

**Universidade de São Paulo**

Organização de Computadores

Dr. Jorge Luiz e Silva

Cap 5

# Operações Lógicas e de Deslocamento

And Or Xor Not Test

Fazer um programa que mascare 8 bits mais significativos de um dado da posição 1000h da memória e coloque o resultado na posição 1002h.

DS - 3000h

Dado - 1000h

Resultado - 1002h

# Solução - LD1

```
MOV     DX,3000H
MOV     DS,DX
MOV     BX,1000H
MOV     AX,[BX]
INC     BX
INC     BX
AND     AX,00FFH
MOV     [BX],AX
INT 20H
```

# Problema - LD2

Fazer um programa que zere o bit 4 e o bit 14 de um dado da posição 1000h da memória.

DS - 3000h

Dado - 1000h

Resultado - 1002h

# Solução - LD2

```
MOV          DX,3000H
MOV          DS,DX
MOV          BX,1000H
MOV          AX,[BX]
AND          AX,1011 1111 1110 1111B
INC          BX
INC          BX
MOV          [BX],AX
INT 20H
```

# Problema - LD3

Fazer um programa que coloque 1 nos 8 bits mais significativos de um dado da posição 1000h da memória

DS                   -3000h

Dado                 -1000h

Resultado           -1002h

# Solução - LD3

```
MOV          DX,3000H
MOV          DS,DX
MOV          BX,1000H
MOV          AX,[BX]
INC          BX
INC          BX
OR           AX,1111 1111 0000 0000B
MOV          [BX],AX
INT 20H
```

# Problema - LD4

Fazer um programa que teste se os 4 bits menos significativos de um dado da posição 1000h são zeros, na forma:

CL ← 00            se iguais

CL ← FF            se diferentes

DS                - 3000h

Dado              - 1000h



# Solução - LD4

```
MOV          DX,3000H
MOV          DS,DX
MOV          CL,0
MOV          BX,1000H
MOV          AX,[BX]
TEST         AX,000FH
JZ           FIM
MOV          CL,0FFH
FIM:         INT 20H
```

# Problema - LD5

Fazer um programa que: supondo uma esteira com barras de ferro de tamanhos diversos, conte quantas barras possuem um tamanho maior que um valor pré-determinado, na forma: dois sensores foto-elétricos estarão apagados ao mesmo tempo se a barra for maior que a medida padrão.

Segmento de dados 3000

bits 0 e 7 do dado da posição 1000 são originados a partir dos sensores.

Contagem na posição 1001

# Solução - LD5

```
MOV     DX,3000H
MOV     DS,DX
MOV     BX,1000
T2:     MOV     AL,[BX]
        TEST    AL,1000001B
        JNZ    T1
        INC    DX
T1:     LOOP   T2
        INT   20H
```

# Programa - LD6

Fazer um programa que coloque em seqüência, a partir da posição 1002h os bits de um dado da posição 1000h

DS	- 3000h
Dado	- 1000h
Bits	- 1002h

# Solução - LD6

```
MOV      DX,3000H
MOV      DS,DX
MOV      BX,1000H
MOV      CX,16
MOV      DX,1
MOV      AX,[BX]
INC      BX
T3      INC      BX
TEST     AX,DX
JNZ      T1
```

# Solução - LD6 cont.

```
MOV        [BX],CH
JMP        T2
T1: MOV     CH,1
MOV        [BX],CH
XOR        CH,CH
T2: ADD     DX,DX
LOOP       T3
INT 20H
```

# Problema - LD7

Fazer um programa que conte quantos bits em um determinado dado memória são iguais a 1.

DS - 3000h

Dado - 1000h

Resultado - 1002h

# Solução - LD7

```
MOV     DX,3000H
MOV     DS,DX
MOV     BX,1000H
MOV     CX,16
MOV     DX,1
MOV     AX,[BX]
XOR     BX,BX
TEST    AX,DX
JZ      T1
INC     BX
T1      ADD     DX,DX
        LOOP   T2
        MOV    [1002H],BX
        INT   20H
```



# Problema - LD8

Fazer um programa que gere o bit de paridade no bit mais significativo de um dado de 7 bits  
(positivo – paridade par)

DS - 3000h

Dado - 1000h

Resultado - 1001h

# Solução - LD8

```
MOV      DX,3000H
MOV      DS,DX
MOV      BX,1000H
MOV      CX,7
MOV      DL,1
MOV      AL,[BX]
T2: INC   BX
MOV      [BX],CH
TEST     AL,DL
JZ       T1
MOV      CH,1
MOV      [BX],CH
MOV      CH,0
```

# Solução - LD8 cont.

```
T1:  ADD     DL,DL
      LOOP  T2
      MOV   CX,6
      MOV   BX,1001H
      MOV   AL,[BX]
T3:  INC    BX
      XOR   AL,[BX]
      LOOP  T3
      MOV   AH,[1000H]
      TEST  AL,1
      JZ    T4
      ADD   AH,80H
T4:  MOV   [1000H],AH
      INT  20H
```

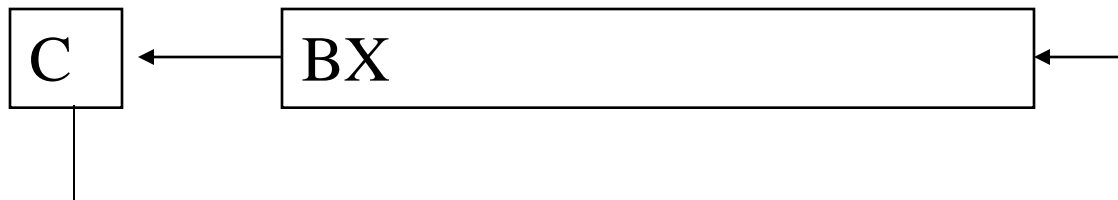
# Operações de Rotação e Deslocamento

RCL ROL SAL SAR

RCR ROR SHL SHR

Ex: RCL [BX],CL

RCL [BX],1



# Programa - RD1

Fazer um programa que determine quantos bits são iguais a 1 em um dado de 16 bits posicionado em 1000h da memória.

DS	- 3000h
Dado	- 1000h
Bits	- 1002h

# Solução - RD1

```
MOV     DX,3000H
MOV     DS,DX
MOV     BX,1000H
MOV     CX,16
MOV     DX,0
MOV     AX,[BX]
T2:    ROL     AX,1
        JNC     T1
        INC     DX
T1:    LOOP   T2
        MOV     [1002H],DX
        INT 20H
```

# Problema - RD2

Fazer um programa que coloque os 8 bits menos significativos de um dado da memória, nos 8 bits mais significativos inserindo zeros.

DS	- 3000H
Dado	- 1000h
Bits	- 1002h

# Solução - RD2

```
MOV     DX,3000H
MOV     DS,DX
MOV     BX,1000H
MOV     AX,[BX]
MOV     CL,8
SAL     AX,CL
INT 20H
```



# Problema - RD3

Fazer um programa que separe um dado de 16 bits da memória em duas palavras de 8 bits.

DS	- 3000h
Dado	- 1000h
8 bits low	- 1002h
8 bits high	- 1003h

# Solução - RD3

```
MOV     DX,3000H
MOV     DS,DX
MOV     BX,1000H
MOV     AX,[BX]
AND     AL,0FF00H
MOV     CL,8
SHR     AX,CL
MOV     DL,AL
MOV     AX,[BX]
AND     AX,0FFH
INC     BX
INC     BX
MOV     [BX],DL
INC     BX
MOV     [BX],AL
INT 20H
```