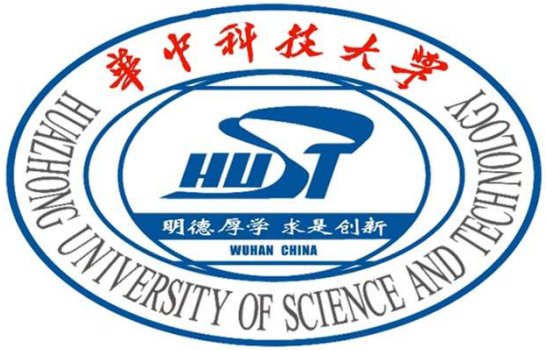
微机原理与接口技术

MIPS运算类指令

华中科技大学　左冬红



运算类指令

- 算术运算类指令
  - 加、减、乘、除
  - 符号数运算
  - 无符号数运算
  - 立即数参与运算
- 逻辑运算类指令
  - 与、或、或非、异或、左移、右移
  - 都是位运算
  - 立即数参与运算时，扩展0

加、减运算

指令格式	指令功能	
add $Rd,$Rs,$Rt	RF[$Rd]=RF[$Rs]+RF[$Rt]	OF
addu $Rd,$Rs,$Rt	RF[$Rd]=RF[$Rs]+RF[$Rt]	
addi $Rt,$Rs,Imm	RF[$Rt]=RF[$Rs]+{D_{15}?16'hffff:16'h0,Imm}	
addiu $Rt,$Rs,Imm	RF[$Rt]=RF[$Rs]+{16'h0,Imm}	
sub $Rd,$Rs,$Rt	RF[$Rd]=RF[$Rs]-RF[$Rt]	OF
subu $Rd,$Rs,$Rt	RF[$Rd]=RF[$Rs]-RF[$Rt]	

乘法指令

指令格式	指令功能
mult $Rs,$Rt	RF[hi]=(RF[$Rs]×RF[$Rt])高32位
$Rs,$Rt符号数	RF[lo]=(RF[$Rs]×RF[$Rt])低32位
multu $Rs,$Rt	RF[hi]=(RF[$Rs]×RF[$Rt])高32位
$Rs,$Rt无符号数	RF[lo]=(RF[$Rs]×RF[$Rt])低32位

差别在哪里呢？

乘法运算方式与加、减法不同，结果符号由参与运算的数的符号异同决定，数值部分由两数绝对值通过移位和加法运算完成

除法指令

指令格式	指令功能
div $Rs,$Rt	RF[hi]=(RF[$Rs]÷RF[$Rt])余数
$Rs,$Rt符号数	RF[lo]=(RF[$Rs]÷RF[$Rt])商
divu $Rs,$Rt	RF[hi]=(RF[$Rs]÷RF[$Rt])余数
$Rs,$Rt无符号数	RF[lo]=(RF[$Rs]÷RF[$Rt])商

商的符号由参与运算的数的符号异同决定，数值部分由两数绝对值通过移位和减法运算完成，余数符号与被除数一致

算术运算指令示例

采用MIPS汇编语言指令完成以下C语言语句功能：
C=A[2]-B[3]-5; //A,B,C都是int型

假定$s0,$s1,$s2分别对应变量C，数组A,B的首地址

临时寄存器存储A[2]的值	lw $t0,8($s1)
临时寄存器存储B[3]的值	lw $t1,12($s2)
执行运算A[2]-B[3]	sub $t2,$t0,$t1
执行运算-5	addi $s0,$t2,-5

基本逻辑运算指令

指令格式	指令功能
and $Rd,$Rs,$Rt	RF[$Rd]=RF[$Rs]位与RF[$Rt]
andi $Rd,$Rs,Imm	RF[$Rd]=RF[$Rs]位与{16'h0,Imm}
or $Rd,$Rs,$Rt	RF[$Rd]=RF[$Rs]位或RF[$Rt]
ori $Rd,$Rs,Imm	RF[$Rd]=RF[$Rs]位或{16'h0,Imm}
nor $Rd,$Rs,$Rt	RF[$Rd]=RF[$Rs]位或非RF[$Rt]
xor $Rd,$Rs,$Rt	RF[$Rd]=RF[$Rs]位异或RF[$Rt]
xori $Rd,$Rs,Imm	RF[$Rd]=RF[$Rs]位异或{16'h0,Imm}

基本逻辑运算指令功能

and,or,nor,xor

and
1 1 0 0 1 0 1 0
1 1 1 1 0 0 0 0
1 1 0 0 0 0 0 0

与1不变，与0清0

or
1 1 0 0 1 0 1 0
1 1 1 1 0 0 0 0
1 1 1 1 1 0 1 0

或1置1，或0不变

nor
1 1 0 0 1 0 1 0
1 1 1 1 0 0 0 0
0 0 0 0 0 1 0 1

或非1清0，或非0取反

xor
1 1 0 0 1 0 1 0
1 1 1 1 0 0 0 0
0 0 1 1 1 0 1 0

异或1取反，异或0不变

基本逻辑运算指令

逻辑运算类指令常用来进行数据不同位的拆分和组合

将$t0的高7位($D_{31}$~$D_{25}$)与$t1的中间8位(D_{24}~D_{17})以及$t2的低17位($D_{16}$~$D_0$)取反之后合并为一个新的32位数，各数据位所处位置不变。

如何保留$t0的高7位($D_{31}$~$D_{25}$)？

and,or,nor,xor

如何保留$t1的中间8位($D_{24}$~$D_{17}$)？

如何将$t2的低17位($D_{16}$~$D_0$)取反？

如何合并多个寄存器不同数据位的数据？

基本逻辑运算指令

and,or,nor,xor

将$t0的高7位($D_{31}$~$D_{25}$)与$t1的中间8位(D_{24}~D_{17})以及$t2的低17位($D_{16}$~$D_{0}$)取反之后合并为一个新的32位数，各数据位所处位置不变。

如何保留$t0的高7位($D_{31}$~$D_{25}$)？

andi $t0,$t0,0xfe000000

如何保留$t1的中间8位($D_{24}$~$D_{17}$)？

andi $t1,$t1,0x01fe0000

?

```
and  1 1 0 0 1 0 1 0
     1 1 1 1 0 0 0 0
     ---------------
     1 1 0 0 0 0 0 0
```

andi指令仅能保留低16位中的某些位

将立即数赋值到寄存器的高16位

```
lui $t3,0xfe00
and $t0,$t0,$t3
```

```
lui $t3,0x01fe
and $t1,$t1,$t3
```

基本逻辑运算指令

and,or,nor,xor

将$t0的高7位($D_{31}$~$D_{25}$)与$t1的中间8位(D_{24}~D_{17})以及$t2的低17位($D_{16}$~$D_0$)取反之后合并为一个新的32位数，各数据位所处位置不变。

如何将$t2的低17位($D_{16}$~$D_0$)取反？

不存在not

哪些指令可执行not功能？

nor,xor

$t2=(~ ($t2||0) &&0x1ffff)

nor不能与立即数直接逻辑运算

nor

```
nor $t2,$t2,$0
lui $t3,0x1
ori $t3,$t3,0xffff
and $t2,$t2,$t3
```

xor

```
lui $t3,0x1
ori $t3,$t3,0xffff
xor $t2,$t2,$t3
and $t2,$t2,$t3
```

基本逻辑运算指令

and,or,nor,xor

将$t0的高7位($D_{31}$~$D_{25}$)与$t1的中间8位(D_{24}~D_{17})以及$t2的低17位($D_{16}$~$D_0$)取反之后合并为一个新的32位数，各数据位所处位置不变。

如何合并多个寄存器不同数据位的数据？

xxxx xxx	**0 0000 0000 0000 0000 0000 0000**
0000 000	**y yyyy yyy0 0000 0000 0000 0000**
0000 000	**0 0000 000z zzzz zzzz zzzz zzzz**

or

```
or $s0,$t0,$t1
or $s0,$s0,$t2
```

基本逻辑运算指令

将$t0的高7位($D_{31}$~$D_{25}$)与$t1的中间8位(D_{24}~D_{17})以及$t2的低17位($D_{16}$~$D_0$)取反之后合并为一个新的32位数，各数据位所处位置不变。

```
lui $t3,0xfe00
and $t0,$t0,$t3
```

```
lui $t3,0x01fe
and $t1,$t1,$t3
```

```
lui $t3,0x1
ori $t3,$t3,0xffff
xor $t2,$t2,$t3
and $t2,$t2,$t3
```

```
nor $t2,$t2,$0
lui $t3,0x1
ori $t3,$t3,0xffff
and $t2,$t2,$t3
```

```
or $s0,$t0,$t1
or $s0,$s0,$t2
```

移位指令

RF[$Rt]、Imm<32 移入补0

指令格式	指令功能
sllv $Rd,$Rs,$Rt	RF[$Rd]=RF[$Rs]<<RF[$Rt]
sll $Rd,$Rt,Imm	RF[$Rd]=RF[$Rs]<<Imm
srlv $Rd,$Rs,$Rt	RF[$Rd]=RF[$Rs]>>RF[$Rt]
srl $Rd,$Rt,Imm	RF[$Rd]=RF[$Rs]>>Imm
srav $Rd,$Rs,$Rt	RF[$Rd]=RF[$Rs]>>RF[$Rt]
sra $Rd,$Rt,Imm	RF[$Rd]=RF[$Rs]>>Imm

$\times 2^{Imm}$、$RF[\$Rt]$

$\div 2^{Imm}$、$RF[\$Rt]$

R型指令中的移位指令

移入符号位

移位指令示例

将$t0的高7位($D_{31}$~$D_{25}$)存储到$t1的中间7位(D_{24}~D_{18})，且$t1的其余数据位值不变。

$t0右移7位，仅保留$D_{24}$~$D_{18}$，$t1保留D_{31}~D_{25}以及D_{17}~D_0再与$t1其余数据位合并

$t0右移7位	srl $t0,$t0,7
仅保留D_{24}~D_{18}	lui $t2,0x01fc and $t0,$t0,$t2
$t1保留$D_{31}$~$D_{25}$以及$D_{17}$~$D_0$	lui $t2,0xfe03 ori $t2,$t2,0xffff and $t1,$t1,$t2
数据位合并	or $t1,$t1,$t0

小结

- 算术运算类指令
  - 加、减
    - 符号数运算
    - 无符号数运算
    - 立即数参与运算，数位扩展
  - 乘、除
    - 结果采用特殊寄存器暂存
- 基本逻辑运算类指令
  - 与、或、或非、异或、
  - 都是位运算
  - 立即数参与运算时，扩展0
- 移位运算
  - 左移、右移
    - 无符号数乘、除2的幂次
    - 移位次数不大于31
    - 都是R型指令

下一讲：程序控制类指令