

ORGANIZAÇÃO BÁSICA DE COMPUTADORES E LINGUAGEM DE MONTAGEM

Introdução à linguagem assembly do 8086 - Sintaxe – continuação

O registrador de sinalizadores (FLAGS)

1. Flags de Status e Flags de Controle

indica o estado do microprocessador após a execução de cada instrução;

conjunto de bits individuais, cada qual indicando alguma propriedade; subdividem-se em: **Flags de Estado** (*status*) e **Flags de Controle**.

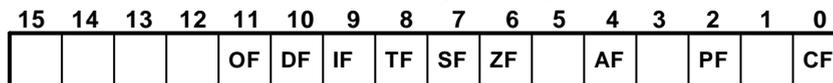
Organização

1 registrador de 16 bits

6 FLAGS de estado

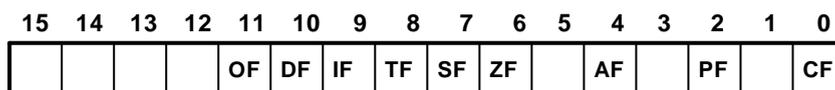
3 FLAGS de controle

7 bits não utilizados (sem função)



ORGANIZAÇÃO BÁSICA DE COMPUTADORES E LINGUAGEM DE MONTAGEM

Introdução à linguagem assembly do 8086 - Sintaxe – continuação



Flags de estado

Nome	Símbolo	Função/característica
Carry Flag	CF	Indicador de "vai-um"
Parity Flag	PF	Indicador de número PAR de 1's no byte inferior
Auxiliary Carry	AF	Indicador de "vai-um" para operações em BCD
Zero Flag	ZF	Indicador de "zero" na última operação
Sign Flag	SF	Indicador de resultado negativo
Overflow Flag	OF	Indicador de erro de transbordamento

Obs: o emprego dos **Flags de Controle** será discutido juntamente com operações com *arrays* e interrupções.

ORGANIZAÇÃO BÁSICA DE COMPUTADORES E LINGUAGEM DE MONTAGEM

Introdução à linguagem assembly do 8086 - Sintaxe – continuação

2 Overflow (erro de transbordamento)

Overflow -> ocorre porque a representação dos números está limitada a uma certa faixa

Tipos	8 bits	16 bits
Não-sinalizado	0 a 255	0 a 65.535
Sinalizado (C2)	-128 a +127	- 32.768 a + 32.767

- Qualquer operação aritmética que tenha como resultado um número fora da faixa de representação, estará produzindo **Overflow**.
- O resultado armazenado no registrador destino estará truncado e terá, portanto, um valor incorreto.

ORGANIZAÇÃO BÁSICA DE COMPUTADORES E LINGUAGEM DE MONTAGEM

Introdução à linguagem assembly do 8086 - Sintaxe – continuação

Tem-se dois Flags que podem indicar *overflow*: CF e OF

CF -> indica se há um vai-um para fora do Bit Mais Significativo do número
MSB (*most significant bit*)

OF -> testa o vem-um que chega e o vai-um gerado no MSB:

se iguais (0 e 0 ou 1 e 1) -> OF = 0

se diferentes, OF = 1

ORGANIZAÇÃO BÁSICA DE COMPUTADORES E LINGUAGEM DE MONTAGEM

Introdução à linguagem assembly do 8086 - Sintaxe – continuação

Exemplos de operações com 8 bits:

ADD AL,BL ;AL contem FFh e BL contem 01h

	repres. não-sinalizada	repres. sinalizada
FFh 1111 1111b	255	-1
01h + 0000 0001b	+ 1	+1
1 0000 0000b ->	256 (fora da faixa)	0 (OK)

Logo após a execução da instrução:

CF = 1 , indicado em **negrito**;

OF = 0 , pois no MSB o "vem-um" é igual ao "vai-um" (ambos 1).

ORGANIZAÇÃO BÁSICA DE COMPUTADORES E LINGUAGEM DE MONTAGEM

Introdução à linguagem assembly do 8086 - Sintaxe – continuação

ADD AL,BL ;ambos AL e BL contém 7Fh

	repres. não-sinalizada	repres. sinalizada
7Fh 0111 1111b	127	+ 127
7Fh + 0111 1111b	+ 127	+ 127
0 1111 1110b ->	254 (OK)	254 (fora)

Logo após a execução da instrução:

CF = 0 , indicado em **negrito**;

OF = 1 , pois no MSB o "vem-um" é diferente do "vai-um".

Portanto:

representação não-sinalizada -> **Flag CF indica overflow;**

representação sinalizada -> **Flag OF indica overflow.**

ORGANIZAÇÃO BÁSICA DE COMPUTADORES E LINGUAGEM DE MONTAGEM

Introdução à linguagem assembly do 8086 - Sintaxe – continuação

3 Como as instruções afetam os Flags

Algumas instruções, imediatamente após a sua execução:

- afetam todos os Flags ;
- afetam apenas alguns;
- não afetam nenhum.

Instrução	Flags afetados
MOV	nenhum
XCHG	nenhum
LEA	nenhum
ADD/SUB	todos
INC/DEC	todos, exceto CF que não é afetado
NEG	todos, CF=1 se o resultado não for zero

ORGANIZAÇÃO BÁSICA DE COMPUTADORES E LINGUAGEM DE MONTAGEM

Introdução à linguagem assembly do 8086 - Sintaxe – continuação

Exemplos:

```

ADD AX,BX           ;onde ambos AX e BX valem FFFFh
FFFFh              1111 1111 1111 1111 b
FFFFh              1111 1111 1111 1111 b
+-----+
FFFEh              1 1111 1111 1111 1110 b
Como resultado:    CF = 1          AF = 1          ZF = 0
                   PF = 0          SF = 1          OF = 0
    
```

```

INC AL              ;onde AL contem FFh
FFh                1111 1111 b
01h                +-----+ 1 b
100h               1 0000 0000 b
Como resultado:    CF = não afetado  AF = 1          ZF = 1
                   PF = 1          SF = 0          OF = 0
    
```

ORGANIZAÇÃO BÁSICA DE COMPUTADORES E LINGUAGEM DE MONTAGEM

Introdução à linguagem assembly do 8086 - Sintaxe – continuação

4 O programa DEBUG

O programa DEBUG do DOS provê um meio de depuração de programas em Linguagem Montadora e permite acompanhar a modificação do conteúdo de registradores (inclusive o de Flags).

Escrevendo um programa de teste e verificação dos Flags:

```
TITLE PROGRAMA PARA VERIFICACAO DOS FLAGS
;
;usado no DEBUG para verificar o registradores de Flags
;
.MODEL SMALL
.STACK 100H
.CODE
    MOV AX,4000H    ;AX = 4000h - valor inicial de AX
    ADD AX,AX      ;AX = 8000h (4000h + 4000h = 8000h)
    SUB AX,0FFFFH ;AX = 8001h (8000h - FFFFh = 8001h)
    NEG AX         ;AX = 7FFFh (C2 de 8001h)
    INC AX         ;AX = 8000h (7FFFh + 0001h = 8000h)
    MOV AH,4CH
    INT 21H       ;saida para o DOS
    END
```

ORGANIZAÇÃO BÁSICA DE COMPUTADORES E LINGUAGEM DE MONTAGEM

Introdução à linguagem assembly do 8086 - Sintaxe – continuação

Acesso ao DEBUG:

C:\ DEBUG <nome_do_programa>.EXE

alguns comandos de linha do DEBUG

- r -> "registers", para exibir o conteúdo dos registradores
- t -> "trace", para executar linha por linha
- g -> "go", para ir até o fim
- q -> "quit", para sair do DEBUG

ORGANIZAÇÃO BÁSICA DE COMPUTADORES E LINGUAGEM DE MONTAGEM

Introdução à linguagem assembly do 8086 - Sintaxe – continuação

Tela do DOS rodando o DEBUG:

```

C:\ Command Prompt - tlink -v eco.obj
Warning messages: None
Passes: 1
Remaining memory: 450k

C:\Tasm>tlink -v eco.obj
Turbo Link Version 7.1.30.1. Copyright (c) 1987, 1996 Borland International

C:\TASM>debug eco.exe
-r
AX=0000 BX=0000 CX=03A1 DX=0000 SP=0100 BP=0000 SI=0000 DI=0000
DS=0B83 ES=0B83 SS=0B95 CS=0B93 IP=0000 NU UP EI PL NZ NA PO NC
0B93:0000 B402      MOV     AH,02
-t
AX=0200 BX=0000 CX=03A1 DX=0000 SP=0100 BP=0000 SI=0000 DI=0000
DS=0B83 ES=0B83 SS=0B95 CS=0B93 IP=0002 NU UP EI PL NZ NA PO NC
0B93:0002 B23F      MOV     DL,3F
-g
?k
k
Program terminated normally
-g
C:\TASM>
    
```

ORGANIZAÇÃO BÁSICA DE COMPUTADORES E LINGUAGEM DE MONTAGEM

Introdução à linguagem assembly do 8086 - Sintaxe – continuação

Simbologia usada para os Flags no Programa Debug

	Símbolo quando 1	Símbolo quando 0
Flag de Estado		
CF	CY (carry)	NC (no carry)
PF	PE (parity even - PAR)	PO (parity odd - IMPAR)
AF	AC (auxiliary carry)	NA (no aux. carry)
ZF	ZR (zero)	NZ (no zero)
SF	NG (negativo)	PL (plus - positivo)
OF	OV (overflow)	NV (no overflow)
Flag de Controle		
DF	DN (down - para baixo)	UP (up - para cima)
IF	EI (permite interrupção)	DI (desabilita interrup.)