第三章 指令

计算机的机器语言

程序的三大结构

- 顺序
- 分支
- 循环

- 分支跳转类指令
  - 分支类Branch
  - 跳转类Jump

分支跳转指令

- Branch If Equal (beq)：相等时转移
  - `beq reg1,reg2,label`
  - 如果reg1值=reg2值，则转移至label处执行
- Branch If Not Equal (bne)：不等时转移
  - `bne reg1,reg2,label`
  - 如果reg1的值 ≠ reg2的值，则转移至label处执行
- Jump (j)：跳转指令
  - `j label`
  - 无条件转移至label处执行
  - 经常需要配合对应C的goto机制

beq构造C代码if-else

C代码：

```
if(i==j) {
   a = b  /* then */
}
else {
   a = -b /* else */
}
```

执行逻辑

- 如果TRUE, 执行THEN语句块
- 否则，执行ELSE语句块

MIPS (beq):

```
# i→$s0, j→$s1
# a→$s2, b→$s3

beq $s0,$s1,then
else:
   sub $s2, $0, $s3
???
j    end
then:
add $s2, $s3, $0
end:
```

??? ← 可以去除该标号

用bne构造if-else

C代码：

```
if(i==j) {
   a = b  /* then */
} else {
   a = -b /* else */
}
```

执行逻辑

- 如果TRUE, 执行THEN语句块
- 如果FALSE，执行ELSE语句块

MIPS (bne):

```
# i→$s0, j→$s1
# a→$s2, b→$s3

bne $s0,$s1,???
???
add $s2, $s3, $0
j   end
else:
sub $s2, $0, $s3
end:
```

与C不同，beq/bne构造if-else条件为TRUE时则转移！

循环

- C语言有3种循环
  - for/while/do…while
  - 3种语句是等价的
- MIPS只需要一种决策机制即可
  - 核心：根据条件转移

循环举例

- C 代码:

```
while (save[i] == k) i += 1;
```

  - i→ $s3, k→ $s5, save[]的基地址在$s6

- MIPS :

```
Loop: sll  $t1, $s3, 2
      add  $t1, $t1, $s6
      lw   $t0, 0($t1)
      bne  $t0, $s5, Exit
      addi $s3, $s3, 1
      j    Loop
Exit: …
```

不等式

- 不等关系：<，<=，>，>=
  - X86是借助于一个叫标志寄存器flag的硬件对用的标志位来实现的
  - MIPS：无标志寄存器！！！
  - 人为借助寄存器作为标志位，通过增加1条额外的指令来支持所有的比较
- Set on Less Than (slt)
  - `slt dst,src1,src2`
  - 如果src1 < src2，dst写入1，否则写入0
- 与bne, beq, and $0组合即可实现所有的比较

不等式

- 实现<

C代码	MIPS汇编

if (a < b) {
... /* then */
}
(假设：a→$s0, b→$s1)
	slt $t0,$s0,$s1
	# $t0=1 if a<b
	# $t0=0 if a>=b
	bne $t0, $0,then
	# go to then
	# if $t0≠0

不等式

- 实现>=

C代码	MIPS汇编

if (a >= b) {
... /* then */
}
(假设：a→$s0, b→$s1)
	slt $t0,$s0,$s1
	# $t0=1 if a<b
	# $t0=0 if a>=b
	beq $t0, $0,then
	# go to then
	# if $t0=0

- Q：请自行完成
  - 交换src1与src2
  - 分别用beq与bne

不等式

- slt的3种变形
  - sltu dst,src1,src2: 无符号数比较
  - slti dst,src,imm: 与常量比较
  - sltiu dst,src,imm: 与无符号常量比较
- 示例：

```
addi  $s0,$0,-1 #$s0=0xFFFFFFFF
slti  $t0,$s0,1   # $t0=1
sltiu $t1,$s0,1   # $t1=0
```

不等式

原条件	等价条件	指令用法	结果寄存器的0/1值含义
a < b		slt 结果寄存器, a, b	0：原条件为假 1：原条件为真
a > b	b < a	slt 结果寄存器, b, a	0：原条件为假 1：原条件为真
a <= b	$\overline{b < a}$	slt 结果寄存器, b, a	0：原条件为真 1：原条件为假
a >= b	$\overline{a < b}$	slt 结果寄存器, a, b	0：原条件为真 1：原条件为假

更多分支指令

- 与0比较的分支指令（简称bxxz类指令）
  - blez（branch less or equal than zero小于等于0转移)
  - bltz（branch less than zero小于0转移）
  - bgtz（branch greater than zero大于0转移）
  - bgez（branch greater or equal than zero大于等于0转移）
- bxxz类指令格式
  - `bxxz rs, label`
  - 由于bxxz指令固定与$0比较，因此指令中无需再描述$0了

小测试

- 根据汇编程序，请选择正确的C代码填入空白处

```
do{i--;
} while(__________);
```

```
Loop:                   # i→$s0, j→$s1
addi $s0,$s0,-1         # i = i - 1
slti $t0,$s1,2          # $t0 = (j < 2)
beq  $t0,$0 ,Loop       # goto Loop if $t0==0
slt  $t0,$s1,$s0        # $t0 = (j < i)
bne  $t0,$0 ,Loop       # goto Loop if $t0!=0
```

a. j ≥ 2 || j < i

b. j ≥ 2 && j < i

c. j < 2 || j ≥ i

d. j < 2 && j ≥ i

- Thanks ！