

MODUL II

THE PROCESSOR STATUS AND FLAGS REGISTER

Flags register merupakan register yang digunakan untuk menunjukkan kondisi dari suatu keadaan (ya atau tidak). Registr ini juga merupakan register 16 bit terdiri dari *Status flags* ditempatkan pada bit 0, 2, 4, 6, 7 dan 11 dan *control flags* ditempatkan pada bit 8, 9, dan 10. Bit-bit lainnya tidak penting.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				OF	DF	IF	TF	SF	ZF		AF		PF		CF

- **Carry Flag.** CF = 1 jika ada *carry out* dari *most significant bit* (MSB) pada operasi penjumlahan, atau ada *borrow in* pada MSB dari suatu operasi pengurangan; dan jika tidak maka CF = 0. CF juga dipengaruhi oleh operasi pergeseran (*shift*) dan rotasi (*rotate*).
- **Parity Flag.** PF = 1 jika ada *low byte* dari hasil operasi menunjukkan logika high berjumlah genap (*even parity*). Sebaliknya PF = 0 jika berjumlah ganjil (*odd parity*). Misalnya jika hasil dari suatu penjumlahan sebuah word adalah FFFEh maka PF = 0 karena jumlah logika high pada *low byte* adalah 7.
- **Auxiliary Carry Flag.** AF = 1 jika ada *carry out* dari bit 3 pada penjumlahan, atau *borrow in* pada bit 3 pada pengurangan. AF digunakan pada operasi bilangan BCD (*Binary Coded Decimal*).
- **Zero Flag.** ZF = 1 jika hasil operasi adalah nol, dan ZF = 0 jika tidak sama dengan nol.
- **Sign Flag.** SF = 1 jika bit MSB adalah 1; yang menunjukkan hasil negatif jika anda menggunakan *sigend interpretation* SF = 0 jika bit MSB adalah 0.
- **Overflow Flag.** OF = 1 jika terjadi *signed overflow*. Dan sebaliknya adalah OF = 0. Arti *overflow* akan dijelaskan berikut:

Jika kita melakukan operasi aritmetika seperti penjumlahan, maka akan ada empat kemungkinan hasilnya: (1) *tidak terjadi overflow*, (2) *signed overflow saja*, (3) *unsigned overflow saja* dan (4) *signed overflow dan unsigned overflow*.

Instruksi	Flag yang berpengaruh
MOV / XCHG	Tidak Ada

ADD , ADC/ SUB, SBB	SF, PF, ZF, CF, OF
INC / DEC	Seluruh Flag, Kecuali CF
NEG	Seluruh Flag (CF=1 kecuali apabila hasil = 0, OF = 1 apabila word operand = 8000H atau Byte Operand=80H

ADD

Untuk menambah dalam bahasa assembler digunakan perintah **ADD** dan **ADC** serta **INC**.

Perintah ADD digunakan dengan syntax :

ADD Tujuan,Asal

Perintah ADD ini akan menambahkan nilai pada Tujuan dan Asal. Hasil yang didapat akan ditaruh pada Tujuan, dalam bahasa pascal sama dengan instruksi **Tujuan:=Tujuan + Asal**. Sebagai contohnya :

MOV AH,15h ; AH:=15h

MOV AL,4 ; AL:=4

ADD AH,AL ; AH:=AH+AL, jadi AH=19h

Perlu anda perhatikan bahwa pada perintah ADD ini antara Tujuan dan Asal harus mempunyai daya tampung yang sama, misalnya register AH(8 bit) dan AL(8 bit), AX(16 bit) dan BX(16 bit). Mungkin ada yang bertanya-tanya, apa yang akan terjadi bila Tujuan tempat hasil penjumlahan disimpan tidak mencukupi seperti pertambahan 1234h dengan F221h.

1234 h Biner --> 0001 0010 0011 0100

F221 h Biner --> 1111 0010 0010 0001

----- + ----- +

10455 h 1 0000 0100 0101 0101

Pada pertambahan diatas dapat dilihat bahwa pertambahan bilangan 1234 dengan F221 akan menghasilkan nilai 10455. Supaya lebih jelas dapat anda lihat pada pertambahan binernya dihasilkan bit ke 17, padahal register terdiri atas 16 bit saja. Operasi pertambahan yang demikian akan menjadikan carry flag menjadi satu, Contoh :

MOV AX,1234h ; Nilai AX:=1234h dan carry=0

MOV BX,0F221h ; Nilai BX:=F221h dan carry=0

ADD AX,BX ; Nilai AX menjadi 0455h dan carry=1

ADC

Perintah ADC digunakan dengan cara yang sama pada perintah ADD, yaitu :

ADC Tujuan,Asal

Perbedaannya pada perintah ADC ini Tujuan tempat menampung hasil pertambahan Tujuan dan Asal ditambah lagi dengan carry flag (Tujuan:=Tujuan+Asal+Carry). Pertambahan yang demikian bisa memecahkan masalah seperti yang pernah kita kemukakan, seperti pertambahan pada bilangan 12345678h+9ABCDEF0h.

Seperti yang telah kita ketahui bahwa satu register hanya mampu menampung 16 bit, maka untuk pertambahan seperti yang diatas bisa anda gunakan perintah ADC untuk memecahkannya,

Contoh:

```
MOV AX,1234h ; AX = 1234h CF = 0
MOV BX,9ABCh ; BX = 9ABCh CF = 0
MOV CX,5678h ; CX = 5678h CF = 0
MOV DX,0DEF0h ; DX = DEF0h CF = 0
ADD CX,DX ; CX = 3568h CF = 1
ADC AX,BX ; AX = AX+BX+CF = ACF1
```

Hasil penjumlahan akan ditampung pada register AX:CX yaitu ACF13568h. Adapun flag-flag yang terpengaruh oleh perintah ADD dan ADC ini adalah CF,PF,AF,ZF,SF dan OF.

INC

Perintah INC(Increment) digunakan khusus untuk pertambahan dengan 1. Perintah INC hanya menggunakan 1 byte memory, sedangkan perintah ADD dan ADC menggunakan 3 byte. Oleh sebab itu bila anda ingin melakukan operasi pertambahan dengan 1 gunakanlah perintah INC.

Syntax pemakaiannya adalah :

INC Tujuan

Nilai pada tujuan akan ditambah dengan 1, seperti perintah Tujuan:=Tujuan+1 dalam Turbo Pascal. Tujuan disini dapat berupa suatu register maupun memory. Contoh : perintah INC AL akan menambah nilai di register AL dengan 1. Adapun flag yang terpengaruh oleh perintah ini adalah OF,SF,ZF,AF dan PF.

SUB

Untuk Operasi pengurangan dapat digunakan perintah SUB dengan syntax:

`SUB Tujuan, Asal`

Perintah SUB akan mengurangi nilai pada Tujuan dengan Asal. Hasil yang didapat akan ditaruh pada Tujuan, dalam bahasa pascal sama dengan instruksi

Tujuan:=Tujuan-Asal.

Contoh :

`MOV AX, 15 ; AX:=15`

`MOV BX, 12 ; BX:=12`

`SUB AX, BX ; AX:=15-12=3`

`SUB AX, AX ; AX=0`

Untuk menolakkan suatu register bisa anda kurangkan dengan dirinya sendiri seperti `SUB AX,AX`.

SBB

Seperti pada operasi penambahan, maka pada operasi pengurangan dengan bilangan yang besar (lebih dari 16 bit), bisa anda gunakan perintah SUB disertai dengan SBB(Substract With Carry).

Perintah SBB digunakan dengan syntax:

`SBB Tujuan, Asal`

Perintah SBB akan mengurangi nilai Tujuan dengan Asal dengan cara yang sama seperti perintah SUB, kemudian hasil yang didapat dikurangi lagi dengan Carry Flag(`Tujuan:=Tujuan-Asal-CF`).

DEC

Perintah DEC(Decrement) digunakan khusus untuk pengurangan dengan 1. Perintah DEC hanya menggunakan 1 byte memory, sedangkan perintah SUB dan SBB menggunakan 3 byte. Oleh sebab itu bila anda ingin melakukan operasi pengurangan dengan 1 gunakanlah perintah DEC.

Syntax pemakaian perintah dec ini adalah:

`DEC Tujuan`

Nilai pada tujuan akan dikurangi 1, seperti perintah `Tujuan:=Tujuan-1` dalam Turbo Pascal. Tujuan disini dapat berupa suatu register maupun memory. Contoh : perintah `DEC AL` akan mengurangi nilai di register AL dengan 1.

Latihan

1. Jika AX berisi FFFFh dan BX berisi FFFFh

ADD AX, BX

		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FFFFh		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
FFFFh	+	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
<hr/>		<hr/>															
1 FFFEh		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0

Hasil yang tersimpan dalam register AX adalah FFFEh = 1111 1111 1111 1110

SF = 1 karena MSB adalah 1

PF = 0 karena ada tujuh (ganjil) bit yang berlogika 1 pada low byte dari hasil

ZF = 0 karena hasilnya tidak nol

CF = 1 karena ada carry out dari MSB

AF = 1 karena ada carry dari bit 3

OF = 0 karena kedua bit tanda (bit 15) sama dengan bit hasil operasi.

2. Jika AL berisi 80h dan BL berisi 80h

ADD AL,BL

		8	7	6	5	4	3	2	1	0
80h										
80h	+	1	0	0	0	0	0	0	0	0
<hr/>		<hr/>								
1 00h		1	0	0	0	0	0	0	0	0

Hasil yang tersimpan dalam register AL adalah 00h = 0000 0000

SF = 0 karena MSB adalah 0

PF = 1 karena semua bit adalah nol

ZF = 1 karena hasilnya nol

CF = 1 karena ada carry out dari MSB

AF = 0 karena tidak ada carry dari bit 3

OF = 1 karena kedua bit tanda (bit 15) tidak sama bit hasil operasi.

Tugas Pendahuluan

1. ORG 100h

```
Proses :  
MOV AH,15h ;  
MOV AL,4 ;  
ADD AH,AL ;  
MOV AX,1234h ;  
MOV BX,0F221h ;  
ADD AX,BX ;  
MOV AX,1234h ;  
MOV BX,9ABCh ;  
MOV CX,5678h ;  
MOV DX,0DEF0h ;  
XCHG AX,CX ;  
ADD CX,DX ;  
ADC AX,BX ;  
INC AL ;  
INT 20h  
RET  
END
```

Jabarkan kejadian yang terjadi pada masing-masing baris instruksi, amati dan jelaskan perubahan kondisi flag register dari tiap baris instruksi?

2. Jelaskan mengenai unsigned Overflow dan Signed Overflow ?

3. Susun algoritma Untuk Tugas Praktikum no.3 ?

Tugas Praktikum

1. Berikan nilai destinasi yang baru dan setting CF, SF,ZF,PF dan OF pada setiap instruksi berikut. Anggap nilai awal flag adalah 0 pada setiap pertanyaan.

a. ADD AX,BX ;ISI AX=7FFFh dan BX=0001h

b. SUB AL,BL ; AL=01h, BL=FFh

c. DEC AL ; AL=00h

d. NEG AL ; AL=7Fh

a. XCHG AX,BX ; AX=1ABCh, BX=712Ah

e. ADD AL,BL ; AL=80h, BL=FFh

f. SUB AX,BX ; AX=0000h, BX=8000h

2. Anggap instruksi ADD AX,BX dieksekusi. Pada setiap bagian berikut, bilangan pertama yang akan ditambahkan berada pada AX dan bilangan kedua berada pada BX. Berikan hasil pada AX dan katakan apakah terjadi *signed overflow* atau *unsigned overflow*

- a. 522Ch
 + 4385h
- b. FF12h
 -1B2Bh
- c. E2F4h
 +D2F3h
- d. 7A32h
 -7200h
- e. 6B89h
 +1276h

3. Buat Program untuk melakukan pengurangan sebagai berikut :

166EFBF-0FEFFFF

High byte disimpan pada variable memory **Tekkom1** dan Low Byte pada Variable **Tekkom2**