MIPS 控制流指令

本节

- MIPS 条件转移指令
- MIPS 无条件转移指令

MIPS条件转移指令

指令举例	操作	说明
beq $1,$2,100	if($1==$2) go to PC+4+100	相等时转移
bne $1,$2,100	if($1!=$2) go to PC+4+100	不相等时转移
slt $1,$2,$3	if($2<$3) $1=1;else $1=0	小于时置位
slti $1,$2,100	if($2<100) $1=1;else $1=0	小于立即数时置位
sltu $1,$2,$3	if($2<$3) $1=1;else $1=0	小于无符号数时置位
sltiu $1,$2,100	if($2<100) $1=1;else $1=0	无符号数小于立即数时置位

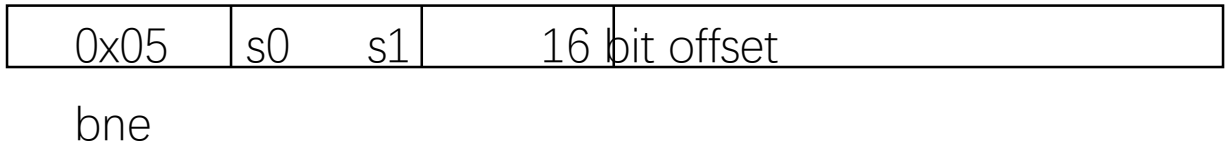
MIPS 条件转移指令

```
bne $s0, $s1, Lbl1  #go to Lbl1 if $s0≠$s1
beq $s0, $s1, Lbl1  #go to Lbl1 if $s0=$s1
```

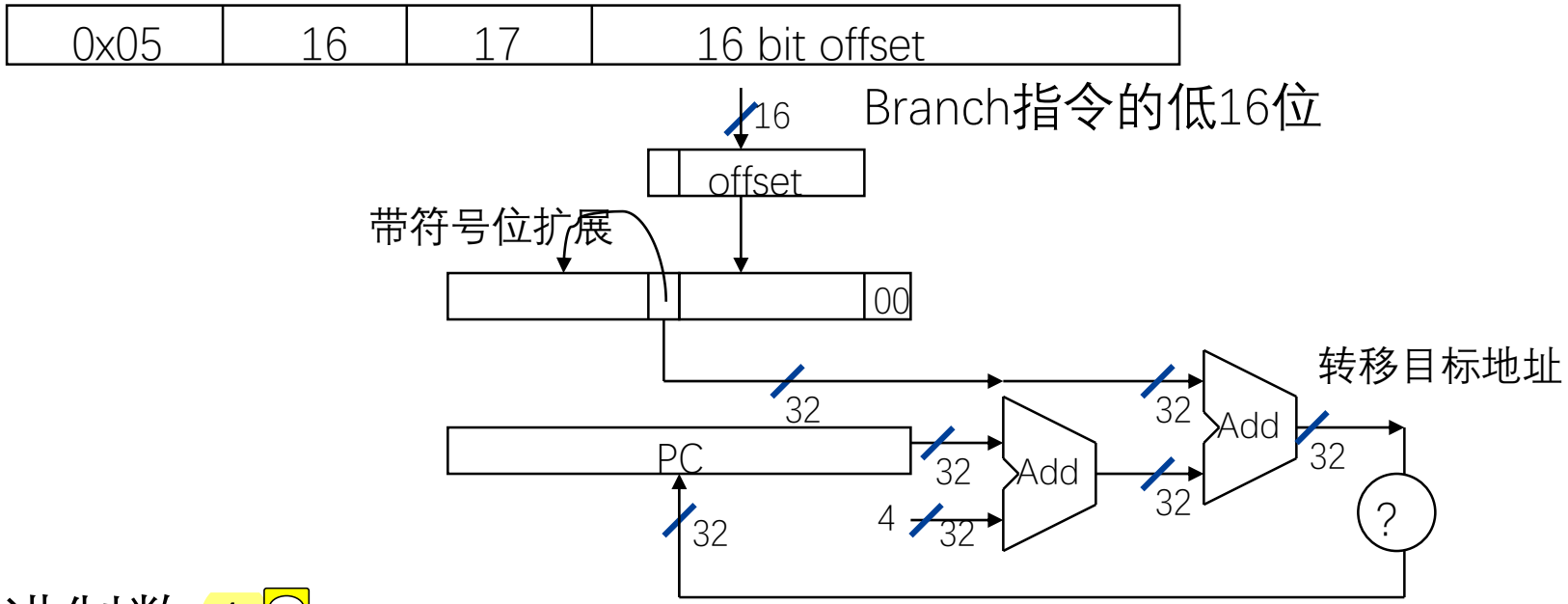
- 例如: `if (i==j) h = i + j;`

```
        bne $s0, $s1, Lbl1
        add $s3, $s0, $s1
Lbl1:   ...
```

- 指令格式 (I 型):



计算转移目标地址



- (PC+4) +16位二进制数*4
- PC+4：下一条要执行的指令
- 16位二进制数：跳转到的指令与当前指令的下一条指令之间间隔的指令条数
- 跳转范围：-2^{15} to $+2^{15}-1$ (word，字)，$2^{15}-1$ 个字 = $2^{17}-1$ （字节： bytes）

更多跳转指令

- 可以使用slt来设置跳转条件
  - 为了表达：blt $s1, $s2, Label，实际上使用：

```
slt  $at, $s1, $s2   #$at set to 1 if #$s1 < $s2
bne  $at, $zero, Label
```

- 类似slt的指令

```
slti $t0, $s0, 25        # if $s0 < 25 then $t0=1 ...
sltu $t0, $s0, $s1       # if $s0 < $s1 then $t0=1 ...
sltiu $t0, $s0, 25       # if $s0 < 25 then $t0=1 ...
```

- 不存在以下MIPS指令：
  - less than or equal to　　ble $s1, $s2, Label
  - greater than　　bgt $s1, $s2, Label
  - great than or equal to　　bge $s1, $s2, Label

MIPS无条件转移指令

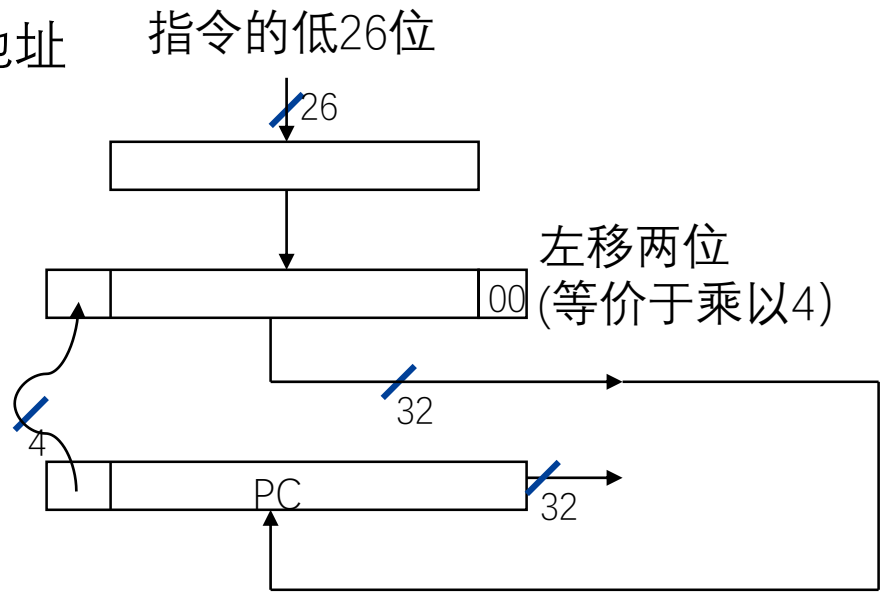
指令举例	操作	说明
j 10000	go to 10000	转移到 10000
jr $31	go to $31	转移到$31
jal 10000	$31=PC+4; go to 10000	转移并链接

无条件转移

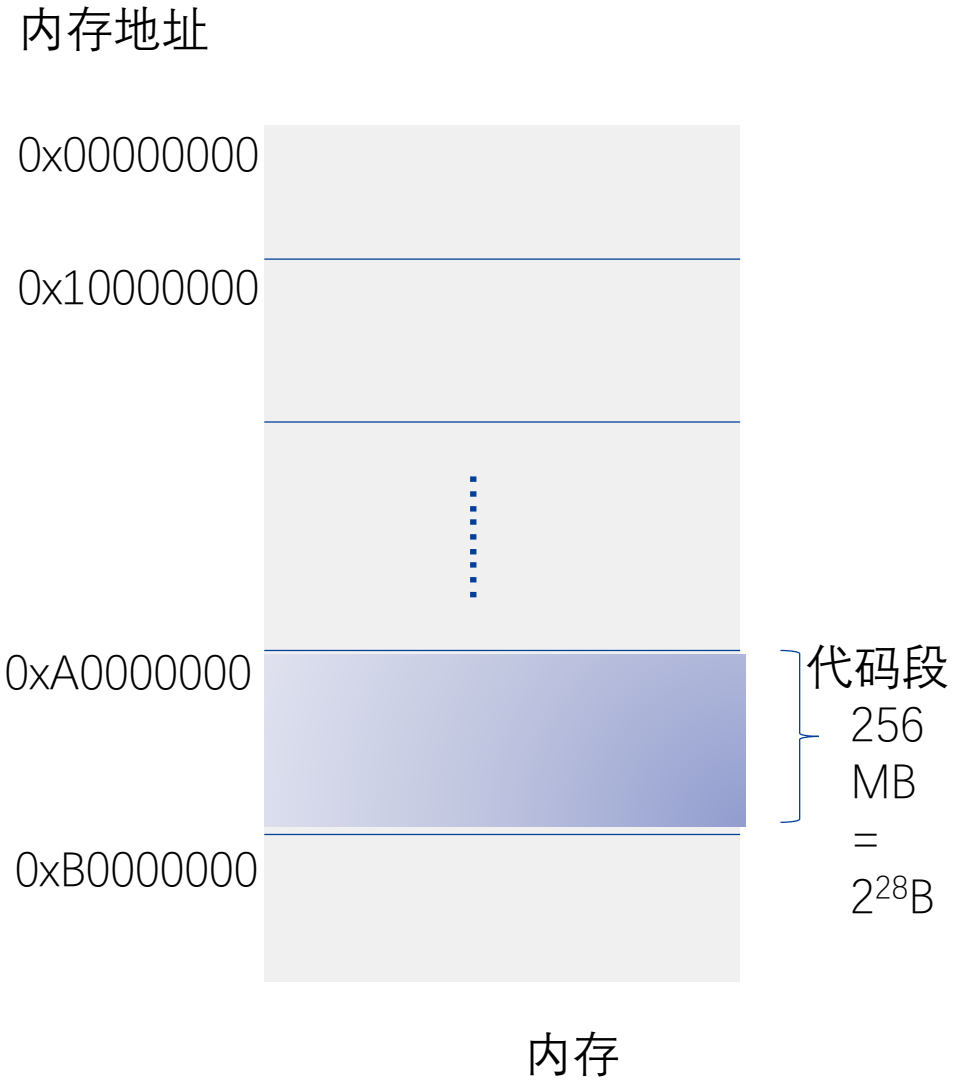
```
j   label       #go to label
```

- 指令格式(Jump 型):

- 计算转移目标地址



- 偏移量是绝对偏移量



跳转到更远的位置

- 如果跳转的位置超过了2^{15}-1个字的偏移范围
  - 编译器将以下指令进行转换
  - ```
    beq  $s0, $s1, L1
    ```

变为:

```
bne  $s0, $s1, L2
j L1

L2:
```
```

举例

- 假设：i 在寄存器$s3中， k 在寄存器 $s5中
- 数组 save 的首地址在寄存器 $s6中

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

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

左移两位，相当于乘4

计算数组元素的地址

读该数组元素

不等于就跳转到出口

修改数组元素所在的下标

跳转至循环开始

小结

主要介绍了以下指令:

- beq
- bne
- slt
- j

谢谢！