

ANEXO

Instruções de Movimentação de dados:

MOV, XCHG, LEA, LDS, LES, LSS, PUSH, POP, PUSHF, POPF

Formato geral:

- MOV DEST, ORIG DEST:=ORIG
- XCHG OPER1, OPER2 OPER1:=OPER2; OPER2:=OPER1
- LEA DEST, ORIG

Carrega o registo especificado por DEST com o endereço efectivo da localização da memória indicada por ORIG.

- PUSH OPER SP:=SP-2
[SS:SP]:=OPER

Transfere a word indicada por OPER para o topo da pilha ([SS:SP]).

- POP OPER OPER:=[SS:SP]
SP:=SP+2

Transfere a word presente no topo da pilha para o operando indicado por OPER.

Formato geral (x ∈ {D, E, S}):

- LxS DEST, ORIG DEST:=[ORIG]
xS:=[ORIG+2]

As instruções MOV, XCHG, LEA, LDS, LES, LSS, PUSH e POP não afectam as flags com a excepção óbvia de PUSHF e POPF.

Instruções de Conversão de dados:

CBW, CWD

Formato geral:

Acção:

- CBW Faz a extensão com sinal do valor de 8 bits presente em AL para AX.
- CWD Faz a extensão com sinal do valor de 16 bits presente em AX para DX:AX.

As instruções CBW e CWD não afectam as flags.

Instruções Aritméticas:

ADD, ADC, INC, SUB, SBB, DEC, CMP, NEG, MUL, IMUL, DIV, IDIV

Formato geral:

- ADD DEST, ORIG DEST:= DEST + ORIG

Flags alteradas: SF, ZF, PF, AF, CF, OF.

- ADC DEST, ORIG DEST:= DEST + ORIG + Flag Carry

Flags alteradas: SF, ZF, PF, AF, CF, OF.

- INC OPER OPER:= OPER + 1

Flags alteradas: SF, ZF, PF, AF, OF.

- SUB DEST, ORIG DEST:= DEST – ORIG

Flags alteradas: SF, ZF, PF, AF, CF, OF.

- SBB DEST, ORIG DEST:= DEST – ORIG - Flag Carry

Flags alteradas: SF, ZF, PF, AF, CF, OF.

- DEC OPER OPER:= OPER – 1

Flags alteradas: SF, ZF, PF, AF, OF.

- CMP DEST, ORIG

Idêntica à instrução SUB mas com uma diferença crucial – não armazena o resultado da subtracção em DEST.

Flags alteradas: SF, ZF, PF, AF, CF, OF.

- NEG OPER OPER := 0 – OPER

Flags alteradas: CF, SF, ZF, PF, AF, OF.

- (I)MUL OPER Se OPER for de 8 bits AX:= AL *OPER
Se OPER for de 16 bits DX:AX:=AX*OPER

Flags alteradas: As flags CF, OF, SF, ZF, PF e AF ficam com valores indefinidos.

- (I)DIV OPER

Se OPER for de 8 bits divide AX por OPER, ficando o quociente em AL e o resto em AH.

Se OPER for de 16 bits divide DX:AX por OPER, ficando o quociente em AX e o resto em DX.

Flags alteradas: CF, SF, ZF, PF, AF, OF.

Instruções Lógicas:

AND, OR, XOR, NOT

Formato geral:

Acção:

- AND OPER1, OPER2 DEST:= OPER1 AND OPER2
- OR OPER1, OPER2 DEST:= OPER1 OR OPER2
- XOR OPER1, OPER2 DEST:= OPER1 XOR OPER2
- NOT OPER OPER:= NOT OPER

À excepção da instrução NOT que não afecta as flags, todas as outras alteram as seguintes flags: CF, SF, ZF, PF, OF, ficando AF com um valor indefinido.

Instruções de Deslocamento e Rotação de bits:

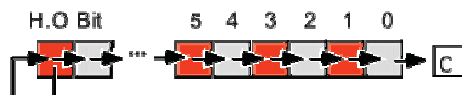
SHL, SAL, SHR, SAR, RCL, ROL, RCR, ROR

Formato geral:

- SHL/SAL DEST, COUNT

Ação:**Formato geral:**

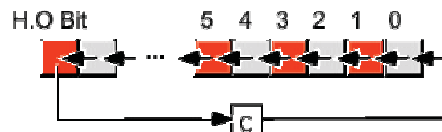
- SAR DEST, COUNT

Ação:**Formato geral:**

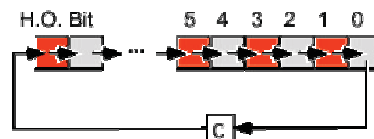
- SHR DEST, COUNT

**Formato geral:**

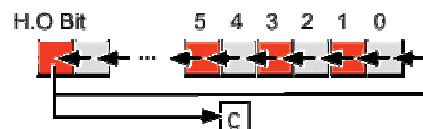
- RCL DEST, COUNT

Ação:**Formato geral:**

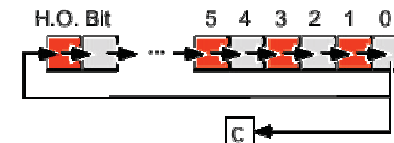
- RCR DEST, COUNT

Ação:**Formato geral:**

- ROL DEST, COUNT

Ação:**Formato geral:**

- ROR DEST, COUNT

Ação:**Flags alteradas:**

As instruções SHL, SAL, SHR, SAR alteram as flags CF, SF, ZF, PF, OF, deixando AF com um valor indefinido.

As instruções RCL, ROL, RCR, ROR alteram as flags CF e OF.

Instruções de manipulação de flags:

TEST, CLC, STC, CMC, CLD, STD, CLI, STI

Formato geral:

- TEST DEST, COUNT

Ação:

Realiza o AND lógico, bit a bit, entre os seus operandos mas não armazena o resultado da operação.

Flags alteradas: CF, SF, ZF, PF, OF, deixando AF com um valor indefinido.

Formato geral:

CLC

STC

CMC

CLD

STD

CLI

STI

Ação:

CLC: Coloca a "0" a CF.

STC: Coloca a "1" a CF.

CMC: Complementa a CF.

CLD: Coloca a "0" a DF.

STD: Coloca a "1" a DF.

CLI: Coloca a "0" a IF.

STI: Coloca a "1" a IF.

Estas instruções modificam as flags correspondentes.

Instruções de manipulação de “Strings”:

MOVS, CMPS, LODS, STOS, SCAS

Formato geral:

- movs {b, w}:
- cmps {b, w}:
- lods {b, w}:
- stos {b, w}:
- scas {b, w}:

Acção:

es:[di] := ds:[si]
 cmp ds:[si], es:[di]
 ax/al := ds:[si]
 es:[di] := ax/al
 cmp ax/al, es:[di]

Todas as instruções mencionadas: MOVS, CMPS, LODS, STOS, SCAS, têm ainda o seguinte comportamento adicional:

```
if DF == 0 then
    di := di + size    ; size= 1 (b - byte), 2 (w - word)
else
    di := di - size
endif
```

Flags alteradas:

As instruções MOVS, LODS, STOS e suas derivadas não alteram as flags. As instruções CMPS, SCAS e suas derivadas alteram as flags AF, CF, OF, PF, SF E ZF.

Instruções de repetição:

REP, REPE, REPZ, REPNE, REPNZ

Formato:

```
REP      MOVS {b,w}
REPE     CMPS {b,w}
REPZ     CMPS {b,w}
REPNE    CMPS {b,w}
REPNZ    CMPS {b,w}
REPE     SCAS {b,w}
REPZ     SCAS {b,w}
REPNE    SCAS {b,w}
REPNZ    SCAS {b,w}
REP      STOS {b,w}
```

Acção:

REP (repete uma operação CX vezes, i. e., enquanto CX ≠ 0).
 REPZ (repete enquanto condição for zero, i. e., CX ≠ 0 e ZF=1).
 REPE (repete enquanto condição representar igualdade, i. e., CX ≠ 0 e ZF=1).
 REPNZ (repete enquanto condição for diferente de zero, i. e., CX ≠ 0 e ZF=0).
 REPNE (repete enquanto condição representar diferença, i. e., CX ≠ 0 e ZF=0.))

Instruções de Controlo de Fluxo do programa:**Formato geral:**

Instrução Label

Salto incondicionais:

JMP – transfere incondicionalmente o controlo do fluxo do programa para *label*.

CALL – transfere incondicionalmente o controlo de fluxo do programa para a primeira instrução do procedimento especificado por label.

RET – permite o retorno de procedimentos ao programa chamante.

Salto condicionais para testar as flags:

JC – Jump if CF=1

JNC – Jump if CF=0.

JS – Jump MP if SF=1

JNS – Jump if SF=0.

JO – Jump if OF=1

JNO – Jump if OF=0.

JP – Jump if PF=1

JNP – Jump if PF=0

JPE – Jump if PF=1 ou (paridade par)

JPO – Jump if PF=0 ou (paridade

ímpar).

Salto condicionais para comparações sem sinal:

JA – Jump if above (>) / JNBE – Jump if not below or equal (not <=).

JAЕ – Jump if above or equal (>=) / JNB – Jump if not below (not <)

JB – Jump if below (<) / JNAE – Jump if not above or equal (not >=)

JBE – Jump if below or equal (<=) / JNA – Jump if not above (not >)

Salto condicionais para comparações com sinal:

JG – Jump if greater than (>) / JNLE – Jump if not less than or equal (not <=).

JGE – Jump if greater than or equal (>=) / JNL – Jump if not less than (not <)

JL – Jump if less than (<) / JNGE – Jump if not greater than or equal (not >=)

JLE – Jump if less or equal (<=) / JNG – Jump if not greater than (not >)

Salto condicionais para comparações com e sem sinal:

JE – Jump if equal.

JNE – Jump if not equal.

NOTA:

AF – Auxiliary Flag.

CF – Carry Flag.

DF – Direction Flag.

IF – Interrupt Flag.

OF – Overflow Flag.

PF – Parity Flag.

SF – Sign Flag.

ZF – Zero Flag.