《AVR单片机C语言开发入门与典型实例》（修订版）

作者：华清远见

第3章 ATmega128（L）的指令系统

本章目标

本章主要介绍 ATmega128（L）的指令系统，首先分析 ATmega128（L）的指令的特点、分类和寻址方式，并详细介绍了指令系统的每条指令，最后介绍了ATmega128（L）汇编语言的一些基本知识。本章的主要内容有以下几个方面。

- 指令系统的特点、分类和寻址方式
- 分类介绍指令系统的每条指令
- 汇编语言的介绍

3.1 ATmega128（L）的指令系统概述

对于 ATmega128（L）来说，它不仅具备比较完善和功能强大的硬件结构和组成，如 SPI、USART、TWI、ADC 和模拟比较器等，而且它的指令系统采用先进的 RISC 体系结构，从而使 ATmega128（L）的整体表现更加出色，可以应用在很多场合。

ATmega128（L）的指令系统是 RISC 结构的精简指令集，完全兼容 AVR 单片机的指令系统，具有高性能的数据处理能力，可以对位、半字节、字节和双字节数据进行各种操作，包括算术和逻辑运算、数据传输、布尔运算、控制转移和硬件乘法等操作。

ATmega128（L）的指令系统中共有 133 条指令，按功能可以分为以下 5 大类。

- 算术和逻辑运算指令（28 条），如表 3.1 所示；
- 比较和跳转指令（36 条），如表 3.2 所示；
- 数据传输指令（38 条），如表 3.3 所示；
- 位操作和位测试指令（28 条），如表 3.4 所示；
- MCU 控制指令（3 条），如表 3.5 所示。

表 3.1　　Tmega128（L）指令系统中的算术和逻辑运算指令（28 条）

助　记　符	操　作　数	说　　明	操　　作	受影响的标志位
ADD	Rd，Rr	两个寄存器相加	Rd←Rd+Rr	Z、C、N、V 和 H
ADC	Rd，Rr	两个寄存器带进位相加	Rd←Rd+Rr+C	Z、C、N、V 和 H
ADIW	Rdl，K	立即数与字相加	Rdh:Rdl←Rdh+Rdl+K	Z、C、N、V 和 S
SUB	Rd，Rr	两个寄存器相减	Rd←Rd−Rr	Z、C、N、V 和 H
SUBI	Rd，K	寄存器与常数相减	Rd←Rd−K	Z、C、N、V 和 H
SBC	Rd，Rr	两个寄存器带进位相减	Rd←Rd−Rr−C	Z、C、N、V 和 H
SBCI	Rd，K	寄存器带常数及进位相减	Rd←Rd−K−C	Z、C、N、V 和 H
SBIW	Rdl，K	字与立即数相减	Rdh:Rdl←Rdh+Rdl−K	Z、C、N、V 和 S
AND	Rd，Rr	两个寄存器逻辑与	Rd←Rd•Rr	Z、N、V 和 S
ANDI	Rd，K	寄存器与常数逻辑与	Rd←Rd•K	Z、N、V 和 S
OR	Rd，Rr	两个寄存器逻辑或	Rd←Rd v Rr	Z、N、V 和 S
ORI	Rd，K	寄存器与常数逻辑或	Rd←Rd v K	Z、N、V 和 S
EOR	Rd，Rr	两个寄存器异或	Rd←Rd ⊕Rr	Z、N、V 和 S
COM	Rd	取反	Rd←$FF−Rd	Z、C、N、V 和 S
NEG	Rd	取补	Rd←$00−Rd	I 和 T 外的所有位
SBR	Rd，K	置位寄存器的某些位	Rd←Rd v K	Z、N、V 和 S
CBR	Rd，K	清零寄存器的某些位	Rd←Rd•($FF−K)	Z、N、V 和 S
INC	Rd	加 1	Rd←Rd+1	Z、N、V 和 S
DEC	Rd	减 1	Rd←Rd−1	Z、N、V 和 S
TST	Rd	测试寄存器值为 0 或负	Rd←Rd•Rd	Z、N、V 和 S
CLR	Rd	清零寄存器	Rd←Rd ⊕Rd	Z、N、V 和 S
SER	Rd	置位寄存器	Rd←$FF	无
MUL	Rd，Rr	无符号数相乘	R1:R0←Rd x Rr	Z 和 C
MULS	Rd，Rr	有符号数相乘	R1:R0←Rd x Rr	Z 和 C
MULSU	Rd，Rr	有符号数与无符号数相乘	R1:R0←Rd x Rr	Z 和 C
FMUL	Rd，Rr	无符号小数相乘	R1:R0←(Rd x Rr)<<1	Z 和 C

FMULS	Rd，Rr	有符号小数相乘	R1:R0←(Rd x Rr)<<1	Z 和 C
FMULSU	Rd，Rr	有符号与无符号小数相乘	R1:R0←(Rd x Rr)<<1	Z 和 C

表 3.2　　ATmega128（L）指令系统中的比较和跳转指令（36 条）

助记符	操作数	说　　明	操　　作	受影响的标志位
RJMP	k	相对跳转	PC←PC+k+1	无
IJMP	无	间接跳转到（Z）	PC←Z	无
JMP	k	直接跳转	PC←k	无
RCALL	k	相对子程序调用	PC←PC+k+1	无
ICALL	无	相对调用到（Z）	PC←Z	无
CALL	k	直接子程序调用	PC←k	无
RET	无	子程序返回	PC←STACK	无
RETI	无	中断返回	PC←STACK	I
CPSE	Rd，Rr	比较，相等则跳过下一条指令	if(Rd=Rr) PC←PC+2or3	无
CP	Rd，Rr	比较	Rd−Rr	Z、N、V、C 和 H
CPC	Rd，Rr	带进位位比较	Rd−Rr−C	Z、N、V、C 和 H
CPI	Rd，K	比较寄存器与立即数	Rd−K	Z、N、V、C 和 H
SBRC	Rr，b	寄存器的某位清 0 即跳转	if(Rr(b)=0) PC←PC+2 or 3	无
SBRS	Rr，b	寄存器的某位置位即跳转	if(Rr(b)=1) PC←PC+2 or 3	无
SBIC	P，b	I/O 寄存器的某位清 0 即跳转	if(P(b)=0) PC←PC+2 or 3	无
SBIS	P，b	I/O 寄存器的某位置位即跳转	if(P(b)=1) PC←PC+2 or 3	无
BRBS	s，k	状态标志置位即跳转	if(SREG(s)=1) then PC←PC+k+1	无
BRBC	s，k	状态标志清 0 即跳转	if(SREG(s)=0) then PC←PC+k+1	无
BREQ	k	相等即跳转	if(Z=1) then PC←PC+k+1	无
BRNE	k	不相等即跳转	if(Z=0) then PC←PC+k+1	无
BRCS	k	进位标志置位即跳转	if(C=1) then PC←PC+k +1	无
BRCC	k	进位标志清 0 即跳转	if(C=0) then PC←PC+k+1	无
BRSH	k	相同或更大即跳转	if(C=0) then PC←PC+k+1	无
BRLO	k	小于即跳转	if(C=1) then PC←PC+k+1	无
BRMI	k	负数即跳转	if(N=1) then PC←PC+k+1	无
BRPL	k	正数即跳转	if(N=0) then PC←PC+k+1	无
BRGE	k	有符号数大于/等于即跳转	if(N ⊕V=0) then PC←PC+k+1	无
BRLT	k	有符号数小于 0 即跳转	if(N ⊕V=1) then PC←PC+k+1	无
BRHS	k	半进位标志置位即跳转	if(H=1) then PC←PC+k+1	无
BRHC	k	半进位标志清 0 即跳转	if(H=0) then PC←PC+k+1	无
BRTS	k	T 标志置位即跳转	if(T=1) then PC←PC+k+1	无
BRTC	k	T 标志清 0 即跳转	if(T=0) then PC←PC+k+1	无
BRVS	k	溢出标志置位即跳转	if(V=1) then PC←PC+k+1	无
BRVC	k	溢出标志清 0 即跳转	if(V=0) then PC←PC+k+1	无
BRIE	k	中断使能即跳转	if(I=1) then PC←PC+k+1	无
BRID	k	中断禁止即跳转	if(I=0) then PC←PC+k+1	无

表 3.3　　ATmega128（L）指令系统中的数据传输指令（38 条）

助记符	操作数	说　　明	操　　作	受影响的标志位
MOV	Rd，Rr	寄存器之间数据转移	Rd←Rr	无

MOVW	Rd，Rr	寄存器字传输	Rd+1:Rd←Rr+1:Rr	无
LDI	Rd，K	加载立即数	Rd←K	无
LD	Rd，X	间接加载	Rd←(X)	无
LD	Rd，X+	间接加载并执行后加操作	Rd←(X)，X←X+1	无
LD	Rd，−X	间接加载并执行预减操作	X←X−1，Rd←(X)	无
LD	Rd，Y	间接加载	Rd←(Y)	无
LD	Rd，Y+	间接加载并执行后加操作	Rd←(Y)，Y←Y+1	无
LD	Rd，−Y	间接加载并执行预减操作	Y←Y−1，Rd←(Y)	无
LDD	Rd，Y+q	带偏移量的间接加载	Rd←(Y+q)	无
LD	Rd，Z	间接加载	Rd←(Z)	无
LD	Rd，Z+	间接加载并执行后加操作	Rd←(Z)，Z←Z+1	无
LD	Rd，−Z	间接加载并执行预减操作	Z←Z−1，Rd←(Z)	无
LDD	Rd，Z+q	带偏移量的间接加载	Rd←(Z+q)	无
LDS	Rd，k	从 SRAM 中直接加载	Rd←(k)	无
ST	X，Rr	间接存储	(X)←Rr	无
ST	X+，Rr	间接存储并执行后加操作	(X)←Rr，X←X+1	无
ST	−X，Rr	间接存储并执行预减操作	X←X−1，(X)←Rr	无
ST	Y，Rr	间接存储	(Y)←Rr	无
ST	Y+，Rr	间接存储并执行后加操作	(Y)←Rr，Y←Y+1	无
ST	−Y，Rr	间接存储并执行预减操作	Y←Y−1，(Y)←Rr	无
STD	Y+q，Rr	带偏移量的间接存储	(Y+q)←Rr	无
ST	Z，Rr	间接存储	(Z)←Rr	无
ST	Z+，Rr	间接存储并执行后加操作	(Z)←Rr，Z←Z+1	无
ST	−Z，Rr	间接存储并执行预减操作	Z←Z−1，(Z)←Rr	无
STD	Z+q，Rr	带偏移量的间接存储	(Z+q)←Rr	无
STS	k，Rr	直接存储到 SRAM	(k)←Rr	无
LPM	无	加载程序存储器	R0←(Z)	无
LPM	Rd，Z	加载程序存储器	Rd←(Z)	无
LPM	Rd，Z+	加载程序存储器并执行后加操作	Rd←(Z)，Z←Z+1	无
ELPM	无	扩展的加载程序存储器操作	R0←(RAMPZ:Z)	无
ELPM	Rd，Z	扩展的加载程序存储器操作	Rd←(RAMPZ:Z)	无
ELPM	Rd，Z+	扩展的加载程序存储器操作并执行后加操作	Rd←(RAMPZ:Z)，RAMPZ:Z←RAMPZ:Z+1	无
SPM	无	存储程序存储器	(Z)←R1:R0	无
IN	Rd，P	读入端口数据	Rd←P	无
OUT	P，Rr	将数据输出到端口	P←Rr	无
PUSH	Rr	将寄存器的数值推入堆栈	STACK←Rr	无
POP	Rd	将寄存器的数值弹出堆栈	Rd←STACK	无

表 3.4　　ATmega128（L）指令系统中的位操作和位测试指令（28 条）

助记符	操作数	说　　明	操　　作	受影响的标志位
SBI	P，b	置位 I/O 寄存器的某一位	I/O(P,b)←1	无
CBI	P，b	清零 I/O 寄存器的某一位	I/O(P,b)←0	无
LSL	Rd	逻辑左移	Rd(n+1)←Rd(n)，Rd(0)←0	Z、C、N 和 V
LSR	Rd	逻辑右移	Rd(n)←Rd(n+1)，Rd(7)←0	Z、C、N 和 V
ROL	Rd	通过进位位向左循环	Rd(0)←C，Rd(n+1)←Rd(n)，	Z、C、N 和 V

			C←Rd(7)	
ROR	Rd	通过进位位向右循环	Rd(7)←C，Rd(n)←Rd(n+1)， C←Rd(0)	Z、C、N 和 V
ASR	Rd	算术右移	Rd(n)←Rd(n+1)，n=0～6	Z、C、N 和 V
SWAP	Rd	高 4 位与低 4 位交换	Rd(3～0)←Rd(7～4)，Rd(7～4)←Rd(3～0)	无
BSET	s	置位标志位	SREG(s)←1	SREG(s)
BCLR	s	清 0 标志位	SREG(s)←0	SREG(s)
BST	Rr，b	将寄存器的某一位保存到 T	T←Rr(b)	T
BLD	Rd，b	将 T 加载到寄存器的某一位	Rd(b)←T	无
SEC	无	置位进位位	C←1	C
CLC	无	清 0 进位位	C←0	C
SEN	无	置位负数标志位	N←1	N
CLN	无	清 0 负数标志位	N←0	N
SEZ	无	置位 0 标志位	Z←1	Z
CLZ	无	清 0 零标志位	Z←0	Z
SEI	无	全局中断标志位置位	I←1	I
CLI	无	全局中断标志位清 0	I←0	I
SES	无	置位符号测试标志位	S←1	S
CLS	无	清 0 符号测试标志位	S←0	S
SEV	无	置位 2 的补码溢出标志	V←1	V
CLV	无	清除 2 的补码溢出标志	V←0	V
SET	无	置位 SREG 的 T 标志	T←1	T
CLT	无	清除 SREG 的 T 标志	T←0	T
SEH	无	置位 SREG 的半进位标志	H←1	H
CLH	无	清除 SREG 的半进位标志	H←0	H

表 3.5　　ATmega128（L）指令系统中的 MCU 控制指令（3 条）

助　记　符	操　作　数	说　　明	操　　作	受影响的标志位
NOP	无	无操作	无	无
SLEEP	无	休眠	参见有关休眠的说明	无
WDR	无	看门狗复位	参见有关看门狗定时器的说明	无
BREAK	无	Break	只用于片上调试	无

表 3.1～3.5 中给出了 ATmega128（L）指令系统的所有指令助记符、操作数及相应的操作，下面对其中所使用的符号的意义进行简单地说明。

（1）状态寄存器及其中的标志位说明如下。

- SREG：状态寄存器；
- C：进位标志位，表示一个算术或逻辑操作之后有无产生进位；
- Z：零标志位，表示一个算术或逻辑操作之后的结果是否为 0；
- N：负数标志位，表示一个算术或逻辑操作之后的结果是否为负数；
- V：二进制的补码溢出标志位，用来支持二进制的补码运算；
- S：符号标志位，由式子“S=N⊕V”决定，即负数标志位 N 和二进制的补码溢出标志位 V 两者的异或值；
- H：半进位标志位，指示了一些 ALU 操作过程中有无半进位的出现；
- T：位拷贝存储位，通常出现在位拷贝指令 BLD 和 BST 中作为目的或源地址；
- I：全局中断使能位。

（2）寄存器和操作码的说明如下。

- Rd：目的（或源）寄存器，取值为 R0～R31 或 R16～R31（取决于具体的指令）；
- Rdl：寄存器 R24、R26、R28 或 R30，通常用于指令 ADIW 和 SBIW 中；
- Rr：源寄存器，取值为 R0-R31；
- b：寄存器中的指定位，取值为常数 0～7；
- s：状态寄存器 SREG 中的指定位，取值为常数 0～7；
- P：I/O 寄存器，取值为 0～31 或 0～63（取决于具体的指令）；
- K：立即数，取值为 0～255；
- k：地址常数，取值取决于具体的指令；
- q：地址偏移量常数，取值为 0～63；
- X、Y 和 Z：地址指针寄存器（X=R27:26；Y=R29:28；Z=R31:R30）。

（3）堆栈的说明如下。

- STACK：堆栈；
- SP：堆栈 STACK 的指针。

3.2 ATmega128（L）的指令操作数的寻址方式

寻址方式指的是指令给出参与运算的数据（即操作数）的方式。在 ATmega128（L）中，指令操作数的寻址方式有：寄存器直接寻址、数据存储器空间直接寻址、数据存储器空间间接寻址、程序存储器空间取常量寻址、程序存储器空间直接寻址、程序存储器空间间接寻址和程序存储器空间相对寻址。具体介绍如下。

- 单寄存器直接寻址方式指的是由指令指定的一个寄存器的内容作为操作数，并在指令中给出该寄存器的直接地址。这种寻址方式的寻址范围为通用工作寄存器组中的 32 个寄存器 R0～R31 或后 16 个寄存器 R16～R31。例如：“TST　Rd”。
- 双寄存器直接寻址方式指的是由指令指定的两个寄存器 Rd 和 Rr 的内容作为操作数，并在指令中给出这两个寄存器的直接地址，最后指令执行的结果存放在寄存器 Rd 中这种寻址方式的寻址范围为通用工作寄存器组中的 32 个寄存器 R0～R31 或后 16 个寄存器 R16～R31 或寄存器 R16～R23。例如：“ADD　Rd，Rr”。
- I/O 寄存器直接寻址方式指的是由指令指定的一个寄存器的内容作为操作数，并在指令中给出该寄存器的直接地址。这种寻址方式的寻址范围为 0～63 或 0～31，即 I/O 寄存器空间的地址$00～$3F。例如：“IN Rd，P”。
- 数据存储器空间直接寻址指的是直接给出 SRAM 中操作数的地址。这种寻址方式的指令为双字节指令，在指令的低字节中给出 16 位的 SRAM 地址。该寻址方式的寻址范围由指令中 16 位 SRAM 地址决定，即 64KB。例如：“LDS　Rd，K”。
- 数据存储器空间的寄存器间接寻址方式指的是由指令指定的某一个 16 位寄存器的内容作为 SRAM 中操作数的地址。在 ATmega128（L）中，常使用 16 位寄存器 X、Y 和 Z 保存该寻址方式下操作数的地址。例如：“LD　Rd，Y”。
- 带后增量的数据存储器空间的寄存器间接寻址方式与数据存储器空间的寄存器间接寻址方式类似，SRAM 中的操作数的地址仍放在寄存器 X、Y 和 Z 中。不同的是指令在间接寻址操作后，会自动的将寄存器 X、Y 和 Z 中的地址加 1，再把加 1 后的地址作为操作数的地址。该寻址方式特别适用于访问矩阵和查表等场合。例如：“LD　Rd，Y+”。
- 带预减量的数据存储器空间的寄存器间接寻址方式与数据存储器空间的寄存器间接寻址方式类似，SRAM 中的操作数的地址仍放在寄存器 X、Y 和 Z 中。不同的是指令在间接寻址操作之前，会自动的将寄存器 X、Y 和 Z 中的地址减 1，再把减 1 后的地址作为操作数的地址。该寻址方式特别适用于访问矩阵和查表等场合。例如：“LD　Rd，−Y”。
- 带位移的数据存储器空间的寄存器间接寻址方式指的是 SRAM 中操作数的地址由寄存器 Y 或 Z 及指令中给出的地址偏移量共同决定。其中，偏移量的范围为 0～63。例如：“LDD　Rd，Y+q”。

- 程序存储器空间常量寻址方式指的是直接给出 Flash 程序存储器中操作数（即常量）的地址。该地址存放在寄存器 Z 中，并且由寄存器 Z 的高 15 位决定（16 位地址的最高位为 0），而寄存器 Z 的最低位用于确定常量是程序存储器单元（程序存储器的存储单元为字）的高字节还是低字节。若最低位为 0，则选择低字节；若最低位为 1，则选择高字节。例如：“LPM　R15，Z”。
- 带后增量的程序存储器空间常量寻址方式与程序存储器空间常量寻址方式类似。不同的是寻址操作后，寄存器 Z 中的地址加 1。该寻址方式只适用于指令“LPM　Rd，Z+”。例如：“LPM　R15，Z+”。

注意 由于 ATmega128（L）的程序存储器空间大小为 64KB，而指令 LPM 只能对低 32KB 的程序存储器空间寻址，因此 ATmega128（L）使用 3 条扩展指令 ELPM 来辅助对高 32KB 的程序存储器空间进行寻址（RAMPZ=0：低 32KB；RAMPZ=1：高 32KB）。

- 程序存储器空间写数据寻址方式主要适用于指令 SPM。该指令将寄存器 R1 和 R0 中的内容组成一个字 R1:R0，然后写入由寄存器 Z 的内容作为地址的程序存储器单元中。例如：“SPM”。
- 程序存储器空间直接寻址方式指的是指令将一个 16 位的操作数送入程序计数器 PC 中，作为下一条指令在程序存储器的哪一存储单元中。该寻址方式适用于指令 JMP 和 CALL。例如：“JMP　$0E4F”。
- 程序存储器空间 Z 寄存器间接寻址方式把下一步要执行的指令在程序存储器中的地址放在寄存器 Z 中，即用寄存器 Z 的值代替 PC 的值。该寻址方式适用于指令 IJMP 和 ICALL。例如：“IJMP”。
- 程序存储器空间相对寻址方式指的是指令中包含一个相对偏移量 k，在指令执行时，先把当前 PC 的值加 1 后再与 k 相加，作为新的 PC 值。该寻址方式只适用于指令 RJMP 和 RCALL。例如：“RJMP　$6789”。
- 数据存储器空间堆栈指针 SP 间接寻址指的是将 16 位的堆栈指针 SP 的值作为 SRAM 空间的操作数的地址。该寻址方式适用于指令 PUSH 和 POP。例如：“PUSH　R3”。

3.3　算术和逻辑运算指令

ATmega128（L）的算术运算指令有加、减、乘、取反、取补、加 1 和减 1 指令；逻辑运算指令有与、或和异或指令。注意在本章的介绍中，指令周期指的是指令执行占用的系统时钟周期。

3.3.1　加法指令

ATmega128（L）的加法指令有：不带进位位加法指令、带进位位加法指令、字加立即数指令和加 1 指令，具体如表 3.6 所示。

表 3.6　　**加法指令**

不带进位位加法指令	指令语法	ADD　Rd，Rr　　0≤d，r≤31
	指令功能	两个寄存器 Rd 和 Rr 中的值相加，结果送目的寄存器 Rd
	指令操作	Rd←Rd+Rr，PC←PC+1
	指令机器码	0000 11rd dddd rrrr
	受影响的标志位	H、S、V、N、Z 和 C
	指令字长	2 个字节
	指令周期	1
带进位位加法指	指令语法	ADC　Rd，Rr　　0≤d，r≤31
	指令功能	两个寄存器 Rd 和 Rr 中的值及进位标志位 C 的值相加，结果送目的寄存器 Rd
	指令操作	Rd←Rd+Rr+C，PC←PC+1
	指令机器码	0001 11rd dddd rrrr
	受影响的标志位	H、S、V、N、Z 和 C

令	指令字长	2 个字节
	指令周期	1
字加立即数指令	指令语法	ADIW Rdl+1:Rdl, K dl 为 24、26、28 或 30，0≤K≤63
	指令功能	寄存器对 Rdl+1:Rdl 中的值（1 个字）与立即数 K（0～63）相加，结果放到寄存器对。此指令只用于最后 4 个寄存器对和指针寄存器
	指令操作	Rdl+1:Rdl←Rdl+1:Rdl+K，PC←PC+1
	指令机器码	1001 0110 KKdd KKKK
	受影响的标志位	H、V、N、Z 和 C
	指令字长	2 个字节
	指令周期	2
加 1 指令	指令语法	INC Rd 0≤d≤31
	指令功能	寄存器 Rd 中的值加 1，结果还送入寄存器 Rd 中。该操作不改变 SREG 中的进位标志位 C，所以指令 INC 可以在多倍字长计算中用作循环计数。注意：当对无符号数操作时，仅有 BREQ（相等跳转）和 BRNE（不相等跳转）指令有效；当对 2 进制补码值进行操作时，所有的带符号跳转指令都有效
	指令操作	Rd←Rd+1，PC←PC+1
	指令机器码	1001 010d dddd 0011
	受影响的标志位	S、V、N 和 Z
	指令字长	2 个字节
	指令周期	1

3.3.2 减法指令

ATmega128（L）的减法指令有：不带进位位减法指令、减立即数指令、带进位位减法指令、带进位位减立即数指令、减立即数指令和减 1 指令，具体如表 3.7 所示。

表 3.7 减法指令

不带进位位减法指令	指令语法	SUB Rd，Rr 0≤d，r≤31
	指令功能	将两个寄存器 Rd 和 Rr 中的值相减，结果放在目的寄存器 Rd 中。被减数放在寄存器 Rd 中，减数放在寄存器 Rr 中
	指令操作	Rd←Rd−Rr，PC←PC+1
	指令机器码	0001 10rd dddd rrrr
	受影响的标志位	H、S、V、N、Z 和 C
	指令字长	2 个字节
	指令周期	1
减立即数指令	指令语法	SUBI Rd，K 0≤d≤31，0≤K≤255
	指令功能	寄存器 Rd 中的值和立即数 K 相减，结果送入目的寄存器 Rd。该指令工作于寄存器 R16～R31 之间，非常适合 X、Y 和 Z 指针的操作
	指令操作	Rd←Rd−K，PC←PC+1
	指令机器码	0101 KKKK dddd KKKK
减立即数指令	受影响的标志位	H、S、V、N、Z 和 C
	指令字长	2 个字节
	指令周期	1

带进位位减法指令	指令语法	SBC Rd，Rr 0≤d，r≤31
	指令功能	两个寄存器 Rd 和 Rr 及进位标志位 C 相减，结果放在目的寄存器 Rd 中
	指令操作	Rd←Rd−Rr−C，PC←PC+1
	指令机器码	0100 10rd dddd rrrr
	受影响的标志位	H、S、V、N、Z 和 C
	指令字长	2 个字节
	指令周期	1
带进位位减立即数指令	指令语法	SBCI Rd，K 0≤d≤31，0≤K≤255
	指令功能	寄存器和立即数带着进位标志位 C 相减，结果放在目的寄存器 Rd 中。被减数放在寄存器 Rd 中，两个减数分别放在寄存器 Rr 和进位标志位 C 中
	指令操作	Rd←Rd−K−C，PC←PC+1
	指令机器码	0100 KKKK dddd KKKK
	受影响的标志位	H、S、V、N、Z 和 C
	指令字长	2 个字节
	指令周期	1
带进位位减立即数指令	指令语法	SBIW Rdl+1:Rdl，K dl=24、26、28 或 30，0≤K≤63
	指令功能	寄存器对 Rdl+1:Rdl 中的值（1 个字）减去一个立即数 K（0～63），并把结果存放在这个寄存器对里
	指令操作	Rdl+1:Rdl←Rdl+1:Rdl-K，PC←PC+1
	指令机器码	1001 0100 KKdd KKKK
	受影响的标志位	S、V、N、Z 和 C
	指令字长	2 个字节
	指令周期	2
减1指令	指令语法	DEC Rd 0≤d≤31
	指令功能	寄存器 Rd 中的值减 1，并将结果送回寄存器 Rd 中。该操作不改变 SREG 中的 C 标志，所以 INC 指令可以在多倍字长计算中用作循环计数。注意：当对无符号数操作时，仅有 BREQ（相等跳转）和 BRNE（不相等跳转）指令有效；当对 2 进制补码值进行操作时，所有的带符号跳转指令都有效
	指令操作	Rd←Rd−1，PC←PC+1
	指令机器码	1001 010d dddd 1010
	受影响的标志位	S、V、N 和 Z
	指令字长	2 个字节
	指令周期	1

3.3.3 取反码和补码指令

ATmega128（L）的取反码和补码指令如表 3.8 所示。

表 3.8 取反码和补码指令

取反码指令	指令语法	COM Rd 0≤d≤31
	指令功能	该指令完成对寄存器 Rd 中的值的二进制反码操作
	指令操作	Rd←$FF−Rd，PC←PC+1
	指令机器码	1001 010d dddd 0000
	受影响的标志位	S、V（0）、N、Z 和 C（1）
	指令字长	2 个字节

	指令周期	1
取补码指令	指令语法	NEG Rd 0≤d≤31
	指令功能	将寄存器 Rd 中的值用它的二进制补码取代
	指令操作	Rd←$00−Rd，PC←PC+1
	指令机器码	1001 010d dddd 0001
	受影响的标志位	H、S、V、N、Z 和 C
	指令字长	2 个字节
	指令周期	1

3.3.4 乘法指令

ATmega128（L）的乘法指令有：无符号数乘法指令、有符号数乘法指令、有符号数与无符号数乘法指令、无符号小数乘法指令、有符号小数乘法指令及有符号小数和无符号小数乘法指令，具体如表 3.9 所示。

表 3.9　　乘法指令

无符号数乘法指令	指令语法	MUL Rd，Rr 0≤d，r≤31
	指令功能	这条指令实现了 8 位无符号数×8 位无符号数→16 位的无符号数乘法。寄存器 Rd 中的被乘数和寄存器 Rr 中的乘数是两个无符号数；16 位的积被放在 R1（高字节）和 R0（低字节）中。注意，如果选定 R0 或 R1 中的值作为被乘数或乘数，运算后的结果将会覆盖掉它们原来的值。即： Rd×Rr→R1:R0 8 位被乘数 Rd×8 位乘数 Rr→积的高字节 R1:积的低字节 R0（16 位）
	指令操作	R1:R0←Rd×Rr，PC←PC+1
	指令机器码	1001 11rd dddd rrrr
	受影响的标志位	Z 和 C
	指令字长	2 个字节
	指令周期	2
有符号数乘法指令	指令语法	MULS Rd，Rr 16≤d，r≤31
	指令功能	这条指令实现了 8 位有符号数×8 位有符号数→16 位的有符号数乘法。寄存器 Rd 中的被乘数和寄存器 Rr 中的乘数是两个有符号数；16 位的积被放在 R1（高字节）和 R0（低字节）中。即： Rd×Rr→R1:R0 8 位被乘数 Rd×8 位乘数 Rr→积的高字节 R1:积的低字节 R0（16 位）
	指令操作	R1:R0←Rd×Rr，PC←PC+1
	指令机器码	0000 0010 dddd rrrr
	受影响的标志位	Z 和 C
	指令字长	2 个字节
	指令周期	2
有符号数与无符	指令语法	MULSU Rd，Rr 16≤d，r≤23
	指令功能	这条指令实现了 8 位有符号数×8 位无符号数→16 位的有符号数和无符号数相乘。寄存器 Rd 中的被乘数是有符号数，寄存器 Rr 中的乘数是无符号数。16 位的结果保存在 R1（高字节）和 R0（低字节）中。即： Rd×Rr→R1:R0 8 位被乘数 Rd×8 位乘数 Rr→积的高字节 R1:积的低字节 R0（16 位）

号数乘法指令	指令操作	R1:R0←Rd×Rr，PC←PC+1
	指令机器码	0000 0011 0ddd 0rrr
	受影响的标志位	Z 和 C
	指令字长	2 个字节
	指令周期	2
无符号小数乘法指令	指令语法	FMUL Rd，Rr 16≤d，r≤23
	指令功能	该指令完成寄存器 Rd 中的值（8 位无符号数）和寄存器 Rr 中的值（8 位无符号数）的内容相乘操作，结果为 16 位的带符号数，并将结果左移一位后保存在 R1:R0 中。R1 为高 8 位，R0 为低 8 位。被乘数 Rd 和乘数 Rr 是两个包含无符号定点小数的寄存器，小数点定位在第 7 位和第 6 位之间。结果为 16 位无符号定点小数，置于 R1（高位）和 R0（低位）之间，其小数点固定在第 14 位和第 15 位之间。即， Rd×Rr→R1:R0 8 位被乘数 Rd×8 位乘数 Rr→积的高字节 R1:积的低字节 R0（16 位）
	指令操作	R1:R0←Rd×Rr，PC←PC+1
	指令机器码	0000 0011 0ddd 1rrr
	受影响的标志位	Z 和 C
	指令字长	2 个字节
	指令周期	2
有符号小数乘法指令	指令语法	FMUL Rd，Rr 16≤d，r≤23
	指令功能	该指令完成寄存器 Rd 中的值（8 位有符号数）和寄存器 Rr 中的值（8 位有符号数）的内容相乘操作，结果为 16 位的有符号数，并将结果左移一位后保存在 R1:R0 中。R1 为高 8 位，R0 为低 8 位。被乘数 Rd 和乘数 Rr 是两个包含无符号定点小数的寄存器，小数点定位在第 7 位和第 6 位之间。结果为 16 位无符号定点小数，置于 R1（高位）和 R0（低位）之间，其小数点固定在第 14 位和第 15 位之间。即， Rd×Rr→R1:R0 8 位被乘数 Rd×8 位乘数 Rr→积的高字节 R1:积的低字节 R0（16 位）
	指令操作	R1:R0←Rd×Rr，PC←PC+1
	指令机器码	0000 0011 1ddd 0rrr
	受影响的标志位	Z 和 C
	指令字长	2 个字节
	指令周期	2
有符号小数和无符号小数乘法指令	指令语法	FMULSU Rd，Rr 16≤d，r≤23
	指令功能	该指令完成寄存器 Rd 中的值（8 位带符号数）和寄存器 Rr 中的值（8 位无符号数）的内容相乘操作，结果为 16 位的带符号数，并将结果左移一位后保存在 R1:R0 中。R1 为高 8 位，R0 为低 8 位。被乘数 Rd 是一个带符号的定点小数，乘数 Rr 是一个无符号的定点小数，而且小数点都定位在第 7 位和第 6 位之间。结果为 16 位无符号定点小数，置于 R1（高位）和 R0（低位）之间，其小数点固定在第 14 位和第 15 位之间。即， Rd×Rr→R1:R0 8 位被乘数 Rd×8 位乘数 Rr→积的高字节 R1:积的低字节 R0（16 位）
	指令操作	R1:R0←Rd×Rr，PC←PC+1
	指令机器码	0000 0011 1ddd 1rrr
	受影响的标志位	Z 和 C
	指令字长	2 个字节

	指令周期	2

注意 （N.Q）表示一个小数点左边有 N 个二进制数位、右边有 Q 个二进制数位的小数。以（N1.Q1）和（N2.Q2）为格式的两个小数相乘，产生格式为((N1+N2).(Q1+Q2))的结果。对于要有效保留小数位的处理应用，输入的数据通常采用（1.7）的格式，产生的结果为（2.14）的格式。因此将结果左移一位，以使高字节的格式与输入的一致。FMUL 指令的执行周期与 MUL 指令相同，但比 MUL 增加了左移操作。

3.3.5 逻辑与指令

ATmega128（L）的逻辑与指令有：寄存器逻辑与指令、与立即数指令、寄存器位清零指令和测试寄存器为 0 或负指令，具体如表 3.10 所示。

表 3.10 逻辑与指令

寄存器逻辑与指令	指令语法	AND Rd，Rr 0≤d，r≤31
	指令功能	寄存器 Rd 和寄存器 Rr 中的值进行逻辑与运算，结果送入目的寄存器 Rd
	指令操作	Rd←Rd·Rr，PC←PC+1
	指令机器码	0010 00rd dddd rrrr
	受影响的标志位	S、V（0）、N 和 Z
	指令字长	2 个字节
	指令周期	1
与立即数指令	指令语法	ANDI Rd，K 16≤d≤31，0≤K≤255
	指令功能	寄存器 Rd 中的值与常数 K 进行逻辑与运算，结果送入目的寄存器 Rd
	指令操作	Rd←Rd·K，PC←PC+1
	指令机器码	0111 KKKK dddd KKKK
	受影响的标志位	S、V（0）、N 和 Z
	指令字长	2 个字节
	指令周期	1
寄存器位清零指令	指令语法	CBR Rd，K 16≤d≤31，0≤K≤255
	指令功能	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，其结果送入寄存器 Rd 中
	指令操作	Rd←Rd·($FF−K)，PC←PC+1
	指令机器码	0111 MMMM dddd MMMM
	受影响的标志位	S、V（0）、N 和 Z
	指令字长	2 个字节
	指令周期	1
测试寄存器为 0 或负	指令语法	TST Rd 0≤d≤31
	指令功能	检测寄存器是否为零或负数，执行了寄存器 Rd 中的值和它本身的逻辑与。该指令只影响标志位而寄存器的内容将保持不变
	指令操作	Rd←Rd·Rd，PC←PC+1
	指令机器码	0010 00dd dddd dddd
	受影响的标志位	S、V（0）、N 和 Z
	指令字长	2 个字节

指令	指令周期	1

注意 MMMM 为 KKKK 的补码。下面内容如未加说明，MMMM 均表示为 KKKK 的补码。

3.3.6 逻辑或指令

ATmega128（L）的逻辑或指令有：寄存器逻辑或指令、或立即数指令、寄存器位置位指令和置位寄存器中的所有位指令，具体如表 3.11 所示。

表 3.11 逻辑或指令

寄存器逻辑或指令	指令语法	OR Rd，Rr 0≤d，r≤31
	指令功能	寄存器 Rd 和寄存器 Rr 中的值进行逻辑或运算，结果送入目的寄存器 Rd 中
	指令操作	Rd←Rd v Rr，PC←PC+1
	指令机器码	0010 10rd dddd rrrr
	受影响的标志位	S、V（0）、N 和 Z
	指令字长	2 个字节
	指令周期	1
或立即数指令	指令语法	ORI Rd，K 16≤d≤31，0≤K≤255
	指令功能	寄存器 Rd 中的值与立即数 K 进行逻辑或运算，结果送入目的寄存器 Rd 中
	指令操作	Rd←Rd v K，PC←PC+1
	指令机器码	0110 KKKK dddd KKKK
	受影响的标志位	S、V（0）、N 和 Z
	指令字长	2 个字节
	指令周期	1
寄存器位置位指令	指令语法	SBR Rd，K 16≤d≤31，0≤K≤255
	指令功能	通过寄存器 Rd 中的值与常数表征码 K 相或来置位寄存器 Rd 中的指定位
	指令操作	Rd←Rd v K，PC←PC+1
	指令机器码	0110 KKKK dddd KKKK
	受影响的标志位	S、V（0）、N 和 Z
	指令字长	2 个字节
	指令周期	1
置位寄存器中的所有位指令	指令语法	SER Rd 16≤d≤31
	指令功能	通过寄存器 Rd 中的值与常数$FF 相或来置位寄存器 Rd 中的所有位
	指令操作	Rd←Rd v $FF，PC←PC+1
	指令机器码	1110 1111 dddd 1111
	受影响的标志位	无
	指令字长	2 个字节
	指令周期	1

3.3.7 逻辑异或指令

ATmega128（L）的逻辑异或指令有：寄存器异或指令和寄存器清零指令，具体如表 3.12 所示。

表 3.12 逻辑异或指令

寄存	指令语法	EOR Rd，Rr 0≤d，r≤31
	指令功能	寄存器 Rd 和 Rr 中的值进行逻辑异或（EOR）运算，并将结果送入目的寄存器 Rd 中

器异或指令	指令操作	Rd←Rd⊕Rr，PC←PC+1
	指令机器码	0010 01rd dddd rrrr
	受影响的标志位	S、V（0）、N 和 Z
	指令字长	2 个字节
	指令周期	1
寄存器清零指令	指令语法	CLR Rd 0≤d≤31
	指令功能	寄存器 Rd 中的值与自己相异或，从而清零寄存器 Rd 中的所有位
	指令操作	Rd←Rd⊕Rd，PC←PC+1
	指令机器码	0010 01dd dddd dddd
	受影响的标志位	S（0）、V（0）、N（0）和 Z（1）
	指令字长	2 个字节
	指令周期	1

3.4 比较和转移指令

3.4.1 比较指令

ATmega128（L）的比较指令有：寄存器比较指令、带进位位比较指令和与立即数比较指令，具体如表 3.13 所示。

表 3.13 比较指令

寄存器比较指令	指令语法	CP Rd，Rr 0≤d，r≤31
	指令功能	两个寄存器 Rd 和 Rr 中的值作比较，而寄存器的内容不改变。该指令后能使用所有条件跳转指令
	指令操作	Rd−Rr，PC←PC+1
	指令机器码	0001 01rd dddd rrrr
	受影响的标志位	H、S、V、N、Z 和 C
	指令字长	2 个字节
	指令周期	1
带进位位比较指令	指令语法	CPC Rd，Rr 0≤d，r≤31
	指令功能	寄存器 Rd 中的值与寄存器 Rr 中的值和进位标志位 C 的值的和作比较，而寄存器的内容不改变。该指令后能使用所有条件跳转指令
	指令操作	Rd−Rr−C，PC←PC+1
	指令机器码	0000 01rd dddd rrrr
	受影响的标志位	H、S、V、N、Z 和 C
	指令字长	2 个字节
	指令周期	1
与立即数比较指令	指令语法	CPI Rd，K 16≤d≤31，0≤K≤255
	指令功能	寄存器 Rd 中的值与立即数 K 作比较，而寄存器的内容不改变。该指令后能使用所有条件跳转指令
	指令操作	Rd−K，PC←PC+1
	指令机器码	0011 KKKK dddd KKKK
	受影响的标志位	H、S、V、N、Z 和 C
	指令字长	2 个字节
	指令周期	1

3.4.2 无条件转移指令

ATmega128（L）的无条件转移指令有：相对跳转指令、间接跳转指令和直接跳转指令，具体如表 3.14 所示。

表 3.14　　无条件转移指令

相对跳转指令	指令语法	RJMP　k　　−2048≤k≤2048
	指令功能	相对跳转到 PC−2K+1～PC+2K（字）范围内的地址。在汇编语言程序中，标号常被用来代替 k 作为该指令的操作数
	指令操作	PC←PC+k+1
	指令机器码	1100 kkkk kkkk kkkk
	受影响的标志位	无
	指令字长	2 个字节
	指令周期	2
间接跳转指令	指令语法	IJMP
	指令功能	间接跳转到由指针寄存器 Z 中的值作为 16 位地址的地址单元。该指令无操作数
	指令操作	PC←Z(15:0)
	指令机器码	1001 0100 0000 1001
	受影响的标志位	无
	指令字长	2 个字节
	指令周期	2
直接跳转指令	指令语法	JMP　k　　0≤k≤4194304
	指令功能	直接跳转到指定的地址（可达整个的 4M 字的程序存储器）处。在汇编语言程序中，标号常被用来代替 k 作为该指令的操作数
	指令操作	PC←k
	指令机器码	1001 010k kkkk 110k kkkk kkkk kkkk kkkk
	受影响的标志位	无
	指令字长	4 个字节
	指令周期	3

3.4.3　条件转移指令

ATmega128（L）的条件转移指令，如表 3.15 所示，指的是根据某个特定条件，程序决定是否转移到别处执行的一类指令。当此特殊条件满足时，程序转移；否则程序顺序执行下一条指令。

表 3.15　　条件转移指令

SREG（s）为 1 转移指令	指令语法	BRBS　s，k　　0≤s≤7，−64≤k≤63
	指令功能	执行该指令时，PC 先加 1，再测试 SREG 中的第 s 位，当该位被置位时，则转移 k 个字（k 为 7 位带符号数），最多可向前转移 63 个字，向后转移 64 个字；否则顺序执行。在汇编语言程序中，标号常被用来代替 k 作为该指令的操作数
SREG（s）为 1 转移指令	指令操作	If SREG(s)=1 then PC←PC+k+1, else PC←PC+1
	指令机器码	1111 00kk kkkk ksss
	受影响的标志位	无
	指令字长	2 个字节
	指令周期	1（条件为假）；2（条件为真）
SREG（s）为 0 转移指令	指令语法	BRBC　s，k　　0≤s≤7，−64≤k≤63
	指令功能	执行该指令时，PC 先加 1，再测试 SREG 中的第 s 位，当该位被清零时，则转移 k 个字（k 为 7 位带符号数），最多可向前转移 63 个字，向后转移 64 个字；否则顺序执行。在汇编语言程序中，标号常被用来代替 k 作为该指令的操作数

	指令操作	If SREG(s)=0 then PC←PC+k+1, else PC←PC+1
	指令机器码	1111 01kk kkkk ksss
	受影响的标志位	无
	指令字长	2 个字节
	指令周期	1（条件为假）；2（条件为真）
相等转移指令	指令语法	BREQ k −64≤k≤63
	指令功能	如果在执行指令 CP、CPI、SUB 或 SUBI 后，立即执行该指令将检查零标志位 Z，如果 Z 位被置位（即寄存器 Rd 中数与寄存器 Rr 中的值相等），则程序相对当前 PC 值转移 k+1 个字（k 为 7 位带符号数），最多可向前转移 63 个字，向后转移 64 个字；否则程序将顺序执行。该指令相当于“BRBS 1，k”
	指令操作	If Rd=Rr (Z=1) then PC←PC+k+1, else PC←PC+1
	指令机器码	1111 00kk kkkk k001
	受影响的标志位	无
	指令字长	2 个字节
	指令周期	1（条件为假）；2（条件为真）
不相等转移指令	指令语法	BRNE k −64≤k≤63
	指令功能	如果在执行指令 CP、CPI、SUB 或 SUBI 后，立即执行该指令将检查零标志位 Z，如果 Z 位被清零（即寄存器 Rd 中数与寄存器 Rr 中的值不相等），则程序相对当前 PC 值转移 k+1 个字（k 为 7 位带符号数），最多可向前转移 63 个字，向后转移 64 个字；否则程序将顺序执行。该指令相当于“BRBC 1，k”
	指令操作	If Rd≠Rr (Z=0) then PC←PC+k+1, else PC←PC+1
	指令机器码	1111 01kk kkkk k001
	受影响的标志位	无
	指令字长	2 个字节
	指令周期	1（条件为假）；2（条件为真）
进位标志位 C 为 1 转移指令	指令语法	BRCS k −64≤k≤63
	指令功能	检查进位标志位 C，如果 C 位被置位，则程序相对当前 PC 值转移 k+1 个字（k 为 7 位带符号数），最多可向前转移 63 个字，向后转移 64 个字。该指令相当于“BRBS 0，k”
	指令操作	If C=1 then PC←PC+k+1, else PC←PC+1
	指令机器码	1111 00kk kkkk k000
进位标志位 C 为 1 转移指令	受影响的标志位	无
	指令字长	2 个字节
	指令周期	1（条件为假）；2（条件为真）
进位标志位 C 为 0 转移指令	指令语法	BRCC k −64≤k≤63
	指令功能	检查进位标志位 C，如果 C 位被清零，则程序相对当前 PC 值转移 k+1 个字（k 为 7 位带符号数），最多可向前转移 63 个字，向后转移 64 个字。该指令相当于“BRBC 0，k”
	指令操作	If C=0 then PC←PC+k+1, else PC←PC+1
	指令机器码	1111 01kk kkkk k000
	受影响的标志位	无
	指令字长	2 个字节
	指令周期	1（条件为假）；2（条件为真）
无符号数大于或等于转移指令	指令语法	BRSH k −64≤k≤63
	指令功能	如果在执行指令 CP、CPI、SUB 或 SUBI 后，立即执行该指令将检查进位标志位 C，如果 C 位被清零（即寄存器 Rd 中的无符号数大于或等于寄存器 Rr 中的无符号数），则程序相对当前 PC 值转移 k+1 个字（k 为 7 位带符号数），最多可向前转移 63 个字，向后转移 64 个字；否则程序将顺序执行。该指令相当于指令“BRBC 0，k”
	指令操作	If Rd≥Rr(C=0) then PC←PC+k+1, else PC←PC+1

	指令机器码	1111 01kk kkkk k000
	受影响的标志位	无
	指令字长	2 个字节
	指令周期	1（条件为假）；2（条件为真）
无符号数小于转移指令	指令语法	BRLO k −64≤k≤63
	指令功能	如果在执行指令 CP、CPI、SUB 或 SUBI 后，立即执行该指令将检查进位标志位 C，如果 C 位被置位（即寄存器 Rd 中的无符号数小于寄存器 Rr 中的无符号数），则程序相对当前 PC 值转移 k+1 个字（k 为 7 位带符号数），最多可向前转移 63 个字，向后转移 64 个字；否则程序将顺序执行。该指令相当于指令“BRBS 0，k”
	指令操作	If Rd＜Rr(C=1) then PC←PC+k+1, else PC←PC+1
	指令机器码	1111 00kk kkkk k000
	受影响的标志位	无
	指令字长	2 个字节
	指令周期	1（条件为假）；2（条件为真）
结果为负跳转指令	指令语法	BRMI k −64≤k≤63
	指令功能	检查负号标志位 N，如果 N 位被置位，则程序相对当前 PC 值转移 k+1 个字（k 为 7 位带符号数），最多可向前转移 63 个字，向后转移 64 个字；否则程序将顺序执行。该指令相当于指令“BRBS 2，k”
	指令操作	If N=1 then PC←PC+k+1, else PC←PC+1
	指令机器码	1111 00kk kkkk k010
	受影响的标志位	无
结果为负跳转指令	指令字长	2 个字节
	指令周期	1（条件为假）；2（条件为真）
结果为正跳转指令	指令语法	BRPL k −64≤k≤63
	指令功能	检查负号标志位 N，如果 N 位被清零，则程序相对当前 PC 值转移 k+1 个字（k 为 7 位带符号数），最多可向前转移 63 个字，向后转移 64 个字；否则程序将顺序执行。该指令相当于指令“BRBC 2，k”
	指令操作	If N=0 then PC←PC+k+1, else PC←PC+1
	指令机器码	1111 01kk kkkk k010
	受影响的标志位	无
	指令字长	2 个字节
	指令周期	1（条件为假）；2（条件为真）
有符号数大于或等于转移指令	指令语法	BRGE k −64≤k≤63
	指令功能	如果在执行指令 CP、CPI、SUB 或 SUBI 后，立即执行该指令将检查进位标志位 S，如果 S 位被清零（即寄存器 Rd 中的有符号数大于或等于寄存器 Rr 中的有符号数），则程序相对当前 PC 值转移 k+1 个字（k 为 7 位带符号数），最多可向前转移 63 个字，向后转移 64 个字；否则程序将顺序执行。该指令相当于指令“BRBC 4，k”
	指令操作	If Rd≥Rr(S=0) then PC←PC+k+1, else PC←PC+1
	指令机器码	1111 01kk kkkk k100
	受影响的标志位	无
	指令字长	2 个字节
	指令周期	1（条件为假）；2（条件为真）
有符号数小于转移指令	指令语法	BRLT k −64≤k≤63
	指令功能	如果在执行指令 CP、CPI、SUB 或 SUBI 后，立即执行该指令将检查进位标志位 S，如果 S 位被置位（即寄存器 Rd 中的有符号数小于寄存器 Rr 中的有符号数），则程序相对当前 PC 值转移 k+1 个字（k 为 7 位带符号数），最多可向前转移 63 个字，向后转移 64 个字；否则程序将顺序执行。该指令相当于指令“BRBS 4，k”
	指令操作	If Rd＜Rr(S=1) then PC←PC+k+1, else PC←PC+1

	指令机器码	1111 00kk kkkk k100
	受影响的标志位	无
	指令字长	2 个字节
	指令周期	1（条件为假）；2（条件为真）
半进位标志位 H 为 1 转移指令	指令语法	BRHS k −64≤k≤63
	指令功能	检查半进位标志位 H，如果 H 位被置位，则程序相对当前 PC 值转移 k+1 个字（k 为 7 位带符号数），最多可向前转移 63 个字，向后转移 64 个字。该指令相当于“BRBS 5，k”
	指令操作	If H=1 then PC←PC+k+1, else PC←PC+1
	指令机器码	1111 00kk kkkk k101
	受影响的标志位	无
	指令字长	2 个字节
	指令周期	1（条件为假）；2（条件为真）
半进位标志位 H 为 0 转移指令	指令语法	BRHC k −64≤k≤63
	指令功能	检查半进位标志位 H，如果 H 位被清零，则程序相对当前 PC 值转移 k+1 个字（k 为 7 位带符号数），最多可向前转移 63 个字，向后转移 64 个字。该指令相当于“BRBC 5，k”
	指令操作	If H=0 then PC←PC+k+1, else PC←PC+1
	指令机器码	1111 01kk kkkk k101
	受影响的标志位	无
	指令字长	2 个字节
	指令周期	1（条件为假）；2（条件为真）
标志位 T 为 1 转移指令	指令语法	BRTS k −64≤k≤63
	指令功能	检查标志位 T，如果 T 位被置位，则程序相对当前 PC 值转移 k+1 个字（k 为 7 位带符号数），最多可向前转移 63 个字，向后转移 64 个字。该指令相当于“BRBS 6，k”
	指令操作	If T=1 then PC←PC+k+1, else PC←PC+1
	指令机器码	1111 00kk kkkk k110
	受影响的标志位	无
	指令字长	2 个字节
	指令周期	1（条件为假）；2（条件为真）
标志位 T 为 0 转移指令	指令语法	BRTC k −64≤k≤63
	指令功能	检查标志位 T，如果 T 位被清零，则程序相对当前 PC 值转移 k+1 个字（k 为 7 位带符号数），最多可向前转移 63 个字，向后转移 64 个字。该指令相当于“BRBC 6，k”
	指令操作	If T=0 then PC←PC+k+1, else PC←PC+1
	指令机器码	1111 01kk kkkk k110
	受影响的标志位	无
	指令字长	2 个字节
	指令周期	1（条件为假）；2（条件为真）
标志位 V 为 1 转移指令	指令语法	BRVS k −64≤k≤63
	指令功能	检查标志位 V，如果 V 位被置位，则程序相对当前 PC 值转移 k+1 个字（k 为 7 位带符号数），最多可向前转移 63 个字，向后转移 64 个字。该指令相当于“BRBS 3，k”
	指令操作	If V=1 then PC←PC+k+1, else PC←PC+1
	指令机器码	1111 00kk kkkk k011
	受影响的标志位	无
	指令字长	2 个字节
	指令周期	1（条件为假）；2（条件为真）

标志位 V 为 0 转移指令	指令语法	BRVC k −64≤k≤63
	指令功能	检查标志位 V，如果 V 位被清零，则程序相对当前 PC 值转移 k+1 个字（k 为 7 位带符号数），最多可向前转移 63 个字，向后转移 64 个字。该指令相当于“BRBC 3，k”
标志位 V 为 0 转移指令	指令操作	If V=0 then PC←PC+k+1, else PC←PC+1
	指令机器码	1111 01kk kkkk k011
	受影响的标志位	无
	指令字长	2 个字节
	指令周期	1（条件为假）；2（条件为真）
标志位 I 为 1 转移指令	指令语法	BRIE k −64≤k≤63
	指令功能	检查标志位 I，如果 I 位被置位，则程序相对当前 PC 值转移 k+1 个字（k 为 7 位带符号数），最多可向前转移 63 个字，向后转移 64 个字。该指令相当于“BRBS 7，k”
	指令操作	If I=1 then PC←PC+k+1, else PC←PC+1
	指令机器码	1111 00kk kkkk k111
	受影响的标志位	无
	指令字长	2 个字节
	指令周期	1（条件为假）；2（条件为真）
标志位 I 为 0 转移指令	指令语法	BRID k −64≤k≤63
	指令功能	检查标志位 I，如果 I 位被清零，则程序相对当前 PC 值转移 k+1 个字（k 为 7 位带符号数），最多可向前转移 63 个字，向后转移 64 个字。该指令相当于“BRBC 7，k”
	指令操作	If I=0 then PC←PC+k+1, else PC←PC+1
	指令机器码	1111 01kk kkkk k111
	受影响的标志位	无
	指令字长	2 个字节
	指令周期	1（条件为假）；2（条件为真）
相等跳行指令	指令语法	CPSE Rd，Rr 0≤d，r≤31
	指令功能	该指令完成两个寄存器 Rd 和 Rr 中的值的比较。若两个值相等，则程序跳过下一条指令执行指令
	指令操作	If Rd=Rr then PC←PC+2 (or 3) else PC←PC+1
	指令机器码	0001 00rd dddd rrrr
	受影响的标志位	无
	指令字长	2 个字节
	指令周期	1（条件为假）；2（条件为真，跳过 1 个字的指令执行）；3（条件为真，跳过 2 个字的指令执行）
寄存器位为 0 跳行指令	指令语法	SBRC Rr，b 0≤r≤31，0≤b≤7
	指令功能	测试寄存器的某一位，如果这一位为 0 则跳过下一条指令
	指令操作	If Rr(b)=0 then PC←PC+2 (or 3) else PC←PC+1
	指令机器码	1111 110r rrrr 0bbb
	受影响的标志位	无
	指令字长	2 个字节
	指令周期	1（条件为假）；2（条件为真，跳过 1 个字的指令执行）；3（条件为真，跳过 2 个字的指令执行）
寄存器位为 1 跳行指令	指令语法	SBRS Rr，b 0≤r≤31，0≤b≤7
	指令功能	测试寄存器的某一位，如果这一位为 1 则跳过下一条指令
	指令操作	If Rr(b)=1 then PC←PC+2 (or 3) else PC←PC+1
	指令机器码	1111 111r rrrr 0bbb
	受影响的标志位	无

	指令字长	2 个字节
	指令周期	1（条件为假）；2（条件为真，跳过 1 个字的指令执行）；3（条件为真，跳过 2 个字的指令执行）
I/O 寄存器位为 0 跳行指令	指令语法	SBIC　P，b　　0≤P≤31，0≤b≤7
	指令功能	测试 I/O 寄存器的某一位，如果这一位为 0 则跳过下一条指令。该指令只适合低 32 个 I/O 寄存器（即 I/O 寄存器地址为 0～31），其地址为地址空间的 0～31
	指令操作	If P(b)=0 then PC←PC+2 (or 3) else PC←PC+1
	指令机器码	1001 1001 PPPP Pbbb
	受影响的标志位	无
	指令字长	2 个字节
	指令周期	1（条件为假）；2（条件为真，跳过 1 个字的指令执行）；3（条件为真，跳过 2 个字的指令执行）
I/O 寄存器位为 1 跳行指令	指令语法	SBIS　P，b　　0≤P≤31，0≤b≤7
	指令功能	测试 I/O 寄存器的某一位，如果这一位为 1 则跳过下一条指令。该指令只适合低 32 个 I/O 寄存器（即 I/O 寄存器地址为 0～31），其地址为地址空间的 0～31
	指令操作	If P(b)=1 then PC←PC+2 (or 3) else PC←PC+1
	指令机器码	1001 1011 PPPP Pbbb
	受影响的标志位	无
	指令字长	2 个字节
	指令周期	1（条件为假）；2（条件为真，跳过 1 个字的指令执行）；3（条件为真，跳过 2 个字的指令执行）

3.4.4　子程序调用与返回指令

ATmega128（L）的子程序调用与返回指令有：相对调用指令、间接调用指令、直接调用指令、子程序返回指令和中断服务程序返回指令，具体如表 3.16 所示。

表 3.16　　子程序调用与返回指令

相对调用指令	指令语法	RCALL　k　　−2048≤k≤2047
	指令功能	调用位于地址 PC−2047 到 PC+2048（字）处的子程序。子程序调用之前，返回地址（即 RCALL 后面那条指令的地址）保存到堆栈当中。在汇编语言程序中，标号常被用来代替 k 作为该指令的操作数
	指令操作	STACK←PC+1，SP←SP−2，PC←PC+1+k
	指令机器码	1101 kkkk kkkk kkkk
相对调用指令	受影响的标志位	无
	指令字长	2 个字节
	指令周期	3
间接调用指令	指令语法	ICALL
	指令功能	间接调用由 Z 寄存器中的值作为地址的子程序。由于地址指针寄存器 Z 为 16 位，允许调用在当前程序存储空间 64K 字（128KB）内的子程序。子程序调用之前，返回地址（即 RCALL 后面那条指令的地址）保存到堆栈当中
	指令操作	STACK←PC+1，SP←SP−2，PC←Z
	指令机器码	1001 0101 0000 1001
	受影响的标志位	无
	指令字长	2 个字节
	指令周期	3

直接调用指令	指令语法	CALL k　　0≤k≤65535
	指令功能	先将 PC+2 后的值（即指令 CALL 后的下一条指令地址）压入堆栈，然后直接调用地址 k 处的子程序
	指令操作	STACK←PC+2，SP←SP−2，PC←k
	指令机器码	1001 010k kkkk 111k kkkk kkkk kkkk kkkk
	受影响的标志位	无
	指令字长	4 个字节
	指令周期	4
子程序返回指令	指令语法	RET
	指令功能	从子程序返回，返回地址从堆栈中弹出
	指令操作	SP←SP+2，PC←STACK
	指令机器码	1001 0101 0000 1000
	受影响的标志位	无
	指令字长	2 个字节
	指令周期	4
中断服务程序返回指令	指令语法	RETI
	指令功能	从中断服务程序中返回，返回地址从堆栈中弹出，且全局中断使能
	指令操作	SP←SP+2，PC←STACK
	指令机器码	1001 0101 0001 1000
	受影响的标志位	无
	指令字长	2 个字节
	指令周期	4

3.5　数据传输指令

ATmega128（L）的数据传输指令主要有三大类：直接数据传输指令、间接数据传输指令和与程序存储器有关的数据传输指令。它们是在使用汇编语言编程时使用最频繁的一类指令。注意：所有的数据传输指令对标志位都没有影响。

3.5.1　直接数据传输指令

ATmega128（L）的直接数据传输指令有：工作寄存器间传输数据指令、SRAM 数据直接送寄存器指令、寄存器数据直接送 SRAM 指令和立即数送寄存器指令，具体如表 3.17 所示。

表 3.17　　直接数据传输指令

工作寄存器间传输数据指令	指令语法	MOV　Rd，Rr　　0≤d，r≤31
	指令功能	将寄存器 Rr 中的值送入寄存器 Rd 中，并且寄存器 Rr 中的值不变
	指令操作	Rd←Rr，PC←PC+1
	指令机器码	0010 11rd dddd rrrr
	受影响的标志位	无
	指令字长	2 个字节
	指令周期	1
SRAM 数据直接送寄存器指令	指令语法	LDS　Rd，k　　0≤d≤31，0≤k≤65535
	指令功能	把 SRAM 中一个存储单元（地址由 16 位数 k 决定）中的值（1 个字节）送入寄存器 Rd 中

	指令操作	Rd←(k)，PC←PC+2
	指令机器码	1001 000d dddd 0000 kkkk kkkk kkkk kkkk
	受影响的标志位	无
	指令字长	4 个字节
	指令周期	2
寄存器数据直接送 SRAM 指令	指令语法	STS k，Rr 0≤r≤31，0≤k≤65535
	指令功能	把寄存器 Rr 中的值（1 个字节）送入 SRAM 中一个存储单元（地址由 16 位数 k 决定）中
	指令操作	（k）←Rr，PC←PC+1
	指令机器码	1001 001d dddd 0000 kkkk kkkk kkkk kkkk
	受影响的标志位	无
	指令字长	4 个字节
	指令周期	2
立即数送寄存器指令	指令语法	LDI Rd，K 16≤d≤31，0≤K≤255
	指令功能	把一个 8 位立即数 K 送入寄存器 Rd 中
	指令操作	Rd←K，PC←PC+1
	指令机器码	1110 KKKK dddd KKKK
	受影响的标志位	无
	指令字长	2 个字节
	指令周期	1

3.5.2 间接数据传输指令

ATmega128（L）的间接数据传输指令有：使用地址指针寄存器 X 间接寻址传输数据指令、使用地址指针寄存器 Y 间接寻址传输数据指令和使用地址指针寄存器 Z 间接寻址传输数据指令。其中，使用地址指针寄存器 X 间接寻址将 SRAM 数据送入到寄存器的指令如表 3.18 所示；使用地址指针寄存器 X 间接寻址将寄存器中的数据送入 SRAM 中的指令如表 3.19 所示。

表 3.18 使用地址指针寄存器 X 间接寻址将 SRAM 数据送入到寄存器的指令

1	指令语法	LD Rd，X 0≤d≤31
	指令功能	将 SRAM 中的存储单元（地址由地址指针寄存器 X 中的值决定）的数据（1 个字节）送入寄存器 Rd 中，地址指针寄存器 X 中的值不变
	指令操作	Rd←(X)，PC←PC+1
	指令机器码	1001 000d dddd 1100
	受影响的标志位	无
	指令字长	2 个字节
	指令周期	2
2	指令语法	LD Rd，X+ 0≤d≤31
	指令功能	先将 SRAM 中的存储单元（地址由地址指针寄存器 X 中的值决定）的数据（1 个字节）送入寄存器 Rd 中，再将地址指针寄存器 X 中的值加 1
	指令操作	Rd←(X)，X←X+1，PC←PC+1
	指令机器码	1001 000d dddd 1101
	受影响的标志位	无
	指令字长	2 个字节
	指令周期	2

3	指令语法	LD Rd，−X 0≤d≤31
	指令功能	先将地址指针寄存器 X 中的值减 1，再将 SRAM 中的存储单元（地址由地址指针寄存器 X 中的值决定）的数据（1 个字节）送入寄存器 Rd 中
	指令操作	X←X−1，Rd←(X)，PC←PC+1
	指令机器码	1001 001r rrrr 1110
	受影响的标志位	无
	指令字长	2 个字节
	指令周期	2

表 3.19　使用地址指针寄存器 X 间接寻址将寄存器中的数据送入 SRAM 中的指令

1	指令语法	ST X，Rr 0≤r≤31
	指令功能	将寄存器 Rr 中的数据（1 个字节）送入 SRAM 中的存储单元（地址由地址指针寄存器 X 中的值决定），地址指针寄存器 X 中的值不变
	指令操作	(X)←Rr，PC←PC+1
	指令机器码	1001 001r rrrr 1100
1	受影响的标志位	无
	指令字长	2 个字节
	指令周期	2
2	指令语法	ST X+，Rr 0≤r≤31
	指令功能	先将寄存器 Rr 中的数据（1 个字节）送入 SRAM 中的存储单元（地址由地址指针寄存器 X 中的值决定），再将地址指针寄存器 X 中的值加 1
	指令操作	(X)←Rr，X←X+1，PC←PC+1
	指令机器码	1001 001r rrrr 1101
	受影响的标志位	无
	指令字长	2 个字节
	指令周期	2
3	指令语法	ST −X，Rr 0≤r≤31
	指令功能	先将地址指针寄存器 X 中的值减 1，再将寄存器 Rr 中的数据（1 个字节）送入 SRAM 中的存储单元（地址由地址指针寄存器 X 中的值决定）
	指令操作	X←X−1，(X)←Rr，PC←PC+1
	指令机器码	1001 001r rrrr 1110
	受影响的标志位	无
	指令字长	2 个字节
	指令周期	2

对于使用地址指针寄存器 Y 间接寻址传输数据指令和使用地址指针寄存器 Z 间接寻址传输数据指令，有一部分指令与使用地址指针寄存器 X 时的情形基本相同，只需将上述六条指令中的 X 换为 Y 或 Z 即可，所不同的是相关指令对应的机器码不同，如表 3.20 所示。

表 3.20　使用地址指针寄存器 Y 或 Z 间接寻址传输数据指令对应的机器码

指　令	指令对应的机器码	指　令	指令对应的机器码
LD Rd，Y	1000 000d dddd 1000	LD Rd，Z	1000 000d dddd 0000
LD Rd，Y+	1001 000d dddd 1001	LD Rd，Z+	1001 000d dddd 0001
LD Rd，−Y	1001 000d dddd 1010	LD Rd，−Z	1001 000d dddd 0010
ST Y，Rr	1000 001r rrrr 1000	ST Z，Rr	1000 001r rrrr 0000
ST Y+，Rr	1001 001r rrrr 1001	ST Z +，Rr	1001 001r rrrr 0001
ST −Y，Rr	1001 001r rrrr 1010	ST −Z，Rr	1001 001r rrrr 0010

使用地址指针寄存器Y间接寻址传输数据指令和使用地址指针寄存器Z间接寻址传输数据指令中与使用地址指针寄存器 X 时的指令不同的指令如表 3.21 所示。

表 3.21　使用地址指针寄存器 Y（或 Z）间接寻址的部分指令（与使用 X 时不同）

1	指令语法	LDD　Rd，Y(or Z)+q　　0≤d≤31，0≤q≤63
	指令功能	将 SRAM 中的存储单元的数据（1 个字节）送入寄存器 Rd 中，存储单元的地址由地址指针寄存器 Y 或 Z 中的值和 q 共同决定，地址指针寄存器 Y 或 Z 中的值不变
1	指令操作	Rd←(Y or Z+q)，PC←PC+1
	指令机器码	10q0 qq0d dddd 1qqq（使用 Y 时）或 10q0 qq0d dddd 0qqq（使用 Z 时）
	受影响的标志位	无
	指令字长	2 个字节
	指令周期	2
2	指令语法	STD　Y(or Z)+q，Rr　　0≤r≤31，0≤q≤63
	指令功能	将寄存器 Rr 中的数据（1 个字节）送入 SRAM 中的存储单元，存储单元的地址由地址指针寄存器 Y 或 Z 中的值和 q 共同决定，地址指针寄存器 Y 或 Z 中的值不变
	指令操作	(Y or Z+q)←Rd，PC←PC+1
	指令机器码	10q0 qq1r rrrr 1qqqq（使用 Y 时）或 10q0 qq1r rrrr 0qqq（使用 Z 时）
	受影响的标志位	无
	指令字长	2 个字节
	指令周期	2

注意　在访问矩阵表或堆栈指针时要经常用到上述 22 条指令。

3.5.3　与程序存储器有关的数据传输指令

ATmega128（L）中与程序存储器有关的数据传输指令有：程序存储器数据送入寄存器 R0 指令、程序存储器数据送入寄存器指令、带后增量的程序存储器数据送入寄存器指令、扩展的程序存储器数据送入寄存器 R0 指令、扩展的程序存储器数据送入寄存器指令、扩展的带后增量的程序存储器数据送入寄存器指令和写程序存储器指令，具体如表 3.22 所示。

程序存储器数据送入寄存器 R0 指令	指令语法	LPM
	指令功能	将程序存储器空间的一个存储单元中的数据（一个字节）送入寄存器 R0 中。存储单元的地址由 16 位地址指针寄存器 Z 中的值决定，寄存器 Z 的高 15 位为程序存储器的字地址（第 16 位地址第 16 位为 0）。寄存器 Z 中的最低位 LSB 为“0”时，存储单元数据的低字节送 R0 中；为“1”时，存储单元数据的高字节送 R0 中。该指令能寻址程序存储器空间为一个 64KB 的页（32K 字）
	指令操作	R0←(Z)，PC←PC+1
	指令机器码	1001 0101 1100 1000
	受影响的标志位	无
	指令字长	2 个字节
	指令周期	3
程序存储器数据送入寄存器指令	指令语法	LPM　Rd，Z　　0≤d≤31
	指令功能	将程序存储器空间的一个存储单元中的数据（一个字节）送入寄存器 Rd 中。存储单元的地址由 16 位地址指针寄存器 Z 中的值决定，寄存器 Z 的高 15 位为程序存储器的字地址（第 16 位地址第 16 位为 0）。寄存器 Z 中的最低位 LSB 为“0”时，存储单元数据的低字节送 Rd 中；为“1”时，存储单元数据的高字节送 Rd 中。该指令能寻址程序存储器空间为一个 64KB 的页（32K 字）

程序存储器数据送入寄存器指令	指令操作	Rd←(Z)，PC←PC+1
	指令机器码	1001 000d dddd 0100
	受影响的标志位	无
	指令字长	2 个字节
	指令周期	3
滞后增量的程序存储器数据送入寄存器指令	指令语法	LPM Rd，Z+　　0≤d≤31
	指令功能	先将程序存储器空间的一个存储单元（地址由寄存器 Z 中的值决定）中的数据（一个字节）送入寄存器 Rd 中，然后寄存器 Z 中的值加 1。存储单元的地址由 16 位地址指针寄存器 Z 中的值决定，寄存器 Z 的高 15 位为程序存储器的字地址（第 16 位地址第 16 位为 0）。寄存器 Z 中的最低位 LSB 为“0”时，存储单元数据的低字节送 Rd 中；为“1”时，存储单元数据的高字节送 Rd 中。该指令能寻址程序存储器空间为一个 64KB 的页（32K 字）
	指令操作	Rd←(Z)，Z←Z+1，PC←PC+1
	指令机器码	1001 000d dddd 0101
	受影响的标志位	无
	指令字长	2 个字节
	指令周期	3
扩展的程序存储器数据送入寄存器 R0 指令	指令语法	ELPM
	指令功能	将程序存储器空间的一个存储单元中的数据（一个字节）送入寄存器 R0 中。存储单元的地址由 16 位地址指针寄存器 Z 和寄存器 RAMPZ 共同决定，寄存器 Z 的高 15 位为程序存储器的字地址的低 15 位，第 16 位地址为寄存器 RAMPZ 中最低位。寄存器 Z 的最低位 LSB 为“0”时，存储单元数据的低字节送 R0 中；为“1”时，存储单元数据的高字节送 R0 中。该指令能寻址整个 128KB（64K 字）的程序存储器空间
	指令操作	R0←(RAMPZ:Z)，PC←PC+1
	指令机器码	1001 0101 1101 1000
	受影响的标志位	无
	指令字长	2 个字节
	指令周期	3
扩展的程序存储器数据送入寄存器指令	指令语法	ELPM　Rd，Z　　0≤d≤31
	指令功能	将程序存储器空间的一个存储单元中的数据（一个字节）送入寄存器 Rd 中。存储单元的地址由 16 位地址指针寄存器 Z 和寄存器 RAMPZ 共同决定，寄存器 Z 的高 15 位为程序存储器的字地址的低 15 位，第 16 位地址为寄存器 RAMPZ 中最低位。寄存器 Z 的最低位 LSB 为“0”时，存储单元数据的低字节送 Rd 中；为“1”时，存储单元数据的高字节送 Rd 中。该指令能寻址整个 128KB（64K 字）的程序存储器空间
	指令操作	Rd←(RAMPZ:Z)，PC←PC+1
	指令机器码	1001 000d dddd 0110
	受影响的标志位	无
	指令字长	2 个字节
	指令周期	3
扩展的滞后增量的程序存储器数据送入寄存器指令	指令语法	LPM　Rd，Z+　　0≤d≤31
	指令功能	先将程序存储器空间的一个存储单元中的数据（一个字节）送入寄存器 Rd 中，然后寄存器 Z 中的值加 1。存储单元的地址由 16 位地址指针寄存器 Z 和寄存器 RAMPZ 共同决定，寄存器 Z 的高 15 位为程序存储器的字地址的低 15 位，第 16 位地址为寄存器 RAMPZ 中最低位。寄存器 Z 的最低位 LSB 为“0”时，存储单元数据的低字节送 Rd 中；为“1”时，存储单元数据的高字节送 Rd 中。该指令能寻址整个 128KB（64K 字）的程序存储器空间
	指令操作	Rd←(RAMPZ:Z)，RAMPZ:Z←RAMPZ:Z+1，PC←PC+1
	指令机器码	1001 000d dddd 0111
	受影响的标志位	无

	指令字长	2 个字节
	指令周期	3
写程序存储器指令	指令语法	SPM
	指令功能	将寄存器对 R1:R0 的数据（16 位字）写入程序存储器空间的存储单元中（地址由寄存器 Z 中的值决定）。寄存器 R1 中保存写入数据的高字节；寄存器 R0 中保存写入数据的低字节。该指令能寻址程序存储器空间为一个 64KB 的页（32K 字）
	指令操作	(Z)←R1:R0，PC←PC+1
	指令机器码	1001 0101 1110 1000
	受影响的标志位	无
	指令字长	2 个字节
	指令周期	取决于具体操作

3.5.4 I/O 口数据传输指令

ATmega128（L）的 I/O 口数据传输指令有：I/O 口数据送入寄存器指令和寄存器数据送 I/O 口指令，具体如表 3.23 所示。

表 3.23　　I/O 口数据传输指令

I/O 口数据送入寄存器指令	指令语法	IN　Rd，P　　0≤d≤31，0≤P≤63
	指令功能	将 I/O 空间（包括端口、定时器、相关的配置寄存器等）的数据传送入寄存器 Rd 中
	指令操作	Rd←P，PC←PC+1
	指令机器码	1011 0PPd dddd PPPP
	受影响的标志位	无
	指令字长	2 个字节
	指令周期	1
寄存器数据送 I/O 口指令	指令语法	OUT　P，Rr　　0≤r≤31，0≤P≤63
	指令功能	将寄存器 Rr 中的数据送到 I/O 空间（包括端口、定时器、相关的配置寄存器等）中
	指令操作	P←Rr，PC←PC+1
	指令机器码	1011 1PPr rrrr PPPP
	受影响的标志位	无
	指令字长	2 个字节
	指令周期	1

3.5.5 堆栈操作指令

ATmega128（L）的堆栈操作指令有：进栈指令和出栈指令，具体如表 3.24 所示。

表 3.24　　堆栈操作指令

进栈指令	指令语法	PUSH　Rd　　0≤d≤31
	指令功能	将寄存器 Rd 中的值送入堆栈中
	指令操作	STACK←Rr，SP←SP−1，PC←PC+1
	指令机器码	1001 001d dddd 1111
	受影响的标志位	无
	指令字长	2 个字节
	指令周期	2
出栈	指令语法	P POP　Rd　　0≤d≤31
	指令功能	将堆栈中的值送入寄存器 Rd 中

指令	指令操作	SP←SP+1，Rd←STACK，PC←PC+1
	指令机器码	1001 000d dddd 1111
	受影响的标志位	无
	指令字长	2 个字节
	指令周期	2

3.6 位操作指令

在 ATmega128（L）中，有 1/4 的指令都是与位操作有关的。这些位操作指令主要有三大类：带进位的位操作指令、位变量传输指令和位变量修改指令。

3.6.1 带进位的位操作指令

ATmega128（L）的带进位的位操作指令有：寄存器逻辑左移指令、寄存器逻辑右移指令、带进位位的寄存器逻辑循环左移指令、带进位位的寄存器逻辑循环右移指令、寄存器算术右移指令和寄存器半字节交换指令，具体如表 3.25 所示。

表 3.25　　带进位的位操作指令

寄存器逻辑左移指令	指令语法	LSL　Rd　　0≤d≤31
	指令功能	寄存器 Rd 中的所有位左移 1 位，第 0 位补 0，第 7 位移入 SREG 中的标志位 C 中。该指令完成一个无符号数乘 2 的操作
	指令操作	C←$b_7b_6b_5b_4b_3b_2b_1b_0$←0，PC←PC+1
	指令机器码	0000 11dd dddd dddd
	受影响的标志位	H、S、V、N、Z 和 C
	指令字长	2 个字节
	指令周期	1
寄存器逻辑右移指令	指令语法	LSR　Rd　　0≤d≤31
	指令功能	寄存器 Rd 中的所有位右移 1 位，第 7 位补 0，第 0 位移入 SREG 中的标志位 C 中。该指令完成一个无符号数除以 2 的操作，标志位 C 被用于结果舍入
	指令操作	0→$b_7b_6b_5b_4b_3b_2b_1b_0$→C，PC←PC+1
	指令机器码	1001 010d dddd 0110
	受影响的标志位	S、V、N（0）、Z 和 C
	指令字长	2 个字节
	指令周期	1
带进位位的寄存器逻辑循环左移指令	指令语法	ROL　Rd　　0≤d≤31
	指令功能	寄存器 Rd 的所有位左移 1 位，标志位 C 移入寄存器 Rd 中的第 0 位，寄存器 Rd 中的第 7 位移入标志位 C
	指令操作	C←$b_7b_6b_5b_4b_3b_2b_1b_0$←C，PC←PC+1
	指令机器码	0001 11dd dddd dddd
	受影响的标志位	H、S、V、N、Z 和 C
	指令字长	2 个字节
	指令周期	1
带进位位的寄存器逻辑循环	指令语法	ROR　Rd　　0≤d≤31
	指令功能	寄存器 Rd 的所有位右移 1 位，标志位 C 移入寄存器 Rd 中的第 7 位，寄存器 Rd 中的第 0 位移入标志位 C
	指令操作	C→$b_7b_6b_5b_4b_3b_2b_1b_0$→C，PC←PC+1
	指令机器码	1001 010d dddd 0111

右移指令	受影响的标志位	S、V、N、Z 和 C
	指令字长	2 个字节
	指令周期	1
寄存器算术右移指令	指令语法	ASR Rd 0≤d≤31
	指令功能	寄存器 Rd 中的所有位右移 1 位，第 7 位保持原逻辑值不变，第 0 位移入标志位 C 中。这个操作实现数的二进制补码值除 2，而不改变符号，标志位 C 用于结果的舍入
	指令操作	$b_7 \rightarrow b_7b_6b_5b_4b_3b_2b_1b_0 \rightarrow C$，PC←PC+1
	指令机器码	1001 010d dddd 0101
	受影响的标志位	S、V、N、Z 和 C
	指令字长	2 个字节
	指令周期	1
寄存器半字节交换指令	指令语法	SWAP Rd 0≤d≤31
	指令功能	寄存器 Rd 中的值的高半字节（高 4 位）和低半字节（低 4 位）交换
	指令操作	$b_7b_6b_5b_4 \leftrightarrow b_3b_2b_1b_0$，PC←PC+1
	指令机器码	1001 010d dddd 0010
	受影响的标志位	无
	指令字长	2 个字节
	指令周期	1

3.6.2 位变量传输指令

ATmega128（L）的位变量传输指令有：寄存器位的值送入标志位 T 指令和标志位 T 的值送入寄存器指定位指令，具体如表 3.26 所示。

表 3.26 位变量传输指令

寄存器位的值送入标志位 T 指令	指令语法	BST Rd，b 0≤d≤31，0≤b≤7
	指令功能	把寄存器 Rd 中的位 b 的值送入标志位 T 中
	指令操作	T←Rr(b)，PC←PC+1
	指令机器码	1111 101d dddd 0bbb
	受影响的标志位	T
	指令字长	2 个字节
	指令周期	1
标志位 T 的值送入寄存器中指定位指令	指令语法	BLD Rd，d 0≤d≤31，0≤b≤7
	指令功能	把标志位 T 的值送入寄存器 Rd 中的位 b 中
	指令操作	Rd(b)←T，PC←PC+1
	指令机器码	1111 100d dddd 0bbb
	受影响的标志位	无
	指令字长	2 个字节
	指令周期	1

3.6.3 位修改指令

ATmega128（L）的位修改指令指的是对状态寄存器 SREG 的位或 I/O 寄存器中的指定位进行置位或清 0 操作，具体如表 3.27 所示。

表 3.27 位修改指令

状态寄存器 SREG 的指定位置位指令	指令语法	BSET s 0≤s≤7
	指令功能	状态寄存器 SREG 的位 s 置位
	指令操作	SREG(s)←1，PC←PC+1
	指令机器码	1001 0100 0sss 1000
	受影响的标志位	I、T、H、S、V、N、Z 和 C
	指令字长	2 个字节
	指令周期	1
状态寄存器 SREG 的指定位清零指令	指令语法	BCLR s 0≤s≤7
	指令功能	状态寄存器 SREG 的位 s 清 0
	指令操作	SREG(s)←0，PC←PC+1
	指令机器码	1001 0100 1sss 1000
	受影响的标志位	I、T、H、S、V、N、Z 和 C
	指令字长	2 个字节
	指令周期	1
I/O 寄存器的指定位置位指令	指令语法	SBI P，b 0≤P≤31，0≤b≤7
	指令功能	将由 P 指定的 I/O 寄存器中的位 b 置位。该指令只在低 32 个 I/O 寄存器（即 I/O 寄存器地址为 0～31）内操作
	指令操作	I/O(P,b)←1，PC←PC+1
	指令机器码	1001 1010 PPPP Pbbb
	受影响的标志位	I、T、H、S、V、N、Z 和 C
	指令字长	2 个字节
	指令周期	2
I/O 寄存器的指定位清零指令	指令语法	CBI P，b 0≤P≤31，0≤b≤7
	指令功能	将由 P 指定的 I/O 寄存器中的位 b 清 0。该指令只在低 32 个 I/O 寄存器（即 I/O 寄存器地址为 0~31）内操作
	指令操作	I/O(P,b)←0，PC←PC+1
	指令机器码	1001 1000 PPPP Pbbb
	受影响的标志位	无
	指令字长	2 个字节
	指令周期	2
进位位置位指令	指令语法	SEC
	指令功能	将进位标志位 C 置位
	指令操作	C←1，PC←PC+1
	指令机器码	1001 0100 0000 1000
	受影响的标志位	C（1）
	指令字长	2 个字节
	指令周期	1
进位位清零指令	指令语法	CLC
	指令功能	进位标志位 C 清 0
	指令操作	C←0，PC←PC+1
	指令机器码	1001 0100 1000 1000
	受影响的标志位	C（0）
	指令字长	2 个字节
	指令周期	1
负数标志	指令语法	SEN

位置位指令	指令功能	负数标志位 N 置位
	指令操作	N←1，PC←PC+1
	指令机器码	1001 0100 0010 1000
	受影响的标志位	N（1）
	指令字长	2 个字节
	指令周期	1
负数标志位清零指令	指令语法	CLN
	指令功能	负数标志位 N 清 0
	指令操作	N←0，PC←PC+1
	指令机器码	1001 0100 1010 1000
	受影响的标志位	N（0）
	指令字长	2 个字节
	指令周期	1
零标志位置位指令	指令语法	SEZ
	指令功能	零标志位 Z 置位
	指令操作	Z←1，PC←PC+1
	指令机器码	1001 0100 0001 1000
	受影响的标志位	Z（1）
	指令字长	2 个字节
	指令周期	1
零标志位清零指令	指令语法	CLZ
	指令功能	零标志位 Z 清 0
	指令操作	Z←0，PC←PC+1
	指令机器码	1001 0100 1001 1000
	受影响的标志位	Z（0）
	指令字长	2 个字节
	指令周期	1
全局中断使能标志位置位指令	指令语法	SEI
	指令功能	全局中断使能标志位 I 置位
	指令操作	I←1，PC←PC+1
	指令机器码	1001 0100 0111 1000
	受影响的标志位	I（1）
	指令字长	2 个字节
	指令周期	1
全局中断使能标志位清零指令	指令语法	CLI
	指令功能	全局中断使能标志位 I 清 0
	指令操作	I←0，PC←PC+1
	指令机器码	1001 0100 1111 1000
	受影响的标志位	I（0）
	指令字长	2 个字节
	指令周期	1
符号标志位置位指令	指令语法	SES
	指令功能	符号标志位 S 置位
	指令操作	S←1，PC←PC+1

续表

符号标志	指令机器码	1001 0100 0100 1000

位置位指令	受影响的标志位	S（1）
	指令字长	2 个字节
	指令周期	1
符号标志位置位清零指令	指令语法	CLS
	指令功能	符号标志位 S 清 0
	指令操作	S←0，PC←PC+1
	指令机器码	1001 0100 1100 1000
	受影响的标志位	S（0）
	指令字长	2 个字节
	指令周期	1
溢出标志位置位指令	指令语法	SEV
	指令功能	溢出标志位 V 置位
	指令操作	V←1，PC←PC+1
	指令机器码	1001 0100 0011 1000
	受影响的标志位	V（1）
	指令字长	2 个字节
	指令周期	1
溢出标志位清零指令	指令语法	CLV
	指令功能	溢出标志位 V 清 0
	指令操作	V←0，PC←PC+1
	指令机器码	1001 0100 1011 1000
	受影响的标志位	V（0）
	指令字长	2 个字节
	指令周期	1
标志位 T 置位指令	指令语法	SET
	指令功能	标志位 T 置位
	指令操作	T←1，PC←PC+1
	指令机器码	1001 0100 0110 1000
	受影响的标志位	T（1）
	指令字长	2 个字节
	指令周期	1
标志位 T 清零指令	指令语法	CLT
	指令功能	标志位 T 清 0
	指令操作	T←0，PC←PC+1
	指令机器码	1001 0100 1110 1000
	受影响的标志位	T（0）
	指令字长	2 个字节
	指令周期	1
半进位标志位置位指令	指令语法	SEH
	指令功能	标半进位标志位 H 置位
	指令操作	H←1，PC←PC+1
	指令机器码	1001 0100 0101 1000
	受影响的标志位	H（1）
	指令字长	2 个字节
	指令周期	1
半进位标志位清零指令	指令语法	CLH
	指令功能	半进位标志位 H 清 0
	指令操作	H←0，PC←PC+1

	指令机器码	1001 0100 1101 1000
	受影响的标志位	H（0）
	指令字长	2 个字节
	指令周期	1

3.7 MCU 控制指令

ATmega128（L）的 MCU 控制指令有：空操作指令、休眠指令和看门狗定时器清零指令。这些指令主要用于控制 MCU 的运行方式以及清零看门狗定时器，具体如表 3.28 所示。

表 3.28 MCU 控制指令

空操作指令	指令语法	NOP
	指令功能	完成一个单周期空操作
	指令操作	PC←PC+1
	指令机器码	0000 0000 0000 0000
	受影响的标志位	无
	指令字长	2 个字节
	指令周期	1
休眠指令	指令语法	SLEEP
	指令功能	该指令使 MCU 根据设置的休眠模式进入休眠状态。当 MCU 从休眠状态被一个中断唤醒时，在中断服务程序执行后，紧跟执行休眠指令后的下一条指令将
	指令操作	PC←PC+1，MCU 进入由 MCU 控制寄存器设置的休眠模式运行
	指令机器码	1001 0101 1000 1000
	受影响的标志位	无
	指令字长	2 个字节
	指令周期	1
看门狗定时器清零指令	指令语法	WDR
	指令功能	该指令清零看门狗定时器。在使能看门狗定时器情况下，系统程序在正常运行中必须在 WD 预分频器（WD Prescaler）给出的限定时间内执行一次该指令，以防止看门狗定时器溢出，从而造成系统复位
	指令操作	清零看门狗定时器，PC←PC+1
	指令机器码	1001 0101 1010 1000
	受影响的标志位	无
	指令字长	2 个字节
	指令周期	1

3.8 ATmega128（L）的汇编语言

所谓的汇编语言就是使用助记符来代替实际二进制机器指令的符号化语言。用汇编语言编写的程序称为汇编语言程序或源程序。注意下面介绍的内容适用于 ATMEL 公司的 AVR 单片机开发环境 AVRStudio 4.12 及相应的汇编器。

3.8.1 汇编语言语句格式及伪指令

ATmega128（L）的汇编语言源程序是由一系列汇编语句组成的，其汇编语句的标准格式有以下两种（“[ ]”内的内容为可选项）。

- [标号:] 伪指令 [操作数] [;注释]
- [标号:] 指令 [操作数] [;注释]

具体说明如下。

（1）标号：汇编语句地址的标记符号，用于帮助对该汇编语句的访问和定位。使用标号时要注意以下几点：

- 标号一般由 ASCII 字符组成，第一个字符为字母；
- 同一标号在一个独立的程序中只能定义一次；
- 不能使用汇编语言中的保留字，如指令助记符、寄存器名、伪指令字等。

（2）伪指令：不属于系统指令集，而且编译时不产生实际的目标机器代码，只是用在汇编程序中对地址、寄存器、数据或常量等进行定义说明，以及对编译过程进行某种控制等。注意：伪指令通常由汇编编译系统给出。

（3）指令：汇编程序源程序的核心部分，即本章所介绍的指令。

（4）操作数：是指令操作时所需要的数据或地址。在编译器的支持下，可以使用多种形式的操作数，如数字、标识符或表达式等，而不是仅局限于指令系统中所定义的操作数格式。

（5）注释：用于对源程序和相关汇编语句进行说明，帮助程序设计人员阅读、理解和修改程序。在汇编语言源程序中，符号“;”后面的内容即为注释，注释内容长度不限。注释内容换行时，开头部分还要使用符号“;”。注意：编译器对注释的内容不予理会，而且注释不产生任何代码。

（6）在汇编语句中，冒号“:”用于标号之后；空格用于分隔指令字和操作数；指令有两个操作数时用逗号“,”分隔两个操作数。注意：所有的标点符号必须在英文状态下输入。

在 AVRstudio 的汇编器中提供了一些汇编伪指令，如表 3.29 所示。这些伪指令不能直接转换生成目标代码，而是用于调整程序存储器中程序的位置、定义宏和初始化程序存储器等。

表 3.29　　汇编器支持的伪指令

伪　指　令	说　　明	伪　指　令	说　　明
BYTE	在 SRAM 中预留存储单元	NOLIST	禁止列表文件生成
CSEG	声明代码段	ORG	设置程序起始位置
DB	定义字节常数	SET	赋值给标识符
DEF	定义寄存器符号名	ELSE	条件汇编
DEVICE	指定为何器件生成汇编代码	ELIF	条件汇编
DSEG	声明数据段	ENDIF	条件汇编
DW	定义字常数	ERROR	输出出错消息
EQU	定义标识符常量	IF	条件汇编
ESEG	声名 E^2PROM 段	IFDEF	条件汇编
EXIT	退出文件	IFNDEF	条件汇编
INCLUDE	包含指定的文件	MESSAGE	输出消息字符串
MACRO	宏定义开始	DD	定义双字常量
ENDMACRO	宏定义结束	DQ	定义四字常量
LISTMAC	列表宏表达式	UNDEF	取消寄存器符号的定义
LIST	使能列表文件生成	WARNING	输出警告消息

3.8.2　表达式

在 AVR 编译器的支持下，在编写汇编语言源程序时，指令的操作数可以使用表达式，而不再需要严格遵守指令定义的操作数的格式。AVR 编译器支持的表达式是由操作数、函数和运算符组成，并且所有的表达式内部都是 32 位的。注意表达式是可以嵌套使用的。

下面来分别介绍表达式中可以使用的操作数、函数和运算符种类。

（1）AVR 汇编器的表达式中使用的操作数有以下几种：

- 自定义的标号，该标号给出了放置标号处的程序计数器的值；
- 用伪指令 SET 定义的变量；
- 用伪指令 EQU 定义的常数；
- 整数常数，常见的有十进制数、十六进制数和二进制数；
- PC：程序计数器的当前值。

（2）AVR 汇编器的表达式中使用的函数有以下几种：

- LOW（表达式）：返回一个表达式值的最低字节；
- HIGH（表达式）：返回一个表达式值的第二个字节；
- BYTE2（表达式）：与 HIGH 函数相同；
- BYTE3（表达式）：返回一个表达式值的第三个字节；
- BYTE4（表达式）：返回一个表达式值的第四个字节；
- LWRD（表达式）：返回一个表达式值的 0～15 位；
- HWRD（表达式）：返回一个表达式值的 16～31 位；
- PAGE（表达式）：返回一个表达式值的 16～21 位；
- EXP2（表达式）：返回（表达式值）2 次幂的值；
- LOG2（表达式）：返回 Log_2（表达式值）的整数部分。

（3）AVR 汇编器的表达式中使用的运算符如表 3.30 所示。注意运算符的优先级数越高，其优先级也就越高。

表 3.30　　汇编器支持的运算符

运算符	名称	优先级	说　明
!	逻辑非	14	一元运算符，表达式值是 0 返回 1；表达式值是 1 返回 0
～	逐位非	14	一元运算符，将表达式的值按位取反
–	负号	14	一元运算符，使表达式值为算术负
*	乘法	13	二元运算符，两个表达式相乘
/	除法	13	二元运算符，左边表达式除以右边表达式，结果取整
+	加法	12	二元运算符，两个表达式相加
–	减法	12	二元运算符，左边表达式减去右边表达式
<<	左移	11	二元运算符，左边表达式值按右边表达式给出的次数左移
>>	右移	11	二元运算符，左边表达式值按右边表达式给出的次数右移
<	小于	10	二元运算符，左边表达式值小于右边表达式值，则为 1；否则为 0
<=	小于等于	10	二元运算符，左边表达式值不大于右边表达式值，则为 1；否则为 0
>	大于	10	二元运算符，左边表达式值大于右边表达式值，则为 1；否则为 0
>=	大于等于	10	二元运算符，左边表达式值不小于右边表达式值，则为 1；否则为 0
==	等于	9	二元运算符，左边表达式值等于右边表达式值，则为 1；否则为 0
!=	不等于	9	二元运算符，左边表达式值不等于右边表达式值，则为 1；否则为 0
&	逐位与	8	二元运算符，两个表达式值之间按位与
^	逐位异或	7	二元运算符，两个表达式值之间按位异或
|	逐位或	6	二元运算符，两个表达式值之间按位或
&&	逻辑与	5	二元运算符，两个表达式值之间逻辑与，全非 0 则为 1；否则为 0

||	逻辑或	4	二元运算符，两个表达式值之间逻辑或，有 1 个非 0 则为 1，全 0 为 0

3.9 本章小结

在本章中，我们对 ATmega128（L）的指令系统做了全面的介绍，重点介绍并分析了指令系统的每一条指令的具体功能。最后介绍了 ATmega128（L）汇编语言编程的一些基本知识。

联系方式

集团官网：www.hqyj.com　　嵌入式学院：www.embedu.org　　移动互联网学院：www.3g-edu.org

企业学院：www.farsight.com.cn　　物联网学院：www.topsight.cn　　研发中心：dev.hqyj.com

集团总部地址：北京市海淀区西三旗悦秀路北京明园大学校内 华清远见教育集团

北京地址：北京市海淀区西三旗悦秀路北京明园大学校区，电话：010-82600386/5

上海地址：上海市徐汇区漕溪路 250 号银海大厦 11 层 B 区，电话：021-54485127

深圳地址：深圳市龙华新区人民北路美丽 AAA 大厦 15 层，电话：0755-25590506

成都地址：成都市武侯区科华北路 99 号科华大厦 6 层，电话：028-85405115

南京地址：南京市白下区汉中路 185 号鸿运大厦 10 层，电话：025-86551900

武汉地址：武汉市工程大学卓刀泉校区科技孵化器大楼 8 层，电话：027-87804688

西安地址：西安市高新区高新一路 12 号创业大厦 D3 楼 5 层，电话：029-68785218

广州地址：广州市天河区中山大道 268 号天河广场 3 层，电话：020-28916067