

计算机组成原理与系统结构

第四章

运算方法与运算器

<http://jpkc.hdu.edu.cn/computer/zcyl/dzkjdx/>

2859.html

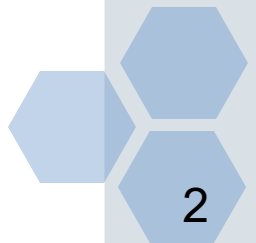




五、MIPS 的加减法指令以及溢出判断

1、指令

- ❖ 加（`add`）、立即数加（`addi`）、减法（`sub`）溢出时引起中断；
- ❖ 无符号数加（`addu`）、无符号数立即数加（`addiu`）、无符号数减法（`subu`）在溢出时不引起中断。





MIPS 的加减法运算举例

❖ `add $t0, $t1, $t2`

`$t0 = $t1 + $t2;`

有符号数的加法运算

❖ `sub $t2, $t3, $t4`

`$t2 = $t3 - $t4`

有符号数的减法运算

❖ `addi $t2, $t3, 5`

`$t2 = $t3 + 5;`
算

操作数是立即数的加法运算

❖ `addu $t1, $t6, $t7`

`$t1 = $t6 + $t7;`

无符号数的加法运算

❖ `subu $t1, $t6, $t7`

`$t1 = $t6 - $t7;`

无符号数的减法运算



MIPS 的加减法指令以及溢出判断

2、测试溢出程序举例

① 无符号数加法 ($\$t0 = \$t1 + \$t2$), 判溢出测试程序:

❖ `addu $t0, $t1, $t2`

❖ `nor $t3, $t1, $zero`

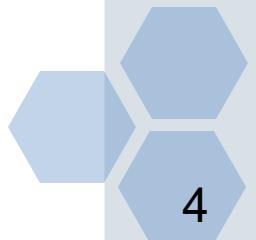
$\# \$t3 = \sim(\$t1 \mid 0)$ 按位或非, 即 $\$t3 = \text{not } \$t1 = 2^{32} - 1 - \$t1$

❖ `sltu $t3, $t3, $t2`

$\# \text{ if } (\$t3 = 2^{32} - 1 - \$t1) < \$t2 \text{ 即 } (2^{32} - 1) < (\$t1 + \$t2)$,

$\# \text{ then } \$t3 = 1; \text{ else } \$t3 = 0$

❖ `bne $t3, $zero, Overflow`





MIPS 的加减法指令以及溢出判断

② 有符号数加法 ($\$t0 = \$t1 + \$t2$), 判溢出测试程序:

- ❖ `addu $t0, $t1, $t2` # $\$t0 = \text{sum}$, 但是没有陷入异常
- ❖ `xor $t3, $t1, $t2` # 检查 2 个操作数符号是否不同
- ❖ `slt $t3, $t3, $zero` # 小于比较, 补码表示, if $\$t3 < 0$, then $\$t3 = 1$, else $\$t3 = 0$, 即符号不同 $\$t3 = 1$
- ❖ `bne $t3, $zero, No_overflow`
if $\$t3 \neq 0$, then go to No_overflow (No_overflow 的地址是 $PC \leftarrow PC + 4 +$ 符号位扩展



MIPS 的加减法指令以及溢出判断

❖ `xor $t3, $t0, $t1`

如果 `$t1` 和 `$t2` 符号相同，那么它们与和的符号也相同吗？如果不同，则 `$t3 < 0`

❖ `slt $t3, $t3, $zero`

如果操作数的符号相同，而结果与操作数符号不同，则置 `$t3=0`

❖ `bne $t3, $zero, verflow`

if `$t3 ≠ 0`, then go to overflow, 即和与操作数符号不同，则溢出

