

4.2.2 指令格式

MIPS 所有指令都是 32 位定长指令,主要有 3 种指令格式,如表 4.3 所示。同一类型的指令格式中对应的比特位包含的数据或操作数相同,其中 opcode 字段 6 位,寄存器字段用 5 位表示,可以访问 32 个通用寄存器,最多可以有 3 个寄存器操作数,5 位的 shamt 字段是移位变量,6 位的 funct 字段用于区分 R 型指令的具体功能。I 型指令的立即数是 16 位,所以能使用的有符号立即数范围为 $[-32768, 32767]$ 。J 型指令的 Address 字段为 26 位,因此最长的局部跳转指令只能跳转 2^{26} B,也就是 $2^{24} = 16\ 777\ 216$ 条指令。

表 4.3 MIPS 指令格式

Type	31~26	25~21	20~16	15~11	10~06	05~00
R-Type	opcode	\$rs	\$rt	\$rd	shamt	funct
I-Type	opcode	\$rs	\$rt	Imm(16 位)		
J-Type	opcode	Address(26 位)				

4.2.3 R 型指令

R 型指令的操作码字段均为 0,不同指令的 funct 字段值不同,除了前 3 个移位指令会使用 shamt 字段外,其他所有 R 型指令均只使用寄存器操作数,如表 4.4 所示。除了运算指令外,R 型指令还包括几条跳转指令和系统调用指令。

表 4.4 MIPS R 型指令功能详解

func	指令助记符	RTL 功能描述	备 注
00	sll\$rd,\$rt,shamt	$R[\$rd] \leftarrow R[\$rt] \ll \text{shamt}$	逻辑左移
02	srl\$rd,\$rt,shamt	$R[\$rd] \leftarrow R[\$rt] \gg \text{shamt}$	逻辑右移
03	sra\$rd,\$rt,shamt	$R[\$rd] \leftarrow R[\$rt] \ggg \text{shamt}$	算术右移
04	sllv\$rd,\$rt,\$rs	$R[\$rd] \leftarrow R[\$rt] \ll R[\$rs]$	可变左移
06	srlv\$rd,\$rt,\$rs	$R[\$rd] \leftarrow R[\$rt] \gg R[\$rs]$	逻辑可变右移
07	srav\$rd,\$rt,\$rs	$R[\$rd] \leftarrow R[\$rt] \ggg R[\$rs]$	算术可变右移
08	jr\$rs	$PC \leftarrow R[\$rs]$	$R[\$rs]$ 值应是 4 的倍数,字对齐
09	jalr\$rd,\$rs	$\text{tmp} \leftarrow R[\$rs]$ $R[\$rd] \leftarrow PC + 8$ $PC \leftarrow \text{tmp}$	$R[\$rs]$ 值应是 4 的倍数,字对齐 Undefined if $\$rs = \rd (无延迟槽则 $PC + 4$)
09	jalr \$rs	等同于 jalr \$31,\$rs	\$31 号寄存器存储子程序返回地址
12	syscall	系统调用	
16	mfhi \$rd	$R[\$rd] \leftarrow HI$	取 HI 寄存器的值
17	mtthi \$rs	$HI \leftarrow R[\$rs]$	

func	指令助记符	RTL 功能描述	备 注
18	mflo \$rd	$R[\$rd] \leftarrow LO$	取 LO 寄存器的值
19	mtlo \$rs	$LO \leftarrow R[\$rs]$	
24	mult \$rs,\$rt	$\{HI, LO\} \leftarrow R[\$rs] * R[\$rt]$	有符号乘
25	multu \$rs,\$rt	$\{HI, LO\} \leftarrow R[\$rs] * R[\$rt]$	无符号乘
26	div \$rs,\$rt	$LO \leftarrow R[\$rs] / R[\$rt]$ $HI \leftarrow R[\$rs] \% R[\$rt]$	有符号除法
27	divu \$rs,\$rt	$LO \leftarrow R[\$rs] / R[\$rt]$ $HI \leftarrow R[\$rs] \% R[\$rt]$	无符号除法
32	add \$rd,\$rs,\$rt	$R[\$rd] \leftarrow R[\$rs] + R[\$rt]$	溢出时产生异常,且不修改 $R[\$rd]$
33	addu \$rd,\$rs,\$rt	$R[\$rd] \leftarrow R[\$rs] + R[\$rt]$	无符号加
34	sub \$rd,\$rs,\$rt	$R[\$rd] \leftarrow R[\$rs] - R[\$rt]$	溢出时产生异常,且不修改 $R[\$rd]$
35	subu \$rd,\$rs,\$rt	$R[\$rd] \leftarrow R[\$rs] - R[\$rt]$	无符号减
36	and \$rd,\$rs,\$rt	$R[\$rd] \leftarrow R[\$rs] \& R[\$rt]$	逻辑与
37	or \$rd,\$rs,\$rt	$R[\$rd] \leftarrow R[\$rs] R[\$rt]$	逻辑或
38	xor \$rd,\$rs,\$rt	$R[\$rd] \leftarrow R[\$rs] \wedge R[\$rt]$	异或
39	nor \$rd,\$rs,\$rt	$R[\$rd] \leftarrow \neg (R[\$rs] R[\$rt])$	或非
42	slt \$rd,\$rs,\$rt	$R[\$rd] \leftarrow R[\$rs] < R[\$rt]$	小于置 1,有符号比较
43	sltu \$rd,\$rs,\$rt	$R[\$rd] \leftarrow R[\$rs] < R[\$rt]$	小于置 1,无符号比较

4.2.4 I 型指令

I 型指令中不同功能的指令操作码不同,所有指令均包含一个 16 位立即数字段,详见表 4.5。

表 4.5 MIPS I 型指令功能详解

opcode	指令助记符	RTL 功能描述	备 注
04	beq \$rs,\$rt,imm	if($R[\$rs] = R[\$rt]$) $PC \leftarrow PC + 4 + \text{SignExt}_{16b}(\{imm, 00\})$	
05	bne \$rs,\$rt,imm	if($R[\$rs] \neq R[\$rt]$) $PC \leftarrow PC + 4 + \text{SignExt}_{16b}(\{imm, 00\})$	
06	blez \$rs,imm	if($R[\$rs] \leq 0$) $PC \leftarrow PC + 4 + \text{SignExt}_{16b}(\{imm, 00\})$	有符号比较
07	bgtz \$rs,imm	if($R[\$rs] > 0$) $PC \leftarrow PC + 4 + \text{SignExt}_{16b}(\{imm, 00\})$	有符号比较
08	addi \$rt,\$rs,imm	$R[\$rt] \leftarrow R[\$rs] + \text{SignExt}_{16b}(imm)$	溢出产生异常
09	addiu \$rt,\$rs,imm	$R[\$rt] \leftarrow R[\$rs] + \text{SignExt}_{16b}(imm)$	

opcode	指令助记符	RTL 功能描述	续表
			备 注
10	slti\$rt,\$rs,imm	$R[rt] \leftarrow R[rs] < \text{SignExt}_{16b}(imm)$	有符号比较
11	sltiu\$rt,\$rs,imm	$R[rt] \leftarrow R[rs] < \text{SignExt}_{16b}(imm)$	无符号比较
12	andi\$rt,\$rs,imm	$R[rt] \leftarrow R[rs] \& \{0 \times 16, imm\}$	
13	ori\$rt,\$rs,imm	$R[rt] \leftarrow R[rs] \{0 \times 16, imm\}$	
14	xori\$rt,\$rs,imm	$R[rt] \leftarrow R[rs] \wedge \{0 \times 16, imm\}$	
15	lui\$rt,imm	$R[rt] \leftarrow \{(imm)[15:0], 0 \times 16\}$	
32	lb\$rt,imm(\$rs)	$R[rt] \leftarrow \text{SignExt}_{8b}(\text{Mem}_{1B}(R[rs] + \text{SignExt}_{16b}(imm)))$	
33	lh\$rt,imm(\$rs)	$R[rt] \leftarrow \text{SignExt}_{16b}(\text{Mem}_{2B}(R[rs] + \text{SignExt}_{16b}(imm)))$	半字对齐
35	lw\$rt,imm(\$rs)	$R[rt] \leftarrow \text{Mem}_{4B}(R[rs] + \text{SignExt}_{16b}(imm))$	字对齐
36	lbu\$rt,imm(\$rs)	$R[rt] \leftarrow \{0 \times 24, \text{Mem}_{1B}(R[rs] + \text{SignExt}_{16b}(imm))\}$	
37	lhu\$rt,imm(\$rs)	$R[rt] \leftarrow \{0 \times 16, \text{Mem}_{2B}(R[rs] + \text{SignExt}_{16b}(imm))\}$	半字对齐
40	sb\$rt,imm(\$rs)	$\text{Mem}_{1B}(R[rs] + \text{SignExt}_{16b}(imm)) \leftarrow (R[rt])[7:0]$	
41	sh\$rt,imm(\$rs)	$\text{Mem}_{2B}(R[rs] + \text{SignExt}_{16b}(imm)) \leftarrow (R[rt])[15:0]$	半字对齐
43	sw\$rt,imm(\$rs)	$\text{Mem}_{4B}(R[rs] + \text{SignExt}_{16b}(imm)) \leftarrow R[rt]$	字对齐

4.2.5 J型指令

J型指令的操作码字段为2或者3,其包含的跳转指令采用伪直接寻址,PC高4位与26位address字段组合后左移两位得到目标地址,详见表4.6。

表 4.6 MIPS J型指令功能详解

opcode	指令助记符	RTL 功能描述	备 注
02	j address	$PC \leftarrow \{(PC+4)[31:28], address, 00\}$	无条件分支
03	jal address	$R[31] \leftarrow PC+8$ (无延迟槽则加4) $PC \leftarrow \{(PC+4)[31:28], address, 00\}$	保存返回地址到31号寄存器,同时跳转,用于子程序调用

4.2.6 MIPS寻址方式

从以上几种基本指令类型不难发现,MIPS指令获取操作数的方式主要包括寄存器寻址(操作数在寄存器中)、立即数寻址(操作数在指令字中)和变址寻址(操作数在主存中)3种寻址方式,只有I型指令中的访存指令load/store指令(lb、lh、lw、lbu、lhu、sb、sh、sw等)可以访问存储器,其他所有指令的操作数均在寄存器中或指令字中,这也是MIPS优化指令流水线的重要特性之一。

访存指令使用如下格式进行加载或存储数据: