第 5 章：汇编语言程序设计

井艳军

沈阳工业大学电气工程学院

主要内容

汇编语言指令格式

系统伪指令

存储器选择方式

常用子程序

PIC 单片机指令系统和其它单片机一样，是较为低级的语言系统，是一套控制和指挥 CPU 工作的编码，即机器语言。

单片机只能识别和执行由二进制数组成的机器语言，然而，这样一种二进制代码的机器语言是很难为人们直接理解和分析。

为了能较好表达人们的设计思路，便于记忆和使用，人们在低级语言之上设计出一种新的符号语言，即汇编语言。

汇编语言指令格式

汇编语言指令格式

标号	操作码 (指令助记符)	操作数	注释
label	opcode	operand	comment

根据指令的功能和作用，只有操作码是必须存在的，它主要决定了指令的操作性质，而其它部分是指令语句的重要补充和说明，有时可以缺省。

汇编语言指令格式

1、标号

标号位于指令助记符前面，它一般是用于表示指令所在的地址，例如表示主程序或子程序的起始地址、转移语句的入口地址等。

汇编语言指令格式

使用标号的要点是:

标号并不是指令的必须部分，只有那些欲被其他语句引用的指令之前，才必须附加标号。标号不一定和语句同行，可以单独在语句上方作为一行使用。

标号最多可以由 32 个字母、数字和其它一些字符组成，但第一个字符必须是字母或下划线。标号不能用系统保留字，即系统禁用指令助记符、寄存器名、标志符等作为标号，如：ADD、PCLATH 等。

汇编语言指令格式

使用标号的要点是:

一个标号只能表示一个地址，不允许多个地址用一个标号重复定义。

标号的定义和引用必须一致，其中的大小写可以混写但必须相同。

标号必须顶格书写，结束不用冒号。

汇编语言指令格式

2、操作码

操作码决定着指令的操作类型和操作性质，是汇编语言语句中的核心要素，每一条汇编指令都不可缺少，而其它三部分有些指令可以缺省。

汇编语言指令格式

有关操作码（指令助记符）的要点是:

操作码所对应的指令助记符，其中的符号大小写可以混写，而不会影响操作码的含意，这一点和标号、操作数符号变量的表达方式是有本质区别。

指令助记符不能顶格书写，当前面没有标号时，必须至少保留一个空格。

操作码核心助记符部分比较简单，初学者必须熟悉复合助记符部分的功能。

汇编语言指令格式

3、操作数

在 PIC 汇编语言语句中，操作数的形式和内容最为丰富，它是指令助记符操作的对象，一般以数据或地址的形式出现，也可以用符号变量所表示的数据或地址。

汇编语言指令格式

各种进制 168 的表示形式

进制	通用形式	默认形式 1	默认形式 2	特定形式
十六进制	H’A8’	0A8H	0A8	0xA8
十进制	D’168’	168D	168	.168
八进制	Q’250’	250Q	250	-
二进制	B’10101000’	10101000B	10101000	-

汇编语言指令格式

使用操作数的要点是:

若操作数有二项，中间应该用逗号（半角）分开。

以 A、B、C、D、E、F 开头的数，前面应加 0 作为引导。

MPASM 编辑环境默认进制为十六进制，也可按用户需要进行重新设置。

操作数部分的符号变量必须区分大小写。

重视 d 参数的应用，目标地址为: F（d=1）; W（d=0）。

汇编语言指令格式

4、注释

注释内容用分号引出，是汇编语言语句功能的一种补充说明，主要是便于人们阅读、分析、修改和程序的调试。

汇编语言指令格式

使用注释的要点是:

用（半角）分号引出注释内容，可以紧跟指令之后，也可以独立一行或多行书写，但每一行均需由分号引出。

注释内容可以英文书写，也能用中文书写（来源于文本编辑内容）。

系统伪指令

系统伪指令

各种单片机的汇编程序除了指令系统语句以外，一般都还定义许多非正式指令的语句，即伪指令。大多数伪指令汇编时并不产生机器码，仅为源程序提供汇编控制信息。

1、定位伪指令：ORG（Origin）

格式： ORG nnnn

说明： ORG 伪指令指出紧跟在该伪指令后的机器码指令的汇编地址，即经汇编后生成的机器码目标程序或数据块在单片机程序存储器中的起始存放地址。

1、定位伪指令：ORG（Origin）

例题 5-1

3 个程序段含义一样

```
1  ;-------------------------------------
2          ORG       0008H
3  START   MOVLW     00H
4  ;-------------------------------------
5  ABC     EQU       08H
6          ORG       ABC
7  START   MOVLW     00H
8  ;-------------------------------------
9  ABC     EQU       04H
10         ORG       ABC+4
11 START   MOVLW     00H
12 ;-------------------------------------
```

2、赋值伪指令：EQU（Equate）

格式： 符号名 EQU nn

说明： EQU 伪指令几乎每一个程序中都用到，其操作含意是使 EQU 两端的值相等。一般在 PIC 的程序设计中，原则上每次遇到新的符号参数，都必须在前面补充定义符号名的初始数值或存储器地址。

符号名一旦被 EQU 赋值，其值便不能被再重新定义。这里的符号名，既可以是 PIC 中的特殊功能寄存器、一个常数，或者是表示一个通用数据存储器地址。

2、赋值伪指令：EQU（Equate）

分析 ABC EQU 20H

实际上对于伪指令：ABC EQU 20H，其中 ABC 既可以认为是符号变量，因为 ABC 代表 20H 地址；又可以认为是符号常量，因为 ABC 可以代表符号常量 20H。

所以对于一个定义的符号量，应结合引用的指令进行分析才能真正确定符号量的类型。

2、赋值伪指令：EQU（Equate）

例题 5-2

正确区别符号变量和符号常量之间的关系

```
1     ABC       EQU       20H
2               ORG       0000H
3               NOP
4               MOVLW     77H
5               MOVWF     20H
6               MOVLW     88H
7               MOVF      ABC,0
8               MOVLW     ABC
9               NOP
10              END
```

3、程序结束伪指令：END

格式： END

说明： END 伪指令表示汇编语言源程序（*.ASM）的结束，MPASM 汇编器汇编时遇到 END 就认为程序已结束，对其后的程序段不再进行汇编。

4、列表选项伪指令：LIST

格式： LIST [可选项，可选项，···]

说明： LIST 伪指令用于设置各种汇编参数，以便控制整个汇编过程或对打印输出的列表文件进行格式化。

1）P = < 设置微控制器类型即单片机型号 >

例如：P = 16F877

2）R = < 定义默认的数值进位制的基数 >

例如：R = DEC（十进制）;

R = HEX（十六进制）;

R = BIN（二进制）;

默认为十六进制。

5、外调程序伪指令：INCLUDE

格式： INCLUDE “文件名”

说明： INCLUDE 伪指令的主要功能是将外部预先编写好的指定文件纳入本源程序的汇编内容，这样可以减少重复劳动，提高编程效率。

如：P16F877.INC 为 F877 单片机的复位矢量、专用寄存器的地址及其控制位和状态位的位地址的原始定义，有些参考书把 P16F877.INC 称为 F877 的头文件。

5、外调程序伪指令：INCLUDE

例题 5-3

通过循环变量 COUNTER，从 RD 端口输出二进制计数过程(00H~0FFH)，采用引入头文件的方法。

6、定义数据伪指令：DB(DW、DE、DATA)

格式： DB(DW、DE、DATA)< 表达式 >,< 表达式 >,···

说明： 数据定义伪指令用来为源程序中被处理的数据安排内存，赋予初值及定义名字的

6、定义数据伪指令：DB(DW、DE、DATA)

例题 5-4

不同定义数据伪指令方法，从 0100H、0200H、0300H、0400H 开始的数据块定义

```
1  ORG     0000H
2  NOP
3  ORG     0100H
4  DB      45H,67H,89H,0ABH,0CDH,0EFH
5  ORG     0200H
6  DB      4567H,89ABH,0CDEFH
7  ORG     0300H
8  DB      'A', 'B', 'C', 'D', 'E', 'F', 'G'
9  ORG     0400H
10 DB      "ABCDEFG"
```

6、定义数据伪指令：DB(DW、DE、DATA)

例题 5-4

不同定义数据伪指令方法，从 0100H、0200H、0300H、0400H 开始的数据块定义

```
1  ORG     0000H
2  NOP
3  ORG     0100H
4  DE      45H,67H,89H,0ABH,0CDH,0EFH
5  ORG     0200H
6  DE      4567H,89ABH,0CDEFH
7  ORG     0300H
8  DE      'A', 'B', 'C', 'D', 'E', 'F', 'G'
9  ORG     0400H
10 DE      "ABCDEFG"
```

6、定义数据伪指令：DB(DW、DE、DATA)

在 PIC16F877 单片机中:

DB 定义方式结果全部错误

DW 和 DATA 定义方式单字节和单字符方式正确

DE 定义方式功能最强，单字节、单字符和字符串三个方式正确

7、进制定义伪指令：RADIX

格式： RADIX < 进制表达式 >

说明： RADIX 伪指令用于设置在 MPLAB-ICD 集成开发环境中采用的进制方式，如定义十进制、八进制和十六进制等参数，MPLAB-ICD 集成开发系统缺省为十六进制。

例如：RADIX DEC ； 定义为十进制

RADIX HEX ； 定义为十六进制

RADIX OCT ； 定义为八进制

存储器选择方式

存储器选择方式

在 PIC 单片机中，有两个概念是令初学者感到头痛的事情，也是本课程的难点和重点，对于正确进行程序设计至关重要。

一个是数据存储器四体的体选方式，需要时刻考虑每一个访问的特殊功能寄存器和通用数据存储器的体位；另一个是程序存储器四页的页选方式，特别是在发生转移或跳转时，须密切注意是否会发生页面转换。

数据存储器体选方式

F877 单片机的数据存储器是一个具有空间为 512 字节的存储器，其中只有 19 个字节是无效存储单元。为了能完全选择 512 字节内的数据，需要 9 根地址线。

而根据 9 根地址线的组合方式不同，形成两种迥然不同的寻址方式：即直接寻址和间接寻址。

数据存储器体域

根据直接寻址和间接寻址操作码携带址址信息情况，一般把512 字节（包括无效地址）的数据存储器分成 4 个区域，在 PIC 中被称为“体”（BANK）。

体 0（000H～07FH）

体 1（080H～1FFH）

体 2（100H～17FH）

体 3（180H～1FFH）

1、直接寻址访问数据存储器

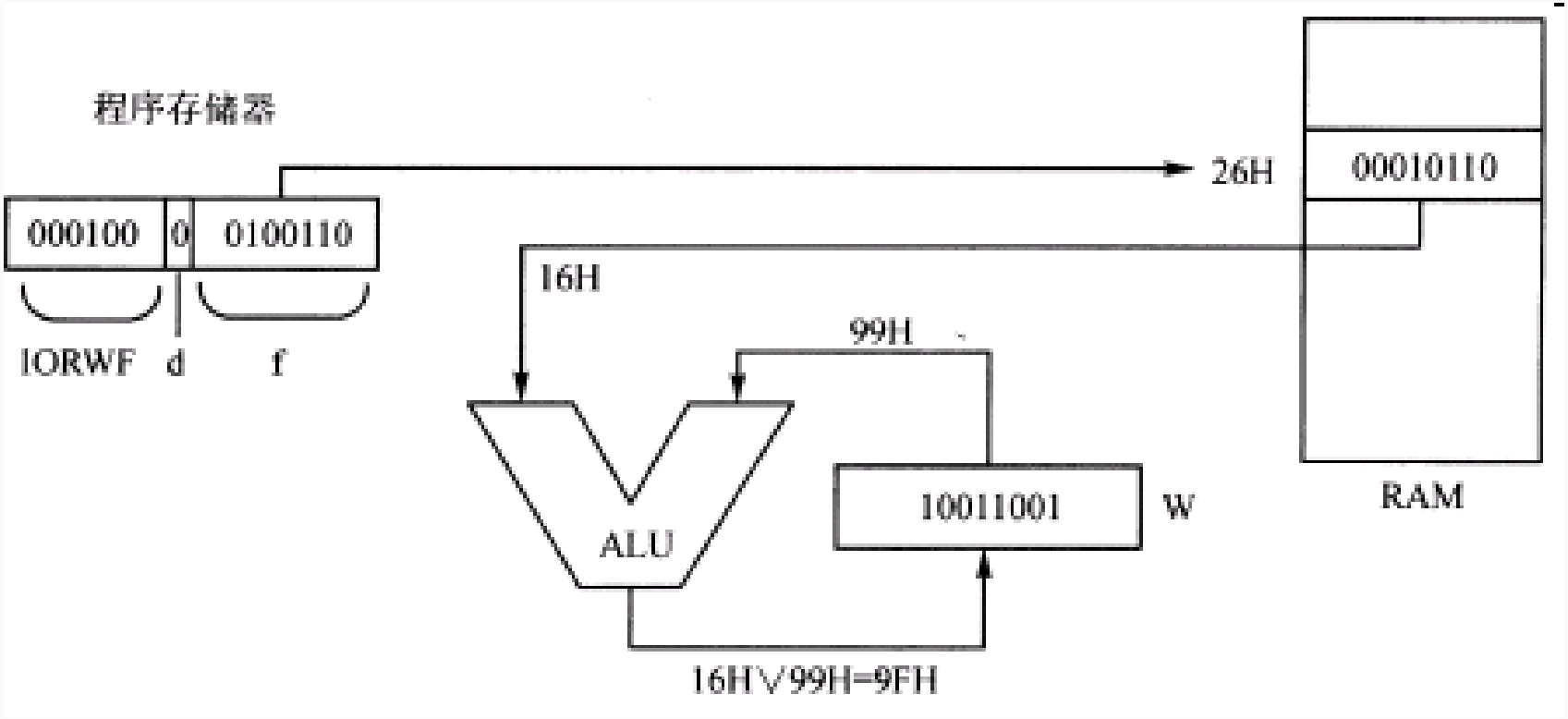
在指令机器码中操作数所携带的寻址信息是低 7 位地址，这不是一个完备的数据信息，每一个体中均会有一个相关的地址与之对应。要惟一确定地址单元，还必须依托其它的数据线进行复合选择。

利用状态标志 STATUS 位中的 RP1 和 RP0，与直接寻址机器码中低 7 位地址共同选择相应数据存储器的内容参与操作。

1、直接寻址访问数据存储器

例 IORWF 26H,W

假设 (26H)=0x16，(W)=0x99H 相或后，结果 9FH 送入 W 中 (d=0) 。参加运算的操作数 0x16 的单元地址 26H，可从指令中得到



2、间接寻址访问数据存储器

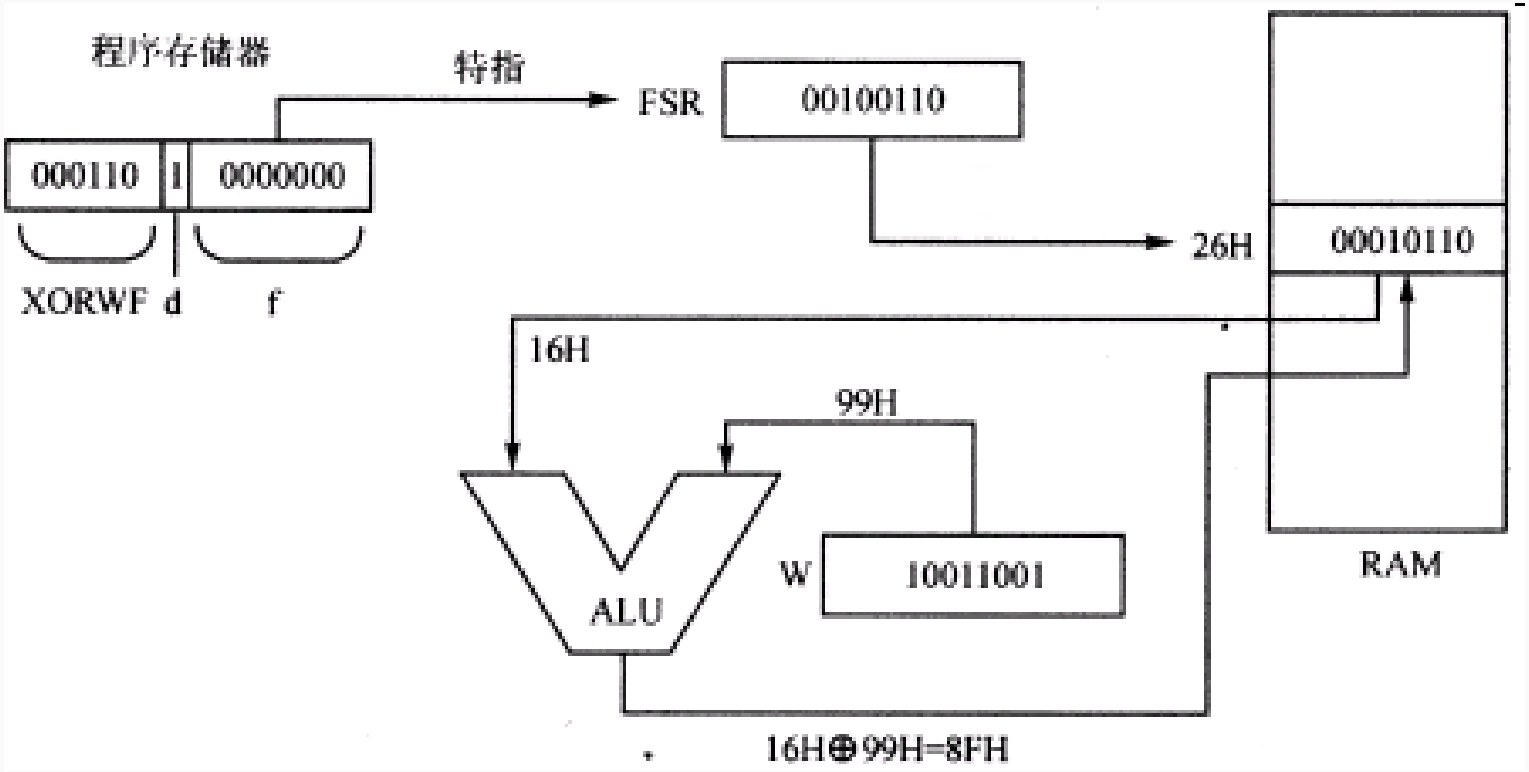
在指令机器码中真正携带的寻址信息是低 8 位地址，这也不是一个完备的数据信息，在整个数据存储器中有二个相关的地址与之对应。

要惟一确定地址单元，也必须依托另一根数据线进行复合选择。主要是依托状态标志位的 IRP，才能准确选择相应数据存储器的内容参与操作。

2、间接寻址访问数据存储器

例 XORWF INDF, F

从指令表面上看，指令是寻址 00H 单元，其实 00H 单元 (INDF) 是一个不存在的寄存器单元，只不过是将 00H 地址专门用以间接寻址。



3、立即寻址

在该寻址方式中，指令中包含了实际操作数，即操作数可在指令中直接获得，而不用到别处去寻找。

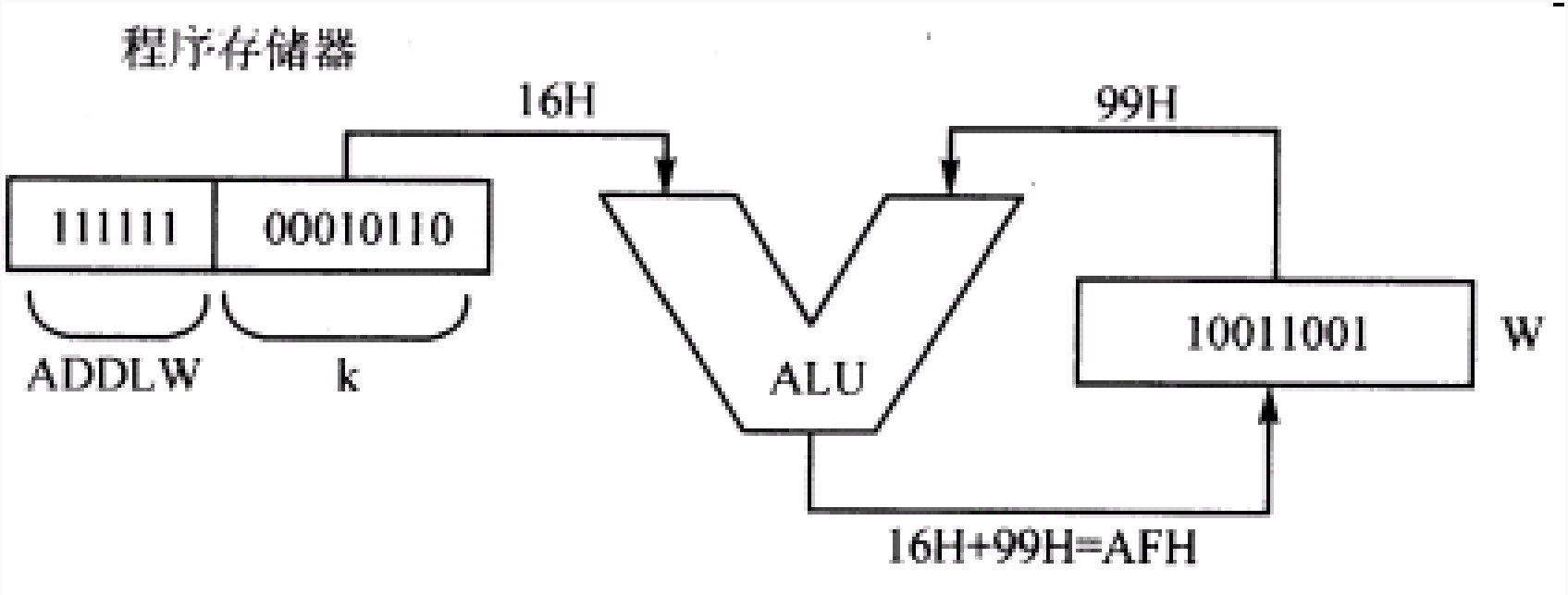
3、立即寻址

例 ADDLW 16H

将立即数 16H 与 W 内容 (假设为 99H) 相加，结果 (AFH) 送到 W。

该指令的二进制形式为：11111100010110

其中前 6 位是指令码，后 8 位就是操作数，如下图所示。



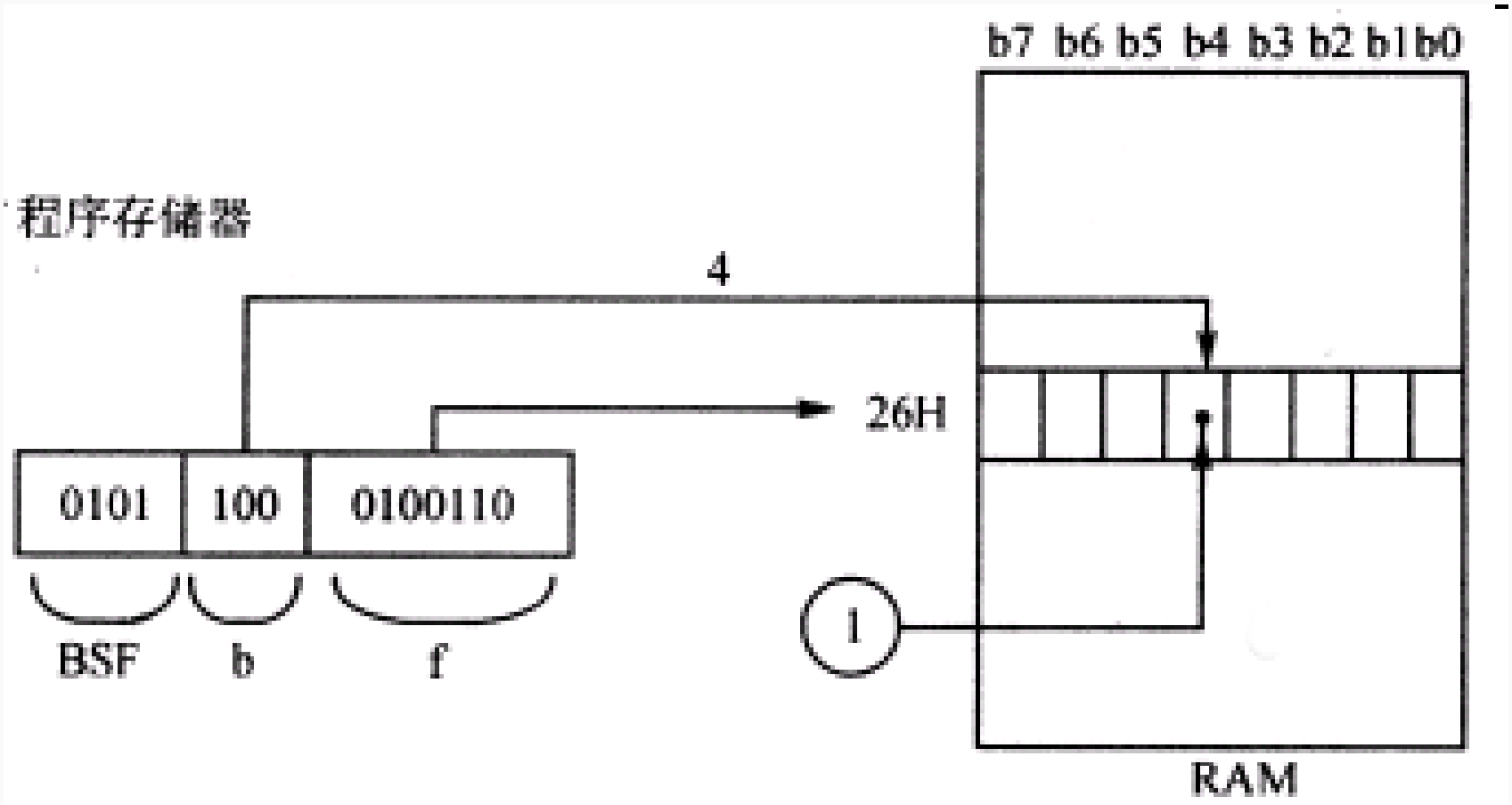
4、位寻址

可以对任一寄存器中的任一位直接寻址访问，即指令中既包含着被访问寄存器的地址，又包含着该寄存器中的位地址。

4、位寻址

例 BSF 26H, 4

把地址为 26H 的寄存器单元内的 bit4 置为 1，如下图所示。



体选伪指令：BANKSEL

```
1  ABC     EQU     20H
2  TEM     EQU     21H
3          ORG     0000H
4          NOP
5          BANKSEL ABC
6          MOVLW   00H
7          MOVWF   ABC
8          BANKSEL TEM
9          MOVLW   00H
10         MOVWF   TEM
```

程序存储器页选择方式

- 复位地址 0000H，直接给出 13 根地址选择线;

程序存储器页选择方式

- 复位地址 0000H，直接给出 13 根地址选择线；
- 中断地址 0004H，直接给出 13 根地址选择线；

程序存储器页选择方式

- 复位地址 0000H，直接给出 13 根地址选择线;
- 中断地址 0004H，直接给出 13 根地址选择线;
- 指令寄存器，是在每一个指令的执行周期自动加 1 而形成当前程序的执行方向;

程序存储器页选择方式

- 复位地址 0000H，直接给出 13 根地址选择线;
- 中断地址 0004H，直接给出 13 根地址选择线;
- 指令寄存器，是在每一个指令的执行周期自动加 1 而形成当前程序的执行方向;
- 执行以 PCL 为目标地址的算术逻辑类指令;

程序存储器页选择方式

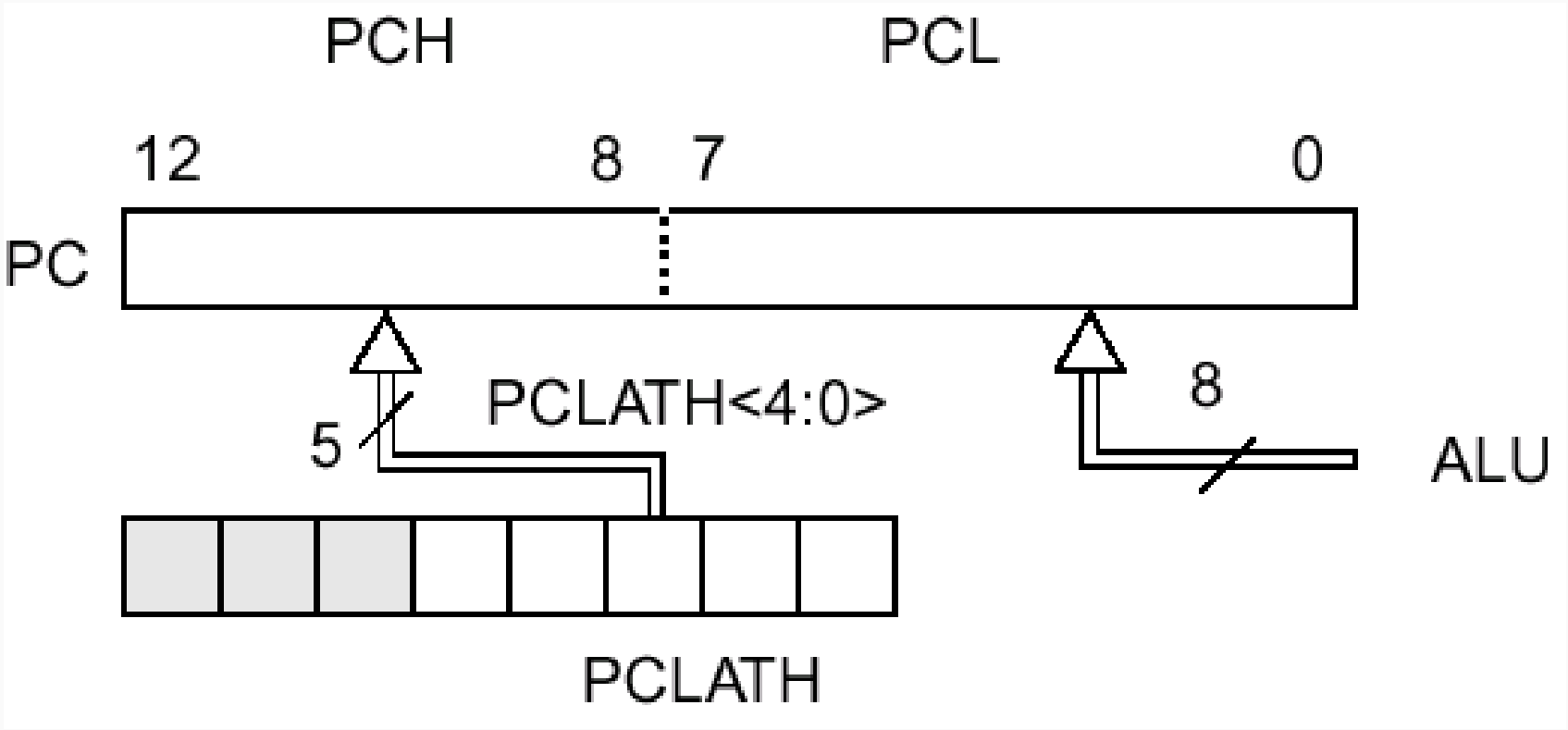
- 复位地址 0000H，直接给出 13 根地址选择线;
- 中断地址 0004H，直接给出 13 根地址选择线;
- 指令寄存器，是在每一个指令的执行周期自动加 1 而形成当前程序的执行方向;
- 执行以 PCL 为目标地址的算术逻辑类指令;
- 转移指令方式，即 GOTO 语句;

程序存储器页选择方式

- 复位地址 0000H，直接给出 13 根地址选择线;
- 中断地址 0004H，直接给出 13 根地址选择线;
- 指令寄存器，是在每一个指令的执行周期自动加 1 而形成当前程序的执行方向;
- 执行以 PCL 为目标地址的算术逻辑类指令;
- 转移指令方式，即 GOTO 语句;
- 调用子程序方式，即 CALL 语句以及相应的返回语句（RETRUN、RETFIE、RETLW）。

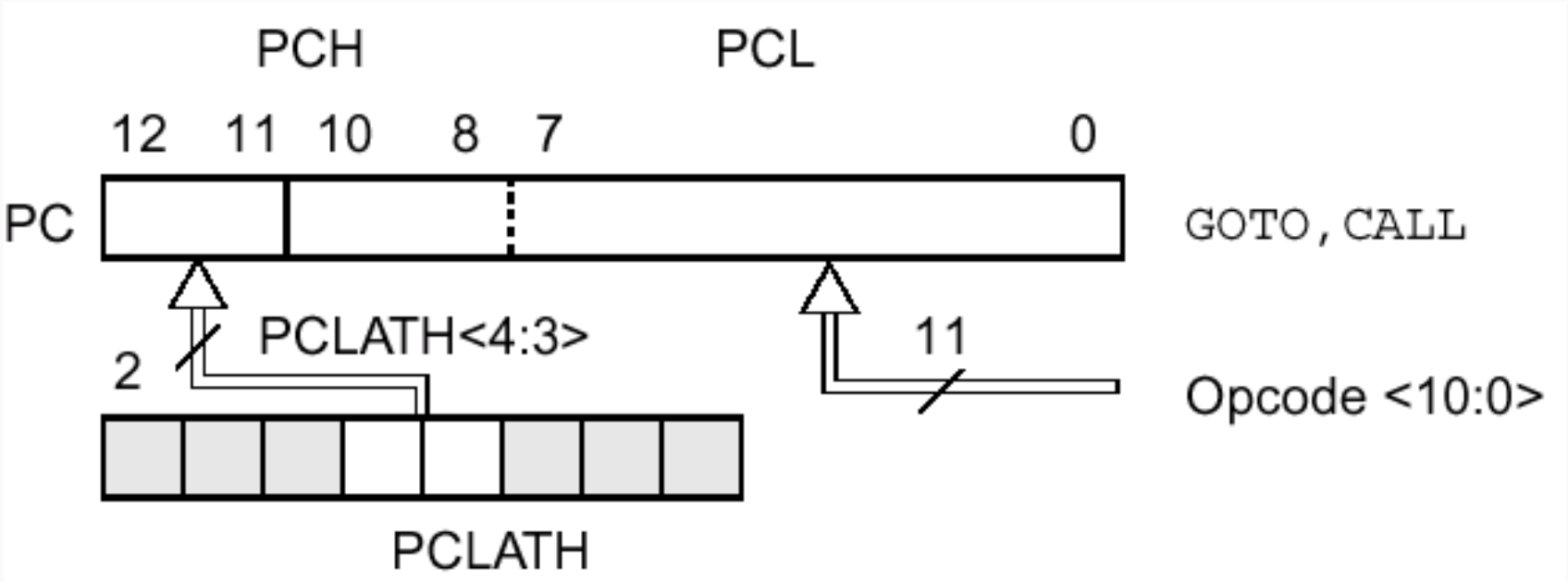
程序存储器页选择方式

执行以 PCL 为目标地址的算术逻辑类指令



程序存储器页选择方式

执行跳转指令（CALL、GOTO、RETURN、RETFIE、RETLW）



页选伪指令：PAGESEL

```
1                ORG         0100H
2                PAGESEL     ABC
3                GOTO        ABC
4
5                ORG         1000H
6   ABC          MOVLW       00H
7                MOVWF       20H
```

页选伪指令：PAGESEL

例题 5-5

利用散转查表方式编写程序，将任意 16 个数据（本例取规则数 00H～0FH）依次送入数据存储器 20H～2FH 中。

```
LIST      P=16F877
INCLUDE  "P16F877.INC"
COUNTER   EQU     30H
          ORG     0000H
          NOP
          MOVLW   20H
          MOVWF   FSR
          CLRF    COUNTER
LOOP      MOVF    COUNTER,W
          CALL    CHABIAO
          MOVWF   INDF
          INCF    COUNTER
          INCF    FSR
          BTFSS   COUNTER,4
          GOTO    LOOP
          GOTO    $
```

```
CHABIAO   ADDWF   PCL,F
          RETLW   00H
          RETLW   01H
          RETLW   02H
          RETLW   03H
          RETLW   04H
           ...
          RETLW   0AH
          RETLW   0BH
          RETLW   0CH
          RETLW   0DH
          RETLW   0EH
          RETLW   0FH
          END
```

```
LIST      P=16F877
INCLUDE  "P16F877.INC"
COUNTER   EQU     30H
          ORG     0000H
          NOP
          MOVLW   20H
          MOVWF   FSR
          CLRF    COUNTER
LOOP      MOVF    COUNTER,W
          CALL    CHABIAO
          MOVWF   INDF
          INCF    COUNTER
          INCF    FSR
          BTFSS   COUNTER,4
          GOTO    LOOP
          GOTO    $
```

```
          ORG     0100H
CHABIAO   ADDWF   PCL,F
          RETLW   00H
          RETLW   01H
          RETLW   02H
          RETLW   03H
          RETLW   04H
           ...
          RETLW   0AH
          RETLW   0BH
          RETLW   0CH
          RETLW   0DH
          RETLW   0EH
          RETLW   0FH
          END
```

```
LIST      P=16F877
INCLUDE "P16F877.INC"
COUNTER   EQU     30H
          ORG     0000H
          NOP
          MOVLW   20H
          MOVWF   FSR
          CLRF    COUNTER
LOOP      MOVF    COUNTER,W
          PAGESEL CHABIAO
          CALL    CHABIAO
          MOVWF   INDF
          INCF    COUNTER
          INCF    FSR
          BTFSS   COUNTER,4
          GOTO    LOOP
          GOTO    $
```

```
          ORG     1EF8H
CHABIAO   ADDWF   PCL,F
          RETLW   00H
          RETLW   01H
          RETLW   02H
          RETLW   03H
          RETLW   04H
           ...
          RETLW   0AH
          RETLW   0BH
          RETLW   0CH
          RETLW   0DH
          RETLW   0EH
          RETLW   0FH
          END
```

PAGESEL 飞溢

例题 5-5

```
1                SUBLW     08H
2                BTFSC     STATUS, C
3                GOTO      POP
4                BSF       PCLATH, 0
5                GOTO      WTO
6   POP          BCF       PCLATH, 0
7   WTO          BSF       PCLATH, 1
8                BSF       PCLATH, 2
9                MOVF      COUNTER, W
10               PAGESEL   CHABIAO
```

常用子程序

程序格式

源程序的编写格式并没有专门的规定，但学习编程时养成一种良好的习惯，对以后会有很大的裨益。以下是一个完整程序的总体布局以供参考。

```
;-------------------------------------------------------------
; 符号名定义和变量定义
;-------------------------------------------------------------
INDF        EQU     00H         ; 把后面程序的指令中将要用到的
TMR0        EQU     01H         ; 寄存器单元地址和位地址
PCL         EQU     02H         ; 用表义性很强的符号名预先定义
X           EQU     20H         ; 对所需的变量进行定义
Y           EQU     21H
```

程序格式

```
;-------------------------------------------------------------
;复位矢量和中断矢量安排(对于16F87X)
;-------------------------------------------------------------
        ORG     0000H           ;地址0000H为复位矢量
        NOP
        GOTO    MAIN            ;跳转到主程序
        ORG     0004H           ;地址0004H为中断矢量
        GOTO    INT_BODY        ;跳转到中断服务程序
;-------------------------------------------------------------
;主程序区
;-------------------------------------------------------------
        ORG     0005H           ;从0005H开始存放主程序
MAIN    CLRW
        CALL    SUB
        …… 
```

程序格式

```
;-------------------------------------------------------------
;子程序区
;-------------------------------------------------------------
SUB     MOVLW     01H       ;子程序
        ……
        RETURN              ;子程序返回
;-------------------------------------------------------------
;中断服务程序区
;-------------------------------------------------------------
INT_BODY                    ;中断服务程序
        MOVLW     0FFH
        RETFIE              ;中断服务程序返回
;-------------------------------------------------------------
        END                 ;全部程序结束
```

常用子程序

在程序设计中，除主程序以外还有一部分很重要的内容就是关于子程序的设计，它是为完成特定的目的而构成的复合程序。

跳转和循环子程序

转移和循环程序，主要是通过跳转、判断和位测试指令来构成的。

1. **跳转指令** GOTO
2. **判断指令** INCFSZ、DECFSZ
3. **位测试指令** BTFSS、BTFSC

跳转和循环子程序

例题 5-6

假定执行某个显示功能 100 次后结束工作，显示子程序是 XSH

```
        ORG     0000H
        MOVLW   D'101'
        MOVWF   20H
LOOP    DECFSZ  20H, F
        GOTO    RRT
        GOTO    PPY
RRT     PAGESEL XSH
        CALL    XSH
        PAGESEL LOOP
        GOTO    LOOP
PPY     END
```

跳转和循环子程序

例题 5-7

比较两个数据寄存器 20H 和 30H 内容的大小，将较大的数送入 40H 中。

```
        MOVF    30H, W
        SUBWF   20H, W
        BTFSC   STATUS, C
        GOTO    L20H
        MOVF    30H, W
        MOVWF   40H
        GOTO    POP
L20H    MOVF    20H, W
        MOVWF   40H
POP     END
```

软件延时子程序

单片机的延时程序，在程序设计中具有很重要的地位。延时的设计，一般可以通过两种方式：硬件延时和软件延时。

所谓硬件延时，就是由单片机系统的定时器实现；而软件延时，是通过循环程序实现。一般来说，前者适用于精确定量延时，而后者常用于初略定性延时。

软件延时子程序

例题 5-8

简单循环的软件延时子程序。

```
COUNTER     EQU     20H
            ORG     0000H
DELAY       MOVLW   0FFH
            MOVWF   COUNTER
LOOP        DECFSZ  COUNTER
            GOTO    LOOP
            RETURN
```

软件延时子程序

例题 5-9

请编写 10ms 软件延时子程序。

```
DEL10MS     MOVLW     0DH
            MOVWF     20H
LOOP1       MOVLW     0FFH
            MOVWF     21H
LOOP2       DECFSZ    21H
            GOTO      LOOP2
            DECFSZ    20H
            GOTO      LOOP1
            RETURN
```

软件延时子程序

例题 5-10

请编写 1s 软件延时子程序。

```
DELAY1S     MOVLW     06H
            MOVWF     20H
LOOP1       MOVLW     0EBH
            MOVWF     21H
LOOP2       MOVLW     0ECH
            MOVWF     22H
LOOP3       DECFSZ    22H
            GOTO      LOOP3
            DECFSZ    21H
            GOTO      LOOP2
            DECFSZ    20H
            GOTO      LOOP1
            RETURN
```

数据查表子程序

数据查表子程序在某些特殊场合是非常有用的，如共阴极 LED 八段显示器以及其它具有固定显示模式的场合，需根据其显示数值去查找对应参考数据库编码输出。

数据查表子程序

例题 5-11

将 RC 端口与共阴极 LED 八段显示器相连，从 0-9 循环显示，间隔时间为 1 秒，请编写相应的软件程序。

数值	编码	数值	编码
1	06H	6	7DH
2	5BH	7	07H
3	4FH	8	7FH
4	66H	9	6FH
5	6DH	0	3FH

```
ABC       EQU     30H
          ORG     0000H
          BSF     STATUS, RP0
          CLRF    TRISC
          BCF     STATUS, RP0
MAIN      MOVLW   00H
          MOVWF   ABC
LOOP      MOVF    ABC, W
          CALL    CHABIAO
          MOVWF   PORTC
          CALL    DELAY1S
          INCF    ABC
          MOVLW   09H
          SUBWF   ABC, W
          BTFSS   STATUS, Z
          GOTO    LOOP
          GOTO    MAIN
```

```
CHABIAO   ADDWF   PCL,F
          RETLW   3FH
          RETLW   06H
          RETLW   5BH
          RETLW   4FH
          RETLW   66H
          RETLW   6DH
          RETLW   7DH
          RETLW   07H
          RETLW   7FH
          RETLW   6FH
DELAY1S   MOVLW   06H
          MOVWF   20H
           ...
          RETURN
          END
```

分支功能跳转子程序

在 PIC 指令系统中并没有类似的语句，但如果借助于 PIC 单片机指令的特殊功能，同样可以轻松地构成分支跳转。

分支跳转实际上是多条件判断指令，条件本身是一个整数或事件，而跳转出口应该是整数的信息返回或事件功能内容的具体表现。

在程序形式上，分支功能跳转子程序与数据查表子程序的结构类似，只是它是用 GOTO 语句替代了 RETLW 语句。

分支功能跳转子程序

例题 5-14

试编写 N 个键盘功能选择子程序

```
                PAGESEL   KEY
                CALL      KEY
                PAGESEL   JIAN
                CALL      JIAN
;-------------------------------------------------
; 根据键情况，确定相应子程序
;-------------------------------------------------
JIAN            ADDWF     PCL, F
                GOTO      PKEY0
                GOTO      PKEY1
                GOTO      PKEY2
                 ...
                GOTO      PKEYN
```

常用数学运算类子程序

本节给出一些常用的数学运算类子程序，主要有加、减、乘、除等子程序，还有为外扩系统设计中的数码显示所需的 BCD 码和二进制数据的互换子程序。

均涉及入口条件及出口条件，在调用时务必加以注意。一般入口条件是指参与操作的相关源数据，用 S1、S2（source）等表示，而操作结果用 R1、R2（result）等表示。高低 8 位数据分别用 H、L 表示，另外用 Z 表示中 8 位数据。

常用数学运算类子程序

可以将这些固定变量参数定义在 PIC16F877 的头文件内，并且单元定义位于映射区域 70H 7FH，这样就可以不受程序所在页面的影响。

常用数学运算类子程序

例题 5-15

将两个无符号 16 位数相加，编写双精度运算程序。

```
INCLUDE "P16F877.INC"
S1H         EQU     50H
S1L         EQU     51H
S2H         EQU     52H
S2L         EQU     53H
R1H         EQU     54H
R1L         EQU     55H
            ORG     0000H
            NOP
            MOVLW   12H
            MOVWF   S1H
            MOVLW   34H
            MOVWF   S1L
            MOVLW   57H
            MOVWF   S2H
            MOVLW   78H
            MOVWF   S2L
            CALL    ADDXY
            GOTO    $
ADDXY       MOVF    S1L,W
            ADDWF   S2L,F
            BTFSC   STATUS,C
            INCF    S2H
            MOVF    S1H,W
            ADDWF   S2H,W
            MOVWF   R1H
            MOVF    S2L,W
            MOVWF   R1L
            RETLW   00H
            END
```

常用数学运算类子程序

例题 5-16

将两个无符号 16 位数相减，编写双精度运算程序。

```
INCLUDE "P16F877.INC"
S1H     EQU     50H
S1L     EQU     51H
S2H     EQU     52H
S2L     EQU     53H
R1H     EQU     54H
R1L     EQU     55H
        ORG     0000H
        NOP
        MOVLW   56H
        MOVWF   S1H
        MOVLW   78H
        MOVWF   S1L
        MOVLW   12H
        MOVWF   S2H
        MOVLW   34H
        MOVWF   S2L
        CALL    SUBXY
        GOTO    $
SUBXY   COMF    S2L,F
        INCF    S2L,F
        BTFSC   STATUS,Z
        DECF    S2H,F
        COMF    S2H,F
ADDXY   MOVF    S1L,W
        ADDWF   S2L,F
        ...
        MOVWF   R1H
        MOVF    S2L,W
        MOVWF   R1L
        RETLW   00H
        END
```

常用数学运算类子程序

例题 5-17

将两个无符号 16 位数相乘，编写双精度运算程序。

```
INCLUDE "P16F877.INC"
S1H     EQU     50H
S1L     EQU     51H
S2H     EQU     52H
S2L     EQU     53H
R1H     EQU     54H
R1L     EQU     55H
R2H     EQU     56H
R2L     EQU     57H
P1H     EQU     58H
P1L     EQU     59H
COUNT   EQU     5AH
```

```
        ORG     0000H
        NOP
        MOVLW   12H
        MOVWF   S1H
        MOVLW   34H
        MOVWF   S1L
        MOVLW   56H
        MOVWF   S2H
        MOVLW   78H
        MOVWF   S2L
CALL    MPXY
GOTO    $
```

```
MPXY    CALL    YIWEI
MPLOOP  RRF     P1H
        RRF     P1L
        BTFSC   STATUS,C
        CALL    MPADD
        RRF     S2H
        RRF     S2L
        RRF     R2H
        RRF     R2L
        DECFSZ  COUNT
        GOTO    MPLOOP
        MOVF    S2H,W
        MOVWF   R1H
        MOVF    S2L,W
        MOVWF   R1L
        RETLW   00H
```

```
;-------------------------------------------------------
; 16 次右移
;-------------------------------------------------------
YIWEI   MOVLW     10H
        MOVWF     COUNT
        MOVF      S2H ,W
        MOVWF     P1H
        MOVF      S2L ,W
        MOVWF     P1L
        CLRF      S2H
        CLRF      S2L
        RETLW     00H
```

```
;-------------------------------------------------------
;加法子程序
;-------------------------------------------------------
MPADD   MOVF    S1L,W
        ADDWF   S2L,F
        BTFSC   STATUS,C
        INCF    S2H,F
        MOVF    S1H,W
        ADDWF   S2H,F
        RETLW   00H
        END
```

例题 5-18

将两个无符号 16 位数相除，编写双精度运算程序。

```
INCLUDE "P16F877.INC"
S1H     EQU     50H
S1L     EQU     51H
S2H     EQU     52H
S2L     EQU     53H
R1H     EQU     54H
R1L     EQU     55H
R2H     EQU     56H
R2L     EQU     57H
P1H     EQU     58H
P1L     EQU     59H
COUNT   EQU     5AH
```

```
        ORG     0000H
        NOP
        MOVLW   12H
        MOVWF   S1H
        MOVLW   34H
        MOVWF   S1L
        MOVLW   67H
        MOVWF   S2H
        MOVLW   89H
        MOVWF   S2L
CALL    DIVXY
GOTO    $
```

```
DIVXY     CALL     YIWEI
          CLRF     R2H
          CLRF     R2L
DIVLOOP   BCF      STATUS,C
          RLF      P1L
          RLF      P1H
          RLF      R2L
          RLF      R2H
          MOVF     S1H,W
          SUBWF    R2H,W
          BTFSS    STATUS,Z
          GOTO     ASP
          MOVF     S1L,W
          SUBWF    R2L,W
```

```
ASP     BTFSS   STATUS,C
        GOTO    PUP
        MOVF    S1L,W
        SUBWF   R2L,F
        BTFSS   STATUS,C
        DECF    R2H,F
        MOVF    S1H,W
        SUBWF   R2H,F
        BSF     STATUS,C
PUP     RLF     S2L
        RLF     S2H
        DECFSZ  COUNT
        GOTO    DIVLOOP
        MOVF    S2H,W
        MOVWF   R1H
        MOVF    S2L,W
        MOVWF   R1L
        RETLW   00H
```

```
;-------------------------------------------------------
; 16 次右移
;-------------------------------------------------------
YIWEI   MOVLW     10H
        MOVWF     COUNT
        MOVF      S2H ,W
        MOVWF     P1H
        MOVF      S2L ,W
        MOVWF     P1L
        CLRF      S2H
        CLRF      S2L
        RETLW     00H
```

例题 5-19

将一个 5 位数 (<65536) 的 BCD 码转换成二进制数。

通过将 BCD 码数向右逐位移到二进制数内，检查每一个数是否大于 7，如果是，则在该位减 3。

```
INCLUDE "P16F877.INC"
S1H     EQU     50H
S1Z     EQU     51H
S1L     EQU     52H
R1H     EQU     53H
R1L     EQU     54H
COUNT   EQU     55H
```

```
ORG     0000H
NOP
MOVLW   01H
MOVWF   S1H
MOVLW   23H
MOVWF   S1Z
MOVLW   45H
MOVWF   S1L
CALL    BCD2BIN
GOTO    $
```

```
BCD2BIN  MOVLW    10H
         MOVWF    COUNT
         CLRF     R1H
         CLRF     R1L
LOOP     BCF      STATUS, C
         RRF      S1H, F
         RRF      S1Z, F
         RRF      S1L, F
         RRF      R1H, F
         RRF      R1L, F
         DECFSZ   COUNT, F
         GOTO     ADJDCT
         RETLW    00H
```

```
ADJDCT  MOVLW   S1L
        MOVWF   FSR
        CALL    ADJBIN
        MOVLW   S1Z
        MOVWF   FSR
        CALL    ADJBIN
        MOVLW   S1H
        MOVWF   FSR
        CALL    ADJBIN
        GOTO    LOOP
```

```
ADJBIN   MOVLW    03H
         BTFSC    INDF, 3
         SUBWF    INDF, F
         MOVLW    30H
         BTFSC    INDF, 7
         SUBWF    INDF, F
         RETLW    00H
         END
```

常用数学运算类子程序

例题 5-20

将一个 16 位二进制数转换成 BCD 码 (<65535)。

通过将 16 位二进制数向左逐位移到 BCD 数内，检查每一个 BCD 码是否大于 4，如果是，则在该位加 3。

```
S1H     EQU     50H
S1L     EQU     51H
R1H     EQU     52H
R1Z     EQU     53H
R1L     EQU     54H
COUNT   EQU     55H
TEMP    EQU     56H
```

```
ORG     0000H
NOP
MOVLW   12H
MOVWF   S1H
MOVLW   34H
MOVWF   S1L
CALL    BIN2BCD
GOTO    $
```

```
BIN2BCD  MOVLW    10H
         MOVWF    COUNT
         CLRF     R1H
         CLRF     R1Z
         CLRF     R1L
LOOP     RLF      S1L, F
         RLF      S1H, F
         RLF      R1L, F
         RLF      R1Z, F
         RLF      R1H, F
         DECFSZ   COUNT, F
         GOTO     ADJDET
         RETLW    00H
```

```
ADJDET   MOVLW    R1L
         MOVWF    FSR
         CALL     ADJBCD
         MOVLW    R1Z
         MOVWF    FSR
         CALL     ADJBCD
         MOVLW    R1H
         MOVWF    FSR
         CALL     ADJBCD
         GOTO     LOOP
```

```
ADJBCD   MOVLW    03H
         ADDWF    INDF, W
         MOVWF    TEMP
         BTFSC    TEMP, 3
         MOVWF    INDF
         MOVLW    30H
         ADDWF    INDF, W
         MOVWF    TEMP
         BTFSC    TEMP, 7
         MOVWF    INDF
         RETLW    00H
         END
```

常用数学运算类子程序

练习

将一个 8 位二进制数开平方。

例题 5-21

将一个 16 位二进制数开平方。

```
S1H     EQU     50H
S1L     EQU     51H
R1L     EQU     52H
COUNT   EQU     53H
TEMP1   EQU     54H
TEMP2   EQU     55H
```

```
ORG     0000H
NOP
MOVLW   02H
MOVWF   S1H
MOVLW   71H
MOVWF   S1L
CALL    SQRT
GOTO    $
```

```
SQRT        MOVLW     0F0H
            ANDWF     S1H, W
            BTFSC     STATUS, Z
            GOTO      KF01
            MOVLW     10H
            SUBWF     S1H, F
            MOVLW     81H
            MOVWF     TEMP1
            GOTO      KF02
KF01          MOVLW     01H
            MOVWF     TEMP1
```

```
KF03    MOVF    TEMP1, W
        SUBWF   S1L, F
        MOVF    TEMP2, W
        BTFSS   STATUS, C
        ADDLW   01H
        SUBWF   S1H, F
        BTFSS   STATUS, C
        GOTO    KF04
        MOVLW   02H
        ADDWF   TEMP1, F
        BTFSC   STATUS, C
        INCF    TEMP2, F
        GOTO    KF03
```

```
KF04        BCF       STATUS, C
            RRF       TEMP2, W
            MOVWF     S1H
            RRF       TEMP1, W
            MOVWF     S1L
            MOVWF     R1L
            RETURN
            END
```