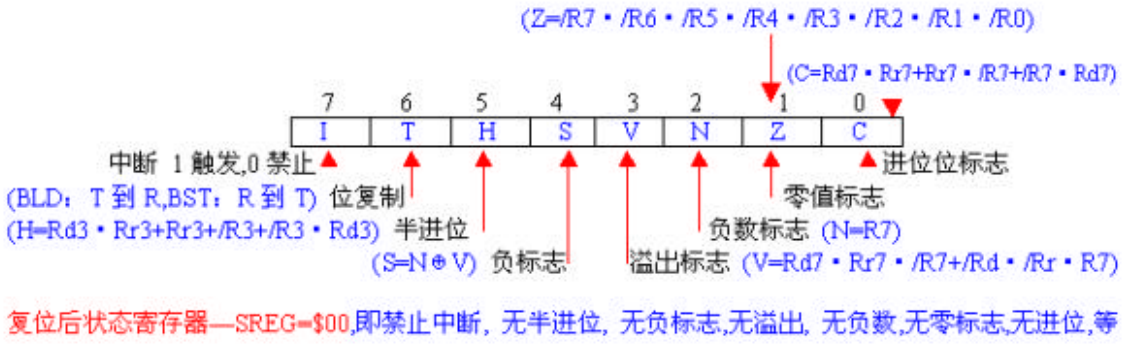
说明:为了使读者和用户对每条指令有一个具体的了解,又有利于大家对单片机映象空间(通用工作寄存器空间、I/O 寄存器空间、片内片外 SRAM 空间、程序存储器空间、EEPROM 数据存储器空间)认识更清楚(软件即完成在单片机映象空间之间或自身之间的传送、运算、检测、处理等操作),我们对每条指令均编一些简单的测试指令,我们约定它的程序编号为第四章第四节第几段第几题,例 4.4.1 加法指令的“1.不带进位加法”,程序编号为 4411.ASM。其余依此类推。

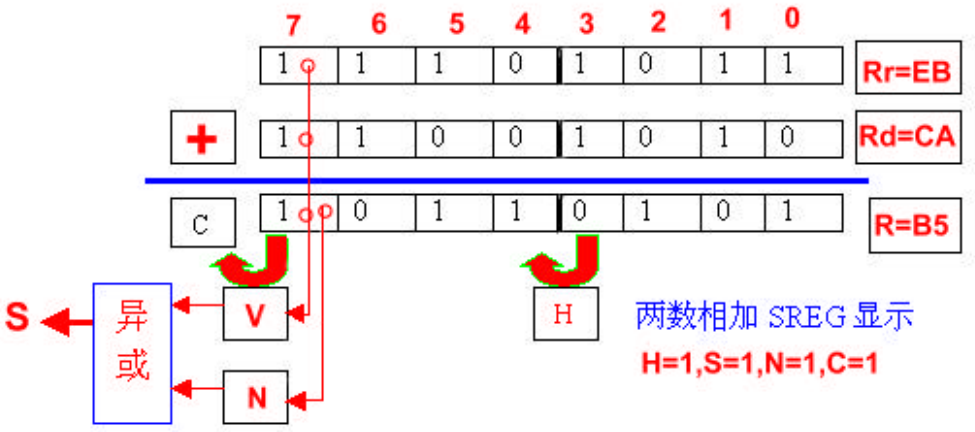
说明:AVR 单片机的指令系统对不同器件有不同指令,选用器件时应注意这一点(详情见本书附录 3),某种器件应使用那些指令,更详细资料请阅相应器件(电子书光盘中)英文指令表。

以下讲述 130 条指令功能及应用

4.4 算术和逻辑指令

AVR 的算术运算指令有加、减、乘、取反、取补、比较指令、增量和减量指令。逻辑运算指令有与、或、异或指令等。





4.4.1 加法指令

1. 不带进位加法

ADD 一不带进位加

说明: 两个寄存器不带进位 C 标志加，结果送目的寄存器 Rd。

操作: Rd←Rd+ Rr

语法:	操作码:	程序计数器:
ADD Rd,Rr	0≤d≤31，0≤r≤31	PC←PC+1

16 位操作码:

0000	11rd	dddd	rrrr

状态寄存器（SREG）和布尔格式:

I	T	H	S	V	N	Z	C
-	-	⇔	⇔	⇔	⇔	⇔	⇔

H: Rd3·Rr3+Rr3+/R3+/R3·Rd3　　N: R7

S: N⊕V,　　Z: /R7·/R6·/R5·/R4./R3·/R2·/R1·/R0

V: Rd7· Rr7./R7+/Rd7·/Rr7·R7　　C: Rd7.Rr7+Rr7·/R7+/R7·Rd7

重要说明:以下所有举的例子仅说明指令书写方法,该例子不能直接汇编,因为程序缺器件配制文件及程序执行地址。实践操作例子*.ASM 可以汇编,可以调试,可以修改(修改需改变文件属性,因为只读文件无法修改)!

约定:注释寄存器的内容(数据)用括号表示,例:(R16)=$16,表示寄存器的内容(数据)为十六进制数 16H!

例子:。(实践操作程序 4411.ASM) 实践操作例子*.ASM,必须编译生成*.OBJ 文件才可调试,如要修改*.ASM,必须修改文件属性,去掉*.ASM 只读文件属性

```
        ldi r16,$11          ;LDI 立即数装入指令,要求寄存器必须符合 16≤d≤31 条件
        ldi r20,$22
        ldi r28,0XAA         ;$,0X 均为十六进制表示法
lp:add  rl6, r20             ; 也可单步执行到此行,
                             ;在调试窗口的对应寄存器 R16,R20 输入数据
                             ;(R16) =     ,(SREG) =
        add  r28, r28        ; 也可单步执行到此行,在调试窗口的对应寄存器 R28 输入数据
                             ;(R28) =     ,(SREG) =
        rjmp lp              ;反复做实验

    Words: 1 ( 2  bytes)
    Cycles: 1
```

2. 带进位加法

ADC——带进位加

说明: 两个寄存器和 C 标志的内容相加, 结果送目的寄存器 Rd。

操作: Rd←Rd+Rr+C

语法:	操作码:	程序计数器:
ADC Rd,Rr	0≤d≤31, 0≤r≤31	PC←PC+1

16 位操作码:

0001	11rd	dddd	rrrr

状态寄存器（SREG）和布尔格式:

I	T	H	S	V	N	Z	C
-	-	⇔	⇔	⇔	⇔	⇔	⇔

H：Rd3·Rr3+Rr3++/R3·Rd3　　N：R7
S：N⊕V，　　Z：/Rd7./Rr7./Rr7·/R7./R7·/Rd7
V：Rd7· Rr7./R7./Rd7·/Rr7·R7　　C：Rd7.Rr7+Rr7·/R7+/R7·Rd7

例子：(实践操作程序 4412.ASM) 实践操作例子*.ASM,必须编译生成*.OBJ 文件才可调试,如要修改*.ASM,必须修改文件属性,去掉*.ASM 只读文件属性

```
    ldi r20,$77          ; LDI 立即数装入指令,要求寄存器必须符合 16≤d≤31 条件
    ldi r21,$99
    LDI R22,0X77
    LDI R23,0X11
  lp:add   r22, r20 ;(r22) =    ,(SREG) =
    ADC   R23, R2l  ; 也可单步执行到此行,在调试窗口的对应寄存器 R23,R21 输入数据
                    ;(R23) =    ,(SREG) =
    rjmp lp         ; 反复做实验

    Words:   1（2 bytes）
    CyCICS:  l
```

3. 立即数据加法（字）

ADIW—立即数加法

说明：寄存器对于立即数值（0~63）相加，结果放到寄存器对。

操作：Rdh:Rdl←Rdh:Rdl+K

语法：	操作码：	程序计数器：
ADIW Rdl, K	d∈{ 24 26 28 30 }, UJ	PC←PC+ 1

16 位操作码：

1001	0110	KKdd	KKKK

状态寄存器（SREG)和布尔格式：

I	T	H	S	V	N	Z	C
-	-	-	⇔	⇔	⇔	⇔	⇔

S：N⊕V
V：Rdh7·R15
N： R15
Z：/R15·/R14·/R13·/R12·/R11·/R10·/R9·/R8·/R7·/R6·/R5·/R4·/R3·/R2·/R1·/R0
C：/R15·Rdh7

例子：(实践操作程序 4413.ASM) 实践操作例子*.ASM,必须编译生成*.OBJ 文件才可调试,如要修改*.ASM,必须修改文件属性,去掉*.ASM 只读文件属性

```
    lp:adiw r24,  1     ; 也可单步执行到此行,在调试窗口的对应寄存器 R24 输入数据
      ADIW R30,  63     ; 也可单步执行到此行,在调试窗口的对应寄存器 R30 输入数据
      rjmp lp           ;反复测试

    WOrdS:   1（ 2   bytes）
    Cycles:  2
```

4. 加 1 指令

INC——加 1

说明：寄存器 Rd 的内容加 1，结果送目的寄存器 Rd 中。该操作不改变 SREG 中的 C 标志，所以 INC 指令允许在多倍字长计算中用作循环计数。当对无符号数操作时，仅有 BREQ(相等转移)和 BRNE(不为零转移)指令有效。当对二进制补码值操作时，所有的带符号转移指令都有效。

操作：Rd←Rd＋ 1

语法：	操作码：	程序计数器：
INC Rd	0≤d≤31	PC←PC＋ 1

16 位操作码：

1001	010d	dddd	0011

状态寄存器（SREG)和布尔格式：

I	T	H	S	V	N	Z	C
-	-	-	⇔	⇔	⇔	⇔	⇔

S： N⊕V　　N：R7

V：R7·/R6·/RS·/R4·/R3·/R2·/RI·/R0　　Z：/R7·/R6·/R5·/R4·/R3·/R2·/R1·/R0

例子：(实践操作程序 4414.ASM)

```
        clr  r22      ;寄存器 R22 清零
   loop:inc  r22      ;+1，也可单步执行到此行,在调试窗口的对应寄存器 R22 输入数据
  loop1:INC  R22
        cpi  r22,$04  ;(R22)与立即数$04 比较
        brnq loop1    ;不相等转移,相等则按顺序执行(观察状态寄存器 Z 标志变化)
        rjmp loop     ;供反复测试
```

Word： l（ 2 bytes）

Cycles： 1

4.4.2 减法指令

1. 不带进位减法

SUB ——不带进位减

说明：两个寄存器相减，结果送目的寄存器 Rd 中。

操作：Rd←Rd—Rr

语法：	操作码：	程序计数器：
SUB Rd,Rr	0≤d≤31，0≤r≤31	PC←PC＋1

16 位操作码

0001	11rd	dddd	rrrr

状态寄存器（SREG）和布尔格式：

I	T	H	S	V	N	Z	C
-	-	⇔	⇔	⇔	⇔	⇔	⇔

H：/Rd3·Rr3＋Rr3·R3＋R3·/Rd3　　N：R7

S：N⊕V　　Z：/R7·/R6·/R5·/R4·/Rr3·/R2·/R1·/R0

V：Rd7·/Rr7·/R7＋/Rd7· Rr7· R7　　C：/Rd7· Rr7＋Rr7·R7＋R7· /Rd7

例子：(实践操作程序 4421.ASM)

```
loop1:  ldi r23,$44     ;寄存器装入立即数
        ldi r22,$11     ;寄存器装入立即数
  loop: sub r23,r22     ;减法 ,也可单步执行到此行,
                        ;在调试窗口的对应寄存器 R23,R22 输入数据
          brne loop     ;r23 内容不为 0 转,为 0 顺执
          rjmp loop1    ;反复测试
```

Words：　1（ 2　bytes）

Cycles：　1

2. 立即数减法（字节）

SUBI—立即数减

说明：一个寄存器和常数相减，结果送目的寄存器 Rd。该指令工作于寄存器 R16 到 R31 之间，非常适合 X、 Y 和 Z 指针的操作。

操作：Rd←Rd–K

语法：	操作码：	程序计数器：
SUBI Rd,K	16≤d≤31，0≤ k≤255	PC←PC＋ 1

16 位操作码：

0101	kkkk	dddd	kkkk

状态寄存器（SREG）和布尔格式：

I	T	H	S	V	N	Z	C
-	-	⇔	⇔	⇔	⇔	⇔	⇔

H：/Rd3·K3＋K3·R3＋R3·/Rd3　　N：R7

S：N⊕V　　Z：/R7·/R6·/R5·/R4·/Rr3·/R2·/R1·/R0

V：Rd7·/K7·/R7＋/Rd7· K7· R7　　C：/Rd7·Rr7＋Rr7·R7＋R7· Rd7

例子：(实践操作程序 4422.ASM)

```
LOOP: LDI    R22, $55
LOOP1: SUBI   R22 , $11   ; 也可单步执行到此行,在调试窗口的对应寄存器 R22 输入数据
       BRNE   LOOP1          ;不为 0 转,为 0 顺执
       RJMP   LOOP           ;反复测试
```

Words：　1（2 bytes）

Cycles：　1

3. 带进位减法

SBC—带进位减

说明：两个寄存器随着 C 标志相减，结果放到目的寄存器 Rd 中。

操作：Rd←Rd—Rr—C

语法：	操作码：	程序计数器：
SBC Rd Rr	0≤d≤31，0≤r≤31	PC←PC＋ 1

16 位操作码

0000	10rd	dddd	rrrr

状态寄存器（SREG）和布尔格式：

I	T	H	S	V	N	Z	C
-	-	⇔	⇔	⇔	⇔	⇔	⇔

H： /Rd3·Rr3＋Rr3· R3＋R3·/Rd3　　N： R7
S： N⊕V　　Z： /R7·/R6·/R5·/R4·/R3·/R2·/R1·/R0.Z
V： Rd7· /Rr7· /R7＋ /Rd7· Rr7· R7　　C： /Rd7·Rr7＋Rr7·R7＋R7·/Rd7

例子：（实践操作程序 4423.ASM）

```
LP: LDI R22,$80      ;寄存器装数(R22)=$80
    LDI R20,$80      ;(R20)=$80
    LDI R23,$23      ;(R23)=$23
    LDI R21,$11      ;(R21)=$11
    ADD R22,R20      ;(R22)=$00,C=1
LP1:SBC  R23,R21     ; 也可单步执行到此行,在调试窗口的对应寄存器 R23,R21 输入数据
                     ;(R23)-(R21)-(C)=
    CPI  R23,$00     ;R23 的内容与立即数$00 比较,
    BRNE LP1         ;R23 的内容不 0 为转,为 0 顺执
    RJMP LP          ;反复测试
```

WOrds： 1（2 byteS）
Cycles： 1

4. 带进位立即数减

SBCI—带进位立即数减

说明：寄存器和立即数随着 C 标志相减，结果放到目的寄存器 Rd 中。

操作：　　Rd←Rd— K— C

语法：	操作码：	程序计数器：
SBCI Rd K	16≤d≤31，0≤K≤255	PC←PC＋1

16 位操作码：

0100	KKKK	dddd	KKKK

状态寄存器（SRE）和布尔格式：

I	T	H	S	V	N	Z	C
-	-	⇔	⇔	⇔	⇔	⇔	⇔

H： /Rd3·K3＋K3·R3＋R3·/Rd3　　N： R7
S： N⊕V　　Z： /R7·/R6·/R5·/R4·/R3·/R2·/R1·/R0·Z
V： Rd7·/K7·/R7＋/Rd7·K7·R7　　C： /Rd7·K7＋K7·R7＋R7·/Rd7

例子：（实践操作程序 4424.ASM）

```
  LP:SEC               ;(C)=1
     LDI   R16,$44     ;(R16)=$44
    SUBI   R16,$22     ;(R16)减立即数$22
```

```
LP1:SBCI    R16, $11    ; 也可单步执行到此行,在调试窗口的对应寄存器 R16 输入数据
                        ;(R16)-$11-1
      CPI    R16,$00    ;比较 R16 内容是否为$00
      BRNE    LP1       ;(R16)不 0 为转,为 0 顺执
      RJMP    LP        ;反复测试

Words:    1（2 bytes）
Cycles:   1
```

5. 立即数减法（字）

SBIW—立即数减法

说明：双寄存器与立即数（0～63）减，结果送双寄存器。该指令操作于四个以上的寄存器对，比较适合对指针寄存器操作。

操作：Rdh ： Rdl←Rdh:Rdl-K

语法：	操作码：	程序计数器：
SBIW Rdl， K	dl∈{24,26，28,30}， 0≤K≤63	PC←PC＋ 1

16 位操作码：

1001	0111	KKdd	KKKK

状态寄存器（SREG）和布尔格式：

I	T	H	S	V	N	Z	C
-	-	—	⇔	⇔	⇔	⇔	⇔

S：NV　　V：Rdh7·/R15　　N：R15　　C：R15·/Rdh7

Z：/R15·/R14·/R13·/R12·/R11·/R10·/R9·/R8·/R7·/R6·/R5·/R4·/R3·/R2·/R1·/R0·Z

例子：（实践操作程序 4425.ASM）

```
LP:LDI    R24,5    ;寄存器装入立即数,寄存器必须符合 16≤R≤31
   LDI    R28,63   ;5,63 为十进制数,十六进制为 0X3F
   sbiw   r24,1    ;
   sbiw   r28,60   ;60 为十进制数
   RJMP   LP       ;反复测试

Words:    1（ 2  bytes）
Cycles:   1
```

6. 减 1 指令

DEC 一减 1

说明：寄存器 Rd 的内容减 1，结果送目的寄存器 Rd 中。该操作不改变 SREG 中的 C 标志，所以 DEC 指令允许在多倍字长计算中用作循环计数。当对无符号值操作时，仅有 BREQ(不相等转移)和 BRNE(不为零转移)指令有效。当对二进制补码值操作时，所有的带符号转移指令都有效。

操作：Rd←Rd－1

语法：	操作码：	程序计数器：
DEC Rd	0≤d≤31	PC←PC ＋1

16 位操作码：

1001	010d	dddd	1010

状态寄存器（SREG）和布尔格式：

I	T	H	S	V	N	Z	C
-	-	—	⇔	⇔	⇔	⇔	⇔

S： N⊕V　　N：R7

V：/R7·R6·R5·R4·R3·R2·R1·R0　　Z：/R7·/R6·/R5·/R4·/R3·/R2·/R1·/R0

例子：（实践操作程序 4426.ASM）

```
LP:    LDI R16,$04   ;多级-1,可作延时子程序用
LOOP:  DEC R16       ; 也可单步执行到此行,在调试窗口的对应寄存器 R16 输入数据
                     ;(R16)-1
       BRNE  LOOP    ;(R16)不为 0 转,为 0 顺执
       DEC   R16     ;第一次((R16)=$00)-1=$FF
       BRNE  LP      ;(R16) 不为 0 转,为 0 顺执
        RET          ;子程序返回
```

Words： 1（2 byteS）

Cycles： 1

4.4.3 乘法指令

注意：AVR 单片机只有 ATmega161 和 Atmeg103L 有乘法指令

MUL 一乘法

说明：该指令完成 8 位 X 8 位→ 16 位的无符号数乘法操作。被乘数 Rr 和乘数 Rd 是两个寄存器。16 位结果放在 R1（高字节）和 R0（低字节）中。注意，如果被乘数和乘数选择了 R0 或 R1，则当进行乘法后，结果将溢出。

操作： R1，R0←Rr*Rd

语：	操作码：	程序计数器：
MUL Rd, Rr	0≤d≤31， 0≤r≤31	PC←PC＋ 1

16 位操作码：

0001	11rd	dddd	rrrr

状态寄存器（SREG）和布尔格式：

I	T	H	S	V	N	Z	C
-	-	—	—	—	—	—	⇔

C： R15

例子：（实践操作程序 4431.ASM,因无相应器件配制文件*.inc,所以汇编时有出错提示）

```
  lp: ldi r6,$04
      ldi r5,#05
      mul r6,r5    ;乘法
      mov r5,r1    ;保存乘积高位
      mov r5,r0    ;保存乘积低位
     rjmp lp
```

Words: l（ 2 byteS）
Cycles: 2

4.4.4 取反码指令

COM — — 取二进制反码
说明: 该指令完成寄存器 Rd 的二进制反码操作。
操作: Rd←$FF — Rd

语法:	操作码:	程序计数器:
COM Rd	0≤d≤31	PC←PC＋ 1

16 位操作码:

1001	010d	dddd	0000

状态寄存器（SREG）和布尔格式:

I	T	H	S	V	N	Z	C
-	-	—	⇔	0	⇔	⇔	1

S: N⊕V Z: /R7./R6./R5./R4./R3./R2./R1./R0
V: 0 C: 1
N: R7
例子: (实践操作程序 4441.ASM)

```
    lp:ldi  R24,$AA
       com  r24       ;取反
       cpi  r24,$aa   ;(r24)与$aa 比较相等吗?
       breq lp1
  lp1:COM  R24        ;取反
      cpi  r24,$aa    ;(r24)与$aa 比较相等吗?
      breq lp         ;相等,反复测试
      BREQ LP
```

Words: 1（ 2 bytes）
Cycles: 1

4.4.5 取补指令

NEG— 二进制补码
说明: 寄存器 Rd 的内容转换成二进制补码, 值$80 是不改变的。
操作: R←$00—Rd

语法:	操作码:	程序计数器:
NEG Rd	0≤d≤31	PC←PC＋ 1

16 位操作码:

1001	010d	dddd	0001

状态寄存器（SREG）和布尔格式：

I	T	H	S	V	N	Z	C
-	-	⇔	⇔	⇔	⇔	⇔	⇔

H：R3./Rd3　　N：R7
S：N⊕V　　Z：/R7./R6./R5./R4./R3./R2.R1./R0
V：R7./R6./R5./R4./R3./R2./R1./R0　　C：R7＋ R6＋ R5＋ R4＋ R3＋ R2＋ R1＋ R0

例子：（实践操作程序 4451.ASM）

```
LP:  SUB  R11,R0 ;开始时在调试窗口中的对应寄存器输入数据
                 ;设(r11)=0b1010101=$AA    (r0)=0b10011001=$99
     BRPL LP1    ;(r11)为正数转移
LP1: NEG  R11
     BRMI LP     ;(r11)为负数转移
```

Words：1（2 bytes）
Cycles：1

4.4.6 比较指令

1. 寄存器比较

CP—比较

说明：该指令完成两个寄存器 Rd 和 Rr 相比较操作，而寄存器的内容不改变。该指令后能使用所有条件转移指令。

操作：Rd—Rr

语法：	操作码：	程序计数器：
CP Rd Rr	0≤d≤31，0≤r≤31	PC←PC＋ 1

16 位操作码：

1001	01rd	dddd	rrrr

状态寄存器（SREG）和布尔格式：

-	-	⇔	⇔	⇔	⇔	⇔	⇔

H：/Rd3.Rr3＋Rr3.R3＋R3./Rd3　　N：R7
S：N⊕V　　Z：Rd7.Rr7.Rr7.R7＋ R7.Rd7
V：Rd7./Rd7./R7＋/Rd7.Rr7.R7　　C：/Rd7.Rr7＋Rr7.R7＋R7./Rd7

例子：（实践操作程序 4461.ASM）

```
lp:cp r24,r19    ;单步执行到此行,开始时在调试窗口的对应寄存器输入数据
                 ;第一次操作设:(r24)=$AA (r19)=$55
                 ;第二次操作设:(r14)=$11 (r19)=$55
  brsh  lp1      ;(r24)≥(r19)则转 lp1, (r24)小于(r19)顺执
  rjmp lp2       ;
lp1:sub  r24,r19 ;(r24)相减(r19)
  brne  lp       ;(r24)不为 0 转,为 0 顺执
lp2:adiw r24,$11 ;立即数加,要求 d∈{ 24 26 28 30 }, 0≤K≤63
```

```
    rjmp lp          ;反复测试

    Words:    l ( 2    bytes)
    CyCICS:   2
```

2. 带进位比较

CPC 一带进位比较

说明: 该指令完成寄存器 Rd 的值和寄存器 Rr 加前位进位的值相比较操作。而寄存器的内容不改变。该指令后能使用所有条件转移指令。

操作: Rd—Rr—C

语法:	操作码:	程序计数器:
CPC Rd Rr	0≤d≤31, 0≤r≤31	PC←PC+1

16 位操作码:

0000	01rd	dddd	rrrr

状态寄存器 (SREG) 和布尔格式:

I	T	H	S	V	N	Z	C
-	-	⇔	⇔	⇔	⇔	⇔	⇔

H: /Rd3·Rr3+Rr3·R3+R3·/Rd3　　N: R7
S: N⊕V　　Z: /R7·/R6· R5·R4·R3·R2· R1·R0·Z
V: Rd7·/Rr7·/R7+/Rd7·Rr7·R7　　C: /Rd7·Rr7+Rr7·R7+R7·/Rd7

例子: (实践操作程序 4462.ASM)

```
 cp r2,r0      ; 单步执行到此行,开始时在调试窗口的对应寄存器输入数据
               ;设:(R2)=$AA,(R0)=$55
 lp:sec        ;(c)=1
cpc r3,r1      ;第一次操作设:(r3)=$11, (r1)=$10,
               ;第二次操作设:设:(R3)=$12,(R1)=$10,
  brne  lp1    ;比较不相等转移,相等顺执
 rjmp lp2      ; 相对转移
  lp1:inc r1   ;+1
  rjmp lp
lp2:dec r1     ;-1
    rjmp lp    ;反复测试

    words:     1 ( 2   bytes)
    Cycles:     l
```

3. 立即数比较

CPI——带立即数比较

说明: 该指令完成寄存器 Rd 和常数的比较操作。寄存器的内容不改变。该指令后能使用所有条件转移指令。

操作: Rd—K

语法:	操作码:	程序计数器:
CPI Rd, K	16≤d≤31, 0≤ K≤255	PC←PC+ 1

16 位操作码:

0011	KKKK	dddd	KKKK

状态寄存器（SREG）和布尔格式:

I	T	H	S	V	N	Z	C
-	-	⇔	⇔	⇔	⇔	⇔	⇔

H: /Rd3 • K3＋K3 • R3＋R3 • /Rd3　　N: R7
S: N⊕V　　Z: /R7 • /R6 • /R5 • /R4 • /R3 • /R2 • /R1 • /R0
V: Rd7 • /K7 • /R7＋/Rd7 • K7 • R7　　C: /Rd7 • K7＋K7 • R7＋R7 • /Rd7

例子: (实践操作程序 4463.ASM)

```
    LP:CPI   R19,3      ; 单步执行到此行,开始时在调试窗口的对应寄存器输入数据
                        ;设(R19)=4
       BRNE LP1         ;不相等转,相等顺执
       INC R19          ;(R19)+1
       RJMP LP          ;反复测试
     LP1:DEC R19        ;(R19)-1
       RJMP LP          ;反复测试
```

Words: 1（ 2 bytes）
Cycles: 1

4.4 .7 逻辑与指令

逻辑与输入		输出
A	B	Y
L	L	L
L	H	L
H	L	L
H	H	H

1. 寄存器逻辑与

AND 一逻辑与 ;全 1 为 1,有 0 即 0,
说明: 寄存器 Rd 和寄存器 Rr 的内容为逻辑与，结果送目的寄存器 Rd。
应用: 清 0,使某位为 0,用 0 去与;保留,用 1 去逻辑与;代硬件与门
操作: Rd←Rd • Rr ;

语法:	操作码:		程序计数器:
AND Rd,Rr	0≤d≤31	0≤d≤31	PC←PC＋1

16 位操作码:

0010	00rd	dddd	rrrr

状态寄存器（SREG）和布尔格式:

I	T	H	S	V	N	Z	C
-	-	∀	⇔	⇔	⇔	⇔	—

S：N⊕V　　N：R7
V：0　　Z：/R7·/R6·/R5·/R4·/R3·/R2·/R1·/R0
例子：(实践操作程序 4471.ASM)

```
LP:add r2,r3        ; 单步执行到此行,开始时在调试窗口的对应寄存器输入数据
                    ;设:(R2)=0B1000 0000,(R3)=0B0001 0011
  LDI R16,1         ;(R16)=0B0000 0001,
   and  r2,r16      ;(R2)=
   RJMP LP          ;反复测试
```

Words：1（2 bytes）
Cycles：1

2. 带立即数与

ANDI—立即数逻辑与 ;全 1 为 1,有 0 即 0
说明：寄存器 Rd 的内容与常数逻辑与，结果送目的寄存器 Rd。
应用：清 0,使某位为 0,用 0 去与;保留,用 1 去逻辑与;代硬件与门
操作：Rd←Rd·K ;

语法：	操作码：	程序计数器：
ANDI Rd K	16≤d≤31， 0≤K≤255	PC←PC＋ 1

16 位操作码：

0001	11rd	dddd	rrrr

状态寄存器（SREG）和布尔格式：

I	T	H	S	V	N	Z	C
-	-	—	⇔	0	⇔	⇔	—

S:N⊕V　　N： R7
V:0　　Z： /R7·/R6·/R5·/R4·/R3·/R2·/R1·/R0
例子：(实践操作程序 4472.ASM)

```
LP:  andi   r17,$0F ; 单步执行到此行,开始时在调试窗口的对应寄存器输入数据
                    ;设:(r17)=$F0 ,(R18)=$EF ,(R19)=$FF,单步执行后,(R17)=
     andi   r18,$10 ;(R18)=
    andi    r19,$AA ;(R19)=
    RJMP  LP        ;反复测试
```

Words：l（2 bytes）
Cycles：1

3. 清除寄存器位

CBR—清除寄存器指定位
说明：清除寄存器 Rd 中的指定位。利用寄存器 Rd 的内容与常数表征码 K 的补码相与完成的，其结果放在寄存器 Rd 中。
操作：Rd←Rd ▪ ($FF—K)

语法：	操作码：	程序计数器：

CBR Rd K 16≤d≤31, 0≤K≤255 PC←PC+1

16 位操作码:

0111	KKKK	dddd	KKKK

状态寄存器（SREG）和布尔格式:

I	T	H	S	V	N	Z	C
-	-	—	⇔	0	⇔	⇔	—

S: N⊕V N: R7

V: 0 Z: /R7·/R6·/R5·/R4·/R3·/R2·/R1·/R0

例子: (实践操作程序 4473.ASM)

```
lp:cbr R16,$F0    ; 单步执行到此行,在调试窗口的对应寄存器输入数据
                  ;(r16)=$FF
 CBR R18,1        ;(r18)=$80
 INC R16
 INC R18
  rjmp lp         ;反复输入数据测试
```

Words: 1（2 bytes）

Cycles: 1

4. 测试零或负

TST 测试零或负

说明: 测试寄存器是否是零或是负。完成同一寄存器之间的逻辑与操作，而寄存器内容不改变。

操作: Rd←Rd Rd

语法:	操作码:	程序计数器
TST Rd	0≤d≤31	PC←PC+1

16 位操作码:

0011	00dd	dddd	dddd

状态寄存器（SREG）和布尔格式:

I	T	H	S	V	N	Z	C
-	-	—	⇔	0	⇔	⇔	—

S: N⊕V N: R7

V:0 Z: /R7·/R6·/RS·/R4·/R3·/R2·/R1·/R0

例子: (实践操作程序 4474.ASM)

```
sub r0,r2       ; 单步执行到此行,在调试窗口的对应寄存器输入数据
                ;并打开状态寄存器 SREG 观察窗口
                ;(r0)=$aa,(r2)=$aa
                ;(r0)=$77,(r2)=$80
lp: tst   r0    ; 测试寄存器 r0 是否是零或是负,
```

```
    breq   lp1   ;如果零标志(Z)位为 1,则转移,为 0 顺执
    dec r0       ;-1
    rjmp lp      ; 再测试
lp1:inc r0       ;+1
    rjmp lp      ;再测试

    Words:     1 ( 2   bytes)
    Cycles:    1
```

4·4·8 逻辑或指令

逻辑或输入		输出
A	B	Y
L	L	L
L	H	H
H	L	H
H	H	H

1. 寄存器逻辑或

OR 一逻辑或 ;有 1 即 1,全 0 为 0,

应用: 置数,使某位为 1,用 1 去或;保留,用 0 去逻辑或;代硬件或门

说明: 完成寄存器 Rd 与寄存器 Rr 的内容逻辑或操作,结果进目的寄存器 Rd 中。

操作: Rd←Rd ∨ Rr

语法:	操作码:	程序计数器:
OR Rd Rr	0≤d≤31, 0≤r≤31	PC←PC+ 1

16 位操作码:

0010	10rd	dddd	rrrr

状态寄存器(SREG)和布尔格式:

I	T	H	S	V	N	Z	C
-	-	—	⇔	0	⇔	⇔	—

S: N⊕V N: R7

V: 0 Z: /R7·/R6·/R5·/R4·/R3·/R2·/R1·/R0

例子: (实践操作程序 4481.ASM)

```
  or  r19,r16   ;逻辑或,单步执行到此行,在调试窗口的对应寄存器输入数据
                ;(R19)=$AA   , (R16)=$44   , (R18)=0B0100 1111=$4F
                ;                          (R18)=0B0000 1111=$0F
  LP:bst  r18,6 ;R18 中的 6 位内容到 SREG 中 T 标志,观察 SREG 中 T 标志的变化
    brts ok     ;SREG 中标志为 1 转移,为 0 顺执
    SER R18     ;置位 R18
    RJMP LP     ;反复测试
  ok:CLR R18    ;清零 R18
    RJMP LP     ;反复测试

Words:    l (2  bytes)
```

Cycles: 1

2. 带立即数或

ORI—立即数逻辑或 ;功能：保留(屏蔽)数据,置数(使某位为 1)

说明：完成寄存器 Rd 的内容与常量逻辑或操作，结果送目的寄存器 Rd 中。

操作： Rd←Rd∨K

语法：	操作码：	程序计数器：
ORI Rd K	16≤d≤31， 0≤K≤255	PC←PC＋ 1

16 位操作码：

0110	KKKK	dddd	KKKK

状态寄存器（SREG）和布尔格式：

I	T	H	S	V	N	Z	C
-	-	—	⇔	0	⇔	⇔	—

S:N⊕V N：R7

V：0 Z：/R7·/R6·/R5·/R4·/R3·/R2·/R1·/R0

例子：(实践操作程序 4482.ASM)

```
  LP:ori r19,$F0    ;立即数或,单步执行到此行,在调试窗口的对应寄存器输入数据
                    ;(R19)=$A0   , (R17)=$45
                    ;(R19)=0B0110 1111=$6F   , (R17)=0B1111 0000=$F0
     ori r17,1      ; 立即数或
     lsl r19        ;r19 逻辑左移
     lsr r17        ;r17 逻辑右移
     RJMP LP        ;反复测试
```

Words： 1（ 2 bytes）

Cycles： 1

3. 置寄存器位

SBR—寄存器位置位

说明：对寄存器 Rd 中指定位置位。完成寄存器 Rd 和常数表征码 K 之间的逻辑直接数或（ORI），结果送目的寄存器 Rd。

操作： Rd←Rd ∨K

p95

语法：	操作码：	程序计数器：
SBR Rd K	16≤d≤31，0≤K≤255	PC←PC＋1

16 位操作码：

0110	KKKK	dddd	KKKK

状态寄存器（SREG）和布尔格式：

I	T	H	S	V	N	Z	C
-	-	—	⇔	0	⇔	⇔	—

S：N⊕V　　N：R7
V：0　　Z：/R7·/R6·/R5·/R4·/R3·/R2·/R1·/R0

例子：（实践操作程序 4483.ASM）

```
                ;设:(R16)=$F0     ,(R17)=$80
 lp:sbr r16,3   ;对 R16 的(0B0000 0011)第 1、0 位置为 1,执行后(R16 )=
    sbr r17,$17 ;对 R17 的(0B0001 0111)第 4、2、1、0 位置 1,执行后(R17)=
    INC R16     ;+1
    DEC R17     ;-1
   rjmp lp      ;反复试验
```

Words：　1（ 2　bytes）
Cycles：　1

4. 置寄存器

SER一置位寄存器的所有位
说明：直接装入$FF 到寄存器 Rd.
操作：　Rd←$FF

语法：	操作码：	程序计数器：
SER　Rd	16≤d≤31	PC←PC＋ 1

16 位操作码：

1110	1111	dddd	1111

状态寄存器（SREG）和布尔格式：

I	T	H	S	V	N	Z	C
—	—	—	—	—	—	—	—

例子：（实践操作程序 4484.ASM）

```
LP:CLR R16     ;(R16)清零
 ser r17       ;(R17)置$FF
 out $18,r16   ;输出到 B 口,打开 I/O 寄存器窗口看$18 变化
 out $18,r17   ;输出到 B 口,打开 I/O 寄存器窗口看$18 变化
   INC R16     ;+1
   INC R17     ;+1
   RJMP LP     ;反复试验
```

Words：　1（ 2　bytes）
Cycles：　1

4.4.9 运辑异或指令

1. 寄存器异或

EOR—异或　;输入相同输出为 0,输入不同输出为 1;也称同或(清零);也称互斥(置 1);见下真值表

说明：完成寄存器 Rd 和寄存器 Rr 的内容相逻辑异或操作，结果送目的寄存器 Rd.

异或输入		输出
A	B	Y
L	L	L
L	H	H
H	L	H
H	H	L

操作: Rd←Rd⊕Rr
语法: 操作码: 程序计数器:
EOR Rd,Rr 0≤d≤31, 0≤r≤31 PC←PC＋ 1

16 位操作码:

0010	01rd	dddd	rrrr

状态寄存器（SREG）和布尔格式:

I	T	H	S	V	N	Z	C
—	—	—	⇔	0	⇔	⇔	—

S: N⊕V N: R7
V:0 Z: /R7·/R6·/R5·/R4·/R3·/R2·/R1·/R0
例子: (实践操作程序 4491.ASM)

```
LP:eor  r4,r4    ;设:(R4)=0B1010 0011,相同数异或为清零,也可称同或清零
   eor  r0,r22   ;设:(R0)=0B1010 0101 设:(R22)=0B0101 0011
   SWAP R4       ;半字节交换
   SWAP R0       ;半字节交换
   SWAP R22      ;半字节交换
   RJMP LP       ;反复实验
```

Words: 1（ 2 bytes）
Cycles: 1

2. 清除寄存器

CLR——寄存器清零

说明: 寄存器清零。该指令采用寄存器 Rd 与自己的内容相异或实现的。寄存器的所有位都被清零。

操作: Rd←Rd ⊕Rd
语法: 操作码: 程序计数器:
CLR Rd 0≤d≤31 PC←PC＋ 1

16 位操作码:

0001	11rd	dddd	rrrr

状态寄存器（SREG）和布尔格式：

I	T	H	S	V	N	Z	C
—	—	—	0	0	0	1	—

S：0 N：0
V：0 Z：1

例子：（实践操作程序 4492.ASM）

```
          LP:CLR  R18         ;R18 清零
          Lp1:inc r18         ;+1
              CPI R18,$05     ;R18 内容与立即数比较
              brne lp1        ;不相等转,相等顺执
              RJMP LP         ;循环测试
```

Words： 1（ 2 bytes）
Cycles： 1