

是 16 位宽，允许在当前程序存储器空间 64K 字（128K 字节）内跳转。

IJMP—间接跳转**优点**:转移范围大,**缺点**:作为子程序模块,移值时需修改转移地址,希望在子程序中不要使用!不要给自己带来麻烦!

注意:只能到你设计的硬件电路所具有的空间,你的器件可有这条指令吗?

操作: $PC \leftarrow Z(15-0)$

$PC(15-0) \leftarrow Z(15-0)$

语法: IJMP 操作码: None 程序计数器: See Operation
16 位操作码:

1001	0100	XXXX	1001
------	------	------	------

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 4512.ASM)

```

ORG $0010
LP:MOV R30,R0 ;设(R0)为 Z 寄存器低 8 位地址,R30 为 Z 间接地址低 8 位
    MOV R31,R1 ;设(R1)为 Z 寄存器高 8 位地址,R31 为 Z 间接地址高 8 位
    ijmp ;根据 Z 寄存器的内容跳转
.ORG $0200
LDI R20,$11 ;地址$0201 ,如设 (R0)=$01, (R1)=02,执行 IJMP 到此行
LDI R21,$22 ;地址$0202 ,如设 (R0)=$02, (R1)=02,执行 IJMP 到此行
LDI R22,$33 ;地址$0203 ,如设 (R0)=$03, (R1)=02,执行 IJMP 到此行
LDI R23,$44 ;地址$0204 ,如设 (R0)=$04, (R1)=02,执行 IJMP 到此行
LDI R24,$55 ;地址$0205 ,如设 (R0)=$05, (R1)=02,执行 IJMP 到此行
LDI R25,$66 ;地址$0206 ,如设 (R0)=$06, (R1)=02,执行 IJMP 到此行
IJMP ;地址$0207 ,如设 (R0)=$07, (R1)=02,执行 IJMP 到此行
;又根据 Z 寄存器的内容跳转到指定地址
    
```

Words: 1 (2 bytes)
Cycles: 1

3. 长跳转

JMP—跳转

说明: 在整个程序存储空间 4M (字) 内跳转, 见 RJMP。

JMP—跳转**优点**:可跳转到程序存储器空间 4M (字) 任何地方;

缺点:不适宜子程序模块中使用;

注意:只能到你设计的硬件电路所具有的空间,你的器件可有这条指令吗!

操作: $PC \leftarrow k$

语法: JMP k 操作码: $0 \leq k \leq 4M$ 程序计数器: $PC \leftarrow k$

32 位操作码:


```

brbs 6,LP1      ; SREG 中的 6 位被置位(T=1)则转移,若为零顺执
SET             ;置 T 标志为 1
RJMP LP        ;反复测试
LP1:CLT        ;T 标志清零
RJMP LP        ;反复测试
    
```

Words: 1 (2 bytes)
 Cycles: 1 if condition is false
 2 if conditlon is true

2. 状态寄存器中位清零转移

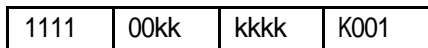
BRBC——SREG 中的位被清零转移

说明: 条件相对转移,测试 SREG 的某一位,如果该位被清零,则相对 PC 值转移. 这条指令相对 PC 转移的方向为: $PC - 64 \leq \text{目的} \leq PC + 63$. 参数 K 为 PC 的偏移,用 2 的补码表示。

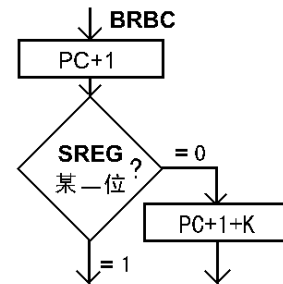
操作: If SREG(S)=0 then $PC \leftarrow (PC+1)+k$, else $PC \leftarrow PC + 1$

语法: BRBC S, k 操作码: 程序计数器:
 $0 \leq S \leq 7, -64 \leq k \leq +63$ $PC \leftarrow (PC+1) + k$
 $PC \leftarrow PC + 1$

16 位操作码:



状态寄存器 (SRE) 和布尔格式:



例子: (实践操作程序 4522.ASM)

```

CPI R20,5      ;R20 中内容与立即数 05 比,设 (R20)=6 或 (R20)=5
LP:BRBC 1,LP1  ;SREG 中位被清零则转移,为 1 顺执
CLZ
RJMP LP        ;反复测试
LP1:SEZ        ;Z=1
RJMP LP        ;反复测试
    
```

Words: 1 (2 bytes)
 Cycles: 1 if condition is false
 2 if condition is true

3. 相等转移

BREQ 相等转移

说明: 条件相对转移, 测试零标志 (Z), 如果 Z 位被置位, 则相对 PC 值转移。如果在执行 CP、CPI、SUB 或 SUBI 指令后立即执行该指令, 且当寄存器 Rd 中无符号或有符号二进制数与寄存器 Rr 中无符号或有符号 H 进制数相等时, 转移将发生。该指令相对 PC 转移的方向

为： $PC-64 \leq \text{目的} \leq PC+63$ 。参数 K 为 PC 的偏移，用 2 的补码表示（相当于指令 BRBSI, K）。

操作：If $Rd=Rr(z=1)$ then $PC \leftarrow (PC+1)+k$, $PC \leftarrow PC+1$

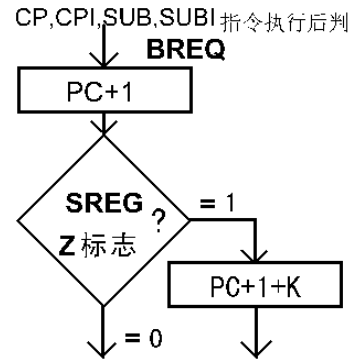
语法：操作码：程序计数器：
 BREQ k $-64 \leq k \leq +63$ $PC \leftarrow (PC+1)+k$
 $PC \leftarrow PC+1$

16 位操作码：

1111	00kk	kkkk	K001
------	------	------	------

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 4523.ASM)

```

lp:cp r1,r2 ;设: (R1)=$AA. (R2)=$AA
breq lp1 ;相等转移, 不相等顺执, 请同时观察 Z 标志
inc r1 ;+1
rjmp lp ;反复验证
lp1:dec r1 ;-1
rjmp lp ;反复验证
    
```

Words: 1 (2 bytes)
 Cycles: 1 if condition is false
 2 if condition is true

4. 不相等转移

BRNE—不相等转移

说明：条件相对转移，测试零标志 (Z)，如果 Z 位被清零，则相对 PC 值转移。如果在执行 CP、CPI、SUB 或 SUBI 指令后立即执行该指令，且当在寄存器 Rd 中的无符号或带符号二进制数不等于寄存器 Rr 中的无符号或带符号二进制数时，转移将发生。该指令相对 PC 转移的方向为： $PC-64 \leq \text{目的} \leq PC+63$ 。参数 K 为 PC 的偏移，用 2 的补码表示（相当于指令 BRBCIK）。

操作：If $Rd \neq Rr(Z=0)$ then $PC \leftarrow (PC+1)+k$, else $PC \leftarrow PC+1$

语法：操作码：程序计数器：
 BRNE k $-64 \leq k \leq +63$ $PC \leftarrow (PC+1)+k$
 $PC \leftarrow PC+1$

16 位操作码：

1111	01kk	kkkk	k001
------	------	------	------

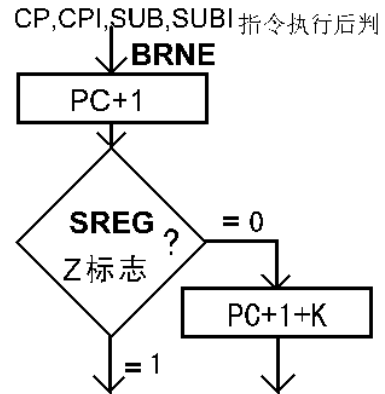
状态寄存器 (SREG) 和布尔格式：

I	T	H	S	V	N	Z	C

```

例子: (实践操作程序 4524.ASM)
lp:cp r1,r2 ;设:(R1)=$AB.(R2)=$AA
brne lp1 ;不相等转移,相等顺执,请同时观察 Z 标志
inc r1 ;+1
rjmp lp ;反复验证
lp1:dec r1 ;-1
rjmp lp ;反复验证
    
```

Words: 1 (2bytes)
 Cycles: 1 if condition is false
 2 if condition is true



5. C 标志位置位转移

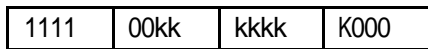
BRCS-进位位置位转移

说明: 条件相对转移, 测试进位标志 (C), 如果 C 位被置位, 则相对 PC 值转移。这条指令相对 PC 转移的方向为: $PC - 64 \leq \text{目的} \leq PC + 63$ 。参数 K 为 PC 的偏移, 用 2 的补码表示 (相当于指令 BRBS 0, K)。

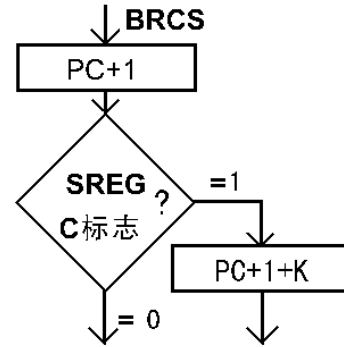
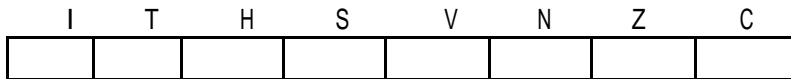
操作: If C=1 then $PC \leftarrow (PC+1)+k$, else $PC \leftarrow PC+1$

语法: 操作码: 程序计数器:
 BRCS k $-64 \leq k \leq +63$ $PC \leftarrow (PC+1)+k$
 $PC \leftarrow PC+1$

16 位操作码:



状态寄存器 (SREG) 和布尔格式:



```

例子: (实践操作程序 4525.ASM)
SEC ;C=1,请同时观察 C 标志
    
```

```

LP:BRCS LP1 ;C=1 转,C=0 顺执
SEC ;C=1
RJMP LP ;重复试验
LP1:CLC ;C=0
RJMP LP ;重复试验
    
```

Words: 1 (2 bytes)
 Cycles: 1 if condition is false
 2 if condition is true

6. C 标志位清除转移

BRCC——进位位清除转移

说明: 条件相对转移, 测试进位标志 (C), 如果 C 位被清除, 则相对 PC 值转移。这条指令相对 PC 转移的方向为: $PC - 64 \leq \text{目的} \leq PC + 63$ 。参数 K 为 PC 的偏移, 用 2 的补码表示

(相当于指令 BRBC 0, K)。

操作: If C=0 then PC←(PC+1)+k, else PC←PC+1

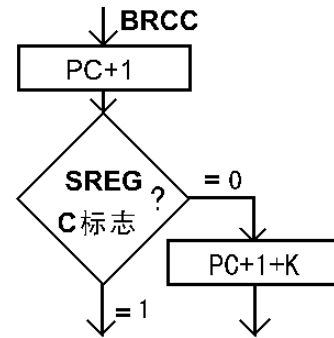
语法: 操作码: 程序计数器:
BRCC k -64 ≤ k ≤ +63 PC←(PC+1)+k
 PC←PC+1

16 位操作码:

1111	01kk	kkkk	K000
------	------	------	------

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 4526.ASM)

```

    CLC            ;C=0,请同时观察 C 标志
LP:BRCC LP1       ;C=0 转,C=1 顺执
    CLC           ;C=0
    RJMP LP       ;重复试验
LP1:SEC           ;C=0
    RJMP LP       ;重复试验
  
```

Words: 1 (2 bytes)
Cycles: 1 if condition is false
 2 if condition is true

7. 大于或等于转移

BRSH-大于等于转移 (无符号)

说明: 条件相对转移, 测试进位标志 (C), 如果 C 位被清零, 则相对 PC 值转移。如果在执行 CP、CPI、SUB 或 SUBI 指令后立即执行该指令, 且当在寄存器 Rd 中无符号二进制数大于等于寄存器 Rr 中无符号二进制数时, 转移将发生。该指令相对 PC 转移的方向为: PC - 64 ≤ 目的 ≤ PC + 63。参数 K 为 PC 的偏移, 用 2 的补码表示 (相当于指令 BRBS 0, K)。

操作: If Rd ≥ Rr (C=0) then PC←(PC+1)+k, else PC←PC+1

语法: 操作码: 程序计数器:
BRSH k -64 ≤ k ≤ +63 PC←(PC+1)+ k
 PC←PC+1

16 位操作码:

1111	01kk	kkkk	K000
------	------	------	------

状态寄存器 (SREG) 和布尔格式:

I	T	H	S	V	N	Z	C

2 if condition is true

9. 负数转移

BRMI — 负数转移

说明：条件相对转移，测试负号标志 (N)，如果 N 被置位，则相对 PC 值转移。该指令相对 PC 转移的方向为： $PC - 64 \leq \text{目的} \leq PC + 63$ 。参数 K 为 PC 的偏移，用 2 的补码表示（相当于指令 BRBS 2, K）。

操作：If N=1 then $PC \leftarrow (PC+1) + k$, else $PC \leftarrow PC+1$

语法：操作码：程序计数器：
BRMI k $-64 \leq k \leq +63$ $PC \leftarrow (PC+1) + k$
 $PC \leftarrow PC+1$

16 位操作码：

1111	00kk	kkkk	K010
------	------	------	------

状态寄存器 (SREG) 和布尔格式：

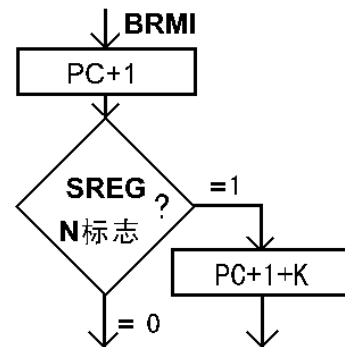
I	T	H	S	V	N	Z	C

例子：(实践操作程序 4529.ASM)

```

lp:subi r18,2 ; 带进位位立即数减,设:(r18)=2
brmi lp1 ; 为负转 N=1,为正顺执 N=0
; 请同时观察 N 标志

inc r18 ;(r18)+1
rjmp lp ;反复实验
lp1:inc r18 ;(r18)+1
inc r18
inc r18
inc r18
rjmp lp ;反复实验
Words: 1 (2 bytes)
Cycles: 1 if condition is false
        2 if condition is true
    
```



10. 正数转移

BRPL — 正数转移

说明：条件相对转移，测试负号标志 (N)，如果 N 被清零，则相对 PC 值转移。该指令相对 PC 转移的方向为： $PC - 64 \leq \text{目的} \leq PC + 63$ 。参数 K 为 PC 的偏移，用 2 的补码表示（相当于指令 BRBC 2, K）。

操作：If N=0 then $PC \leftarrow (PC+1) + k$, else $PC \leftarrow PC+1$

语法：操作码：程序计数器：
BRPL k $-64 \leq k \leq +63$ $PC \leftarrow (PC+1) + k$
 $PC \leftarrow PC+1$

16 位操作码：

1111	01kk	kkkk	K010
------	------	------	------

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 45210.ASM)

```

lp:subi r18,2      ; 带进位位立即数减,设:(r18)=2
BRPL lp1         ; 为正转 N=0,为负顺执 N=1 请同时观察 N 标志
inc r18          ;(r18)+1
inc r18
inc r18
inc r18
rjmp lp          ;反复实验
lp1:dec r18      ;(r18)+1
dec r18
rjmp lp          ;反复实验
    
```

positive: nop

Words: 1 (2 bytes)
 Cycles: 1 if condition is false
 2 if condition is true

11. 大于或等于转移

BRGE—大于或等于转移 (带符号)

说明: 条件相对转移, 测试符号标志 (S), 如果 S 位被清零, 则相对 PC 值转移。如果在执行 CP、CPI、SUB 或 SUBI 指令后立即执行该指令, 且当在寄存器 Rd 中带符号二进制数大于或等于寄存器 Rr 中带符号二进制数时, 转移将发生。该指令相对 PC 转移的方向为: $PC - 64 \leq \text{目的} \leq PC + 63$ 。参数 K 为 PC 的偏移, 用 2 的补码表示 (相当于指令 BRBC 4, K)。

操作: If $Rd \geq Rr$ ($N \vee = 0$) then $PC \leftarrow (PC+1)+k$, else $PC \leftarrow PC+1$

语法: 操作码: 程序计数器:
 BRGE k $-64 \leq k \leq +63$ $PC \leftarrow (PC+1) + k$
 $PC \leftarrow PC + 1$

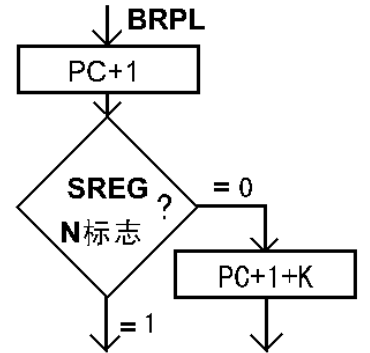
16 位操作码:

1111	01kk	kkkk	K100
------	------	------	------

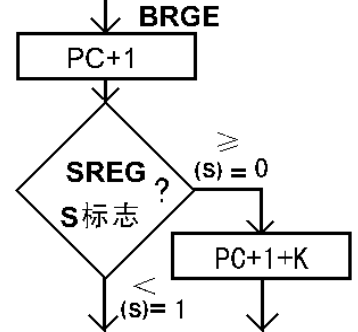
状态寄存器 (SREG) 和布尔格式:

I	T	H	S	V	N	Z	C

例子: (实践操作程序 45211.ASM)



CP,CPI,SUB,SUBI 指令执行后判



```

lp:subi r18,2 ; 带进位位立即数减,设:(r18)=2
brge lp1 ;为大于或等于转 S=0,小于顺执 S=1 请同时观察 S 标志
inc r18 ;(r18)+1
inc r18
inc r18
inc r18
rjmp lp ;反复实验
lp1:dec r18 ;(r18)+1
dec r18
rjmp lp ;反复实验
    
```

Words: 1 (2 bytes)
 Cycles: 1 if condition is false
 2 if condition is true

12. 小于转移

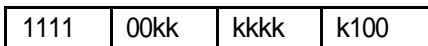
BRLT—小于转移 (有符号)

说明: 条件相对转移, 测试符号标志 (S), 如果 S 位被置位, 则相对 PC 值转移。如果在执行 CP、CPI、SUB 或 SUBI 指令后立即执行该指令, 且当在寄存器 Rd 中带符号二进制数小于在寄存器 Rr 中带符号二进制数时, 转移将发生。该指令相对 PC 转移的方向为: $PC - 64 \leq \text{目的} \leq PC + 63$ 。参数 K 为 PC 的偏移, 用 2 的补码表示 (相当于指令 BRBS 4, K)。

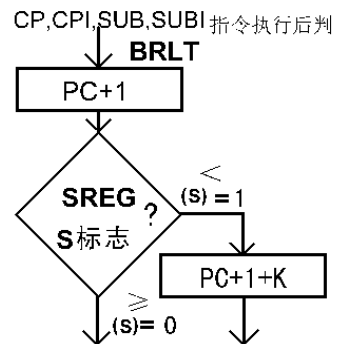
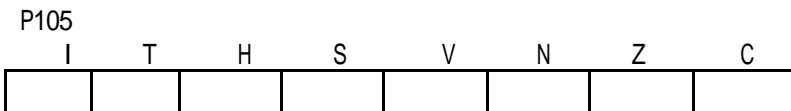
操作: If $Rd < Rr$ ($NV=1$) then $PC \leftarrow (PC+1)+k$, else $PC \leftarrow PC+1$

语法: BRLT k $-64 \leq k \leq +63$
 操作码: 程序计数器: $PC \leftarrow (PC+1) + k$
 $PC \leftarrow PC+1$

16 位操作码:



状态寄存器 (SREG) 和布尔格式:



例子: (实践操作程序 45212.ASM)

```

lp:subi r18,2 ; 带进位位立即数减,设:(r18)=2
brlt lp1 ;为小于于转 S=0,大于或等顺执 S=1 请同时观察 S 标志
inc r18 ;(r18)+1
inc r18
inc r18
inc r18
rjmp lp ;反复实验
lp1:dec r18 ;(r18)-1
dec r18
rjmp lp ;反复实验
    
```

Words: 1 (2 bytes)

Cycles: 1 if condition is false
2 if condition is true

13. 半进位标志置位转移

BRHS -- 半进位标志置位转移

说明: 条件相对转移, 测试半进位标志 (H), 如果 H 被置位, 则相对 PC 值转移。该指令相对 PC 转移的方向为: $PC - 64 \leq \text{目的} \leq PC + 63$ 。参数 K 为 PC 的偏移, 用 2 的补码表示 (相当于指令 BRBS 5, K)。

操作: If H=1 then $PC \leftarrow (PC+1)+k$, else $PC \leftarrow PC+1$

语法: 操作码: 程序计数器:
BRHS k $-64 \leq k \leq +63$ $PC \leftarrow (PC+1) + k$
 $PC \leftarrow PC + 1$

16 位操作码:

1111	00kk	kkkk	k101
------	------	------	------

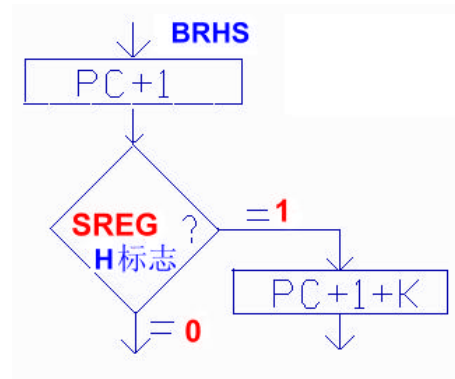
状态寄存器 (SREG) 和布尔格式:

I	T	H	S	V	N	Z	C

例子: (实践操作程序 45213.ASM)

```
lp:add r10,r11      ;加法设:(r10)=5 ,(r11)=5 ,(r12)=5
brhs lp1           ;半进位标志(H)=1 转移
                   ;(H)=0 顺执
dec r10            ;(R10)-1
dec r10
rjmp lp           ;反复实验
lp1:add r10,r12    ;加法
rjmp lp           ;反复实验

Words: 1 (2 bytes)
Cycles: 1 if condition is false
        2 if condition is true
```



14. 半进位标志清零转移

BRHC -- 半进位标志被清零转移

说明: 条件相对转移, 测试半进位标志 (H), 如果 H 位被清零, 则相对 PC 值转移。该指令相对 PC 转移的方向为: $PC - 64 \leq \text{目的} \leq PC + 63$ 。参数 K 为 PC 的偏移, 用 2 的补码表示 (相当于指令 BRBC 5, K)。

操作: If H=0 then $PC \leftarrow (PC+1)+k$, else $PC \leftarrow PC+1$

语法: 操作码: 程序计数器:
BRHC k $-64 \leq k \leq +63$ $PC \leftarrow (PC+1) + k$
 $PC \leftarrow PC + 1$

16 位操作码:

1111	01kk	kkkk	k101
------	------	------	------

状态寄存器 (SREG)和布尔格式:

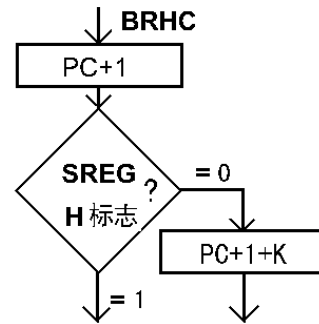
I	T	H	S	V	N	Z	C

例子: (实践操作程序 45214.ASM)

```

lp:add r10,r11      ;加法设:(r10)=5 ,(r11)=5 ,(r12)=5
brhc lp1           ;半进位标志(H)=0 转移,(H)=1 顺执
dec r10            ;(R10)-1
dec r10
rjmp lp            ;反复实验
lp1:add r10,r12     ;加法
rjmp lp            ;反复实验
    
```

Words: 1 (2 bytes)
 Cycles: 1 if condition is false
 2 if conditlon is true



15. T 标志置位转移

BRTS—T 标志被置位转移

说明: 条件相对转移, 测试 T 标志, 如果 T 被置位, 则相对 PC 值转移。该指令相对 PC 转移的方向为: $PC - 64 \leq \text{目的} \leq PC + 63$ 。参数 K 为 PC 的偏移, 用 2 的补码表示 (相当于指令 BRBS 6, K)。

操作: If T=1 then $PC \leftarrow (PC+1)+k$, else $PC \leftarrow PC+1$

语法: 操作码: 程序计数器:
 BRTS k $-64 \leq k \leq +63$ $PC \leftarrow (PC+1) + k$
 $PC \leftarrow PC+1$

16 位操作码:

1111	00kk	kkkk	k110
------	------	------	------

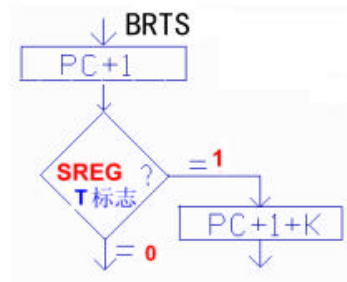
状态寄存器 (SRE) 和布尔格式:

I	T	H	S	V	N	Z	C

例子: (实践操作程序 45215.ASM)

```

lp:bst r3,5        ;R3 中的 5 位到 SREG 中 T 标志, 设:(R3)=$AA
brts LP1          ;(T)=1 转移,(T)=0 顺执
rol r3            ;r3 通过进位位左循环
rjmp lp           ;反复实验
LP1:rol r3         ;r3 通过进位位左循环
rjmp lp           ;反复实验
    
```



Words: 1 (2 bytes)
 Cycles : 1 if condition is false
 2 if conditlon is true

16. T 标志清零转移

BRTC—T 标志被清零转移

说明：条件相对转移，测试 T 标志，如果 T 被清零，则相对 PC 值转移。该指令相对 PC 转移的方向为： $PC-64 \leq \text{目的} \leq PC+63$ 。参数 K 为 PC 的偏移，用 2 的补码表示（相当于指令 BRBC 6, K）。

操作：If T=0 then PC ← (PC+1)+k, else PC ← PC+1

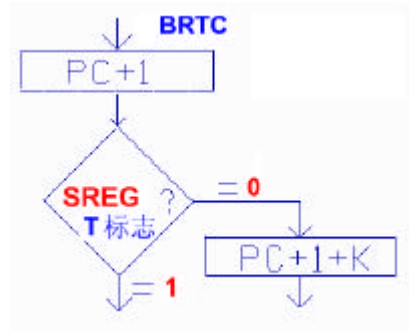
语法：操作码：程序计数器：
 BRTC k $64 \leq k \leq +63$ $PC \leftarrow (PC+1)+k$
 $PC \leftarrow PC+1$

16 位操作码：

1111	01kk	kkkk	k110
------	------	------	------

状态寄存器 (SREG) 和布尔格式：

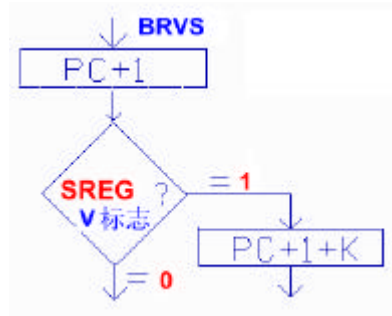
I	T	H	S	V	N	Z	C



例子：(实践操作程序 45216.ASM)

```
bst r3,5 ;R3 中的 5 位到 SREG 中 T 标志, 设 R3=$55
lp:brtc lp1 ;T 标志置 0 转移, 置 1 顺执
clt ;(T)=0
rjmp lp ;反复测试
lp1:set ;(T)=1
rjmp lp ;反复测试
```

Words: 1 (2 bytes)
 Cycles: 1 if condition is false
 2 if condition is true



17. 溢出标志置位转移

BRVS —溢出标志被置位转移

说明：条件相对转移，测试溢出标志 (V)，如果 V 被置位，则相对 PC 值转移。该对 PC 转移的方向为： $PC-64 \leq \text{目的} \leq PC+63$ 。参数 K 为 PC 的偏移，用 2 的补码表示（相当于指令 BRBS3, K）。

操作：If V=1 then PC ← (PC+1)+k, else PC ← PC+1

语法：操作码：程序计数器：
 BRVS k, $-64 \leq k \leq +63$ $PC \leftarrow (PC+1)+k$
 $PC \leftarrow PC+1$

16 位操作码：

1111	00kk	kkkk	k011
------	------	------	------

状态寄存器 (SRE) 和布尔格式:

I	T	H	S	V	N	Z	C

例子: (实践操作程序 45217.ASM)

```

lp:brvs lp1      ;溢出标志位(V)=1 转移,(V)=0 顺执
   sev          ;溢出标志位(V)置 1
   rjmp lp      ;反复测试
lp1:clv         ;溢出标志位(V)置 0
   rjmp lp      ;反复测试
    
```

```

words:      1 ( 2 bytes)
Cycles:     1 if condition is false
            2 if condition is true
    
```

18. 溢出标志清零转移

BRVC——溢出标志被清零转移

说明: 条件相对转移, 测试溢出标志 (V), 如果 V 被清零, 则相对 PC 值转移。该对 PC 转移的方向为: $PC-64 \leq \text{目的} \leq PC+63$ 。参数 K 为 PC 的偏移, 用 2 的补码表示 (相当于指令 BRBC3, K)。

```

操作: IfV=0thenPC←(PC+1)+k,  else PC←PC+1
语法:          操作码:          程序计数器:
BRVC k         -64≤k≤+63        PC←(PC+1)+k
                                     PC←PC+1
    
```

16 位操作码:

1111	01kk	kkkk	k011
------	------	------	------

状态寄存器 (SREG) 和布尔格式:

I	T	H	S	V	N	Z	C

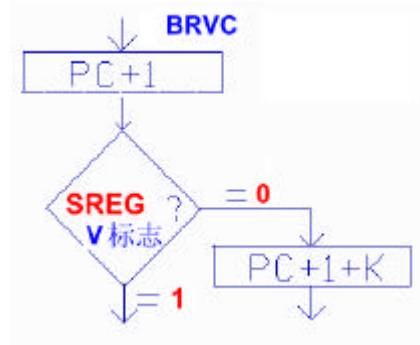
例子: (实践操作程序 45218.ASM)

```

lp:brvc lp1     ;溢出标志位(V)=0 转移,(V)=1 顺执
   clv         ;溢出标志位(V)置 0
   rjmp lp     ;反复测试
lp1:sev        ;溢出标志位(V)置 1
   rjmp lp     ;反复测试
    
```

```

Words:      1 (2 bytes)
Cyclse:     lif condition is false
            2 if conition is true
    
```



19. 中断标志触发转移

16 位操作码:

1111	01kk	kkkk	k111
------	------	------	------

状态寄存器 (SREG) 和布尔格式:

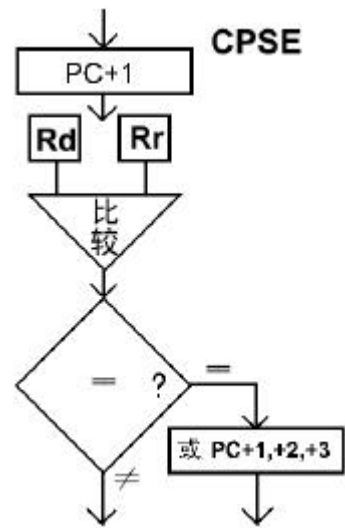
I	T	H	S	V	N	Z	C

例子: (实践操作程序 45220.ASM)

```

LP:BRID LP1 ;全局中断标志(I)=0 转移, (I)=1 顺执
  CLI      ;(I)=0
  RJMP LP  ;反复测试
LP1:SEI    ;(I)=1
  RJMP LP  ;反复测试
    
```

Words: 1 (2 bytes)
 Cycles: 1 if condition is false
 2 if condition is true



二、测试条件符合跳行转移指令

21. 相等跳行

CPSE—比较相等跳行

说明: 该指令完成两个寄存器 Rd 和 Rr 的比较, 若 Rd=Rr, 则跳行执行指令。

操作: If Rd=Rr then PC←PC+2 (or 3) else PC←PC+1

语法: 操作码: 程序计数器:
 CPSE Rd, Rr 0 ≤ d ≤ 31, 0 ≤ r ≤ 31 PC←PC+ 1
 PC←PC+ 2
 PC←PC+ 3

16 位操作码:

0001	00rd	dddd	rrrr
------	------	------	------

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 45221.ASM)

```

lp:inc    r4      ;设(r4)=$77 ,(r0)=$77
           ;设(r4)=$76 ,(r0)=$77
lp1:cpse r4,r0   ;r4 与 r0 比较相等跳行,不等顺执
lds r10,$0100   ;这是四字节指令把 SRAM 地址 $0100 的数据装入 R10
nop
dec r4          ;(r4)-1
    
```

```

cpse r4,r0      ; r4 与 r0 比较相等跳行,不等顺执
rjmp lp1       ;这是二字节指令
mov r10,r0     ;拷贝
rjmp lp       ;反复测试
    
```

Words: 1 (2 bytes)
Cycles: 1

22. 寄存器位清零跳行

SBRC 一寄存器位被清零跳行

说明: 该指令测试寄存器某位, 如果该位被清零, 则跳下一行执行指令。

操作: If Rd (b) = 0 then PC←PC+2 (or 3) eles PC←PC+ 1

语法: 操作码: 程序计数器:
SBRC Rr, b 0≤r≤31, 0≤b≤7 PC←PC+1
 pC←pC+2
 PC←PC+3

16 位操作码:

1111	110r	rrrr	Xbbb
------	------	------	------

状态寄存器 (SREG) 和布尔格式:

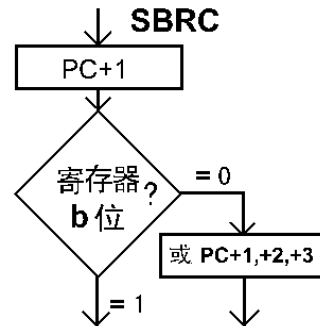
I	T	H	S	V	N	Z	C

例子: (实践操作程序 45222.ASM)

```

lp:sub r0,r1    ;设:(r0)=$80,(r1)=$70
sbrc r0,7      ;r0 的 7 位清零跳行,为 1 顺执
sub r0,r1     ;这是二字节指令
swap r0       ;r0 半字节交换
rjmp lp       ;反复测试
    
```

Words: 1 (2 bytes)
Cycles: 1 if condition is false(no skip)
 2 if condition is true (skip is exected)



23. 寄存器位置位跳行

SBRS 一寄存器位置位跳行

说明: 该指令测试寄存器某位, 如果该位被置位, 则跳下一行执行指令。

操作: If Rr(b) = 1 then PC←PC+ 2 (or 3) eles PC←PC+ 1

语法: 操作码: 程序计数器:
SBRS Rr, b ≤r≤31, 0≤b≤7 PC←PC+1
 PC←PC+2
 PC←PC+3

16 位操作码:

1111	111r	rrrr	Xbbb
------	------	------	------

rjmp lp

Words: 1 (2 bytes)

Cycles: 1 if condition is false (no skip)

2 if condition is true (skip is executed)

25. I/O 寄存器位置位跳行

SBIS—I/O 寄存器的位置位跳行

说明: 该指令测试 I/O 寄存器某位, 如果该位被置位, 则跳一行执行指令。该指令在低 32 个 I/O 寄存器内操作, 地址为 0~31。

操作: If I/OP, b=1 then PC←PC+2 (or 3) else PC←PC+1

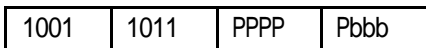
语法: 操作码: 程序计数器:

SBIS p ,b 0 ≤ P ≤ 31, 0 ≤ b ≤ 7 PC←PC+1

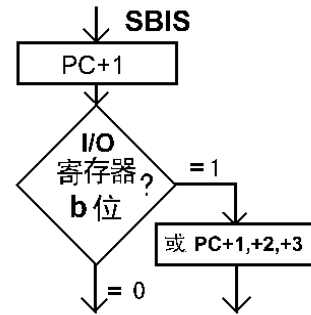
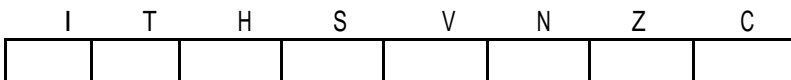
PC←PC+2

PC←PC+3

16 位操作码:



状态寄存器 (SREG) 和布尔格式:



例子: (实践操作程序 45225.ASM)

lp: ldi R16,\$0f ;

OUT \$1C,R16 ;置 i/o 寄存器 1 位为 1

lp1: sbis \$1c,1 ;复位后(EECR)=\$00, .equ EECR=\$1c,测试 1 位,为跳 1 行,为 0 顺执

;请打开 I/O 窗口观察,

;注意:这条指令只能连续测试 1 次,第 2 次会改变 \$1C 数据(反复测试不能用该指令),见 45225B.ASM

rjmp lp ;这是二字节指令

ldi R16,\$00 ;

OUT \$1C,R16 ;置 i/o 寄存器 1 位为 0

rjmp lp1

Words: 1 (2 bytes)

Cycles: 1 if condition is false (no skip)

2 if condition is true (skip is executed)

三、调用和返回指令

在程序设计中通常把具有一定功能模块的公用程序段定义为子程序。为了实观调用子程序的功能, 指令系统中都有调用指令。也称转移子程序指令, 它与转移指令的区别如下。执行调用子程序时, 把下一条指令地址 (PC 值) 保留到堆栈中, 即断点保护, 然后把子程序的起始地址置入 PC。子程序执行完毕, 再从断点返回, 从断点处继续执行原程序。而转移指令即不保护断点, 也不返回原程序。在每个子程序中都必须有返回指令, 返回指令的功能就是把调用前压入堆栈的断点弹出置入 PC, 恢复调用前的原程序。

在一个程序中, 子程序中还会调用别的子程序, 这称为子程序嵌套。每次调用子程序时必须将下条指令地址保存起来, 返回时按后进先出原则依次取出旧 PC 值。堆栈就是按后进先出规律存取数据的, **调用指令和返回指令**具有自动的保存和恢复 PC 内容的功能, 即**自动进栈, 自动出栈**。

26. 相对调用

RCALL—相对调用于程序

说明：在 PC+1 后（2K 字（4K 字节））范围内调用子程序。返回地址（RCALL 后的指令地址）存储到堆栈（见 CALL）。

操作： $PC \leftarrow (PC+1)+k$

语法： RCALL k 操作码： 程序计数器：
 $-2K \leq k \leq 2K$ $PC \leftarrow (PC+1)+k$

16 位操作码：

1101	kkkk	kkkk	kkkk
------	------	------	------

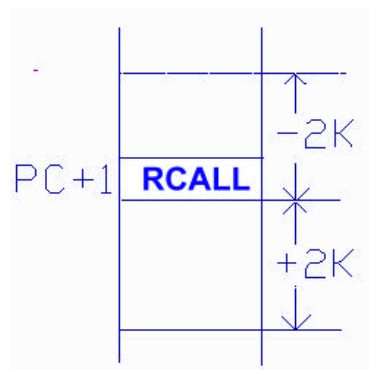
状态寄存器（SREG）和布尔格式：

I	T	H	S	V	N	Z	C

例子：（实践操作程序 DIP40LED.ASM）

```

ser temp          ;直接装入$FF,
out DDRA, temp   ;口的方向寄存器设定,为输出
forever:
clr temp          ;硬件设低电平 LED 灯亮
out PORTA, temp  ;PORTA 口 LED 灯亮
ldi temp,1200    ;装延时常数(十进制),灯亮延时,
rcall delay_mS   ;调用延时子程序
ser temp          ;硬件设高电平 LED 灯灭
out PORTA, temp  ;PORTA 口 LED 灯灭
ldi temp,1200    ;装延时常数,灯灭延时,可修改该参数
rcall delay_mS   ;调用延时子程序
rjmp forever     ;无限循环
    
```



```

delay_mS:         ;延时子程序
ldi zl,low(2500) ;装入延时参数,参数可修改
ldi zh,high(2500);参数可修改
delay_loop:
dec zl            ;-1
brne delay_loop ;不相等转移,相等按顺序执行
dec zh            ;-1
brne delay_loop ;不相等转移,相等按顺序执行
dec temp         ;-1
brne delay_mS    ;不相等转移,相等按顺序执行
ret              ;子程序返回
    
```

Words: 1 (2 bytes)
 Cycles: 3

27. 间接调用

ICALL—间接调用于程序

说明：间接调用由寄存器区中的 Z（16 位）指针寄存器指向的子程序。Z 指针寄存器

是 16 位宽, 允许调用当前程序存储空间 64K 字 (128 字节) 内的子程序。

操作: PC (15-0) ← Z (15-0)

PC (15-0) ← Z (15-0)

语法:

ICALL

操作码:

None

程序计数器:

See operation

16 位操作码:

1001	0101	XXXX	1001
------	------	------	------

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 45227.ASM)

.ORG \$0010

LP:MOV R30,R0 ;设(R0)为 Z 寄存器低 8 位地址,R30 为 Z 间接地址低 8 位

MOV R31,R1 ;设(R1)为 Z 寄存器高 8 位地址,R31 为 Z 间接地址高 8 位

ICALL ;根据 Z 寄存器的内容调用, 该步用进入子程序图标  调试

.ORG \$0200

LDI R20,\$11 ;地址\$0200 ,如设(R0)=\$00,(R1)=02,执行 ICALL 到此行

LDI R21,\$22 ;地址\$0201 ,如设(R0)=\$01,(R1)=02,执行 ICALL 到此行

LDI R22,\$33 ;地址\$0202 ,如设(R0)=\$02,(R1)=02,执行 ICALL 到此行

LDI R23,\$44 ;地址\$0203 ,如设(R0)=\$03,(R1)=02,执行 ICALL 到此行

LDI R24,\$55 ;地址\$0204 ,如设(R0)=\$04,(R1)=02,执行 ICALL 到此行

LDI R25,\$66 ;地址\$0205 ,如设(R0)=\$05,(R1)=02,执行 ICALL 到此行

ICALL ;地址\$0205 ,如设(R0)=\$06,(R1)=02,执行 ICALL 到此行

;又根据 Z 寄存器的内容调用到指定地址

move r30, r0

icall

Words: 1 (Zbytes)

Cycles: 3

28. 长调用

CALL—子程序长调用

说明: 在整个程序存储器区内调用子程序。返回地址 (调用后返回的指令地址) 将存储在堆栈 (见 RCALL 指令) 中。

操作: PC←k

PC←k

语法:

CALL k

操作码:

0 ≤ k ≤ 64K

程序计数器:

PC←k STACK←PC+ 2

SP←SP-2

CALL k

0 ≤ k ≤ 4M

PC←k STACK←PC+ 2

SP←SP-3

32 位操作码:


1001	010k	kkkk	111k
kkkk	kkkk	kkkk	kkkk

状态寄存器 (SRE) 和布尔格式:

I	T	H	S	V	N	Z	C

例子: (实践操作程序 45228.ASM)

```

move    r16,r0
call    delay_mS           ;长调用延时子程序,该步用进入子程序图标  调试
.ORG $0200
delay_mS:                   ;延时子程序
    ldi z1,low(1990)
    ldi zh,high(1990)
delay_loop:
    dec z1
    brne delay_loop
    dec zh
    brne delay_loop
    dec temp
    brne delay_mS
    ret                     ;子程序返回

Words:   2 (4 bytes)
Cycles:  4
    
```

29. 从子程序返回

RET—子程序返回

说明: 从子程序返回。返回地址从堆栈中弹出。

操作: PC (15—0) ← STACK

PC (21—0) ← STACK

语法:	操作码:	程序计数器:	堆栈:
RET	None	See Operation	SP←SP+ 2
RET	None	See Operation	SP←SP+ 3

16 位操作码:

1001	0101	0XX0	1000
------	------	------	------

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 45228.ASM)

