

AVR 单片机指令系统快速记忆方法及速查表

计算机的指令系统是一套控制计算机操作的代码，称之为机器语言。计算机只能识别和执行机器语言的指令。为了便于人们理解、记忆和使用，通常用汇编语言指令来描述计算机的指令系统。汇编语言指令可通过汇编器翻译成计算机能识别的机器语言。

AVR 单片机指令系统是 RISC 结构的精简指令集，是一种简明易掌握、效率高的指令系统。

AVR 单片机指令系统速查表，不同器件使用不同的指令表，见附录 3：

- (1) 89 条指令器件:AT90S1200,最基本指令;
- (2) 90 条指令器件(□):Attiny11/12/15/22; 90 条指令=□+89 条基本指令
- (3) 118 条指令器件(◇):AT90S2313/2323/2343/2333,/4414/4433/4434/8515/90S8534/8535
;118 条指令=◇+ 90 条;

118 条指令分类：

数据传送指令 (31)	{	寄存器传送(4): MOV, LD, SER, LDS, X 寄存器(3): LD Rd, X; LD Rd, X+; LD Rd, -X Y 寄存器(4): LD Rd, Y; LD Rd, Y+; LD Rd, -Y; LDD Rd, Y+q Z 寄存器(4): LD Rd, Z; LD Rd, Z+; LD Rd, -Z; LDD Rd, Z+q X 寄存器(3): ST X, Rr; ST X+, Rr; ST -X, Rr Y 寄存器(4): ST Y, Rr; ST Y+, Rr; ST -Y, Rr; STD Y+q, Rr Z 寄存器(4): ST Z, Rr; ST Z+, Rr; ST -Z, Rr; STD Z+q, Rr 读存储器(1): LPM I/O 指令(2): IN, OUT 栈指令(2): PUSH, POP
算术指令与 (10)	{	加(4): ADD, ADC, ADIW, INC; 减(6): SUB, SUBI, SBC, SBCI, SBIW, DEC
逻辑运算指令 (12)	{	与(2): AND, ANDI, 或(3): OR, ORI, EOR 清 0 置 1: (7) COM, NEG, CBR, SBR, CLR, TST, SER,
转移调用指令 (34)	{	转移指令: (2) RJMP, IJMP (JMP 转移) 调用指令: (2) RCALL, ICALL (CALL 调用) 返回: (2) RET, RETI (RET 返回) 比较指令: (3) CP, CPC, CPI (CP 比较) 分支指令: (5) CPSE, SBRC, SBRS, SBIC, SBIS, (S 为 1 跳, C 为 0 跳, R 寄存器, I 为 I/O) 基本条件分支指令: (2) BRBS, BRBC (S 为 1 转, C 为 0 转) (状态字为 I, T, H, S, V, N, Z, C) 派生指令: (18) BRIE, BRID; BRIS, BRIC; BRHS, BRHC; BRGE, BRGT; BRVS, BRVC; BRMI, BRPL; BREQ, BRNE; BRCS, BRCC; BRSH, BRLO;
位操作及其他 (31)	{	位 I/O 操作指令: (2) SBI, CBI (S 置 1, C 清 0) 状态寄存器指令: (2) BSET, BCLR (状态字为 I, T, H, S, V, N, Z, C) 派生指令: (16) BSET -- SEI, SET, SEH, SES SEV, SEN, SEZ, SEC (SE 置 1 某位) BCLR -- CLI, CLT, CLH, CLS, CLV, CLN, CLZ, CLC (CL 清 0 某位) 位传送: (2) BST, BLD, 移位指令: (6) LSL, LSR, ASR, ROL, ROR, SWAP (LS 右移, R0 左移, AS 算术右移, R 寄存器) 其他指令: (3) NOP, SLEEP, WDR

AVR 器件(指令速查表) 118 条指令器件

AT90S2313/2323/2343/2333 ,AT90S4414/4433/4434/8515,AT90S8534/8535

算术和逻辑指令(22)		☐BRCC k	C=0 清零转	位指令和位测试指令(31)	
ADD Rd,Rr	加法	☐BRSH k	C=1 无符号数≥转	SBI P,b	置位 I/O 位
ADC Rd,Rr	带进位加	☐BRLO k	C=0 无符号数小于转	CBI P,b	清零 I/O 位
◇ ADIW Rdl,K	加立即数	☐BRMI k	N=1 负数转移	LSL Rd	左移
SUB Rd,Rr	减法	☐BRPL k	N=0 正数转移	LSR Rd	右移
SUBI Rd,Rr	减立即数	☐BRGE k	S=0(带符号数)≥转	ROL Rd	带进位左循环
SBC Rd,Rr	带借位减	☐BRLT k	S=1(带符号)小于转	ROR Rd	带进位右循环
SBCI Rd,K	带 C 减立即数	☐BRHS k	H=1 置位转移	ASR Rd	算术右移
◇ SBIW Rdl,K	字减立即数	☐BRHC k	H=0 清零转移	SWAP Rd	半字节交换
AND Rd,Rr	与	☐BRTS k	T=1 置位转移	☆BSET s	置位 SREG
ANDI Rd,K	与立即数	☐BRTC k	T=0 清零转移	☆BCLR s	清零 SREG
OR Rd,Rr	或	☐BRVS k	V=1 置位转移	BST Rr,b	Rr 的 b 位送 T
ORI Rd,K	或立即数	☐BRVC k	V=0 清零转移	BLD Rd	T 送 Rr 的 b 位
EOR Rd,Rr	异或	☐BRIE k	I=1 中断位置位转移	☐SEC	置位 C
COM Rd	取反	☐BRID k	I=0 中断位清零转移	☐CLC	清零 C
NEG Rd	取补	数据传送指令(31)		☐SEN	置位 N
SBR Rd,K	寄存器位置位	MOV Rd,Rr	寄存器传送	☐CLN	清零 N
CBR Rd,K	寄存器位清零	◇ LDI Rd,K	装入立即数	☐SEZ	置位 Z
INC Rd	加 1	◇ LD Rd, X	X 间接取数	☐CLZ	清零 Z
DEC Rd	减 1	◇ LD Rd, X+	X 间接取数后+	☐SEI	置位 I
TST Rd	测试零或负	◇ LD Rd, -X	X 间接取数先-	☐CLI	清零 I
CLR Rd	寄存器清零	◇ LD Rd, Y	Y 间接取数	☐SES	置位 S
SER Rd	寄存器置 FF	◇ LD Rd, Y+	Y 间接取数后+	☐CLS	清零 S
条件转移指令(34)		◇ LD Rd, -Y	Y 间接取数先-	☐SEV	置位 V
RJMP k	相对转移	◇ LDD Rd, Y+q	Y 间接取数+q	☐CLV	清零 V
◇ IJMP	间接转移(Z)	LD Rd, Z	Z 间接取数	☐SET	置位 T
RCALL k	相对调用	◇ LD Rd, Z+	Z 间接取数后+	☐CLT	清零 T
◇ ICALL	间接调用(Z)	◇ LD Rd, -Z	Z 间接取数先-	☐SEH	置位 H
RET	子程序返回	◇ LDD Rd, Z+q	Z 间接取数+q	☐CLH	清零 H
RETI	中断返回	◇ LDS Rd,K	从 SRAM 装入	NOP	空操作
CPSE Rd,Rr	比较相等跳行	◇ ST X, Rr	X 间接存数	SLEEP	休眠指令
CP Rd,Rr	比较	◇ ST X+, Rr	X 间接存数后+	WDR	看门狗复位
CPC Rd,Rr	带进位比较	◇ ST -X, Rr	X 间接存数先-	90 条指令为 Attiny11/12/15/22= ☐+89 条基本指令是 AT90S1200	
CPI Rd,K	与立即数比较	◇ ST Y, Rr	Y 间接存数		
SBRC Rr,b	位清零跳行	◇ ST Y+, Rr	Y 间接存数后+		
SBRS Rr,b	位置位跳行	◇ ST -Y, Rr	Y 间接存数先-		
SBIC P,b	I/O 位清零跳行	◇ STD Y+q, Rr	Y 间接存数+q	118 条指令器件= ◇+ 90 条指令器件	
SBIS P,b	I/O 位置位跳行	ST Z, Rr	Z 间接存数		
☆BRBS s,k	SREG 位置位转	◇ ST Z+, Rr	Z 间接存数后+		
☆BRBC s,k	SREG 位清零转	◇ ST -Z, Rr	Z 间接存数先-		
☐BREK k	Z=1 相等转移	◇ STD Z+q, Rr	Z 间接存数+q	☆ 基本指令 ☐ 派生出来的指令	
☐BRNE k	Z=0 不相等转移	◇ STS k, Rr	数据送 SRAM		
☐BRCS k	C=1 置位转	☐ LPM	从程序区取数		
程序状态字 SREG		IN Rd,P	从 I/O 口取数		
7 6 5 4 3 2 1 0		OUT P, Rdr	存数 I/O 口		
I T H S V N Z C		PUSH Rr	压栈		
		POP Rd,	出栈		

AVR 单片机指令系统简介

4.4 算术和逻辑指令

4.4.1 加法指令

1. ADD 不带进位加法

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：ADD Rd,Rr $Rd \leftarrow Rd + Rr$ $0 \leq d \leq 31, 0 \leq r \leq 31$

2. ADC 带进位加法

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：ADC Rd,Rr $Rd \leftarrow Rd + Rr + C$ $0 \leq d \leq 31, 0 \leq r \leq 31$

3. ADIW 立即数据加法（字）

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：ADIW RdI, K $Rdh:RdI \leftarrow Rdh:RdI + K$ $d \in \{24, 26, 28, 30\}$, UJ

4. INC 加 1 指令

说明：寄存器 Rd 的内容加 1，结果送目的寄存器 Rd 中。

操作：INC Rd $Rd \leftarrow Rd + 1$ $0 \leq d \leq 31$

4.4.2 减法指令

1. SUB 不带进位减法

说明：两个寄存器相减，结果送目的寄存器 Rd 中。

操作：SUB Rd,Rr $Rd \leftarrow Rd - Rr$ $0 \leq d \leq 31, 0 \leq r \leq 31$

2. SUBI 立即数减法（字节）

说明：一个寄存器和常数相减，结果送目的寄存器 Rd。

操作：SUBI Rd,K $Rd \leftarrow Rd - K$ $16 \leq d \leq 31, 0 \leq k \leq 255$

3. SBC 带进位减法

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。

4. 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：SBCI Rd K $Rd \leftarrow Rd - K - C$ $16 \leq d \leq 31, 0 \leq K \leq 255$

5. SBIW 立即数减法（字）

说明：双寄存器与立即数（0~63）减，结果送双寄存器。

操作：SBIW RdI, K $Rdh : RdI \leftarrow Rdh:RdI - K$ $dI \in \{24, 26, 28, 30\}, 0 \leq K \leq 63$

6. DEC 减 1 指令

说明：寄存器 Rd 的内容减 1，结果送目的寄存器 Rd 中。

操作：DEC Rd $Rd \leftarrow Rd - 1$ $0 \leq d \leq 31$

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 COM $Rd \leftarrow \$FF - Rd$ $0 \leq d \leq 31$

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: $\text{NEG Rd } R \leftarrow \$00 - R_d \quad 0 \leq d \leq 31$

4.4.6 比较指令

1. CP 寄存器比较

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。

操作: $\text{CP Rd Rr } R_d - R_r \quad 0 \leq d \leq 31, 0 \leq r \leq 31$

2. CPC 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。

操作: $\text{CPC Rd Rr } R_d - R_r - C \quad 0 \leq d \leq 31, 0 \leq r \leq 31$

3. CPI 立即数比较

说明: 该指令完成寄存器 Rd 和常数的比较操作。

操作: $\text{CPI Rd, K } R_d - K \quad 16 \leq d \leq 31, 0 \leq K \leq 255$

4.4.7 逻辑与指令

1. AND 寄存器逻辑与;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。

操作: $\text{AND Rd, Rr } R_d \leftarrow R_d \cdot R_r \quad 0 \leq d \leq 31 \quad 0 \leq r \leq 31;$

2. ANDI 带立即数与;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。

操作: $\text{ANDI Rd, K } R_d \leftarrow R_d \cdot K \quad 16 \leq d \leq 31, 0 \leq K \leq 255;$

3. CBR 清除寄存器位

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: $\text{CBR Rd, K } R_d \leftarrow R_d \cdot (\$FF - K) \quad 16 \leq d \leq 31, 0 \leq K \leq 255$

4. 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: $\text{TST Rd } R_d \leftarrow R_d \quad 0 \leq d \leq 31$

4.4.8 逻辑或指令

1. OR 寄存器逻辑或;有 1 即 1,全 0 为 0

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

操作: $\text{OR Rd, Rr } R_d \leftarrow R_d \vee R_r \quad 0 \leq d \leq 31, 0 \leq r \leq 31$

2. ORI 带立即数或 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: $\text{ORI Rd, K } R_d \leftarrow R_d \vee K \quad 16 \leq d \leq 31, 0 \leq K \leq 255$

3. 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: $\text{SBR Rd, K } R_d \leftarrow R_d \vee K \quad 16 \leq d \leq 31, 0 \leq K \leq 255$

4. 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: $\text{SER Rd } R_d \leftarrow \$FF \quad 16 \leq d \leq 31$

4.4.9 逻辑异或指令

1. EOR 寄存器异或 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)

操作: $\text{EOR Rd, Rr } R_d \leftarrow R_d \oplus R_r \quad 0 \leq d \leq 31, 0 \leq r \leq 31$

2. 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：CLR Rd $Rd \leftarrow Rd \oplus Rd$ $0 \leq d \leq 31$

4.5 转移指令

4.5.1 无条件转移指令

1. RJMP 相对跳转

说明：相对跳转到 PC - 2K 和 PC + 2K (字) 范围内的地址。在汇编程序中，标号用于替代相对操作。

操作：RJMP k $PC \leftarrow (PC + 1) + k$ $-2K \leq k \leq 2K$

2. IJMP 间接跳转

说明：间接跳转到由寄存器区中的 Z (16 位) 指针寄存器指向的地址。Z 指针寄存器是 16 位宽，允许在当前程序存储器空间 64K 字 (128K 字节) 内跳转。

操作：IJMP $PC \leftarrow Z(15-0)$ $PC(15-0) \leftarrow Z(15-0)$

3. JMP 长跳转

说明：在整个程序存储空间 4M (字) 内跳转，见 RJMP。

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：JMP k $PC \leftarrow k$ $0 \leq k \leq 4M$

4.5.2 条件转移指令

一、测试条件符合转移指令

1. BRBS 状态寄存器中位置位转移

说明：条件相对转移，测试 SREG 的某一位，如果该位被置位，则相对 PC 值转移。

操作：BRBS S, k If SREG(S)=1 then $PC \leftarrow (PC + 1) + k$, else $PC \leftarrow PC + 1$
 $0 \leq S \leq 7$ $-64 \leq k \leq +63$ $PC \leftarrow (PC + 1) + k$

2. BRBC 状态寄存器中位清零转移

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。

操作：BRBC S, k If SREG(S)=0 then $PC \leftarrow (PC + 1) + k$, else $PC \leftarrow PC + 1$
 $0 \leq S \leq 7$, $-64 \leq k \leq +63$ $PC \leftarrow (PC + 1) + k$

3. 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。

操作：BREQ k If $Rd = Rr (Z=1)$ then $PC \leftarrow (PC + 1) + k$, $PC \leftarrow PC + 1$
 $-64 \leq k \leq +63$ $PC \leftarrow (PC + 1) + k$

4. 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。

操作，BRNE k If $Rd \neq Rr (Z=0)$ then $PC \leftarrow (PC + 1) + k$, else $PC \leftarrow PC + 1$ $-64 \leq k \leq +63$

5. BRCS 进位 C 标志位置位转移

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。

操作：BRCS k If C=1 then $PC \leftarrow (PC + 1) + k$, else $PC \leftarrow PC + 1$ $-64 \leq k \leq +63$

6. BRCC 进位 C 标志位清除转移

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。

- 操作: $\text{BRCC } k \text{ If } C = 0 \text{ then } PC \leftarrow (PC+1)+k, \text{ else } PC \leftarrow PC+1 \quad -64 \leq k \leq +63$
7. BRSH (无符号) 大于或等于转移
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。
操作: $\text{BRSH } k \text{ If } Rd \geq Rr (C=0) \text{ then } PC \leftarrow (PC+1)+k, \text{ else } PC \leftarrow PC+1 \quad -64 \leq k \leq +63$
8. BRLO (无符号) 小于低于转移
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。
操作: $\text{BRLO } k \text{ If } Rd < Rr (C=1) \text{ then } PC \leftarrow (PC+1)+k, \text{ else } PC \leftarrow PC+1 \quad -64 \leq k \leq +63$
9. 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。
操作: $\text{BRMI } k \text{ If } N = 1 \text{ then } PC \leftarrow (PC+1)+k, \text{ else } PC \leftarrow PC+1 \quad -64 \leq k \leq +63$
10. 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。
操作: $\text{BRPL } k \text{ If } N = 0 \text{ then } PC \leftarrow (PC+1)+k, \text{ else } PC \leftarrow PC+1 \quad -64 \leq k \leq +63$
11. 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。
操作: $\text{BRGE } k \text{ If } Rd \geq Rr (N \vee S = 0) \text{ then } PC \leftarrow (PC+1)+k, \text{ else } PC \leftarrow PC+1 \quad -64 \leq k \leq +63$
12. 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。
操作: $\text{BRLT } k \text{ If } Rd < Rr (N \vee S = 1) \text{ then } PC \leftarrow (PC+1)+k, \text{ else } PC \leftarrow PC+1 \quad -64 \leq k \leq +63$
13. 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。
操作: $\text{BRHS } k \text{ If } H = 1 \text{ then } PC \leftarrow (PC+1)+k, \text{ else } PC \leftarrow PC+1 \quad -64 \leq k \leq +63$
14. BRHC 半进位标志清零转移
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。
操作: $\text{BRHC } k \text{ If } H = 0 \text{ then } PC \leftarrow (PC+1)+k, \text{ else } PC \leftarrow PC+1 \quad -64 \leq k \leq +63$
15. BRTS T 标志置位转移
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。
操作: $\text{BRTS } k \text{ If } T = 1 \text{ then } PC \leftarrow (PC+1)+k, \text{ else } PC \leftarrow PC+1 \quad -64 \leq k \leq +63$
16. BRTC T 标志清零转移
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。
操作: $\text{BRTC } k \text{ If } T = 0 \text{ then } PC \leftarrow (PC+1)+k, \text{ else } PC \leftarrow PC+1 \quad -64 \leq k \leq +63$
17. BRVS 溢出标志置位转移
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。
操作: $\text{BRVS } k, \text{ If } V = 1 \text{ then } PC \leftarrow (PC+1)+k, \text{ else } PC \leftarrow PC+1 \quad -64 \leq k \leq +63$
18. BRVC 溢出标志清零转移
操作: $\text{BRVC } k \text{ If } V = 0 \text{ then } PC \leftarrow (PC+1)+k, \text{ else } PC \leftarrow PC+1 \quad -64 \leq k \leq +63$
19. BRIE—全局中断标志触发转移
说明: 条件相对转移, 测试全局中断标志 (I), 如果 I 被置位 1, 则相对 PC 值转移。
操作: $\text{BRIE } k \text{ If } I = 1 \text{ then } PC \leftarrow (PC+1)+k, \text{ else } PC \leftarrow PC+1 \quad -64 \leq k \leq +63$
20. BRID 全局中断标志禁止转移
说明: 条件相对转移。测试全局中断标志 (I), 如果 I 被清零, 则相对 PC 但转移。
操作: $\text{BRID } k \text{ If } I = 0 \text{ then } PC \leftarrow (PC+1)+k, \text{ else } PC \leftarrow PC+1 \quad -64 \leq k \leq +63$

二、测试条件符合跳行转移指令

21. 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：CPSE Rd, Rr If $Rd=Rr$ then $PC\leftarrow PC+2$ (or 3) else $PC\leftarrow PC+1$

$0\leq d\leq 31, 0\leq r\leq 31$

22. SBRC—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：SBRC Rr, b If $Rr(b)=0$ then $PC\leftarrow PC+2$ (or 3) else $PC\leftarrow PC+1$

$0\leq r\leq 31, 0\leq b\leq 7$

23. SBRS—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：SBRS Rr, b If $Rr(b)=1$ then $PC\leftarrow PC+2$ (or 3) else $PC\leftarrow PC+1$

$0\leq r\leq 31, 0\leq b\leq 7$

24. SBIC—I/O 寄存器位清零跳行

说明：该指令测试 I/O 寄存器某位，如果该位被清零，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：SBIC P, b If I/O, $b=0$ then $PC\leftarrow PC+2$ (or 3) else $PC\leftarrow PC+1$

$0\leq P\leq 31, 0\leq b\leq 7$

25. SBIS—I/O 寄存器位置位跳行

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：SBIS p, b If I/O, $b=1$ then $PC\leftarrow PC+2$ (or 3) else $PC\leftarrow PC+1$

$0\leq P\leq 31, 0\leq b\leq 7$

三、调用和返回指令

26. RCALL—相对调用子程序

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：RCALL k $PC\leftarrow (PC+1)+k$ $-2K\leq k\leq 2K$

27. ICALL—间接调用子程序

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：ICALL $PC(15-0)\leftarrow Z(15-0)$

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

28. 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：CALL k $PC\leftarrow k$ $0\leq k\leq 64K$

CALL k $PC\leftarrow k$ $0\leq k\leq 4M$

29. RET—从子程序返回

说明：从子程序返回。返回地址从堆栈中弹出。

操作：RET $PC(15-0)\leftarrow STACK$

RET $PC(21-0)\leftarrow STACK$

30. RETI—从中断程序返回

说明：从中断程序中返回。返回地址从堆栈中弹出，且全局中断标志被置位。

注意：1. 主程序应跳过中断区，防止修改补充中断程序带来麻烦；

2. 不用的中断入口地址写上 RETI - 中断返回，有抗干扰作用；

操作: RETI PC (15-0) ←STACK
 RETI PC (21-0) ←STACK

4.6 数据传送指令

4.6.1 直接数据传送指令

1. MOV 寄存器拷贝数据

说明: 该指令将一个寄存器拷贝到另一个寄存器。源寄存器 Rr 的内容不改变, 而目的寄存器 Rd 拷贝了 Rr 的内容。

操作: MOV Rd Rr Rd ← Rr 0 ≤ d ≤ 31, 0 ≤ r ≤ 31

2. LDS SRAM 数据直接送寄存器

说明: 把 SRAM 中 1 个字节装入到寄存器。必须提供一个 16 位地址。

操作: LDS Rd k Rd ← (k) 0 ≤ d ≤ 31, 0 ≤ k ≤ 65535

3. STS 寄存器数据直接送 SRAM

说明: 将寄存器的内容直接存储到 SRAM。必须提供一个 16 位的地址。

操作: STS k, Rr (k) ← Rr 0 ≤ r ≤ 31, 0 ≤ k ≤ 65535

4. LDI—立即数送寄存器

说明: 装入一个 8 位立即数到寄存器 R16~R31 中。

操作: LDI Rd K Rd ← K 16 ≤ d ≤ 31, 0 ≤ K ≤ 255

4.6.2 间接数据传送指令

一、使用 X 寄存器间接传送数据

1. 使用变址 X 间接将 SRAM 中内容送入到寄存器

LD—使用变址 X 间接将 SRAM 中内容送入到寄存器

说明: 从 SRAM 中间送入一个字节到寄存器, SRAM 中的位置由寄存器区中的 X (16 位) 指针寄存器指出。存储器访问被限制在当前 64K 字节的 SRAM 页中。为访问另外 SRAM 页, 则 I/O 范围内的寄存器 RAMPX 需改变。在指令执行中, X 指针寄存器值要么不改变, 要么就加 1 或减 1 操作。使用 X 指针寄存器的这些特性, 特别适合于访问矩阵、表和堆栈指针等。

操作: LD Rd, X Rd ← (X) ; 送数, X 指针寄存器值不改变
 LD Rd, X+ Rd ← (X) X ← X+1 ; 先送数, 后 X 指针寄存器值加 1
 LD Rd, -X X ← X-1 Rd ← (X) ; 先 X 指针寄存器值减 1, 后送数

2. 使用变址 X 间接将寄存器内容传送到 SRAM

ST—使用变址 X 间接将寄存器内容传送到 SRAM

说明: 间接将寄存器的一个字节传送到 SRAM。SRAM 的位置由寄存器区中的 X (16 位) 指针寄存器指出。存储器访问被限制在当前 64K 字节的 SRAM 页中。为访问另外 SRAM 页, 则 I/O 范围内的寄存器 RAMPX 将被修改。在操作之后, X 指针寄存器要么不改变, 要么是加 1 或减 1。使用 X 指针寄存器的这些特性, 特别适合作堆栈指针。

操作: ST X, Rr (X) ← Rr 0 ≤ d ≤ 31 送数, X 指针不改变
 ST X+, Rr X ← Rr 0 ≤ d ≤ 31 先送数, 后 X 指针加 1
 ST -X, Rr X ← X-1 (X) ← Rr 0 ≤ d ≤ 31 先 X 指针减 1, 后送数

二、使用 Y 寄存器间接传送数据

3. 使用变址 Y 间接将 SRAM 中的内容传送到寄存器

LD (LDD)—使用变址 Y 间接将 SRAM 中的内容传送到寄存器

说明: 带或不带偏移间接从 SRAM 中传送一个字节到寄存器, SRAM 中的位置由寄存器区中的 Y (16 位) 指针寄存器指出。存储器访问被限制在当前 64K 字节的 SRAM 页中。为访问另外 SRAM 页, 则 I/O 范围内的寄存器 RAMPY 需改变。在指令执行后, Y 指针寄存器值要么不改变, 要么就加 1 或减 1 操作。使用 Y 指针寄存器的这些特性, 特别适合于访问矩阵、表和堆栈指针等。

操作: LD Rd, Y Rd ← (Y) 0 ≤ d ≤ 31 送数, Y 指针不改变
 LD Rd, Y+ Rd ← (Y) 0 ≤ d ≤ 31 先送数, 后 Y 指针加 1

LD Rd, -Y $Y \leftarrow Y-1$ $Y \leftarrow Y+1$ $0 \leq d \leq 31$ 先 Y 指针减 1, 后送数
 LDD Rd Y+q $Rd \leftarrow (Y+q)$ $Rd \leftarrow (Y)$ $0 \leq d \leq 31$, 先 Y 指针加 q, 后送数
 $0 \leq q \leq 63$ 执行后 Y 指针(Y 不含 q)不变

4. 使用变址 Y 间接将寄存器内容传送到 SRAM

ST (STD) —使用变址 Y 间接将寄存器内容传送到 SRAM

说明: 间接将带或不带偏移的寄存器的一个字节传送到 SRAM。SRAM 的位置由寄存器区中的 Y (16 位) 指针寄存器指出。存储器访问被限制在当前 64K 字节的 SRAM 页。为访问另外 SRAM 页, 则 I/O 范围的寄存器 RAMPY 将被修改。在操作之后, Y 指针寄存器要么不改变, 要么是加 1 或减 1。使用 Y 指针寄存器的这些特性, 特别适合作堆栈指针。

操作: ST Y, Rr $(Y) \leftarrow Rr$ $0 \leq d \leq 31$ 送数, Y 指针不改变
 ST Y+, Rr $(Y) \leftarrow Rr$ $Y \leftarrow Y+1$ $0 \leq d \leq 31$ 先送数, 后 Y 指针加 1
 ST -Y, Rr $Y \leftarrow Y-1$ $(Y) \leftarrow Rr$ $0 \leq d \leq 31$ 先 Y 指针减 1, 后送数
 STD Y+q, Rr $(Y+q) \leftarrow Rr$ $0 \leq d \leq 31$, 先 Y 指针加 q, 后送数
 $0 \leq q \leq 63$ 执行后 Y 指针(Y 不含 q)不变

三、使用 Z 寄存器间接传送数据

5. 使用变址 Z 间接将 SRAM 中的内容传送到寄存器

LD (LDD) —使用变址 Z 间接将 SRAM 中的内容传送到寄存器

说明: 带或不带偏移间接从 SRAM 中传送一个字节到寄存器, SRAM 中的位置由寄存器区中的 Z (16 位) 指针寄存器指出。存储器访问被限制在当前 64K 字节的 SRAM 页中。为访问另外 SRAM 页, 则 I/O 范围内的寄存器 RAMPZ 需改变。在指令执行后, Z 指针寄存器值要么不改变, 要么就加 1 或减 1 操作。使用 Z 指针寄存器的这些特性, 特别适合于堆栈指针, 因为 Z 指针寄存器能用于直接子程序调用, 直接跳转和查表。Z 指针寄存器用作为专用堆栈指针要比 X、Y 指针方便。

用 Z 指针在程序存储器中查表, 可参见 LPM 指令。

操作: LD Rd, Z $Rd \leftarrow (Z)$ $0 \leq d \leq 31$ 送数, Z 指针不改变
 LD Rd Z+ $Rd \leftarrow (Z)$ $0 \leq d \leq 31$ 先送数, 后 Z 指针加 1
 LD Rd, -Z $Z \leftarrow Z-1$ $0 \leq d \leq 31$ 先 Z 指针减 1, 后送数
 LDD Rd Z+q $Rd \leftarrow (Z+q)$ $0 \leq d \leq 31$, 先 Z 指针加 q, 后送数
 $0 \leq q \leq 63$ 执行后 Z 指针(Z 不含 q)不变

6. 使用变址 Z 间接将寄存器内容传送到 SRAM

ST (STD) —使用变址 Z 间接将寄存器内容传送到 SRAM

说明: 间接将带或不带偏移的寄存器的一个字节传送到 SRAM。SRAM 的位置由寄存器区中的 Z (16 位) 指针寄存器指出。存储器访问被限制在当前 64K 字节的 SRAM 页。为访问另外 SRAM 页, 则 I/O 范围的寄存器 RAMPZ 将被修改。在操作之后, Z 指针寄存器要么不改变, 要么是加 1 或减 1。使用 Z 指针寄存器的这些特性, 特别适合作堆栈指针。因为 Z 指针寄存器能适用于间接子程序调用, 间接跳转和查表, 所以 Z 指针寄存器像一个专用堆栈指针, 用起来比 X 和 Y 指针更方便。

操作: ST Z, Rr $(Z) \leftarrow Rr$ $0 \leq d \leq 31$ 送数, Z 指针不改变
 ST Z+, Rr $(Z) \leftarrow Rr$ $Z \leftarrow Z+1$ $0 \leq d \leq 31$ 先送数, 后 Z 指针加 1
 ST -Z, Rr $Z \leftarrow Z-1$ $(Z) \leftarrow Rr$ $0 \leq d \leq 31$ 先 Z 指针减 1, 后送数
 STD Z+q, Rr $(Z+q) \leftarrow Rr$ $0 \leq d \leq 31$, 先 Z 指针加 q, 后送数
 $0 \leq q \leq 63$ 执行后 Z 指针(Z 不含 q)不变

3.6.3 从程序存储器直接取数据指令

1. LPM—装入程序存储器

说明: 将 Z 寄存器指向的一个字节传送到寄存器 0 (R0)。该指令使 100 % 空间有效, 常量初始化或常数取数特别有用。程序存储器被编为 16 位字, Z (16 位) 指针的最低位 (LSB) 选择为 0 是低字节, 选择为 1 是高字节。该指令能寻址程序存储器第一个 64K 字节 (32 字)。

操作: LPM R0←(Z)

4.6.4 I/O 口数据传送

1. IN—I/O 口数据传送到寄存器

说明: 将 I/O 空间 (口, 定时器, 配置寄存器等) 的数据传送到寄存器区中的寄存器 Rd 中。

操作: IN Rd P Rd←P $0 \leq d \leq 31$, $0 \leq P \leq 63$

2. OUT--寄存器数据送 I/O 口

说明: 将寄存器区中寄存器 Rr 的数据传送到 I/O 空间 (口、定时器、配置寄存器等)。

操作: OUT P, Rr P←Rr $0 \leq r \leq 31$, $0 \leq P \leq 63$

4.6.5 堆栈操作指令

AVR 单片机的特殊功能寄存器中有一个堆栈指针 SP。它指出栈顶的位置, 在指令系统中有两条用于数据传送的栈操作指令。

1. PUSH—压寄存器到堆栈, 进栈指令

说明: 该指令存储寄存器 Rr 的内容到堆栈。

操作: PUSH Rr STACK←Rr $0 \leq d \leq 31$

2. POP—堆栈弹出到寄存器, 出栈指令

说明: 该指令将堆栈中的字节装入到寄存器 Rd 中。

操作: POP Rd Rd←STACK $0 \leq d \leq 31$

4.7 位指令和位测试指令

4.7.1 带进位逻辑操作指令

1. LSL—逻辑左移

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: LSL Rd $0 \leq d \leq 31$

2. 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: LSR Rd $0 \leq d \leq 31$

3. 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: ROL Rd $0 \leq d \leq 31$

4. ROR—通过进位右循环

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

操作: ROR Rd $0 \leq d \leq 31$

5. 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: ASR Rd $0 \leq d \leq 31$

6. 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: SWAP Rd $0 \leq d \leq 31$ R(7~4)←R(3~0)←R(3~0)←Rd(7~4)

4.7.2 位变量传送指令

1. 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: BST Rd b T←Rd(b) $0 \leq d \leq 31, 0 \leq b \leq 7$

2. BLD — SREG 中的 T 标志装入寄存器中的某一位
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: $\text{BLD Rd,d Rd}(b) \leftarrow T \quad 0 \leq d \leq 31, 0 \leq b \leq 7$
- 4.7.3 位变量修改指令
 1. 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: $\text{BSET s SREG}(S) \leftarrow 1 \quad 0 \leq s \leq 7$
 2. BCLR—清状态寄存器的位
说明: 清零 SREG 状态寄存器中的一个标志位。
操作: $\text{BCLR s SREG}(S) \leftarrow 0 \quad 0 \leq s \leq 7$
 3. 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: $\text{SBIP, b I/O}(P, b) \leftarrow 1 \quad 0 \leq P \leq 31, 0 \leq b \leq 7$
 4. 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: $\text{CBI P, b I/O}(P, b) \leftarrow 0 \quad 0 \leq P \leq 31, 0 \leq b \leq 7$
 5. SEC—置进位标志
说明: 置位 SREG (状态寄存器) 中的进位标志 (C)。
操作: $\text{SEC C} \leftarrow 1$
 6. 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: $\text{CLC C} \leftarrow 0$
 7. 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: $\text{SEN N} \leftarrow 1$
 8. CLN—清零负数标志
说明: 清零 SREG (状态寄存器) 中的负数标志 (N)。
操作: $\text{CLN N} \leftarrow 0$
 9. SEZ—置位零标志
说明: 置位 SREG (状态寄存器) 中的零标志 (Z)。
操作: $\text{SEZ Z} \leftarrow 1$
 10. CLZ—清零零标志
说明: 清零 SREG (状态寄存器) 中的零标志 (Z)。
操作: $\text{CLZ Z} \leftarrow 0$
 11. SEI—置位全局中断标志
说明: 置位 SREG (状态寄存器) 中的全局中断标志 (I)。
操作: $\text{SEI I} \leftarrow 1$
 12. 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: $\text{CLI I} \leftarrow 0$
 13. SES—置位符号 S 标志
说明: 置位 SREG (状态寄存器) 中的符号标志 (S)。
操作: $\text{SES s} \leftarrow 1$
 14. CLS—清零符号 S 标志
说明: 清零 SREG (状态寄存器) 中的符号标志 (S)。

- 操作: CLS $S \leftarrow 0$
15. SEV—置溢出标志位
说明: 置位 SREG 状态寄存器) 中的溢出标志 (V)。
操作: SEV $V \leftarrow 1$
16. CLV—清溢出标志位
说明: 清零 SREG (状态寄存器) 中的溢出标志 (V)。
操作: CLV $V \leftarrow 0$
17. SET 置 T 标志位
说明: 置位 SREG 状态寄存器) 中的 T 标志。
操作: SET $T \leftarrow 1$
18. CLT—清 T 标志位
说明: 清零 SREG (状态寄存器) 中的 T 标志。
操作: CLT $T \leftarrow 0$
19. SEH—置半进位标志
说明: 置位 SREG (状态寄存器) 中的半进位标志 (H)。
操作: SEH $H \leftarrow 1$
20. CLH—清半进位标志
说明: 清零 SREG (状态寄存器) 中的半进位标志 (H)。
操作: CLH $H \leftarrow 0$
- 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: NOP No
2. SLEEP—休眠指令
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: SLEEP
3. 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: WDR WD timer restart.

广州市天河双龙电子有限公司 <http://WWW.SL.COM.CN>

广州双龙: 广州天河路 561 号新赛格电子城 331 室(510630)

电话: 020-87578852、87505012 传真: 分机 620

北京双龙: 北京海淀知春路 132 号中发大厦 616 室(100086)

电话: 010-82623551、62653785 传真: 010-82623550

上海双龙: 上海北京东路 668 号科技京城东楼 12H2 室(200001)

电话: 021-53081501、53081502 传真: 分机 213

广州市天河双龙电子有限公司 <http://WWW.SL.COM.CN>