

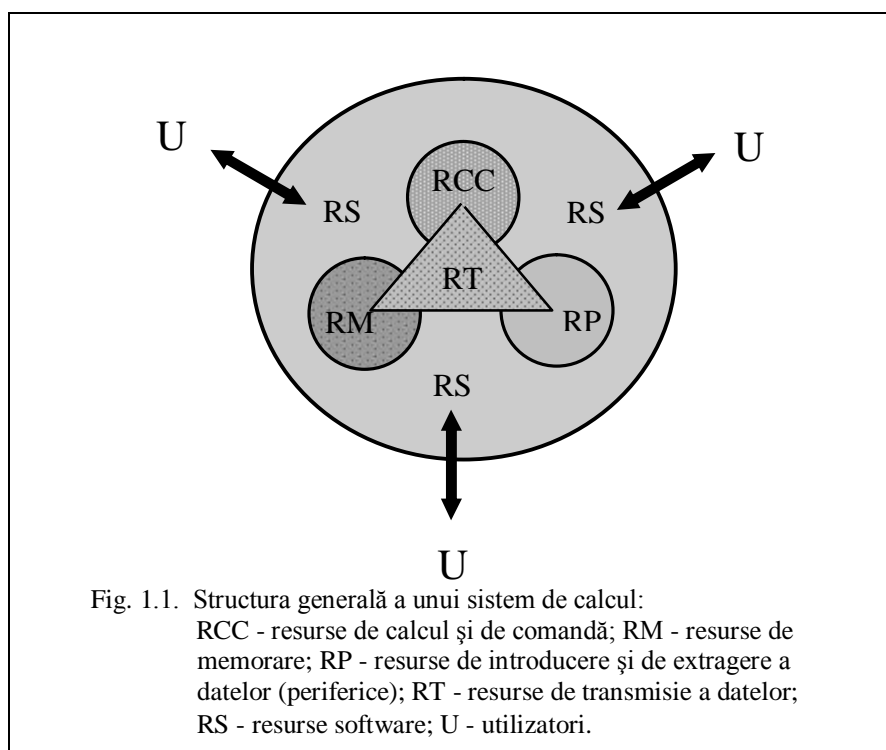
1. Funcțiile și structura unei unități centrale de procesare

Unitatea centrală de prelucrare (UCP) reprezintă un subsistem al unui sistem de calcul numeric care efectuează operațiile esențiale de prelucrare a datelor și coordonează activitatea celorlalte subsisteme. Sintetic spus UCP asigură disponibilitățile de calcul și de comandă a echipamentului numeric.

Pornind de funcțiile menționate, UCP este structurată într-o *diviziune de calcul și una de comandă*. Practic UCP este realizată sub forma unui procesor central care realizează funcțiile menționate. Ultima generație de calculatoare a impus procesorul realizat pe un singur circuit integrat pe scară foarte largă (VLSI - Very Large Scale Integrated) numit *microprocesor* (μP).

1.1. Subsisteme ale Unității Centrale de procesare

Un sistem de calcul este constituit dintr-un ansamblu de componente funcționale fizice (*hardware*) și logice (*software de bază*) care interacționează în scopul satisfacerii unor cerințe, adesea contradictorii, ale utilizatorilor. Sistemul de calcul prelucrează date într-o formă specifică și oferă rezultatele într-o formă accesibilă utilizatorilor. În figura 1.1 este ilustrat modelul topologic de bază al unui calculator care evidențiază componentele fizice și poziția acestora în raport cu cele logice .



După cum se observă componenta software apare ca o anvelopă a componentelor hardware, oferind exploatarea facilităților acestora și interfața cu utilizatorii. Din punct de vedere conceptual, funcționarea unui sistem de calcul poate fi văzută ca un flux de instrucțiuni a căror execuție este inițiată de introducerea datelor de prelucrat și finalizată de obținerea rezultatelor.

Pornind de la cele două fluxuri, *instrucțiuni și date*, pot fi realizate patru tipuri de sisteme de calcul și anume:

- a) sisteme cu fluxuri de instrucțiuni și date unice (**SISD** - **Single Instruction Single Data**);
- b) sisteme cu flux de date unic și mai multe fluxuri de date (**SIMD** - **Single Instruction Multiple Data**);
- c) sisteme cu mai multe fluxuri de instrucțiuni și flux de date unic (**MISD** - **Multiple Instruction Single Data**);
- d) sisteme cu fluxuri de instrucțiuni și date multiple (**MIMD** - **Multiple Instruction Multiple Data**).

Arhitectura **SISD** este reprezentată de sistemele clasice *monoprosesor* în timp ce arhitecturii **SIMD** îi sunt specifice sistemele cu *procesare paralelă*. În ceea ce privește arhitectura **MISD** aceasta este reprezentată de sistemele cu procesoare “*pipeline*”. Caracteristice pentru arhitectura **MIMD** sunt sisteme de calcul *multiprosesor* și *rețelele de sisteme de calcul*. În continuare vor fi prezentate trăsături ale UCP aferente sistemelor SISD. Principalele componente ale unei astfel de UCP sunt *unitatea aritmetico-logică, unitatea de comandă și memoria*.

Unitatea aritmetico-logică (UAL) reprezintă un subsistem al UCP care efectuează operațiile aritmetice și logice elementare. Operațiile pot fi efectuate *paralel*, când se prelucrează simultan toți biții unui cuvânt, sau *paralel - serie* când se prelucrează simultan biții unei zone a cuvântului transmis serial. Durata unei operații (cu excepția înmulțirii și împărțirii) este egală cu durata unui ciclu al procesorului.

UAL preia operanzii pe care îi prelucrează, din memorie sau din registre, rezultatele fiind transmise înapoi în memorie sau în mediul extern prin intermediul subsistemului de intrare-ieșire. În general o UAL primește doi operanzi și un cod al operației și furnizează un rezultat și eventuale informații despre acesta. În figura 1.2 este prezentată structura principală a unei UAL simple cu operanzii exprimați în virgulă fixă.

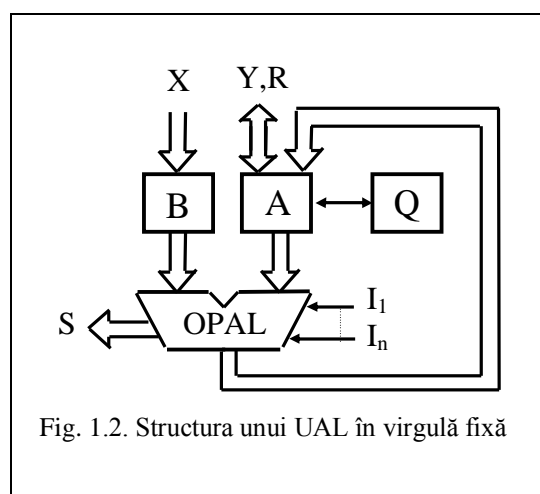


Fig. 1.2. Structura unui UAL în virgulă fixă

UAL este constituită dintr-un operator aritmetico-logic OPAL și 3 registre A,B,Q. OPAL reprezintă un circuit combinațional ce constă dintr-un sumator paralel și un bloc asociat funcțiilor logice. Registrul A păstrează la un moment dat unul din operanzi sau rezultatul operației, iar registrul B conține al doilea operand. În ceea ce privește registrul Q acesta este destinat realizării de înmulțiri / împărțiri prin deplasări succesive. Selectarea operației ce se va efectua la un moment dat se face cu ajutorul variabilelor I_1, I_2, \dots, I_n aplicate de asemenea la intrarea OPAL. În afara rezultatului care se obține în registrul A, la ieșirea OPAL se mai obține un cuvânt de stare format din indicatori de condiție.

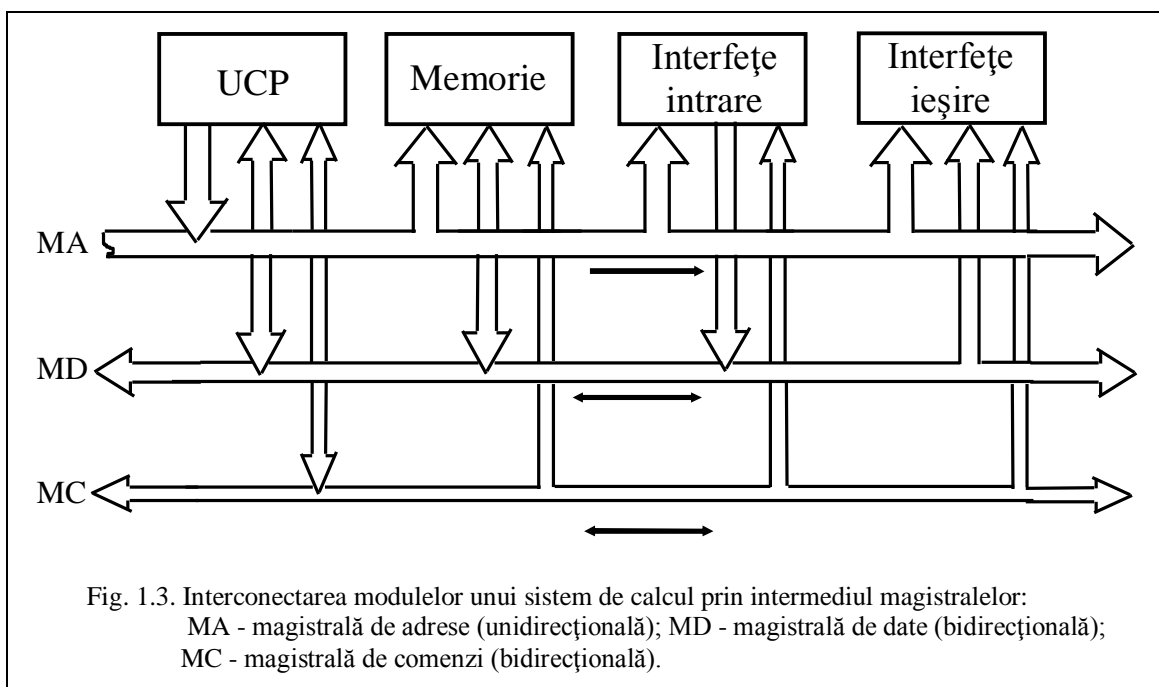
Unitatea de comandă (UC) reprezintă un subsistem al UCP care generează succesiunea de semnale de comandă pentru coordonarea activității întregului sistem de calcul (*transferuri de informație, operații aritmetico-logice, citire, interpretare și execuție instrucțiuni*). În funcție de modul de implementare se întâlnesc UC convenționale (cablate) și UC microprogramate.

UC *convențională* reprezintă un automat secvențial în care semnalele de stare testate de aceasta sunt considerate semnale de intrare, iar semnalele generate pentru controlul resurselor sistemului sunt semnale de ieșire.

UC *microprogramată* generează semnalele de comandă necesare funcționării sistemului numeric prin citirea unor cuvinte de comandă stocate într-o memorie specială.

Realizarea funcțiilor UCP necesită conectarea acestora cu celelalte unități ale sistemului de calcul. Schimbul de informație se realizează prin intermediul *magistrelor de comunicație* care reprezintă grupuri de interconexiuni cu funcții similare ce leagă diversele secțiuni ale unui sistem de calcul. La nivelul unui sistem de calcul principial există două categorii de magistrale: *interne și externe*. Magistralele interne conectează unități ale CPU iar cele externe interconectează modulele sistemului. Fiecare din cele două categorii de magistrale poate fi divizată funcție de tipul informației transferate în magistrale de *adrese*, de *date* și de *comenzi*.

Funcție de tipul transmisiei magistralele pot fi *paralele* (MP) sau *seriale* (MS). MP transferă toți biții unui cuvânt binar simultan, deci numărul de linii este egal cu lungimea cuvântului. MS sunt formate dintr-o singură linie prin care cuvintele binare se transmit bit cu bit. Practic, toate magistralele interne ale UCP și cele care conectează UCP cu memoria și cu interfețele de intrare-ieșire sunt organizate ca MP. În ceea ce privește MS acestea sunt utilizate pentru a conecta unele periferice (mouse, imprimante etc.) sau pentru interconectarea sistemelor de calcul în cadrul rețelelor. În figura 1.3 este reprezentat modul de interconectare a principalelor module ale unui sistem de calcul prin intermediul magistrelor.



Magistrala de date este destinată atât transferului unidirecțional de instrucțiuni de la memorie, cât și a celui bidirecțional de date între UCP, memorie și interfețele de intrare-ieșire. Sensul transferului de informație este supervizat de către unitatea de comandă a UCP. Numărul de linii al magistralei de date și implicit lungimea cuvântului de date (8, 16, 32, 64 biți) constituie o caracteristică importantă a sistemului de calcul.

Magistrala de adrese este destinată transferului unidirecțional al cuvântului binar ce reprezintă adresa locației de memorie al cărui conținut urmează a fi citit. Numărul de biți ce poate fi transferat prin magistrala de adrese determină o altă caracteristică a sistemului și anume capacitatea memoriei ce poate fi adresată.

Magistrala de control vehiculează semnale care asigură coordonarea activităților într-un sistem de calcul. Numărul liniilor acestei magistrale este egal cu numărul semnalelor de control al UCP, dintre care pot fi menționate: semnalul de ceas pentru sincronizare, semnalele de citire/scriere din și în memorie sau de la interfețele de intrare/ieșire etc.

1.2. Conceptul de microprocesor

Microprocesorul (μP) reprezintă un circuit logic de o maximă complexitate, reprezentat din punct de vedere funcțional printr-o structură formată din *elemente secvențiale și combinaționale*. Caracteristica specifică a unui μP , în raport cu o configurație clasică, este reprezentată de flexibilitatea sa la nivel software. Aceasta face ca implementarea unei structuri logice pentru o aplicație dată să nu reprezinte o intervenție la nivel hardware ci să se reducă la implementarea unui program specific.

Termenul de *microprocesor* a fost introdus de firma Intel în anul 1972 și este legat de realizarea de către această firmă a primelor μP , care se numeau 4004 și 8008. Acestea aveau

lățimea magistralei de date de 4 și respectiv 8 biți și se consideră că aparțin generației întâi de μP .

Anul 1976 marchează apariția generației a doua de μP , prin lansarea produselor 8080 al aceleiași firme și Z80 al firmei Zilog. Este de remarcat faptul că Z80 a fost un μP cu performanțe deosebite pentru generația sa, în acest sens prefigurând într-o anumită măsură noi concepte specifice viitoarelor generații de μP .

Lansarea pe piață, de către firma Intel a μP 8086 marchează pe lângă trecerea la μP cu lungimea magistralei de date de 16 biți apariția generației a treia. La scurt timp și firmele Motorola și Zilog lansează produsele M68000 și Z8000 din cadrul aceleiași generații. O manevră interesantă a fost lansarea de către Intel în 1979 a μP 8088, care este identic în interior cu 8086 iar în exterior lucrează pe 8 biți. Această apariție venea în sprijinul deținătorilor de sisteme pe 8 biți care voiau să beneficieze de facilitățile oferite de noul produs pe 16 biți. Consacrarea definitivă a produselor Intel se produce după ce firma IBM lansează microsistemele **IBM PC-XT** care folosesc μP 8086/8088.

În anul 1993 Intel lansează primul μP al generației a cincea și anume Pentium. Aceasta inaugurează seria μP pe 64 biți cu o arhitectură puternic diferită față de μP precedente. Considerațiile privind evoluția μP au avut în vedere dispozitive ale căror caracteristici și arii posibile de aplicație s-au dezvoltat pornind de la tipurile de bază lansate în anii '70. Acestea sunt cunoscute ca *microprocesoare de uz general* sau pur și simplu *microprocesoare*.

Pot fi identificate însă și alte direcții de dezvoltare spectaculoasă a dispozitivelor de procesare a datelor.

Procesoarele **RISC (Reduced Instruction Set Computer)** prezintă un set redus de instrucțiuni în vederea creșterii vitezei de prelucrare. Ideea de bază a arhitecturii RISC constă în reducerea setului instrucțiunilor de bază la un minimum necesar, punând acces pe cele folosite mai des și optimizându-le pentru cea mai rapidă execuție posibilă. Restul instrucțiunilor trebuie să poată fi executate prin combinarea celor selectate. Viteza de execuție la un procesor RISC este de 1,5-1,7 ori mai mare decât la unul **CISC (Complex Instruction Set Computer)**.

O evoluție interesantă a au **transputerele**, procesoare care pot asigura o prelucrare paralelă a datelor. Tot în categoria procesoarelor situate în afara uzului general trebuie menționate circuitele **ASIC (Application Specific Integrated Circuit)** în rândul cărora un loc aparte îl ocupă procesoarele de semnal.

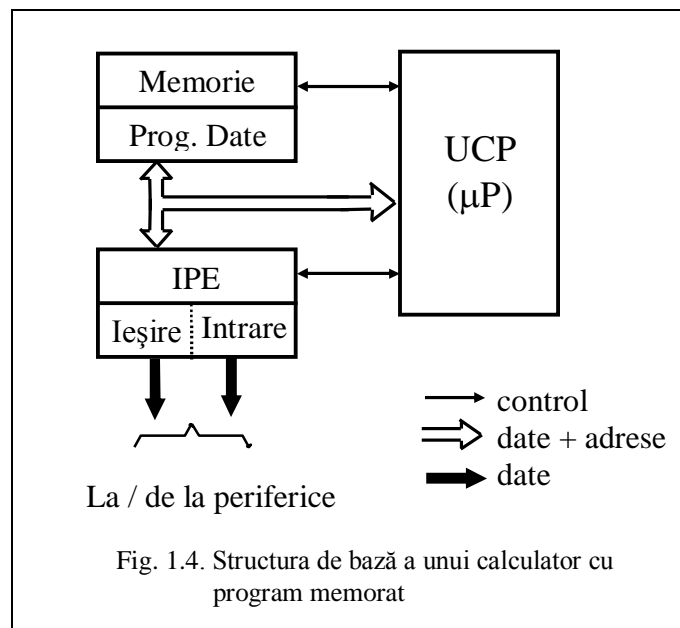
Dinamica extraordinară a tehnicii de calcul poate fi mai bine receptată, dacă amintim faptul că primul calculator electronic a apărut în anul 1947, iar evoluția acestora a fost marcată de progresul continuu al tehnologiei electronice. Calculatoarele la care UCP este realizată cu μP aparțin generației a patra, celelalte generații având ca tehnologii dominante: tubul electronic pentru generația întâi, componentele statice (tranzistoare) discrete pentru generația a doua și circuitele integrate pe scară redusă pentru generația a treia. Datorită unui raport performanță/cost în continuă creștere, calculatorul penetrează în mediile cele mai diverse fiind considerat principalul factor favorizant al trecerii la societatea post-industrială informatizată.

Indiferent cărei generații aparțin, toate tipurile de calculatoare prezintă două caracteristici comune și anume:

- lucrează pe baza unui program memorat;

- comunicarea programatorului cu calculatorul se realizează prin intermediul limbajelor artificiale, înlocuirea acestor limbaje cu cel natural reprezentând unul din obiectivele importante ale generației a cincea de calculatoare.

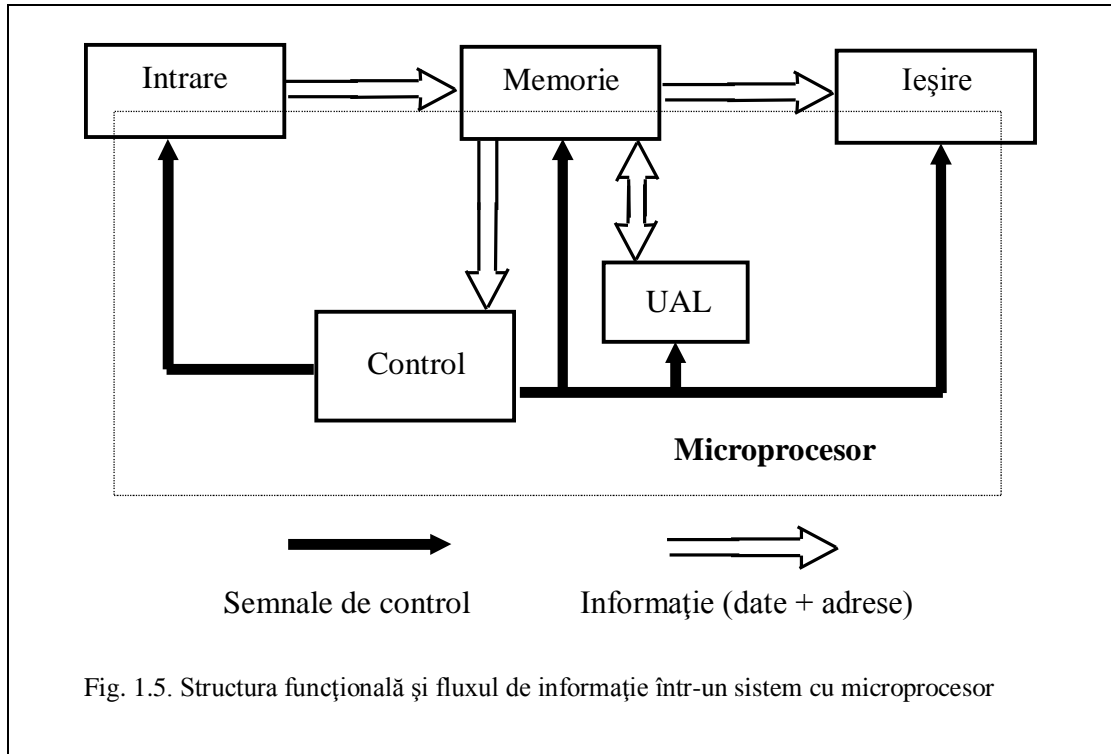
Conceptul de calculator numeric cu program memorat a fost introdus de John von Neumann în anul 1947. Potrivit acestui concept instrucțiunile și datele sunt memorate împreună și sunt accesibile în același mod. În figura 1.4, care prezintă structura de bază a unui calculator numeric bazat pe o mașină von Neumann, este evidențiat faptul că în memorie sunt stocate atât secvențele de instrucțiuni ce constituie programul, cât și datele (de intrare, intermediare, de ieșire). Prin secțiuni specifice calculatorul schimbă informație cu echipamentele periferice care pot fi atât convenționale (tastatură, ecran, imprimantă etc.), cât și de proces (traductoare și elemente de execuție).



Practic un calculator numeric realizează cinci funcții de bază și anume:

- funcția de *INTRARE* care permite conectarea lumii exterioare la sistem;
- funcția de *IEȘIRE* care permite conectarea sistemului la lumea exterioară;
- funcția de *MEMORARE* care asigură păstrarea datelor și a instrucțiunilor programului;
- funcția *ARITMETICO - LOGICĂ* ce permite efectuarea operațiilor de calcul (aritmetice și logice) în sistem;
- funcția de *CONTROL* care grupează totalitatea operațiilor de secvențializare și coordonare în cadrul sistemului.

Celor cinci funcții le corespund tot atâtea secțiuni care nu sunt însă integral acoperite de către UCP (μP). În general secțiunile de intrare-ieșire și memoria sunt în mare măsură exterioare μP. După cum se observă din figura 1.5 secțiunile **aritmetico-logică** și **control** sunt integrate în μP.



Uzual, informația este adusă prin intermediul funcției de INTRARE în memoria sistemului. Din memorie, informația (instrucțiuni și date) este citită și decodificată, executându-se secvențial instrucțiunile programului. Rezultatele sunt apoi transferate, prin intermediul funcției IEȘIRE în afara sistemului. Toate operațiile, a căror înlănțuire temporală este redată simplificat în figura 1.6, sunt coordonate prin atribute ale funcției CONTROL.

În intervalul T_1 informația (instrucțiuni și date) se memorează activându-se secțiunile *intrare și memorie*. În T_2 codul este preluat din memorie și executat iar în T_3 cu ajutorul funcției *ieșire* rezultatele sunt transferate în memorie.

După cum s-a arătat, totalitatea informației (instrucțiuni, date, rezultate) se memorează și se prelucrează în formă binară. Ele trebuie să fie *interpretabile* în mod unic de către μP , prezentându-se din acest motiv într-un format specific. Acest format conține o combinație de lungime fixă de simboluri binare care constituie *cuvinte de instrucțiuni* respectiv *cuvinte de date*.

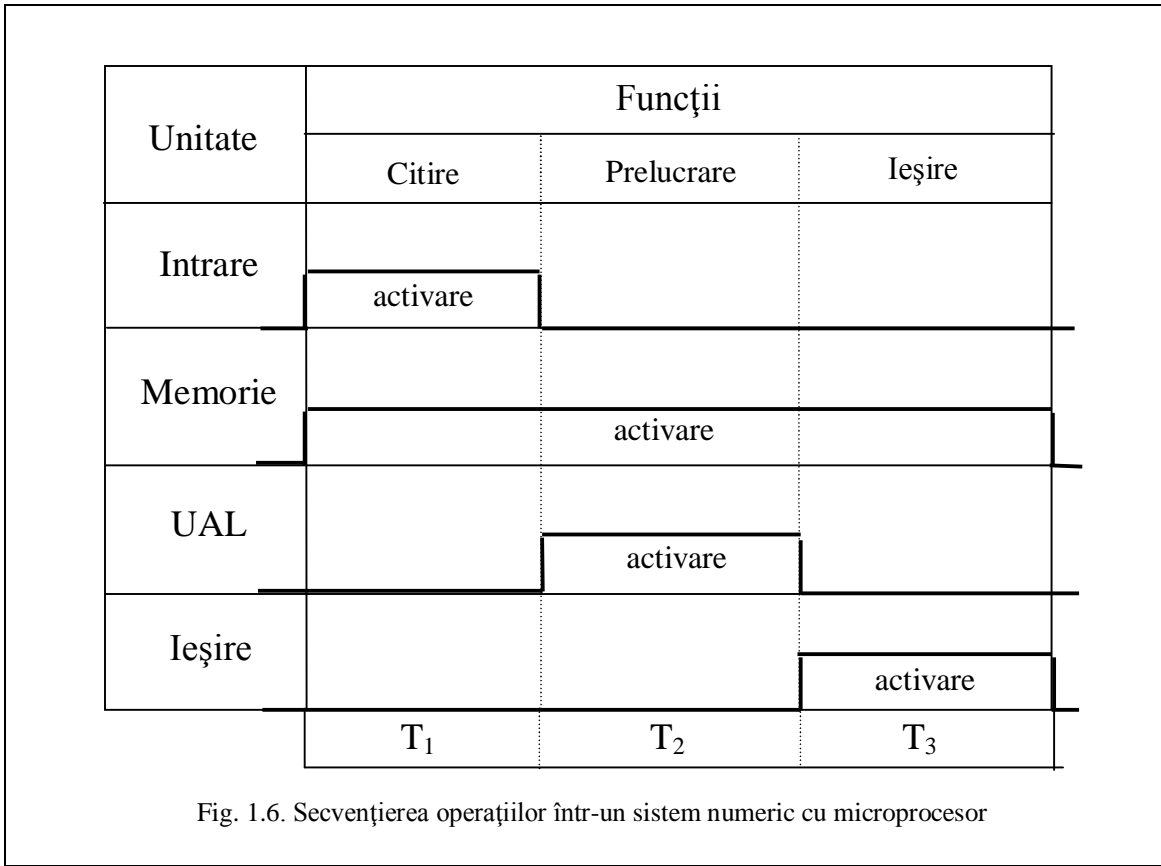


Fig. 1.6. Secvențierea operațiilor într-un sistem numeric cu microprocesor