

## 附录 A

实验所附的 coe 文件中所有指令均包含于下表中，可供查询 opcode 及 funct 所代表的具体指令。

表 3.1 MIPS 的 31 种指令

助记符	指 令 格 式						示 例	示例含义	操作及解释
BIT #	31..26	25..21	20..16	15..11	10..6	5..0			
R-类型	op	rs	rt	rd	shamt	func			
add	000000	rs	rt	rd	00000	100000	add \$1,\$2,\$3	$S1 = S2 + S3$	$(rd) \leftarrow (rs) + (rt); rs = S2, rt = S3, rd = S1$
addu	000000	rs	rt	rd	00000	100001	addu \$1,\$2,\$3	$S1 = S2 + S3$	$(rd) \leftarrow (rs) + (rt); rs = S2, rt = S3, rd = S1$ , 无符号数
sub	000000	rs	rt	rd	00000	100010	sub \$1,\$2,\$3	$S1 = S2 - S3$	$(rd) \leftarrow (rs) - (rt); rs = S2, rt = S3, rd = S1$
subu	000000	rs	rt	rd	00000	100011	subu \$1,\$2,\$3	$S1 = S2 - S3$	$(rd) \leftarrow (rs) - (rt); rs = S2, rt = S3, rd = S1$ , 无符号数
and	000000	rs	rt	rd	00000	100100	and \$1,\$2,\$3	$S1 = S2 \& S3$	$(rd) \leftarrow (rs) \& (rt); rs = S2, rt = S3, rd = S1$
or	000000	rs	rt	rd	00000	100101	or \$1,\$2,\$3	$S1 = S2   S3$	$(rd) \leftarrow (rs)   (rt); rs = S2, rt = S3, rd = S1$
xor	000000	rs	rt	rd	00000	100110	xor \$1,\$2,\$3	$S1 = S2 \wedge S3$	$(rd) \leftarrow (rs) \wedge (rt); rs = S2, rt = S3, rd = S1$
nor	000000	rs	rt	rd	00000	100111	nor \$1,\$2,\$3	$S1 = \sim(S2   S3)$	$(rd) \leftarrow \sim((rs)   (rt)); rs = S2, rt = S3, rd = S1$
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; 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; rs = \$2, rt = \$3, rd = \$1, 无符号数
sll	000000	00000	rt	rd	shamt	000000	sll \$1,\$2,10	$S1 = S2 \ll 10$	$(rd) \leftarrow (rt) \ll \text{shamt}, rt = S2, rd = S1, \text{shamt} = 10$
srl	000000	00000	rt	rd	shamt	000010	srl \$1,\$2,10	$S1 = S2 \gg 10$	$(rd) \leftarrow (rt) \gg \text{shamt}, rt = S2, rd = S1, \text{shamt} = 10$ , (逻辑右移)
sra	000000	00000	rt	rd	shamt	000011	sra \$1,\$2,10	$S1 = S2 \gg 10$	$(rd) \leftarrow (rt) \gg \text{shamt}, rt = S2, rd = S1, \text{shamt} = 10$ , (算术右移, 注意符号位保留)
sllv	000000	rs	rt	rd	00000	000100	sllv \$1,\$2,\$3	$S1 = S2 \ll S3$	$(rd) \leftarrow (rt) \ll (rs), rs = S3, rt = S2, rd = S1$
srlv	000000	rs	rt	rd	00000	000110	srlv \$1,\$2,\$3	$S1 = S2 \gg S3$	$(rd) \leftarrow (rt) \gg (rs), rs = S3, rt = S2, rd = S1$ , (逻辑右移)
srav	000000	rs	rt	rd	00000	000111	srav \$1,\$2,\$3	$S1 = S2 \gg S3$	$(rd) \leftarrow (rt) \gg (rs), rs = S3, rt = S2, rd = S1$ , (算术右移, 注意符号位保留)
jr	000000	rs	00000	00000	00000	001000	jr \$31	goto \$31	$(PC) \leftarrow (rs)$
I-类型	op	rs	rt	immediate					
addi	001000	rs	rt	immediate			addi \$1,\$2,10	$S1 = S2 + 10$	$(rt) \leftarrow (rs) + (\text{sign\_extend})\text{immediate}, rt = S1, rs = S2$
addiu	001001	rs	rt	immediate			addiu \$1,\$2,10	$S1 = S2 + 10$	$(rt) \leftarrow (rs) + (\text{sign\_extend})\text{immediate}, rt = S1, rs = S2$
andi	001100	rs	rt	immediate			andi \$1,\$2,10	$S1 = S2 \& 10$	$(rt) \leftarrow (rs) \& (\text{zero\_extend})\text{immediate}, rt = S1, rs = S2$
ori	001101	rs	rt	immediate			ori \$1,\$2,10	$S1 = S2   10$	$(rt) \leftarrow (rs)   (\text{zero\_extend})\text{immediate}, rt = S1, rs = S2$
xori	001110	rs	rt	immediate			xori \$1,\$2,10	$S1 = S2 \wedge 10$	$(rt) \leftarrow (rs) \wedge (\text{zero\_extend})\text{immediate}, rt = S1, rs = S2$
lui	001111	00000	rt	immediate			lui \$1,10	$S1 = 10 * 65536$	$(rt) \leftarrow \text{immediate} \ll 16 \& 0FFFF0000H$ , 将 16 位立即数放到目的寄存器高 16 位, 目的寄存器的低 16 位填 0
lw	100011	rs	rt	offset			lw \$1,10(\$2)	$S1 = \text{Memory}[S2 + 10]$	$(rt) \leftarrow \text{Memory}[(rs) + (\text{sign\_extend})\text{offset}], rt = S1, rs = S2$
sw	101011	rs	rt	offset			sw \$1,10(\$2)	$\text{Memory}[S2 + 10] = S1$	$\text{Memory}[(rs) + (\text{sign\_extend})\text{offset}] \leftarrow (rt), rt = S1, rs = S2$
beq	000100	rs	rt	offset			beq \$1,\$2,40	if (\$1 = \$2) goto PC+4+40	if ((rt) = (rs)) then (PC) ← (PC) + 4 + (Sign-Extend)offset < 2, rs = \$1, rt = \$2

bne	000101	rs	rt	offset	bne \$1,\$2,40	if(\$1≠\$2) goto PC+4+40	if ((rt)≠(rs)) then (PC)←(PC)+4+( (Sign-Extend) offset<<2) , rs=\$1, rt=\$2
slti	001010	rs	rt	immediate	slti \$1,\$2,10	if(\$2<10) \$1=1 else \$1=0	if ((rs)<(Sign-Extend)immediate) then (rt)←-1; else (rt)←0, rs=\$2, rt=\$1
sltiu	001011	rs	rt	immediate	sltiu \$1,\$2,10	if(\$2<10) \$1=1 else \$1=0	if ((rs)<(Zero-Extend)immediate) then (rt)←-1; else (rt)←0, rs=\$2, rt=\$1
<b>J-类型</b>	<b>op</b>	<b>address</b>					
j	000010	address			j 10000	goto 10000	(PC)←( (Zero-Extend) address<<2), address=10000/4
jal	000011	address			jal 10000	\$31=PC+4 goto 10000	(\$31)←(PC)+4; (PC)←( (Zero-Extend) address<<2), address=10000/4