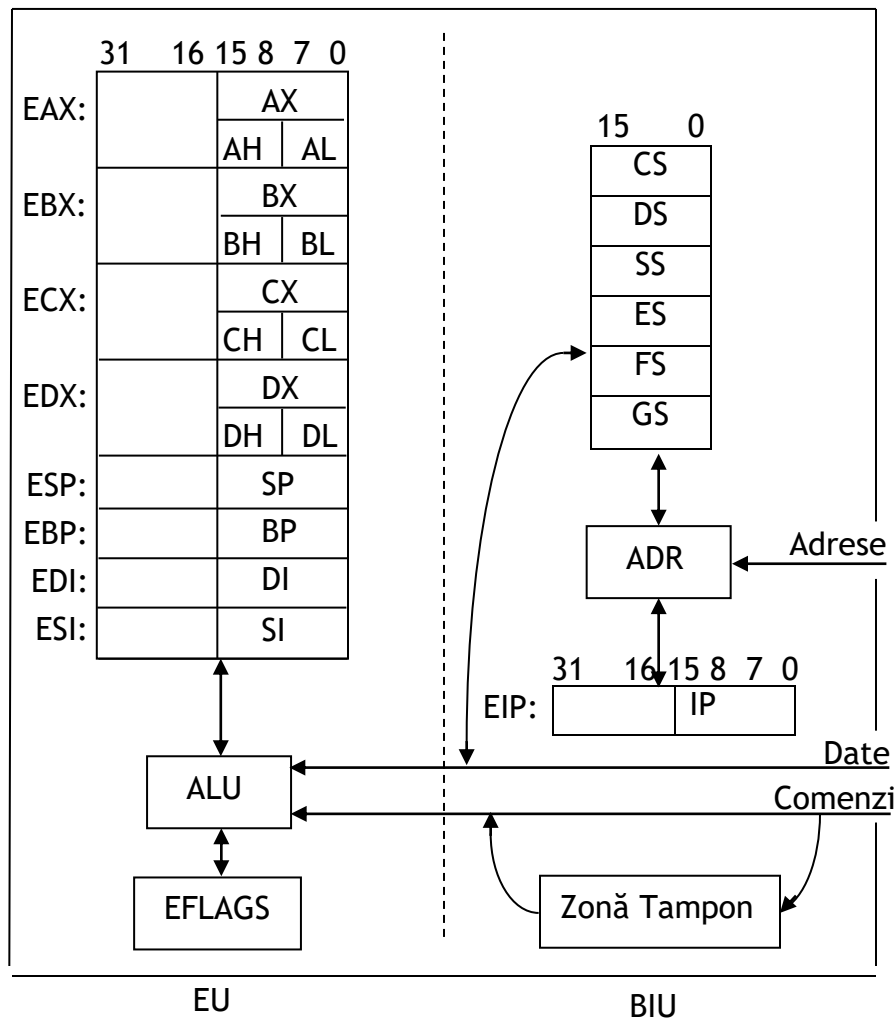


2.6. ARHITECTURA MICROPROCESOARELOR x86 (IA-32)

2.6.1. Structura microprocesorului

Microprocesorul x86 este format din două componente principale:

- **EU** (*Executive Unit*) - execută instr. mașină prin intermediul componentei **ALU** (*Aritmetic and Logic Unit*).
- **BIU** (*Bus Interface Unit*) - pregătește execuția fiecărei instrucțiuni mașină. Citește o instrucțiune din memorie, o decodifică și calculează adresa din memorie a unui eventual operand. Configurația rezultată este depusă într-o zonă tampon cu dimensiunea de 15 octeți, de unde va fi preluată de **EU**.



EU si **BIU** lucrează în paralel - în timp ce **EU** execută instrucțiunea curentă, **BIU** pregătește instrucțiunea următoare. Cele două acțiuni sunt sincronizate - cea care termină prima așteaptă după cealaltă.

2.6.2. Regiștrii generali EU

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

Registrul **EBX** - *registru general* (provine ca denumire de la Base Register – registru de baza – folosit în această accepțiune la programarea pe 16 biți).

Registrul **ECX** - *registru de numărare (registru contor)* pt instr (repetitive) care au nevoie de indicații numerice (bucle – limite de ciclare etc).

Registrul **EDX** - *registru de date*. Împreună cu EAX se folosește în calculele ale căror rezultate depășesc un dublucuvânt (32 biți).

"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.

Regiștrii **ESP** și **EBP** sunt regiștri destinați lucrului cu *stivă*. O stivă se definește 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*).

Regiștrii **EDI** și **ESI** sunt *regiștrii de index* utilizați de obicei pentru accesarea elementelor din șiruri de octeți sau de cuvinte. Denumirile lor (*Destination Index* și *Source Index*) precum și rolurile lor vor fi clarificate în cap. 4.

Fiecare dintre regiștrii EAX, EBX, ECX, EDX, ESP, EBP, EDI, ESI au capacitatea de 32 biți. Fiecare dintre ei poate fi privit în același timp ca fiind format prin concatenarea (alipirea) a doi (sub)regiștri de câte 16 biți. Subregistrul superior, care conține cei mai semnificativi 16 biți ai registrului de 32 biți 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, și DX sunt fiecare la rândul lor, formați din câte doi alți subregiștri a câte 8 biți. Există astfel regiștrii **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).