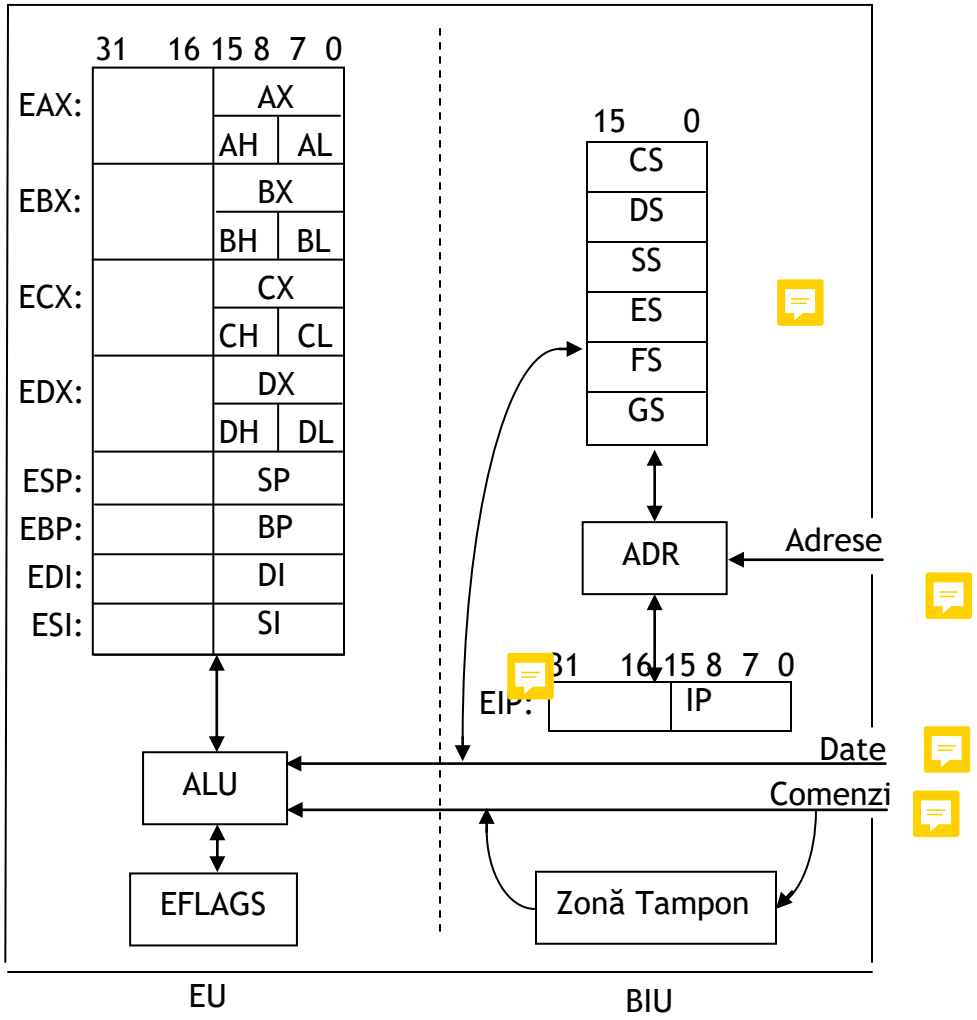
2.6. ARHITECTURA MICROPROCESOARELOR x86 (IA-32)

2.6.1. Structura microprocesorului

Microprocesorul x86 este format din două componente principale:

– **EU** (*Executive Unit*) - execută instr. maşină prin intermediul componentei **ALU** (*Aritmetic and Logic Unit*).

– **BIU** (*Bus Interface Unit*) - pregăteşte execuţia fiecărei instrucţiuni maşină. Citeşte o instrucţiune din memorie, o decodifică şi calculează adresa din memorie a unui eventual operand. Configuraţia rezultată este depusă într-o zonă tampon cu dimensiunea de 15 octeţi, de unde va fi preluată de **EU**.



EU si **BIU** lucrează în paralel - în timp ce **EU** execută instrucţiunea curentă, **BIU** pregăteşte instrucţiunea următoare. Cele două acţiuni sunt sincronizate - cea care termină prima aşteaptă după cealaltă.

2.6.2. Regiştrii generali EU

Registrul **EAX** este *registrul acumulator*. El este folosit de către majoritatea instrucţiunilor ca unul dintre operanzi.

Registrul **EBX** - *registru general*

Registrul **ECX** - *registru de numărare (registru contor)* pt instr care au nevoie de indicaţii numerice.

Registrul **EDX** - *registru de date*. Împreună cu EAX se foloseşte în calculele ale căror rezultate depăşesc un dublucuvânt (32 biţi).

"Word size" refers to the number of bits processed by a computer's CPU in one go (these days, typically 32 bits or 64 bits). Data bus size, instruction size, address size are usually multiples of the word size.

Just to confuse matters, for backwards compatibility, Microsoft Windows API defines a WORD as being A DATA TYPE on 16 bits, a DWORD being A DATA TYPE on 32 bits and a QWORD as 64 bits, regardless of the processor.

Regiştrii **ESP** şi **EBP** sunt regiştri destinaţi lucrului cu *stiva*. O stivă se defineşte ca fiind o zonă de memorie în care se pot depune succesiv valori, extragerea lor ulterioară făcându-se în ordinea inversă depunerii.

Registrul **ESP** (*Stack Pointer*) punctează spre elementul ultim introdus în stivă (elementul din *vârful stivei*).

Registrul **EBP** (*Base pointer*) punctează spre primul element introdus în stivă (indică *baza stivei*).

Regiştrii **EDI** şi **ESI** sunt *regiştrii de index* utilizaţi de obicei pentru accesarea elementelor din şiruri de octeţi sau de cuvinte. Denumirile lor (*Destination Index* şi *Source Index*) precum şi rolurile lor vor fi clarificate în cap. 4.

Fiecare dintre regiştrii EAX, EBX, ECX, EDX, ESP, EBP, EDI, ESI au capacitatea de 32 biţi. Fiecare dintre ei poate fi privit în acelaşi timp ca fiind format prin concatenarea (alipirea) a doi (sub)regiştri de câte 16 biți. Subregistrul superior, care conţine cei mai semnificativi 16 biţi ai registrului de 32 biţi din care face parte, nu are denumire și nu este disponibil separat. Subregistrul inferior poate însă fi accesat individual, având astfel regiștrii de 16 biți **AX, BX, CX, DX, SP, BP, DI, SI**. Dintre aceștia, regiștrii AX, BX, CX, si DX sunt fiecare la rândul lor, formați din câte doi alți subregiștri a câte 8 biți. Există astfel regiştrii **AH, BH, CH, DH**, conținând cei 8 biți superiori (partea HIGH a regiștrilor AX, BX, CX și DX), respectiv **AL, BL, CL, DL**, conținând cei 8 biți inferiori (partea LOW).

2.6.3. <u>Flagurile</u>

Un *flag* este un indicator reprezentat pe un bit. O configuraţie a *registrului de flaguri* indică un rezumat sintetic a execuţiei fiecărei instrucţiuni. Pentru x86 registrul EFLAGS (the *status* register) are 32 biţi dintre care sunt folosiţi uzual numai 9.

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

CF (*Carry Flag*) este flagul de transport. Are valoarea 1 în cazul în care în cadrul ultimei operatii efectuate (UOE) s-a efectuat transport în afara domeniului de reprezentare a rezultatului si valoarea 0 in caz contrar.

PF (*Parity Flag*) - Valoarea lui se stabileşte a.î. împreună cu numărul de biţi 1 din octetul cel mai puţin semnificativ al reprezentarii rezultatului UOE să rezulte un număr impar de cifre 1.

AF (*Auxiliary Flag*) indică valoarea transportului de la bitul 3 la bitul 4 al rezultatului UOE. De exemplu, în adunarea de mai sus transportul este 0.

ZF (*Zero Flag*) primeşte valoarea 1 dacă rezultatul UOE este egal cu zero şi valoarea 0 la rezultat diferit de zero.

SF (*Sign Flag*) primeşte valoarea 1 dacă rezultatul UOE este un număr strict negativ şi valoarea 0 în caz contrar.

TF (*Trap Flag*) este un flag de depanare. Dacă are valoarea 1, atunci maşina se opreşte după fiecare instrucţiune.
IF (*Interrupt Flag*) este flag de întrerupere. Detalii în cap.5 din manual.

DF (*Direction Flag*) - pt operare asupra şirurilor de octeţi sau de cuvinte. Dacă are valoarea 0, atunci deplasarea în şir se face de la început spre sfârşit, iar dacă are valoarea 1 este vorba de deplasări de la sfârşit spre început.

OF (*Overflow Flag*) este flag pentru depăşire **CU SEMN**. Dacă rezultatul ultimei instrucţiuni în interpretarea CU SEMN a operanzilor nu a încăput în spaţiul rezervat operanzilor (intervalul de reprezentare admisibil), atunci acest flag va avea valoarea 1, altfel va avea valoarea 0.

Categorii de flag-uri

Flag-urile se pot împărţi în două categorii :
a). cu efect anterior generat de către Ultima Operaţie Efectuată (UOE) : CF, PF, AF, ZF, SF şi OF
b). cu efect ulterior setării lor de către programator pentru influenţarea modului de operare al instrucţiunilor care urmează : CF, TF, DF şi IF.

Instrucţiuni specifice de setare a valorilor unor flag-uri

Având în vedere categoria b) este normal ca limbajul de asamblare să ne pună la dispoziţie intrucţiuni specifice de setare a valorii flag-urilor care vor avea un efect ulterior. Aceste instrucţiuni sunt în număr de 7:

CLC – efectul fiind CF=0,
STC – efectul fiind CF=1,
CMC – complementarea valorii din CF

CLD – avand ca efect DF=0,
STD – avand ca efect DF=1

CLI – avand ca efect IF=0, //nu pot fi activate in mod protejat
STI – avand ca efect IF=1

Având în vedere riscul major de setare accidentală a valorii din TF precum şi rolul său absolut special în dezvoltarea de depanatoare, NU există disponibile instrucţiuni de acces direct la valoarea din TF !!