

计算机组成原理

第五章 指令系统

5.6 MIPS指令详解

lw \$t0, 0(\$2)

lw \$t1, 4(\$2)

sw \$t1, 0(\$2)

sw \$t0, 4(\$2)

0000 1001 1100 0110 1010 1111 0101 1000
1010 1111 0101 1000 0000 1001 1100 0110
1100 0110 1010 1111 0101 1000 0000 1001
0101 1000 0000 1001 1100 0110 1010 1111

1

R型指令



操作数和保存结果均通过寄存器进行;

- ◆ op: 操作码, 所有R型指令中都全为0;
- ◆ rs: 寄存器编号, 对应第1个源操作数;
- ◆ rt: 寄存器编号, 对应第2个源操作数;
- ◆ rd: 寄存器编号, 据此保存结果;
- ◆ shamt: 常数, 在移位指令中使用;
- ◆ funct: 功能码, 指定指令的具体功能;

1

R型指令

指令	[31:26]	[25:21]	[20:16]	[15:11]	[10:6]	[5:0]	指令功能
add	000000	rs	rt	rd	<u>00000</u>	100000	寄存器加
sub	000000	rs	rt	rd	<u>00000</u>	100010	寄存器减
and	000000	rs	rt	rd	<u>00000</u>	100100	寄存器与
or	000000	rs	rt	rd	<u>00000</u>	100101	寄存器或
xor	000000	rs	rt	rd	<u>00000</u>	100110	寄存器异或
sll	000000	<u>00000</u>	rt	rd	sa	000000	逻辑左移
srl	000000	<u>00000</u>	rt	rd	sa	000010	逻辑右移
sra	000000	<u>00000</u>	rt	rd	sa	000011	算术右移
jr	000000	rs	<u>00000</u>	<u>00000</u>	<u>00000</u>	001000	寄存器跳转

1

R型指令

■ R型指令存在3种不同类型

◆ 3寄存器R型指令

指令	[31:26]	[25:21]	[20:16]	[15:11]	[10:6]	[5:0]	指令功能
add	000000	rs	rt	rd	<u>00000</u>	100000	寄存器加
sub	000000	rs	rt	rd	<u>00000</u>	100010	寄存器减
and	000000	rs	rt	rd	<u>00000</u>	100100	寄存器与
or	000000	rs	rt	rd	<u>00000</u>	100101	寄存器或
xor	000000	rs	rt	rd	<u>00000</u>	100110	寄存器异或

指令功能: $\$rd \leftarrow \$rs \text{ op } \$rt$

1

R型指令

◆ 2寄存器R型指令

指令	[31:26]	[25:21]	[20:16]	[15:11]	[10:6]	[5:0]	指令功能
sll	000000	<u>00000</u>	rt	rd	sa	000000	逻辑左移
srl	000000	<u>00000</u>	rt	rd	sa	000010	逻辑右移
sra	000000	<u>00000</u>	rt	rd	sa	000011	算术右移

指令功能: $\$rd \leftarrow \$rt \text{ shift } sa$

1

R型指令

◆ 1寄存器R型指令

指令	[31:26]	[25:21]	[20:16]	[15:11]	[10:6]	[5:0]	指令功能
jr	000000	rs	<u>00000</u>	<u>00000</u>	<u>00000</u>	001000	寄存器跳转

jr rs;

 $PC \leftarrow rs$

2

I 型指令



操作数中涉及立即数，结果保存到寄存器；

- ◆ op: 标识指令的操作功能；
- ◆ rs: 第1个源操作数，是寄存器操作数；
- ◆ rt: 目的寄存器编号，用来保存运算结果；
- ◆ imm: 第2个源操作数，立即数；

2

I 型指令

指令	[31:26]	[25:21]	[20:16]	[15:0]	指令功能
addi	001000	rs	rt	imm	寄存器和立即数“加”
andi	001100	rs	rt	imm	寄存器和立即数“与”
ori	001101	rs	rt	imm	寄存器和立即数“或”
xori	001110	rs	rt	imm	寄存器和立即数“异或”
lw	100011	rs	rt	imm	从存储器中读取数据
sw	101011	rs	rt	imm	把数据保存到存储器
beq	000100	rs	rt	imm	寄存器相等则转移
bne	000101	rs	rt	imm	寄存器不等则转移
lui	001111	<u>00000</u>	rt	imm	设置寄存器的高16位

■ I型指令存在4种不同类型

◆ 面向运算的I型指令

指令	[31:26]	[25:21]	[20:16]	[15:0]	指令功能
addi	001000	rs	rt	imm	寄存器和立即数 “加”
andi	001100	rs	rt	imm	寄存器和立即数 “与”
ori	001101	rs	rt	imm	寄存器和立即数 “或”
xori	001110	rs	rt	imm	寄存器和立即数 “异或”

addi/andi/ori/xori rt, rs, imm; # $\$rt \leftarrow \$rs \text{ op } E(\text{imm})$

第一条指令是进行符号扩展，其余是0扩展

◆ 面向访存的I型指令

指令	[31:26]	[25:21]	[20:16]	[15:0]	指令功能
lw	100011	rs	rt	imm	从存储器中读取数据
sw	101011	rs	rt	imm	把数据保存到存储器

MIPS32中唯一两条访问存储器的指令(RISC)

lw rt, imm(rs) # $\$rt \leftarrow \text{mem}[\$rs + E(\text{imm})]$

sw rt, imm(rs) # $\text{mem}[\$rs + E(\text{imm})] \leftarrow \rt

◆ 面向数位设置的I型指令

指令	[31:26]	[25:21]	[20:16]	[15:0]	指令功能
lui	001111	<u>00000</u>	rt	imm	设置寄存器的高16位

lui rt, imm # $\$rt \leftarrow \text{imm} \ll 16$ (空位补0)

◆ 面向条件转移(分支)的I型指令

指令	[31:26]	[25:21]	[20:16]	[15:0]	指令功能
beq	000100	rs	rt	imm	寄存器相等则转移
bne	000101	rs	rt	imm	寄存器不等则转移

beq rs, rt, imm #if(\$rs==\$rt) $PC \leftarrow PC + E(imm) \ll 2$

bne rs, rt, imm #if(\$rs!=\$rt) $PC \leftarrow PC + E(imm) \ll 2$

是标准的PC相对寻址方式

其中imm要先“带符号扩展”成32位，再左移2位。

指令	[31:26]	[25:0]	指令功能
j	000010	address	无条件跳转
jal	001100	address	调用与联接

j address; $\$PC \leftarrow (\$PC + 4)_{H4} \cup (\text{address} \ll 2)$

jal address; $\$ra \leftarrow \$PC + 4$ (保存返回地址)
 $\$PC \leftarrow (\$PC + 4)_{H4} \cup (\text{address} \ll 2)$

4

关于MIPS指令的学习建议

- MIPS CPU设计过程中数据通路的设计与指令的执行流程密切相关
- 每条指令更为详细的功能，请课后查阅MIPS指令手册
- MIPS 指令学习过程中，建议配合使用MIPS汇编器和运行模拟器Mars，直观体验指令执行的流程。