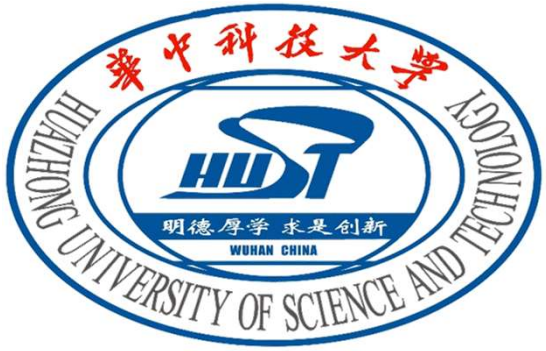
微机原理与接口技术

MIPS程序控制类指令应用

华中科技大学　左冬红



C程序控制语句

```
if(条件表达式)
    {……}
    else
        {……}
```

```
switch(变量)
{
    case xx:……
    case yy:……
    ……
    default:……
}
```

分支语句

```
    do
    {
    ……
    }
while(条件表达式);
```

```
while(条件表达式)
    {
    ……
    }
```

```
for(初始化;条件表达式;增量)
    {
    ……
    }
```

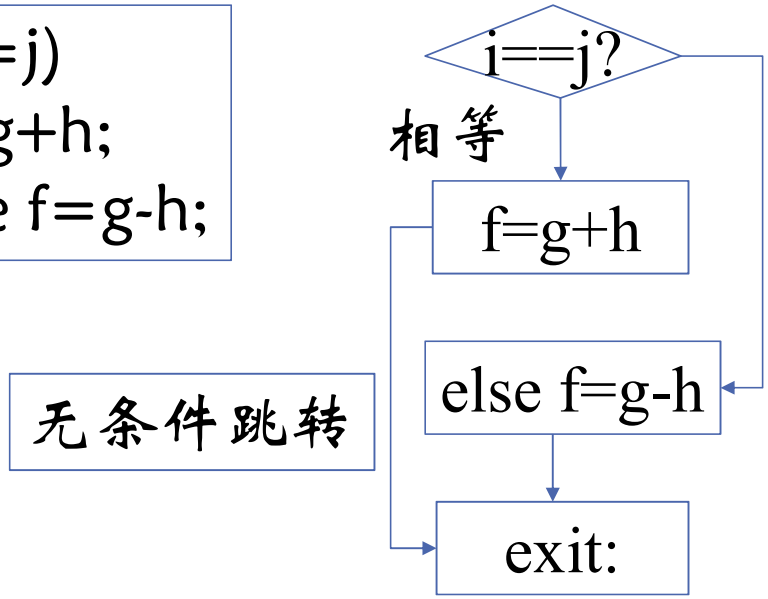
循环语句

if控制

文件中各语句存储顺序

MIPS汇编指令序列

```
if (i==j)
    f=g+h;
    else f=g-h;
exit:
```



无条件跳转

```
bne $s0,$s1,else
add $s2,$s3,$s4
j exit
else:sub $s2,$s3,$s4
exit:
```

i	j	f	g	h
$s0	$s1	$s2	$s3	$s4

条件跳转采用beq指令时
如何修改汇编指令序列？

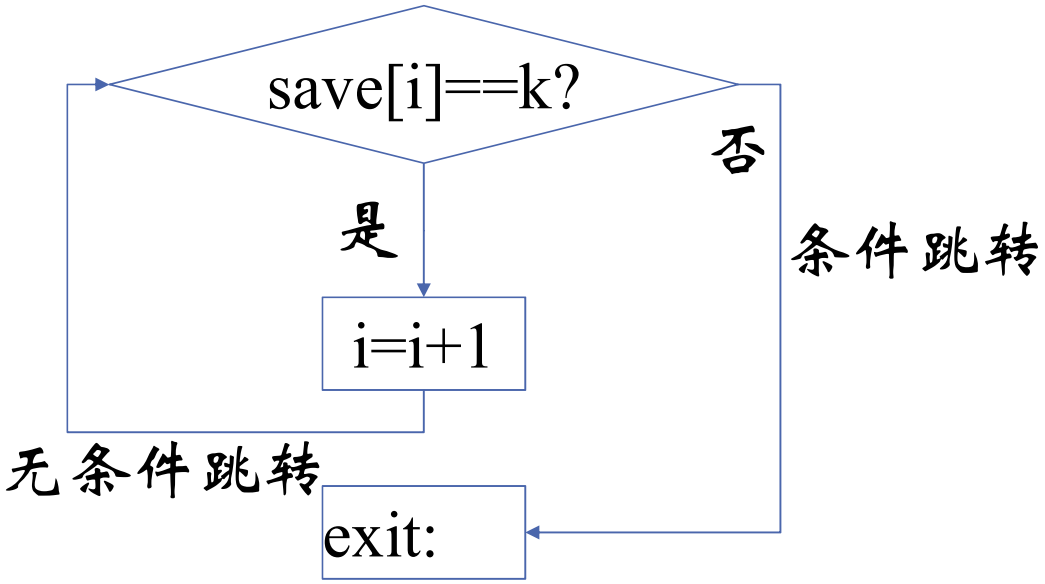
while语句

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

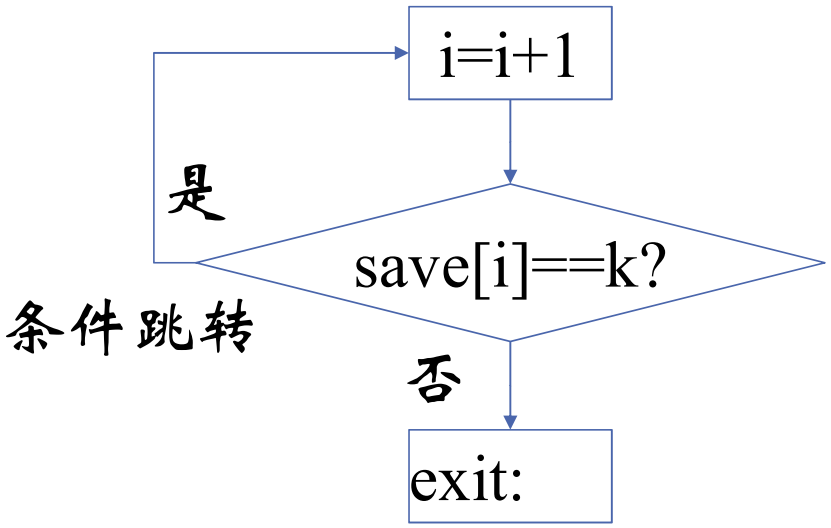
功能是否一致?

性能是否一致?

语句执行流程解释1



语句执行流程解释2



while语句

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

i	k	save
$s0	$s2	$s3

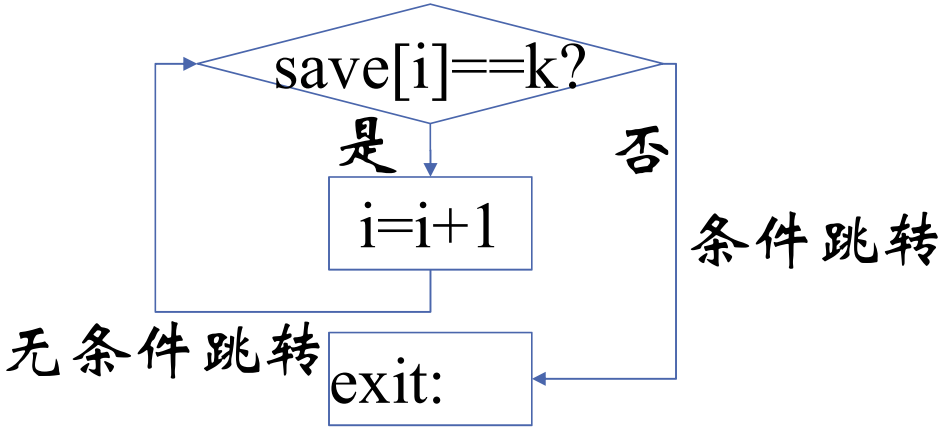
```
int save[];
```

save[i]能否直接与k比较？

如何获取save[i]的值？

&save[i]=save+4×i

语句执行流程解释1



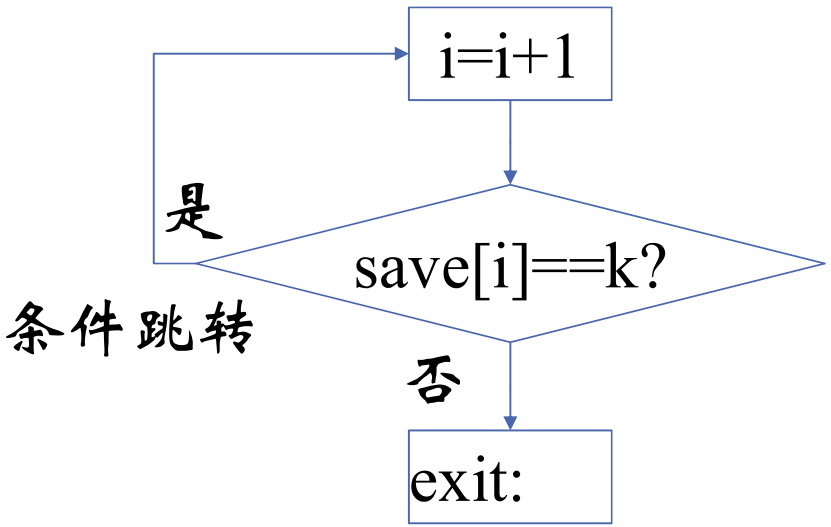

```
again: sll $t0,$s0,2
       add $t0,$t0,$s3
       lw $t0,0($t0)
       bne $t0,$s2,exit
       addi $s0,$s0,1
       j again
exit:
```

while语句

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

i	k	save
$s0	$s2	$s3

语句执行流程解释2



条件跳转

```
        addi $s0,$s0,-1
again:  addi $s0,$s0,1
        sll $t0,$s0,2
        add $t0,$t0,$s3
        lw $t0,0($t0)
        beq $t0,$s2,again
exit:
```

while语句

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

性能有差别吗？

重复执行的指令段

语句执行流程解释1

```
again: sll $t0,$s0,2
       add $t0,$t0,$s3
       lw $t0,0($t0)
       bne $t0,$s2,exit
       addi $s0,$s0,1
       j again
exit:
```

语句执行流程解释2

```
       addi $s0,$s0,-1
again: addi $s0,$s0,1
       sll $t0,$s0,2
       add $t0,$t0,$s3
       lw $t0,0($t0)
       beq $t0,$s2,again
exit:
```

更优

大小比较条件判断

beq、bne仅能判断相等、不等

小于0、小于等于0、大于0、大于等于0判断

bltz | blez | bgtz | bgez

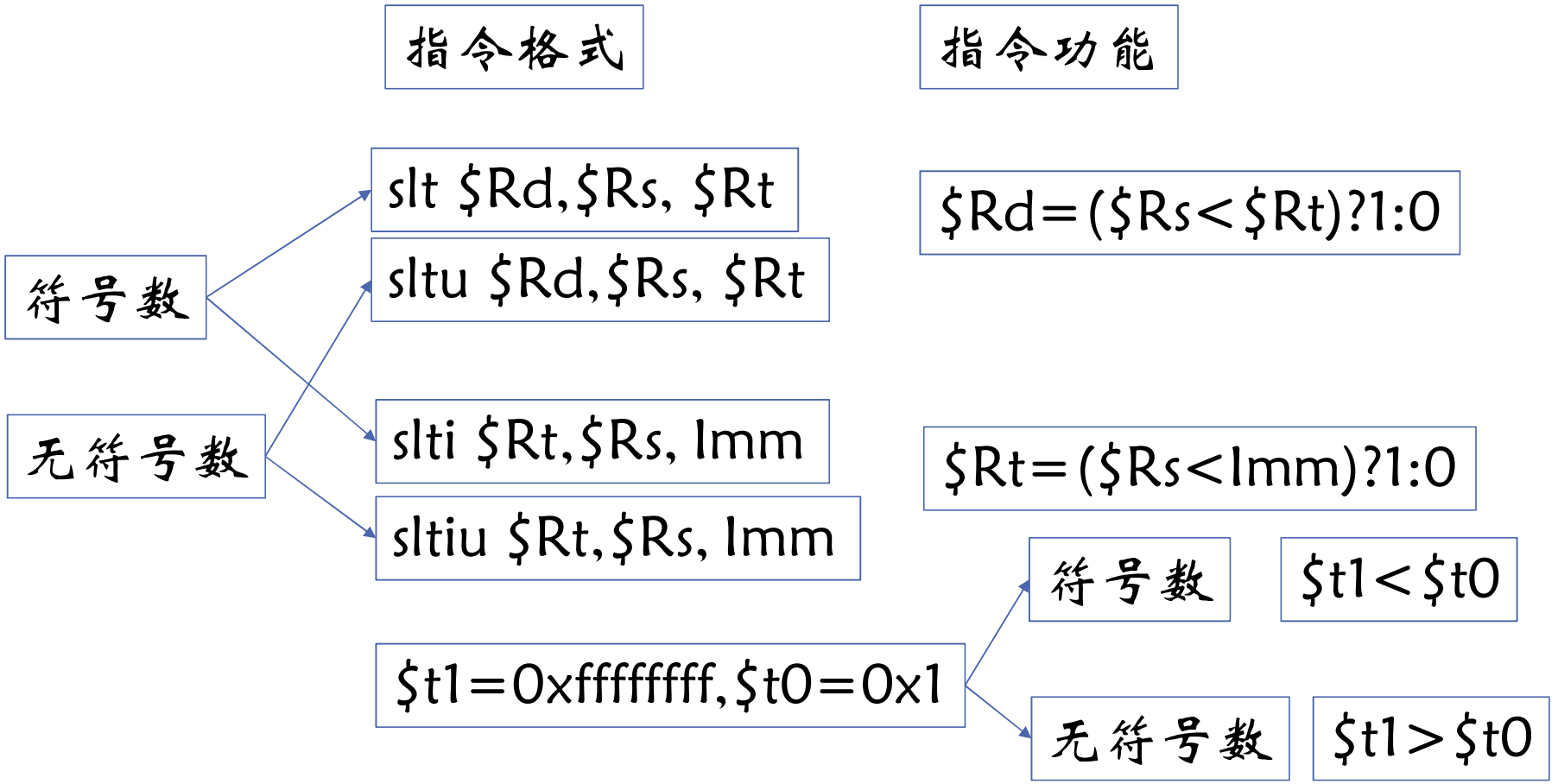
指令格式　Op $Rs,lable

i>j,i<j,i<=j;i>=j

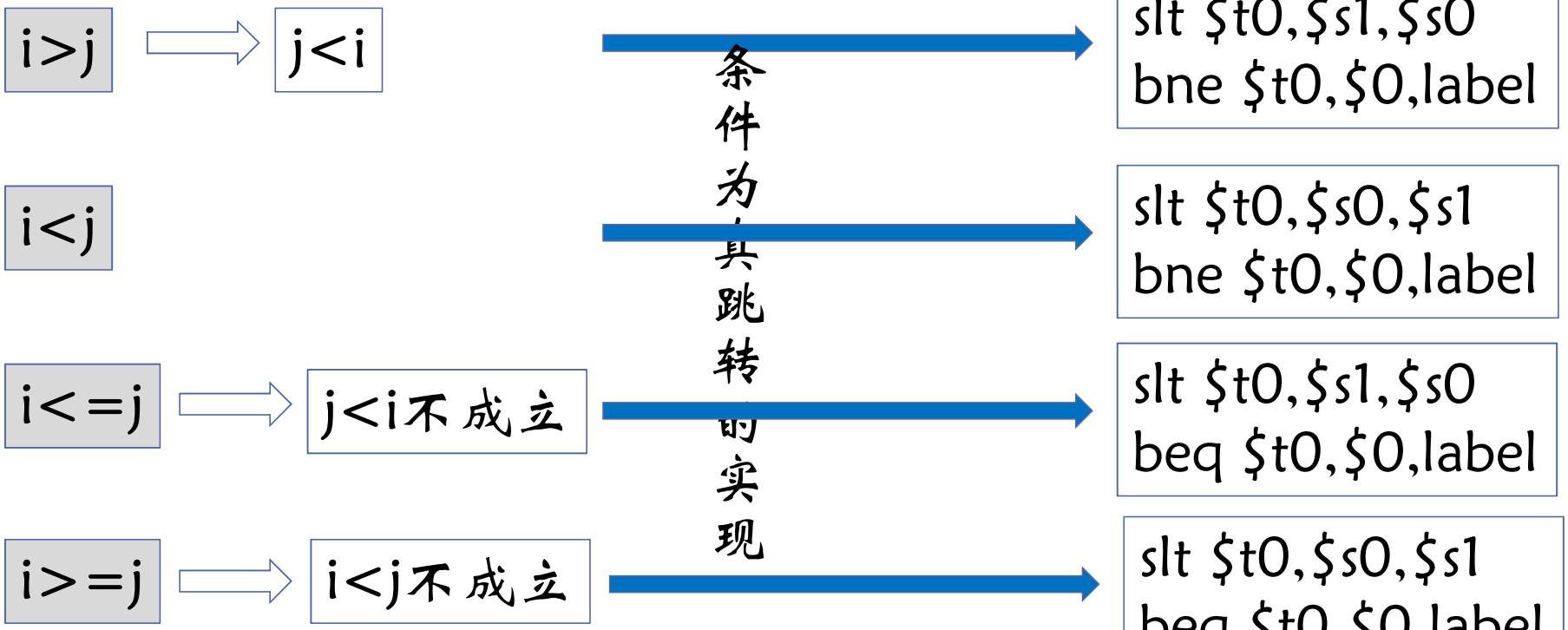
小于设置配合相等、不等比较跳转

slt、sltu、slti、sltiu

小于设置指令


指令格式
指令功能
符号数
无符号数
slt $Rd,$Rs, $Rt
sltu $Rd,$Rs, $Rt
slti $Rt,$Rs, Imm
sltiu $Rt,$Rs, Imm
$Rd=($Rs<$Rt)?1:0
$Rt=($Rs<Imm)?1:0
$t1=0xffffffff,$t0=0x1
符号数
$t1<$t0
无符号数
$t1>$t0

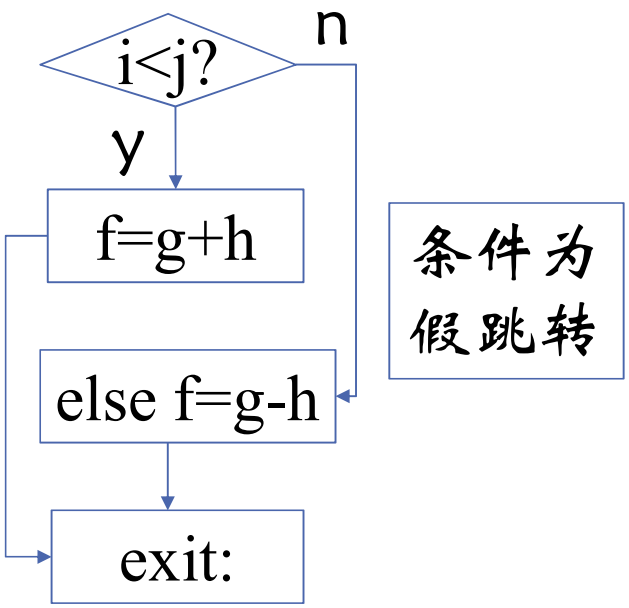
大小比较条件判断



i	j
$s0	$s1

大小比较条件判断

```
if (i<j)
    f=g+h;
    else f=g-h;
exit:
```



条件为假跳转

i	j	f	g	h
$s0	$s1	$s2	$s3	$s4

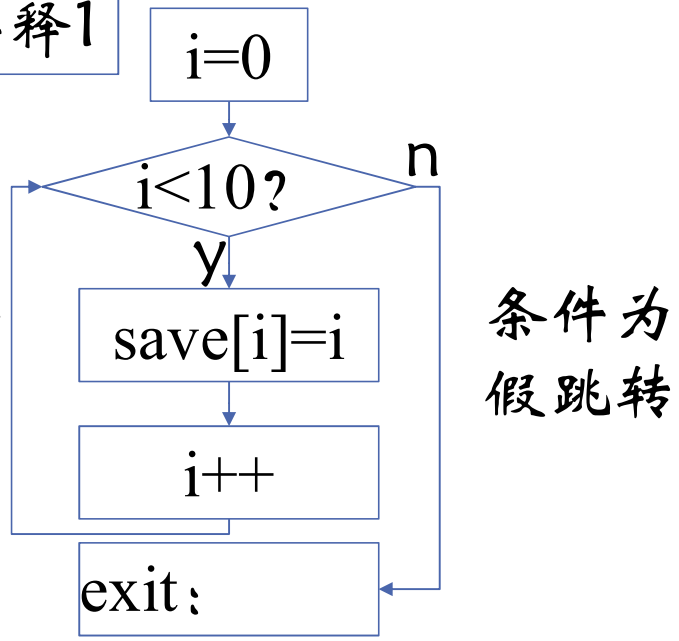
```
slt $t0,$s0,$s1
beq $t0,$zero,else
add $s2,$s3,$s4
j exit
else:sub $s2,$s3,$s4
exit:
```

for语句

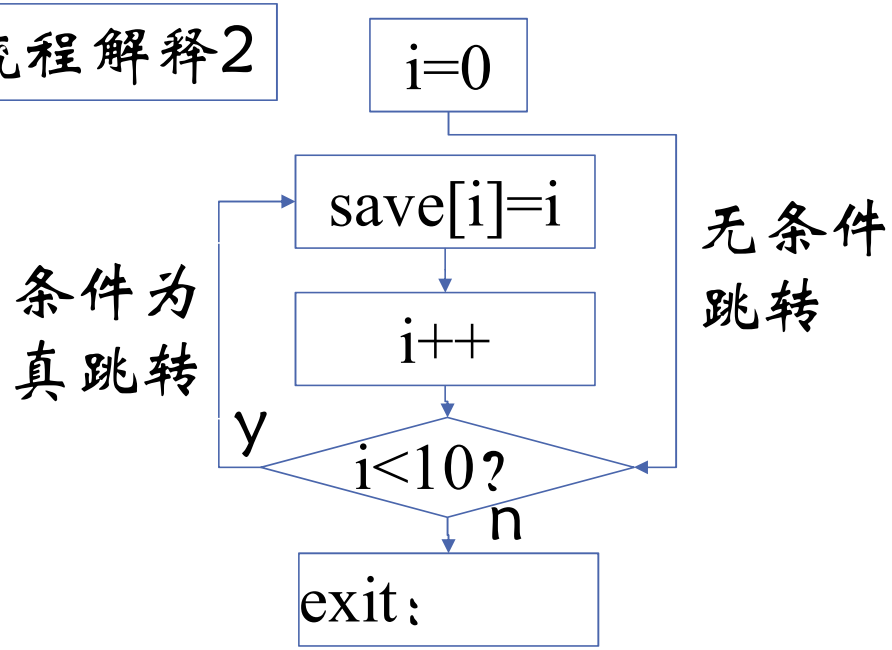
```
for (i=0;i<10;i++)
    save[i]=i;
exit:
```

两种流程功能、性能有差别吗？

流程解释1



流程解释2

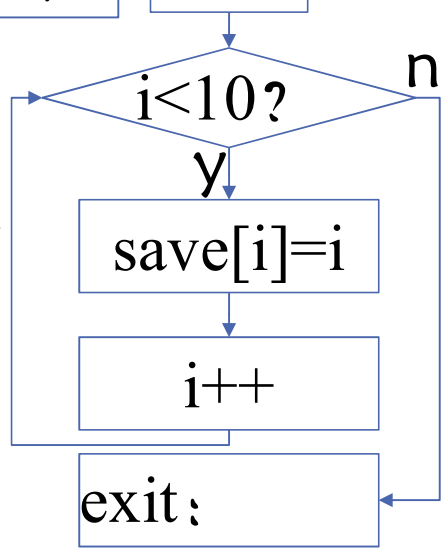


for语句

i	save
$s0	$s3

```
for (i=0;i<10;i++)
    save[i]=i;
exit:
```

流程解释1



无条件跳转

条件为假跳转

给寄存器赋初值0

```
lui $s0,0
add $s0,$0,$0
again: slti $t0,$s0,10
       beq $t0,$0,exit
       sll $t0,$s0,2
       add $t0,$t0,$s3
       sw $s0,0($t0)
       addi $s0,$s0,1
       j again
exit:
```

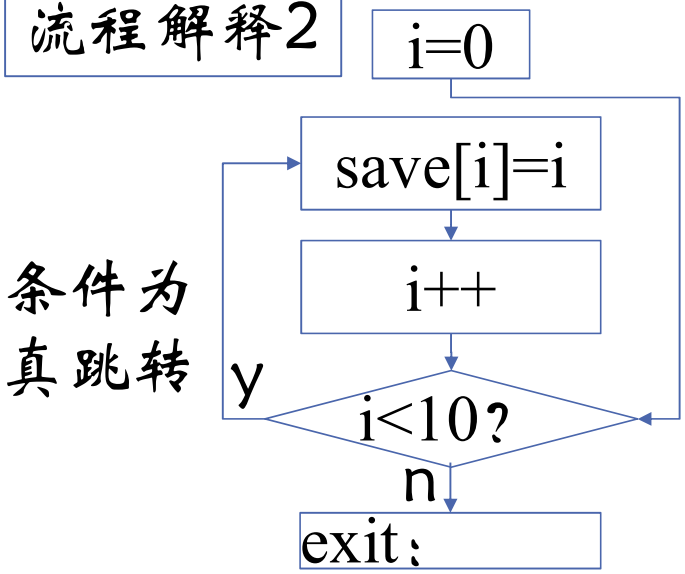
条件为假跳转 beq

for语句

```
for (i=0;i<10;i++)
    save[i]=i;
exit:
```

i	save
$s0	$s3

流程解释2



条件为真跳转 bne

```
    add $s0,$0,$0
    j check
again: sll $t0,$s0,2
    add $t0,$t0,$s3
    sw $s0,0($t0)
    addi $s0,$s0,1
check: slti $t0,$s0,10
    bne $t0,$0,again
exit:
```

for语句

两种流程功能、性能有差别吗？

流程解释1

```
    add $s0,$0,$0
again: slti $t0,$s0,10
       beq $t0,$0,exit
       sll $t0,$s0,2
       add $t0,$t0,$s3
       sw $s0,0($t0)
       addi $s0,$s0,1
       j again
exit:
```

循环执行语句

循环体内

流程解释2

```
    add $s0,$0,$0
    j check
again: sll $t0,$s0,2
       add $t0,$t0,$s3
       sw $s0,0($t0)
       addi $s0,$s0,1
check: slti $t0,$s0,10
       bne $t0,$0,again
exit:
```

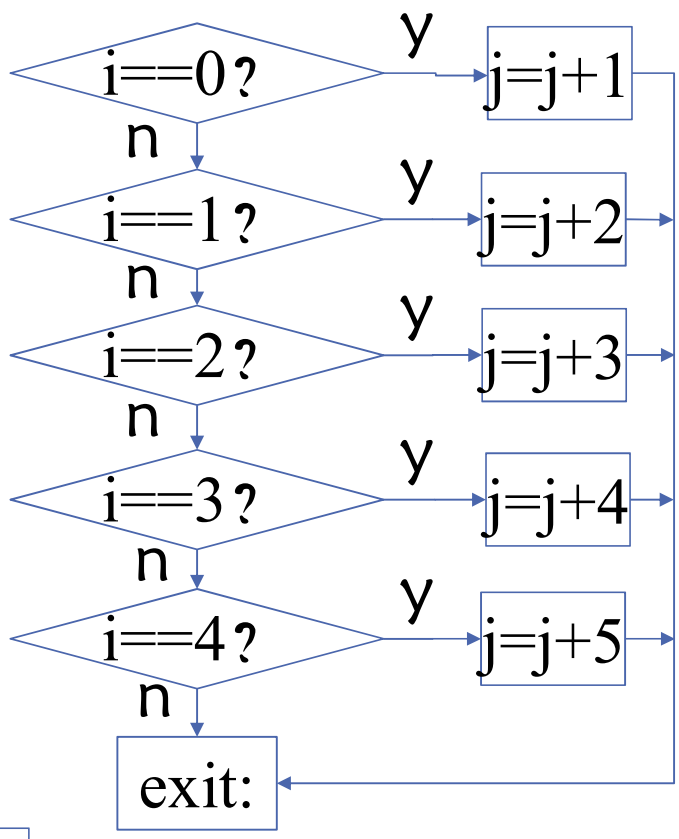
循环体外

更优

switch语句

```
switch (i)
{
case 0:j=j+1;break;
case 1:j=j+2;break;
case 2:j=j+3;break;
case 3:j=j+4;break;
case 4:j=j+5;break;
};
exit:
```

i:$s0, j:$s1, 0~4:$t0~$t4




```
    beq $s0,$t0,ca0
    beq $s0,$t1,ca1
    beq $s0,$t2,ca2
    beq $s0,$t3,ca3
    beq $s0,$t4,ca4
    j exit
ca0: addi $s1,$s1,1
    j exit
ca1: addi $s1,$s1,2
    j exit
ca2: addi $s1,$s1,3
    j exit
ca3: addi $s1,$s1,4
    j exit
ca4: addi $s1,$s1,5
exit:
```

switch语句

```
int address[5]={ca0,ca1,ca2,ca3,ca4}
```

```
switch (i)
{
case 0:j=j+1;break;
case 1:j=j+2;break;
case 2:j=j+3;break;
case 3:j=j+4;break;
case 4:j=j+5;break;
};
exit:
```

i:$s0, j:$s1, 0~4:$t0~$t4

address:$s3

判断i的合法性 0<=i<5

```
bltz $s0,exit
slti $t0,$s0,5
beq $t0,$0,exit
```

获取跳转目标地址

```
sll $t0,$s0,2
add $t0,$t0,$s3
lw $t0,0($t0)
```

跳转到目标地址

```
jr$t0
```

```
ca0: addi $s1,$s1,1
     j exit
ca1: addi $s1,$s1,2
     j exit
ca2: addi $s1,$s1,3
     j exit
ca3: addi $s1,$s1,4
     j exit
ca4: addi $s1,$s1,5
exit:
```

switch语句

case少优

增加beq语句

比较跳转

case数目增多时程序段是否需发生变化?

```
    beq $s0,$t0,ca0
    beq $s0,$t1,ca1
    beq $s0,$t2,ca2
    beq $s0,$t3,ca3
    beq $s0,$t4,ca4
    j exit
ca0: addi $s1,$s1,1
    j exit
ca1: addi $s1,$s1,2
    j exit
ca2: addi $s1,$s1,3
    j exit
ca3: addi $s1,$s1,4
    j exit
ca4: addi $s1,$s1,5
exit:
```

case多优

查表跳转

访问存储器，效率低

```
    bltz $s0,exit
    slt $t0,$s0,5
    beq $t0,$0,exit
    sll $t0,$s0,2
    add $t0,$t0,$s3
    lw $t0,0($t0)
    jr$t0
ca0: addi $s1,$s1,1
    j exit
ca1: addi $s1,$s1,2
    j exit
ca2: addi $s1,$s1,3
    j exit
ca3: addi $s1,$s1,4
    j exit
ca4: addi $s1,$s1,5
exit:
```

小结

- if、while、for、switch语句
  - 汇编实现
  - 优化
- 大于、小于、大于等于、小于等于条件判断
  - slt、bne、beq

下一讲：子程序原理