|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **MIPS 指令集(共31条）** | | | | | | | | | | | **助记符** | **指令格式** | | | | | | **示例** | **示例含义** | **操作及其解释** | | Bit # | 31..26 | 25..21 | 20..16 | 15..11 | 10..6 | 5..0 |  |  |  | | R-type | op | rs | rt | rd | shamt | func |  |  |  | | add | 000000 | rs | rt | rd | 00000 | 100000 | add $1,$2,$3 | $1=$2+$3 | rd <- rs + rt   ；其中rs＝$2，rt=$3, rd=$1 | | addu | 000000 | rs | rt | rd | 00000 | 100001 | addu $1,$2,$3 | $1=$2+$3 | rd <- rs + rt   ；其中rs＝$2，rt=$3, rd=$1,无符号数 | | sub | 000000 | rs | rt | rd | 00000 | 100010 | sub $1,$2,$3 | $1=$2-$3 | rd <- rs - rt   ；其中rs＝$2，rt=$3, rd=$1 | | subu | 000000 | rs | rt | rd | 00000 | 100011 | subu $1,$2,$3 | $1=$2-$3 | rd <- rs - rt   ；其中rs＝$2，rt=$3, rd=$1,无符号数 | | and | 000000 | rs | rt | rd | 00000 | 100100 | and $1,$2,$3 | $1=$2 & $3 | rd <- rs & rt   ；其中rs＝$2，rt=$3, rd=$1 | | or | 000000 | rs | rt | rd | 00000 | 100101 | or $1,$2,$3 | $1=$2 | $3 | rd <- rs | rt   ；其中rs＝$2，rt=$3, rd=$1 | | xor | 000000 | rs | rt | rd | 00000 | 100110 | xor $1,$2,$3 | $1=$2 ^ $3 | rd <- rs xor rt   ；其中rs＝$2，rt=$3, rd=$1(异或） | | nor | 000000 | rs | rt | rd | 00000 | 100111 | nor $1,$2,$3 | $1=~($2 | $3) | rd <- not(rs | rt)   ；其中rs＝$2，rt=$3, rd=$1(或非） | | slt | 000000 | rs | rt | rd | 00000 | 101010 | slt $1,$2,$3 | if($2<$3)    $1=1 else    $1=0 | if (rs < rt) rd=1 else rd=0 ；其中rs＝$2，rt=$3, rd=$1 | | sltu | 000000 | rs | rt | rd | 00000 | 101011 | sltu $1,$2,$3 | if($2<$3)    $1=1 else    $1=0 | if (rs < rt) rd=1 else rd=0 ；其中rs＝$2，rt=$3, rd=$1   (无符号数） | | sll | 000000 | 00000 | rt | rd | shamt | 000000 | sll $1,$2,10 | $1=$2<<10 | rd <- rt << shamt  ；shamt存放移位的位数，   也就是指令中的立即数，其中rt=$2, rd=$1 | | srl | 000000 | 00000 | rt | rd | shamt | 000010 | srl $1,$2,10 | $1=$2>>10 | rd <- rt >> shamt ；(logical) ，其中rt=$2, rd=$1 | | sra | 000000 | 00000 | rt | rd | shamt | 000011 | sra $1,$2,10 | $1=$2>>10 | rd <- rt >> shamt  ；(arithmetic) 注意符号位保留  其中rt=$2, rd=$1 | | sllv | 000000 | rs | rt | rd | 00000 | 000100 | sllv $1,$2,$3 | $1=$2<<$3 | rd <- rt << rs  ；其中rs＝$3，rt=$2, rd=$1 | | srlv | 000000 | rs | rt | rd | 00000 | 000110 | srlv $1,$2,$3 | $1=$2>>$3 | rd <- rt >> rs  ；(logical)其中rs＝$3，rt=$2, rd=$1 | | srav | 000000 | rs | rt | rd | 00000 | 000111 | srav $1,$2,$3 | $1=$2>>$3 | rd <- rt >> rs  ；(arithmetic) 注意符号位保留  其中rs＝$3，rt=$2, rd=$1 | | jr | 000000 | rs | 00000 | 00000 | 00000 | 001000 | jr $31 | goto $31 | PC <- rs | | jalr | 000000 | rs | 00000 | rd | 00000 | 001001 | jalr $1,$2 | goto $2  rd<-PC+4 | PC <- rs,rd<-PC+4 | | div | 000000 | rs | rt | 00000 | 00000 | 011010 | div $1,$2 | LO=$1/$2  HI=$1%$2 | LO<-rs/rt  HI<-rs%rt | | divu | 000000 | rs | rt | 00000 | 00000 | 011011 | divu $1,$2 | LO=$1/$2  HI=$1%$2 | LO<-rs/rt  HI<-rs%rt无符号 | | mult | 000000 | rs | rt | 00000 | 00000 | 011000 | mult $1,$2 | LO=$1\*$2[31:0]  HI=$1\*$2[63:32] | LO<-rs\*rt[31:0]  HI<-rs\*rt[63:32] | | multu | 000000 | rs | rt | 00000 | 00000 | 011001 | multu $1,$2 | LO=$1\*$2[31:0]  HI=$1\*$2[63:32] | LO<-rs\*rt[31:0]  HI<-rs\*rt[63:32]无符号 | | mfhi | 000000 | 00000 | 00000 | rd | 00000 | 010000 | mfhi $1 | $1<-HI | rd<-HI | | mflo | 000000 | 00000 | 00000 | rd | 00000 | 010010 | mflo $1 | $1<-LO | rd<-LO | | mthi | 000000 | rs | 00000 | 00000 | 00000 | 010001 | mthi $1 | HI<-$1 | HI<-rs | | mtlo | 000000 | rs | 00000 | 00000 | 00000 | 010011 | mtlo $1 | LO<-$1 | LO<-rs | | break | 000000 | code | | | | 001101 | break | 触发断点例外。 | SignalException(Breakpoint) | | syscall | 000000 | code | | | | 001100 | syscall | 触发系统调用例外。 | SignalException(SystemCall) | | I-type | op | rs | rt | imm | | |  |  |  | | addi | 001000 | rs | rt | imm | | | addi $1,$2,100 | $1=$2+100 | rt <- rs + (sign-extend)imm ；其中rt=$1,rs=$2 | | addiu | 001001 | rs | rt | imm | | | addiu $1,$2,100 | $1=$2+100 | rt <- rs + (zero-extend)imm ；其中rt=$1,rs=$2 | | andi | 001100 | rs | rt | imm | | | andi $1,$2,10 | $1=$2 & 10 | rt <- rs & (zero-extend)imm ；其中rt=$1,rs=$2 | | ori | 001101 | rs | rt | imm | | | andi $1,$2,10 | $1=$2 | 10 | rt <- rs | (zero-extend)imm ；其中rt=$1,rs=$2 | | xori | 001110 | rs | rt | imm | | | andi $1,$2,10 | $1=$2 ^ 10 | rt <- rs xor (zero-extend)imm ；其中rt=$1,rs=$2 | | lui | 001111 | 00000 | rt | imm | | | lui $1,100 | $1=100\*65536 | rt <- imm\*65536 ；将16位立即数放到目标寄存器高16位，目标寄存器的低16位填0 | | lw | 100011 | rs | rt | imm | | | lw $1,10($2) | $1=memory[$2  +10] | rt <- memory[rs + (sign-extend)imm] ；rt=$1,rs=$2 | | lb | 100000 | base | rt | imm | | | lb $1,10(00000) | vAddr=(sign\_extend)10+0;  $1=(sign\_extend) (byte)memory[vAddr]; | vAddr<-(sign\_extend)imm+base;  rt<-(sign\_extend) (byte)memory[vAddr]; | | lbu | 100100 | base | rt | imm | | | lbu $1,10(00000) | vAddr=(sign\_extend)10+0;  $1=(0\_extend) (byte)memory[vAddr]; | vAddr<-(sign\_extend)imm+base;  rt<-(0\_extend) (byte)memory[vAddr]; | | lh | 100001 | base | rt | imm | | | lh $1,10(00000) | vAddr=(sign\_extend)10+0;  if(vAddr[0]!=0)AddrEr;  $1=(sign\_extend) (2byte)memory[vAddr]; | vAddr<-(sign\_extend)imm+base;  if(vAddr[0]!=0)AddrEr;  rt<-(sign\_extend) (2byte)memory[vAddr]; | | lhu | 100101 | base | rt | imm | | | lhu $1,10(00000) | vAddr=(sign\_extend)10+0;  if(vAddr[0]!=0)AddrEr;  $1=(0\_extend) (2byte)memory[vAddr]; | vAddr<-(sign\_extend)imm+base;  if(vAddr[0]!=0)AddrEr;  rt<-(0\_extend) (2byte)memory[vAddr]; | | sw | 101011 | rs | rt | imm | | | sw $1,10($2) | memory[$2+10]  =$1 | memory[rs + (sign-extend)imm] <- rt ；rt=$1,rs=$2 | | sb | 101000 | base | rt | imm | | | sb $1,10(00000) | vAddr=(sign\_extend)10+0;  memory[vAddr]=$1[7:0] | vAddr<-(sign\_extend)imm+base;  memory[vAddr]<-rt[7:0]; | | sh | 101001 | base | rt | imm | | | sh $1,10(00000) | vAddr=(sign\_extend)10+0;  if(vAddr[0]!=0)AddrEr;  memory[vAddr]=$1[15:0] | vAddr<-(sign\_extend)imm+base;  if(vAddr[0]!=0)AddrEr;  memory[vAddr]<-rt[15:0] | | beq | 000100 | rs | rt | imm | | | beq $1,$2,10 | if($1==$2)    goto PC+4+40 | if (rs == rt) PC <- PC+4 + (sign-extend)imm<<2 | | bne | 000101 | rs | rt | imm | | | bne $1,$2,10 | if($1!=$2)   goto PC+4+40 | if (rs != rt) PC <- PC+4 + (sign-extend)imm<<2 | | bgez | 000001 | rs | 00001 | immediate | | | bgez $1,10 | if($1>=0)   goto PC+4+40 | if (rs >= 0) PC <- PC+4 + (sign-extend)immediate<<2 | | bgtz | 000111 | rs | 00000 | immediate | | | bgtz $1,10 | if($1>0)   goto PC+4+40 | if (rs >=0) PC <- PC+4 + (sign-extend)immediate<<2 | | blez | 000110 | rs | 00000 | immediate | | | blez $1,10 | if($1<=0)   goto PC+4+40 | if (rs <= 0) PC <- PC+4 + (sign-extend)immediate<<2 | | bltz | 000001 | rs | 00000 | immediate | | | bltz $1,10 | if($1<0)   goto PC+4+40 | if (rs < 0) PC <- PC+4 + (sign-extend)immediate<<2 | | bgezal | 000001 | rs | 10001 | immediate | | | bgezal $1,10 | if($1>=0)   goto PC+4+40  $31<-PC+4; | if (rs >= 0) PC <- PC+4 + (sign-extend)immediate<<2 | | bltzal | 000001 | rs | 10000 | immediate | | | bltzal $1,10 | if($1<0)   goto PC+4+40  $31<-PC+4; | if (rs < 0) PC <- PC+4 + (sign-extend)immediate<<2 | | slti | 001010 | rs | rt | imm | | | slti $1,$2,10 | if($2<10)    $1=1 else    $1=0 | if (rs <(sign-extend)imm) rt=1 else rt=0 ；    其中rs＝$2，rt=$1 | | sltiu | 001011 | rs | rt | imm | | | sltiu $1,$2,10 | if($2<10)    $1=1 else    $1=0 | if (rs <(zero-extend)imm) rt=1 else rt=0 ；   其中rs＝$2，rt=$1 | | J-type | op | address | | | | |  |  |  | | j | 000010 | address | | | | | j 10000 | goto 10000 | PC <- (PC+4)[31..28],address,0,0   ；address=10000/4 | | jal | 000011 | address | | | | | jal 10000 | $31<-PC+4;  goto 10000 | $31<-PC+4；PC <- (PC+4)[31..28],address,0,0    ；address=10000/4 | | 特权指令 |  |  | | | | |  |  |  | | eret | 010000 | 1 | 000 0000 0000 0000 0000 | | | 011000 | eret | 从终端例外处返回 | PC<-epc;Status.EXL<-0;刷新流水线 | | mfc0 | 010000 | 00000 | rt | rd | 0000 0000 | sel | mfc0 $1,$2,010 | $1=CP0[$2,010] | rt<-CP0[rd,sel] | | mtc0 | 010000 | 00100 | rt | rd | 0000 0000 | sel | mtc0 $1,$2,010 | CP0[$2,010]=$1 | CP0[rd,sel]<-rt | |

注意：因为MIPS16只有16个16位的寄存器，所以JAL指令中$31改成$15, 所有立即数均无需扩展，LUI指令直接就是将立即数付给RT寄存器。