

1. ARHITECTURA MICROPROCESOARELOR

1.1. Microprocesorul. Noțiuni generale

Progresele tehnologice și electronice, înregistrate în ultimele decenii, au determinat apariția microprocesoarelor care a reprezentat un moment important, atât în ceea ce privește domeniile de aplicație a acestor componente, cât și în ceea ce privește concepția și realizarea sistemelor cu microprocesor.

Începând cu microprocesoare de 4 biți, de viteză relativ redusă, cu costuri ridicate și dispunând de puține elemente de dezvoltare a aplicațiilor, piața microprocesoarelor a înregistrat progrese uimitoare, în urmă cu câțiva ani, evoluând într-o progresie exponențială.

Obținerea "calculatoarelor pe un cip", cu performanțe de-a dreptul spectaculoase, s-a făcut extrem de rapid, iar estimările pentru următorii ani prevăd o expansiune a sistemelor microprocesor.

Din punctul de vedere al caracteristicilor putere de calcul / cost / complexitate, microprocesoarele și microcalculatoarele (calculatoarele care utilizează ca unitate centrală un microprocesor), se situează pe poziții avantajate, comparativ cu celelalte tipuri "clasice", de calculatoare.

Figura 1 prezintă unele dintre aceste caracteristici pentru diversele tipuri de echipamente de calcul.

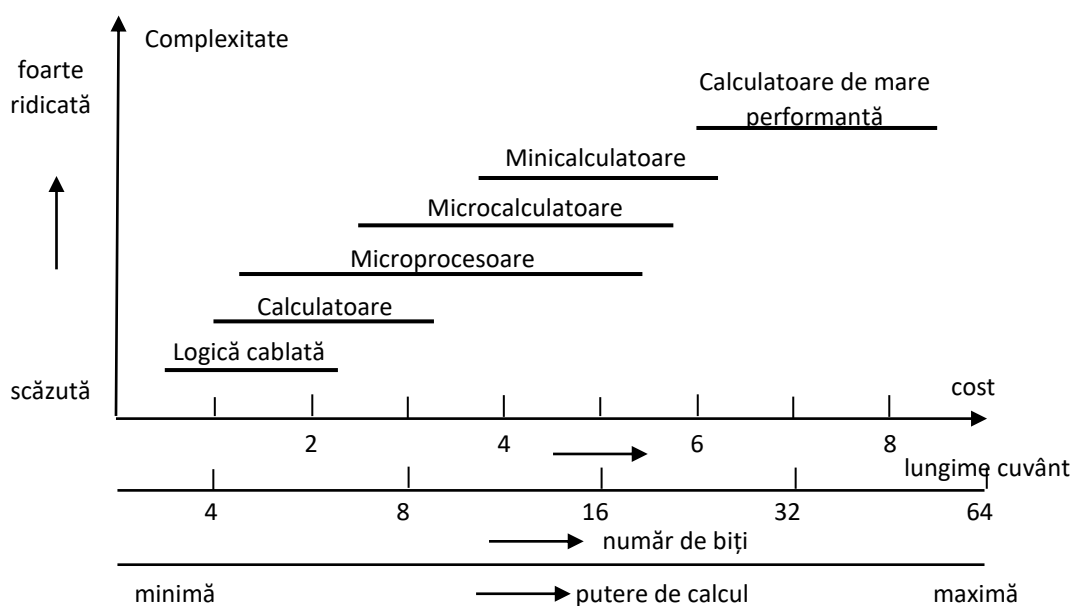


Fig.1. Compararea costului, puterii de calcul și capacității diverselor calculatoare numerice

Termenul de "*microprocesor*" a fost introdus în 1972 de către firma americană INTEL, realizatoarea primului microprocesor de 4 biți, în anul 1971. Au urmat apoi microprocesoarele de 8, 16 și 32 de biți (în 1972, 1974 și 1981, respectiv). Numărul de componente pe cip a crescut de peste 500 de ori iar frecvența de lucru a circuitelor de peste 100 de ori.

Utilizarea microprocesoarelor a fost orientată inițial cu predilecție ca *microcontrolere* în structuri dedicate, pentru aplicații de control al proceselor. Setul de instrucțiuni al acestora era relativ limitat, iar programarea se făcea direct în cod mașină. Dezvoltarea ulterioară a arhitecturilor evoluate, apariția limbajelor de programare de nivel înalt, deosebit de performante, creșterea puterii de calcul și a flexibilității, apariția și expansiunea sistemelor de dezvoltare, au lărgit considerabil aria de aplicații ale microprocesoarelor.

Unul dintre cele mai importante criterii, pentru evaluarea și selecția microprocesoarelor, este mărimea *cuvântului de date* al acestora. Cuvinte de date mai mari implică evident o putere de calcul și adresabilitate sporite. În prezent, microprocesoarele de 16 și 32 biți domină piața de microcalculatoare, deși o gamă relativ importantă de aplicații, în domeniul controlului proceselor industriale, este implementată cu microprocesoare de putere mai mică (uzual, microprocesoare de 8 biți), suficiente ca putere de calcul, capacitate de adresare și competitive ca preț de cost.

Progrese importante s-au făcut și în domeniul "*microprogramarii*". Procesoarele microprogramate, (spre deosebire de microprocesoarele "clasice", cu structura fixă și set de instrucțiuni impus de fabricant), permit (deși la o viteză relativ mai redusă), obținerea unor structuri mult mai flexibile, în ceea ce privește realizarea și modificarea setului de instrucțiuni, de către utilizator, adaptate aplicației abordate.

În decursul anilor, a crescut sensibil și *capacitatea de adresare* a microprocesoarelor. Numărul de moduri auxiliare de adresare a crescut. Multe firme oferă circuite specializate pentru controlul și coordonarea funcționării memoriei sistemului.

Microprocesoarele diferă sensibil în ceea ce privește *capacitățile de memorare și manipulare* a diverselor tipuri de date. Dacă reprezentarea și manipularea informațiilor sub forma octeților și a cuvintelor de date este uzuală la toate tipurile de microprocesoare, alte tipuri de operații, ca de exemplu, informații reprezentate pe biți, *BCD* (Binary Coded Decimal = zecimal codat binar), reale (virgula mobilă), texte (șiruri de caractere), etc., nu sunt întotdeauna direct acceptate. De exemplu, date manipulate ca biți individuali nu sunt acceptate de către microprocesorul 8086, deși sunt foarte utilizate în aplicațiile de proces. Cu cât acceptă o gamă mai extinsă de tipuri de date, microprocesorul devine mai potrivit unei game tot mai largi de posibile aplicații.

Revoluția microprocesoarelor este generată de tendința de a se implementa toate componentele unui calculator cu ajutorul unui număr relativ redus de cipuri. Orice calculator, indiferent de mărime, încorporează trei componente de baza:

- **unitatea centrala de prelucrare (CPU - Central Processing Unit = unitate centrală de prelucrare)**, conținând elementele de calcule aritmetice, logice și funcțiile de control;
- **memoria (RAM - pentru citire/scriere, și ROM - doar citire)**
- **interfețele de intrare/ieșire** pentru controlul periferiei calculatorului.

Odată cu progresele tehnologice înregistrate, tot mai multe componente pot fi integrate pe același cip, și astfel tot mai multe funcții auxiliare se pot implementa pe cipul microprocesorului însuși. Se obțin astfel calculatoare construite cu foarte puține cipuri.

Un microcalculator combină un microprocesor cu memorie și capacități de intrare/ieșire pe unul sau mai multe cipuri.

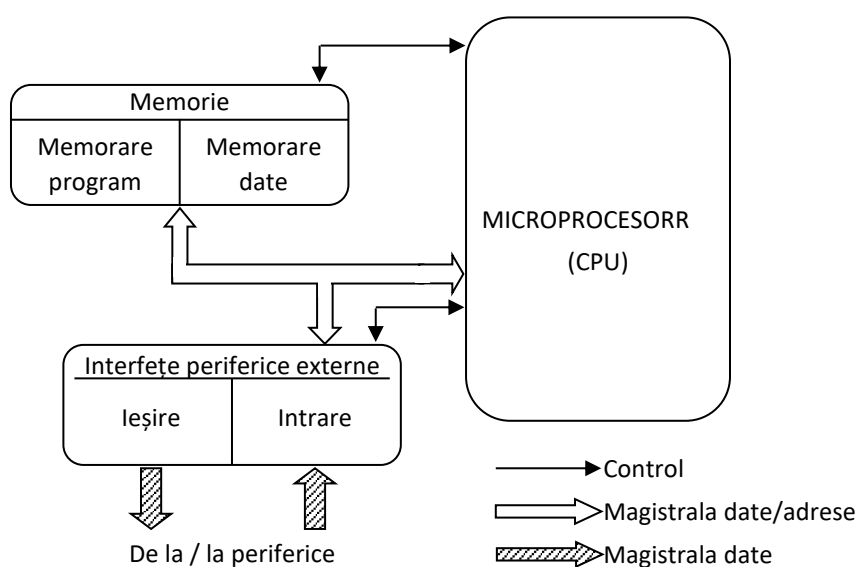


Fig. 2. Structura de bază a unui sistem microcalculator

Funcțional, un sistem microprocesor (ca de altfel orice calculator numeric), prezintă cinci funcții de bază:

1. **Funcția de intrare (INPUT)**, care permite legătura între lumea exterioară și sistem;
2. **Funcția de ieșire (OUTPUT)**, care permite legătura între sistem și lumea exterioară;
3. **Funcția de memorare (MEMORY)**, care permite păstrarea informațiilor (date, rezultate), și (uneori) a instrucțiunilor programului;
4. **Funcția aritmetico-logică**, (implementată prin *ALU* -Arithmetic Logic Unit = unitate aritmetico - logică), care permite efectuarea operațiilor aritmetice și logice în sistem;
5. **Funcția de control**, care cuprinde totalitatea acțiunilor de coordonare și control ale activității sistemului.

1.2. MICROPROCESORUL - PREZENTARE HARDWARE

Structura funcțională a microprocesorului va fi prezentată din perspectiva utilizatorului orientat spre hardware-ul sistemului.

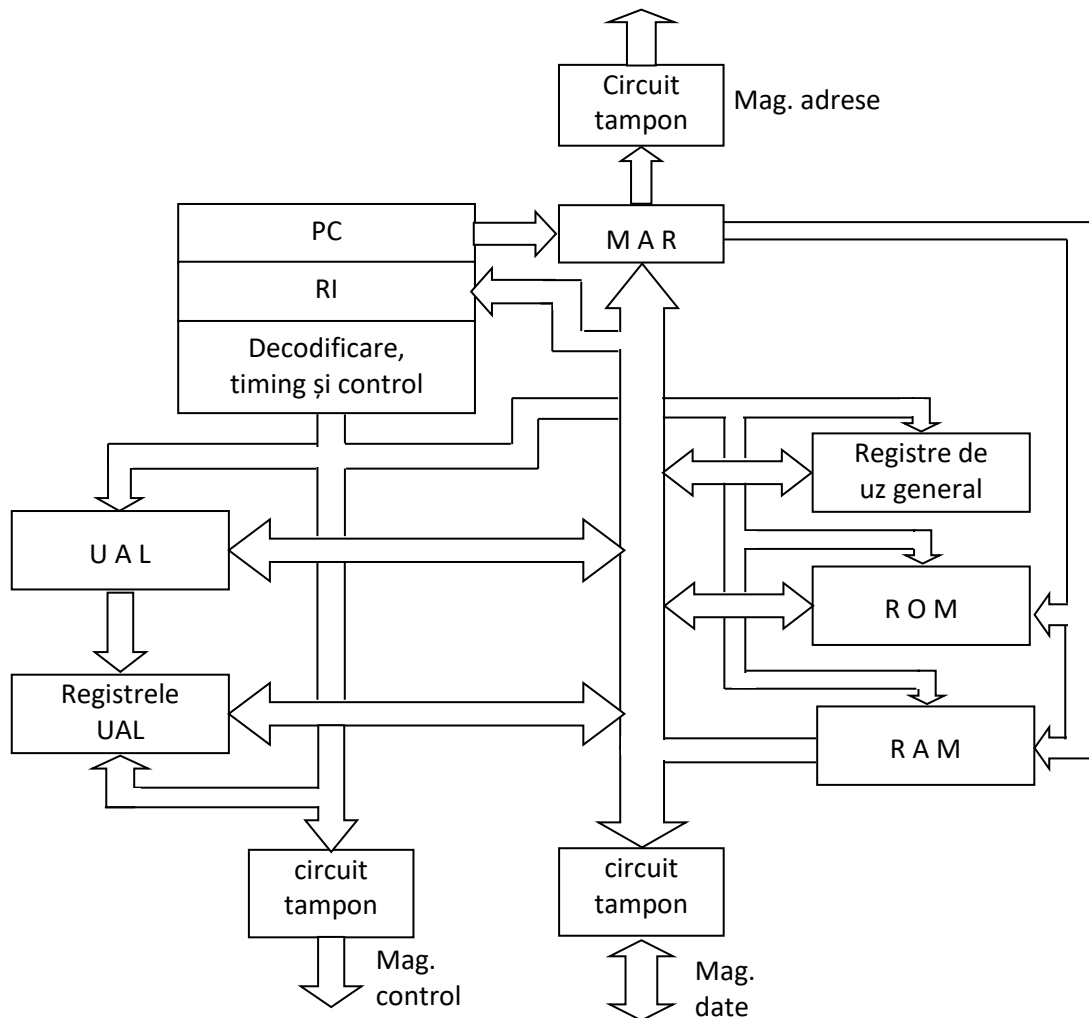


Fig. 3. Structura internă a unui microprocesor

Notății:

PC – program counter - registru contor de program

MAR - registrul de adresă a memoriei

UAL - unitate aritmetico – logică

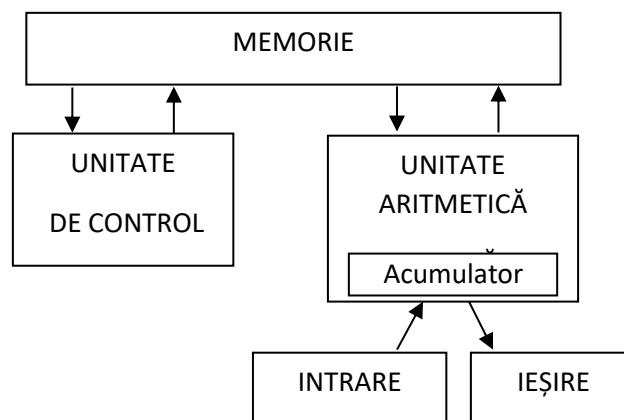
1.3. CARACTERISTICI ARHITECTURALE ALE UNITĂȚII CENTRALE

Arhitectura unității centrale de calcul (CPU - Central Processing Unit) este unul din elementele cele mai importante care trebuie avut în vedere în analiza oricărui sistem de calcul.

Principalele concepte luate în considerare și întâlnite aici sunt următoarele:

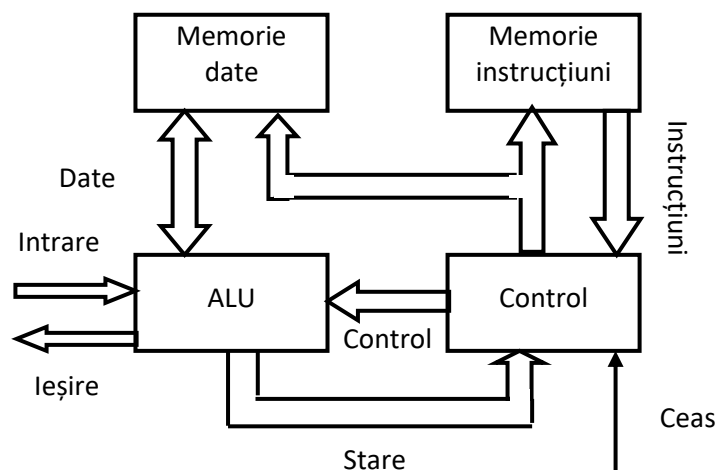
a. Arhitecturi de tip " von Neumann "

Cele mai multe microcontrolere sunt realizate pe baza acestei arhitecturi de sistem. Microcontrolerele bazate pe această arhitectură au o unitate centrală (CPU) caracterizată de existența unui singur spațiu de memorie utilizat pentru memorarea atât a codului instrucțiunilor cât și a datelor ce fac obiectul prelucrării. Există deci o singură magistrală internă (bus) care este folosită pentru preluarea instrucțiunilor și a datelor. Efectuarea celor două operații separate, în mod secvențial, are ca efect, încetinirea operațiilor. Este arhitectura standard (cea mai des întâlnită) și pentru microprocesoarele de uz general.



b. Arhitecturi de tip " Harvard "

La această arhitectură există spații de memorie separate pentru program și date. În consecință ar trebui să existe și magistrale separate (de adrese și date) pentru codul instrucțiunilor și respectiv pentru date. Principial există astfel posibilitatea execuției cvasiparalele (suprapunerii) a celor două operații menționate anterior. Codul unei instrucțiuni poate fi preluat din memorie în timp ce se execută operațiile cu datele aferente instrucțiunii anterioare. Este posibilă o execuție mai rapidă, pe seama unei complexități sporite a microcircuitului. Este arhitectura standard pentru procesoarele numerice de semnal (DSP - Digital Signal Processing). Datorită costului mare al implementării unei astfel de arhitecturi, în cazul microcontrolerelor se întâlnește mai ales o arhitectură Harvard modificată, cu spații de memorie separate pentru program și date, dar cu magistrale comune pentru adrese și date.



c. CISC

Aproape toate microcontrolerele au la baza realizării CPU conceptul CISC (Complex Instruction Set Computer). Aceasta înseamnă un set uzual de peste 80 instrucțiuni, multe din ele foarte puternice și specializate. De obicei multe din aceste instrucțiuni sunt foarte diferite între ele: unele operează numai cu anumite spații de adrese sau registre, altele permit numai anumite moduri de adresare, etc. Pentru programatorul în limbaj de asamblare există unele avantaje prin utilizarea unei singure instrucțiuni complexe în locul mai multor instrucțiuni simple (analog macroinstrucțiunilor clasice dintr-un limbaj de asamblare).

d. RISC

RISC (Reduced Instruction Set Computer) este un concept de realizare a CPU care a început să fie utilizat cu succes de ceva timp și la realizarea microcontrolerelor. Prin implementarea unui set redus de instrucțiuni care se pot executa foarte rapid și eficient, se obține o reducere a complexității microcircuitului, suprafața disponibilizată putând fi utilizată în alte scopuri. Printre caracteristicile asociate de obicei unui CPU RISC se pot menționa:

- arhitectură Harvard modificată sau von Neumann;

- viteză sporită de execuție prin implementarea unui "pipeline" pentru instrucțiuni. „Pipeline” este o tehnică de creștere a vitezei de execuție totale a procesoarelor, fără a ridica tactul. Ea constă în subdivizarea fiecărei instrucțiuni într-un număr de etape sau segmente, fiecare etapă fiind executată de câte o unitate funcțională separată a procesorului (segment pipeline). Segmentele "pipeline" sunt conectate între ele într-un mod analog asamblării unei conducte din segmente de țevă.

- set de instrucțiuni ortogonal (simetric): orice instrucțiune operează cu orice spațiu de adrese (de memorie) sau orice registru, instrucțiunile nu prezintă combinații speciale, excepții, restricții sau efecte colaterale.

1.4. MICROPROCESORUL INTEL 8086

Ca urmare a succesului de piață repurtat cu primul microprocesor pe 8 biți - 8080 (magistrala de adrese de 16 biți, magistrala de date de 8 biți) și a dezvoltării tehnologiei H-MOS, firma *INTEL* realizează microprocesorul 8086 (magistrală multiplexată - 20 linii de adresă, 16 linii de date) cu care se implantează solid și în domeniul microprocesoarelor de 16 biți.

Patru concepte arhitecturale noi, la momentul respectiv, au stat la baza proiectării și realizării familiei lui 8086:

1. **Segmentarea memoriei**, prin care se permite programarea modulară, deschizându-se astfel calea implementării tehnicilor de protecție și partajare a memoriei.
2. Implementarea cu ușurință a **modurilor de adresare** caracteristice limbajelor de programare de nivel înalt.
3. Creșterea puterii de calcul pe baza unui **set de registre optimizat** pentru a realiza funcții generale și specializate de procesare a datelor, precum și introducerea unităților aritmetice de calcul de tipul coprocesoarelor aritmetice. Aceste echipamente care se conectează pe magistralele microprocesorului în paralel cu acesta, cresc mult viteza de lucru a sistemului în unele aplicații.
4. Un **set de instrucțiuni** codificat anume pentru utilizarea la viteză crescută și cu eficiență maximă a memoriei.

Microprocesorul 8086 extinde performanțele unui microprocesor la niveluri neatinse de predecesorul său pe 8 biți:

- operații aritmetice pe 16 biți cu numere cu și fără semn (inclusiv înmulțiri și împărțiri);
- operații complexe de manipulare a șirurilor de caractere și la nivel de bit;
- adresarea directă a unui megabyte de memorie;
- posibilitatea configurării unor structuri foarte variate din punct de vedere al complexității sistemului;
- simplitatea implementării unor sisteme complexe multi-microprocesor.

Realizarea acestor performante cu un circuit în capsulă cu doar 40 de pini (ca și 8080) se îndeplinește pe două căi. Acestea se referă la două caracteristici hardware esențiale ce definesc funcționarea microprocesorului *I 8086*, și anume:

1. Multiplexarea în timp a magistralei de adrese și date;

2. **Microprocesorul are o configurație internă comutabilă** pentru adaptarea la nivelul de complexitate a sistemului în care este utilizat. Astfel, în sistemele simple, 8086 își generează și controlează singur semnalele de pe magistrala de control; în sistemele complexe, magistrala de control este generată de un circuit specializat din familia lui 8086, circuit denumit "controler de magistrală" (*8288 Bus Controller*), opt dintre conexiunile fizice ale microprocesorului fiind comutate pentru a îndeplini funcțiunile de coordonare necesare. Un singur pin al lui 8086, pinul 33 (*MN / MX*), conectat la masă sau la alimentare, comandă comutarea configurației interne a microprocesorului pentru adaptarea la sistemul extern.

1.4.1. ARHITECTURA MICROPROCESORULUI INTEL 8086

După cum s-a prezentat în capitolul precedent, execuția unui program într-un sistem cu microprocesor are ca efect repetarea ciclică a pașilor de mai jos:

- Extragerea următoarei instrucțiuni din memorie.
- Citirea unui operand (dacă instrucțiunea o cere).
- Execuția instrucțiunii.
- Scrierea rezultatului (dacă instrucțiunea o cere).

Datorită arhitecturii specifice, execuția acestor pași are loc în două unități separate de procesare a datelor din cadrul CPU:

- EU (*execution unit*) — unitatea de execuție
- BIU (*bus interface unit*) — unitatea de interfață cu magistrala.

În EU are loc execuția instrucțiunilor, iar în BIU se extrag instrucțiunile, se citesc operanzii și se scriu rezultatele în memorie. Cele două unități pot opera independent una de cealaltă și pot asigura, în majoritatea cazurilor, suprapunerea în timp a etapei de extragere a unei instrucțiuni cu etapa de execuție a unei instrucțiuni extrase anterior din memorie. În acest mod, practic "dispare" timpul necesar extragerii instrucțiunilor din memorie, crescând viteza de lucru a microprocesorului deoarece EU execută instrucțiuni al căror cod a fost deja adus de către BIU din memorie în microprocesor.

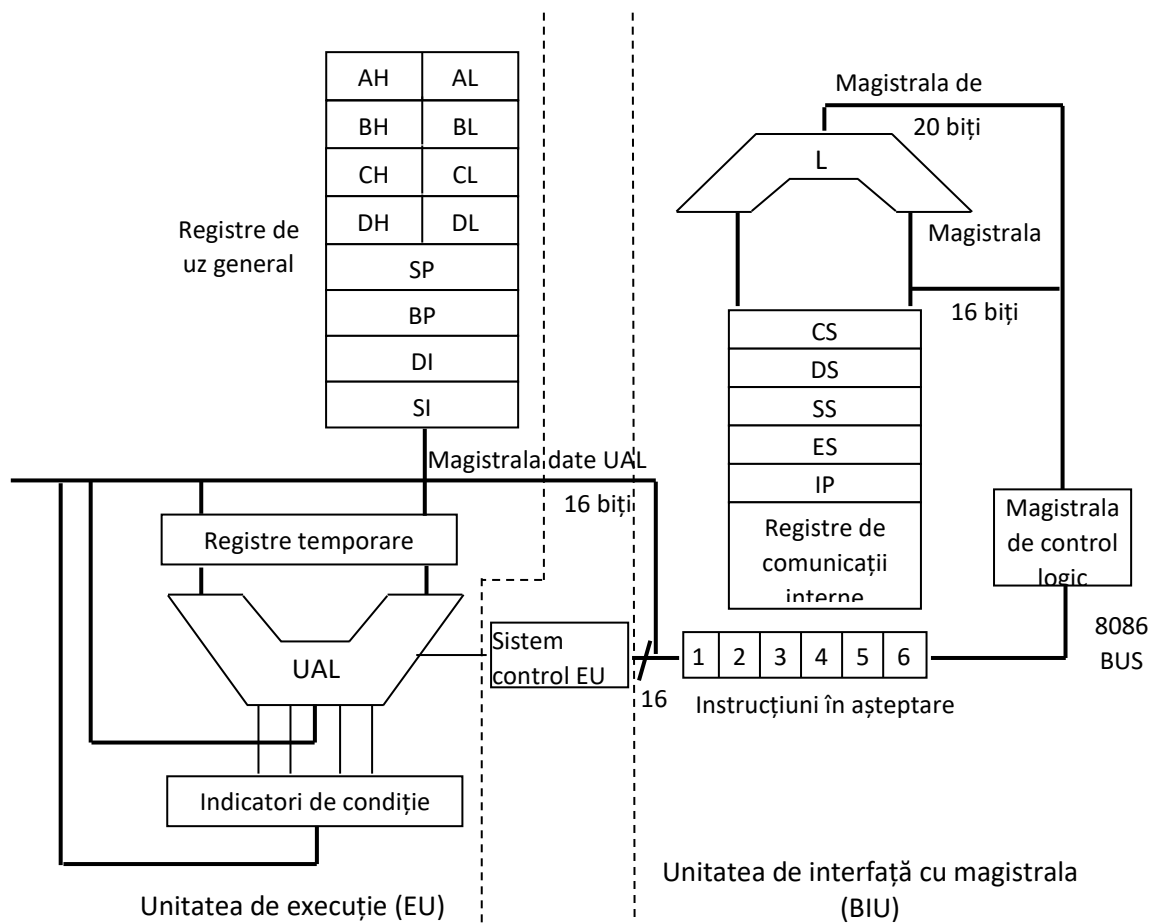


Fig. 4. Microprocesorul 8086 – schema bloc

EU — Unitatea de execuție

Unitatea de execuție conține registrele de uz general, unitatea aritmetico-logică, registrul indicatorilor de condiție, un bloc logic de control și o magistrală internă de date de 16 biți (Figura 4.1).

Funcțiile EU acoperă execuția tuturor instrucțiunilor, furnizarea datelor și a adreselor către BIU, controlul registrelor de uz general și al indicatorilor de condiție. Cu excepția câtorva pini de control, unitatea de execuție este complet izolată de "lumea exterioară". Așa cum se observă în figura 4.1, unitatea EU preia instrucțiunea următoare de executat dintr-o serie de așteptare alimentată continuu de unitatea de interfață cu magistrala.

BIU — Unitatea de interfață cu magistrala

Unitatea de interfață cu magistrala execută toate operațiile externe de magistrală ce sunt necesare pe parcursul extragerii și execuției unei instrucțiuni. Ea se compune din registre de segment, un registru de tip contor de program denumit **pointer de instrucțiuni** (*instruction pointer*), registre de comunicație internă, o schemă logică pentru generarea adresei pe cele 20 de linii de adresă ale microprocesorului 8086 și pentru controlul magistralei multiplexate precum și o **serie de instrucțiuni în așteptare**. Această funcție este realizată cu o memorie RAM de 6 octeți și conține instrucțiuni care sunt extrase în avans de BIU și urmează să fie preluate de EU pentru decodificare și execuție.

Cele două unități de procesare ale CPU operează independent una de alta, în sensul că, ori de câte ori doi sau mai mulți octeți din seria de așteptare sunt liberi, iar EU nu solicită BIU la efectuarea vreunui ciclu de magistrală, unitatea de interfață cu magistrala execută în avans cicluri de extragere de instrucțiuni pentru a realimenta locațiile libere din seria de instrucțiuni în așteptare. Instrucțiunile extrase în avans de BIU sunt cele care urmează în mod logic într-o procesare serială a programului, ele aflându-se în memorie în locații adiacente și la adrese superioare adresei instrucțiunii care se execută la un moment dat.

Registrele de uz general

Microprocesorul 8086 are 8 registre generale de 16 biți grupate în două seturi de câte 4 registre fiecare:

1. **registrele de date** (uneori denumite grupul registrelor H&L de la *high* și *low*)
2. **registrele pointer și index** (denumite și grupul P&I).

Un registru aparținând grupului H&L se caracterizează prin faptul că poate fi adresat ca registru de 16 biți, dar se compune din două entități de 8 biți, partea *high* și partea *low*, care pot fi adresate la rândul lor separat, ca registre de 8 biți. Registrele pointer și index nu pot fi adresate decât ca registre de 16 biți.

Atât registrele de date cât și registrele pointer și index pot fi folosite în majoritatea operațiilor aritmetice și logice, oricare dintre ele putând juca rolul registrului "acumulator" existent la generațiile precedente de microprocesoare. Pentru a permite utilizarea unui set compact, dar puternic de instrucțiuni, anumite registre sunt folosite în mod implicit de unele instrucțiuni, așa cum arată tabelul 4.1.

Aceste registre, în forme evoluat se întâlnesc și la microprocesoarele care îi urmează lui 80286 în serie (80386, 80486, Pentium). Pot fi însă pe 32 de biți, caz în care sunt denumite EAX, EBX, ECX și EDX sau pe 64 de biți.

REGISTRU	OPERAȚII
AX	Înmulțiri, împărțiri și I/E pe cuvânt
AL	Înmulțiri, împărțiri și I/E pe octet, translatări, aritmetică zecimală
AH	Înmulțiri, împărțiri pe octet
BX	Translatări
CX	Operații cu șiruri de caractere, contor pentru operații repetate
CL	Deplasări și rotații cu mai mult de o poziție
DX	Înmulțiri și împărțiri pe cuvânt, I/E cu adresare indirectă
SP	Operații cu stiva
SI, DI	Operații cu șiruri de caractere

Registrele de segment

Spațiul fizic de memorie de 1 Mb direct adresabil de microprocesorul 8086 este divizat în segmente logice de până la 64 kb fiecare. CPU are acces direct, în orice moment, la patru segmente logice ale căror adrese de bază (adresele de început ale segmentelor) se află în registrele de segment ale microprocesorului.

Registrul segmentului de cod (**CS**) conține adresa de început a segmentului din care sunt extrase instrucțiunile — segmentul de cod.

Stiva programului se afla în așa numitul segment de stiva căruia îi corespunde registrul segmentului de stiva (**SS**). Mai există două segmente de date, unul propriu-zis (**DS**) și unul suplimentar (**ES**), fiecăruia fiindu-i asociat câte un registru de segment ce conține adresa de început respectivă. Și registrele de segment sunt accesibile programatorului, conținutul lor putând fi modificat de anumite instrucțiuni.

Registrul pointerului de instrucțiuni

Registrul **pointerului de instrucțiuni** (*Instruction Pointer*) este similar registrului **contor de program** (*Program Counter*) al microprocesoarelor pe 8 biți. Acest registru este actualizat de către BIU și conține **offsetul** (distanța în octeți) a următoarei instrucțiuni, măsurat de la începutul segmentului curent de cod. Altfel spus, IP reprezintă, în mod normal, un pointer către următoarea instrucțiune ce urmează a fi *extrasă* de către BIU, iar atunci când este salvat în stivă, se modifică automat pentru a indica offsetul următoarei instrucțiuni ce urmează a fi *executată* de EU. Operarea cu conținutul IP, odată salvat în stivă, constituie calea prin care acesta poate fi modificat indirect în decursul execuției unui program.

Indicatorii de condiție

Microprocesorul 8086 are 6 biți de **stare** și 3 biți de **control** grupați în **regISTRUL indicatorilor de condiție** (*flags*). Cei de stare sunt poziționați de unitatea de execuție pentru a reflecta anumite proprietăți ale rezultatului unei operații aritmetice sau logice. Aceștia pot fi utilizați de un grup al setului de instrucțiuni pentru a modifica secvențialitatea execuției programului, în funcție de rezultatul operației anterioare. În general, indicatorii stării programului reflectă următoarele condiții:

- **AF: Auxiliary Carry Flag.** Dacă AF=1, a existat un transport dinspre bitul 7 spre bitul 8 sau un împrumut dinspre bitul 8 către bitul 7. Acest indicator este folosit mai ales în cazul instrucțiunilor ce implică operații aritmetice cu numere zecimale codificate binar.
- **CF: Carry Flag.** Dacă CF=1, a existat un transport dinspre, sau un împrumut către, cel mai semnificativ bit (*MSB*) al rezultatului reprezentat pe 8 sau 16 biți. Acest indicator este utilizat de instrucțiunile ce implică operații de adunare sau scădere cu numere reprezentate pe unul sau mai mulți octeți. Biți ai operanzilor din memorie sau registre pot fi izolați în CF prin intermediul instrucțiunilor de rotire și deplasare.
- **OF: Overflow Flag.** Dacă OF=1, a apărut o depășire aritmetică, adică s-a pierdut cel mai semnificativ bit al rezultatului datorită faptului că dimensiunea acestuia a depășit capacitatea de reprezentare a locației destinație. Este de remarcat faptul că există o instrucțiune (*INTerrupt On Overflow*) care generează o cerere de întrerupere pentru semnalarea apariției acestei situații.
- **SF: Sign Flag.** SF=1 indică faptul că, cel mai semnificativ bit al rezultatului are valoarea 1. Cum numerele întregi cu semn sunt reprezentate în *complement față de 2*, rezultă că SF arată semnul rezultatului (0=pozitiv, 1=negativ).
- **PF: Parity Flag.** PF=1 atunci când rezultatul are un număr par de biți poziționați pe 1. Poate fi utilizat pentru verificarea erorilor de transmisie a datelor.
- **ZF: Zero Flag.** Valoarea zero a rezultatului este evidențiată prin ZF=1.

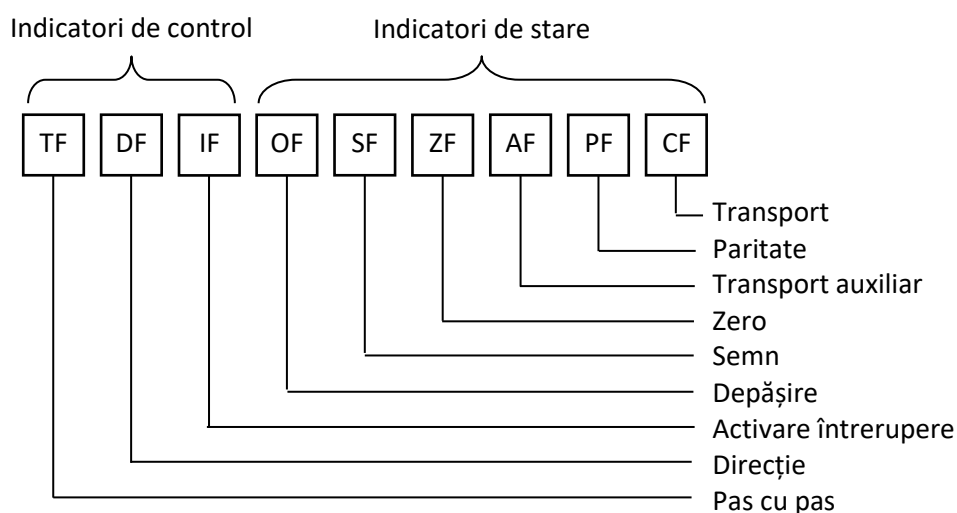


Fig. 5. Indicatorii de condiții

Cei 3 indicatori de control pot fi modificați prin program pentru a determina anumite operații ale procesorului. Astfel:

- **DF: Direction Flag.** Acest indicator este utilizat de instrucțiunile ce operează cu șiruri de caractere (*string*) și semnalizează autodecrementarea (DF=1) respectiv autoincrementarea (DF=0) registrelor SI și DI care conțin offsetul adreselor sursa și destinație ce intervin în transfer.
- **IF: Interrupt — enable Flag.** Setarea IF prin instrucțiunea *SeT Interrupt — enable flag* permite CPU să recunoască cererile de întrerupere externă mascabile, în timp ce resetarea aceluiași indicator cu instrucțiunea *CLear Interrupt — enable flag* le va dezactiva. Modificarea valorii lui IF nu are efect asupra întreruperilor generate intern în CPU sau a celor externe nemascabile.
- **TF: Trap Flag.** Dacă TF=1, procesorul intra în modul de **operare pas cu pas** în care CPU generează automat o întrerupere internă după fiecare instrucțiune pentru a permite examinarea stării programului și deci depanarea acestuia.