PIC系列单片机开发应用技术

电工电子实验中心 郭万有

2000.12

第一章　PIC单片机概述

第一节　PIC单片机主要特点

一. 高性能的RISC结构CPU

- 1. 哈佛双总线结构

  程序总线和数据总线独立, 避免了瓶颈现象.

- 2. RISC指令集（精简指令集）

  具有指令 33/35/58条， 因此易学易用.

- 3. 指令周期

  大部分为单周期指令 (分支指令除外) .

- 4.执行速度

  大部分指令周期在(200ns-1us)

  PIC17XX是目前执行速度最快的8位单片机。
- 5.多级硬件堆栈

  可为2、8、16级。
- 6.寻址方式

  直接、简接、相对等方式寻址。
- 7.功能完备的存储器

  PIC单片机均含有程序存储器和数据存储器

  有的还含有E^2PROM数据存储器.

二.丰富的功能部件

- 1. I/O引脚驱动电流大

  I/O引脚驱动电流达25mA，可直接驱动LED

- 2. I/O脚可双向独立编程设置
- 3. 1~4个8/16位的定时器/计时器/分频器

  8位预分频或后分频或8位周期寄存器

  一个自振式的看门狗定时器WDT

- 4.并行通信口

  中级以上PIC可以将某个双向I/O口当作8位并行口使用.

- 5. 多种串行通讯接口

  支持I^2C/SPI及SCI/USART操作，可以满足各种通信规程的要求.

- 6.内置A/D转换器

  某些中级产品内置4~8通道8位A/D转换器,有的高级产品具有12路10位A/D转换器.

- 7. 具有1-2个内置比较器.

- 8. CCP(捕捉/比较/PWM)接口

  中高级产品具有1~4 路CCP接口.

- 9.某些产品具有LCD显示接口 .

三.微控制器特征

- 1. 内置上电复位POR电路
  上电延时时序，复位定时器.
- 2. 4种可选择的振荡方式
  RC、LP、XT、HS
- 3. 程序保密位
  可以防止程序代码的非法拷贝.
- 4. 低功耗睡眠方式
  耗电量小于1uA. 可用几种方式唤醒CPU.
- 5.内置掉电复位锁定电路
  对供电电压进行检测,避免了系统失控.
- 6. 内置上电定时器PWRT
  保障工作电压的稳定建立.

- 7. 内置振荡启动定时器

  保证建立稳定的振荡.

- 8. 多种中断源.

  外部INT触发中断

  定时器TMR0溢出中断

  端口B电平变化中断

  CCP中断

  SCI中断

  SSP中断

  A/D中断

  E^2PROM写完成中断

四.CMOS电气工艺特性

- 1.低功耗, 用于电池供电的产品中

  <2mA @ 5V,4MHz

  <15uA @ 3V,32kHz

  <1uA @ Sleep摸式

- 2.宽工作电压范围

  2.5V~5V

- 3.工作温度范围宽

  商用级 0~70 ^{0}C

  工业级 -40~85 ^{0}C

  汽车级 -40~125 ^{0}C

第二节　PIC单片机系列简介

- PIC单片机的4种类型

  按程序存储器类型，可划分为OTP、E^2PROM、FLASH存储器及掩膜ROM等四种类型

- PIC单片机的三个层次(按性能分）

1.基本级：12位指令系统,8位数据线,常用于嵌入式控制.典型器件如PIC16C57等.

2.中级：14位指令系统,8位数据线.多级中断.常用于各种高.中.低档电子产品,典型器件如PIC16C74等.

3.高级：16位指令系统,8位数据线,多级中断.常用于高精度电机控制.工业过程控制等.典型器件如PIC17XX.PIC18XX等.

第二章　PIC单片机结构

第一节　PIC单片机的内部结构

一、PIC单片机的硬件结构特点:

1. 指令总线与数据总线分离的哈佛结构.

在PIC系列单片机中,程序和数据是从不同的存储器中通过独立的总线存取的，因此改善了冯.诺伊曼结构的带宽。相互分离的指令总线和数据总线使得指令位宽可以不同于8位数据总线.

指令宽度为(12/14/16)位, 使得所有的指令都为单字节指令，在一个机器周期内就可以完成操作.

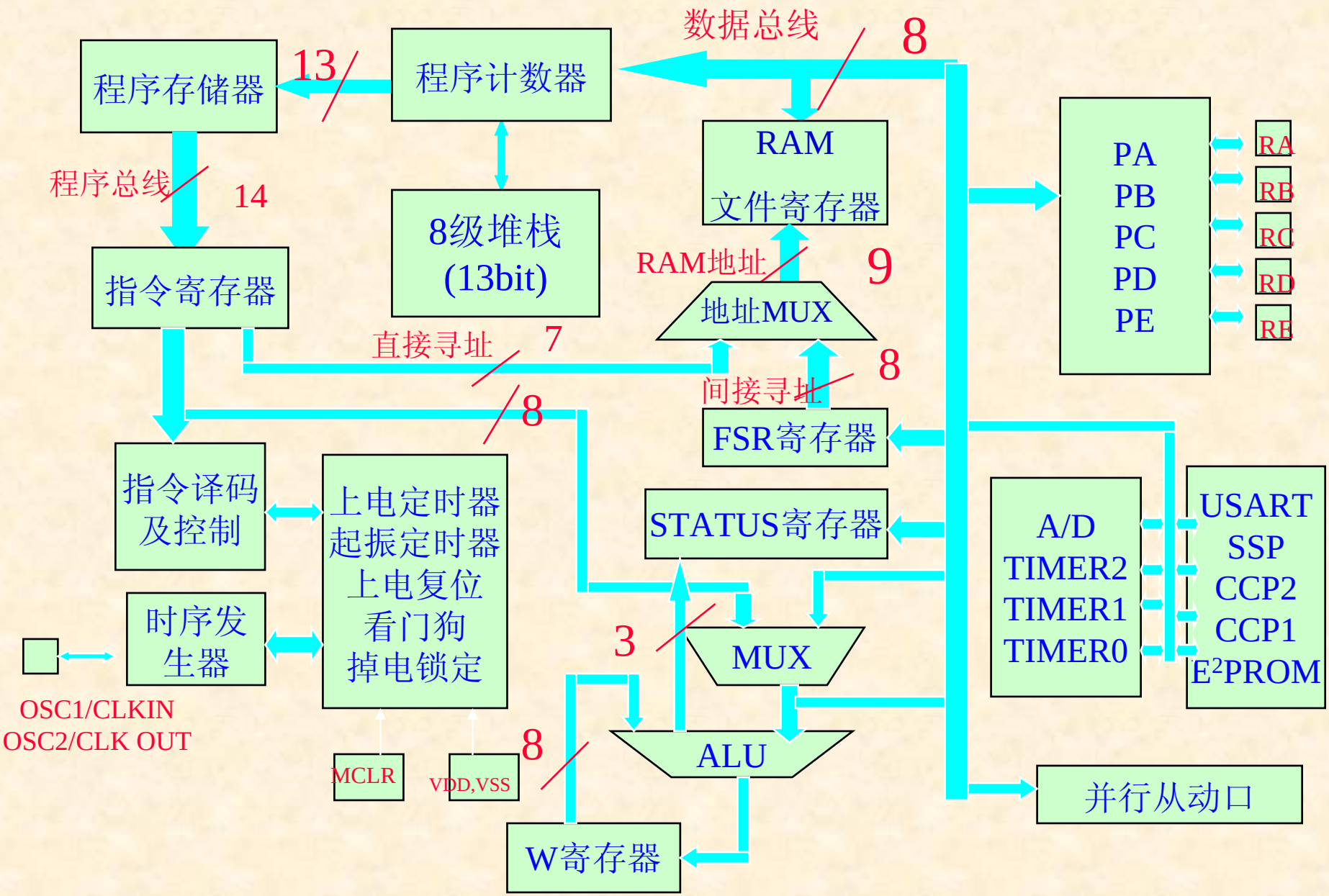
2. 流水作业的指令运行方式

指令周期由4个机器周期Q1~Q4组成，执行指令贯穿在Q1~Q4四个节拍中，取指和执行各占一个指令周期。

由于采用了哈佛结构,就有可能在执行一条指令的时候,取出下一条指令,即采用流水作业方式取出指令和执行指令.这样以来，每条指令执行时间等效为一个指令周期 。

PIC单片机指令的流水作业方式,也是RISC结构单片机的特点,使PIC单片机的运行速度比一般单片机提高了数倍.

二、PIC16F874/877的内部结构



第二节　程序存储器

一、程序存储器类型

1. 可重擦型

EPROM型：　紫外线可擦写存储器，适用于开发过程中。

E^2PROM型：　电可擦写型存储器，适用于那些可能会经常改动程序的应用场合。

Flash memory型：　这种存储器允许无数次擦除及编程的能力,支持在线擦写。

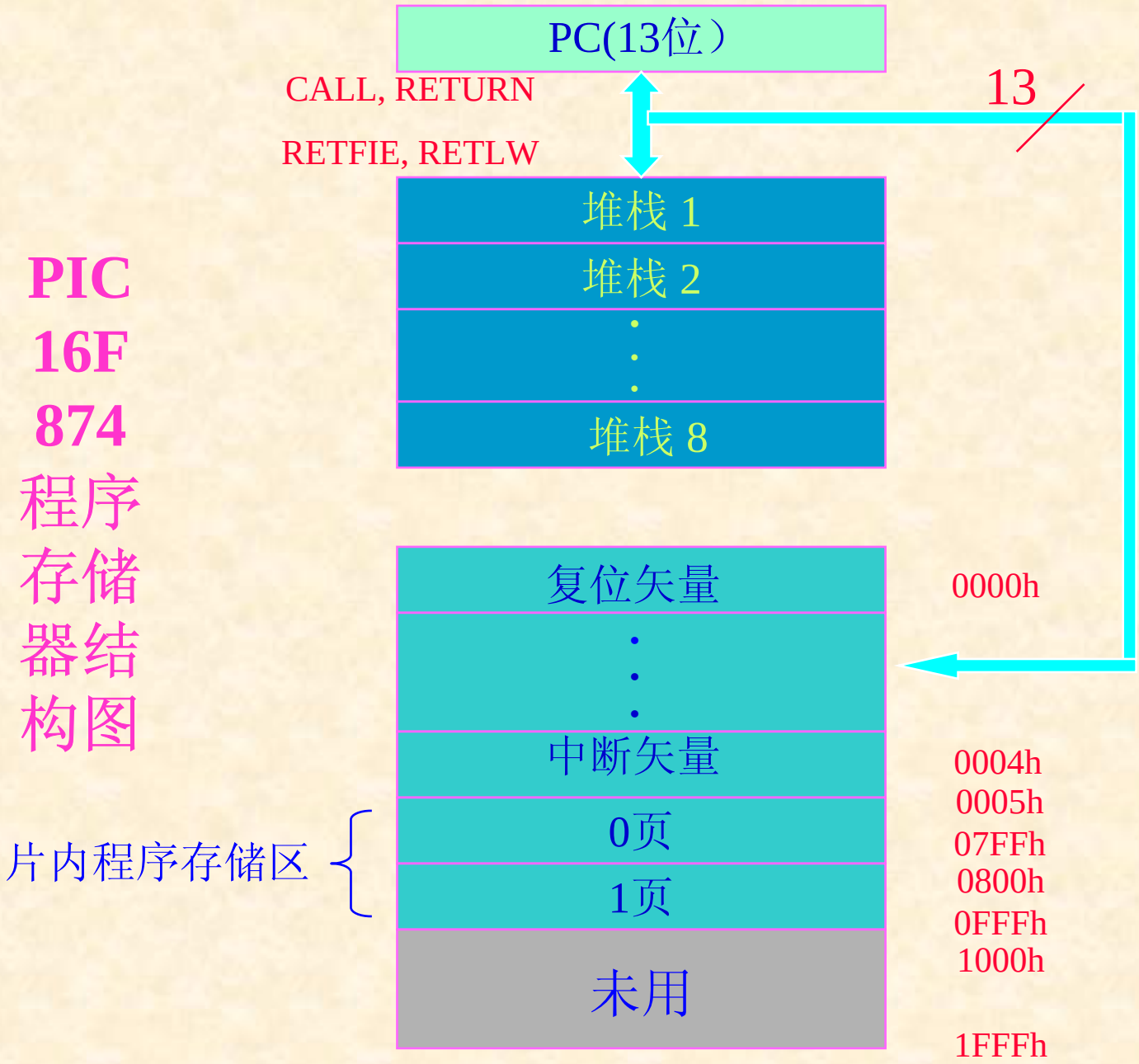
2. OTP(一次性编程)型

用编程器进行一次性编程，适合中小批量生产采用。带OTP存储器的PIC芯片，最少384x12位,最大16384x16位。

3. 掩膜型(MASK)

当产品已经稳定,程序无需进一步修改后,可以选择掩膜型芯片以进一步降低成本.

PIC 16F 874 程序存储器结构图



二、程序计数器

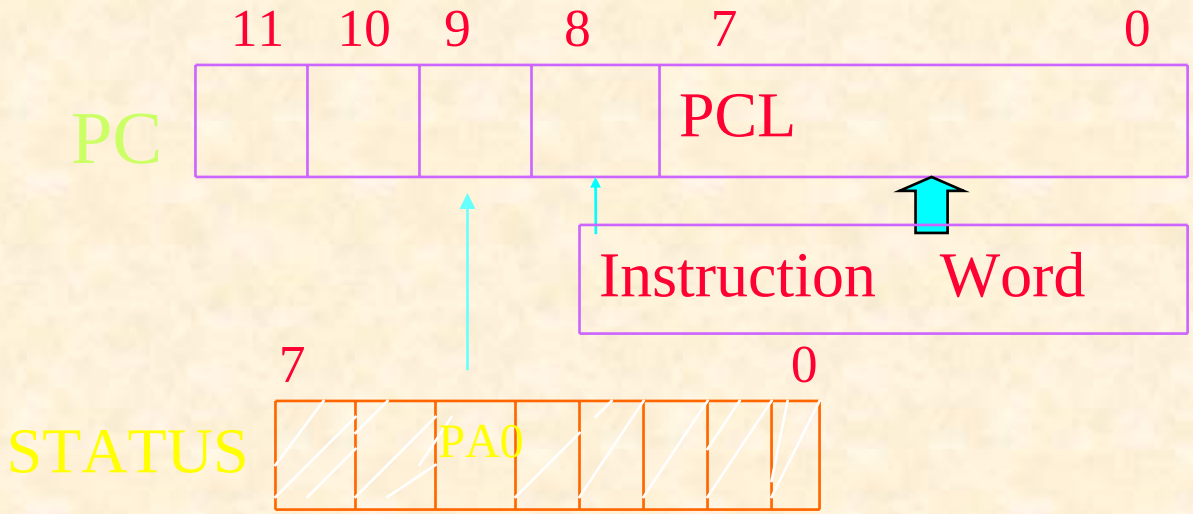
PIC12C5XX的程序计数器为12位宽，可寻址4K空间;

PIC16CXX程序计数器为13位宽，可寻址8K空间;

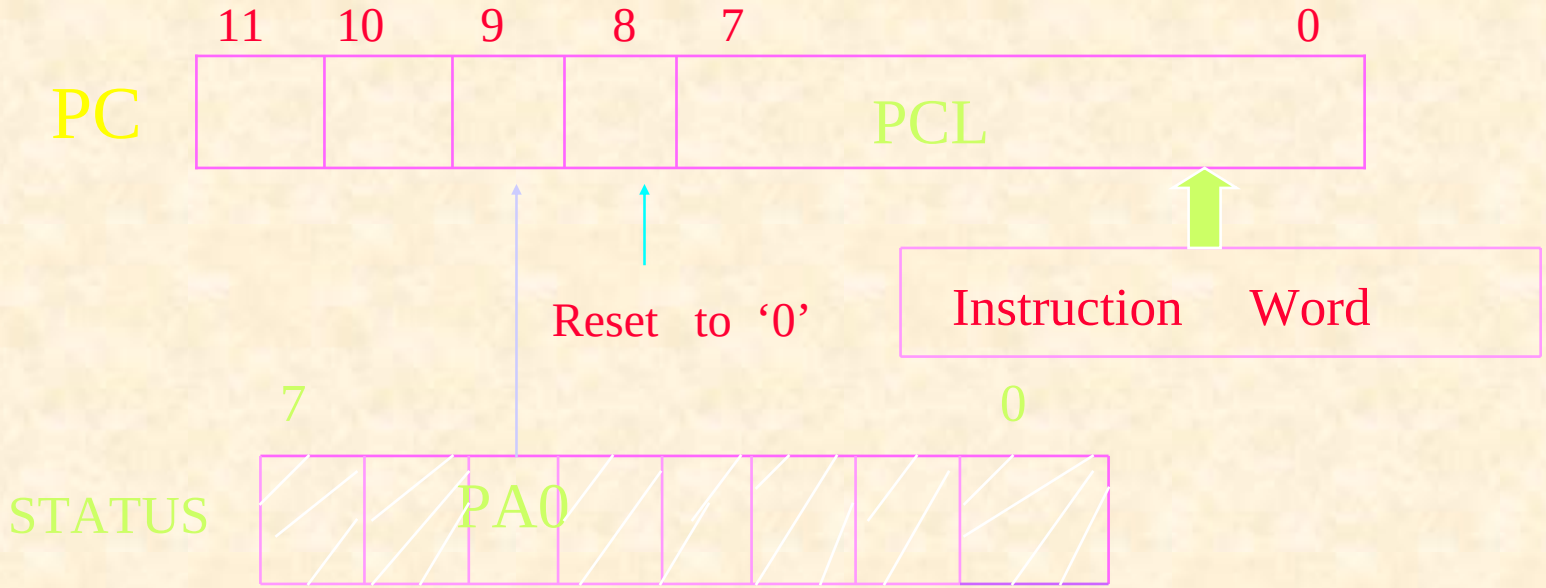
PIC17CXX程序计数器为16位长，可寻址64K空间.

1.PIC12C5XX的程序计数器

PIC12CXX程序存储器以512个字节为单位进行分页管理,例如，PIC12C509有2个页面存储区，由状态寄存器STATUS的第5位PA0指定存储区的页面。其 GOTO指令的寻址方式如下图所示:



PIC12C5XX的CALL指令码中仅包含目的地址的低8位,即PC<7:0>,PC<8>总是会被硬件自动清零,其页面地址也由STATUS<5>置入PC<9>,见下图.由此可见,子程序的起始地址必须放在每个页面的上半部,即头256个字节内,在执行CALL指令前,也要把PA0置成的值.



2. 程序计数器PC及PCLATH

PIC16CXX的PC是一个13位寄存器,其低8位PCL是可读写的,高5位PCH不能直接读写,而要从PCLATH载入.大多数指令的目的地址是8位或不足8位,它们作为ALU的结果置入PCL.CALL和GOTO指令包含11位地址信息,可以在2K空间内跳转,所以PIC16XX的每页为2K.

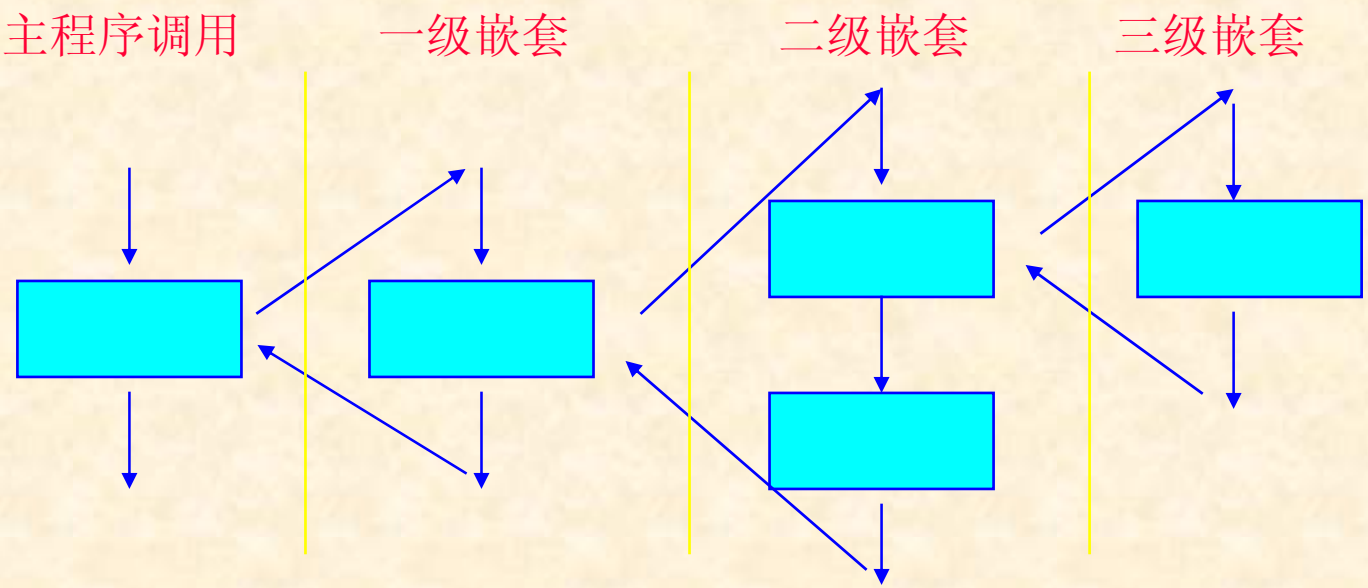
当要发生跨页的CALL或GOTO时,应先预置PCLATH<3>,使其指向所需的页面.当调用发生时,13位的PC值全部压入堆栈保存,调用结束执行RETURN指令时,程序显然会返回原来的页面,所以RETURN指令前不必考虑页面问题.

3. PIC17CXX的PC及PCLATH

PIC17CXX的PC是16位的,它分成8位的PCH,其低8位PCL是可读写的,高8位PCH不能直接读写,而要从PCLATH载入，其处理类似PIC16CXX

三、堆栈

PIC单片机的堆栈位宽与PC一致,它们不占用户程序存储器.PIC12C5XX基本级产品有2级堆栈, 因此只能容纳2层程序嵌套调用. PIC16XX的堆栈有16级,所以中高级PIC单片机具有很强的程序嵌套调用能力.



第三节　数据存储器

在PIC单片机中，称内部数据存储器为文件存储器,它被划分成若干个体(Bank)，由状态寄存器中的RP0、RP1等位选定。

每个体又由通用寄存器和专用寄存器两部分组成。每个体最大可扩充到128字节。

在每个体中，专用寄存器被安排在地址空间低端，通用寄存器安排在地址空间高端。一个体中的某些专用寄存器可以映射到其它体中，以压缩代码量，提高访问速度

一、通用数据寄存器

通用数据寄存器是静态RAM. PIC基本级产品只有25个字节.中级产品的通用数据寄存器稍多,但除了PIC16C66/76/77有368个字节外,多数只有80,或128,或176,或192个字节．高级产品内部数据寄存器较多,最多的PIC17C756多达902个字节.

由于数据存储器的低地址区是特殊功能寄存区,所以通用数据寄存器的地址不是从00H开始的.

二、专用寄存器（特殊功能寄存器）

特殊功能寄存器有两大类,一类是用来控制CPU操作的,另一类是控制功能部件操作的.

PIC单片机的所有特殊寄存器或数据存储器包括程序计数器都映射到数据存储器.

PIC单片机的指令系统支持用任何寻址方式对任何寄存器进行任何操作运算, 从而使编程变得十分简便.

与CPU有关的特殊功能寄存器

1.状态寄存器STATUS

状态寄存器包含了ALU的算术状态,芯片复位状态及程序页面位.不同型号的PIC单片机的状态寄存器虽然有不同,但差别很小.如下图是PIC12CXX的状态寄存器.

GPW	-	PA0	T0	PD	Z	DC	C

如下图是PIC16CXX的状态寄存器,仅仅是6,7,8三位与PIC12CXX的状态寄存器有所不同.

IRP	RP1	RP0	T0	PD	Z	DC	C

PIC17CXX状态寄存器的功能由CPUSTA、LUST和存储器选择寄存器BSR一起来实现.

和其他寄存器一样,状态寄存器也可以作为各种指令的目的操作数.

如果状态寄存器作为对Z,DC或C有影响的指令的目的操作数,则对这几个状态位的写无效的,这些位仅仅根据芯片的逻辑设定或清除.因此,以状态寄存器为目的操作数的指令的执行结果可能会与期望的不同.

建议用户使用BCF、BSF、SWAPF及MOVWF指令来改动STATUS寄存器,因为这条指令对Z, C, 或DC没有影响.

- 2.参数定义寄存器OPTION

OPTION寄存器包含了对TMR0/WDT预分频,设定外部中断INT和TMR0的触发极性,TMR0的时钟源及I/O口弱上拉选择,电平唤醒选择和TMR0的组态信息.如下图是PIC12CXX的OPTION寄存器.

GPWU	GPUU	T0CS	SRT0	PSA	PS2	PSIC	PS0

值得注意的是,PIC12CXX和PIC16C5X的OPTION寄存器是不可读写的,必须执行“OPTION”指令把W寄存器的内容置入OPTION寄存器,如下例:

```
MOVLW     7      ; W=“00000111”
OPTION           ; W    OPTION
```

在PIC16CXX中,OPTION寄存器是可读的,PIC16CX的OPTION寄存器如下图所示:

OPWU	GPPU	T0CS	SRT0	PSA	PS2	PSIC	PS0

其中低6位是与PIC12CXX的OPTION寄存器相同的.

3.间址寄存器INDF和文件选择寄存器FSR

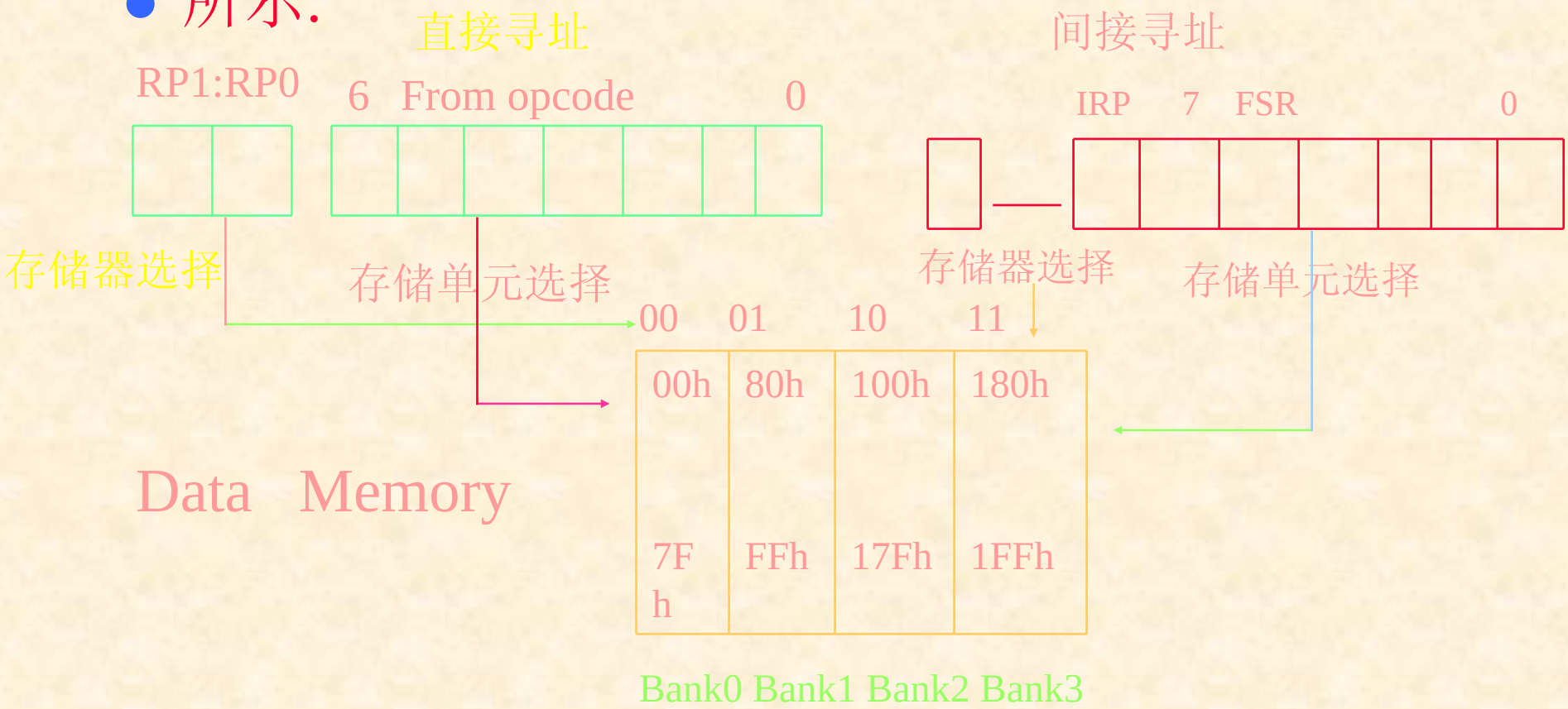
间址寄存器并不是一个物理上存在寄存器.利用间址寄存器INDF可以实现间接寻址.任何使用INDF寄存器的指令实际上是存取由文件寄存器FSR的内容所指向的存储单元.

例: 用间接寻址方式将20H-2FH的寄存器清零.

```
MOVLW    0X20
MOVF     FSR     ;首址 → FSR
```

- NEXT CLRF INDF ;清除FSR所指向的单元
- INCF FSR ;FSR+1
- BTFSS FSR ,4 ;FSR=2FH?等于则跳过下一语句
- GOTO NEXT ;NO,循环
- CONTINUE ;YES,程序向下址行
- 值得注意的是,在直接寻址和间接寻址两种方式下,包括寄存器体在内的9位有效地址的形成方式是不同的.直接寻址方式下,由STATUS寄存器的RP1,RP0加上指令中的7位操作数组成;而在间接寻址方式下,是由STATUS的第7位IRP加上FSR的8位数据形成9位有效地址的,如下图

- 所示:


直接寻址
RP1:RP0
6 From opcode 0
存储器选择
存储单元选择
间接寻址
IRP 7 FSR 0
存储器选择
存储单元选择
00 01 10 11
00h 80h 100h 180h
7F h FFh 17Fh 1FFh
Data Memory
Bank0 Bank1 Bank2 Bank3

- 4.电源控制寄存器PCON

PIC单片机可以有多达五种不同的复位方式,包括上电复位,正常状态下的MCLR端复位,SLEEP状态下MCLR复位,正常状态下WDT溢出复位和SLEEP状态下WDT溢出复位.PIC单片机的各种寄存器在不同的复位后的值是不同的.例如,在SLEEP状态下,WDT的溢出不会改变寄存器的内容,这显然有利于系统复位后的正常运行.所以有多种复位状态是有益的,但是发生复位后,必须要能区分出是何种复位,才能保证系统复位后正确运行.各种PIC单片机的电源控制寄存器PCON不尽相同.如图是PIC16C62/64/65的PCON寄存器,只使用bit1.

-	-	-	-	-	-	POR	-

- 如图是PIC16C62A/63/64/65A PCON寄存器,

-	-	-	-	-	-	POR	BOR

- 除了POR电源上电复位标志位外,还有BOR掉电锁定复位标志位.利用PCON寄存器和STATUS寄存器所提供的信息,可以判断出发生了何种复位.

第三章　PIC单片机的RISC指令系统

- PIC单片机有(33/35/58)条指令
- 三种类型的指令

  面向字节操作类

  面向位操作类

  面向常数和控制操作类.

第一节　PIC单片机的寻址方式

PIC单片机的寻址方式分为寄存器简接寻址、立即数寻址、直接寻址和位寻址4种.

一.寄存器间接寻址

```
MOVLW   05H  ; W=5
MOVWF    4   ; W(=5)   ⟶  F4 (FSR)
MOVLW   55H  ; W=55H
MOVWF    0   ; W(=55H) ⟶  F5
```

上面这段程序把55H送入F5寄存器。间接寻址方式主要用于编写查表、写表程序。

二.立即数寻址

这种寻址方式的操作数为立即数,可以直接从指令中获取. 例:

```
MOVLW    16H  ; 16H  ⟶ W
```

三. 直接寻址

对任何一个寄存器直接寻址访问.

例: MOVWF 8 ; W ⟶ F8寄存器

MOVF 8,W ; F8 ⟶ W

四. 位寻址

对寄存器中的某一位进行操作

例: BSF 11,0 ; 把F11的第0位置为1

第二节　基本级和中级PIC单片机指令详解

PIC单片机指令包括操作码和操作数。 基本级指令12位， 中级指令14位。 由于操作码是相同的， 所以中档机14位长的指令相对低档机指令而言, 多了两位操作数。指令码结构如下:

(11-----6)　(5)　(4----0)

OPCODE	d	ffff

a.12位指令

(13-------8)　(7)　(6-----0)

OPCODE	d	fffffff

b.14位指令

(13------8)　(7)　(6--------0)

OPCODE	d	fffff	ff

c.12/14位指令

指令操作码说明：

f ： 文件寄存器（数据寄存器）

W： 工作寄存器 （累加器）

b ： 数据寄存器位址

k： 常数

x ： 不需要关心的位

d ： 目标寄存器 d=0 W ;d=1 f

s ： 目标寄存器 s=0 W和f ;s=1 f

p ： 端口地址（指PIC17CXX)

i ： 指针控制。i=0 指针不变； i=1执行后指针加1

t ： 字节高低位选择

u ： 未使用的位

一.面向字操作类指令

共18条,包括数据传送，算术和逻辑运算，数据移位和交换等操作。这些操作都是在工作寄存器W和数据寄存器f之间进行的。

NOP	空操作	INCFSZ f,d	f递增，为0跳
MOVWF	w送f	DECFSZ f,d	f递减，为0跳
MOVF f,d	f送d	ANDWF f,d	f按位与
CLRW	清W	IORWF f,d	f按位或
CLRF f	清 f	COMF f,d	f按位取反
ADDWF f,d	f加W	XORWF f,d	f按位异或
SUBWF f,d	f减W	RRF f,d	循环右移
INCF f,d	f递加	RLF f,d	循环左移
DECF f,d	f递减	SWAPE f,d	半字节交换

1.寄存器加

格式　　ADDWF　f,d

指令码

000011	d	fffff	ff

指令周期:　1

操作:　　W+f　⟶　d

影响状态位:　C, DC, Z

说明:　　将f和W相加,结果存入f(d=1)或W(d=0).

例:　ADDWF　8,0　; F8+W　⟶　W

2.寄存器与指令

格式　　ANDWF　f, d

指令码

000101	d	ffff	ff

指令周期:　　1

操作:　　W^f ⟶ d

影响状态位:　　Z

说明:　将f和W做逻辑与运算,结果存入f(d=1)
或 W(d=0).

例:　ANDWF　10,0　; F10^W ⟶ W

ANDWF　10,1　; F10^W ⟶ F10

3.寄存器清零指令

格式　　CLRF　f

指令码

000001	1	fffff	ff

指令周期:　1

操作:　0 ⟶ f, 1 ⟶ z

影响状态位:　Z

说明:　将f寄存器清零.

例:　CLRF　8 ; F8清零(0 ⟶ F8)

4.W清零指令

格式　　CLRW(DT)

指令码

000001	0	00000	00

指令周期: 1

操作: 0 → W , 1 → z

影响状态位: Z

说明: 将W寄清零,状态位Z将被置为1.

5.寄存器取反指令

格式　　COMF　　f,d

指令码

001001	d	fffff	ff

指令周期:　1

操作:　　$\overline{f} \longrightarrow d$

影响状态位:　　Z

说明: 将f寄存器内容组做逻辑求反运算,结果存入f(d=1)或W(d=0).

例:　COMF　12，0　　；F12取反 → F12

　　COMF　12，1　　；F12取反 → W

6.寄存器减1指令

格式　　DECF　　f,d

指令码

000011	d	fffff	ff

指令周期:　1

操作:　　f　－　1　⟶　d

影响状态位:　　Z

说明: 将f寄存器内容减1,结果存入f(d=1)或W(d=0).

例:　　DECF　15,1　;　F15－　1 ⟶　F15

　　DECF　15,0　;　F15－　1　⟶　W

7.寄存器减1,结果为零则跳指令

格式　　DECFSZ　f,d

指令码

001011	d	fffff	ff

指令周期: 1或2

操作:　　f-1 ⟶ d, 结果为零则间跳(PC+1 ⟶ PC)

影响状态位: 无

说明: 将f内容减1结果存入f(d=1)或W(d=0), 如果结果为零,则跳过下一条指令,否则顺序执行下一条指令.

例:　　MOVWF　8

LOOP　DECFSZ　8,1

GOTO　LOOP

……

8.寄存器加1

格式　　INCF　f,d

指令码

001010	d	fffff	ff

指令周期: 1或2

操作:　　f + 1⟶ d, 结果为零则跳(PC+1⟶ PC)

影响状态位: 无

说明: 将f内容加1结果存入f(d=1)或W(d=0).

例:　　MOVWF　8

LOOP　DECFSZ　8,1

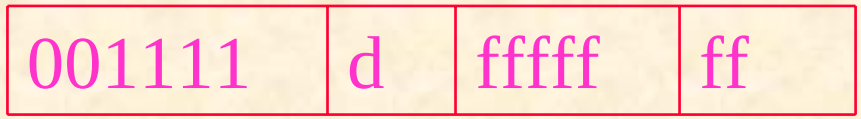
GOTO　LOOP

……

9.寄存器加1,结果为零则跳指令

格式　　INCFSZ　f,d

指令码



指令周期: 1或2

操作:　f + 1 ⟶ d, 结果为零则间跳(PC+1 ⟶ PC)

影响状态位: 无

说明: 将f内容加1结果存入f(d=1)或W(d=0), 如果结果为零,则跳过下一条指令,否则顺序执行下一条指令.

例:
```
        MOVWF  8
LOOP    INCFSZ   8,1
        GOTO   LOOP
        ……
```

10.寄存器或指令

格式　　IORWF　　f,d

指令码

000100	d	fffff	ff

指令周期:　1

操作:　　W ∨ f ⟶ d

影响状态位: Z

说明: 将f和W做逻辑或运算,结果存入f(d=1)或W(d=0).

例:　　IORWF　10,0　;　F10 V　W ⟶ W

　　IORWF　10,1　;　F10 V W ⟶ F10

11. f寄存器传送指令

格式　MOVF　f,d

指令码

001000	d	fffff	ff

指令周期:　1

操作:　　f　⟶d

影响状态位:　Z

说明: 将f内容传送至本身f(d=1)或W(d=0),如果是传送至本身,一般是用来影响状态位Z,用来判断f是否为零.

例: MOVF　　10,1 ;　F10　⟶F10

　　BTFSS　　3,2　;　判断F3的第2位，即Z状态位,若Z=1，即F10=0，则跳过CLRW执行ANDLW 55H

　　CLRW

　　ANDLW　55H

12.W寄存器传送指令

格式　　MOVWF　　f

指令码

000000	1	fffff	ff

指令周期:　　1

操作:　　W　　f

影响状态位:　无

说明:　　将W寄存器内容传送至f寄存器.

例:　　MOVWF　6　;　W⟶ F6(B口)

13.空操作指令

格式　NOP

指令码

00000	0	00000	00

指令周期: 1

操作: 无任何操作

影响状态位:无

说明:不做任何操作,仅仅使PC加1.

14.带进位左移指令

格式　　RLF　f,d

指令码

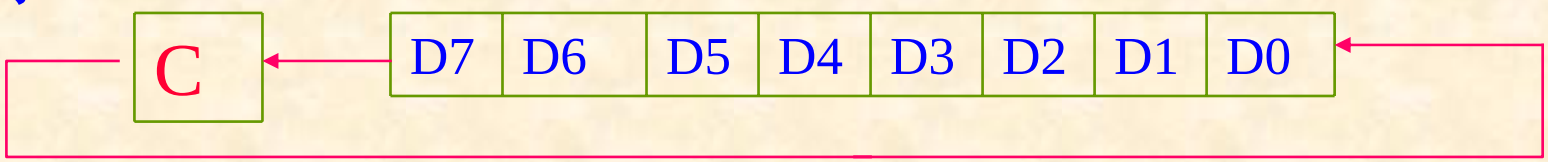
001101	d	fffff	ff

指令周期: 1

操作:　f(n) ⟶ d(n+1)　, f(7) ⟶ c,c ⟶ d(0)

影响状态位:　C

说明: 将f寄存器内容左移,结果存入f(d=1)或W(d=0).左移时,其最高位移入状态位,如下图所示.



例：RLF　8,1　;F8左移 ⟶ F8

　　RLF　8,0　;F8左移 ⟶ W

15.带进位位右移指令

格式　　RRF　f,d

指令码

001100	d	fffff	ff

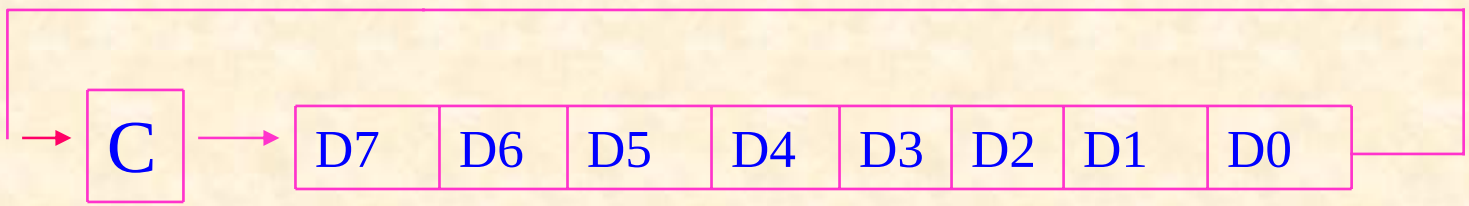
指令周期: 1

操作: f(n) → d(n-1), f(0) → c, c → d(7)

影响状态位: C

说明: 将f寄存器内容右移,结果存入f(d=1)或W(d=0).

右移时,其最抵位移入状态位C,而原来的状态位C移入F最高位,如下图所示.



例:　RRF　8,1　;F8右移 → F8

　　RRF　8,0　;F8右移 → W

16.寄存器减法指令

格式　　SUBWF　　f,d

指令码

000010	d	fffff	ff

指令周期:　1

操作:　　f － W ⟶ d

影响状态位: C, DC, Z

说明:　将f寄存器内容减去W内容,结果存入f(d=1)或W(d=0).

例:　CLRF　20　　; F20=0

　　MOVLW　1　　; W=1

　　SUBWF　20,1　; F20-W=0-1=1⟶F20, C=0,结果为负.

17.寄存器交换指令

格式　　SWAPF　f,d

指令码

001110	d	fffff	ff

指令周期: 1

操作: f(0-3) ⟶ d(4-7), f(4-7) ⟶ d(0-3)

影响状态位: 无

说明: 将f寄存器内容的高4位和抵4位交换,
结果存入f(d=1)或W(d=0).

例: MOVLW　56H
MOVWF　8　; F8=56
SWAPF　8, 1 ; F8交换,结果存F8,F8=65H

18.寄存器异或运算指令

格式：　XORWF　f,d

指令码

000110	d	fffff	ff

指令周期: 1

操作: W ⊕ f ⟶ d

影响状态位: Z

说明: 将f寄存器内容与W进行异或运算,结果存入f(d=1)或W(d=0).

例:　XORWF　5,1　　; F5 ⊕ W ⟶ F5

　　XORWF　5,0　　; f5 ⊕ W ⟶ W

二.面向位操作类指令

共4条,其指令码的基本结构为:

操作码	bbb	fffff	ff

高4位操作码,操作码之后是3位位地址,然后是5位或7位寄存器地址.

```
BCF f, b      ; 清f的b位
BSF f, b      ; 置f的b位
BTFSC f, b    ; f的b位为0间跳
BTFSS f, b    ; f的b位为1间跳
```

19.位清零指令

格式　　BCF　f,b

指令码

0100	bbb	fffff	ff

指令周期:　1

操作:　0 →f（b）

影响状态位:　无

说明:　将f寄存器的b位清为0.

例　:　BCF　8,2　;将F8的第2位清为0.

20. 位置1指令

格式 ： BSF f, b

指令码：

0101	bbb	fffff	ff

指令周期：1

操作： 1 ⟶ f(b)

影响状态位： 无

说明： 将f寄存器的b置为1

例： BSF 8,2 ;将F8的第2位置为1.

21. 位测试,为零则间跳指令

格式 ： BTFSC f,b

指令码：

0110	bbb	fffff	ff

指令周期: 1或2

操作: 如果f(b)=0则间跳(PC+1→ PC)

影响状态位: 无

说明: 测试f寄存器的b位,如果b位为零则跳过下一条指令，否则顺序执行程序。

例: BTFSC 8,2 ;测试F8的第2位,如果为0则跳到INCF9,1执行

```
MOVF   5,0
INCF   9,1
```

22.位测试,为1则间跳指令

格式 ： BTFSS f,b

指令码 :

0111	bbb	fffff	ff

指令周期: 1或2

操作: 如果f(b)=1则跳(PC+1⟶ PC)

影响状态位: 无

说明: 测试f寄存器的b位,如果b位为1则跳过下一条指令, 否则顺序执行程序.

例: BTFSS 8,2 ;测试F8的第2位, 如果为 1 则跳到语句INCF9,1执行

```
MOVF  5,0
INCF   9,1
```

三、常数和控制操作类指令

这类指令共11条,由于12位与14位指令的操作码不尽相同,所以指令码分别列出.

OPTION ; 写OPTION	MOVLW k ; 置W常数
SLEEP ;置睡眠状态	IORLW k ; W或常数
CLRWDT ;清WDT	ANDLW k ; W与常数
TRIS f ;置I/O状态	XORLW k ; W异或常数
RETLW k ;子程序带参返回	
CALL k ;调子程序	
GOTO k ;跳转	

23.常数与指令

格式 ： ANDLW K

指令码：

1110	kkkkkkkk

11	1001	kkkk	kkkk

指令周期: 1

操作: W^ K ⟶ W

影响状态位: Z

说明: W寄存器和常数k按位逻辑与运 算,结果存入W.

例: ANDLW 55H,0 ; W与55H W

24.子程序调用指令

格式　：CALL　k

指令码:

1001	kkkkkkkk

01	0kkk	kkkk	kkkk

操作:　12位指令: PC+1 → 堆栈,K → PC(0-7),

0 → PC(8), PA1.PA0 → PC(10-9).

14位指令: PC+1→ 堆栈,K→ PC(10,0),

PCLATH(4,3) → PC(12,11)

指令周期:　2

影响状态位:　无

说明: 将 PC加1后推入堆栈,将入口地址K置入PC,同时修改页面地址即PCLATH(14位指令) 或STATUS的PA1和PA0（12位指令）,从而在PC中形成子程序的入口地址.

对于12位指令的单片机,还要注意虽然子程序可以放在任何一个页面,但是必须放在页面的上半部,因为执行CALL指令会将PC的第九位清为零.

```
例:          CALL   DELAY  ;调用子程序DELAY
                :
                :
DELAY   MOVLW  80H    ;子程序
                :
                :
             RETLW  0
```

25.看门狗计数器清零指令

格式　　CLRWDT

指令码

0000	000	0100

00	0000	0110	0100

指令周期：　　1

操作：　0 → WDT，0 → WDT预分频器

影响状态位：1 → T0，1 → PD

说明：清除WDT，使其不能计时溢出，如果已将预分频器分配给WDT，则预分频器也清为零。

26。无条件跳转指令

格式:　GOTO　k

指令码:

101k	kkkkkkkk		
10	1kkk	kkkk	kkkk

指令周期：2

操作：K → PC（8-0），PA1、PA0 → PC(10、9）

或 K → PC(10-0），PALATH(4、3） → PC(12、11）。

影响状态位：　无

说明: 无条件跳转,将常数k置入PC,同时把14位指令的页面地址即PCLATH(4,3)　　PC(12,11). 12位指令的PIC单片机有的有4个页面,则把程序页面位即STATUS的PA1和PA0　PC(10,9); 有的有2个页面,就把PA0　　PC(9),因此GOTO指令可以跳转到程序的任何地方.

例: LOOP　MOVLW　　80H

:

GOTO　　LOOP　;无条件跳转

:

:

27.常数或指令

格式　　IORLW　k

指令码

1101	kkkkkkkk

11	1000	kkkk	kkkk

指令周期:　1

操作:　W或K　　W

影响状态位:　　Z

说明:　将W寄存器和常数k做逻辑或运算,结果存入W。

例:　IORLW　80H　;W或80H⟶W

28.常数传送指令

格式　MOVLW　k

指令码

1100	kkkkkkkk

11	00xx	kkkk	kkkk

指令周期:　1

操作:　k ⟶ W

影响状态位:　无

说明:　将常数k置入W寄存器,这条指令不会影响任何状态位.

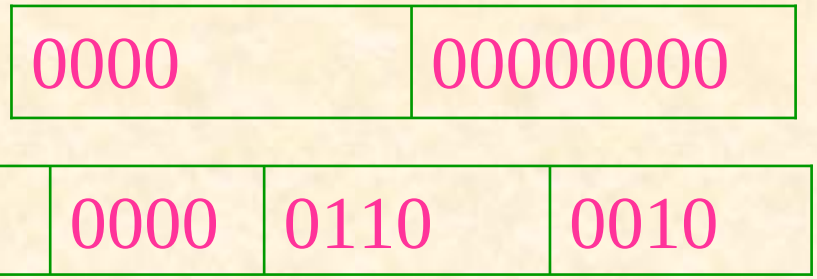
例:　MOVLW　0　; 0 ⟶ W,状态位Z保持原来的值.

MOVLW　88H　; 88H ⟶ W

29. 写OPTION寄存器指令

格式　OPTION

指令码



指令周期:　1

操作:　W ⟶ OPTION寄生器

影响状态位:　无

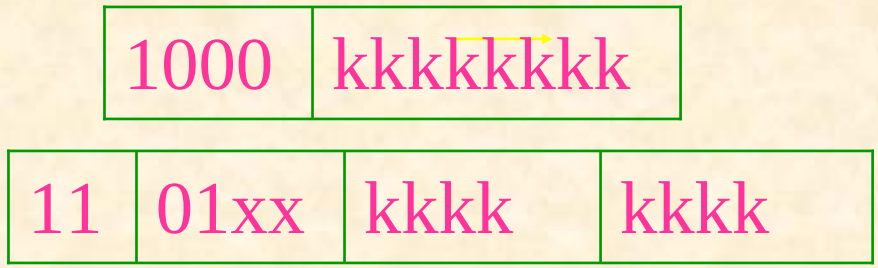
说明: 将W内容置入OPTION寄存器.

```
例:     MOVLW    07H      ; 7 ⟶W
        OPTION            ; OPTION=W=7
```

30. 子程序返回指令

格式 RETLW K

指令码



指令周期: 2

操作: K⟶ W,栈顶⟶ PC

影响状态位: 无

说明: 子程序带参数返回,栈顶内容到PC,8位常数K⟶W

例: RETLW 0 ;返回, 0⟶ W

31进入低功耗状态指令

格式　　SLEEP

指令码

	0000	0000	0011
00	0000	0110	0011

指令周期:　　1

操作:　　0 ⟶ PD,1 ⟶ T0,00⟶WDT,

0 ⟶ WDT预分频器

影响状态位:　　PD,T0

说明:　　停止芯片振荡,使PIC进入低功耗睡眠状态这条指令还会清零WDT和预分频器,并将STATUS的PD位清零,TO位置1.进入低功耗模式后,I/O状态保持不变,WDT清零后重新计时,　一旦计时溢出即把PIC从SLEEP模式中唤醒。

32.设置I/O控制寄存器指令

格式　TRIS　f

指令码

0000	0000	0ffff

00	0000	0110	0fff

指令周期: 1

操作: W ⟶ I/O控制寄存器TRISf(f 为某端口寄存器）

影响状态位:　无

说明: 将W寄存器内容置入GP口控制寄存器,以设定GP口的输入/输出方向.　f=6, 对应于GP口.

例: MOVLW　0FH

TRIS　6　　;指定I/O口高4位为输出，低四位为输入

在PIC16\17系列中，TRIS寄存器可直接读写

33.常数异或指令

格式　　XORLW　k

指令码

1111	kkkk	kkkk

11	1010	kkkk	kkkk

指令周期:　1

操作:　　W ⊕ k → W

影响状态位:　Z

说明:　　将W寄存器内容与常数k异或运算,结果存入W.

例:　XORLW　33H　　;W ⊕ 33 → W

四.PIC16和PIC17独有的指令

下面是PIC16CXX等中级单片机具有的4条常数和控制类操作指令:

```
ADDLW   ;立即数加法
RETFIE  ;中断返回
RETURN  ;子程序不带参数返回
SUBLW   ;常数减法
```

34.立即数加法指令

格式　　ADDLW　　K

指令码

11	111X	kkkk	kkkk

指令周期:　　1

操作:　W+K ⟶ W

影响状态位: C，DC，Z

说明:　将W寄存器内容与会代表位立即数K相加,
结果存入W.

例:　ADDLW　60H　; W+60H ⟶　W

35.中断返回指令

格式　　RETFIE

指令码

00	0000	0000	1001

指令周期: 2

操作:　　栈顶 ⟶ PC,1 ⟶GIE位

影响状态位:　GIE位

说明:　中断服务子程序返回指令.栈顶 为返回地址,弹回PC. 同时总中断使能位（ INTCON中的GIE位被置为“1”.

36.子程序不带参数返回

格式 ： RETURN

指令码：

11	01xx	kkkk	kkkk

指令周期： 2

操作： 栈顶 → PC

影响状态位： 无

说明： 子程序返回,栈顶内容弹回PC,返回到子程序调用处,但是不带回任何参数.

37.常数减法指令

格式　　SUBLW　　K

指令码

11	110x	kkkk	kkkk

指令周期：1

操作：　　K－W ⟶W

影响状态位：　C、DC、Z

说明：　　8位常数K减W寄存器内容，结果放入W．PIC的减法运算是用补码加法实现的.

例：MOVLW　01H　；1⟶W

SUBLW　02H　；2－W＝2－1＝1⟶W

C＝1，结果为正

MOVLW　2　；2 ⟶ W

SUBLW　1　；1-W＝1-2＝-1 =FEH ⟶ W

C＝0，结果为负

第三节　高级产品具有功能更强大的指令系统

ＰＩＣ单片机高级产品具有乘法指令，求反指令，读表/写表指令，各种比较指令，长调用指令等等，不但方便了编程，而且提高了运行速度，增加了单片机的功能.

第四章.中断

第一节 PIC单片机的丰富的中断功能

在计算机中,“中断”是一个很重要的概念.中断功能的强弱已成为衡量一台计算机尤其是用于控制场合的计算机功能完善与否的重要标志.

PIC单片机是世界上最有影响力的主要嵌入式控制器之一,具有丰富中断功能.

第二节.中断的开放禁止和响应标志

对每个中断,其开放和禁止是由中断控制寄存器INTCON和外设中断使能寄存器PIE的某一位控制的.当某个中断条件满足时,中断标志寄存器相应的位就会被置位.

一、 中断控制器寄放器I NTCON

I NTCON 中断控制寄存器包括TMR0、B口、外部中断INT引脚的中断使能位和标志位,它还包括外设中断使能位和所有中断的总使能位.它也是一个可读写的寄存器.PIC16C6X和部分PIC16C7X的INTCON寄存器如图.

GIE	PEIE	TOI E	INTE	RBIE	T0IF	INTF	RBIF

其他PIC16C7X产品,如PIC1670/71/71A增加了A/D转换器,因此其INTCON中的第6位为A/D转换中断使能位,其他各位与PIC16C6X的INTCON相同,如图

GIE	ADIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF

PIC16C8X有64*8个字节的E²PROM数据存储器,由于E²PROM 写操作需要10ms时间,所以提供了E²PROM 写操作完成中断.PIC16C8X的INTCON把第7位用作 E²PROM 写完成中断使能位. E²PROM 的中断标志位在EECON1中. PIC16C8X的INTCON寄存器的如下图:

GIE	EEIE	T0IE	INTE	RBIE	T0IF	INTE	RBIF

PIC17CXX也具有丰富的中断功能,完成相当于PIC16C6X的INTCON寄存器的功能是INTSTA寄存器,其结果与INTCON类似,不再枚举.

二、外设中断使能寄存器PIE1

PIE1包括各个外设即各个功能模块的中断使能位,如图:

PSPIE		RCIE	TXTE	SSPIE	CCP1IF	TMR2IE	TMR1IE

PIC16C6X的PIE1中未使用的第6位在PIC16C7X中用作A/D中断使能位即ADIE.

三、外设中断标志寄存器PIR1

该寄存器包括了各个外设,即各个功能模块的中断标志位,如图:

PSPIE	-	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF

PIC16C6X的PIR1中未使用的第6位 在PIC16C7X中用作A/D中断标志位ADIF.在PIC16C8X的E^2PROM写操作完成中断标志位是E^2PROM 的控制寄存器EECON1的第4位EEIF.

四、CCP2中断使能寄存器PIE2

PIC16C63/65/73/74,PIC17C42/43/44等有两个捕捉/比较/脉宽调制模块.PIC17C52/56有4个CCP模块,所以功能更强大.CCP2由PIE2寄存器使能,如图

-	-	-	-	-	-	-	CCP2IE

五、CCP2中断标志寄存器PIE2

PIR2是CCP2的中断标志寄存器,如图:

-	-	-	-	-	-	-	CCP2IF

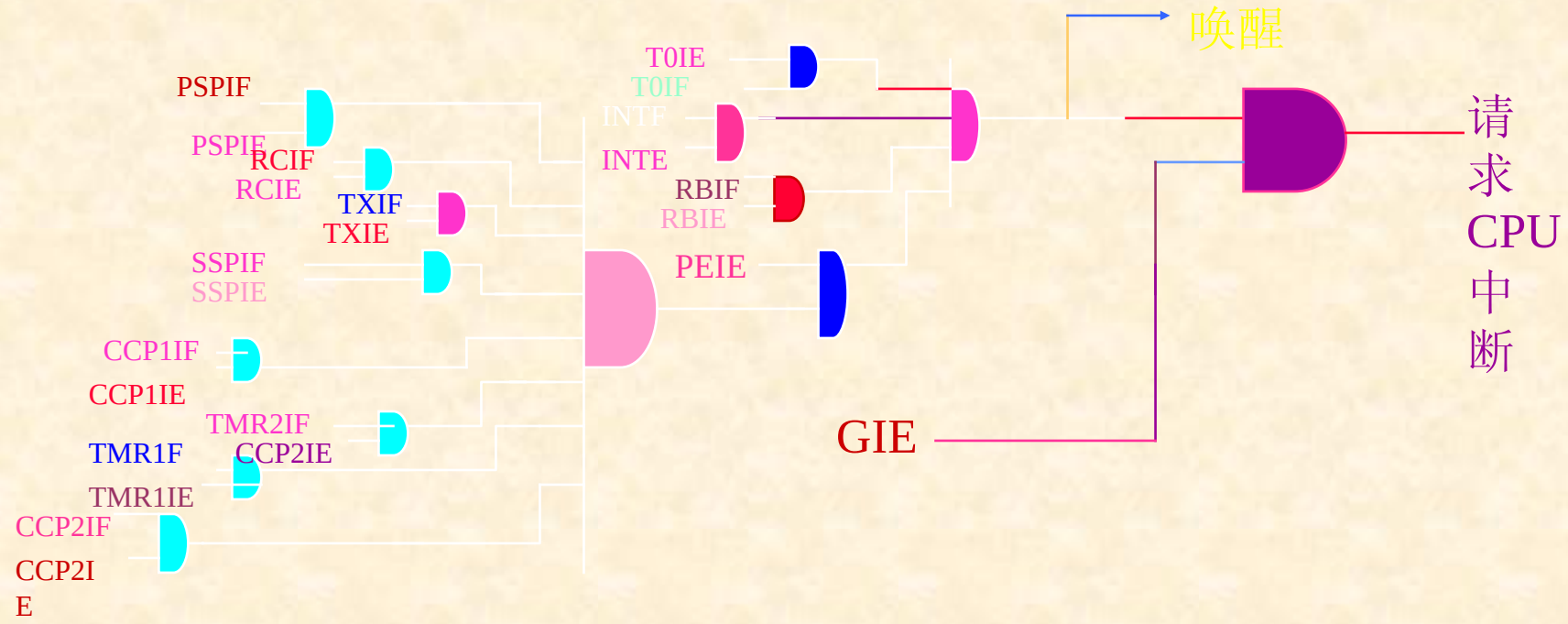
第三节 中断响应过程

- 在有关中断的寄存器IINTCON,PIE1,PIE2,PIR1

PIR2中包括了各种中断的使能位和标志以及中断总使能位GIE.芯片复位后硬件自动置GIE=0关闭所有中断,而中断返回指令“RETFIE”执行后将置GIE=1重新开放中断.不管各种中断使能位或中断总使能位GIE是什么状态,中断条件满足时都会发出中断请求,相应的中断标志会被置位,但是CPU是否响应中断则要根据其使能位的状态而定.CPU响应中断后,硬件自动清GIE=0关闭所有中断以免发生重复中断,然后把当前PC值压入堆栈,PC寄存器置以中断向量地址而开始执行中断服务程序.进入中断服务程序后,程序必须检查中断源,这是通过检查中断标志位实现的.

第三节 中断响应过程(续)

- 一旦确定了中断源,就用软件把该中断标志位清零,以免执行中断返回指令RETFIE后,由于中断标志位仍为1而引起重复中断.PIC16C62的中断请求逻辑电路如图



第三节　　中断响应过程(续)

- 中断响应过程当然要引起一定的延时,对于外部触发中断,如INT和RB口中断,由于中断请求发生的时机不同,延时时间稍有差别,为3~4个指令周期.

第四节 中断现场保护

- 中断现场保护是中断技术的重要组成部分,由于在PIC单片机指令系统中没有PUSH,POP指令,所以要用一段程序来实.由于这段程序可能会影响W寄存器和STATUS寄存器,所以首先要把这两个寄存器保护起来,然后再保存用户认为应保留的其他寄存器.由于没有PUSH,POP指令,这些中断现场数据并不是保留在堆栈中的,而是保留在用户自己选择的寄存器中的,一般应选择通用寄存器来保护现场.下面两段程序分别是PIC16C61和其他PIC16C6X的中断现场保护程序.
- 1.PIC16C6X的中断现场保护程序:

```
MOVWF     W-TEMP      ;将W保护到W-TEMP
SWAPF     STATUS,W    ;STATUS → W
MOVWF     STATUS-TEMP
```

第四节 中断现场保护(续)

：
：

中断服务程序

：
：
：

SWAPF　STATUS_TEMP,M
MOVWF　ＳＴＡＴＵＳ
ＳＷＡＰＦ　Ｗ－ＴＥＭＰ，Ｆ
ＳＷＡＰＦ　Ｗ－ＴＥＭＰ，Ｗ

２．ＰＩＣ１６Ｃ６Ｘ的中断现场保护程序：

ＭＯＶＷＦ　Ｗ－ＴＥＭＰ；保护Ｗ和ＳＴＡＴＵＳ
ＳＷＡＰＦ　ＳＴＡＴＵＳ，Ｗ

第四节　中断现场保护（续）

```
BCF   STATUS，RP0
MOV   STATUS－TEMP
  :
  :
```

中断服务程序

```
  :
  :
SWAPF   STATUS－TEMP，W
MOVWF   STATUS
SWAPE   W－TEMP，F
SWAPE   W-TEMP,W
```

与第一段程序相比,第二段程序多了“BCF STATUS,PR0”

第四节　中断现场保护(续)

- 这条指令,因为在有的PIC单片机如PIC16C61中,BANK1是完成映射到BANR0的,而其他PIC16C6X的寄存器体由于有BANK0和BANK1的区别,要通过设定RP0规定使用哪个存储器,在编制中断现场保护程序时要注意:一是W-TEMP必须同时定义在BANK0和BANK1,例如若W-TEMP定义在0X20,则0XA0也必须分配给它,二是STATUS必须定义在BANK0.

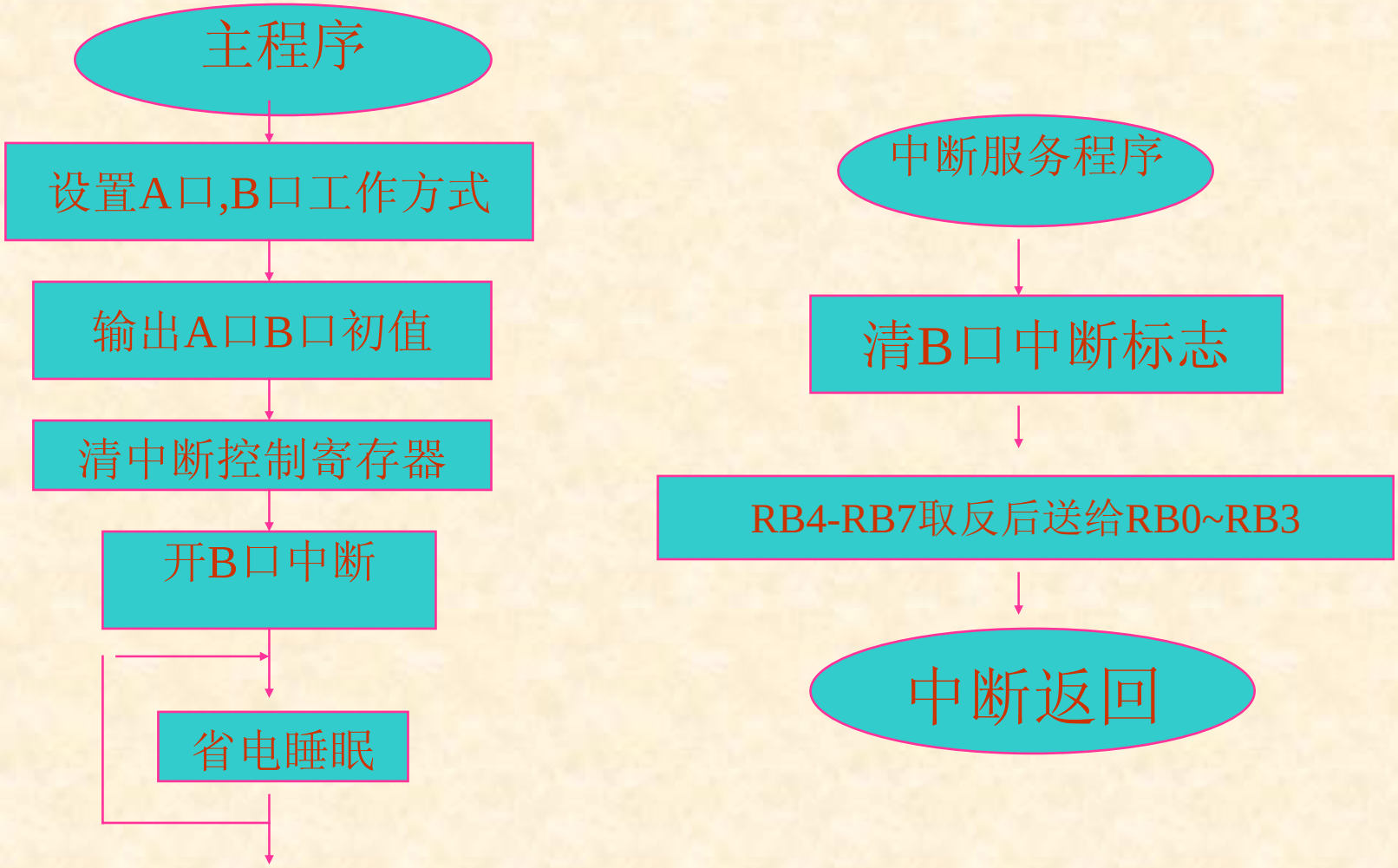
第五节　中断程序实例

- 下面是一个利用按电唤醒CPU的简单的例子,用来说明如何设置和响应中断.
- 1.电路设计

  采用的是PIC16C71单片机.CPU平时处在睡眠状态,当按下或松开某个Swi时,由于RB口电平变化引起中断,CPU就被唤醒并且相应的LEDi点亮或熄灭,然后CPU又回到省电睡眠状态.
- 2.程序流程图

  采用PIC16C71单片机,利用按键唤醒CPU程序如图:

利用按键唤醒CPU流程图



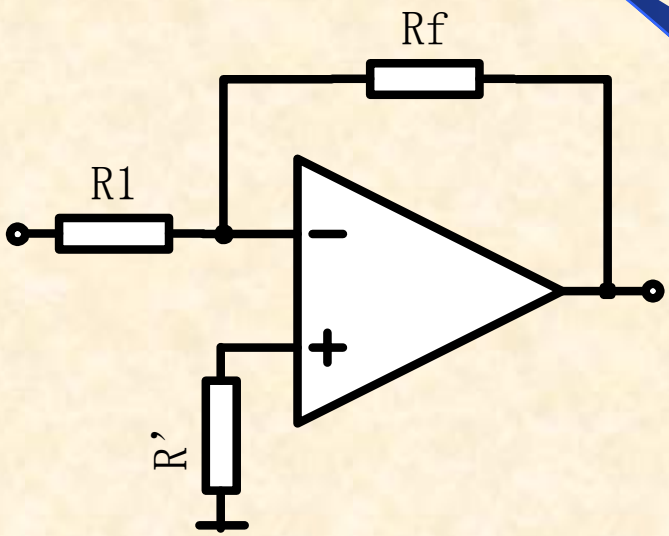
第五节 中断程序实例(续)

3.程序清单

本程序演示PIC16C71的按键唤醒特性.B口的RB4-RB7设置成带内部上拉电阻的输入方式,RB4-RB7上由电平变化引起的中断把芯片从睡眠状态中唤醒.中断向量地址为佳0004.而RB0-RB3设置成输出方式驱动4个LED显示器,每LED和相应的按钮相对应,既RB0对应RB4……执行SLEEP指令处理器进入睡眠状态后,按下任何一个按钮,处理器就会被唤醒并且转到中断向量地址执行中断服务程序,使相应的LED点亮.当按钮释放后,LED熄灭,系统又进入等待状态,等待下一次唤醒.

§ 2.1信号运算—比例放大电路

- 反相输入比例放大电路



Vo = - Fr/R1 Vi

约束条件：

R’ = R1 // Rf

性能：

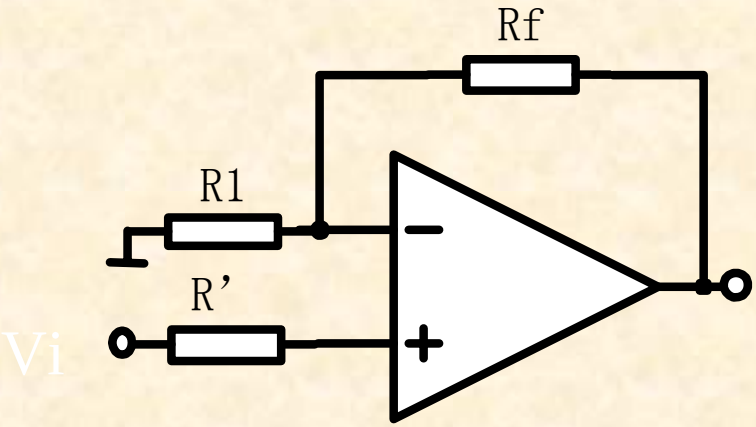
Ac = - Rf/R1

Ri = R1

Ro = 0

§ 2.1信号运算—比例放大电路

- 同相输入比例放大电路



$$V_o = (1 + R_f/R_1)\, V_i$$

约束条件：

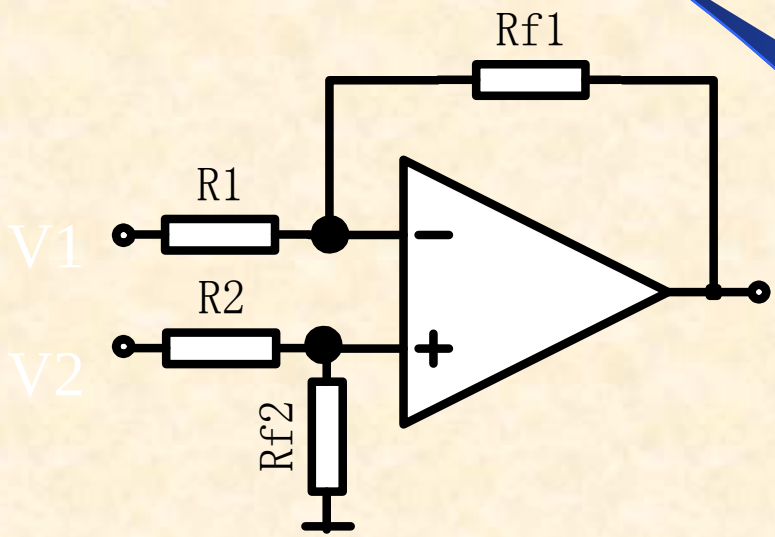
$$R' = R_1 // R_f$$

性能：

$$\begin{cases} A_c = 1 + R_f/R_1 \\ R_i = \infty \\ R_o = 0 \end{cases}$$

§2.1信号运算—比例放大电路

- 基本差动放大电路（单运放）



$$Vo=(Rf/R)Vid$$

$$Vid=V2-V1$$

约束条件：

$$R1=R2=R$$

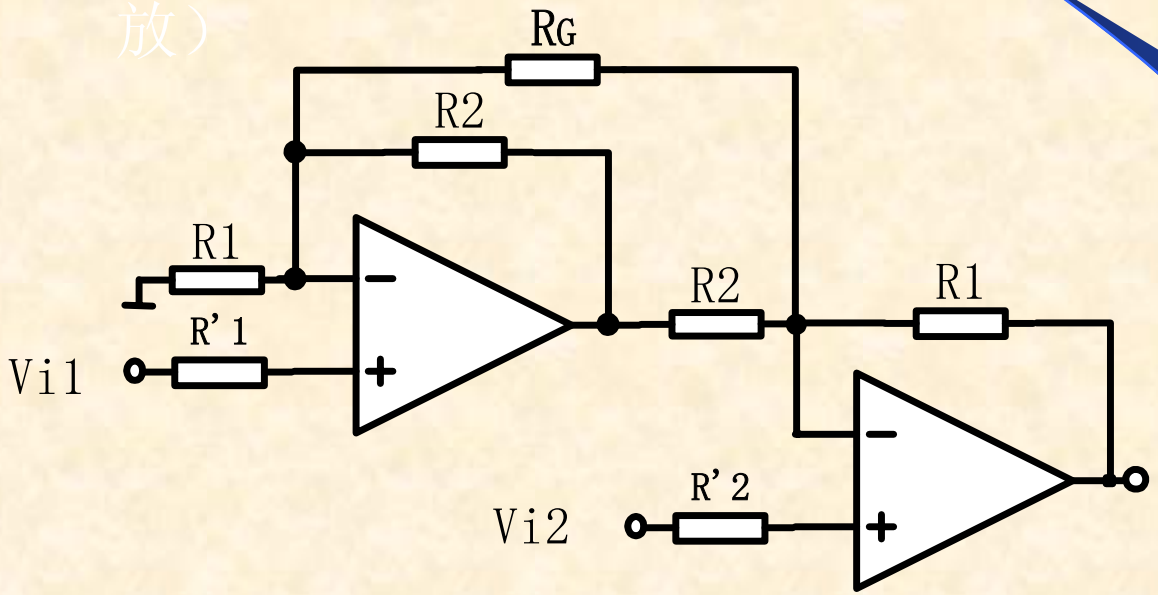
$$Rf1=Rf2=Rf$$

性能：

$$\begin{cases} Ac = Rf/R \\ Rid = R1+R2 = 2R \\ Ric = (R+Rf)/2 \\ Ro = 0 \end{cases}$$

§2.1信号运算—比例放大电路

- 同相串联差动放大电路（双运放）



Vid=Vi2-Vi1

Vo=(1+R1/R2+2R1/RG)Vid

约束条件：

R'1=R'2

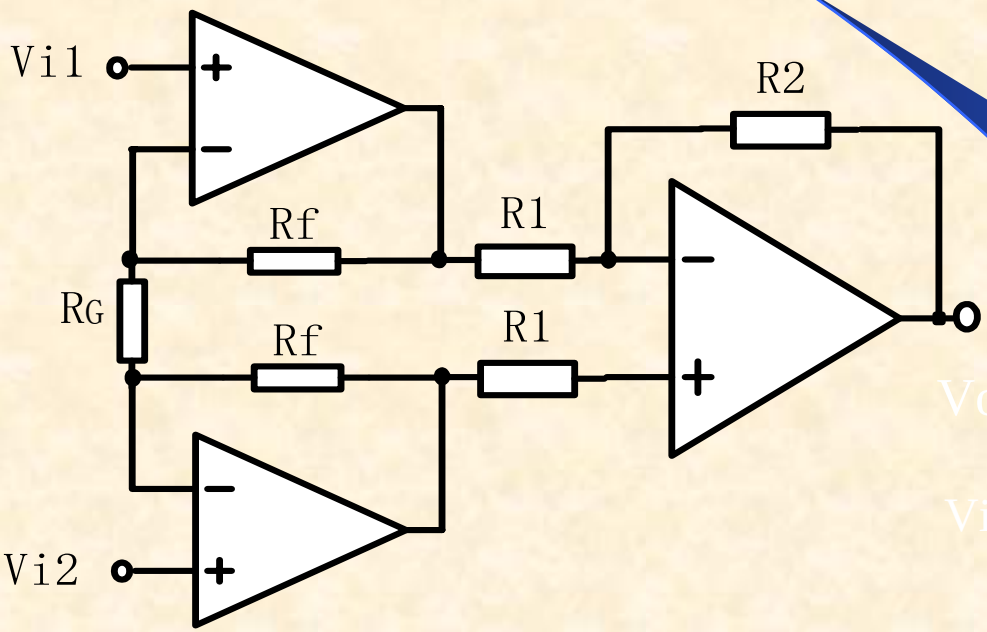
=R1//R2

性能：

- Ac = 1+R1/R2+2R1/RG
- Rid= 2Rc
- Ric= Rc/2
- Ro = 0

§2.1信号运算—比例放大电路

- 同相并联差动放大电路（3运放）



$V_o = (1+2R_f/R_G) \times R_2/R_1 \; V_{id}$

$V_{id} = V_2 - V_1$

约束条件：

运放CMMR匹配

性能：

$A_c = (1+2R_f/R_G)R_2/R_1$

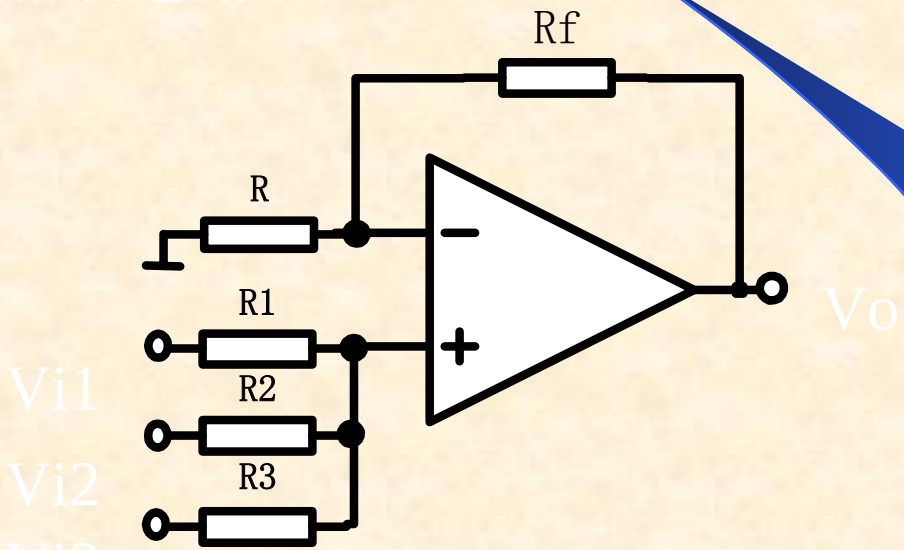
$R_{id} = 2R_c$

$R_{ic} = R_c/2$

$R_o = 0$

§2.1信号运算—加、减运算电路

- 同相输入加法电路



约束条件：

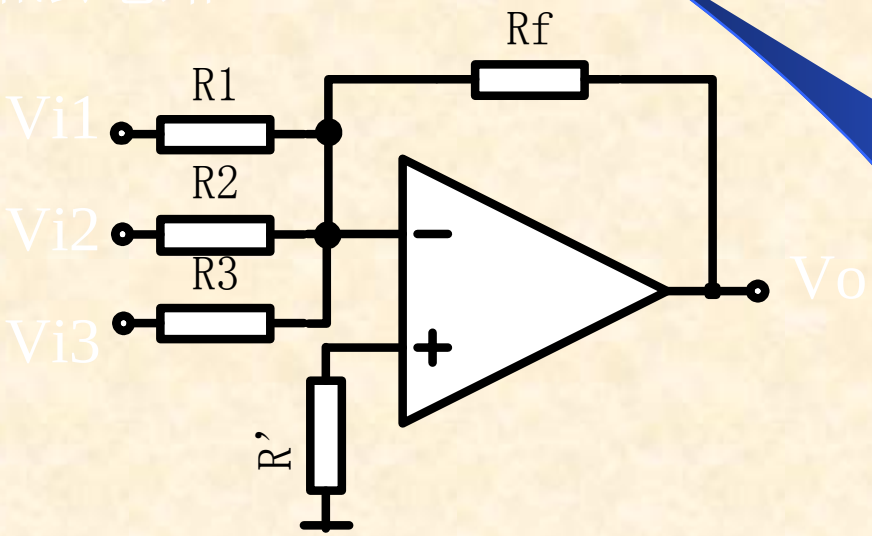
R//Rf’=R1//R2//R3

性能：

Vo = Rf（Vi1/R1+ Vi2/R2+ Vi3/R3）

§2.1信号运算—比例放大电路

- 反相输入加法电路



约束条件：

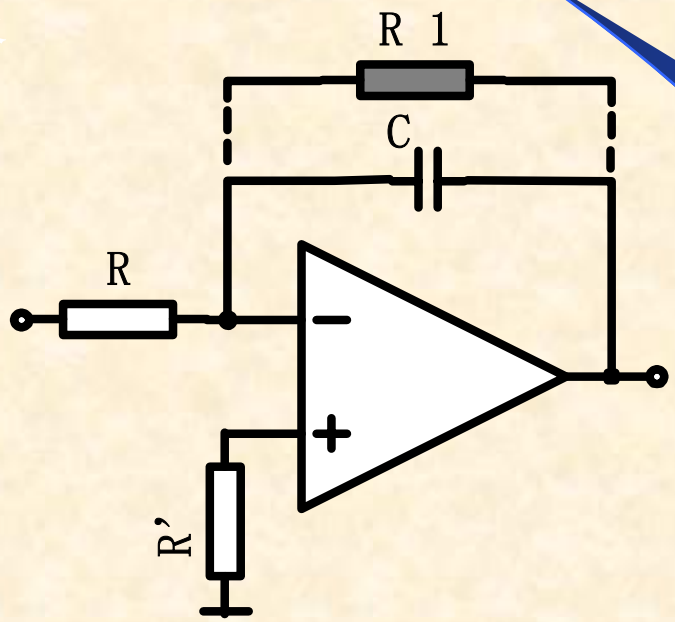
R’=R1//R2//R3//Rf

性能：

Vo = —Rf（Vi1/R1+ Vi2/R2+ Vi3/R3）

§2.1信号运算—积分、微分电路

- 基本积分电路



$Vo = -1/RC \int Vidt$

约束条件：

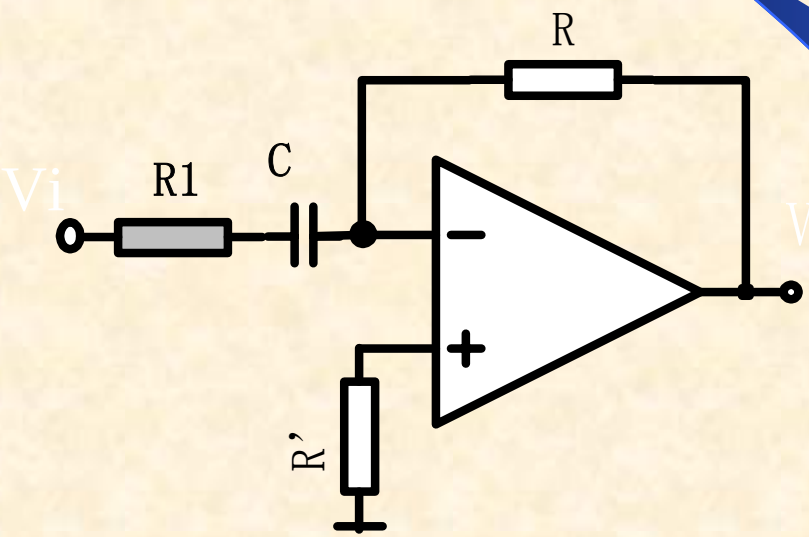
R’=R//R1

说明：

R1引入负反馈抑制漂移

§2.1信号运算—积分、微分电路

- 基本微分电路



$V_o = -RC \frac{dV_i}{dt}$

约束条件：

$R' = R // R1$

说明：

引入R1抑制高频噪声提高输入阻抗