

# Linguagens de montagem

## Capítulo 6 – Instruções lógicas e de deslocamento

Ricardo Anido  
Instituto de Computação  
Unicamp

# Instruções lógicas

	<hr/> E-lógico <hr/>	<hr/> OU-lógico <hr/>	<hr/> OU-exclusivo <hr/>
Operando1	00001111	00001111	00001111
Operando2	00110011	00110011	00110011
Resultado	<hr/> 00000011 <hr/>	<hr/> 00111111 <hr/>	<hr/> 00111100 <hr/>

# Instruções de processamento

COM	Nome	Operação
AND	E-lógico	$Rd \leftarrow Rn \wedge \text{Operando2}$
EOR	Ou-exclusivo	$Rd \leftarrow Rn \oplus \text{Operando2}$
SUB	Subtração	$Rd \leftarrow Rn - \text{Operando2}$
RSB	Subtração reversa	$Rd \leftarrow \text{Operando2} - Rn$
ADD	Adição	$Rd \leftarrow Rn + \text{Operando2}$
ADC	Adição com vai-um	$Rd \leftarrow Rn + \text{Operando2} + C$
SBC	Subtração com empresta-um	$Rd \leftarrow Rn - \text{Operando2} + C - 1$
RSC	Subtração reversa com empresta-um	$Rd \leftarrow \text{Operando2} - Rn + C - 1$
TST	Testa bits	$Rn \wedge \text{Operando2}$
TEQ	Testa equivalência	$Rn \oplus \text{Operando2}$
CMP	Comparação	$Rn - \text{Operando2}$
CMN	Comparação negativa	$Rn + \text{Operando2}$
ORR	Ou-lógico	$Rd \leftarrow Rn \vee \text{Operando2}$
MOV	Move registrador	$Rd \leftarrow \text{Operando2}$
BIC	Desliga bit	$Rd \leftarrow Rn \wedge \text{not Operando2}$
MVN	Move registrador negado	$Rd \leftarrow \text{not Operando2}$

# Problema

Escreva um trecho de programa para trocar os bits mais e menos significativos de `r0`, sem alterar os bits restantes, como ilustrado na Figura abaixo.



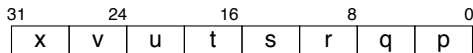
Antes



Depois

# Comparando elementos de uma estrutura

suponha que uma estrutura de oito elementos com quatro bits cada é implementada usando uma palavra de 32 bits, como mostrado na Figura abaixo:

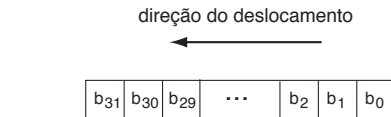


Suponha agora que desejamos testar se o elemento p da estrutura tem o mesmo valor que o elemento q.

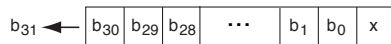
# Instruções de deslocamento

- ▶ As instruções de deslocamento do LEG operam sempre sobre um registrador, e deslocam os bits do registrador operando para a direita ou para a esquerda.
- ▶ Todos os bits são deslocados ao mesmo tempo.
- ▶ Por exemplo, no deslocamento para a esquerda, o bit  $b_0$  (menos significativo) do registrador é deslocado para a posição do bit  $b_1$ , que por sua vez é deslocado para a posição do bit  $b_2$ , e assim por diante.

# Instruções de deslocamento

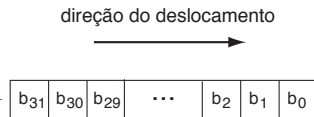


*Antes*

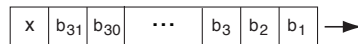


*Depois*

(a) Deslocamento para a esquerda



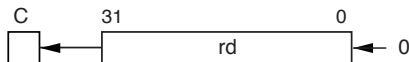
*Antes*



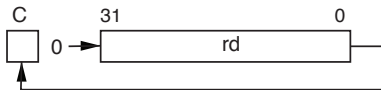
*Depois*

(b) Deslocamento para a direita

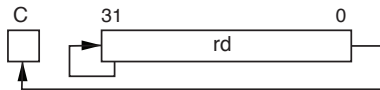
# Instruções de deslocamento



SHL  
deslocamento para a esquerda



SHR  
deslocamento para a direita



SAR  
deslocamento aritmético para a direita



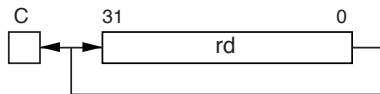
Suponha que `r12` contém o endereço inicial de uma cadeia de caracteres '0' ou '1' que representa um número em notação binária, sendo que `r12` aponta para o caractere que representa o “bit mais significativo” da cadeia. O número de caracteres da cadeia, entre 1 e 32, é dado no registrador `r1`. Escreva um trecho de programa para colocar em `r0` o valor que a cadeia dada representa. Por exemplo, se a cadeia de caracteres é '01011011' (em hexadecimal, os valores dos caracteres são 0x30, 0x31, 0x30, 0x31, 0x31, 0x30, 0x31, 0x31), o valor de `r0` ao final do trecho deve ser 0x5b.

# Instruções de rotação

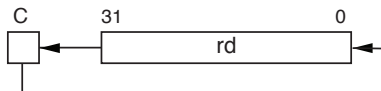
Instruções de rotação são similares a instruções de deslocamento, mas o bit ejetado do registrador é injetado novamente no operando, no lado oposto de onde o bit foi ejetado.



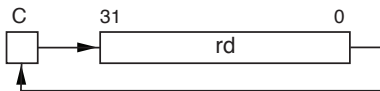
ROL  
rotação para a esquerda



ROR  
rotação para a direita



RCL  
rotação com vai-um para a esquerda



RCR  
rotação com vai-um para a direita