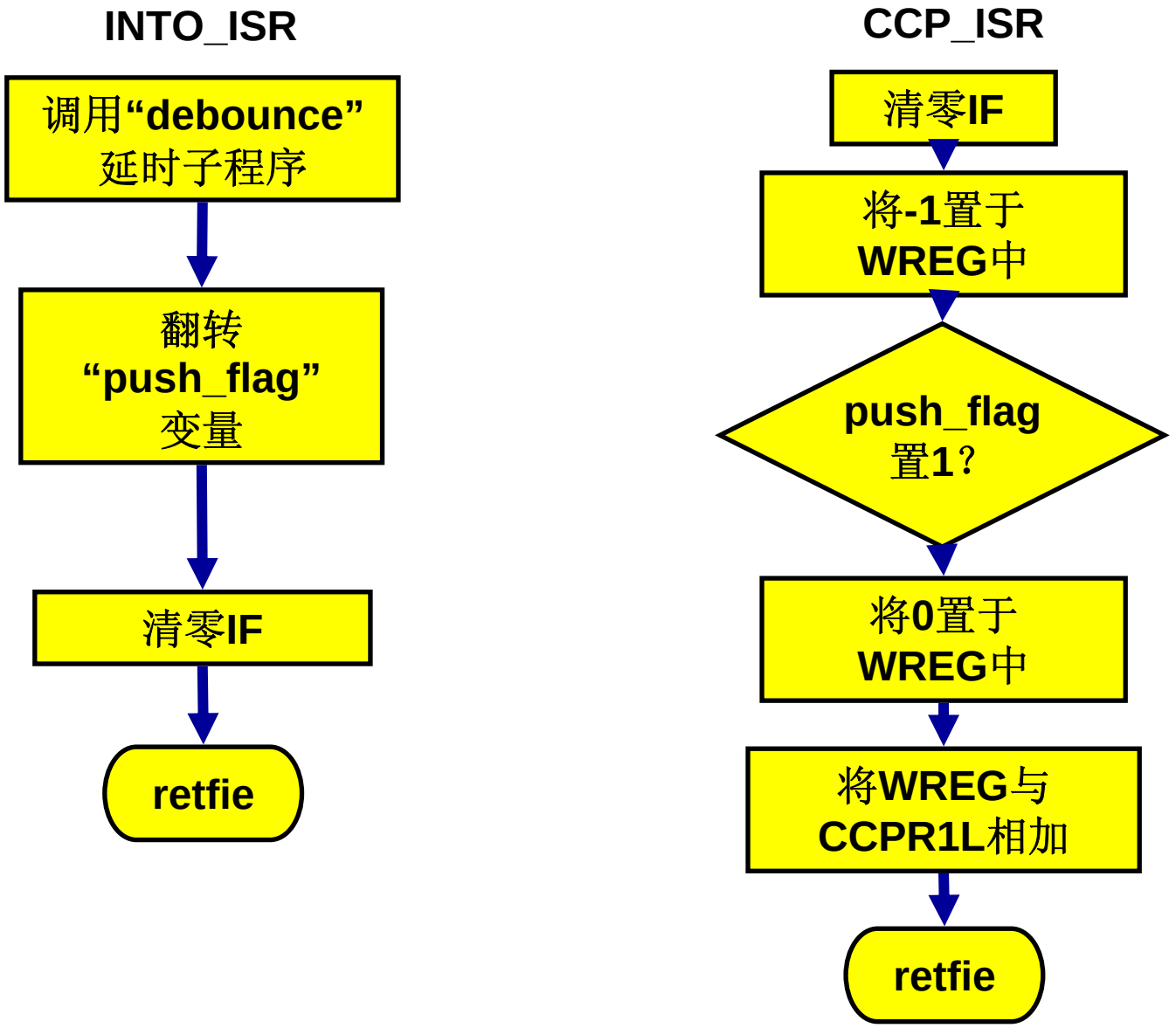
主程序



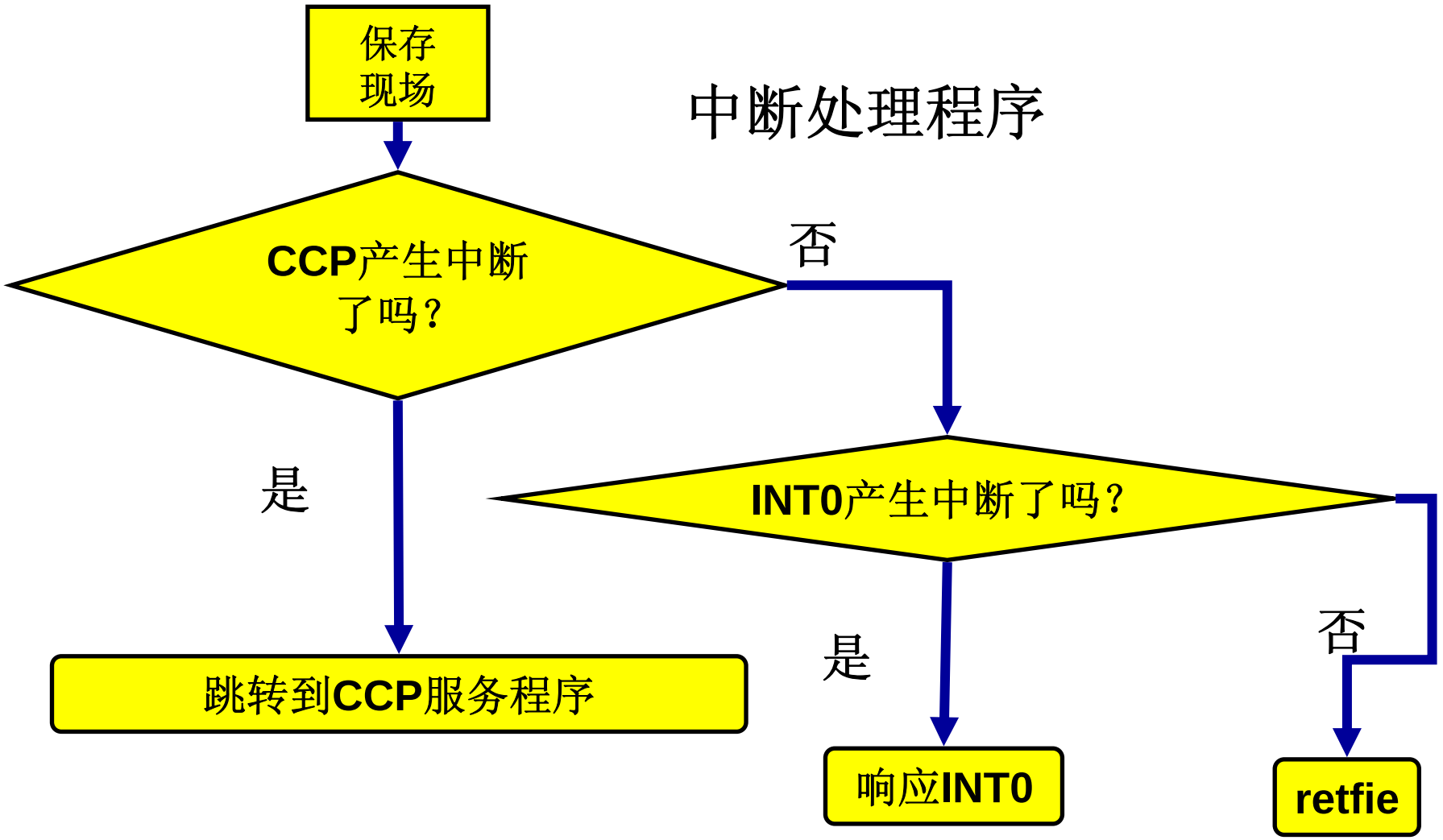
转下一张幻灯片

Microchip Technology Inc.
RTC
Regional Training Centers

实验概述



实验概述（续）



Microchip Technology Inc.
RTC
Regional Training Centers

实验细节

- 本实验位于：
  - C:\RTC\201_ASP\Lab8-MXINT

- 提供了两个中断服务程序（INT0_ISR和CCP_ISR）

- 完成代码的以下部分
  - 中断发生时，确定相应ISR的原因并进行中断请求的传递控制
  - 设置SFR以允许产生INT0和CCP1中断

本实验中需要了解的内容

- 本实验使用了INTCON和PIR 特殊功能寄存器

实验解决方案

向量解析部分

```
NT_VECTOR  CODE     0x004     ; interrupt vector location
;
;      Save Wreg, STATUS, and PCLATH during Interrupt Service
;
       call        save_regs;

       btfsc       INTCON,INTF       ; test for INT0 interrupt
       goto        INT0_ISR
       btfsc       PIR1,CCP1IF       ; test for CCP Interrupt request
       goto        CCP_ISR
;
```

Microchip Technology Inc.
RTC
Regional Training Centers

实验解决方案（续）

```
        call        save_regs           ; Save W, STATUS, and PCLATH
;
; Enable CCP1 interrupts, INT0, Peripheral Interrupts and Global Interrupts
;
        bsf         PIE1,CCP1IE
        bsf         INTCON,INTE
        bsf         INTCON,GIE
        bsf         INTCON,PEIE
        bcf         STATUS,RP0          ; return to bank0
;
        call        restore_regs
        retfie
```

Microchip Technology Inc.
RTC
Regional Training Centers

实验问答

问：为什么在按下S3时会非常安静？

答：因为在中断期间调用了“debounce”函数，并清零了GIE位，在这段时间内不允许执行翻转蜂鸣器的CCP1中断服务程序。所以蜂鸣器没有发出声音

实验问答（续）

问：如何消除静音，并使蜂鸣器继续运行？

答：

1. 在主程序中捕获S3按下事件，并在GIE置1时调用debounce
2. 使用定时器完成此延时
3. 在INT0中断服务程序中重新允许中断

实验问答（续）

问： 三个方法中哪个最好？

答： **依具体情况而定 !!**

- 每个实验都有自己的优点和缺点

201ASP总结

Microchip Technology Inc.
RTC
Regional Training Centers

Microchip Technology Inc.
RTC
Regional Training Centers

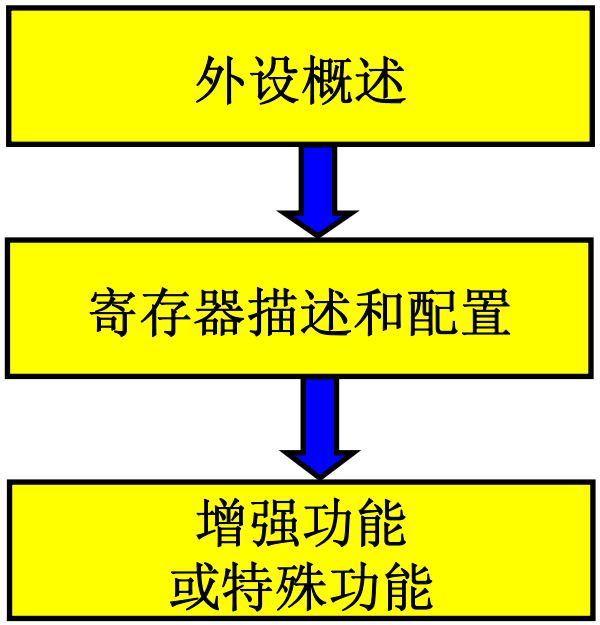
201ASP总结

- 现在的中档系列包含了下列外设
  - I/O端口
  - 中断结构和处理
  - 定时器（timer0、timer1和timer2）
  - ECCP模块（输出比较、输入捕捉和PWM）
  - 比较器和模数转换器
    - 参考电压
  - EUSART – 串行端口
  - 使用MSSP模块的I^2C和SPI

Microchip Technology Inc.
RTC
Regional Training Centers

总结

- 本次讨论遵循标准MCHP数据手册流程：



使用这部分内容：
- 开发逻辑流程图或伪代码
（避免复杂而混乱的编程!!）

其他提示：
- 详细注释代码
- 为用户定义的寄存器选择描述性的名称

*封装和电气规范在数据手册的末尾

Microchip Technology Inc.
RTC
Regional Training Centers

资源

- 访问www.microchip.com网站并跟踪相应的链接以了解以下信息：
  - 24/7技术支持
  - 应用笔记
  - 网上研讨会
  - 代码示例
  - 数据手册
  - 以及更多详细信息！

谢谢!!

Microchip Technology Inc.
RTC
Regional Training Centers