

## Arhitectura Calculatoarelor

### T.5 – Procesorul

Cipul microprocesorului. Unități funcționale. Unitatea centrală de prelucrare. Caracteristicile microprocesorului. Istoric al evoluției microprocesoarelor. Caracteristicile ultimelor microprocesoare. Exemple de microprocesoare – Intel, UltraSPARC II, INTEL 8086. ARHITECTURA ȘI FUNCȚIONAREA UCP. Descrierea unităților funcționale. Caracteristici arhitecturale. Schema generală a unui microprocesor 8086 Modelul programatorului de assembler. Ciclul instrucțiune. Registrele microprocesorului Adresarea memoriei. Segmentarea memoriei. Formatul instrucțiunii. Moduri de adresare

Scopul Lecției: De a face cunoștință cu istoricul dezvoltării microprocesoarelor. Caracteristicile lor de bază și Schema generală a microprocesorului 8086

Studentul trebuie să cunoască:

§ Noțiunea de microprocesor

§ Caracteristicile microprocesoarelor

§ Arhitectura și funcționarea 8086

§ Modelul programatorului assembler

§ Registrele microprocesorului 8086

§ Segmentarea memoriei. Formatul instrucțiunii. Moduri de adresare

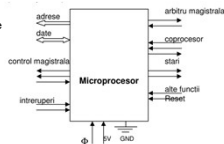
Conf. Univ. Dr. Crețu Vasilii

### Cipul microprocesorului

Microprocesorul este "creierul" calculatorului. El este o unitate centrală de procesare încorporată într-o singură pastilă de circuit integrat (un cip sau chip) care citește instrucțiunile unui program depus în memoria principală, le decodifică și le execută secvențial (una după alta).

Cipurile microprocesor comunica cu lumea exterioară prin "pini". În funcție de direcția pe care circula semnalele acestea se clasifică în:

- pini de intrare – prin care microprocesorul primește semnale de la celelalte unități funcționale ale sistemului de calcul
- pini de ieșire – prin care microprocesorul transmite semnale celorlalte unități funcționale
- pini de intrare/ieșire – prin care microprocesorul poate primi și transmite semnale.



În funcție de tipul datelor vehiculate prin pini avem:

- pini de adresă
- pini de date
- pini de control (magistrală, intreruperi, semnalizare, etc.)
- pini de stare
- pini de alimentare și masa

### Unități funcționale

Unitățile funcționale componente ale microprocesoarelor actuale sunt:

**CPU** (Central Processing Unit) – unitatea centrală de prelucrare. Reprezintă unitatea de procesare (execuție) a instrucțiunilor, aritmetica numerelor întregi și de coordonare a întregului sistem.

**FPU** (Floating Point Unit) – unitatea în virgulă mobilă, specializată în aritmetica numerelor reale reprezentate în format virgulă mobilă (standard IEEE 754)

**MMU** (Memory Management Unit) – unitatea de gestiune a memoriei, realizează automat gestiunea memoriei.

**MMX** (MultiMedia eXtension) – unitate multimedia specializată în prelucrări grafice

**Unitatea centrală de prelucrare**

CPU include următoarele unități funcționale:

- UAL (**Unitatea Aritmetică și Logică**) reprezintă o unitate combinațională cu două intrări și o ieșire care execută operații aritmetice și logice.
  - **UC (Unitatea de Comandă)** este unitatea funcțională care programează execuția secvențială a tuturor operațiilor necesare efectuării instrucțiunilor, generând semnale de comandă pentru tot sistemul, dirijând fluxul de date, corelând viteza de lucru a unității centrale de prelucrare cu memoria, etc.. Activitatea unității de comandă este pilotată de un *semnal de ceas* a cărui frecvență este acum de ordinul sutelor de MHz.
  - **Registrii** Aceștia reprezintă elemente de memorare în care se stochează temporar date sau adrese. Unii sunt folosiți pentru urmărirea execuției instrucțiunilor (registrii de contor de program), alții sunt folosiți în calcule (registri cu scop general), alții păstrează starea programului în execuție (registru de stare), alții pentru calculul adreselor de memorie (registri de adresă). Aceștia reprezintă cea mai rapidă formă de memorie din sistem fiind direct conectați la UAL.
- Efectuarea transferurilor de date și comenzi între unitățile funcționale ale microprocesorului se face pe **magistrala internă de date** a microprocesorului. Semnalele electrice prin care microprocesorul dă comenzi de execuție către memorie și către celelalte componente din sistem se numesc **semnale de comandă**. Semnalele electrice prin care microprocesorul culege informații privind starea componentelor din sistem se numesc **semnale de stare**.

*lungimea* (numărul de biți) regiștrilor interni se corelează de obicei cu *lățimea* (numărul de linii) ale magistralei de date. Aceasta e **măsura numărului de biți** ai microprocesorului. Microprocesoarele cu structură fică sunt de 8,16,32,64biți. Lungimea de curvânt a microcalculatoarelor realizate cu microprocesoare « bit slice » (felii de bit), a căror structură e flexibilă, va fi un multiplu întreg al numărului de biți ai unei felii. Registrul de adrese, respectiv lățimea magistralei de adrese definește spațiul de memorie adresabil direct de microprocesor. O magistrală de adrese de 16 biți permite adresarea a  $2^{16}=65536$  celule distincte, iar 20 linii de adresă ne duc în lumea megaoctetilor:  $2^{20}=1.048.576$  celule adresabile.

Memoria adresată direct de microprocesor se poate împarti în memorie program și memorie de date. Memoria program conține instrucțiuni executabile de către microprocesor, iar memoria de date date utilizate de instrucțiunile programului. Datele utilizate în program pot fi adrese (de adresare) sau date stocate în memorie (adresate).

**Caracteristicile microprocesorului :**

- Frecvența ceasului
- Lățimea magistralei de date
- Lățimea magistrală de adrese
- Setul de instrucțiuni
- Arhitectura UCP

**Istoric al evoluției microprocesoarelor****Microprocesoare INTEL.**

1964	Gordon Moore – enunt lege "numarul de tranzistori planari pe pastiala de siliciu se dubleaza anual"
⇒1986	primul CI pe 222 elemente (de 4 biți)
1968	firma Intel a luat fiinta (Robert Noyce ← Gordon Moore) inventator al CI
1969	primul RAM static – INTEL
1970	primul RAM dynamic – INTEL
1971	- primul microprocesor – INTEL 4004 pe 45 date, 125 adrese, ceas de 740 KHz, PMOS - primul EPROM – avantajul posibilitatii stergerii informatiei cu raze ultraviolete TED HOFF - inventator al microprocesorului - microprocesorul a fost privit ca un circuit programabil ce poate inlocui logica cablata
1972	- microprocesorul 8008 Intel: - 8 biți date, 14 biți adrese (16 Koct memorie adresabila), ceas 800 KHz - set de instrucțiuni simple pentru operatii aritmetice (adunare, scadere) și logice pentru operanzi pe 85 - stiva implementare hardware, mica

1973-1977	<ul style="list-style-type: none"> <li>- perioada specifica microprocesorului pe 8 biti si adrese 16 biti</li> <li>- set de instructiuni extinse pentru manipulare date pe 16 biti (se pot evalua adrese)</li> <li>- stiva mutata din CPU in memorie =&gt; posibilitatea de manipulare a subrutinelor imbricate</li> <li>- limbaj de programare -&gt; limbaj de asamblare</li> <li>- posibilitate de utilizare a perifericelor specializate in aritmetica intregilor extinsa pe axa numerelor reale in format VM</li> </ul> <p>→ dar acestea aveau viteza mica =&gt; consumuri suplimentare (overhead) mari</p> <ul style="list-style-type: none"> <li>- reprezentant tipic microprocesorul 8080 (1974): <ul style="list-style-type: none"> <li>- domenii de aplicatie : sisteme in timp real, sisteme pentru conducerea proceselor industriale</li> <li>- tehnologii de realizare NMOS =&gt; viteza mare ceas 2-4 MHz =&gt; viteza mare de executie a instructiunilor</li> </ul> </li> <li>- alte microprocesoare - Intel 8085, Z80 (ZILLOG)</li> <li>- Intel 8086 - 16 biti date, 20 biti adrese, ceas 4 - 8 MHz</li> <li>- intel 8088 - compatibil cu circuite existente la acea vreme(1979) =&gt; idem 8086 dar date pe 8 biti</li> </ul> <p>Caracteristicile microprocesoarelor de la inceputul anilor '80</p> <ul style="list-style-type: none"> <li>- extinderea lungimii cuvintelor de date la 16 biti si adrese 20 biti (1 Moct adresabil)</li> <li>- aritmetica numerelor intregi extinsa la 32 biti (usurinta in manipulare adreelor)</li> <li>- s-au introdus instructiuni pentru manipularea sirurilor cu lungime variabila de biti</li> <li>- s-a perfectionat mecanismul CALL/RETURN, cu posibilitati sporite de implementare a limbajului HLL si recursive</li> <li>- au aparut noi tipuri de coprocesoare pentru prelucrarea in VM</li> <li>- avans in tehnologia de realizare CMOS</li> </ul>
-----------	---

1982	- Intel 286 - 16 biti date, 24 biti adrese (16 Moct), ceas 6 ÷ 16 MHz
1984	<ul style="list-style-type: none"> <li>- apare primul calculator PC AT cu microprocesor 286 la baza</li> <li>- cea mai importanta facilitate - cea legata de protectia memoriei ceea ce presupune o organizare hardware speciala destinata oferirii de suport pentru taskuri specifice sistemului de operare</li> <li>- s-au dezvoltat noi tipuri de aplicatii: <ul style="list-style-type: none"> <li>- legate de calculatoarele personale</li> <li>- statii de lucru CAD</li> <li>- sisteme functionand cu divizarea timpului (time sharing)</li> </ul> </li> <li>- aplicatiile datorita gradului de paralelism oferite in executia instructiunilor se scriu in limbaje de nivel inalt (HLL)</li> <li>- posibilitatea accesarii de memorie virtuala - implementat deocamdata software ptr MMU</li> </ul>
1985	- Intel 386 - 32 biti date - cu FPU integrat pe cip - cache - incorporat
1989	- Intel 486 - viteza > 386 - cu FPU integrat pe cip - cache - incorporat
1993	- Pentium - frecventa 60 - 233 MHz - 2 benzi de asamblare - componenta MMX - unele modele mai noi
1995	- Pentium Pro - memorie intermediara pe 2 nivele incorporata 150 - 200 MHz

1997	<ul style="list-style-type: none"> <li>- Pentium II - Pentium PRO+ MMX</li> <li>- 233 MHz → Celeron - performanta scazuta</li> <li>→ Xeon - profesional</li> </ul>
1999	- Pentium III - 400 - 700 MHz
2000	<ul style="list-style-type: none"> <li>- Pentium IV (Titanium)</li> <li>- frecventa &gt; 1,4 GHz</li> <li>- arhitectura Netburst (suport pentru Internetul vizual)</li> <li>- ALU - se numeste Rapid Execution Engine</li> <li>- se prelucreaza comenzi la jumatate de ciclu de tact</li> <li>- Pentium IV la 1,4 GHz = 3x mai rapid la Pentium III la 1000 MHz = 7% mai rapid ca Pentium III la 1400 MHz (29 milioane tranzistori)</li> <li>- 42 milioane tranzistori</li> <li>- suport Internet → schimburi de date direct intre PC- urile utilizatorului internet</li> <li>→ strategia Netster</li> </ul>

**Caracteristicile ultimelor microprocesoare**  
 - bazele pe existenta a 4 unitati majore :  
 • CPU (unitatea centrala de prelucrare)  
 • FPU (unitatea de prelucrare in VM)  
 • MMU (unitatea de gestiune a memoriei)  
 • MMX (unitatea multimedia)

CPU - pentru executarea lucrarilor de uz general  
 FPU - specializata in operatii aritmetice in VM  
 MMU - suport pentru functii de memorie virtuala si suport hard pentru protectia memoriei  
 MMX - suport pentru functii multimedia de prelucrare video si sunet  
 - pe langa acestea, ultimele microprocesoare mai cuprind pe placheta circuite ce inainte erau

exteriorizate :- controller de memorie cache  
 - coprocesor de periferice pentru operatii rapide de I/O  
 - suport pentru grafica de mare viteza  
 - suport pentru multiprocesare

Aria disponibila a pastilei de siliciu s-a utilizat in 2 scopuri :  
 1. oferirea de suport pentru implementarea sistemelor complexe de operare prin introducerea de hardware specializat si instructiuni specifice  
 2. executia eficienta a programelor scrise in HLL prin utilizarea unor moduri de adresare si a unor instructiuni mai complexe.

---

---

---

---

---

---

---

---

---

---

**Exemple de microprocesoare – Intel**

Facilitati speciale:	286	- gestiunea memoriei virtuale - implementarea de medii protejate pentru executie - gestiunea task-urilor in regim de multiprocesare - gestiunea memoriei - se face de MMU integrata cu implementări speciale pentru cod, date, stiva, date suplimentare - teste implementate hard - privind corectitudinea accesului la memorie : - limita segmentelor - separarea task-urilor de cele privilegiate - validitatea indicatorilor si operatiilor cu subrutine
	386	- viteza mai mare - facilitati de optimizare in regim multitasking si suport SO - spatiu de adresare marit - posibilitati de testare si depanare
	486	- conducta de executie (pipeline) - cache pentru date - coprocesor aritmetic pe chip

---

---

---

---

---

---

---

---

---

---

	Pentium	- integrare inalta (3 milioane tranzistori si trasee = 0,5 µm) - arhitectura superscalară – prin executarea instructiunilor cu tehnologie pipeline 2 conducte in 5 trepte → prefetch → de code1 → de code2 → executie → writeback ⇒ mai multe instructiuni pe ciclu instructiuni sunt executate - anumite instructiuni sunt cablate nu microcodate - cache suportat pentru cod si date de cate 8 Koct. – asociative cu 2 cai pentru cautarea datelor pe 32 biti fara a baleia intreaga memorie cache de date – foloseste 2 tehnici: → writeback – ce transfera datele in memoria principala numai cand sunt solicitate prin metoda write through – se transfera de cate ori erau rescrise in cache ⇒ performanta sistemului prin reducerea utilizarii magistralelor → protocolul MESI (Modified Exclusive Shared Invalid) – pentru consistenta datelor - predictia Branch – ce se refera la salturile pentru continuarea executiei unei branch sau iesirea din el si se bazeaza pe predictia ca branch-ul anterior se va folosi din nou. Se folosesc doua buffere de prefetch → unul realizeaza prefetch-ul pentru urmatoarea instructiune → altul pentru adresa de la inceputul buclei
--	---------	--

---

---

---

---

---

---

---

---

---

---



UltraSPARC II	
1970	- Andy Bachtolsheim – la Universitatea Stanford a realizat prima statie de lucru in retea numita SUN 1 (Standford University Network) → cu microprocessor
1982	- fondare Sun Microsystem SUN2, SUN3 – folosesc microprocessor Motorola - cu placa de retea conexiune Ethernet, soft TCP/ IP
1987	- Sun realizeaza primul UCP propriu cu tehnologii RISC II numit SPARC (Scalable Processor ARChitecture = arhitectura scalabila de procesor) masina pe 32 biti la 36 MHz instructiuni pentru intregi
1995	- UltraSPARC – versiunea 9 (V9) - pe 64 biti - prelucrare, imagini si semnale audio/video - set de noi instructiuni numite VIS = Visual Instruction Set - destinat aplicatiilor profesionale ca servere Web multiprocessor cu zeci de UCPUri si memorie fizica de ordinul 2 TB (1 terabyte = 1012 octeti)
UltraSPA RC I – UltraSPA RC II –	cu frecventa crescuta de ceas

---

---

---

---

---

---

---

---

---

---

Cipuri PicoJava	
- Sun defineste o masina virtuala numita JVM cu memorie de 32 biti, 226 instructiuni. - exista compiler care compileaza ptr JVM. - exista interpretor pentru masina compilatoare (compilare + executie) - majoritatea browserelor au interpretor Java pentru a executa appleturi - exista cipuri JVM hardware – pe care lucreaza direct programe executabile JVM fara a necesita interpretare sau compilare  firma Sun – PicoJava II	

---

---

---

---

---

---

---

---

---

---

INTEL 8086 ARHITECTURA ȘI FUNCȚIONAREA UCP	
<b>Componente funcționale</b> UCP include următoarele unități funcționale:	
- UAL (Unitatea Aritmetica si Logica) reprezinta o unitate combinatională cu două intrări și o ieșire care executa operații aritmetice si logice.	
- UC (Unitatea de Comanda) este unitatea functionala care programează execuția secvențială a tuturor operațiilor necesare efectuării instrucțiunilor, generând semnale de comandă pentru tot sistemul, dirijând fluxul de date, corelând viteza de lucru a unității centrale de prelucrare cu memoria, etc.. Activitatea unității de comandă este pilotată de un <i>semnal de ceas</i> a cărui frecvență este acum de ordinul sutelor de MHz.	
- <b>Registrii</b> Aceștia reprezintă elemente de memorare în care se stochează temporar date sau adrese. Unii sunt folosiți pentru urmărirea execuției instrucțiunilor (registrii de stare), alții sunt folosiți în calcule (registri cu scop general), alții păstrează starea programului în execuție (registri de stare), alții pentru calculul adreselor de memorie (registri de adresă). Aceștia reprezintă cea mai rapidă formă de memorie din sistem fiind direct conectați la UAL.	
Efectuarea transferurilor de date și comenzi între unitățile funcționale ale microprocesorului se face pe <b>magistrala internă de date</b> a microprocesorului. Semnalele electrice prin care microprocesorul dă comenzi de execuție către memorie și către celelalte componente din sistem se numesc <b>semnale de comandă</b> . Semnalele electrice prin care microprocesorul culege informații privind starea componentelor din sistem se numesc <b>semnale de stare</b> .	

---

---

---

---

---

---

---

---

---

---

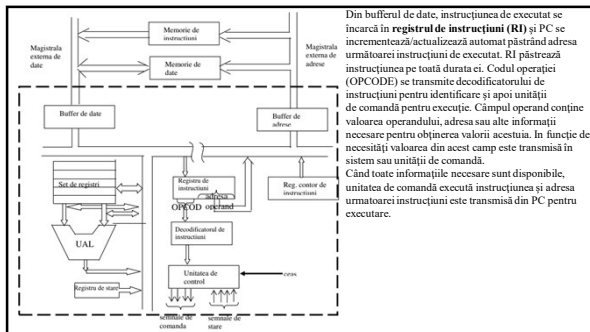
**Lungimea** (numărul de biți) regiștrilor interni se corelează de obicei cu **lățimea** (numărul de linii) ale magistralei de date. Aceasta e **măsura numărului de biți ai microprocesorului**. Microprocesoarele cu structură fixă sunt de 8, 16, 32, 64 biți. Lungimea de cuvânt a microcalculatoarelor realizate cu microprocesoare « bit slice » (felii de biți), a căror structură e flexibilă, va fi un multiplo înțreg al numărului de biți ai unei felii.

Regiștrul de adrese, respectiv lățimea magistralei de adrese definește spațiul de memorie adresabil direct de microprocesor. O magistrală de adrese de 16 biți permite adresarea a  $2^{16}=65536$  celule distincte, iar 20 linii de adresă ne duc în lumea megaocteților:  $2^{20}=1.048.576$  celule adresabile.

Cuvântul instrucțiune conține un număr de biți ce exprimă cimpul codului operației (**OPCODE**) și un cimp corespunzător valorii sau adresei operandului.

**Funcționarea UCP**

Programul executat de microprocesor se află stocat în memorie în format codificat. Fiecare instrucțiune este codificată corespunzător formatului anterior specificat (OPCODE și operand). Instrucțiunile sunt memorate în ordinea de execuție. Un regiștru special păstrează ordinea de execuție a instrucțiunilor prin adresa următoarei instrucțiuni de executat. Acesta este regiștrul contor de instrucțiuni, numit și PC (Program counter) sau numărător de adrese. Pentru executarea unui program, PC se încarcă cu adresa primei instrucțiuni de executat. Această adresă este trimisă către memorie pentru a se obține codul instrucțiunii de executat. Pentru menținerea liniilor de adresă pe durata citirii memoriei se folosește un **regiștru tampon de adrese AB (Address Buffer)**. Informația codificată, citită din memorie este depusă într-un **regiștru tampon de date DB (Data Buffer)**. Liniiile electrice pe care se generează cuvântul binar de adresă formează **magistrala de adrese**, iar cele dedicate datelor citite/scrise din memorie **magistrala de date**.



**Descrierea unităților funcționale**

**UAL** este circuitul din structura microprocesorului care procesează informația realizând operații aritmetice și logice. Este un circuit combinational cu două intrări și o ieșire și necesită registre de memorare temporară atât pentru intrări cât și pentru ieșire. Rezultatul operației este stocat în tot într-unul din regiștrii de intrare. Acești regiștri se numesc de tip acumulator.

1. **Numaratorul de adrese** (program counter, PC) pastreaza adresa locației care conține următoarea instrucțiune executat. Programul este stocat în memorie ca o succesiune de instrucțiuni ce trebuie executate secvențial de către microprocesor. La executarea unei instrucțiuni, conținutul PC este automat marit cu o unitate pentru a indica următoarea instrucțiune de executat. Exista posibilitatea de prescriere a regiștrului PC, i.e. se introduce a altor valori decât celel obținute prin ordinea naturală (salturi în program necesare în decizii și bucle). Pentru inițierea unui program H încarcă PC cu adresa de început corespunzătoare. Prin comanda de RESET aplicată MICROPROCESORULUI, număratorul de programe este încarcat cu o adresa fixată de către producător, în general aceasta este 0000H. Comanda RESET ( inițializare) se da automat la punerea în funcțiune a microprocesorului după conectarea tensiunii sau se aplica din exterior.
2. **Regiștrul de adresare a memoriei**. Acest regiștru tampon de adresare denumit buffer de adresare e conectat la magistrala de adresare a memoriei, sau a posturilor de I/O. Conținutul regiștrului PC e transferat în bufferul de ieșire care va aplica pe magistrala exterioara de adresare un cuvânt binar de un bit ce reprezintă adresa unei locații de memorie sau adresa unui port de I/E. Dar încarcarea bufferului de adresare se poate face nu numai la PC, cât și de la alte elemente ale microprocesorului rezultă ca pe magistrala de adresare se pot aplica și avinute de adresa diferite de conținutul regiștrului PC. Unele instrucțiuni pot încarca regiștrele de ieșire cu o adresa rezultată din conținutul lui PC în care se adaugă sau se scade un număr rezultat în urma anumitor calcule (adesea generind multiple variante de adresare).



**3. Registrul de I/O (buffer I/O).** Prin acest buffer de I/O se realizeaza legatura dintre magistrala de date interioara a microprocesorului si magistrala de date exterioara a sistemului, deci vehiculeaza curentele de date si instructiuni.

**4. Registrul de instructiuni, RI.** Dupa ce un cuvânt instructiune e adus din memorie prin bufferul de I/O pe magistrala internă a microprocesorului. O copie a acestui cuvânt va fi înscrisă în registrul de instructiuni. Registrul RI pastreaza instructiunea pe durata executării acesteia. O data copiata instructiunea in RI continutul numaratorului de adrese este automat incrementat cu o unitate PC+1. Instructiunea este divizata in 2 câmpuri: câmpul codului operatiei, OPCODE si câmpul operandului (sau adresei operandului). Bitii din codul operatiei se aplica decodificatorului instructiunii care, apoi prin unitatea de control, va genera toate semnalele de control necesare executiei instructiunii reprezentate de codul operatiei. Câmpul adresa operandului se aplica bufferului de adrese pentru a forma adresa din memorie unde se afla operandul necesar operatiei.

**5. Registri de tip acumulator** Aceștia sunt regiștri, din structura  $\mu p$ , cu cea mai frecventa utilizare. În acestia se pătrează operanzi expresiilor aritmetice sau logice. Rezultatul operației efectuate de UAL se depune în unul din regiștri de intrare, aliterand continutul vechii al regiștrului. Microprocesorul permite efectuarea unor operații (cu un singur operand) folosind acești regiștri: stergerea acumulatorului (toti bitii pusi pe 0), inscrierea tuturor bitilor cu valoarea 1, deplasarea dreapta, stânga, complementarea continutului etc.

**6. Registrul indicatorilor de condiții.** Prin această denumire se intelege un grup de bistabile (flaguri, fanioane) asamblate sub forma unui regiștru si cite simultan vor genera împreuna cuvântul de stare al programului PSW (Program Status Words). Bitii cuvântului de stare sunt înscrisi la valoarea 1 în urma unor teste din timpul executiei operatorilor aritmetice si logice ale programului. Setul de instructiuni contine si instructiuni condiționale (instructiunea se execută numai dacă fanionul respectiv e setat). O instructiune condițională e utilizată pentru realizarea unei ramificații (salt) în program, adică se schimbă succesiunea de citire din ordinea naturală a instructiunilor de memorie, prin încarcarea PC cu o anumită adresă .

**Unitatea de control** - e partea care supervizeaza functionarea corecta a sistemului de calcul. Comenzile generate de unitatea de control se obțin în urma decodificării instructiunilor, a cererilor de intrerupere (prinime de la elementele microsistemului) si a impulsului de ceas.

Modalitatile de implementare a unitatii de control sunt:

- prin microprogramare. Instructiunile reprezintă microprograme, alcătuite din microinstructiuni (înscrise într-o memorie speciala) executate prin interpretare.
- Hardware – instructiunile sunt direct executate de hardware. Se spune ca unitatea de comanda este cablata.

**Caracteristici arhitecturale.**

Caracteristicile arhitecturale ca cele mai puternice influențe asupra eficienței de execuție a programelor scrise în limbaje de nivel înalt sunt:

- *tipurile de date* – deoarece ele pot fi aceleași su cele ale mașinii
- *modurile de adresare* – ele definesc mecanismul de access la date și pot fi utilizate eficient pentru reprezentarea structurilor complexe de date
- *setul de instructiuni* – reflectă operațiile tipice cerute de execuția programului

Alte caracteristici arhitecturale ale microprocesoarelor sunt:

- *organizarea regiștrilor*
- *aritmetica numerelor în virgulă mobilă*
- *intreruperi și capace*
- *mijloace de depanare.*

**Schema generală a unui microprocesor 8086 Modelul programatorului de asamblare**

Microprocesorul 8086 are două componente:

- Unitatea de execuție (UE);
- Unitatea de interfață cu bus-urile(UI)

Structura acestui microprocesor este de tip pipeline, adică este o structură în bandă de asamblare cu două segmente: UE și UI. UE decodifică instructiunile numerice, elaborează comenzi interne pentru efectuarea calculului și comenzi externe către cea de-a doua unitate. UE conține 8 locații de memorie internă numite registre de uz general.

UI calculează adresele de memorie și de intrare/ieșire, transferă datele între UE și memorie, între UE și I/O și transferă către UE codurile numerice ale instructiunilor citite în memorie.

Cele două segmente efectuează autonom secvențe de operații proprii, transferându-și în același timp informații. Secvențele de operații efectuate de cele două unități ale microprocesorului pentru a executa instructiunile se numesc *cicliuri de instructiune*, pentru UE și *cicliuri mașină de bus*, pentru UI.

**a) Structura UE**

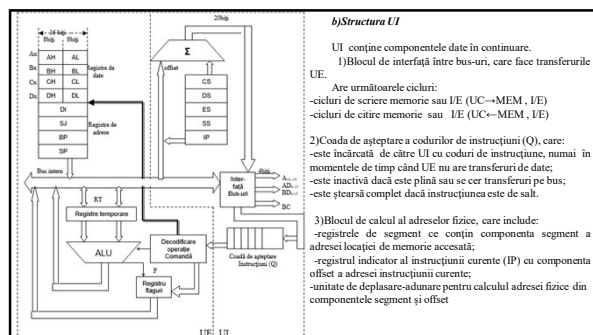
UE are componentele:

- Unitatea aritmetică-logică (UAL), în engleză Arithmetic-Logic Unity (ALU), care execută operații aritmetice și logice, deplasări și rotații.
- Registrele temporare (RT), care preiau operanzii de pe bus-ul intern și îi oferă unității UAL. Împună cu UAL formează un automat.
- Registrul de flaguri, F, care conține indicatorii de stare ALU ai ultimei operații; regiștrul este actualizat de către ALU.

Blocul de comandă, care decodifică codul instructiunii curente, preluată din coada de așteptare Q, execută operația și elaborează comenzi externe pentru UI.

UE este complet separată de exterior, toate sarcinile privind transferul cu exteriorul revenind UI-ului.






---

---

---

---

---

---

---

---

---

---

---

---

**Ciclul instrucțiune**

Execuția instrucțiunilor se realizează într-o secvență de pași numită **ciclul instrucțiune**.

Un ciclu instrucțiune este compus din mai multe cicluri masina. Ciclul esențial în funcționarea oricărui instrucțiuni este ciclul extrage-decodifică-execuția (fetch-decodeexecute).

Secvența de pași este următoarea:

1. Se transferă instrucțiunea următoare (indicată de PC) în registrul de instrucțiuni.
2. Se schimbă PC a.i. să conțină adresa următoarei instrucțiuni de executat.
3. Se determină timpul instrucțiunii extrase.
4. Dacă în instrucțiune se folosește conținutul unei locații de memorie, se găsește.
5. Se execută instrucțiunea
  - prin interpretare - microprogram - care este conținut în memorii rapide numai pentru citire numite memorii de control în care se stochează microinstrucțiunile) sau
  - direct de către hardware (cablat).
6. Se reia pasul 1.

Microprocesorul mai execută și alte cicluri masina:

- ciclul de citire din memorie
- ciclul de scriere în memorie
- ciclul de calcul.

---

---

---

---

---

---

---

---

---

---

---

---

**Registrele microprocesorului**

Registrele microprocesorului 8086 au o capacitate de 16 biți și pot fi clasificate în patru grupe, în funcție de rolul pe care îl au în execuția instrucțiunilor.

- Registre generale.
- Registre segment.
- Registru indicator al adresei instrucțiunii curente IP.
- Registru de flaguri, F, și registrul IP.

16 biți	8 biți	8 biți
Ax	AH	AL
Bx	BH	BL
Cx	CH	CL
Dx	DH	DL

**Registre generale**

Registrele generale se împart în două seturi de registre:

- Registre de date: Ax, Bx, Cx, Dx.
- Registre de adresare: SP, BP, SI, DI.

**a) Registre de date**

Există patru registre de date de 16 biți:

- Ax-accumulator;
- Bx-baza în adresarea datelor;
- Cx-conitor;
- Dx-date.

Fiecare din aceste registre pot fi împărțite în două registre de 8 biți:

---

---

---

---

---

---

---

---

---

---

---

---

Registrele de date sunt utilizate în majoritatea instrucțiunilor aritmetice și logice. Majoritatea instrucțiunilor aritmetice utilizează în același mod toate registrele. Există și instrucțiuni aritmetice pentru care anumite registre generale au întrebunișiri speciale. De exemplu:

- Ax - operații de intrare/ieșire pe 16 biți, înmulțiri și împărțiri pe 16 biți;
- AL - operații de intrare/ieșire pe 8 biți, translații, aritmetice BCD, înmulțiri și împărțiri pe 8 biți;
- AH - înmulțiri și împărțiri pe 8 biți;
- Bx - operații cu memoria, adrese indirectă, translații;
- Cx - operații cu șiruri, bucle program;
- CL - operații de deplasare sau rotație cu mai mult de o poziție;
- Dx - operații de intrare/ieșire, adresare indirectă, înmulțiri și împărțiri pe 16 biți.

**b) Registre de adresare**

Sunt de două tipuri:

- 1) Registre indicatoare de adresă în stivă (pointer):
  - SP (Stack Pointer), care conține adresa curentă a vârfului stivei;
  - BP (Base Pointer), care conține adresa bază pentru adresarea indirectă a stivei.
- 2) Registre indicatoare de adresă pentru șiruri (index):
  - DI (Destination Index) - conține adresa curentă pentru șirul destinației;
  - SI (Source Index) - conține adresa curentă pentru șirul sursei.

Registrele de adresare pot fi utilizate și pentru anumite instrucțiuni aritmetico-logice.

Registrele pointer conțin componente offset ale adreselor de stivă (adresele relative în segmentul de stivă curent).

Registrul BP poate fi utilizat și pentru adresarea în cadrul altor segmente.

Registrele index conțin componente offset ale adreselor variabilelor (adrese relative în segmentul de date curent). Ele sunt utilizate ca registre de adresare și instrucțiunile de transfer sau prelucrare de șiruri de caractere. În acest ultim caz registrul SI conține adresa relativă curentă a șirului destinație în cadrul segmentului de date curent (DS), iar DI conține adresa relativă curentă a șirului sursă în cadrul segmentului de date suplimentar (ES).

---

---

---

---

---

---

---

---

---

---

**Registrele segment**

Spațiul de memorie pe care poate să-l adreseze un microprocesor 8086 este împărțit în segmente logice de lungime 64Kb. Există patru registre segment:

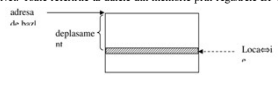
- CS (Cod Segment), care conține componenta segment a adreselor codului (instrucțiunile programului);
- DS (Data Segment), care conține componenta segment a adreselor variabilelor (segment date curent);
- ES (Extra Segment), care conține componenta segment a adreselor variabilelor (segment suplimentar);
- SS (Stack Segment), care conține componenta segment a adreselor datelor din segmentul stivă.

Instrucțiunea care urmează să fie executată se găsește în segmentul a cărui adresă se află în registrul CS, la adresa relativă conținută în IP.

Conținutul registrului DS definește segmentul de date curent. Toate referirile la datele din memorie, cu excepția celor prin registrele BP și SP sau registrul DI în instrucțiunile pentru șiruri, utilizează în mod implicit segmentul referit de DS.

Conținutul registrului ES definește segmentul de date suplimentar. Referirile la date în instrucțiunile pentru șiruri utilizează în mod implicit segmentul referit de ES.

Conținutul registrului SS definește segmentul curent al stivei. Toate referirile la datele din memorie prin registrele BP și SP utilizează în mod implicit segmentul referit de registrul SS.




---

---

---

---

---

---

---

---

---

---

**Registrele IP și F**

Registrul indicator al adresei curente, IP (Instrucțiun pointer), este un registru de 16 biți care conține componenta offset a adresei instrucțiunii în segmentul de cod curent. Programele nu au acces direct la IP, dar există instrucțiuni care îl modifică și îl încarcă sau îl descarcă prin stivă. Registrul de flaguri F cuprinde biții indicatori de stare și control, numiți și flaguri. Aceste flaguri sunt utilizate pentru a memora informații referitoare la rezultatul unor operații aritmetice și logice (OF, SF, ZF, AF, PF, CF) și pentru memorarea unor informații de control al microprocesorului (TF, DF, IF). Semnificațiile acestor flaguri sunt date în continuare.

- CF (Carry Flag) reflectă transportul în exterior al bitului cel mai semnificativ al rezultatului operațiilor aritmetice. CF=1 înseamnă un transport la adunare. CF mai este modificat la operațiile de rotație și deplasare.
- PF (Parity Flag) este indicator de paritate, el fiind 1 dacă rezultatul are paritate pară; de asemenea acest indicator este utilizat și de instrucțiunile de aritmetică zecimală.
- AF (Auxiliary Carry Flag) este indicator de transport auxiliar și este 1 dacă a fost transport de la jumătatea de octet inferioară la jumătatea de octet superioară; este utilizat la instrucțiunile de aritmetică zecimală. -ZF (Zero Flag) este indicatorul de zero, având valoarea 1 dacă rezultatul operației a fost zero;
- SF (Sign Flag), indicatorul de semn, este 1 dacă cel mai semnificativ bit al rezultatului este 1, adică, în reprezentarea numerelor în complement față de 2, rezultatul este negativ.
- OF (Overflow Flag), indicatorul de depășire aritmetică, a gamei de valori posibile de reprezentat, este 1 dacă dimensiunea rezultatului depășește capacitatea locașei memoriei.
- IF (Interrupt Flag), indicatorul de validare a întreruperilor, este 1 dacă se pot valida întreruperile externe mascabile și 0 dacă întreruperile externe sunt invalidate. De remarcat că acest flag nu afectează întreruperile interne sau pe cele externe nemascabile.
- DF (Direction Flag) este utilizat de instrucțiunile pe șiruri de caractere și specifică direcția de parcurgere a lor:
  - 0 - de la adrese mici spre adrese mari;
  - 1 - de la adrese mari spre adrese mici.
- TF (Trace Flag) este utilizat pentru controlul execuției instrucțiunilor în regim pas cu pas în scopul depănării programelor. Dacă este 1, după execuția fiecărei instrucțiuni se va genera un semnal de întrerupere intern.

---

---

---

---

---

---

---

---

---

---

**Adresarea memoriei. Segmentarea memoriei.**

Intel 8086 poate adresa un spațiu de memorie din memoria principală (MP) de 1 Moctet. Conform convenției INTEL, datele formate din mai mulți octeți sunt memorate cu octetul cel mai semnificativ la locația de adresă cea mai mare, adică octetul cel mai puțin semnificativ este memorat la adresa cea mai mică.

Intel 8086 vede memoria principală organizată ca un grup de segmente. Un segment este un bloc de memorie de dimensiune 64 Kocteți. Fiecare segment poate fi accesat, în scriere sau în citire, în mod independent.

Procesorul poate lucra în două moduri: modul *real* și modul *protejat*.

Procesorul real adresează memoria principală printr-o adresă fizică directă.

În modul protejat, procesorul adresează memoria principală ca pe o memorie virtuală. O adresă virtuală este, de fapt, un număr pentru o locație de memorie pe care procesorul o translatează într-o adresă fizică corespunzătoare. O adresă virtuală are două componente: o *adresă de bază* (segment) și un *deplasament* (offset). Notația consacrată pentru adresa logică este:

Segment : offset

Trecerea de la adresa logică la adresa fizică se face astfel:

Adresa logică = adresă fizică

Adresă fizică = segment x 10<sub>16</sub> + offset

Înmulțirea cu zece a unui număr în ... înseamnă deplasarea spre stânga cu o poziție. Exemple:

1) adresa logică ABC4 : EF88

    adresa fizică = ABC40 + EF88 = BACF8

2) adresa logică AB00 : CD00

    adresa fizică = AB000 + CD00 = B7D00

Adresă fizică este calculată în UI în funcție de modul de adresare.

Structura pe segmente a memoriei face posibilă scrierea unor programe care sunt independente de poziția lor în memorie, adică sunt *relocabile dinamic*. Pentru ca un program să fie relocabil trebuie să fie scris astfel încât să nu altereze registrele sale segment și să nu facă transferuri directe de la o locație în afara segmentului de cod. Aceasta permite programului să fie mutat oriunde în memoria disponibilă, atâta timp cât registrele segment sunt actualizate cu noua adresă de bază.

---

---

---

---

---

---

---

---

---

---

---

**Formatul instrucțiunii**

Instrucțiunile, codificate în binar, pot ocupa în memorie de la 1 la 6 octeți. Codul instrucțiunii este format din:

- codul operației care ocupă 1 sau 2 octeți și care specifică:
  - tipul operației;
  - tipul operandilor (8 sau 16 biți);
  - sursa operandilor (internă sau externă);
  - destinația rezultatelor operațiilor ALU;
  - modul de calcul al EA;
- operandii de tip imediat: date și adrese.

**Formatul instrucțiunii cu cod pe un octet.**

Semnificațiile câmpurilor din figura sunt date în continuare.

**D** – destinație. Specifică direcția rezultatului în combinație cu câmpurile MOD și R/M.

**W** – word. Specifică tipul de transfer. W = 0 înseamnă transfer pe un octet iar W = 1, transfer pe 2 octeți (1 cuvânt).

**MOD** este o codificare a modului de calcul al adresei efective sau lungimea deplasamentului. Este utilizat pentru a determina adresa efectivă împreună cu câmpul r/m.

MOD = 11, câmpul r/m este un câmp de registru cu următoarele semnificații:

MOD = 00, câmpul deplasament nu este prezent

MOD = 01, câmpul deplasament are 8 biți (low)

MOD = 10, câmpul deplasament are 16 biți.

r/m	registru	
	[w=1] cuvânt	[w=0] octet
000	Ax	AL
001	Cx	CL
010	Dx	DL
011	Bx	BL
100	SP	AXI
101	BP	CH
110	SI	DH
111	DI	BH

---

---

---

---

---

---

---

---

---

---

---

R/M conține:

- adresa unui registru (pentru MOD=11)
- o codificare utilizată pentru calculul adresei efective.

R/M	Adresa efectivă
000	BX + SI + deplasament
001	BX + DI + deplasament
010	BP + SI + deplasament
011	BP + DI + deplasament
100	SI + deplasament
101	DI + deplasament
110	BP + deplasament
111	BP + deplasament

Unitatea de execuție (UE) are acces la operandi imediați și de registre; când este nevoie de un operand de memorie, se transmite la UI deplasamentul acestuia și registrul de segment IC determină adresa fizică a operandului în funcție de modul de adresare.

---

---

---

---

---

---

---

---

---

---

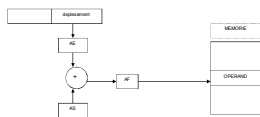
---

**Moduri de adresare**

Există, în principal, cinci moduri de adresare.

**Adresare directă**

Adresa efectivă (AE) a operandului este reprezentată de deplasamentul conținut în instrucțiune.



Adresare directă. AE, adresă efectivă. AS, adresă segment. AF, adresă fizică.

---

---

---

---

---

---

---

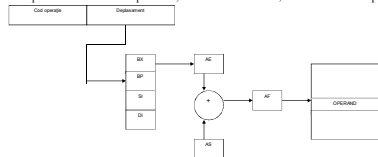
---

---

---

**Adresare indirectă prin registre**

În câmpul calculat nu se află operandul, ci o altă adresă de operand.



Adresare indirectă prin registre.

Sintaxa instrucțiunilor în limbajul de asamblare utilizează pentru adresarea indirectă operatorul [ ]. De exemplu: mov ax, [bx], se deplasează la adresa conținută de bx.

---

---

---

---

---

---

---

---

---

---

**Adresare indexată**

La calculul adresei participă și un registru index (SI sau DI în cazul procesoarelor 8086).

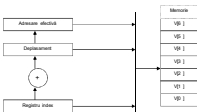


**Adresare indexată**

Adresa se obține din suma registrului index și deplasamentul din instrucțiune.

Acest mod de indexare este utilizat, în cele mai multe cazuri, pentru referirea elementelor unui vector. Deplasamentul marchează începutul vectorului iar registrul index selectează elementul prin poziția sa relativă în cadrul vectorului. Deoarece toate elementele vectorului sunt de aceeași lungime, prin operații aritmetice elementare asupra registrului index se va selecta orice element. De aceea se poate specifica un factor de scală (1,2,3,4) pentru index, pentru a referi vectori cu componente de lungime fixă de 1,2,3,4 octeți.

Referirea vectorilor de lungime fixă în adresarea indexată.




---

---

---

---

---

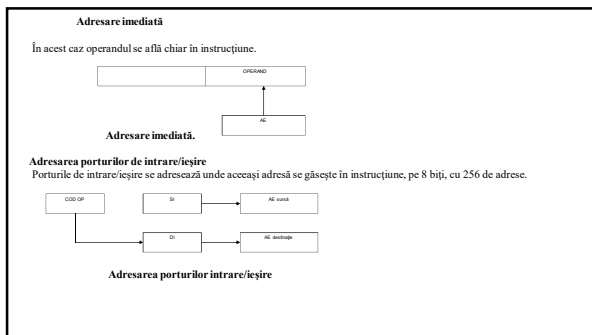
---

---

---

---

---



---

---

---

---

---

---

---

---