|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 表1 MIPS基础指令特性归纳表 | | | | | | |
| 指令类型 | 汇编指令 | 指令码 | 1 | 2 | 目 | 功能描述 |
| R型指令 | add rd,rs,rt | 0000 00ss ssst tttt dddd d000 0010 0000 | [rs] | [rt] | rd | rd←rs+rt |
| addu rd,rs,rt | 0000 00ss ssst tttt dddd d000 0010 0001 | [rs] | [rt] | rd | rd←rs+rt |
| sub rd,rs,rt | 0000 00ss ssst tttt dddd d000 0010 0010 | [rs] | [rt] | rd | rd←rs-rt |
| subu rd,rs,rt | 0000 00ss ssst tttt dddd d000 0010 0011 | [rs] | [rt] | rd | rd←rs-rt |
| and rd,rs,rt | 0000 00ss ssst tttt dddd d000 0010 0100 | [rs] | [rt] | rd | rd←rs&rt |
| or rd,rs,rt | 0000 00ss ssst tttt dddd d000 0010 0101 | [rs] | [rt] | rd | rd←rs|rt |
| nor rd,rs,rt | 0000 00ss ssst tttt dddd d000 0010 0111 | [rs] | [rt] | rd | rd←~(rs|rt) |
| xor rd,rs,rt | 0000 00ss ssst tttt dddd d000 0010 0110 | [rs] | [rt] | rd | rd←rs^rt |
| slt rd,rs,rt | 0000 00ss ssst tttt dddd d000 0010 1010 | [rs] | [rt] | rd | if sign\_ope(rs<rt) rd=1 |
| sltu rd,rs,rt | 0000 00ss ssst tttt dddd d000 0010 1011 | [rs] | [rt] | rd | if unsign\_ope(rs<rt) rd=1 |
| sll rd,rt,sa | 0000 00ss ssst tttt dddd daaa aa00 0000 |  | [rt] | rd | rd←rt<<sa |
| srl rd,rt,sa | 0000 00ss ssst tttt dddd daaa aa00 0010 |  | [rt] | rd | rd←zero\_ext(rt>>sa) |
| sra rd,rt,sa | 0000 0000 000t tttt dddd daaa aa00 0011 |  | [rt] | rd | rd←zero\_ext(rt>>sa) |
| sllv rd,rt,rs | 0000 00ss ssst tttt dddd d000 0000 0100 | [rs] | [rt] | rd | rd←rt<<rs |
| srlv rd,rt,rs | 0000 00ss ssst tttt dddd d000 0000 0110 | [rs] | [rt] | rd | rd←zero\_ext(rt>>rs) |
| srav rd,rt,rs | 0000 00ss ssst tttt dddd d000 0000 0111 | [rs] | [rt] | rd | rd←sign\_ext(rt>>rs) |
| mov rd,rs | 0000 00ss sss0 0000 dddd d000 0010 0001 | [rs] | $0 | rd | rd←rs+0 |
| jr rs | 0000 00ss sss0 0000 0000 0000 0000 1000 | [rs] |  |  | PC←[rs] |
| I型指令 | addi rt,rs,im | 0010 00ss ssst tttt iiii iiii iiii iiii | [rs] | im | rt | rt←rs+sign\_ext(imm) |
| addiu rt,rs,imm | 0010 01ss ssst tttt iiii iiii iiii iiii | [rs] | im | rt | rt←rs+zero\_ext(imm) |
| andi rt,rs,im | 0011 00ss ssst tttt iiii iiii iiii iiii | [rs] | im | rt | rt←rs&sign\_ext(imm) |
| ori rt,rs,im | 0011 01ss ssst tttt iiii iiii iiii iiii | [rs] | im | rt | rt←rs|sign\_ext(imm) |
| xori rt,rs,im | 0011 10ss ssst tttt iiii iiii iiii iiii | [rs] | im | rt | rt←rs^sign\_ext(imm) |
| lw rt,off(rs) | 1000 11ss ssst tttt iiii iiii iiii iiii | [rs] |  | rt | rt←MEM(rs+sign\_ext(off)) |
| lb rt,off(rs) | 1000 00ss ssst tttt iiii iiii iiii iiii | [rs] |  | rt | rt←MEM(rs+ sign\_ext(off)) |
| sw rt,off(rs) | 1010 11ss ssst tttt iiii iiii iiii iiii | [rs] |  | rt | MEM(rs+ sign\_ext(off))←rt |
| sb rt,off(rs) | 1010 00ss ssst tttt iiii iiii iiii iiii | [rs] |  | rt | MEM(rs+ sign\_ext(off)) ←0xFF&rt |
| beq rs,rt,off | 0001 00ss ssst tttt iiii iiii iiii iiii | [rs] |  |  | if rs==rt PC←PC+4+sign\_ext(off<<2) |
| bgez rs,off | 0000 01ss sss0 0001 iiii iiii iiii iiii | [rs] |  |  | if rs>=0 PC←PC+4+ sign\_ext(off<<2) |
| bgtz rs,off | 0001 11ss sss0 0000 iiii iiii iiii iiii | [rs] |  |  | if rs>0 PC←PC+4+ sign\_ext(off<<2) |
| blez rs,off | 0001 10ss sss0 0000 iiii iiii iiii iiii | [rs] |  |  | if rs<=0 PC←PC+4+ sign\_ext(off<<2) |
| bltz rs,off | 0000 01ss sss0 0000 iiii iiii iiii iiii | [rs] |  |  | if rs<0 PC←PC+4+ sign\_ext(off<<2) |
| bne rs,rt,off | 0001 01ss ssst tttt iiii iiii iiii iiii | [rs] |  |  | if rs!=rt PC←PC+4+sign\_ext(off<<2) |
| J型指令 | j target | 0000 10ii iiii iiii iiii iiii iiii iiii |  |  |  | PC←target |
| jal target | 0000 11ii iiii iiii iiii iiii iiii iiii |  |  |  | $31←PC+4;PC←target |
| halt | 1111 1100 0000 0000 0000 0000 0000 0000 |  |  |  |  |