Set CF=1 ; No way !

* Utilizand directive

BITS 16

………….

A[7] = \*(A+7); “+ = []”

Mov eax, [a1] ;

Mov eax, [8:1000h];

Segment code

Start:

Jmp Real\_start

V db 17

V1 dw 54321

Real\_Start:

Mov ax, [v]

Add ebx, eax

Mul [v1]

………

Instructiunile de setare a valorilor flag-urilor sunt instructiuni care NU au operanzi expliciti.

Sub 16 biti, adresa unei locatii se putea manipula in mod explicit de catre programator. Sub 32 de biti, calculul de adresa este efectuat EXCLUSIV de catre componenta ADR din BIU de o maniera total opaca (de tip black box) fata de utilizator.

Limita (dimensiunea) = size of segment

Un selector de segment este definit si furnizat de catre sistemul de operare!

La nivelul codului scris de un programator se pot utiliza DOAR SPECIFICARI DE ADRESA, NICIODATA ADRESE fizice finale efective, acestea pot fi gestionate EXCLUSIV de catre componenta ADR din BIU.

Spre deosebire de adresa de inceput de segment asupra careia NU avem NICIUN CONTROL ca si programatori, offset-urile sau deplasamentele trebuie sa le gestionam NOI!

Obligatorii pentru executia oricarui program trebuie sa existe segmentul de COD si cel de STIVA !

Pentru noi ca programatori este obligatoriu sa il scriem pe cel de COD, segmentul de stiva fiind atunci generat automat de catre asamblor.

CODUL MASINA (forma binara) a unui program **.exe** (executabil) NU contine DOAR echivalentul instructiunilor de executat , ci va contine si segmentul de date asociat acestui program generat in mod corespunzator de catre asamblor/compilator!

Deci cu alte cuvinte, segmentul de DATE este si el SALVAT (alaturi de segmentul de COD) ca parte a programului meu **.exe !!**