附录 A　MIPSsim 的指令列表

（MIPS 64 指令集的一个子集）

MIPSsim 是一个指令级和流水线级的 MIPS 模拟器。它能够执行用 MIPS 汇编语言（子集）编写的程序。下面给出它能够执行的 MIPS 指令的列表。

符号说明：

① 在指令助记符中，".W"表示 32 位整数、".L"表示 64 位整数、".S"表示单精度浮点数、".D"表示双精度浮点数。".fmt"表示多种格式的数据，fmt∈(S, D, W, L)。

② 助记符的最后一个字母为 U 表示无符号操作，I 表示与立即值操作，IU 表示无符号立即值操作。助记符的第一个字母为 D 表示是双字（64 位）操作。

③ 为简洁起见，直接用 rs 来表示 rs 寄存器中的内容，其他的如 rt、rd、fs、ft、fd 等都是如此。

④ fs、ft、fd 表示浮点寄存器。一般来说，fs 和 ft 表示源操作数（寄存器），fd 表示结果寄存器。

⑤ rs、rt、rd 表示整数寄存器，也称为通用寄存器。一般来说，rs 和 rt 表示源操作数（寄存器），rd 表示目的寄存器。

⑥ 以下是两个特殊寄存器：

LO——常用来存放乘积的低 32 位（或 64 位）以及除法的商；

HI——常用来存放乘积的高 32 位（或 64 位）以及除法的余数。

1. 算术运算指令

名称	格式	描述
寄存器加（ADD）	ADD rd, rs, rt	rd←rs + rt 32 位，按有符号数操作
立即值加（ADDI）	ADDI rt, rs, immediate	rt←rs + immediate 32 位，按有符号数操作， immediate 都是 16 位的。下同

续表

名称	格式	描述
无符号立即值加(ADDIU)	ADDIU rt, rs, immediate	rt←rs + immediate 32 位,按无符号数操作
无符号加(ADDU)	ADDU rd, rs, rt	rd←rs + rt 32 位,按无符号数操作
双字寄存器加(DADD)	DADD rd, rs, rt	rd ←rs + rt 按有符号数操作
双字立即值加(DADDI)	DADDI rt, rs, immediate	rt ←rs + immediate 按有符号数操作
双字无符号立即值加(DADDIU)	DADDIU rt, rs, immediate	rt ←rs + immediate 按无符号数操作
双字无符号加(DADDU)	DADDU rd, rs, rt	rd ←rs + rt 按无符号数操作
寄存器减(SUB)	SUB rd, rs, rt	rd←rs - rt 32 位,按有符号数操作
无符号减(SUBU)	SUBU rd, rs, rt	rd←rs - rt 32 位,按无符号数操作
双字寄存器减(DSUB)	DSUB rd, rs, rt	rd ←rs - rt 按有符号数操作
双字无符号减(DSUBU)	DSUBU rd, rs, rt	rd ←rs - rt 按无符号数操作
寄存器乘 1(MUL)	MUL rd, rs, rt	rd←rs×rt 32 位,按有符号数操作
寄存器乘 2(MULT)	MULT rs, rt	(LO,HI)←rs×rt,32 位,按有符号数操作。所得到的积的低 32 位按符号扩展后送特殊寄存器 LO,高 32 位按符号扩展后送特殊寄存器 HI
无符号寄存器乘(MULTU)	MULTU rs, rt	按无符号数操作。其余同 MULT
双字寄存器乘(DMULT)	DMULT rs, rt	(LO,HI)←rs×rt 积的低 64 位送 LO,高 64 位送 HI,按有符号数操作
双字无符号乘(DMULTU)	DMULTU rs, rt	按无符号数操作。其余同 DMULT

续表

名称	格式	描述
寄存器除(DIV)	DIV rs, rt	(LO,HI)←rs/rt 32 位商送 LO,32 位余数送 HI,按有符号数操作
无符号寄存器除(DIVU)	DIVU rs, rt	按无符号数操作。其余同 DIV
双字寄存器除(DDIV)	DDIV rs, rt	(LO,HI)←rs/rt 64 位商送 LO,64 位余数送 HI,按有符号数操作
双字无符号除(DDIVU)	DDIVU rs, rt	按无符号数操作。其余同 DDIV
小于比较(SLT)	SLT rd, rs, rt	if (rs < rt)　rd←1 else rd←0 按有符号数操作
无符号小于比较(SLTU)	SLTU rd, rs, rt	if (rs < rt)　rd←1 else rd←0 按无符号数操作
立即值小于比较(SLTI)	SLTI rt, rs, immediate	if (rs < immediate) rt←1 else rt←0 按有符号数操作
无符号立即值小于比较(SLTIU)	SLTIU rt, rs, immediate	if (rs < immediate) rt←1 else rt←0 按无符号数操作
字节符号位扩展(SEB)	SEB rd, rt	rd←rt 的末字节按符号位扩展
半字符号位扩展(SEH)	SEH rd, rt	rd←rt 的后半字按符号位扩展

2. 逻辑运算指令

名称	格式	描述
与(AND)	AND rd, rs, rt	rd←rs AND rt
立即值与(ANDI)	ANDI rt, rs, immediate	rt←rs AND immediate
取立即值高位(LUI)	LUI rt, immediate	16 位 immediate 低位拼接 16 位 0,然后按符号位扩展后装入 rt
或非(NOR)	NOR rd, rs, rt	rd←rs NOR rt
或(OR)	OR rd, rs, rt	rd←rs OR rt

续表

名称	格式	描述
立即值或(ORI)	ORI rt, rs, immediate	rt←rs OR immediate
异或(XOR)	XOR rd, rs, rt	rd←rs XOR rt
立即值异或(XORI)	XORI rt, rs, immediate	rt←rs XORI immediate

3. CPU 移位指令

名称	格式	描述
按立即值逻辑左移(SLL)	SLL rd, rt, sa	rd←rt << sa rt 中的低 32 位进行逻辑左移,结果按符号位扩展,然后放入 rd。移动的位数由立即值 sa 给出
按立即值算术右移(SRA)	SRA rd, rt, sa	rd←rt >> sa rt 中的低 32 位进行算术右移,其余同 SLL
按立即值逻辑右移(SRL)	SRL rd, rt, sa	rd←rt >> sa rt 中的低 32 位进行逻辑右移,其余同 SLL
按变量逻辑左移(SLLV)	SLLV rd, rt, rs	rd←rt << rs rt 中的低 32 位进行逻辑左移,结果进行符号位扩展,然后放入 rd。移动的位数由寄存器 rs 给出
按变量算术右移(SRAV)	SRAV rd, rt, rs	rd←rt >> rs rt 中的低 32 位进行算术右移,其余同 SLLV
按变量逻辑右移(SRLV)	SRLV rd, rt, rs	rd←rt >> rs rt 中的低 32 位进行逻辑右移,其余同 SLLV
按立即值循环右移(ROTR)	ROTR rd, rt, sa	rd← rt 中的低 32 位进行循环右移。移动的位数由立即值 sa 给出
按变量循环右移(ROTRV)	ROTRV rd, rt, rs	rd← rt 中的低 32 位进行循环右移。移动的位数由寄存器 rs 给出
双字按立即值逻辑左移(DSLL)	DSLL rd, rt, sa	rd←rt << sa 移动的位数由立即值 sa 给出

续表

名称	格式	描述
双字按立即值逻辑右移(DSRL)	DSRL rd, rt, sa	rd←rt >> sa 移动的位数由立即值 sa 给出
双字按立即值算术右移(DSRA)	DSRA rd, rt, sa	rd←rt >> sa 移动的位数由立即值 sa 给出
双字按变量逻辑左移(DSLLV)	DSLLV rd, rt, rs	rd←rt << rs 移动的位数由寄存器 rs 给出
双字按变量逻辑右移(DSRLV)	DSRLV rd, rt, rs	rd←rt >> rs 移动的位数由寄存器 rs 给出
双字按变量算术右移(DSRAV)	DSRAV rd, rt, rs	rd←rt >> rs 移动的位数由寄存器 rs 给出
双字按立即值循环右移(DROTR)	DROTR rd, rt, sa	rd← rt 中的双字循环右移。 移动的位数由立即值 sa 给出
双字按变量循环右移(DROTRV)	DROTRV rd, rt, rs	rd← rt 中的双字循环右移。 移动的位数由寄存器 rs 给出

4. CPU 传送指令

名称	格式	描述
等于 0 传送(MOVZ)	MOVZ rd, rs, rt	If(rt=0) then rd←rs
不等于 0 传送(MOVN)	MOVN rd, rs, rt	If(rt!=0) then rd←rs
从 HI 传送至整数寄存器(MFHI)	MFHI rd	rd←HI
从 LO 传送至整数寄存器(MFLO)	MFLO rd	rd←LO
从整数寄存器传送至 HI(MTHI)	MTHI rs	HI ←rs
从整数寄存器传送至 LO(MTLO)	MTLO rs	LO ←rs
浮点条件码为假整数传送(MOVF)	MOVF rd, rs, cc	if FPConditionCode(cc) = 0 then rd←rs
浮点条件码为真整数传送(MOVT)	MOVT rd, rs, cc	if FPConditionCode(cc) = 1 then rd←rs

5. 浮点传送指令

名称	格式	描述
把浮点控制寄存器的内容传送至整数寄存器(CFC1)	CFC1 rt, fs	把 fs 中的 32 位数据按符号位扩展成 64 位后送入 rt
把整数寄存器的内容传送至浮点控制寄存器(CTC1)	CTC1 rt, fs	fs ← rt
从浮点寄存器传送双字到整数寄存器(DMFC1)	DMFC1 rt, fs	rt←fs
从整数寄存器传送双字到浮点寄存器(DMTC1)	DMTC1 rt, fs	fs ← rt
从浮点寄存器传送 32 位至整数寄存器(MFC1)	MFC1 rt, fs	rt←fs 符号位扩展后
从整数寄存器传送 32 位至浮点寄存器(MTC1)	MTC1 rt, fs	fs←rt
单精度浮点传送(MOV.S)	MOV.S fd, fs	fd←fs
双精度浮点传送(MOV.D)	MOV.D fd, fs	
等于 0 单精度浮点传送(MOVZ.S)	MOVZ.S fd, fs, rt	If(rt=0) then fd←fs
等于 0 双精度浮点传送(MOVZ.D)	MOVZ.D fd, fs, rt	
不等于 0 单精度浮点传送(MOVN.S)	MOVN.S fd, fs, rt	If(rt!=0) then fd←fs
不等于 0 双精度浮点传送(MOVN.D)	MOVN.D fd, fs, rt	
浮点条件码为假单精度浮点传送(MOVF.S)	MOVF.S fd,fs, cc	if FPConditionCode(cc) = 0 then fd←fs
浮点条件码为假双精度浮点传送(MOVF.D)	MOVF.D fd,fs, cc	if FPConditionCode(cc) = 0 then fd←fs
浮点条件码为真单精度浮点传送(MOVT.S)	MOVT.S fd,fs, cc	if FPConditionCode(cc) = 1 then fd←fs
浮点条件码为真双精度浮点传送(MOVT.D)	MOVT.D fd,fs, cc	if FPConditionCode(cc) = 1 then fd←fs

6. 分支指令

名称	格式	描述
无条件转移(B)	B offset	用以下指令实现： BEQ r0, r0, offset
等于 0 转移(BEQZ)	BEQZ rs, rt, offset	if(rs=0)　then action action 表示以 offset 作为相对于 PC+4 的偏移量进行转移。下同
不等于 0 转移(BNEZ)	BNEZ rs, rt, offset	if(rs!=0)　then action
相等转移(BEQ)	BEQ rs, rt, offset	if(rs=rt)　then action
不相等转移(BNE)	BNE rs, rt, offset	if(rs!=rt)　then action
大于等于 0 转移(BGEZ)	BGEZ rs, offset	if(rs>=0)　then action
大于 0 转移(BGTZ)	BGTZ rs, offset	if(rs>0)　then action
小于等于 0 转移(BLEZ)	BLEZ rs, offset	if(rs<=0)　then action
小于 0 转移(BLTZ)	BLTZ rs, offset	if(rs<0)　then action
大于等于 0 转移并链接(BGEZAL)	BGEZAL rs, offset	If (rs>=0)　then action,并将返回地址(当前的 PC 值)保存到 R31
小于 0 转移并链接(BLTZAL)	BLTZAL rs, offset	If (rs<0)　then action,并将返回地址保存到 R31
从异常处理返回(ERET)	ERET	

7. 跳转指令

名称	格式	描述
跳转(J)	J target	无条件转移到目标地址 target
寄存器跳转(JR)	JR rs	无条件转移,目标地址由 rs 给出
跳转并链接(JAL)	JAL target	无条件转移到目标地址 target,并将返回地址 PC+4 保存到 R31
寄存器跳转并链接(JALR)	JALR rd, rs	无条件转移到 rs 给出的地址,并将返回地址 PC+4 保存到 R31

8. 访存指令

从存储器读出数据或将数据写入存储器。存储器的地址按“16 位偏移量 offset+定点寄存器 base 的内容”计算。

名称	格式	描述
取字节(LB)	LB rt, offset(base)	rt←memory[base+offset] 按有符号数操作
取半字(LH)	LH rt, offset(base)	
取字(LW)	LW rt, offset(base)	
取无符号字节(LBU)	LBU rt, offset(base)	rt←memory[base+offset] 按无符号数操作
取无符号半字(LHU)	LHU rt, offset(base)	
取无符号字(LWU)	LWU rt, offset(base)	
取双字(LD)	LD rt, offset(base)	rt←memory[base+offset]
取浮点数(LDC1)	LDC1 ft, offset(base)	ft←memory[base+offset]
存字节(SB)	SB rt, offset(base)	memory[base+offset]←rt
存半字(SH)	SH rt, offset(base)	
存字(SW)	SW rt, offset(base)	
存双字(SD)	SD rt, offset(base)	
存浮点数(SDC1)	SDC1 ft, offset(base)	memory[base+offset]←ft

9. 浮点访存指令

名称	格式	描述
取单精度浮点数(L.S)	L.S ft, offset(base)	ft←memory[base+offset] ft 及下面的 fd、fs 为浮点寄存器
取双精度浮点数(L.D)	L.D ft, offset(base)	
取字到 FPR(LWC1)	LWC1 ft, offset(base)	
取双字到 FPR(LDC1)	LDC1 ft, offset(base)	
存单精度浮点数(S.S)	S.S ft, offset(base)	memory[base+offset] ←ft
存双精度浮点数(S.D)	S.D ft, offset(base)	
存 FPR 中的单字到存储器(SWC1)	SWC1 ft, offset(base)	memory[base+offset] ←ft
存 FPR 中的双字到存储器(SDC1)	SDC1 ft, offset(base)	
变址取单字到 FPR(LWXC1)	LWXC1 fd, index(base)	fd←memory[base+index]
变址取双字到 FPR(LDXC1)	LDXC1 fd, index(base)	

续表

名称	格式	描述
变址存 FPR 中的单字(SWXC1)	SWXC1 fs, index(base)	memory[base+index]←fs
变址存 FPR 中的双字(SDXC1)	SDXC1 fs, index(base)	

10. 自陷指令

名称	格式	描述
等于自陷(TEQ)	TEQ rs, rt	If rs=rt then trap
不等于自陷(TNE)	TNE rs, rt	If rs!=rt then trap
大于等于自陷(TGE)	TGE rs, rt	If rs>=rt then trap
小于自陷(TLT)	TLT rs, rt	If rs<rt then trap
大于等于自陷(TGEU)	TGEU rs, rt	If rs>=rt then trap
小于自陷(TLTU)	TLTU rs, rt	If rs<rt then trap
等于立即值自陷(TEQI)	TEQI rs, immediate	If rs=immediate then trap
不等于立即值自陷(TNEI)	TNEI rs, immediate	If rs!=immediate then trap
大于等于立即值自陷(TGEI)	TGEI rs, immediate	If rs>=immediate then trap
小于立即值自陷(TLTI)	TLTI rs, immediate	If rs<immediate then trap
大于等于无符号立即值自陷(TGEIU)	TGEIU rs, immediate	If rs>=immediate then trap
小于无符号立即值自陷(TLTIU)	TLTIU rs, immediate	If rs<immediate then trap

11. 浮点运算指令

名称	格式	描述
单精度求绝对值(ABS.S)	ABS.S fd, fs	fd←abs(fs)
双精度求绝对值(ABS.D)	ABS.D fd, fs	
单精度浮点加(ADD.S)	ADD.S fd, fs, ft	fd←fs + ft
双精度浮点加(ADD.D)	ADD.D fd, fs, ft	

续表

名称	格式	描述
单精度浮点减(SUB.S)	SUB.S fd, fs, ft	fd←fs - ft
双精度浮点减(SUB.D)	SUB.D fd, fs, ft	
单精度浮点乘(MUL.S)	MUL.S fd, fs, ft	fd←fs×ft
双精度浮点乘(MUL.D)	MUL.D fd, fs, ft	
单精度浮点除(DIV.S)	DIV.S fd, fs, ft	fd←fs / ft
双精度浮点除(DIV.D)	DIV.D fd, fs, ft	
单精度浮点求负(NEG.S)	NEG.S fd, fs	fd← -fs
双精度浮点求负(NEG.D)	NEG.D fd, fs	fd← -fs
单精度求平方根(SQRT.S)	SQRT.S fd, fs	fd←SQRT(fs)
双精度求平方根(SQRT.D)	SQRT.D fd, fs	

12. 浮点分支指令

浮点条件码为假转移(BC1F)	BC1F cc,offset	if FPConditionCode(cc) = 0 then 转移
浮点条件码为真转移(BC1T)	BC1T cc,offset	if FPConditionCode(cc) = 1 then 转移

13. 浮点比较指令

单精度浮点比较并设置条件码 (C.cond.S)	C.cond.S fs, ft	cond ∈ (LT, GT, LE, GE, EQ, NE),按 cond 的关系对 fs 和 ft 进行比较,如果关系成立,则置浮点条件码 cc 为 1,否则置为 0
双精度浮点比较并设置条件码 (C.cond.D)	C.cond.D fs, ft	按双精度比较。其余同 C.cond.S

14. 浮点转换指令

名称	格式	描述
单精度浮点数转换成双精度(CVT.D.S)	CVT.D.S fd, fs	CVT.x.y 把 fs 中的 y 类型的数据转换成 x 类型，并送给 fd。x，y∈(L(64 位整数)，W(32 位整数)，D(双精度)，S(单精度))
32 位整数转换成双精度浮点数(CVT.D.W)	CVT.D.W fd, fs	
64 位整数转换成双精度浮点数(CVT.D.L)	CVT.D.L fd, fs	
32 位整数转换成单精度浮点数(CVT.S.W)	CVT.S.W fd, fs	
64 位整数转换成单精度浮点数(CVT.S.L)	CVT.S.L fd, fs	
双精度浮点数转换成单精度(CVT.S.D)	CVT.S.D fd, fs	
单精度浮点数转换成 32 位整数(CVT.W.S)	CVT.W.S fd, fs	
双精度浮点数转换成 32 位整数(CVT.W.D)	CVT.W.D fd, fs	
单精度浮点数转换成 64 位整数(CVT.L.S)	CVT.L.S fd, fs	
双精度浮点数转换成 64 位整数(CVT.L.D)	CVT.L.D fd, fs	
单精度转换并截断成单字整数(TRUNC.L.S)	TRUNC.L.S fd, fs	fd←fs 转换并截断
双精度转换并截断成单字整数(TRUNC.L.D)	TRUNC.L.D fd, fs	
单精度转换并截断成双字整数(TRUNC.W.S)	TRUNC.W.S fd, fs	
双精度转换并截断成双字整数(TRUNC.W.D)	TRUNC.W.D fd, fs	