

# INE5607 – Organização e Arquitetura de Computadores

Linguagem de Montagem e de Máquina

## Aula 10: Instruções de tomada de decisão: if-then-else

Prof. Laércio Lima Pilla

laercio.pilla@ufsc.br



# Sumário

- Desvios
- If-then-else
- Comparadores
- Switch
- Exemplo avançado
- Considerações finais

# DESVIOS

# Desvios

- Computador != calculadora
  - Máquina: dá significado a operações e **testes**
  - Capacidade de tomar decisões
- Linguagens de programação
  - **if-then-else, while, for, repeat until, switch**
- Suporte na arquitetura do conjunto de instruções
  - **Desvio condicional (*branch*)**
  - **Desvio incondicional (*jump*)**

# Desvios

- Desvios condicionais

- beq: *branch on equal*

- **beq \$s0, \$s1, Igual**

- Se  $\$s0 == \$s1$ , desvie para o endereço do label Igual

- bne: *branch on not equal*

- **bne \$s0, \$s1, Diferente**

- Se  $\$s0 != \$s1$ , desvie para o endereço do label Diferente

- Tipo I

op	rs	rt	constante/endereço
----	----	----	--------------------

# Desvios

- Desvios incondicionais

- j: *jump*

- j Fim

- Desvie para o endereço do label Fim

- Tipo J

op	endereço
6 bits	26 bits

# Desvios

- Desvios incondicionais

- jal: *jump and link*

- Salve o endereço atual em \$ra e desvie
    - Suporte para chamadas de procedimentos
    - Tipo J

- jr: *jump register*

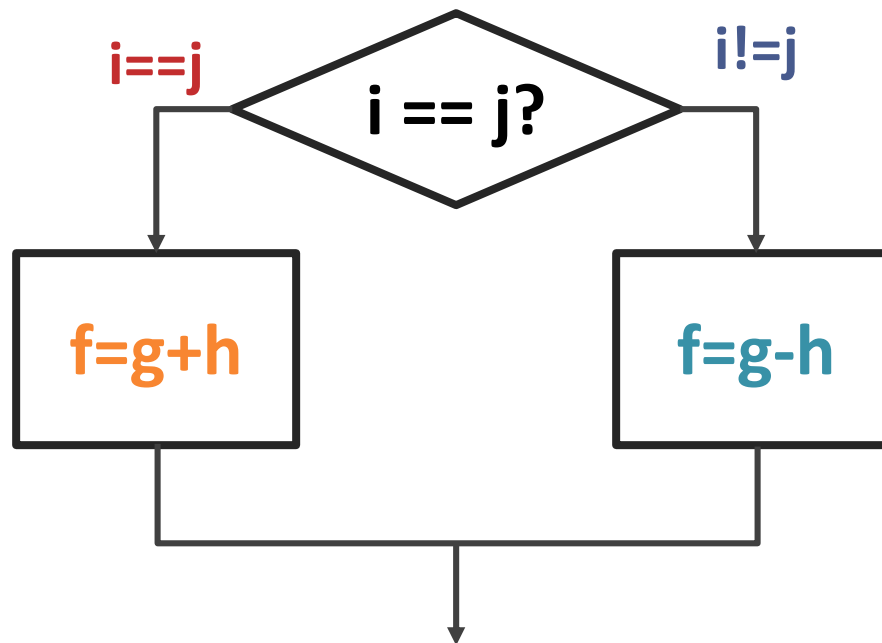
- Desvie para o endereço no registrador X
    - Suporte para o retorno de procedimentos
    - Tipo R

# IF-THEN-ELSE



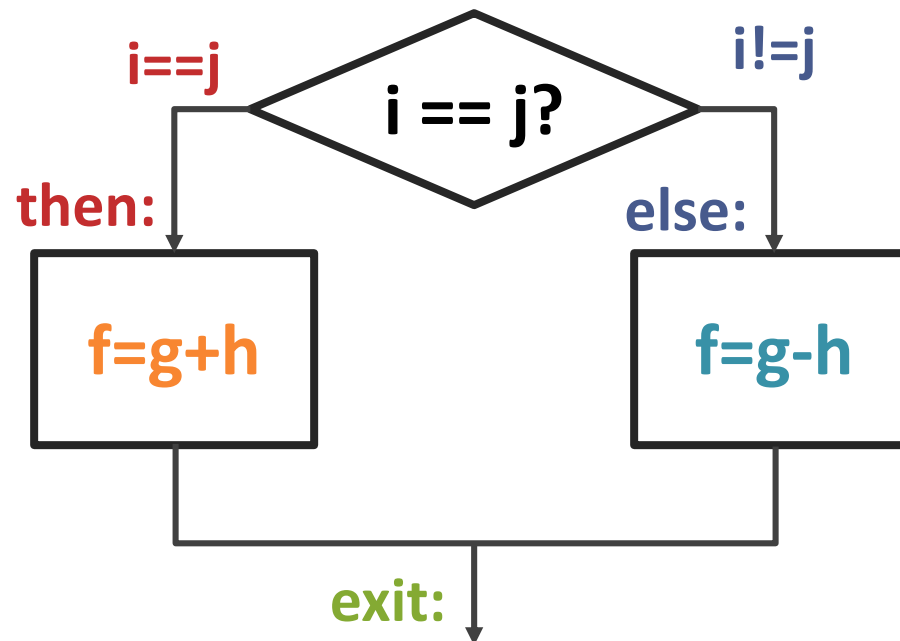
# If-then-else

- Exemplo em alto nível
  - `if (i == j) {f = g + h;} else {f = g - h;}`
  - Convenção  $(f, g, h, i, j) = (\$s0, \$s1, \$s2, \$s3, \$4)$
- Programa monolítico em fluxograma



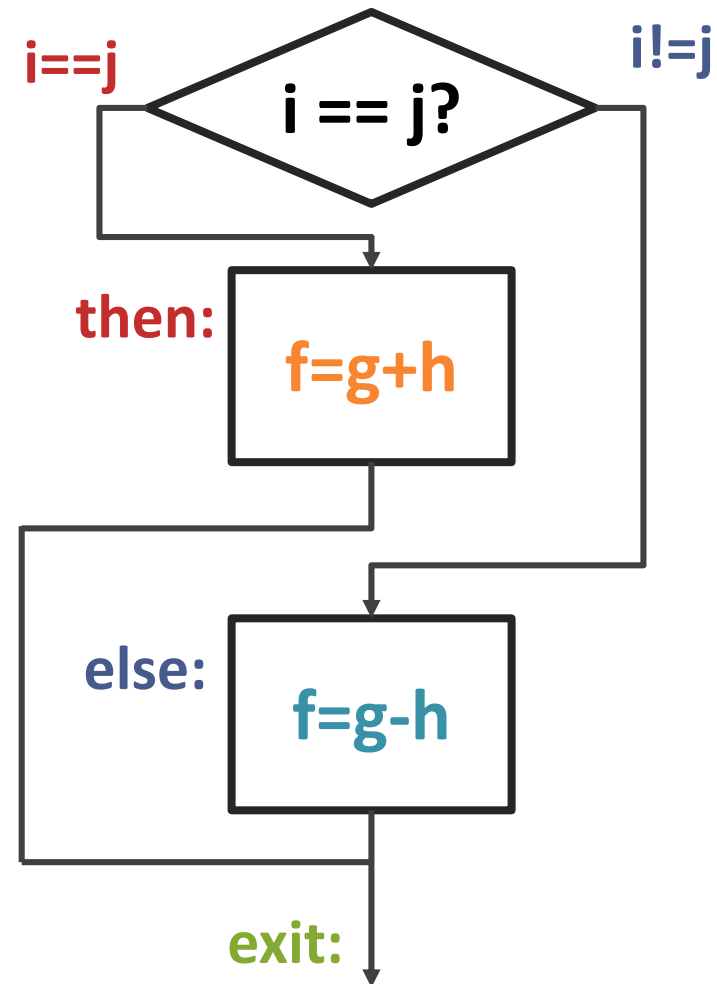
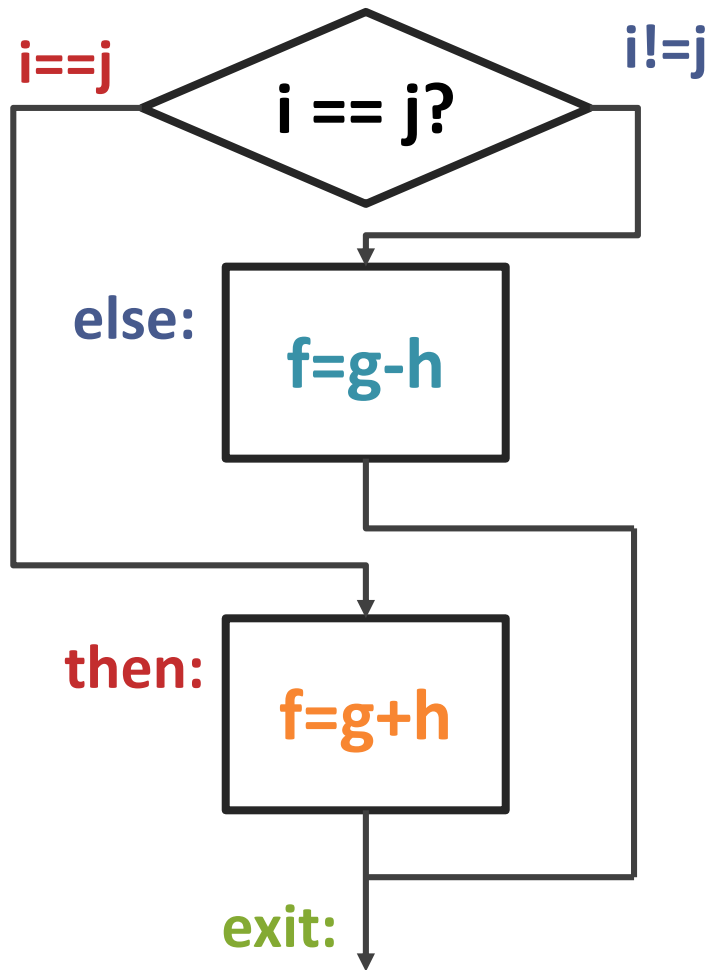
# If-then-else

- Exemplo em alto nível
  - `if (i == j) {f = g + h;} else {f = g - h;}`
  - Convenção  $(f,g,h,i,j) = (\$s0,\$s1,\$s2,\$s3,\$4)$
- Programa monolítico em fluxograma



# If-then-else

- Programa monolítico em fluxograma



# If-then-else

- Exemplo em alto nível
  - `if (i == j) {f = g + h;} else {f = g - h;}`
  - Convenção  $(f,g,h,i,j) = (\$s0,\$s1,\$s2,\$s3,\$s4)$
- Linguagem de montagem

`beq $s3, $s4, then`

`sub $s0, $s1, $s2`

`j exit`

`then: add $s0, $s1, $s2`

`exit: ...`

`bne $s3, $s4, else`

`add $s0, $s1, $s2`

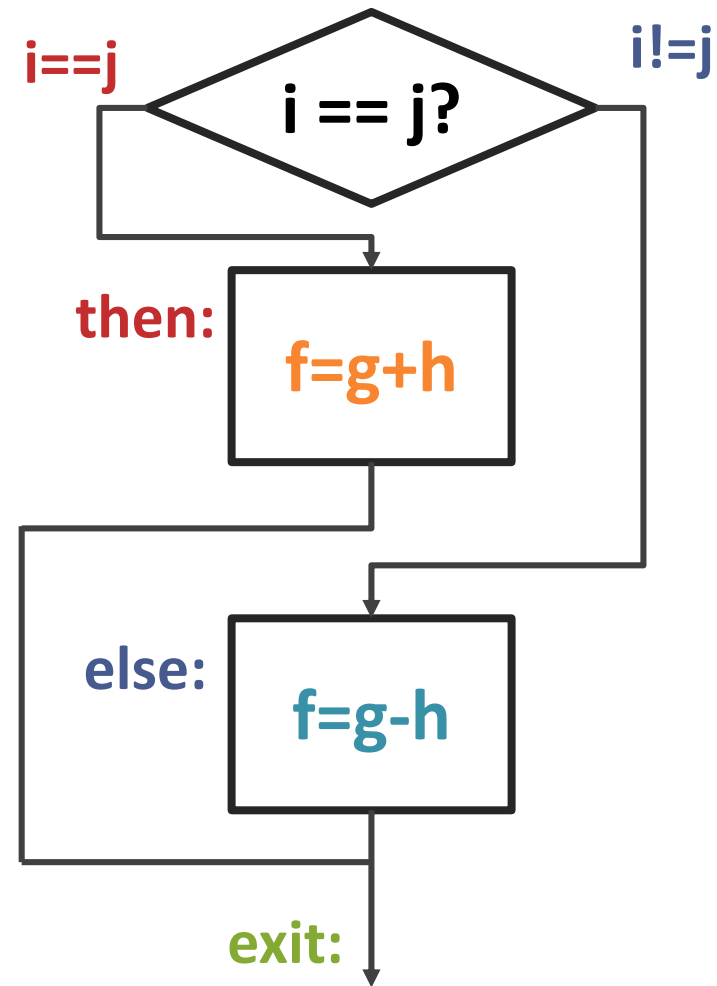
`j exit`

`else: sub $s0, $s1, $s2`

`exit: ...`

# If-then-else

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



# If-then-else

- Exemplo em alto nível

```
int quantos_iguais(int x, int y, int z){  
    int resultado = 0;  
    if (x == y) resultado++;  
    if (x == z) resultado++;  
    if (y == z) resultado++;  
    return resultado;  
}
```

Errado :)

# If-then-else

- Exemplo em alto nível

```
int quantos_iguais(int x, int y, int z){  
    int resultado = 0;  
    if (x == y) {  
        if (x == z) resultado = 3;  
        else resultado = 2;  
    } else {  
        if (x == z) resultado = 2;  
        else if (y == z) resultado = 2;  
    }  
    return resultado;  
}
```

# If-then-else

- Exemplo em alto nível
  - `#define MIN(a,b) ((a) < (b) ? (a) : (b))`
  - `c = MIN(a,b)`
    - comofas



# COMPARADORES

# Comparadores

- Testes mais populares
  - Igualdade e desigualdade
    - Embutidos em desvios condicionais
  - Teste “menor que”
    - Instrução de comparação
    - *slt: set less than*
      - Variações: *slti*, *sltu*, *sltiu*

# Comparadores

- Comparação entre duas variáveis
  - **slt** \$t0, \$s3, \$s4
    - \$t0 = 1 se \$s3 < \$s4
    - \$t0 = 0 caso contrário
- Comparação entre variável e constante
  - **slti** \$t0, \$s2, 10
    - \$t0 = 1 se \$s2 < 10
    - \$t0 = 0 caso contrário

# Comparadores

- Exemplo em alto nível
  - `#define MIN(a,b) ((a) < (b) ? (a) : (b))`
  - `c = MIN(a,b)`
  - `if (a < b) c = a; else c = b;`

# Comparadores

- Exemplo em alto nível
  - if ( $a < b$ )  $c = a$ ; else  $c = b$ ;
  - $a = \$s1$ ,  $b = \$s2$ ,  $c = \$s0$
  - **slt**  $\$t0$ ,  $\$s1$ ,  $\$s2$
  - beq  $\$t0$ ,  $\$zero$ , **else**
  - add  $\$s0$ ,  $\$zero$ ,  $\$s1$   $\#\$t0 == 1, a < b$
  - **j exit**
  - **else:** add  $\$s0$ ,  $\$zero$ ,  $\$s2$   $\#\$t0 == 0, a \geq b$
  - **exit:** ...

# Comparadores

- Para implementar relacionais MIPS só usa:
  - **beq, bne**
  - **slt, slti**
  - **registrador \$zero**
- Comparação e desvio em instrução única
  - Relacionais **=, ≠**
- Comparação e desvio em duas instruções
  - Relacionais **<, >, ≤, ≥**

# SWITCH

# Switch

- Instrução Case/Switch

```
switch (minhaPrimeiraVariavel){
```

```
    case -1:
```

```
        minhaPrimeiraResposta = 10;
```

```
        Break;
```

```
        if(mPV == -1) mPR = 10;
```

```
    case 1:
```

```
        else if(mPV == 1) mPR = 20;
```

```
        minhaPrimeiraResposta = 20; else mPR = 30;
```

```
        Break;
```

```
    default:
```

```
        minhaPrimeiraResposta = 30;
```

```
}
```

ou jump address table e jr



# EXEMPLO AVANÇADO

# Exemplo avançado

```
Class LimitedCounter{  
    private int value;  
    private int limit;  
    public LimitedCounter(int _limit){ ...  
value = 0; limit = _limit;}  
    public int getValue(){ return value;}  
    public int getLimit(){ return limit;}  
    ...  
}
```

# Exemplo avançado

```
Class LimitedCounter{  
    ...  
    public int increment(){  
        if ( value < limit ) value++;  
        return value;  
    }  
}
```

# Exemplo avançado

- Memória

Endereço rel. objeto	Significado	Valor
0x0000	Id	0x2345 6789
0x0004	Value	0x0000 000c
0x0008	Limit	0x0000 1000
0x000C	End. LimitedCounter()	0x0060 0000
0x0010	End. getValue()	0x0060 1000
0x0014	End .getLimit()	0x0060 2000
0x0018	End. increment()	0x0060 3000

- Como implementar increment()?

- Sabendo que o endereço do objeto está em \$a0

# CONSIDERAÇÕES FINAIS

# Considerações finais

- Instruções de desvio incondicional
  - j, jal, jr
- Instruções de desvio condicional
  - beq, bne
  - Suporte: slt, slti
- if-then-else
  - Visão de programa monolítico
  - Switch como encadeamento de if-then-else

# Considerações finais

- Próxima aula
  - Instruções de tomada de decisão: while, for, repeat

# INE5607 – Organização e Arquitetura de Computadores

Linguagem de Montagem e de Máquina

## Aula 10: Instruções de tomada de decisão: if-then-else

Prof. Laércio Lima Pilla

laercio.pilla@ufsc.br

