**`CF si OF exprima eventualele depasiri DOAR pt operatiile de ADUNARE si SCADERE !!!!!!!**

**Setarea lor la valoarea 1 exprimă dpdv semantic INCORECTITUDINEA MATEMATICA a acelei operaţii de adunare sau scădere din respectiva interpretare în baza 10.**

**Inmultirea NU va furniza niciodata depasire deoarece DIN START ii este rezervat spatiu suficient pt reprezentarea rezultatului final !!!!!**

**b\*b à b (2\*3 = 6) à CF=OF=0 (NO « overflow » !!)**

**b\*bà w (255 \* 254 =….) à CF=OF=1 (« Overflow » !!)**

**numar/0 -> operatie imposibila matematic , deoarece numar/0 = infinit**

**In analiza matematica numar/(epsilon ce TINDE la zero) = rezultat ce TINDE la infinit**

**w/b à b 1002/3 = 334 = situatie de depasire (overflow) in cazul impartirii – fatal – Run time error (**‘Divide overflow’, ‘Division by zero’ sau ‘Zero divide’)

**numar/0 – Zero divide = ESTE OVERFLOW PT CA INFINIT NU INCAPE IN NIMIC !!!!!!!!!**

**De ce am nevoie SIMULTAN de CF si OF in EFLAGS ?? Nu ajunge un singur flag pt a imi arata PE RAND daca am sau nu depasire fie in interpretarea cu semn fie in cea fara semn ? NU, pt ca** In momentul efectuarii unei operatii de adunare sau scadere in baza2 se efectueaza de fapt SIMULTAN 2 operatii in baza10: una în interpretarea cu semn şi cealaltă în interpretarea fără semn.

In consecinta e nevoie SIMULTAN de două flaguri diferite care sa se ocupe fiecare de una din cele 2 interpretari posibile in baza 10:

* CF – pt interpretarea fara semn ; OF – pt interpretarea cu semn

Aceasta se întâmplă deoarece operaţia de adunare sau scădere exprimată IN BAZA 2 se efectuează IDENTIC, la fel deci, INDIFERENT DE INTERPRETAREA cu semn sau fără semn a operanzilor şi a rezultatului !!!! Acesta este şi motivul pt care în limbaj de asamblare NU EXISTA IADD sau ISUB ! Pt ca ele şi dacă ar exista NU ar functiona diferit de ADD si respectiv SUB !  
**ADD = IADD, SUB = ISUB – Pt ca in baza 2 operatia exprimata se efectueaza la fel INDIFERENT DE INTERPRETARE !!!!**

* **DE CE AM NEVOIE DE IMUL si IDIV ???**

Pentru ca spre deosebire de adunare si scadere, care functioneaza la fel in baza2 , indiferent de interpretare (cu semn sau fara semn) inmultirea si impartirea CU SEMN si FARA SEMN functioneaza diferit în cazul cu semn comparativ cu cazul fără semn !!

Ca urmare la adunare si la scadere nu exista necesitatea de a preciza ANTERIOR desfasurarii lor cum dorim sa fie interpretati operanzii si rezultatul, deoarece cele 2 operatii oricum functioneaza la fel indiferent de cum dorim sa îi interpretam. Este suficient sa ne decidem ULTERIOR efectuării operaţiei cum dorim sa fie interpretaţi operanzii şi rezultatul.

In schimb inmultirea si impartirea nu functioneaza la fel in cele 2 interpretari, aici existand necesitatea de a preciza ANTERIOR unei inmultiri sau impartiri cum dorim sa fie interpretati operanzii, iar acest lucru se face tocmai prin precizarea MUL şi DIV (dacă dorim operanzi fără semn) sau respectiv IMUL şi IDIV (dacă dorim operanzi cu semn).