



第6章 STC单片机CPU指令系统

何宾

2018.03



程序分支指令

8051支持有条件和无条件的程序分支指令，这些程序分支指令用于修改程序的执行顺序。

程序分支指令 --调用指令

ACALL addr11

- 该命令无条件的调用在指定地址处的子程序。目标地址由递增PC的高5位、操作码的第7到第5位和指令第2字节并置组成。所以，所调用的子程序的首地址必须与ACALL后面指令的第一个字节在同一个2KB区域内。

ACALL addr11 指令的内容

助记符	操作	标志	操作码	字节数	周期数
ACALL addr11	$(PC) \leftarrow (PC) + 2$ $(SP) \leftarrow (SP) + 1$ $((SP)) \leftarrow (PC_{7-0})$ $(SP) \leftarrow (SP) + 1$ $((SP)) \leftarrow (PC_{15-8})$ $(PC_{10-0}) \leftarrow \text{页面地址}$	无	$a_{10}a_9a_81\textcolor{red}{0001}$	2	4

程序分支指令 --调用指令

【例】假设堆栈指针的初值为07H，标号SUBRTN位于程序存储器地址为0345H的位置，如果执行位于地址0123H处的指令：

ACALL SUBRTN

结果：

堆栈指针的内容变成09H，内部RAM地址为08H和09H的位置保存的内容为25H和01H，PC值变为0345H。

程序分支指令 --调用指令

LCALL addr16

该命令无条件的调用首地址为addr16处的子程序。执行该指令时，将PC加3，以获得下一条指令的地址。然后将指令第2，第3字节所提供的16位目标地址送入PC₁₅₋₀，程序转向子程序的首地址执行。所调用的子程序的首地址可以在64KB的范围内。

LCALL addr16 指令的内容

助记符	操作	标志	操作码	字节数	周期数
LCALL addr16	$(PC) \leftarrow (PC) + 3$ $(SP) \leftarrow (SP) + 1$ $((SP)) \leftarrow (PC_{7-0})$ $(SP) \leftarrow (SP) + 1$ $((SP)) \leftarrow (PC_{15-8})$ $(PC) \leftarrow \text{页面地址}$	N	00010010	3	4

程序分支指令 --调用指令

【例】假设堆栈指针的初值为07H，标号SUBRTN位于程序存储器地址为1234H的位置，如果执行位于地址0123H处的指令：

LCALL SUBRTN

结果：

堆栈指针的内容变成09H，内部RAM地址为08H和09H的位置保存的内容为26H和01H，PC值变为1234H。

程序分支指令

--返回指令

RET

- 该命令将栈顶高地址和低地址字节连续的送给PC的高字节和低字节，并把堆栈指针减2，返回ACALL或LCALL的下一条指令，继续往下执行，该指令的操作不影响标志位。

RET 指令的内容

助记符	操作	标志	操作码	字节数	周期数
RET	$(PC_{15-8}) \leftarrow ((SP))$ $(SP) \leftarrow (SP) - 1$ $(PC_{7-0}) \leftarrow ((SP))$ $(SP) \leftarrow (SP) - 1$	N	00100010	1	4

程序分支指令 --调用指令

【例】堆栈指针的内容为0BH，内部RAM地址为0AH和0BH的位置保存的内容为23H和01H，如果执行指令：

RET

结果：

堆栈指针的内容变成09H，程序将从地址为0123H的地方继续执行。

程序分支指令

--返回指令

RETI

- 该命令将从中断程序返回，并清除相应的内部中断状态寄存器。CPU在执行RETI后，至少要再执行一条指令，才能响应新的中断请求。

RETI 指令的内容

助记符	操作	标志	操作码	字节数	周期数
RETI	$(PC_{15-8}) \leftarrow ((SP))$ $(SP) \leftarrow (SP) - 1$ $(PC_{7-0}) \leftarrow ((SP))$ $(SP) \leftarrow (SP) - 1$	N	00110010	1	4

程序分支指令

--返回指令

【例】堆栈指针的内容为0BH，结束在地址0123H处的指令执行结束期间产生中断，内部RAM地址为0AH和0BH的位置保存的内容为23H和01H，如果执行指令：

RETI

结果：

堆栈指针的内容变成09H，中断返回后继续从程序代码地址为0123H的位置继续执行。

程序分支指令

--无条件转移指令

AJMP addr11

- 该命令实现无条件跳转。绝对跳转操作它的目标地址是由PC递增两次后值的高5位，操作码的7-5位，和第二个字节并置而成。目标地址必须包含AJMP指令后第一条指令的第一个字节在内的2KB范围内。

AJMP addr11指令的内容

助记符	操作	标志	操作码	字节数	周期数
AJMP addr11	$(PC) \leftarrow (PC) + 2$ $(PC_{10-0}) \leftarrow \text{页面地址}$	N	a ₁₀ a ₉ a ₈ 00001	2	3

程序分支指令

--无条件转移指令

【例】假设标号JMPADR位于程序存储器的0123H的位置，如果指令：

AJMP JMPADR

位于程序存储器地址为0345H的位置

结果：

执行完该指令后，PC的值变为0123H。

程序分支指令

--无条件转移指令

LJMP addr16

- 该命令实现无条件长跳转操作，跳转的16位目的地址由指令的第2和第3字节组成。因此，程序指向的目标地址可以包含程序存储器的整个64KB的空间。

LJMP 指令的内容

助记符	操作	标志	操作码	字节数	周期数
LJMP addr16	$(PC) \leftarrow \text{addr}_{15} \dots \text{addr}_0$	N	00000010	3	4

注：在操作码后面带着一个字节的目標地址的 **A₁₅~A₈** 位和一个字节的目標地址的 **A₇~A₀** 位。

程序分支指令

--无条件转移指令

【例】假设标号JMPADR位于程序存储器的1234H的位置，如果指令：

LJMP JMPADR

位于程序存储器地址为1234H的位置

结果：

执行完该指令后，PC的值变为1234H。

程序分支指令

--无条件转移指令

SJMP rel

- 该命令实现无条件短跳转操作，跳转的目的地址是由PC递增两次后的值和指令的第二字节带符号的相对偏移量相加而成的。

SJMP rel 指令的内容

助记符	操作	标志	操作码	字节数	周期数
SJMP rel	$(PC) \leftarrow (PC) + 2$ $(PC) \leftarrow (PC) + rel$	N	10000000	2	3

注：在操作码后面带着一个字节的偏移量rel。

思考与练习：rel的最大值？

程序分支指令

--无条件转移指令

【例】 假设标号RELADR位于程序存储器的0123H的位置，如果指令：

SJMP RELADR

位于程序存储器地址为0100H的位置。

结果：

执行完该指令后，PC的值变为0123H。

程序分支指令

--无条件转移指令

JMP @A+DPTR

- 该命令实现无条件的跳转操作，跳转的目标地址是将累加器A中的8位无符号数与数据指针DPTR的内容相加而得。相加运算不影响累加器A和数据指针DPTR的原内容。若相加结果大于64KB，则从程序存储器的零地址往下延续。

JMP @A+DPTR指令的内容

助记符	操作	标志	操作码	字节数	周期数
JMP @A+DPTR	$(PC) \leftarrow (A) + (DPTR)$	N	01110011	1	5

程序分支指令

--无条件转移指令

【例】假设累加器A中的值是偶数（0~6）。下面的指令序列将使程序跳转到位于跳转表JMP_TBL的4条AJMP指令中的某一条去执行：

```
MOV DPTR,#JMP_TBL
```

```
JMP @A+DPTR
```

```
JMP_TBL: AJMP LABEL0
```

```
AJMP LABEL1
```

```
AJMP LABEL2
```

```
AJMP LABEL3
```

如果开始执行上面指令时，累加器A中的值为04H，那么程序最终会跳到标号为LABEL2的地方执行。

程序分支指令

--有条件转移指令

JNZ rel

- 该命令实现有条件跳转。判断累加器A的内容是否不为0，如果不为0，则跳转到(PC) + rel指定的目标地址；否则，程序转向下一条指令。

JNZ rel指令的内容

助记符	操作	标志	操作码	字节数	周期数
JNZ rel	$(PC) \leftarrow (PC) + 2$ 如果 $(A) \neq 0$, 则 $(PC) \leftarrow (PC) + rel$	N	01110000	2	4

注：在操作码后面带着一个字节的偏移量rel。

程序分支指令

--有条件转移指令

【例】假设累加器A的内容00H，则执行指令：

JNZ LABEL1 ; 跳转条件不成立

INC A ; 累加器的内容加1

JNZ LABEL2 ; 跳转条件成立

结果：

程序跳转到标号LABEL2的地方执行。

程序分支指令

--有条件转移指令

JZ rel

- 该命令实现有条件跳转。判断累加器A的内容是否为0，如果为0，则跳转到(PC) + rel指定的目标地址；否则，程序转向下一条指令。

JZ rel 指令的内容

助记符	操作	标志	操作码	字节数	周期数
JZ rel	$(PC) \leftarrow (PC) + 2$ 如果 $(A) = 0$, 则 $(PC) \leftarrow (PC) + rel$	N	01100000	2	4

注：在操作码后面带着一个字节的偏移量rel。

程序分支指令

--有条件转移指令

【例】 假设累加器A的内容01H，则执行指令：

```
JZ LABEL1      ; 跳转条件不成立  
DEC A           ; 累加器的内容减1  
JZ LABEL2      ; 跳转条件成立
```

结果： 程序跳转到标号LABEL2的地方执行。

程序分支指令

--有条件转移指令

CJNE A, direct, rel

- 该命令对累加器A和直接寻址单元内容相比较，若它们的值不相等则程序转移到(PC) + rel指向的目标地址。若直接寻址单元的内容小于累加器内容，则清除进位标志CY；否则，置位进位标志CY。

CJNE A, direct, rel指令的内容

助记符	操作	标志	操作码	字节数	周期数
CJNE A, direct, rel	$(PC) \leftarrow (PC) + 3$ 如果 $(A) \neq (\text{direct})$ 则 $(PC) \leftarrow (PC) + \text{rel}$ 如果 $(A) < (\text{direct})$, 则 $(CY) \leftarrow 1$ 否则 $(CY) \leftarrow 0$	N	10110101	3	3

程序分支指令

--有条件转移指令

CJNE A, #data, rel

- 该命令将比较累加器A的内容和立即数，若它们的值不相等，则程序转移(PC) + rel指向的目标地址。进位标志CY设置同上，该指令不影响累加器A的内容。

CJNE A, #data, rel指令的内容

助记符	操作	标志	操作码	字节数	周期数
CJNE A, #data, rel	$(PC) \leftarrow (PC) + 3$ 如果 $(A) \neq (data)$ 则 $(PC) \leftarrow (PC) + rel$ 如果 $(A) < (data)$, 则 $(CY) \leftarrow 1$ 否则 $(CY) \leftarrow 0$	N	10110101	3	3

程序分支指令

--有条件转移指令

CJNE Rn, #data, rel

- 该命令将寄存器Rn的内容和立即数进行比较，若它们的值不相等，则程序转移到(PC) + rel指向的目标地址。进位标志CY设置同上。

CJNE Rn, #data,rel 指令的内容

助记符	操作	标志	操作码	字节数	周期数
CJNE Rn, #data,rel	$(PC) \leftarrow (PC) + 3$ 如果 $(Rn) \neq data$, 则 $(PC) \leftarrow (PC) + rel$ 如果 $(Rn) < data$, 则 $(CY) \leftarrow 1$ 否则 $(CY) \leftarrow 0$	CY	10111rrr	3	4

注：rrr为寄存器的编号，因此机器码范围是B8H~BFH。

程序分支指令

--有条件转移指令

CJNE @Ri , #data , rel

- 该命令将间接寻址的内容和立即数相比较，若它们的值不相等，则程序转移到(PC) + rel指向的目标地址。进位标志CY设置同上。

CJNE @Ri , #data,rel指令的内容

助记符	操作	标志	操作码	字节数	周期数
CJNE @Ri , #data,rel	$(PC) \leftarrow (PC) + 3$ 如果 $((Ri)) \neq (data)$, 则 $(PC) \leftarrow (PC) + rel$ 如果 $((Ri)) < data$, 则 $(CY) \leftarrow 1$ 否则 $(CY) \leftarrow 0$	CY	1011011i	3	5

注：i表示R0或者R1。当i=0时，表示R0寄存器；当i=1时，表示R1寄存器。

程序分支指令

--有条件转移指令

【例】假设累加器A的内容34H，寄存器R7的内容为56H。则执行指令：

```
CJNE R7, #60H, NOT_EQ
.....          ; R7的内容为60H
NOT_EQ: JC  REQ_LOW  ; 如果R7<60H
.....          ; R7>60H
```

结果：

第一条指令将进位标志CY设置为1，程序跳转到标号NOT_EQ的地方。接下来测试进位标志CY，可以确定寄存器R7的内容大于还是小于60H。

程序分支指令

--有条件转移指令

DJNZ Rn,rel

- 该命令实现有条件跳转。每执行一次指令，寄存器Rn的内容减1，并判断其内容是否为0。若不为0则转向(PC) + rel指向的目标地址，继续执行循环程序，否则，结束循环程序，执行下一条指令。

DJNZ Rn , rel指令的内容

助记符	操作	标志	操作码	字节数	周期数
DJNZ Rn , rel	$(PC) \leftarrow (PC) + 2$ $(Rn) \leftarrow (Rn) - 1$ 如果 $(Rn) \neq 0$, 则 $(PC) \leftarrow (PC) + rel$	N	11011rrr	2	4

程序分支指令

--有条件转移指令

DJNZ direct,rel

- 该命令实现有条件跳转。每执行一次指令，直接寻址单元的内容减1，并判断其内容是否为0。若不为0则转向(PC) + rel指向的目标地址，继续执行循环程序，否则，结束循环程序，执行下一条指令。

DJNZ direct , rel 指令的内容

助记符	操作	标志	操作码	字节数	周期数
DJNZ direct , rel	$(PC) \leftarrow (PC) + 3$ $(direct) \leftarrow (direct) - 1$ 如果 $(direct) \neq 0$ 则 $(PC) \leftarrow (PC) + rel$	N	11010101	3	5

程序分支指令

--有条件转移指令

【例】假设内部RAM地址为40H、50H和60H的单元分别保存着数据01H、70H和15H，则执行指令：

DJNZ 40H, LABEL_1

DJNZ 50H, LABEL_2

DJNZ 60H, LABEL_3

结果：

程序将跳转到标号LABEL_2处执行，且相应的3个RAM单元的内容变成00H、6FH和15H。

程序分支指令

--空操作指令

NOP

- 该命令表示无操作，PC+1。

NOP 指令的内容

助记符	操作	标志	操作码	字节数	周期数
NOP	$(PC) \leftarrow (PC) + 1$	N	0x00	1	1

程序分支指令 --空操作指令

【例】假设期望在端口P2的第7位引脚上输出一个长时间的低电平脉冲，该脉冲持续5个及其周期（精确）。若仅仅使用SETB和CLR指令序列，生成的脉冲只能持续一个机器周期。因此，需要设法增加4个额外的机器周期，可以按照下面的方式实现所要求的功能（假设在此期间没有使能中断）：

CLR P2.7

NOP

NOP

NOP

NOP

SETB P2.7