



# 第6章 STC单片机CPU指令系统

何宾

2018.03

# 算术指令

## --减法指令

### SUBB A,Rn

- 该指令从累加器A中减去寄存器Rn和进位标志CY内的内容，将结果保存在累加器A中。

SUBB A,Rn 指令的内容

助记符	操作	标志	操作码	字节数	周期数
SUBB A,Rn	$(PC) \leftarrow (PC) + 1$ $(A) \leftarrow (A) - (C) - (Rn)$	CY,AC,OV	10011rrr	1	1

注：rrr为寄存器的编号，因此机器码范围是98H~9FH。

# 算术指令

## --减法指令

- 如果第7位需要一个借位，则设置进位（借位）标志；否则，清除CY标志。
- 如果第3位需要一个借位，则设置AC标志；否则，清除AC标志。
- 如果第6位需要借位，而7位没有借位时；或者第7位有借位，而第6位没有借位时，在这两种情况下都会设置OV标志。
- 或者可以这样说，当减去有符号的整数时，当一个正数减去一个负数，产生一个负数结果时；或者一个负数减去一个正数时，产生一个正数结果时，设置OV标志。

# 算术指令

## --减法指令

【例】假设累加器A中的数据为C9H，R2寄存器中的数据为54H，进位标志为1时，当执行指令：

**SUBB A,R2**

结果：

(A) =74H, (AC) =0, (CY) =0, (OV) =1

计算过程为：

$$\begin{array}{r} 1100,1001 \\ 0101,0100 \\ - \quad \quad 1 \\ \hline 0111,0100 \end{array}$$

# 算术指令

## --减法指令

### SUBB A,direct

- 该指令从累加器A中减去直接寻址单元的内容和进位标志CY的内容，然后结果保存在累加器A中。CY、AC、OV设置如上。

SUBB A,direct 指令的内容

助记符	操作	标志	操作码	字节数	周期数
SUBB A,direct	$(PC) \leftarrow (PC) + 2$ $(A) \leftarrow (A) - (C) - (\text{direct})$	CY,AC,OV	10010101	2	2

注：在操作码后面跟着一个字节的直接地址。

# 算术指令

## --减法指令

### SUBB A,@Ri

- 该指令从累加器A中减去间接寻址单元的内容和进位标志CY的内容，然后结果保存在累加器A中。CY、AC、OV设置如上。

SUBB A,@Ri 指令的内容

助记符	操作	标志	操作码	字节数	周期数
SUBB A,@Ri	$(PC) \leftarrow (PC) + 1$ $(A) \leftarrow (A) - (C) - ((Ri))$	CY,AC,OV	1001011i	1	2

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。

# 算术指令

## --减法指令

### SUBB A,#data

- 该指令从累加器A中减去一个立即数和进位标志CY的内容，然后结果保存在累加器A中。CY、AC、OV设置如上。

SUBB A,#data 指令的内容

助记符	操作	标志	操作码	字节数	周期数
SUBB A,#data	$(PC) \leftarrow (PC) + 2$ $(A) \leftarrow (A) - (C) - data$	CY,AC,OV	10010100	2	2

注：在操作码后面跟着一个字节的立即数。

# 算术指令

## --递增指令

### INC A

- 该指令将累加器A的内容加1，结果保存在累加器A中。若累加器的结果为0xFF时，将其内容设置为0。

INC A 指令的内容

助记符	操作	标志	操作码	字节数	周期数
INC A	$(PC) \leftarrow (PC) + 1$ $(A) \leftarrow (A) + 1$	N	00000100	1	1



# 算术指令

## --递增指令

### INC Rn

- 该指令将寄存器Rn的内容加1，结果保存在Rn中。若Rn的结果为0xFF时，将其内容设置为0。

INC Rn 指令的内容

助记符	操作	标志	操作码	字节数	周期数
INC Rn	$(PC) \leftarrow (PC) + 1$ $(Rn) \leftarrow (Rn) + 1$	N	00001rrrr	1	2

注：rrr为寄存器的编号，因此机器码范围是08H~0FH。

# 算术指令

## --递增指令

### INC direct

- 该指令将直接寻址单元的内容加1，结果保存在直接地址单元中。若直接地址单元的结果为0xFF时，将其内容设置为0。

INC direct 指令的内容

助记符	操作	标志	操作码	字节数	周期数
INC direct	$(PC) \leftarrow (PC) + 2$ $(direct) \leftarrow (direct) + 1$	N	00000101	2	3

注：在操作码后面跟着一个字节的直接地址。

# 算术指令

## --递增指令

### INC @Ri

- 该指令将间接寻址单元的内容加1，结果保存在间接地址单元中。若间接地址单元的结果为0xFF时，将其内容设置为0。

INC @Ri 指令的内容

助记符	操作	标志	操作码	字节数	周期数
INC @Ri	$(PC) \leftarrow (PC) + 1$ $((Ri)) \leftarrow ((Ri)) + 1$	N	0000011i	1	3

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。

# 算术指令

## --递增指令

### INC DPTR

- 该指令将DPTR的内容加1，结果保存在DPTR中。若DPTR的结果为0xFFFF时，将其内容设置为0x0000。

INC DPTR 指令的内容

助记符	操作	标志	操作码	字节数	周期数
INC DPTR	$(PC) \leftarrow (PC) + 1$ $(DPTR) \leftarrow (DPTR) + 1$	N	10100011	1	1

# 算术指令

## --递增指令

【例】假设寄存器R0中的数据为7EH，内部RAM地址为7EH和7FH单元的数据分别为FFH和40H，即：(7E) = FFH，(7F) = 40H，则当执行指令：

INC @R0 ; 内部RAM地址为7EH单元的内容加1，变成0

INC R0 ; 寄存器R0中的数据变为7FH

INC @R0 ; 内部RAM地址为7FH单元的内容加1，变成41H

结果：

(R0) = 7FH，内部RAM地址为7EH和7FH单元的数据变为00H和41H。

# 算术指令

## --递减指令

### DEC A

- 该指令将累加器A的内容减1，结果保存在累加器A中。如果累加器A中的内容为0，则变为0xFF。

DEC A 指令的内容

助记符	操作	标志	操作码	字节数	周期数
DEC A	$(PC) \leftarrow (PC) + 1$ $(A) \leftarrow (A) - 1$	N	00010100	1	1

# 算术指令

## --递减指令

### DEC Rn

- 该指令将寄存器Rn的内容减1，结果保存在寄存器Rn中。如果Rn的内容为0，则变为0xFF。

DEC Rn 指令的内容

助记符	操作	标志	操作码	字节数	周期数
DEC Rn	$(PC) \leftarrow (PC) + 1$ $(Rn) \leftarrow (Rn) - 1$	N	00011rrr	1	2

注：rrr为寄存器的编号，因此机器码范围是18H~1FH。

# 算术指令

## --递减指令

### DEC direct

- 该指令将直接寻址单元的内容减1，结果保存在直接地址单元中。如果直接寻址单元的内容为0，则变为0xFF。

DEC direct 指令的内容

助记符	操作	标志	操作码	字节数	周期数
DEC direct	$(PC) \leftarrow (PC) + 2$ $(direct) \leftarrow (direct) - 1$	N	00010101	2	3

注：在操作码后面跟着一个字节的直接地址。



# 算术指令

## --递减指令

DEC @Ri

- 该指令将间接寻址单元的内容减1，结果保存在间接地址单元中。如果间接寻址单元的内容为0，则变为0xFF。

DEC @Ri 指令的内容

助记符	操作	标志	操作码	字节数	周期数
DEC @Ri	$(PC) \leftarrow (PC) + 1$ $((Ri)) \leftarrow ((Ri)) - 1$	N	0001011i	1	3

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。

# 算术指令

## --递减指令

【例】假设寄存器R0中的数据为7FH，内部RAM地址为7EH和7FH单元的数据分别为00H和40H，即：(7F) = 00H，(7E) = 40H，则当执行指令：

DEC @R0 ; 内部RAM地址为7FH单元的内容减1，变成FFH

DEC R0 ; 寄存器R0中的数据变为7EH

DEC @R0 ; 内部RAM地址为7EH单元的内容减1，变成3FH

结果：

(R0) = 7EH，内部RAM地址为7EH和7FH单元的数据变为FFH和3FH。

# 算术指令

## --乘法指令

### MUL AB

- 该指令将累加器A和寄存器B中的两个无符号8位二进制数相乘，所得的16位乘积的低8位结果保存在累加器A中，高8位结果保存在寄存器B中。
- 如果乘积大于255，则溢出标志OV置1；否则OV清零。
- 在执行该命令时，总是清除进位标志CY。

MUL AB 指令的内容

助记符	操作	标志	操作码	字节数	周期数
MUL AB	$(PC) \leftarrow (PC) + 1$ $(A) \leftarrow (A) \times (B)$ 结果第7位到第0位 $(B) \leftarrow (A) \times (B)$ 结果第15位到第8位	CY,OV	10100100	1	2

# 算术指令

## --乘法指令

【例】假设累加器A中的数据为  
(80)<sub>10</sub>=50H，寄存器B中的数据为  
(160)<sub>10</sub>=A0H，则执行指令：

MUL AB

结果：

乘积为(12800)<sub>10</sub>=3200H，则

(A) =00H, (B) =32H,

(CY) =0, (OV) =1。

```
      01010000
    × 10100000
    ──────────
      00000000
      00000000
      00000000
      00000000
      00000000
      00000000
      01010000
      00000000
      +01010000
    ──────────
    0011001000000000
```

# 算术指令

## --除法指令

### DIV AB

- 该指令用累加器A中的无符号整数除以寄存器B中无符号整数。所得的商保存在累加器A中，余数保存在寄存器B中。
- 当除数（B寄存器的内容）为0时，结果不定，溢出标志OV置1。
- 在执行该指令时，清除进位标志CY。

DIV AB 指令的内容

助记符	操作	标志	操作码	字节数	周期数
DIV AB	$(PC) \leftarrow (PC) + 1$ $(A)_{15-8} \leftarrow (A)/(B)$ $(B)_{7-0} \leftarrow (A)/(B)$	CY,OV	10000100	1	6

# 算术指令

## --除法指令

【例】假设累加器A中的数据为 $(251)_{10} = \text{FBH}$ ，寄存器B中的数据为 $(18)_{10} = 12\text{H}$ ，则执行指令：

**DIV AB**

结果：(A) = 0DH, (B) = 11H, (CY) = 0, (OV) = 0

```
      00001101
00010010 ) 11111011
            -10010
              11010
              -10010
                100011
                -010010
                  10001
```

# 算术指令

## --BCD调整指令

### DA A

- 该指令的功能是对BCD码的加法结果进行调整。
- 两个压缩型BCD码按十进制数相加后，须经此指令的调整才能得到压缩型BCD码的和。
- 本指令是根据A的最初数值和程序状态字PSW的状态，决定对A进行加06H、60H或66H操作的。

DA A 指令的内容

助记符	操作	标志	操作码	字节数	周期数
DA A	$(PC) \leftarrow (PC) + 1$ 如果 $[(A_{3-0}) > 9] \vee [(AC) = 1]$ 则： $(A_{3-0}) \leftarrow (A_{3-0}) + 6$ 如果 $[(A_{7-4}) > 9] \vee [(C) = 1]$ 则： $(A_{7-4}) \leftarrow (A_{7-4}) + 6$	CY	11010100	1	3