

Linguagem Assembly

Multiplicação e Divisão



MULTIPLICAÇÃO DE INTEIROS

```
mul $s0, $t0, $tl # s0 = t0 * tl
```

Multiplicar dois inteiros resulta em outro inteiro.
Logo, não há novidades em relação ao que já fizemos!

MULTIPLICAÇÃO POR POTÊNCIAS DE DOIS

- Multiplicar números por **potências de 2** é trivial para o **computador**. Basta realizar a operação de *shift left*, que significa mover os bits para a esquerda.
- Se movermos os bits de um número binário uma casa para a esquerda, multiplicaremos por 2
- Se movermos os bits de um número binário duas casas para a esquerda, multiplicaremos por 4
- ...
- Se movermos os bits de um número binário n casas para a esquerda, multiplicaremos por 2^n

MULTIPLICAÇÃO POR POTÊNCIAS DE DOIS

Ex.:

$$\begin{array}{l} 000110_2 = 6_{10} \quad \text{SHIFT LEFT 1} \\ 001100_2 = 12_{10} \quad \text{SHIFT LEFT 1} \\ 011000_2 = 24_{10} \quad \text{SHIFT LEFT 1} \\ 110000_2 = 48_{10} \quad \text{SHIFT LEFT 1} \end{array}$$

Conclusão: 6 SHIFT LEFT 3 dá 48

SHIFT LEFT LÓGICO

sll \$t0, \$t1, n # faz shift left de t1 n casas para
a esquerda

DIVISÃO DE INTEIROS

div \$t0, \$t1 #realiza a divisão inteira t0/t1

#a parte inteira vai para lo

#o resto vai para hi

mflo \$s0 #move o conteúdo de lo para s0

mfhi \$s1, #move o conteúdo de hi para s1