

RISC-V 汇编语言

类别	指令	示例	含义	注解
算术运算	加	add x5, x6, x7	$x5 = x6 + x7$	三寄存器操作数；加
	减	sub x5, x6, x7	$x5 = x6 - x7$	三寄存器操作数；减
	立即数加	addi x5, x6, 20	$x5 = x6 + 20$	用于加常数
数据传输	取双字	ld x5, 40(x6)	$x5 = \text{Memory}[x6 + 40]$	从存储器取双字到寄存器
	存双字	sd x5, 40(x6)	$\text{Memory}[x6 + 40] = x5$	从寄存器存双字到存储器
	取字	lw x5, 40(x6)	$x5 = \text{Memory}[x6 + 40]$	从存储器取字到寄存器
	取字（无符号数）	lwu x5, 40(x6)	$x5 = \text{Memory}[x6 + 40]$	从存储器取无符号字到寄存器
	存字	sw x5, 40(x6)	$\text{Memory}[x6 + 40] = x5$	从寄存器存字到存储器
	取半字	lh x5, 40(x6)	$x5 = \text{Memory}[x6 + 40]$	从存储器取半字到寄存器
	取半字（无符号数）	lhu x5, 40(x6)	$x5 = \text{Memory}[x6 + 40]$	从存储器取无符号半字到寄存器
数据传输	存半字	sh x5, 40(x6)	$\text{Memory}[x6 + 40] = x5$	从寄存器存半字到存储器
	取字节	lb x5, 40(x6)	$x5 = \text{Memory}[x6 + 40]$	从存储器取字节到寄存器
	取字节（无符号数）	lbu x5, 40(x6)	$x5 = \text{Memory}[x6 + 40]$	从存储器取无符号字节到寄存器
	存字节	sb x5, 40(x6)	$\text{Memory}[x6 + 40] = x5$	从寄存器存字节到存储器
	取保留字	lr.d x5, (x6)	$x5 = \text{Memory}[x6]$	取；原子交换的前半部分
	存条件字	sc.d x7, x5, (x6)	$\text{Memory}[x6] = x5; x7 = 0/1$	存；原子交换的后半部分
	取立即数高位	lui x5, 0x12345	$x5 = 0x12345000$	取左移12位后的20位立即数
逻辑运算	与	and x5, x6, x7	$x5 = x6 \& x7$	三寄存器操作数；按位与
	或	or x5, x6, x8	$x5 = x6 x8$	三寄存器操作数；按位或
	异或	xor x5, x6, x9	$x5 = x6 \wedge x9$	三寄存器操作数；按位异或
	与立即数	andi x5, x6, 20	$x5 = x6 \& 20$	寄存器与常数按位与
	或立即数	ori x5, x6, 20	$x5 = x6 20$	寄存器与常数按位或
	异或立即数	xori x5, x6, 20	$x5 = x6 \wedge 20$	寄存器与常数按位异或
移位操作	逻辑左移	sll x5, x6, x7	$x5 = x6 \ll x7$	按寄存器给定位数左移
	逻辑右移	srl x5, x6, x7	$x5 = x6 \gg x7$	按寄存器给定位数右移
	算术右移	sra x5, x6, x7	$x5 = x6 \gg x7$	按寄存器给定位数算术右移
	逻辑左移立即数	slli x5, x6, 3	$x5 = x6 \ll 3$	根据立即数给定位数左移
	逻辑右移立即数	srl i x5, x6, 3	$x5 = x6 \gg 3$	根据立即数给定位数右移
	算术右移立即数	srai x5, x6, 3	$x5 = x6 \gg 3$	根据立即数给定位数算术右移
	相等即跳转	beq x5, x6, 100	if (x5 == x6) go to PC+100	若寄存器数值相等则跳转到PC相对地址
条件分支	不等即跳转	bne x5, x6, 100	if (x5 != x6) go to PC+100	若寄存器数值不等则跳转到PC相对地址
	小于即跳转	blt x5, x6, 100	if (x5 < x6) go to PC+100	若寄存器数值比较结果小于则跳转到PC相对地址
	大于等于即跳转	bge x5, x6, 100	if (x5 >= x6) go to PC+100	若寄存器数值比较结果大于或等于则跳转到PC相对地址
	小于即跳转（无符号）	bltu x5, x6, 100	if (x5 < x6) go to PC+100	若寄存器数值比较结果小于则跳转到PC相对地址（无符号）
	大于等于即跳转（无符号）	bgeu x5, x6, 100	if (x5 >= x6) go to PC+100	若寄存器数值比较结果大于或等于则跳转到PC相对地址（无符号）
	跳转-链接	jal x1, 100	$x1 = PC+4; \text{go to } PC+100$	用于PC相关的过程调用
	跳转-链接（寄存器地址）	jalr x1, 100(x5)	$x1 = PC+4; \text{go to } x5+100$	用于过程返回；非直接调用

U型指令LUI的应用

- LUI将长立即数写入目的寄存器的高20位，并清除低12位。

- LUI与ADDI一起可在寄存器中创建任何32位值

```
LUI    x10, 0x87654    # x10 = 0x87654000
ADDI   x10, x10, 0x321  # x10 = 0x87654321
```

U型指令LUI的应用

- 如何创建 0xDEADBEEF?

```
LUI    x10, 0xDEADB    # x10 = 0xDEADB000
ADDI   x10, x10, 0xEEF  # x10 = 0xDEADAEFF
```

- ADDI 立即数总是进行符号扩展，如果高位为1，将从高位的20位中减去1

```
LUI    x10, 0xDEADC    # x10 = 0xDEADC000
ADDI   x10, x10, 0xEEF  # x10 = 0xDEADBEEF
```

- 伪指令 li x10, 0xDEADBEEF # 创建两条指令

U型指令

- AUIPC（Add Upper Immediate to PC）

- 将长立即数值加到PC并写入目的寄存器
- 用于PC相对寻址

```
Label: AUIPC x10, 0    # 将Label地址放入x10
```

注意：RV 汇编语法比较简单，但需要熟练掌握（带偏移量的指令比如 ld, sd 等的括号要括在寄存器外），见黑书第二章 P44-P45