

MATA49

Programação de Software Básico

Operações com bits

Leandro Andrade
leandrojsa@ufba.br

Operações com bits

- Deslocamento (shift):
 - Permite o deslocamento de uma cadeia de bits para direita ou para esquerda

Original	1	1	1	0	1	0	1	0
Left shifted	1	1	0	1	0	1	0	0
Right shifted	0	1	1	1	0	1	0	1

Operações com bits

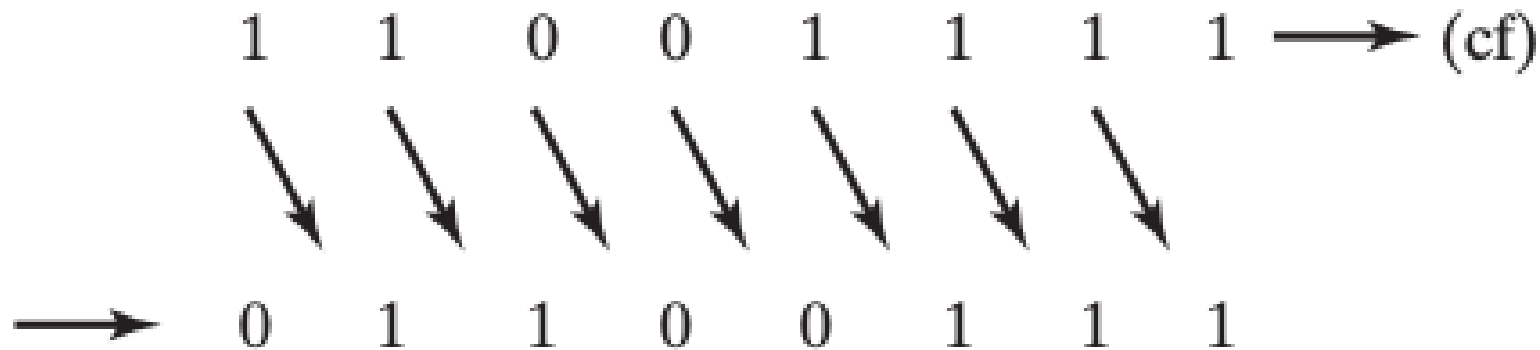
- SHL e SHR

- SHL: deslocamento para esquerda (left)
- SHR: deslocamento para direita (right)
- Sintaxe:
SHL/SHR <destino>,<num_deslocamentos>
- O último bit deslocado para fora do registrador é armazenado no Carry Flag (CF)

Operações com bits

- SHL e SHR

- Exemplo de deslocamento de 1 bit para direita:



Operações com bits

- SHL e SHR

- Exemplos:

- MOV AL, 10011100b

- SHL AL, 1

- ; AL= 00111000 CF= 1

- MOV AL, 10110011b

- SHR AL, 2

- ; AL = 00101100 CF=1

Operações com bits

- SHL e SHR
 - Deslocamentos podem ser uma opção para operações de multiplicação e divisão por potências de 2 em inteiros sem sinal
 - São operações muito mais rápidas que a MUL e DIV

Operações com bits

- Aritméticos (SAL e SAR):
 - Permite que números inteiros com sinais sejam deslocados conservando o sinal
 - SAL/SAR <destino>, <num_deslocamentos>
 - A instrução **SAL** funciona de modo idêntico a SHL, pois o deslocamento a esquerda não implica em nenhuma alteração de sinal do inteiro

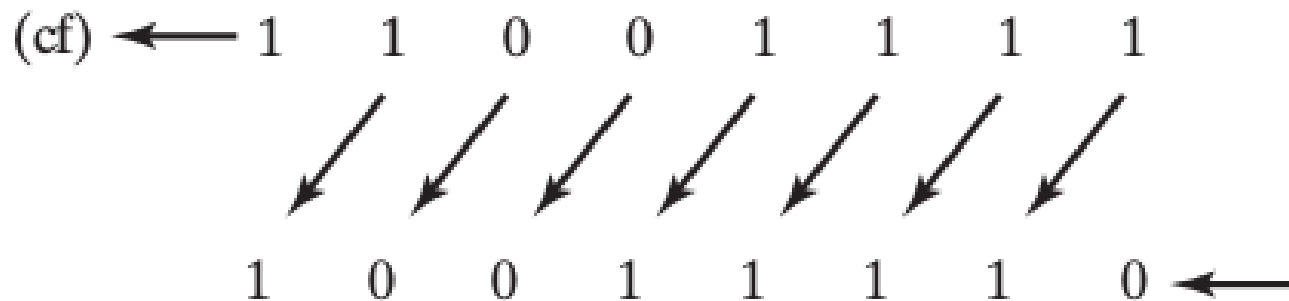
Operações com bits

- Aritméticos (SAL e SAR):

- SAL

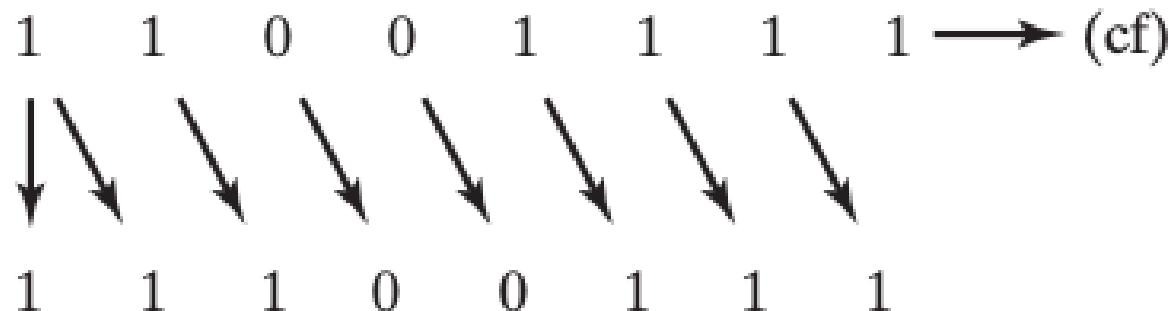
- Exemplo de deslocamento de 1 bit:

- Note que o bit acrescentado é 0 e não ocorre nenhuma alteração no sinal



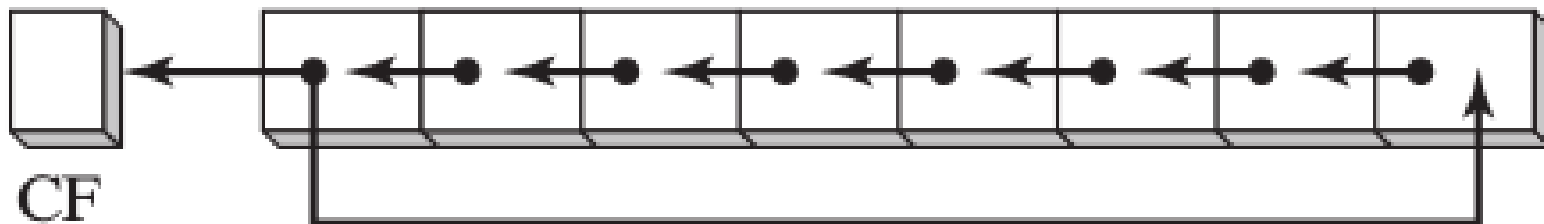
Operações com bits

- Aritméticos (SAL e SAR):
 - SAR
 - Exemplo de deslocamento de 1 bit:
 - Note que o bit acrescentado é 1, para conservar o sinal do número inteiro (negativo)



Operações com bits

- Rotação (ROL e ROR):
 - Faz um deslocamento circular, ou seja os bits deslocados irão para fim/início da cadeia
 - Sintaxe:
 - ROL/ROR <destino>, <num_deslocam>



Operações com bits

- Rotação (ROL e ROR):
 - Exemplos:

```
mov  al,40h           ; AL = 01000000b
rol  al,1             ; AL = 10000000b, CF = 0
rol  al,1             ; AL = 00000001b, CF = 1
rol  al,1             ; AL = 00000010b, CF = 0
```

```
mov  al,00000100b
ror  al,3             ; AL = 10000000b, CF = 1
```

Operações com bits

- Exemplo:
 - Conta o número de bits “on” de EAX

```
1      mov     bl, 0           ; bl will contain the count of ON bits
2      mov     ecx, 32        ; ecx is the loop counter
3  count_loop:
4      shl     eax, 1         ; shift bit into carry flag
5      jnc     skip_inc       ; if CF == 0, goto skip_inc
6      inc     bl
7  skip_inc:
8      loop    count_loop
```

Operações com bits

- Exercício:
 - Como armazenar o resultado de uma multiplicação localizada nos registradores DX:AX em um registrador de 32-bits com as instruções de descolamento de bits?

Operações com bits

- Operações Booleanas:

- AND:

- Realiza a operação booleana “e” bit-a-bit nos operandos

- Sintaxe: AND <destino>, <fonte>

destino = destino AND fonte

x	y	$x \wedge y$
0	0	0
0	1	0
1	0	0
1	1	1

Operações com bits

- Operações Booleanas:

- AND:

- Exemplos:

- ```
mov al, 11001111b
```

- ```
and    al, 00001100b
```

- ```
; al = 00001100
```

- ```
mov al,10101110b
```

- ```
and al,11110110b
```

- ```
;al = 10100110
```

Operações com bits

- Operações Booleanas:
 - AND:
 - Exemplo de aplicação conversão de caractere de minúsculo para maiúsculo (vice-versa):

0 1 **1** 0 0 0 0 1 = 61h ('a')

0 1 **0** 0 0 0 0 1 = 41h ('A')

Operações com bits

- Operações Booleanas:

- OR:

- Aplica a operação booleana “ou” bit-a-bit nos operandos

- Sintaxe:

OR <destino>, <fonte>

destino = destino OR fonte

x	y	$x \vee y$
0	0	0
0	1	1
1	0	1
1	1	1

Operações com bits

- Operações Booleanas:

- OR:

- Exemplo:

- ```
mov al, 11001111b
```

- ```
or     al, 00001100b
```

- ```
; al = 11001111
```

# Operações com bits

- Operações Booleanas:

- XOR:

- Aplica o “ou exclusivo” bit-a-bit nos operandos
    - Sintaxe:

XOR <destino>, <fonte>

destino = destino XOR fonte

| x | y | $x \oplus y$ |
|---|---|--------------|
| 0 | 0 | 0            |
| 0 | 1 | 1            |
| 1 | 0 | 1            |
| 1 | 1 | 0            |

# Operações com bits

- Operações Booleanas:

- XOR:

- Exemplo:

- ```
mov    al, 11001111b
```

- ```
xor al, 00001100b
```

- ```
; al = 11000011
```

Operações com bits

- Operações Booleanas:

- NOT

- Aplica o complemento de 1, ou seja inverte o valor de cada bit
 - Sintaxe: NOT <destino>

ATENÇÃO! A operação NOT é diferente da NEG.

- NEG: aplica complemento de 2
 - NOT: aplica complemento de 1

Operações com bits

- Operações Booleanas:
 - NOT

Exemplo:

```
mov    al, 11001111b
```

```
not    al ; al = 00110000
```

Operações com bits

- TEST
 - Aplica uma operação AND porém não armazena o resultado.
 - Modifica os bits Flags de acordo com o resultado da operação (Zero Flag)
 - Caso o resultado tenha algum bit com valor 1 o ZF=0
 - Caso o resultado tenha todos os bits com valor 0 o ZF=1
 - Sintaxe: TEST <destino>, <fonte>

Operações com bits

- TEST

- Exemplos:

```
0 0 1 0 0 1 0 1  <- input value
0 0 0 0 1 0 0 1  <- test value
0 0 0 0 0 0 0 1  <- result: ZF = 0

0 0 1 0 0 1 0 0  <- input value
0 0 0 0 1 0 0 1  <- test value
0 0 0 0 0 0 0 0  <- result: ZF = 1
```


Operações com bits

- Exercício:

O timestamp de um determinado arquivo pode ser representado em um número binário de 16 bits, onde do bit 0 ao bit 4 representa o valor em segundos, do bit 5 ao bit 10 representa os minutos e do bit 11 ao 15 representa as horas. Escreva um código em assembly que leia o **timestamp previamente armazenado no registrador BX** e extraia os **bits referentes aos minutos** e armazene na **variável ValMinutos (não inicializada)**.

ValMinutos resw 1

; registrador bx já inicializado com timestamp

Para melhor ilustração considere o exemplo abaixo:

Horas					Minutos						Segundos				
0	1	1	1	0	1	0	0	1	0	0	0	1	0	1	1
14 horas					36 minutos						11 segundos				
<u>ValMinutos</u> = 100100b = 36															