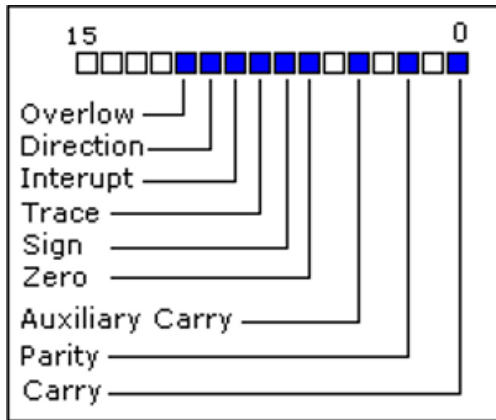


Садржај FLAG регистра

Аритметичке и логичке инструкције асемблера утичу на FLAG регистар. Са слике се види да овај регистар има 16 битова, сваки се назива флег и може имати вредност 1 или 0.

CF се сетује када постоји недозвољени пренос без знака: $255+1=256$ (мак 255)

ZF се сетује када је резултат било какве операције 0

SF има вредност највишег значајног бита

OF је 1 када постоји недозвољени пренос са знаком: $100+50=150$ (-128..127)

PF се сетује када је паран број 1 у 8 нижих битова резултата

AF се сетује када је пренос без знака за нижи нибл (нижа 4 бита)

Када је IF сетован, процесор реагује на прекиде од спољних уређаја

Када је DF 0 рад са битовима се креће у лево а када је 1 у десно, бит по бит.

Аритметичке инструкције

ADD додаје вредност другог операнда првом операнду. Утиче на флегове CF ZF SF OF PF AF

SUB одузима други операнд од првог. Утиче на флегове CF ZF SF OF PF AF

MUL множење без знака. Када је операнд бајт: $AX = AL * operand$. Када је операнд реч: $(DX AX) = AX * operand$. Ова инструкција утиче само на CF OF флегове. Када је резултат изван величине операнда ови флегови се сетују а када је резултат унутар величине операнда флегови се ресетују.

DIV дељење без знака. Када је операнд бајт: $AL = AX / operand$, $AH = ostatak$. Када је операнд реч: $AX = (DX AX) / operand$, $DX = ostatak$. Флегови су недефинисани.

INC инкрементира вредност операнда. Утиче на флегове ZF SF OF PF AF.

DEC декрементира вредност операнда. Утиче на флегове ZF SF OF PF AF.

Логичке инструкције

AND упоређује два бита логичком операцијом I. $1 AND 1 = 1$, $1 AND 0 = 0$, $0 AND 1 = 0$, $0 AND 0 = 0$. CF ZF SF OF PF AF

OR упоређује два бита логичком операцијом II. $1 OR 1 = 1$, $1 OR 0 = 1$, $0 OR 1 = 1$, $0 OR 0 = 0$. CF ZF SF OF PF AF

XOR упоређује два бита логичком операцијом ексклузивно III. $1 XOR 1 = 0$, $1 XOR 0 = 1$, $0 XOR 1 = 1$, $0 XOR 0 = 0$. CF ZF SF OF PF AF

NOT обрће вредност сваког бита у операнду. Не утиче на флегове.

NEG прави операнд негативним (у комплементу двојке). Заправо обрће сваки бит и додаје 1. CF ZF SF OF PF AF

Пример01 : Исписивање садржаја регистра ах током извршења петље током које се садржај ах инкрементира.

```
#start=led.display.exe#
```

Ово је bin фајл. Приступа се виртуелним портovima који се емулирају у фајлу

```
#make_bin#
```

c:\emu8086.io. Ово је начин за додавање екстерних виртуелних уређаја за

```
name "aritm1"
```

емулатор. Програм користи фајл led.display.exe за приказ садржаја жељеног

```
mov ax, 1234
```

регистра. Програм се налази у: c:\emu8086\devices\led_display.exe.

```
out 199, ax
```

Инструкција out приказује садржај AL или AX регистра на порт. OUT oper1, oper2

```
mov ax, -5678
```

Први операнд је број порта.

```
out 199, ax
```

```
mov ax, 0
```

```
x1:
```

```
out 199, ax
```

```
inc ax
```

; бесконачна петља исписује вредности на порт 199

```
jmp x1
```

```
hlt
```

Пример02 : Исписивање садржаја регистра ах током извршења петље током које се садржај ах инкрементира.

```

name "add-sub"
org 100h
mov al, 5          ; al = 00000101b
mov bl, 10         ; bl = 00001010b
add bl, al         ; bl = 00001111b
sub bl, 1          ; bl = 00001110b
mov cx, 8          ; postavljanje brojaca
print: mov ah, 2   ; print funkcija
      mov dl, '0'
      test bl, 10000000b; testiranje prvog bita
      jz zero
      mov dl, '1'
zero: int 21h
      shl bl, 1
loop print
mov dl, 'b'
int 21h           ; ispisuje b na kraju
mov ah, 0
int 16h
ret

```

Инструкција **TEST** ради исто што и AND функција али нигде не уписује резултат. Утиче само на флегове ZF, SF, PF.

Инструкција **SHL** помера битове првог операнда улево за онолико места колико је дефинисано у другом операнду.