

4.7 位指令和位测试指令

AVR 单片机指令系统中有四分之一的指令为位和位测试指令。位指令的灵活应用，极大地提高了系统的逻辑控制和处理能力。

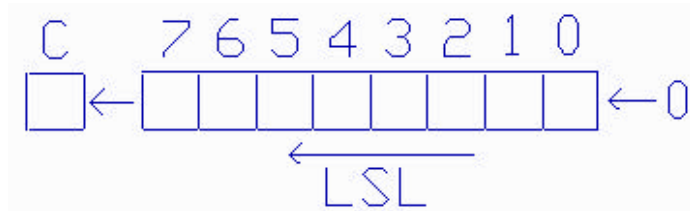
4.7.1 带进位逻辑操作指令

1. 逻辑左移

LSL—逻辑左移

说明：寄存器 Rd 中所有位左移 1 位。第 0 位被清零，第 7 位移到 SREG 中的 C 标志。该指令完成一个无符号数乘 2 的操作。

操作：



语法：

LSL Rd

操作码：

$0 \leq d \leq 31$

程序计数器：

$PC \leftarrow PC + 1$

16 位操作码：

0000	11dd	dddd	dddd
------	------	------	------

P125

状态寄存器 (SREG) 和布尔格式：

I	T	H	S	V	N	Z	C
-	-	↔	↔	↔	↔	↔	↔

H: Rd3

N: R7

S: NV

Z: /R7·/R6·/RS·/R4·/R3·/R2·/R1·/R0

V: NC

C: Rd7

例子：(实践操作程序 4711.ASM)

LP:

ldi r16,\$01 ;将 01 送入到 R16 中

ldi r17,\$02 ;将 02 送入到 R17 中

add r16,r17 ;R16 与 R17 相加

lsl r16 ;加后再左移一位 (01+02=03, 左一位, 即乘以 2。所以 r=06)

rjmp lp ;继续实验

Words: 1 (2 bytes)

Cycles: 1

2. 逻辑右移

LSR—逻辑右移

说明：寄存器 Rd 中所有位右移 1 位。第 7 位被清零，第 0 位移到 SREG 中的 C 标志。该指令完成一个无符号数除 2 的操作。C 标志被用于结果舍入。



操作：

语法：

LSR Rd

16 位操作码：

操作码：

$0 \leq d \leq 31$

程序计数器：

$PC \leftarrow PC + 1$

1001	010d	dddd	0110
------	------	------	------

状态寄存器 (SREG) 和布尔格式：

I	T	H	S	V	N	Z	C
-	-		↔	↔	↔	↔	↔

S: N V

V: NC

N: 0

例子：(实践操作程序 4712.ASM)

LP:

add r0,r1 ;将 R0 与 R1 的内容相加 (r0)= , (r1)=

lsl r0 ;将 R0 的内容右转移一位

rjmp lp ;循环继续做

Words: 1 (2 bytes)

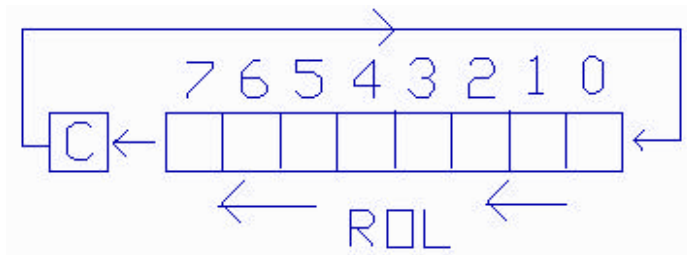
Cycles: 1

3. 通过进位左循环

ROL—通过进位左循环

说明：寄存器 Rd 的所有位左移 1 位，C 标志被移到 Rd 的第 0 位，Rd 的第 7 位移到 C 标志。

操作：



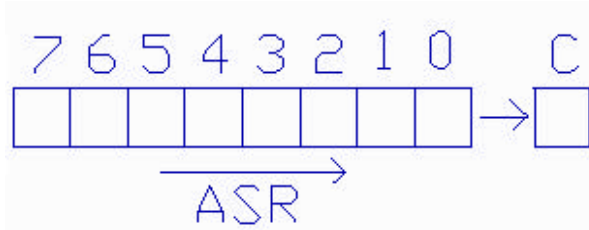
-	-		↔	↔	↔	↔	↔
---	---	--	---	---	---	---	---

S: N V Z: /R7·/R6·..R·/R4·/R3·/R2·/R1·R0
 V: N C C: Rd0
 N: R7

例子: (实践操作程序 4714.ASM)

```
LP:
    ror r15      ;将 R15 中的内容通过进位循环右移,设:(r15)=
    ror r15
    nop         ;
    rjmp lp     ;
```

Words: 1 (2 bytes)
 Cycles: 1



5. 算术右移

ASR—算术右移

说明: 寄存器 Rd 中的所有位右移 1 位, 而位 7 保持常量, 位 0 被装入 SREG 的 C 标志位。这个操作实现 2 的补码值除 2, 而不改变符号, 进位标志用于结果的舍入。

操作:

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

1001	010d	dddd	0101
------	------	------	------

状态寄存器 (SREG) 和布尔格式:

	I	T	H	S	V	N	Z	C
-	-		↔	↔	↔	↔	↔	↔

S: N V Z: R7·R6·R5·R4·R3·R2·R1·R0
 V: N C C: Rd0
 N: R7

例子: (实践操作程序 4715.ASM)

```
LP:
    ldi r16,$10 ;
    ldi r17,$fc ;
    asr r16     ;
    asr r17     ;
    rjmp lp    ;
```

words: 1 (2 bytes)
 Cycles: 1

6. 半字节交换

SWAP—半字节交换

说明：寄存器中的高半字节和低半字节交换。

操作：R (7~4) ←Rd (3~0) ←R (3~0) ←Rd (7~4)

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

16 位操作码：

1001	010d	dddd	0010
------	------	------	------

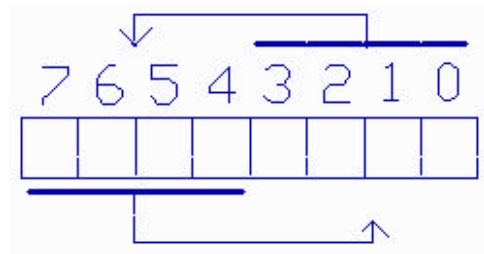
状态寄存器 (SREG) 和布尔格式：

I	T	H	S	V	N	Z	C

例子：(实践操作程序 4716.ASM)

```

LP:
    inc r1    ;
    inc r2    ;
    swap r1   ;
    swap r2   ;
    rjmp lp   ;
    
```



Words: 1 (2 bytes)
 Cycles: 1

4.7.2 位变量传送指令

1. 寄存器中的位存储到 SREG 中的 T 标志

BST—寄存器中的位存储到 SREG 中的 T 标志

说明：把寄存器中的位 b 存储到 SREG (状态寄存器) 中的 T 标志。

操作：T ← Rd (b)

语法： BST Rd b 操作码： 程序计数器：
 0 ≤ d ≤ 31, 0 ≤ b ≤ 7 PC ← PC + 1

16 位操作码：

1111	101d	dddd	Xbbb
------	------	------	------

状态寄存器 (SREG) 和布尔格式：

I	T	H	S	V	N	Z	C

T: 0 if bit b in Rd is cleared. Set to 1 otherwise.

例子:

```
bst r1,2 ; T←R1 (2)
bld r0,4 ; R0 (4) ←T
```

Words: 1 (2 bytes)

Cycles: 1

2. SREG 中的 T 标志装入寄存器中的某一位

BLD 一位装入，将 SREG 中的 T 标志装入到寄存器中的某一位。

说明：拷贝 SREG（状态寄存器）的 T 标志到寄存器 Rd 中的位 b。

操作：Rd (b) ←T

语法:

操作码:

程序计数器:

BLD Rd,d 0 ≤ d ≤ 31, 0 ≤ b ≤ 7 PC ← PC + 1

16 位操作码:

1111	100d	dddd	0bbb
------	------	------	------

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 4722.ASM)

```
bst r1,2 ; T←R1 (2)
bld r0,4 ; R0 (4) ←T
```

Words: 1 (2 bytes)

Cycles: 1

4.7.3 位变量修改指令

1. 置状态寄存器的位

BSET—置状态寄存器的位

说明：置状态寄存器 (SREG) 的某一标志或某一位。

操作：SREG (S) ←1

语法:

操作码:

程序计数器:

BSET s 0 ≤ s ≤ 7

PC ← PC + 1

16 位操作码:

1001	0100	0sss	1000
------	------	------	------

状态寄存器 (SREG) 和布尔格式:

I	T	H	S	V	N	Z	C
↔	↔	↔	↔	↔	↔	↔	↔

I: 1 if s=7 V: 1 if s=3
 T: 1 if s=6 N: 1 if s=3
 H: 1 if s=5 Z: 1 if s=1
 S: 1 if s=4 C: 1 if s=0

例子:

```

    bset 6 ; SREG (6) ← 1
    bset 7 ; SREG (7) ← 1
    
```

Words: 1 (2 bytes)
 Cycles: 1

2. 清状态寄存器的位

BCLR——SREG 中的位清零

说明: 清零 SREG 状态寄存器中的一个标志位。

操作: SREG (S) ← 0

语法:

操作码:

程序计数器:

BCLR s 0 ≤ s ≤ 7

PC ← PC + 1

16 位操作码:

1001	0100	1sss	1000
------	------	------	------

状态寄存器 (SREG) 和布尔格式:

I	T	H	S	V	N	Z	C
↔	↔	↔	↔	↔	↔	↔	↔

I: 0 if s=7 V: 0 if s=3
 T: 0 if s=6 N: 0 if s=2
 H: 0 if s=5 Z: 0 if s=1
 S: 0 if s=4 C: 0 if s=0

例子:

```

    bclr 0 ; SREG (0) ← 0
    bclr 7 ; SREG (7) ← 0
    
```

Words: 1 (2 bytes)
 Cycles: 1

3. 置 I/O 寄存器的位

SBI——置 I/O 寄存器的位

说明: 对 I/O 寄存器指定的位置位, 该指令在低 32 个 I/O 寄存器内操作, I/O 寄存器地址为 0~31。

操作: I/O (P b) ← 1

语法:

操作码:

程序计数器:

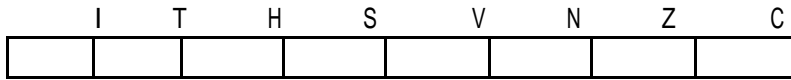
SBIP, b 0 ≤ P ≤ 31, 0 ≤ b ≤ 7

PC ← PC + 1

16 位操作码:

1001	1010	PPPP	Pbbb
------	------	------	------

状态寄存器 (SREG) 和布尔格式:



例子:

```
out $1e,r0 ;(EEARL 寄存器)←(R0)
sbi $1c,0 ;( EECR 寄存器 0 位)←1
in r1,$1d ;(R0)←(EEDR 寄存器)
```

WordS: 1 (2 bytes)

Cycles: 2

4. 清 I/O 寄存器的位

CBI—清 I/O 寄存器的位

说明: 清零 I/O 寄存器中的指定位, 该指令用在寄存器最低的 32 个 I/O 寄存器上, I/O 寄存器地址为 0~31。

操作: I/O (P, b) ← 0

语法:

操作码:

程序计数器:

CBI P, b 0 ≤ P ≤ 31, 0 ≤ b ≤ 7

PC ← PC + 1

16 位操作码:



状态寄存器 (SREG) 和布尔格式:



C: 1

例子:

```
cbi $18,7 ; I/O (PORTB 寄存器的 7 位) ← 0
```

Words: 1 (2 bytes)

Cycles: 2

5. 置进位位

SEC — 置位进位标志

说明: 置位 SREG (状态寄存器) 中的进位标志 (C)。

操作: C ← 1

语法:

操作码:

程序计数器:

SEC

None

PC ← PC + 1

16 位操作码:



状态寄存器 (SREG) 和布尔格式:



							1
--	--	--	--	--	--	--	---

C: 1

例子:

```

sec          ; C←1
adc r0, r1  ;带进位位加
words:      1 ( 2 bytes)
Cycles:     1

```

6. 清进位位

CLC 清零进位标志

说明: 清零 SREG (状态寄存器) 中的进位标志 (C)。

操作: $C \leftarrow 0$

语法:	操作码:	程序计数器:
CLC	None	$PC \leftarrow PC + 1$

16 位操作码:

1001	0100	1000	1000
------	------	------	------

状态寄存器 (SREG) 和布尔格式:

I	T	H	S	V	N	Z	C
							0

C: 0

例子:

```

add r0, r0 ;加
clc       ; C←0
Words:    1 ( 2 bytes)
Cycles:   1

```

7. 置位负标志位

SEN—置位负数标志

说明: 置位 SREG (状态寄存器) 中的负数标志 (N)。

操作: $N \leftarrow 1$

语法	操作码:	程序计数器:
SEN	None	$PC \leftarrow PC + 1$

16 位操作码:

1001	0100	0010	1000
------	------	------	------

状态寄存器 (SREG) 和布尔格式:

I	T	H	S	V	N	Z	C
					1		

`set` ; $I \leftarrow 1$
 Words: 1 (2 bytes)
 Cycles: 1

12. 禁止全局中断位

CLI—清零全局中断标志

说明: 清除 SREG (状态寄存器) 中的全局中断标志 (I)。

操作: $I \leftarrow 0$

语法: 操作码: 程序计数器:
 CLI None $PC \leftarrow PC + 1$

16 位操作码:

1001	0100	1111	1000
------	------	------	------

状态寄存器 (SREG) 和布尔格式:

I	T	H	S	V	N	Z	C
0							

I: 0

例子: (实践操作程序 47312.ASM)

```

cli          ; I ← 0
in r13,$16  ; (r13) ← (PINB 寄存器数据)
set         ; I ← 1
  
```

Words: 1 (2 bytes)

Cycles: 1

13. 置 S 标志位

SES—置位符号标志

说明: 置位 SREG (状态寄存器) 中的符号标志 (S)。

操作: $s \leftarrow 1$

语法: 操作码: 程序计数器:
 SES None $PC \leftarrow PC + 1$

16 位操作码:

1001	0100	0100	1000
------	------	------	------

状态寄存器 (SREG) 和布尔格式:

I	T	H	S	V	N	Z	C
			1				

S: 1

例子: (实践操作程序 47313.ASM)

```

add r2, r19 ;加
ses          ; s ← 1
  
```

Words: 1 (2 bytes)

Cycles: 1

操作: $V \leftarrow 0$

语法:

CLV

16 位操作码:

操作码:

None

程序计数器:

 $PC \leftarrow PC + 1$

1001	0100	1011	1000
------	------	------	------

状态寄存器 (SREG) 和布尔格式:

I	T	H	S	V	N	Z	C
				0			

V: 0

例子:

```

add    r2, r3 ;加
      clv    ; V←0

```

Words: 1 (2 bytes)

Cycles: 1

17. 置 T 标志位

SET 置位 T 标志

说明: 置位 SREG (状态寄存器) 中的 T 标志。

操作: $T \leftarrow 1$

语法:

SET

16 位操作码:

操作码:

None

程序计数器:

 $PC \leftarrow PC + 1$

1001	0100	0110	1000
------	------	------	------

状态寄存器 (SREG) 和布尔格式:

I	T	H	S	V	N	Z	C
	1						

T: 1

例子:

```

set    ; T←1

```

Words: 1 (2 bytes)

Cycles: 1

18. 清 T 标志位

CLT—清零 T 标志

说明: 清零 SREG (状态寄存器) 中的 T 标志。

操作: $T \leftarrow 0$

语法:

CLT

16 位操作码:

操作码:

None

程序计数器:

 $PC \leftarrow PC + 1$

1001	0100	1110	1000
------	------	------	------

状态寄存器 (SREG) 和布尔格式:

I	T	H	S	V	N	Z	C
	0						

T: 0

例子:

`clt` ; T←0

Words: 1 (2 bytes)

Cycles: 1

19. 置半进位标志

SEH—置位半进位标志

说明: 置位 SREG (状态寄存器) 中的半进位标志 (H)。

操作: H←1

语法:

操作码:

程序计数器:

SEH

None

PC←PC+ 1

16 位操作码:

1001	0100	0101	1000
------	------	------	------

状态寄存器 (SREG) 和布尔格式:

I	T	H	S	V	N	Z	C
		1					

H: 1

例子:

`seh` ; H←1

Words: 1 (2 bytes)

Cycles: 1

20. 清半进位标志

CLH—清零半进位标志

说明: 清零 SREG (状态寄存器) 中的半进位标志 (H)。

操作: H←0

语法:

操作码:

程序计数器:

CLH

None

PC←PC+ 1

16 位操作码:

1001	0100	1101	1000
------	------	------	------

状态寄存器 (SREG) 和布尔格式:

I	T	H	S	V	N	Z	C
---	---	---	---	---	---	---	---

		0					
--	--	---	--	--	--	--	--

H: 0

例子:

 clh ; H←0

Words: 1 (2 bytes)

Cycles: 1

4.7.4 其它指令

1. 空指令

NOP—空操作

说明: 该指令完成一个单周期空操作。

应用: 延时等待;产生方波;抗干扰,在无程序单元写上空操作,空操作指令最后转到\$000H

操作: No

语法: 操作码: 程序计数器:

NOP None PC←PC+ 1

16 位操作码:

0000	0000	0000	0000
------	------	------	------

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 4741.ASM)

LP:

nop ;空操作

add r0 ,r1 ;R1 与 R0 相加

nop ;空操作

add r0, r1 ;R1 与 R0 相加

rjmp lp ;循环 P138

Words: 1 (2 bytes)

Cycles: 1

2. 休眠指令

SLEEP—休眠

说明: 该指令设置电路休眠模式,由 MCU 控制寄存器定义。当在休眠状态由一个中断唤醒时,在中断程序执行后,紧跟在休眠指令后的指令被执行。

应用:省电,尤其对便携式仪器特别有用

操作:

语法: 操作码: 程序计数器:

SLEEp None PC~PC+ 1

16 位操作码:

1001	0101	100x	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 4742.ASM)

```

mov ro, r11 ;拷贝
sleep      ;休眠
words:    1 ( 2 bytes)
Cycles:   1

```

3. 看门狗复位

WDR—看门狗复位

说明: 该指令复位看门狗定时器, 在 WD 预定比例器给出限定时间内必须执行。参见看门狗定时器硬件部分。

应用: 抗干扰; 延时

操作: WD timer restart.

语法: 操作码:

WDR None

16 位操作码:

程序计数器:

PC~PC+ 1

1001	0101	101x	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 4743.ASM)

```

PLYDEL: LDI TEMP,185 ;用 WDR 做延时子程序
DT3:    LDI TEMP1,04 ;
DT2:    LDI TEMP2,250
DT1:    WDR ;1T
        WDR ;2T
        WDR ;3T
        WDR ;4T
        WDR ;5T
        DEC TEMP2 ;
        BRNE DT1 ;
        DEC TEMP1 ;
        BRNE DT2 ;
        DEC TEMP ;
        BRNE DT3 ;
        RET
Words:  1 ( 2 bytes)
Cycles:  1

```