MICROPROCESORUL X86

**Microprocesorul x86** este format din doua componente prncipale:

* **BIU** (Bus Interface Unit) – pregateste executia fiecarei instructiuni masina. Citeste o instructiune din memorie, o decodifica si calculeaza adresad in memorie a unui eventual operand. Configuratia rezultata este depusa intr-o zona tampon cu dimensiunea de 15 octeti, de unde va fi preluata de EU
* **EU** (Executive Unit) – executa instructiunile masina prin intermeiul componentei **ALU** ( Aritmetic and Logic Unit)

EU si BIU lucreaza in paralel – in timp ce EU executa instructiunea curenta, BIU pregateste urmatoarea instructiune. Cele doua actiuni sunt sincronizate – cea care termina prima o asteapta pe urmatoarea.

**Registrii** sunt capacitati de memorare foarte mici (8, 16, 32, 64) dar foarte rapide ca viteza de acces, utilizati pt stocarea temporara a operanzilor instructiunilor cu care opereaza in mod curent un procesor.

**Registrii generali EU**

Fiecare dintre registrii generali EAX, EBX, ECX, EDX, ESP, EBP, EDI, ESI au capacitate de 32 de biti. Fiecare dintre ei poate fi privit in acelasi timp ca fiind format prin concatenarea a doi (sub)registrii de cate 16 biti. Registrul superior, care contine cei mai semnificativi 16 biti, nu are denumire si nu este disponibil separate. Subregistrul inferior este individual si poate fi accesat astfel prin: AX, BX, CX, DX, SP, BP, DI, SI. Dintre acestia, registrii AX, BX, CX, DX sunt formati la randul lor din lati doi subregistrii a cate 8 bti. Exista astfel registrii continand cei 8 biti superiori (partea high) AH, BH, CH, DH, respective registrii ocntinand cei 8 biti inferiori (partea low) AL, BL, CL, DL.

EAX (registrul acumulator)

* cel mai utilizat registru din procesor
* folosit de majoritatea instructiunilor procesorului drept unul dintre operanzi in operatii aritmetice

EBX (registrul de baza) ex: a[7] a🡪baza 7🡪index

* folosit ca pointer la date

ECX (registrul contor)

* folosit in principal pt instructiuni de ciclare indicand nr de pasi ce trebuie efectuat sau pentru instructiuni care au nevoie de indicatii numerice

EDX (registru de date)

* folosit impreuna cu EAX in calculele ale caror rezultate depasesc un double word (32 biti)

ASC CURS 2 – 11.10.2023

**STIVA** = o zona de memorie in care se pot depune succesiv valori, extragerea lor facandu-se in ordinea inversa depunerii

ESP (pointer de stiva) – puncteaza spre ultimul element introdus in stiva (varful stivei)

EBP (pointer de baza) – puncteaza spre primul element introdus in stiva (baza stivei)

SS (Stack Segment)

EDI (destination index) + ESI (source index) – utilizati pt accesarea elem din siruri de byte sau word

**Exista 3 registrii de lucru cu stiva (prioritara pt procesor) si 0 registrii pt coada**

– ordinea de activare/dezactivare a functiilor dintr un program este ordinea activata de stiva, nu de coada, respectand disciplina LIFO (last in first out); spatiul de executie a oricarui program fiind stiva de executie (Run-Time Stack)

**De ce sunt 3 registrii pt lucrul cu stiva? ESP+EBP+ SS**

– baza reala a stivei ca zona de memorie este SS, iar BP este baza stack frameului principal curent

**ASM nu stie ce inseamna notiunea de tablou**. Sirurile trebuie sa fie simulate de catre programator prin gestionarea unei baze (inceputul sirului) si a unui index (gestionat prin EDI sau ESI).