MC404

ORGANIZAÇÃO BÁSICA DE COMPUTADORES E LINGUAGEM DE MONTAGEM

2006

Prof. Paulo Cesar Centoducatte

ducatte@ic.unicamp.br

www.ic.unicamp.br/~ducatte

MC404

ORGANIZAÇÃO BÁSICA DE COMPUTADORES E LINGUAGEM DE MONTAGEM

"Introdução à linguagem assembly do 8086 (continuação)"

Introdução à linguagem assembly do 8086 (cont.)
Sumário

- Instruções de Controle de Fluxo
- Desvios Condicionais e Incondicionais
  - Jxxx
  - Jump
  - loop
- Comparação
- Saltos sinalizados e não-sinalizados
- Algumas Estruturas de Linguagens de Alto Nível
  - If-Then-Else
  - For
  - While
  - Repeat

Instruções de controle de fluxo

Instruções de laço (*loop*) e de salto (*jump*) permitem que:

- o programa "tome" certas decisões, alterando seu curso;
- certas partes de um programa sejam repetidas um número controlado de vezes.

Exemplo: Exibição na tela de todos os caracteres ASCII

```
TITLE   EXIBICAO DE CARACTERES ASCII
.MODEL  SMALL
.STACK   100H
.CODE
;inicializacao de alguns registradores
;
        MOV AH,2            ;funcao DOS para exibicao de caracter
        MOV CX,256          ;contador com o numero total de caracteres
        MOV DL,00H          ;DL inicializado com o primeiro ASCII
;
;definicao de um processo repetitivo de 256 vezes
;
PRINT_LOOP:
        INT 21H             ;exibir caracter na tela
        INC DL              ;incrementa o caracter ASCII
        DEC CX              ;decrementa o contador
        JNZ PRINT_LOOP      ;continua exibindo enquanto CX nao for 0
;
;quando CX = 0, o programa quebra a sequencia do loop    ;saida para o DOS
;
        MOV AH,4CH
        INT 21H
        END
```

Instrução de comparação

CMP destino,fonte

CMP (Compare) compara os conteúdos destino e fonte, que podem ser:

- **registrador e registrador**
- **registrador e uma posição de memória**
- **um número diretamente como operando fonte**

Combinações legais de operandos

	Operando destino	
Operando fonte	Registrador de dados	Posição de memória
Reg. de dados	sim	sim
Posição de memória	sim	**não**
Constante	sim	sim

Instrução de comparação

CMP calcula a subtração: (destino) - (fonte)

O resultado não é armazenado, porém Todos os Flags de Estado são afetados.

Exemplos de instruções válidas:

```
CMP DX,BX          ;compara os conteúdos de DX e BX
CMP AX,[WORD1]     ;compara o conteúdo do registrador AX com o da
                   ;posição de memória WORD1
CMP AH,'A'         ;compara o conteúdo de AH com o caracter ASCII
    'A'
                   ;hexa 41H
```

Saltos Condicionais

Jxxx rótulo_de_destino

xxx define uma condição dependente de algum dos Flags de Estado

Se a condição xxx é verdadeira:

- a próxima instrução a ser executada é aquela definida pelo rótulo_de_destino;

- a CPU ajusta o registrador IP para apontar para a posição de memória dada por rótulo_de_destino.

Se a condição xxx é falsa:

- a próxima instrução é aquela que imediatamente segue o salto.

Saltos Condicionais

Faixa de endereçamento do rótulo_de_destino:

- **deve preceder Jxxx não mais do que 126 bytes;**
- **deve suceder Jxxx não mais do que 127 bytes.**

Há três classes de saltos condicionais:

- **saltos sinalizados: dependem de um resultado na forma de um número sinalizado;**
- **saltos não-sinalizados: dependem de um resultado na forma de um número não-sinalizado;**
- **saltos de Flag simples: que dependem do *status* de algum dos Flags.**

Obs.: A instrução Jxxx não altera nenhum Flag.

Saltos Condicionais

Tipos de saltos condicionais

Saltos sinalizados		
Símbolo	**Descrição**	**Condições**
JG ou JNLE	salto se maior do que OU salto se não menor do que ou igual a	ZF = 0 E SF = OF
JGE ou JNL	salto se maior do que ou igual a OU salto se não menor do que	SF = OF
JL ou JNGE	salto se menor do que OU salto se não maior do que ou igual a	SF ≠ OF
JLE ou JNG	salto se menor do que ou igual a OU salto se não maior do que	ZF = 1 OU SF ≠ OF

Saltos Condicionais

Tipos de saltos condicionais

Saltos não-sinalizados				
Símbolo		**Descrição**		**Condições**
JÁ JNBE	ou	salto se acima de salto se não abaixo de ou igual a	OU	CF = 0 ZF = 0	E
JAE JNB	ou	salto se acima de ou igual a salto se não abaixo de	OU	CF = 0	
JB JNAE	ou	salto se abaixo de salto se não acima de ou igual a	OU	CF = 1	
JBE JNA	ou	salto se abaixo de ou igual a salto se não acima de	OU	CF = 1 ZF = 1	OU

Saltos Condicionais

Tipos de saltos condicionais

Saltos de Flag simples		
Símbolo	**Descrição**	**Condições**
JE ou JZ	salto se igual OU salto se igual a zero	ZF = 1
JNE ou JNZ	salto se não igual OU salto se não igual a zero	ZF = 0
JC	salto se há VAI-UM (*carry*)	CF = 1
JNC	salto se não há VAI-UM (*not carry*)	CF = 0
JO	salto se há *overflow*	OF = 1
JNO	salto se não há *overflow*	OF = 0
JS	salto se o sinal é negativo	SF = 1
JNS	salto se o sinal é não-negativo (+)	SF = 0
JP ou JPE	salto se a paridade é PAR (*even*)	PF = 1
JNP ou JPO	salto se a paridade é IMPAR (*odd*)	PF = 0

Saltos Condicionais

Diferença entre Saltos sinalizados e não-sinalizados

a) Trecho de programa que supõe quantidades não-sinalizadas:

```
;supondo que AX contem 7FFFh e BX contem 8000h
        ...
        CMP  AX,BX
        JA   PT2        ;o salto não ocorre porque 7FFFh < 8000h
        ...
        ...
PT2:    MOV ...                  ;continuação do programa
```

Saltos Condicionais

Diferença entre Saltos sinalizados e não-sinalizados

b) Trecho de programa que supõe quantidades sinalizadas:

```
;supondo que AX contem 7FFFh e BX contem 8000h
        ...
        CMP  AX,BX
        JG   PT2        ;o salto ocorre porque 7FFFh (+) > 8000h (-)
        ...
        ...
PT2:    MOV ...                  ;continuação do programa
```

Salto Condicionais

Exemplo: Supondo que AX e BX contenham números sinalizados, escreva um trecho de programa que coloque o maior deles em CX.

```
        ...
        MOV  CX,AX      ;AX já é pressuposto ser o maior deles
        CMP  AX,BX
        JNL  ABAIXO     ;poderia ser também JGE  ABAIXO
        MOV  CX,BX      ;caso BX seja de fato o maior deles
ABAIXO:     ...         ;continuação do programa
        ...
```

Salto Incondicional

JMP rótulo_de_destino

Rótulo_de_destino é uma posição no programa, no mesmo segmento de código onde JMP aparece

Não há restrição de faixa de endereçamento como em JXXX

JMP pode ajudar a solucionar o problema de faixa de endereçamento das instruções JXXX

Como?

Salto Incondicional

Exemplo: trecho utilizando JMP para resolver a restrição no alcance de JXXX

```
TOPO:       ...
            ; mais do que 126 bytes de instruções: limitação
            ; para JXXX
            ...
            ; corpo de algum laço
            ;
            DEC CX  ; se Zero termina o laço
            JZ  CONTINUA
            JMP TOPO
CONTINUA:   MOV ...       ; o programa continua
            ...
```

Exercícios Sugeridos

- 1) Escreva um programa que apresente uma '?', leia em seguida duas letras maiúsculas e exiba-as na próxima linha, em ordem alfabética.

- 2) Modifique o programa de exibição de caracteres ASCII (visto em aula), de forma a exibir 16 caracteres por linha separados por espaços em branco.

- 3) Escreva um programa que pergunte ao usuário para teclar um dígito hexadecimal, exiba na próxima linha o seu valor decimal e pergunte ao usuário se deseja continuar a utilizar o programa: se for digitado S (sim), o programa se repete desde o começo; se for digitado outro caracter, o programa termina. Teste se o dígito hexa está na faixa de valores correta. Se não estiver, exiba uma mensagem para o usuário tentar de novo.

Exercícios Sugeridos

- **4) Crie um trecho de código modificando o programa do exercício (3) acima, tal que se o usuário falhar em entrar com um dígito hexa na faixa correta mais do que três tentativas, o programa exibe uma mensagem adequada e termina.**

- **5) Crie um programa que implemente uma multiplicação por meio de somas sucessivas. Faça as considerações que achar necessárias.**

- **6) Crie um programa que implemente uma divisão por meio de subtrações sucessivas, exibindo o quociente e o resto com mensagens adequadas. Faça as considerações que achar necessárias.**

Algumas Estruturas de Linguagens de Alto Nível

1) Estrutura IF - THEN – ELSE

Em linguagem de alto nível:

```
IF      (condição)
        THEN  (seqüência 1)
        ELSE  (seqüência 2)
END_IF
```

Exemplo: Suponha que AL e BL contenham dois caracteres ASCII; exiba aquele que seja o primeiro em ordem alfabética.

Algumas Estruturas de Linguagens de Alto Nível

Em linguagem de alto nível:

```
IF      AL (menor ou igual a)  BL
        THEN    (exibir AL)
        ELSE    (exibir BL)
END_IF
```

Em linguagem de montagem:

```
                        ....
; if AL menor ou igual a BL
                        MOV AH,2h
                        CMP AL,BL
                        JA  TROCA
; then
                        MOV DL,AL
                        INT 21h
                        JMP FIM
; else
  TROCA:                MOV DL, BL
                        INT 21h
; end_if
  FIM:                  .....
```

Algumas Estruturas de Linguagens de Alto Nível

Repetição

LOOP rótulo_de_destino

Tem como contador implícito o registrador CX, que deve ser inicializado antes do laço.

Salta para rótulo_de_destino enquanto o conteúdo de CX não for zero.

Quando CX = 0, a próxima instrução após LOOP será executada.

CX é decrementado automaticamente quando LOOP é executada.

Nenhum FLAG é afetado.

Algumas Estruturas de Linguagens de Alto Nível

Exemplo de instruções válidas:

```
LOOP   PT1
LOOP   TOPO
LOOP   RETORNO
```

Obs: são equivalentes as seqüências:

```
        MOV CX, (valor_inicial)
TOPO:   ...
        ...
        ...
        LOOP TOPO
```

```
MOV CX, (valor_inicial)
TOPO:   ...

        DEC CX

        JNZ     TOPO
```

Algumas Estruturas de Linguagens de Alto Nível

2) FOR loop

Em linguagem de alto nível:

```
FOR     (número_de_vezes)      DO
        (seqüência de instruções)
END_FOR
```

Exemplo: Exiba uma seqüência de 80 asteriscos no monitor de vídeo.

Algumas Estruturas de Linguagens de Alto Nível

Em linguagem de alto nível:

```
FOR    (80 vezes)          DO
       (exibir " * " )
END_FOR
```

Em linguagem de montagem:

```
          ...
;for 80 vezes
                    MOV CX,80d
                    MOV AH,2h
                    MOV DL," * "
;do
          TOPO:     INT 21h
                    LOOP  TOPO
;end_for
                                ...
```

Exercício: modifique o programa que exibe todos os caracteres ASCII (visto em aula), para utilizar a instrução LOOP.

Algumas Estruturas de Linguagens de Alto Nível

3) WHILE loop

Em linguagem de alto nível:

```
WHILE(condição_verdadeira) DO
        (seqüência de instruções)
END_WHILE
```

Exemplo: Ler caracteres ASCII do teclado, contando sua quantidade, até que seja lido o caracter *Carriage Return* (CR).

Algumas Estruturas de Linguagens de Alto Nível

Em linguagem de alto nível:

```
WHILE          (caracter diferente de CR?)    DO
        (ler caracter do teclado e armazená-lo)
        (contador = contador +1)
END_WHILE
```

Em linguagem de montagem:

```
                    ...
                    MOV DX,0h                   ; inicialização
                    MOV AH,1h
                    INT 21h
;while
    LOOP:           CMP AL,0Dh                  ; é o caracter CR?
                    JE          FIM             ; salto quando caracter é igual a CR
                    MOV AL, (algum lugar)       ; salvando o caracter lido
                    INC DX                      ; conta número de caracteres
                    INT 21h
                    JMP LOOP                    ; fecha o loop WHILE
;end_while
    FIM:
```

Algumas Estruturas de Linguagens de Alto Nível

4) REPEAT loop

Em linguagem de alto nível:

```
REPEAT
        (seqüência de instruções)
UNTIL (condição_verdadeira)
```

Exemplo: Ler caracteres ASCII do teclado, contando sua quantidade, até que seja lido o caracter *Carriage Return* (CR).

Algumas Estruturas de Linguagens de Alto Nível

Em linguagem de alto nível:

```
    REPEAT
          (ler caracter do teclado e armazená-lo)
          (contador = contador + 1)
    UNTIL  (caracter igual a CR)
```

Em linguagem de montagem:

```
                    ...
                    MOV DX,0h                       ;inicialização
                    MOV AH,1h
;repeat
    LOOP:           INT 21h
                    MOV AL, (algum lugar)   ;salvando o caracter lido
                    INC DX                  ;conta número de caracteres
                    CMP AL,0Dh              ;é o caracter CR?
                    JNE      LOOP           ;salto enquanto caracter não é
    CR
;until
                    ...
```