

Arhitectura sistemelor de calcul

Curs 3 - flag-uri + complementul

Registrul EFLAGS are 32 de biti dintre care sunt folosiți uzual numai 9. Un flag este reprezentat printr-un bit. configurație a registrului de flag-uri indică un rezultat a execuției unor instrucțiuni.

31	30	...	12	11	10	9	8	7	6	5	4	3	2	1	0
x	x	...	x	OF	DF	IF	TF	SF	ZF	x	AF	x	PF	x	CF

- a) Flag-uri ce raportează efectul generat de UOE: CF, PF, AF, ZF, SF, OF
b) Flag-uri cu efect ulterior stării lor de programator: CF, TF, IF, DF

CF: carry flag este flagul de transport și are valoarea 1 dacă în cazul UOE s-a efectuat transport în/din afara domeniului de reprezentare și valoarea 0 în caz contrar. CF are implicare doar în adunări și scăderi.

De exemplu:
$$\begin{array}{r} 10010011 \\ 01110011 \\ \hline 100000110 \end{array} \Rightarrow CF=1$$
 (când adunare / scădere binară are maxim o cifră de transport)

PF: parity flag este flagul de paritate și are valoarea 1 dacă în cazul UOE octetul cel mai puțin semnificativ conține un număr par de biti care au valoarea 1, în caz

contrar PF are val 0.

De exemplu:

1000	0000	1101	1110	+
0000	1110	1010	1101	
<hr/>				
1000	1111	1000	1011	=> PF=1

AF: auxiliary flag are valoarea 1 dacă în cazul UOE s-a efectuat transport de la bitul 3 la bitul 4, iar în caz contrar are valoarea 0.

De exemplu:

1000	1010	+
0010	1001	
<hr/>		
1011	0011	=> AF=1

(denumirea unei jumătăți de octet este nibble)

ZF: zero flag are valoarea 1 dacă rezultatul UOE este egal cu zero, iar în caz contrar are valoarea 0

De exemplu:

1001	0000	-
1001	0000	
<hr/>		
0000	0000	=> ZF=1

SF: sign flag are valoarea 1 dacă rezultatul UOE este un număr negativ (în interpretarea cu semn), adică bitul high este 1, iar în caz contrar SF are valoarea 0

De exemplu:

0100	0000	+
0100	0000	
<hr/>		
1000	0000	

TF: trap flag dacă are valoarea 1 atunci procesorul se oprește după fiecare instrucțiune. De exemplu când folosim

debuggerul TP are valoarea 1. Dacă rulăm codul normal fără debugger atunci TP are valoarea 0.

IF: interrupt flag are valoarea 1 dacă permite interupserile hardware, adică tastatură, mouse, sistemul nu mai răspunde la comenzi, iar valoarea 0 în caz contrar. De exemplu când $IF = 1$ și apăsăm o tastă, tastatura generează o interupcere, iar procesorul o procesează. Dacă $IF = 0$ procesorul ignorează interupserile.

DF: direction flag are valoarea 0 dacă deplasarea în șir se face de la început spre sfârșit și valoarea 1 dacă deplasarea în șir se face de la sfârșit spre început (ESI, EDI).

OF: overflow flag este flagul de depășire și are valoarea 1 dacă rezultatul VOE nu a încăput în intervalul de reprezentare (signed) și valoarea 0 în caz contrar.

De exemplu:

0	1	1	1	1	1	1	1	+	1	2	7	+
							1			1		
1	0	0	0	0	0	0	0		1	2	8	∉ [-128, 127] ⇒ OF = 1

Instrucțiuni de setare a flag-urilor există doar pentru unele din categorii b deoarece trebuie să avem la dispoziție instrucțiuni de setare a flag-urilor cu efect ulterior, acestea sunt în număr de 7 și nu au operații explicite:

CLC: clear carry flag ($CF = 0$)

STC: set carry flag ($CF = 1$)

CMC: complement carry flag

CLD: clear direction flag ($DF = 0$)

STD: set direction flag ($DF = 1$)

CLI: clear interrupt flag ($IF = 0$)

STI: set interrupt flag ($IF = 1$)

} au efect doar în programarea în 16 biți (putem

programa în 16 biți dacă folosim la începutul programului
directiva `bits 16` la începutul programelor)

Complementul față de 2

10010011 = 93h = 147, deci în interpretarea fără semn 10010011 = 147. Fiind un număr care începe cu 1 în interpretarea cu semn, acest număr este negativ, iar valoarea sa este -(complementul față de 2 al configurației inițiale) = -(01101101) = -109

Cum calculăm complementul?

$$\begin{array}{r} 10000000 \\ - 10010011 \\ \hline 01101101 \end{array}$$

binar

hexa

fără semn

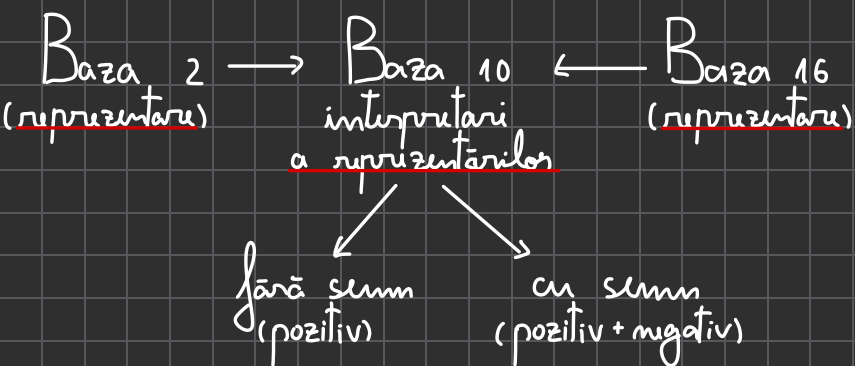
cu semn

10010011 +	93h +	147 +	-109 +
01101101	73h	115	115
100000110	106h	262	6

P
Pe 8 biti se pot reprezenta 2^8 valori = 256 valori ^{unsigned} [0...255] sau ^{signed} [-128...+127]

P
Pe 16 biti se pot reprezenta 2^{16} valori = 65536 valori ^{unsigned} [0...65536] sau ^{signed} [-32768...+32767]

P
Pe 32 biti se pot reprezenta 2^{32} valori ...



Ex: Să se exprime numărul 10010011 în bazele 2, 10, 16 în fiecare interpretare (cu semn, fără semn)

binar: 10010011

decimal $\begin{cases} \text{cu semn: } 147 \\ \text{fără semn: } -109 \end{cases}$

hexadecimal: 93h

Pe n biți se pot reprezenta 2^n valori: fie valorile $[0, 2^n - 1]$ în interpretarea fără semn sau valorile $[-2^{n-1}, 2^{n-1} - 1]$ în interpretarea cu semn.

Ex: Căru este numărul minim de biți pe care se poate reprezenta -147?

$$-147 \in [-256, 255] = [-2^{n-1}, 2^{n-1} + 1], n = 9$$

Suma valorilor absolute a celor două valori complementare trebuie să fie cardinalul intervalului de reprezentare a valorilor.

Ex: 1 0 1 1 0 0 1 1 $\begin{cases} ? & \text{interpretarea cu semn} \\ ? & \text{interpretarea fără semn} \end{cases}$

1 0 1 1 0 0 1 1 = 179 (fără semn) \Rightarrow cu semn

avem $-(0 1 0 0 1 1 0 1) = -77$, $179 + |-77| = 256$ semn

1 0 1 1 0 0 1 1 + 0 1 0 0 1 1 0 1 = 100000000 = 256