

MOV, MOVZ, MOVS, XCHG

Como os dados são representados nos registradores

• Os dados dos registradores podem ser modificados utilizando diferentes tamanhos de dados.

• Exemplo:

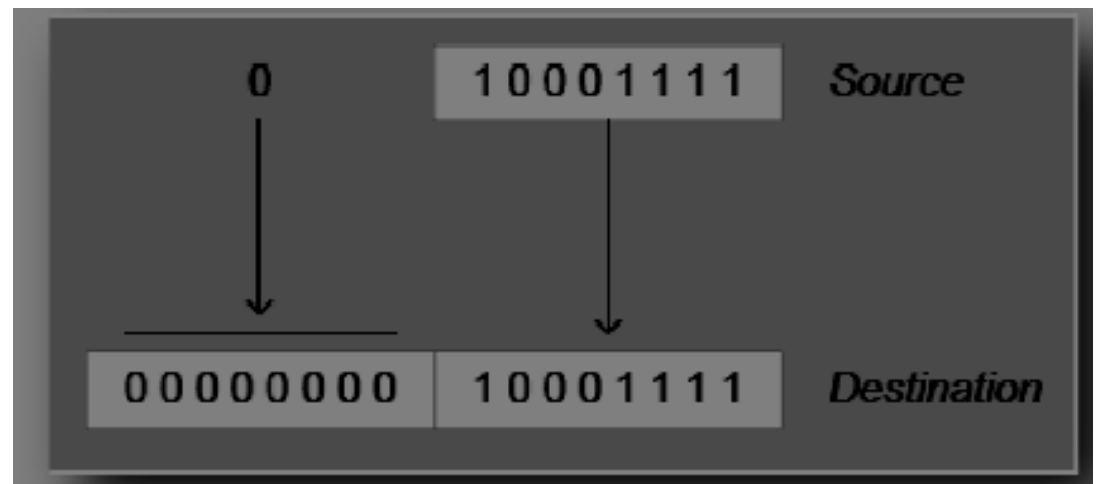
```
Section .data
OneByte DB 78h
OneWord DW 1234h
OneDword DD 12345678h
Caractere DB 'B'
```

```
Section .text
global CMAIN
CMAIN:
Mov eax, [caractere] ; EAX = 00000042h
Mov eax, 0 ; EAX = 00000000h
Mov al, [OneByte] ; EAX = 00000078h
Mov ax, [oneWord] ; EAX = 00001234h
Mov eax, [oneDWord] ; EAX = 12345678h
Mov ax, 0 ; EAX = 12340000h
ret
```

MOVZX

• Quando copia um valor menor do que o destino, o MOVZX preenche com zeros as posições do destino que não foram preenchidas

```
Mov bl, 10001111b  
Movzx ax, bl
```



MOVZX

•De registrador para registrador

```
mov bx, 0A69Bh
```

```
Movzx eax, bx          ;    EAX=0000A69Bh
```

```
Movzx edx, bl          ;    EDX=0000009Bh
```

```
Movzx cx, bl           ;    CX=009Bh
```

•De memória para registrador

```
Section .data
```

```
Byte1 DB  9Bh
```

```
Word1 DW 0A69Bh
```

```
Section .text
```

```
global CMAIN
```

```
CMAIN:
```

```
    Movzx  eax, [word1]      ; EAX = 0000A69B
```

```
    Movzx  edx, [byte1]     ; EDX=0000009Bh
```

```
    Movzx  cx, [byte1]      ; CX = 009Bh
```

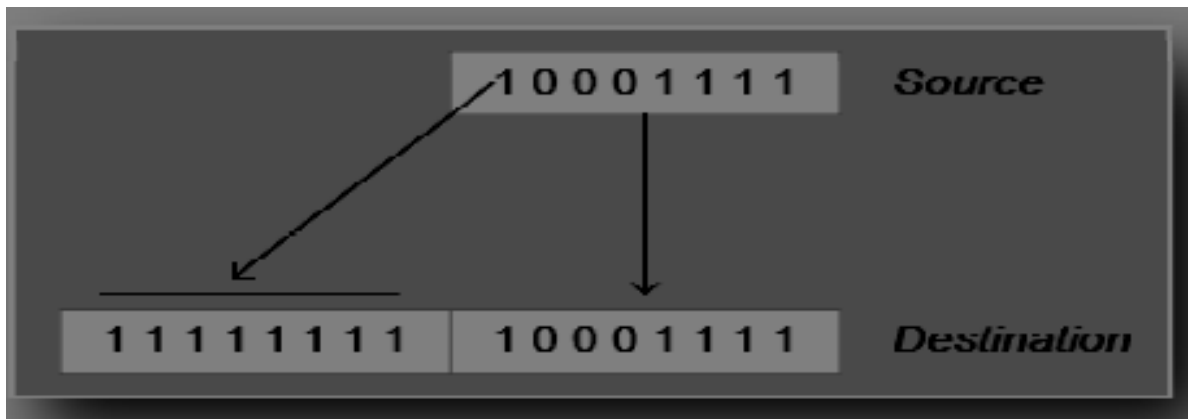
```
ret
```

MOVSX

• Quando copia um valor menor do que o destino, o MOVSX preenche com o bit mais significativo as posições do destino que não foram preenchidas

Mov bl, 10001111b

Movsx ax, bl



MOVSX

•De registrador para registrador

```
mov bx, 0A69Bh
```

```
Movsx eax, bx          ;    EAX=FFFFA69Bh
```

```
Movsx edx, bl          ;    EDX=FFFFFF9Bh
```

```
Movsx cx, bl           ;    CX=FF9Bh
```

•De memória para registrador

```
Section .data
```

```
Byte1 DB 9Bh
```

```
Word1 DW 0A69Bh
```

```
Section .text
```

```
global CMAIN
```

```
CMAIN:
```

```
    Movsx  eax, [word1]      ; EAX = FFFFA69B
```

```
    Movsx  edx, [byte1]     ; EDX= FFFFFFF9Bh
```

```
    Movsx  cx, [byte1]      ; CX = FF9Bh
```

```
ret
```

XCHG

- Swap – realiza troca entre dados
- Na troca de dados não é necessário um terceiro local de memória para a realização da troca
- Exemplo:

XCHG ax, dx ; troca o conteúdo de ax por dx e vice versa
XCHG bl, ch ; troca o conteúdo de bl por ch e vice versa

XCHG

•Tipos de troca

xchg ax, bx ; troca entre registradores de 16 bits
xchg ah, al ; troca entre registradores de 8 bits
xchg var1,bx ; entre memoria e registrador
xchg eax, ebx ; troca entre registradores de 32 bits

•Exemplo:

Section .data

val1 DW 1000h

val2 DW 2000h

Section .text

global CMAIN

CMAIN:

mov ax, [val1] ; AX = 1000h

xchg ax, [val2] ; AX= 2000h , val2 = 1000h

mov [val1], ax ; val1 = 2000h

ret

Exercicios

- Repetir os exemplos utilizando MOV.
- Indique os valores nos registradores após a execução do código abaixo:

```
mov bx, 0A69Bh
```

```
Movsx eax, bx      ;
```

```
Movsx edx, bl      ;
```

```
Movsx cx, bl       ;
```

Exercicios

- 1 - Repetir os exemplos utilizando MOV.
- 2 -Indique os valores nos registradores após a execução do código abaixo:

```
mov bx, 0A69Bh  
Movsx eax, bx      ;  
Movsx edx, bh      ;  
Movsx dl, bl       ;
```

Exercicios

3 - Indique os valores nos registradores após a execução do código abaixo:

```
mov bx, 0A69Bh  
Movzx eax, bx      ;  
Movzx edx, bh      ;  
Movzx dl, bl       ;
```

Exercicios

4 - Indique os valores nos registradores após a execução do código abaixo:

```
mov bx, 0A69Bh
```

```
Movsx eax, bx      ;
```

```
Movsx edx, bl      ;
```

```
Movsx cx, bl       ;
```

Exercicios

5 -Indique os valores nos registradores após a execução do código abaixo:

```
Section .data
```

```
    Byte1 DB 9Bh
```

```
    Word1 DW 0A69Bh
```

```
Section .text
```

```
global CMAIN
```

```
CMAIN:
```

```
    movzx  eax, word1      ;
```

```
    xchg  eax, edx
```

```
    movzx  edx, byte1      ;
```

```
    movzx  cx, byte1       ;
```

```
ret
```

Exercicios

6 -Indique os valores nos registradores após a execução do código abaixo:

```
Section .data
```

```
    Byte1 DB 9Bh
```

```
    Word1 DW 0A69Bh
```

```
Section .text
```

```
global CMAIN
```

```
CMAIN:
```

```
    movsx  eax, word1      ;
```

```
    xchg  al, dl
```

```
    movsx  edx, byte1      ;
```

```
    movsx  cx, byte1       ;
```

```
ret
```