

Linguagem de montagem

4. Instruções de desvio e laços

Prof. John L. Gardenghi

Adaptado dos slides do livro

Instruções de desvio

- Desvio condicional
 - desvia para uma instrução rotulada se a condição for verdadeira
 - senão, continua sequencialmente
 - `beq rs, rt, L1`
 - se $(rs == rt)$ desvia para a instrução L1;
 - `bne rs, rt, L1`
 - se $(rs != rt)$ desvia para a instrução L1;
- Desvio incondicional
 - `j L1`
 - desvia incondicionalmente para a instrução L1

Compilando ifs

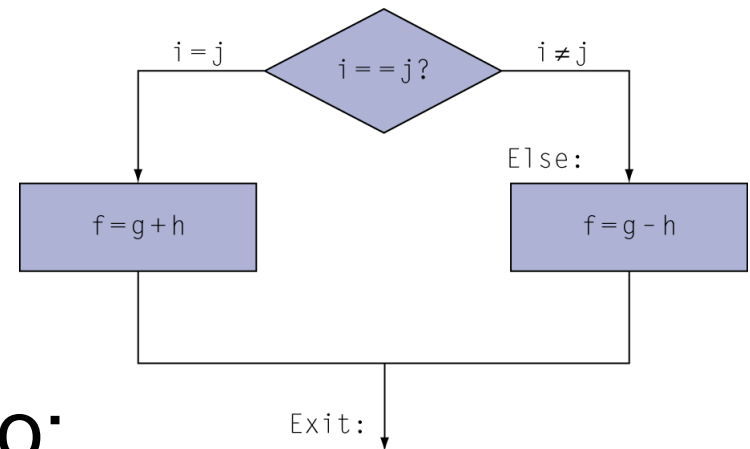
- Código C:

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

- f, g, ... in \$s0, \$s1, ...

- Código MIPS compilado:

```
        bne $s3, $s4, Else  
        add $s0, $s1, $s2  
        j   Exit  
Else:   sub $s0, $s1, $s2  
Exit:   ...
```



O assembler calcula o endereço

Compilando laços

- Código C:

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

- i em \$s3, k em \$s5, endereço de save em \$s6

- Código MIPS compilado:

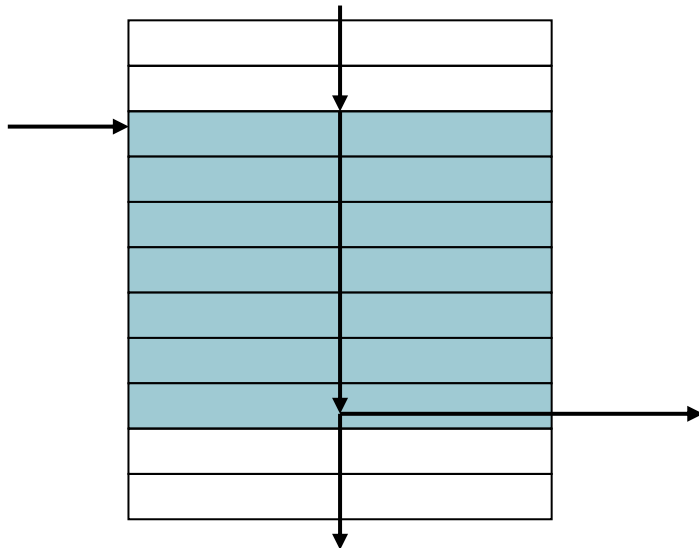
```
Loop:  sll    $t1, $s3, 2
       add   $t1, $t1, $s6
       lw    $t0, 0($t1)
       bne   $t0, $s5, Exit
       addi  $s3, $s3, 1
       j    Loop
Exit:  ...
```

Design das inst. de desvio

- Por que não b1t, bge, etc?
- O hardware para $<$, \geq , ... é mais lento que $=$, \neq
 - Para checar $=$ e \neq , basta realizar uma subtração
 - Para checar $<$, \geq , ... são necessárias mais de uma instrução
 - Em bilhões de instruções, isso faz diferença!
- beq e bne são os casos mais comuns

Blocos básicos

- Um bloco básico é uma sequência de instruções
 - Sem ramificações (exceto ao final)
 - Sem destino de desvio (exceto no início)



- O compilador identifica blocos básicos para otimização
- Um processador avançado pode acelerar a execução desses blocos

Mais instruções condicionais

- `slt/slt`: *set on less than (immediate)*
 - Define o resultado como 1 se a condição for verdadeira
 - Caso contrário, define como zero
- `slt rd, rs, rt`
 - Se $(rs < rt)$ $rd = 1$; senão $rd = 0$;
- `slti rt, rs, constant`
 - Se $(rs < constant)$ $rt = 1$; senão $rt = 0$;
- É comum usar junto com `beq`, `bne`

```
slt $t0, $s1, $s2 # if ($s1 < $s2)
bne $t0, $zero, L # branch to L
```

Com e sem sinal

- Comparação com sinal: `slt`, `slti`
- Comparação sem sinal: `sltu`, `sltui`
- Exemplo:
 - `$s0 = 1111 1111 1111 1111 1111 1111 1111 1111`
 - `$s1 = 0000 0000 0000 0000 0000 0000 0000 0001`
 - `slt $t0, $s0, $s1 # signed`
 - $-1 < +1 \Rightarrow \$t0 = 1$
 - `sltu $t0, $s0, $s1 # unsigned`
 - $+4,294,967,295 > +1 \Rightarrow \$t0 = 0$