
第 5 章 CPU 和 ALU

目录

本章包括下面一些主要内容：

5.1	简介	5-2
5.2	指令的一般格式	5-4
5.3	中央处理单元 (CPU)	5-4
5.4	指令时钟	5-4
5.5	算术逻辑单元 (ALU)	5-5
5.6	状态寄存器	5-6
5.7	OPTION_REG 寄存器	5-8
5.8	电源控制寄存器	5-9
5.9	设计技巧	5-10
5.10	相关应用笔记	5-11
5.11	版本历史	5-12

5.1 简介

中央处理单元（CPU）通过执行程序存储器中的信息（指令）来控制器件的运行。其中许多指令是对数据存储器进行操作。对数据存储器的操作需要使用算术逻辑单元（ALU）。除了执行算术和逻辑操作外，ALU 还控制状态位（在状态寄存器中）。一些指令的执行结果会根据结果的状态而改变状态位。

CPU 的机器码如表 5-1 中所示（MPASM™ 用来产生这些机器码的指令助记符也列在表中）。

表 5-1: 中档系列单片机指令集

助记符, 操作数	描述	周期	14 位宽指令字				影响的状 态位	备注	
			最高 位			最低位			
针对字节的数据寄存器操作									
ADDWF	f, d	W 加 f	1	00	0111	dfff	ffff	C,DC,Z	1,2
ANDWF	f, d	W 和 f 与运算	1	00	0101	dfff	ffff	Z	1,2
CLRF	f	f 清零	1	00	0001	1fff	ffff	Z	2
CLRW	-	W 清零	1	00	0001	0xxx	xxxx	Z	
COMF	f, d	f 取反	1	00	1001	dfff	ffff	Z	1,2
DECF	f, d	f 减 1	1	00	0011	dfff	ffff	Z	1,2
DECFSZ	f, d	f 减 1, 为 0 则跳转	1(2)	00	1011	dfff	ffff		1,2,3
INCF	f, d	f 增 1	1	00	1010	dfff	ffff	Z	1,2
INCFSZ	f, d	f 增 1, 为 0 则跳转	1(2)	00	1111	dfff	ffff		1,2,3
IORWF	f, d	W 和 f 或运算	1	00	0100	dfff	ffff	Z	1,2
MOVF	f, d	f 送到 d	1	00	1000	dfff	ffff	Z	1,2
MOVWF	f	W 送到 f	1	00	0000	1fff	ffff		
NOP	-	空操作	1	00	0000	0xx0	0000		
RLF	f, d	f 循环左移	1	00	1101	dfff	ffff	C	1,2
RRF	f, d	f 循环右移	1	00	1100	dfff	ffff	C	1,2
SUBWF	f, d	f 减去 W	1	00	0010	dfff	ffff	C,DC,Z	1,2
SWAPF	f, d	f 半字节交换	1	00	1110	dfff	ffff		1,2
XORWF	f, d	W 和 f 异或运算	1	00	0110	dfff	ffff	Z	1,2
针对位的数据寄存器操作									
BCF	f, b	清除 f 的 bit b	1	01	00bb	bfff	ffff		1,2
BSF	f, b	置 f 的 bit b	1	01	01bb	bfff	ffff		1,2
BTFSC	f, b	测试 f 的 bit b, 为 0 则跳转	1 (2)	01	10bb	bfff-	ffff		3
BTFSS	f, b	测试 f 的 bit b, 为 1 则跳转	1 (2)	01	11bb	bfff	ffff		3
立即数操作和控制操作									
ADDLW	k	立即数加 W	1	11	111x	kkkk	kkkk	C,DC,Z	
ANDLW	k	立即数和 W 与运算	1	11	1001	kkkk	kkkk	Z	
CALL	k	调用子程序	2	10	0kkk	kkkk	kkkk		
CLRWDT	-	清除看门狗定时器	1	00	0000	0110	0100	$\overline{TO,PD}$	
GOTO	k	跳转	2	10	1kkk	kkkk	kkkk		
IORLW	k	立即数和 W 或运算	1	11	1000	kkkk	kkkk	Z	
MOVLW	k	立即数送到 W	1	11	00xx	kkkk	kkkk		
RETFIE	-	中断返回	2	00	0000	0000	1001		
RETLW	k	立即数送到 W 的子程序返回	2	11	01xx	kkkk	kkkk		
RETURN	-	子程序返回	2	00	0000	0000	1000		
SLEEP	-	进入休眠状态	1	00	0000	0110	0011	$\overline{TO,PD}$	
SUBLW	k	立即数减 W	1	11	110x	kkkk	kkkk	C,DC,Z	
XORLW	k	立即数和 W 异或运算	1	11	1010	kkkk	kkkk	Z	

注 1: 当 I/O 寄存器用自身内容修改自己时 (如 MOVF PORTB, 1), 使用的值是引脚上的当前值。例如, 某个设置成输入的引脚, 其数据锁存器中的值为 “1”, 但此时被外部器件拉为低电平, 执行指令后, 数据锁存器的值变为 “0”。

2: 如果预分频器分配给 Timer0 模块, 对 TMR0 寄存器执行该指令 (且适用时 d=1) 时, 将清零预分频器。

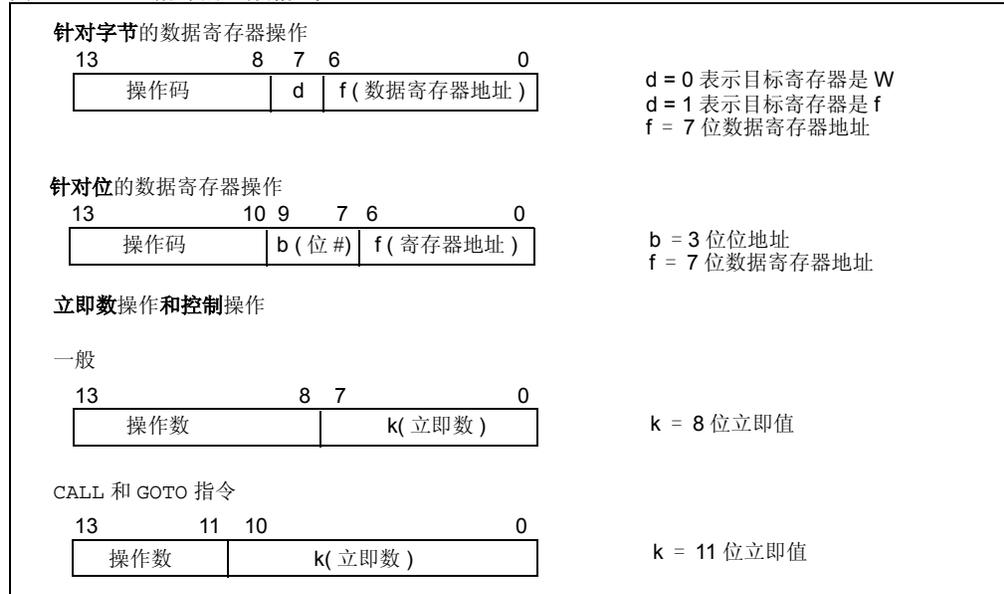
3: 改写程序计数器 (PC) 或条件测试为真时, 需要 2 个周期执行指令, 第二个周期执行空操作指令 NOP。

PICmicro 中档单片机系列

5.2 指令的一般格式

中档系列单片机的指令有四种一般格式，如图 5-1 所示。指令的操作码从 3 位到 6 位不等。这种可变长度的操作码组成了 35 条指令。

图 5-1: 指令的一般格式



5.3 中央处理单元 (CPU)

CPU 被视为器件的“大脑”，它负责获取正确的执行指令、译码并且执行该指令。

CPU 有时和 ALU 配合工作来完成指令的执行（如算术或逻辑操作）。

CPU 控制程序存储器的地址总线、数据存储器的地址总线以及对堆栈进行存取。

5.4 指令时钟

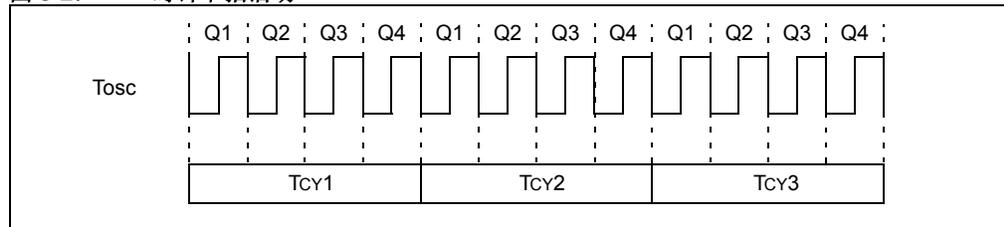
每个指令周期 (Tcy) 由 4 个时钟节拍 (Q1-Q4) 构成。时钟节拍和器件振荡周期 (Tosc) 相同。在各个时钟节拍分别对指令的译码、读取、处理和写操作等进行计时 / 标示。下图表明了时钟节拍和指令周期之间的关系。

组成指令周期 (Tcy) 的 4 个时钟节拍归纳如下：

- Q1: 指令的译码周期或强制性空操作
- Q2: 指令的读数据周期或空操作
- Q3: 处理数据
- Q4: 指令的写数据周期或空操作

每条指令都有具体的时钟节拍操作。

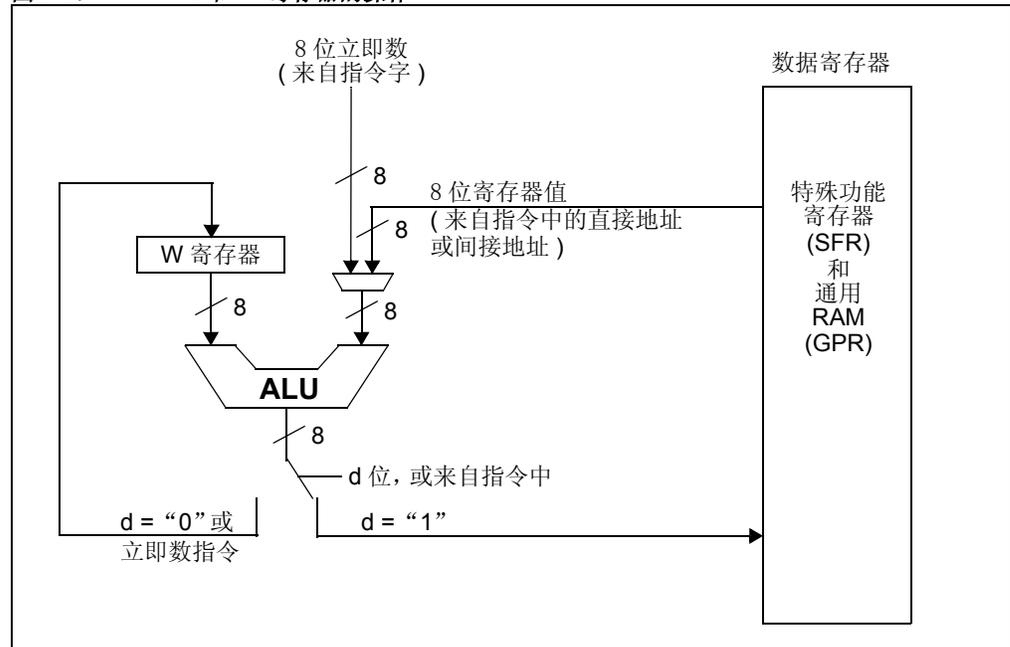
图 5-2: 时钟节拍活动



5.5 算术逻辑单元 (ALU)

PICmicro[®] 单片机包含一个 8 位 ALU 和一个 8 位工作寄存器。ALU 是一个通用的算术逻辑单元，它对工作寄存器和数据寄存器中的数据进行算术和布尔运算。

图 5-3: ALU 和 W 寄存器的操作



ALU 是 8 位宽，能够进行加、减、移位和逻辑操作。除非特别指明，算术运算一般是以 2 的补码形式进行。在 2 个操作数的指令中，典型情况下，其中一个操作数是在工作寄存器（W 寄存器）中，另一个操作数放在一个数据寄存器中或是一个立即数。在单操作数指令中，操作数放在 W 寄存器中或某个数据寄存器中。

W 寄存器是一个 8 位宽、用于 ALU 运算的工作寄存器，它是一个不可寻址的寄存器。

根据所执行的指令，ALU 可以影响状态寄存器中的进位标志位 C、辅助进位标志位 DC 和全零标志位 Z。在减法操作中，C 和 DC 位就分别作为借位和辅助借位之反。例如指令 SUBLW 和 SUBWF。

5.6 状态寄存器

状态寄存器（如图 5-1 所示）含有 ALU 的算术运算结果状态、复位状态及数据存储区的选择位。因为数据存储区的选择是由状态寄存器控制的，所以各存储区里都有状态寄存器的映射。而且，这些映射在每个存储区的相对位置（偏移位置）都相同（见第 6 章“存储器构成”中图 6-5“寄存器映射”）。

状态寄存器和其它寄存器一样，可以作为任何指令的目标寄存器。如果状态寄存器作为一条指令的目标寄存器，而这条指令又影响 Z、DC 或 C 标志位，那么这三个标志位的状态不能由指令直接写入。这些标志位的状态要根据器件逻辑操作的结果来置 1 或清零。此外，不能对 TO 和 PD 位进行写操作，所以当执行一条把状态寄存器作为目标寄存器的指令后，状态寄存器的结果可能和预想的不一樣。

例如，指令 CLRF STATUS 将状态寄存器的高 3 位清零，将 Z 标志位置 1。操作后状态寄存器的结果为 000u u1uu（u 表示未变化）。

因此，建议仅使用位操作指令 BCF、BSF 或传送指令 MOVWF 来改变状态寄存器，因为这些指令不影响该寄存器中的 Z、C 或 DC 标志位。关于其它不影响任何状态位的指令，请见表 5-1。

注 1: 一些器件不需要 IRP 和 RP1 位（STATUS<7:6>）。本章（CPU 和 ALU）未使用这些位，应该将这些位保持为零。不建议将这些位作为通用读 / 写位，因为这可能会影响代码对未来产品的向上兼容性。

注 2: 在减法运算中，C 和 DC 位分别作为借位和辅助借位之反。

寄存器 5-1: 状态寄存器

R/W-0	R/W-0	R/W-0	R-1	R-1	R/W-x	R/W-x	R/W-x
IRP	RP1	RP0	\overline{TO}	\overline{PD}	Z	DC	C
bit 7						bit 0	

bit 7 **IRP:** 寄存器组选择位 (用于间接寻址)
 1: 选择 Bank 2, Bank3 (100h - 1FFh)
 0: 选择 Bank 0, Bank1 (00h - FFh)

对于只有 Bank0 和 Bank1 的器件, 保留 IRP 位, 且应始终保持为 0。

bit 6:5 **RP1:RP0:** 寄存器组选择位 (用于直接寻址)
 11: Bank 3 (180h - 1FFh)
 10: Bank 2 (100h - 17Fh)
 01: Bank 1 (80h - FFh)
 00: Bank 0 (00h - 7Fh)

每组 128 个字节。对于只有 Bank 0 和 Bank1 的器件, 保留 IRP 位, 且应始终保持为 0。

bit 4 **TO:** 超时位
 1 = 上电、执行 CLRWDT 或 SLEEP 指令后
 0 = 发生看门狗定时器超时

bit 3 **PD:** 低功耗标志位
 1 = 上电或执行 CLRWDT 指令后
 0 = 执行 SLEEP 指令后

bit 2 **Z:** 零标志位
 1 = 算术或逻辑运算结果为 0
 0 = 算术或逻辑运算结果不为 0

bit 1 **DC:** 辅助进位 / 借位标志位 (ADDWF、ADDLW、SUBLW 和 SUBWF 指令) (借位的极性是相反的)
 1 = 结果的低 4 位向高 4 位进位 / 低 4 位向高 4 位无借位
 0 = 结果的低 4 位没有向高 4 位进位 / 低 4 位向高 4 位借位

bit 0 **C:** 进位 / 借位标志位 (ADDWF、ADDLW、SUBLW 和 SUBWF 指令)
 1 = 结果的最高位有进位 / 最高位无借位
 0 = 结果的最高位无进位 / 最高位有借位

注: 借位的极性是相反的。减法指令通过加上第二个操作数 2 的补码来实现。对于移位指令 (RRF 和 RLF), C 位值来自源寄存器的最高位或最低位。

图注
R = 可读位 W = 可写位
U = 未用位, 读为 “0” -n = 上电复位时的值

PICmicro 中档单片机系列

5.7 OPTION_REG 寄存器

OPTION_REG 寄存器是可读写寄存器，它包含配置 TMR0/WDT 的预分频器、外部 INT 中断、TMR0 和 PORTB 弱上拉的各个控制位。

寄存器 5-2: OPTION_REG 寄存器

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
$\overline{\text{RBP}}\text{U}$	INTE $\overline{\text{D}}\text{G}$	T0CS	T0SE	PSA	PS2	PS1	PS0
bit 7						bit 0	

- bit 7 **$\overline{\text{RBP}}\text{U}$** : PORTB 上拉使能位
 1 = 禁止 PORTB 上拉
 0 = 按各个端口锁存器值使能 PORTB 上拉
- bit 6 **INTE $\overline{\text{D}}\text{G}$** : 中断触发边沿选择位
 1 = INT 引脚的上升沿触发中断
 0 = INT 引脚的下降沿触发中断
- bit 5 **T0CS**: TMR0 时钟源选择位
 1 = T0CKI 引脚上的外部时钟
 0 = 内部指令周期时钟 (CLKOUT)
- bit 4 **T0SE**: TMR0 计数边沿选择位
 1 = T0CKI 引脚上的下降沿递增
 0 = T0CKI 引脚上的上升沿递增
- bit 3 **PSA**: 预分频器分配位
 1 = 预分频器分配给 WDT
 0 = 预分频器分配给 Timer0 模块
- bit 2-0 **PS2:PS0**: 预分频比选择位

位值	TMR0 比率	WDT 比率
000	1:2	1:1
001	1:4	1:2
010	1:8	1:4
011	1:16	1:8
100	1:32	1:16
101	1:64	1:32
110	1:128	1:64
111	1:256	1:128

图注
 R = 可读位 W = 可写位
 U = 未用位，读为“0” -n = 上电复位时的值

注: 如果需要 TMR0 寄存器得到 1:1 的预分频值，可以把预分频器分配给看门狗定时器 WDT (即 PSA=1)。

5.8 电源控制寄存器

利用电源控制 (PCON) 寄存器中的标志位和 TO 及 PD 位, 用户可以区别各种不同的器件复位。

注 1: $\overline{\text{BOR}}$ 在上电复位时的值是未知的。它必须由用户软件置位, 并在以后复位时检查该位是否为零, 可判断有无欠压发生。如果禁用欠压电路 (通过清除配置字里的 BODEN 位), 则不必关心 $\overline{\text{BOR}}$ 状态位, 也没必要预测。

注 2: 建议在检测到上电复位时, 将 $\overline{\text{POR}}$ 位清零, 以便检测后续的上电复位。

寄存器 5-3: 电源控制寄存器

R-u	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
MPEEN	—	—	—	—	PER	POR	BOR
bit 7							bit 0

- bit 7 **MPEEN:** 存储器奇偶校验错误状态位
该位反映 MPEEN 配置位的值。
- bit 6:3 **未用位:** 读为 “0”
- bit 2 **PER:** 存储器奇偶校验错误复位状态位
1 = 没有错误发生
0 = 发生程序存储器取奇偶校验错误
(在上电复位后必须用软件置 1)
- bit 1 **POR:** 上电复位状态位
1 = 未发生上电复位
0 = 发生上电复位 (上电复位后, 由软件置 1)
- bit 0 **BOR:** 欠压复位状态位
1 = 未发生欠压复位
0 = 发生欠压复位 (欠压复位后, 由软件置 1)

图注

R = 可读位

W = 可写位

U = 未用位, 读为 “0”

-n = 上电复位时的值

5.9 设计技巧

问 1: *我的程序算法好象工作不正确。*

答 1:

1. 指令的目标寄存器可能被指定为 W 寄存器 (d=0)，而不是数据寄存器 (d = 1)。
2. 可能没有正确选择寄存器组选择位 (RP1:RP0 或 IRP)。而且如果使用了中断的话，在退出中断处理时，可能没有正确恢复寄存器组选择位。

问 2: *我不能修改状态寄存器的标志位。*

答 2:

如果一条影响 Z、DC 或 C 位的指令的目标寄存器是状态寄存器，会禁止对这些位进行直接写操作。这些位是根据器件的逻辑操作结果来置位或清零。因此，要修改状态寄存器里的这些位，建议使用指令 BCF 和 BSF。

5.10 相关应用笔记

本部分列出了与本章内容相关的应用笔记。这些应用笔记并非都是专门针对中档单片机系列而写的（即有些针对低档系列，有些针对高档系列），但是其概念是相近的，通过适当修改并受到一定限制，即可使用。目前与 CPU 和 ALU 相关的应用笔记有：

标题	应用笔记 #
Fixed Point Routines	AN617
IEEE 754 Compliant Floating Point Routines	AN575
Digital Signal Processing with the PIC16C74	AN616
Math Utility Routines	AN544
Implementing IIR Digital Filters	AN540
Implementation of Fast Fourier Transforms	AN542
Tone Generation	AN543
Servo Control of a DC Brushless Motor	AN532
Implementation of the Data Encryption Standard using the PIC17C42	AN583
PIC16C5X / PIC16CXX Utility Math Routines	AN526
Real Time Operating System for PIC16/17	AN585

5.11 版本历史

版本 A

这是描述 CPU 和 ALU 的初始发行版。