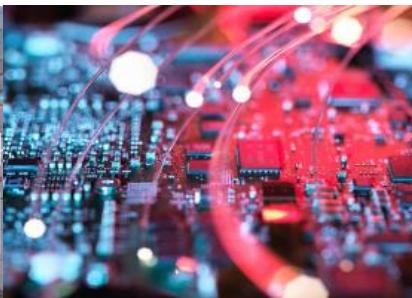

Arhitecturi de calculatoare și sisteme de operare (ACSO)

AC1.

(11.10.2024, ora 8, sala P02)

<https://ocw.mit.edu/search/?t=Computer%20Design%20and%20Engineering&type=course>



AC1. Introducere în Arhitectura Calculatoarelor

AC1 – 4.10.2024

- **Definirea arhitecturii calculatoarelor**
- **Istoric**

AC1 - 11.10.2024 – Concepte fundamentale

- Tipuri de arhitecturi (Von Neumann vs. Harvard)
- Elementele de bază ale unui calculator (unitatea centrală de procesare, memorie, periferice)



Calculatorul digital (CD)

CD – echipamente complexe care conțin milioane de componente electronice elementare

- **Sistem ierarhic**- format din câteva subsisteme ce interacționează până la subsistemul cel mai de jos;
- Imposibil de definit;
- Nu există definiție unică.



Calculator digital. Definiții

Def.1 - sistem destinat automatizării calculelor informației discrete în conformitate cu anumiți algoritmi.

Def. 2 - o reuniune între ansamblul de echipamente fizice care constituie **HARDWARE-ul**, ansamblul de microprograme care constituie **FIRMWARE-ul** și ansamblul de programe care constituie **SOFTWARE-ul**, ce permite prelucrarea logică și aritmetică a **informației discrete** la o **viteză foarte mare** (de ordinul milioanelor de operații pe secundă) în conformitate cu **algoritmii** dați.

Def. 3 - automat finit. Un sistem cu stare finită care prelucreză cu o **viteză foarte mare**, informația discretă în conformitate cu **algoritmii** impuși.

Calculatorul digital. Caracteristici.

Trăsături comune specifice tuturor calculatoarelor:

- ❑ Informația este discretă, datele sunt numerice;
- ❑ Prelucrările sunt complet automatizate;
- ❑ Prelucrarea se face în conformitate cu algoritmii impuși;
- ❑ Viteza de prelucrare este foarte mare;
- ❑ Natura prelucrărilor este aritmetică și logică.

Diferențele între diferite calculatoare sunt date de mărime, preț de cost, performanță, domeniul de aplicabilitate și organizare.

Studiul sistemelor de calcul

Studierea sistemelor de calcul se face pe 3 niveluri:

1. **Arhitectura calculatoarelor** – atributele unui sistem văzut de un programator, care au **impact direct asupra execuției logice** a unui program.

Arhitectura calculatorului - comportamentul unui sistem de calcul aşa cum este văzut de un programator, referindu-se la:

- natura tipurilor de date;
- operațiile executate;
- organizarea memoriei;
- setul de registre accesibile unui programator;
- setul și formatul instrucțiunilor;
- tehniciile de adresare;
- mecanismele de I/O.

Studiul sistemelor de calcul

2. Organizarea calculatoarelor – unităile operaționale și la interconectarea lor care realizează specificațiile arhitecturale.

Atributele de organizare includ acele **detalii Hardware transparente pentru un programator**, cum ar fi:

- semnale de control;
- detalierea structurală a unor blocuri funcționale;
- interfața între calculator și periferice;
- tehnologia de memorare;
- frecvența de ceas etc.

Studiul sistemelor de calcul

3. Implementarea calculatoarelor - Hardware, la **structura fizică** care include:

- tehnici de proiectare logică;
- așezarea pe circuite;
- interconectarea fizică;
- alimentarea;
- metode de testare;
- interfața între semnale, etc.

Arhitectura Von Neumann vs. Arhitectura Harvard

Două modele de arhitectură pentru sistemele de calcul, fiecare având un mod diferit de gestionare a memoriei și a comunicațiilor dintre unitatea centrală de procesare (CPU) și memorie.

Aspect	Arhitectura Von Neumann	Arhitectura Harvard
Memorie	Memorie comună pentru date și instrucțiuni	Memorie separată pentru date și instrucțiuni
Bus	Bus unic pentru date și instrucțiuni	Bus separat pentru date și instrucțiuni
Performanță	Posibilă limitare din cauza gâtului de sticlă	Performanță mai mare datorită accesului paralel
Flexibilitate	Mai flexibilă pentru manipularea instrucțiunilor ca date	Mai puțin flexibilă, dar mai eficientă
Utilizare	PC-uri, servere generale	Sisteme embedded, microcontrolere

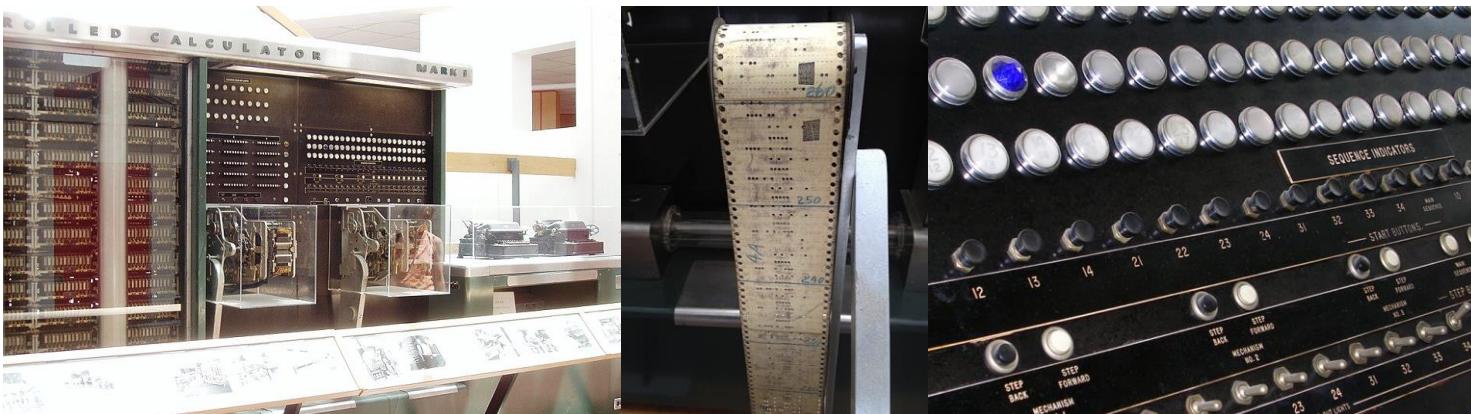
<https://www.youtube.com/watch?v=i1zjGlmnqxE>

<https://www.youtube.com/watch?v=4nY7mNHLrLk>

Harvard Mark I - 1944

Harvard Mark I, sau IBM Automatic Sequence Controlled Calculator (ASCC) – cel mai mare calculator electromecanic din istorie, construit în timpul războiului de o echipă (Claire D. Lake, Francis E. Hamilton, Benjamin M. Durfee) sub conducerea lui Howard Aiken, conform ideii sale din 1937.

Arhitectura Harvard. Harvard Mark I avea aproape 51 de picioare (16 m) lungime și 8 picioare (2,4 m) înălțime, cântărind peste 9.445 de livre (4,3 t). Continea peste 530 de mile (850 km) de fire cu trei milioane de conexiuni. A fost lansat la Laboratorul de Calcul Harvard (din mai 1944 – anterior a fost localizat la IBM Endicott Labs).



https://pl.wikipedia.org/wiki/Harvard_Mark_I

Harvard Mark I - 1944

- mașină universală**, proiectată pentru:
 - tabele de funcții
 - integrare numerică
 - rezolvarea ecuațiilor diferențiale
 - rezolvarea sistemelor de ecuații algebrice liniare
 - analiză armonică
 - calcule statistice.





Harvard Mark I - 1944



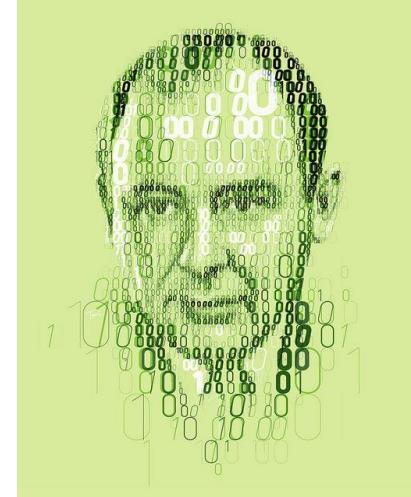
Grace Brewster Hopper

prima care a conceput teoria limbajelor de programare independente de mașină și a folosit această teorie pentru a dezvolta limbajul de programare **FLOW-MATIC** și **COBOL**, un limbaj de programare de nivel înalt timpuriu, care este încă utilizat astăzi. De asemenea, a fost una dintre primele programatoare ale computerului **Harvard Mark I**. El i se atribuie scrierea primului manual de utilizare a unui calculator, „**A Manual of Operation for the Automatic Sequence Controlled Calculator.**”

https://pl.wikipedia.org/wiki/Harvard_Mark_I

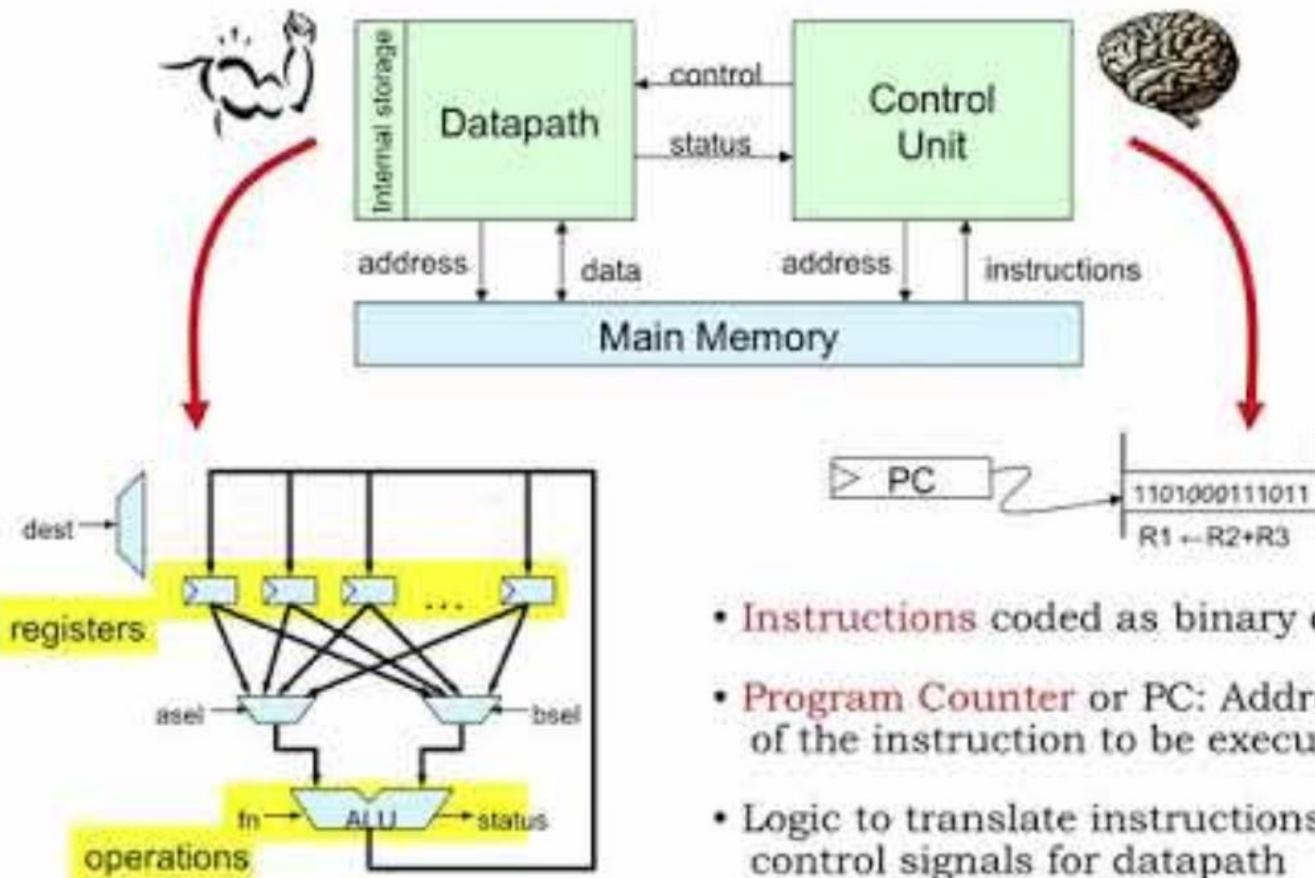
Principiul comenzii după program

- Modelul pe care se bazează majoritatea calculatoarelor contemporane: **John von Neumann**;
- Anterior: **Principiul programului memorat** sau principiul comenzii după program;
- Structură: mașină cu program memorat;
- Programul scris în cod mașină este memorat în memoria calculatorului împreună cu datele de prelucrat, => sunt posibile manipulări asupra programului în mod asemănător cu prelucrarea datelor;
- Conceptul de **program memorat** constă în reprezentarea algoritmilor necesari pentru rezolvarea unei probleme cu ajutorul unei scheme logice care cuprinde două tipuri de operatori: - operatori de procesare; - operatori de secvențiere.



1903-1957

Anatomy of a von Neumann Computer



Principiul comenzii după program

- Operatorii de **procesare** specifică modul de transformare a datelor; O
- Operatorii de **secvențiere** stabilesc succesiunea de execuție a operatorilor de prelucrare;
- Acest principiu poate fi implementat prin diferite scheme;
- Cel mai răspândit model: **principiul von Neumann** de comandă după program;
 -> John von Neumann, 1945; => arhitectura von Neumann;
- Trăsătura principală: proceduralitatea.

Orice problemă trebuie prezentată calculatorului sub forma unei secvențe de operații (program).

Principiul von Neumann

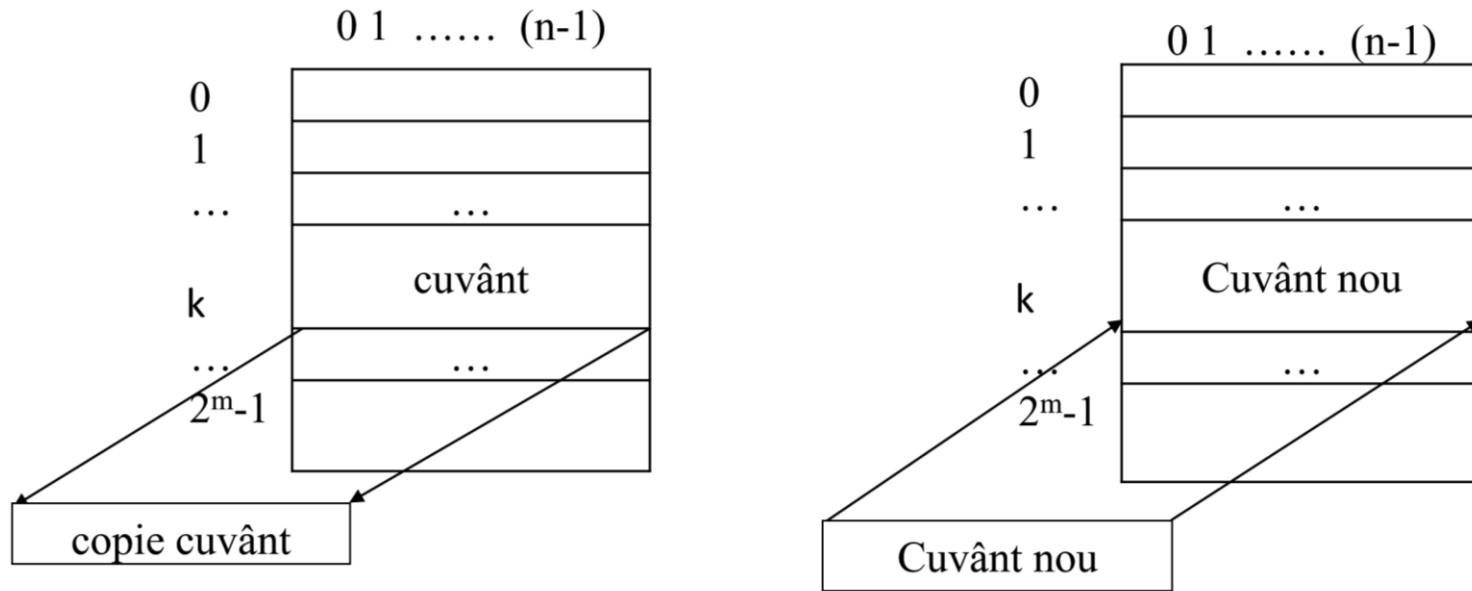
- ❑ **5 principii** stau la baza definirii modelului von Neumann a unui calculator numeric;
- ❑ **Scop:** simplificarea proiectării și implementării unui calculator;

1. **Informația** în calculatorul digital **este codificată binar:**
 - este reprezentată prin cifre binare numite biți (0 sau 1);
 - n biți: un cuvânt:
 - n=8 biți => octetul sau byte;
 - n=16 biți => semicuvântul sau 2 bytes;
 - n=32 biți => cuvânt sau 4 bytes;
 - n=64 biți => dublucuvânt sau 8 bytes.
 - un cuvânt: o dată -> **cuvânt dată** sau o comandă -> **cuvânt de comandă** sau **instrucțiune**.

Principiul von Neumann

2. Diferențierea între cuvântul dată și cuvântul instrucțiune se face prin modul de utilizare (contextul de utilizare).

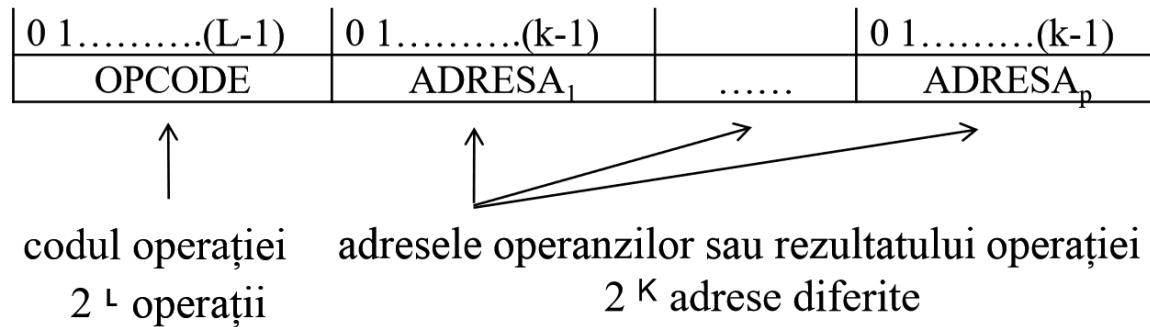
3. Cuvintele dată și comandă sunt depuse în locații de memorie care se identifică printr-un număr de locație numit adresă.



Prinzip von Neumann

4. Algoritmul în execuție se reprezintă sub forma unei succesiuni de cuvinte de comandă (instrucțiuni).

Instructiune: formă generală:



5. Efectuarea calculelor corespunzând algoritmului problemei de rezolvat este univoc determinată de execuția secvențială a instrucțiunilor care constituie programul pentru algoritmul implementat.

Prima instrucțiune executată dintr-un program corespunde adresei de start a programului.

Principiul von Neumann

Adresa următoarei instrucțiuni:

- este determinată în mod automat de către CPU și specificată prin adresa sa din memorie;
- poate fi adresa secvențial următoare sau orice altă adresă din spațiul de adrese;
- este determinată de un numărator de program (Program Counter – PC) – pointer la adresa respectivă.

Definiție: Multimea programelor acceptate de un calculator se numește **clasa funcțiilor realizabile de calculator**;

Concluzie: proceduralitate;

Orice problemă trebuie prezentată calculatorului ca o secvență de operații, adică o secvență de instrucțiuni.

Principiul von Neumann

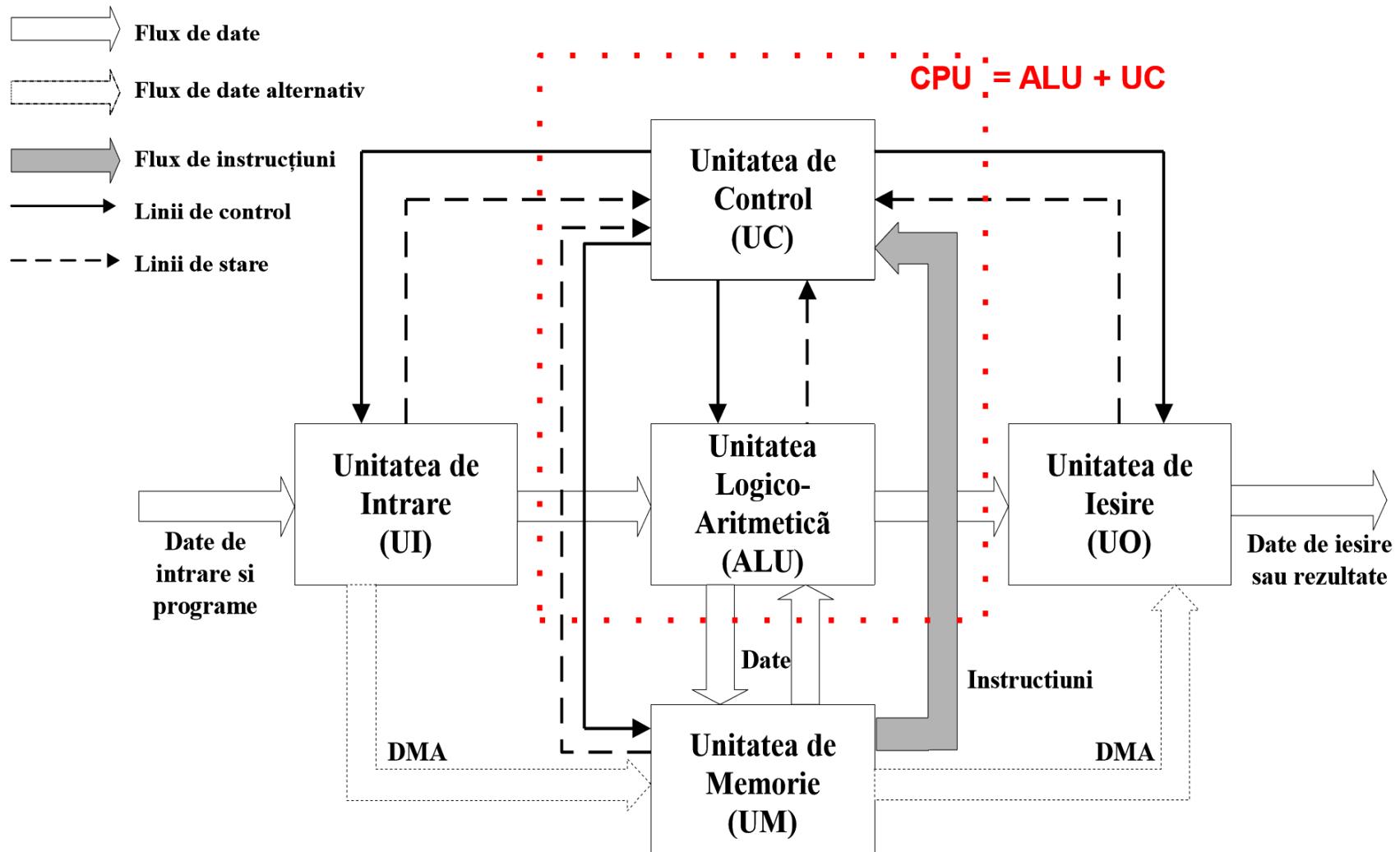
- În conformitate cu principiile Von Neumann;
- Orice calculator numeric: un sistem format din mai multe subsisteme sau unități.
- Acestea sunt organizate într-o structură generală care este aproape standardizată și corespunde modelului von Neumann.
- Un calculator digital trebuie să conțină 5 componente:
 - un **mediu de intrare**: care să asigure preluarea datelor sau instrucțiunilor;
 - un **mediu de memorare**: în care se depozitează programul și datele aferente;
 - un **mediu de prelucrare**: care asigură funcțiile de natură aritmetică și logică pentru prelucrarea datelor;
 - un **mediu de ieșire**: care să asigure extragerea rezultatelor ;
 - un **mediu de control**: capabil să interpreteze instrucțiunile extrase din memorie și să controleze întreg procesul de automatizare a calculelor.

Structura de bază a unui calculator

Structura fundamentală a unui calculator digital – “structura în 5 unități”:

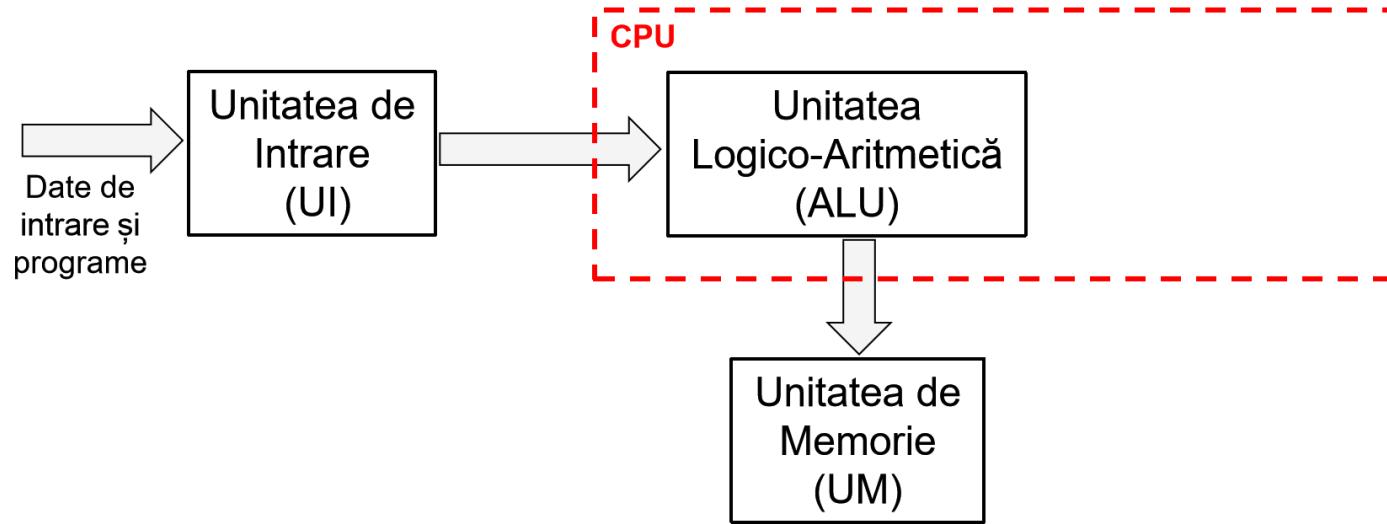
1. Unitatea de Intrare (UI) – Input Unit (IU);
2. Unitatea Logico-Aritmetică (ALU) – Arithmetic and Logic Unit;
3. Unitatea de Memorie (UM) – Memory Unit (MU);
4. Unitatea de Ieșire (UO) – Output Unit (OU);
5. Unitatea de Control (UC) – Control Unit (CU).

Structura de bază a unui calculator



Structura de bază a unui calculator

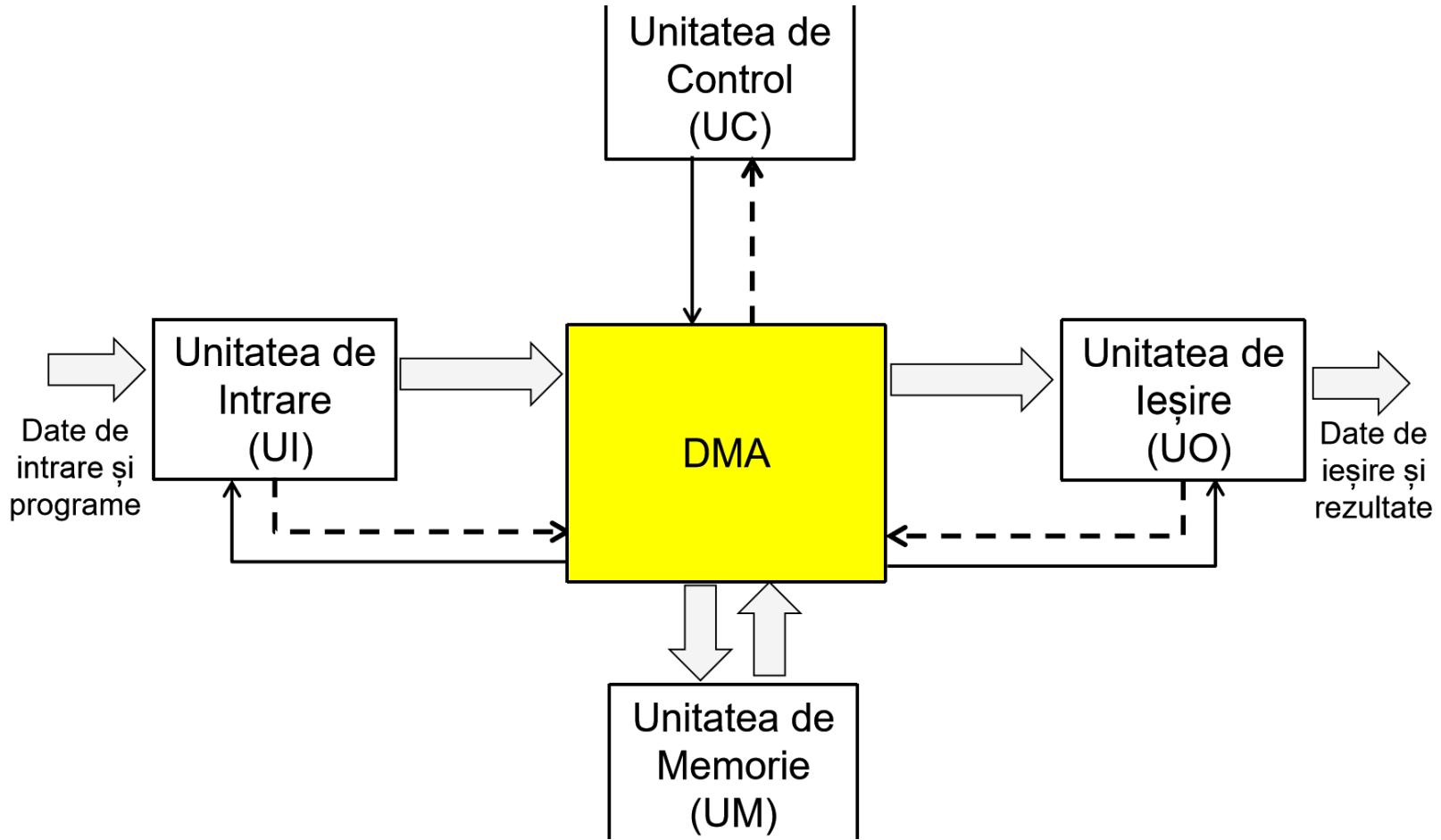
Datele și programele sunt introduse în calculator prin intermediul UI;
Initial: datele și programele erau transferate în UM prin ALU;



DAR: transferul de date către UM este un proces lent ce îngreunează activitatea CPU (ALU+UC);

Funcționarea calculatorului digital

Astăzi: transferul se face prin DMA (Direct Memory Access), astfel:

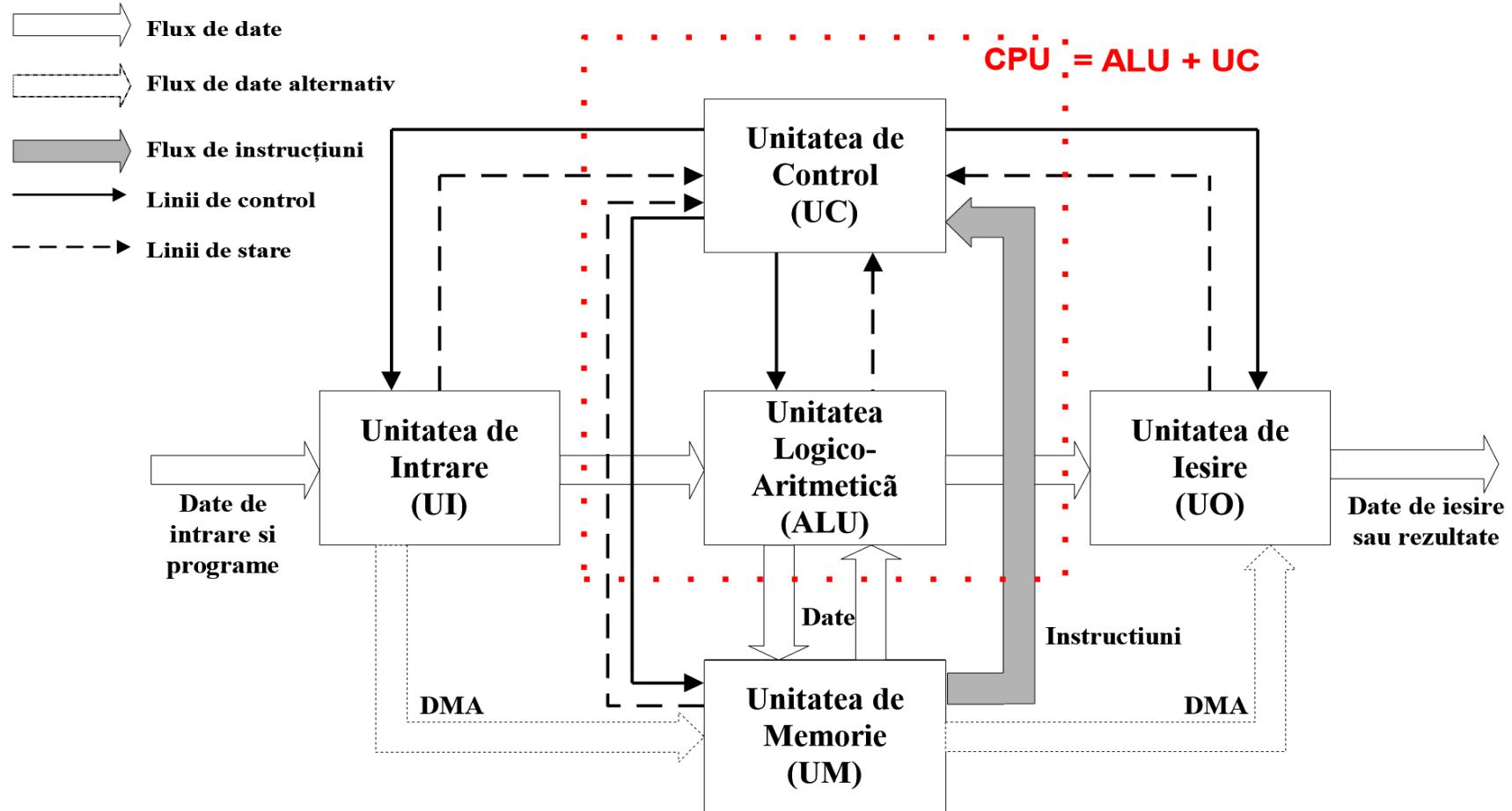


Funcționarea calculatorului digital

Datele de intrare și programele sunt introduse în calculator prin intermediul UI;

- UI informează DMA despre prezența datelor/programelor la intrare;
- DMA informează UC despre prezența datelor/programelor la intrare;
- UC dă comandă către DMA spre a iniția transferul de date/programe;
- DMA transmite comanda mai departe către UI;
- UI pornește transferul datelor/programelor către DMA;
- DMA înaintează datele/programele către UM;
- Analog transferul de date/rezultate UM-UO via DMA.

Funcționarea calculatorului digital



Funcționarea calculatorului digital

După ce transferul de date/programe în UM a fost realizat, UM notifică UC;

- UC transmite comanda către UM pentru a-i fi înaintată instrucțiunea în curs de completare;
- UM transferă instrucțiunea la UC;
- UC interpretează instrucțiunea (OPCODE + ADRESE) și: -

Transmite comandă la UM prin care cere ca operanzele localizate la adresele specificate în corpul instrucțiunii să fie transferați în ALU;

- UM transferă operanzele în ALU;
- UC dă comandă la ALU să efectueze operația specificată în OPCODE cu operanzele primite din UM;
- Rezultatul operației este stocat în ALU, în registrul Acumulator.

Funcționarea calculatorului digital

- Rezultatul din Acumulator (ALU) poate:
 - rămâne în ALU spre a fi supus la o nouă operație inițiată de o nouă instrucțiune;
 - transferat și stocat în UM (drept rezultat al instrucțiunii MOVE).
- Rezultatul stocat în UM poate fi afișat de către UO prin intermediul DMA;
- Transferul de date/rezultate din UM către UO este realizat într-un mod similar transferului UI-UM via DMA.

Unitatea de Intrare - UI

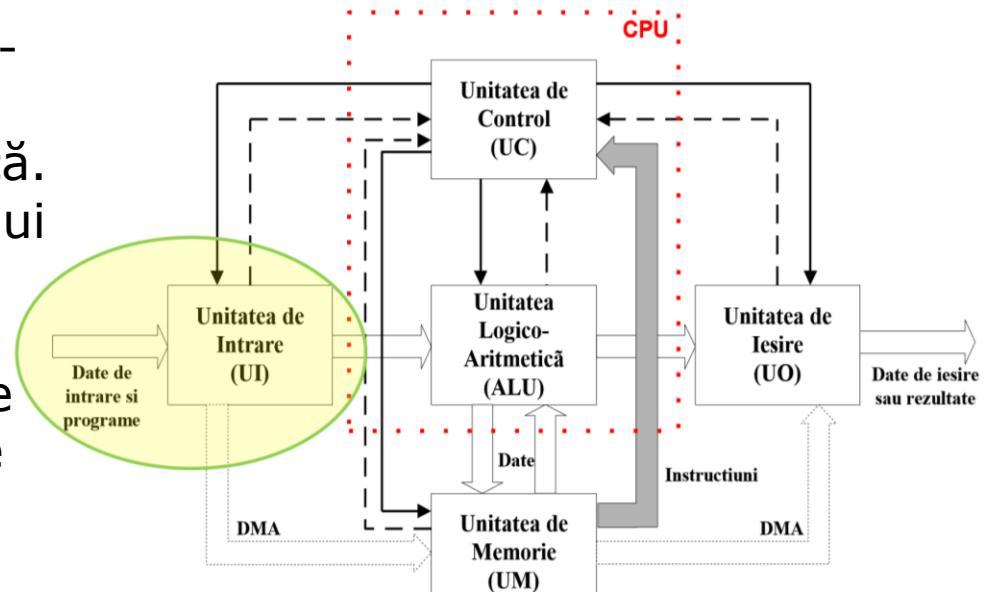
Face legătura dintre calculator și mediul exterior;

- Permite introducerea datelor și programelor în calculator;
- Două moduri de funcționare:

Direct: prin intermediul tastaturii, mouse-ului, scanner-ului etc.; mai simplu; viteza de operare este mai mică.

Indirect: prin intermediul unui mediu intermediar prin care se face citirea informațiilor: ex. cititoare de cartele, cititoare de bandă perforată, de microfilme etc.;

proces: operator uman -> document -> mediu intermediar -> cititor -> calculator; rapid; mult timp consumat la pregătirea datelor.

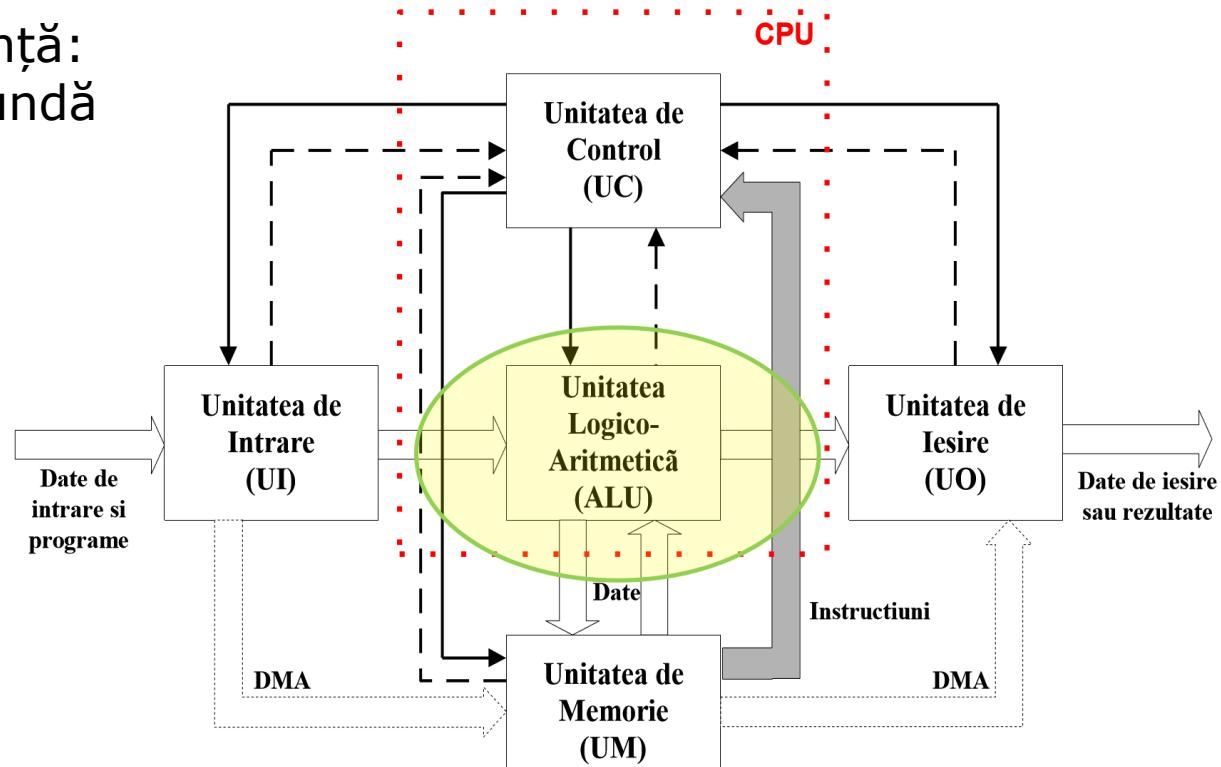


Unitatea Aritmetică și Logică- ALU

Parte componentă a Unității Centrale de Procesare (CPU);

- Singura unitate în care se generează informație nouă;
- Prelucrările sunt de natură logică și aritmetică;
- Viteză ridicată;
- Cea mai performantă și cea mai productivă unitate din calculator;

Criteriu de performanță:
de adunări pe secundă



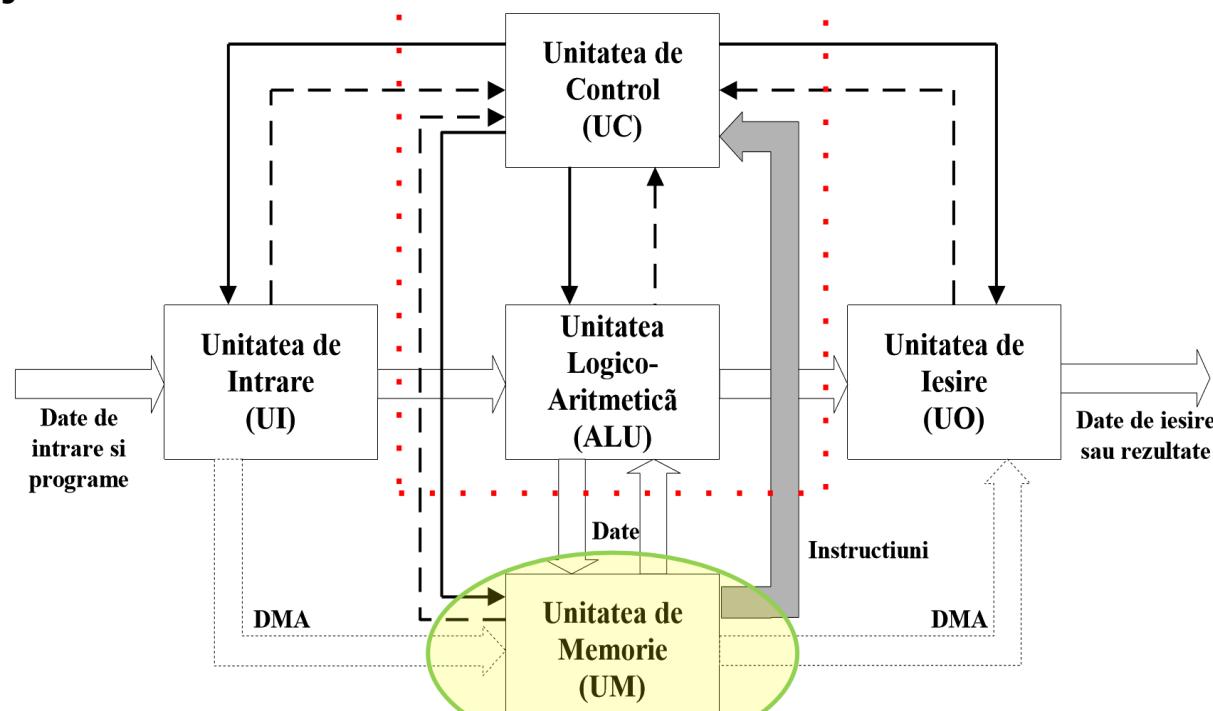
Unitatea de Memorie - UM

Stochează datele și programele;

□ 3 funcții de bază:

1. **Scriere** (înregistrare sau Write);
2. **Memorare** (stocare sau conservare sau Store);
3. **Citire** (extragere sau Read).

□ Unitatea de memorie nu are rol de procesare (păstrează informația fără să o altereze)

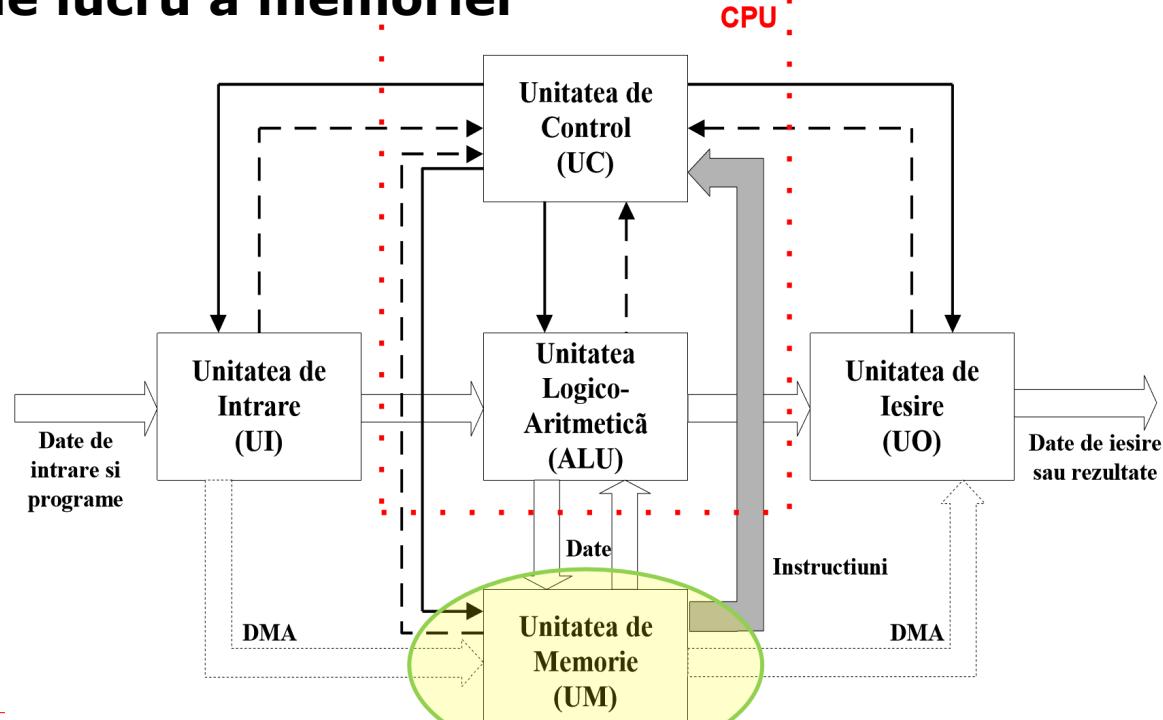


Unitatea de Memorie - UM

UM se caracterizează prin:

- **Capacitate:** volumul de date ce sunt memorate (biți, bytes sau cuvinte). Unități de măsură: kilo(2¹⁰), mega(2²⁰), giga(2³⁰), tera(2⁴⁰).
- **Timpul de acces:** intervalul de timp între momentul când se dă comanda de citire și momentul în care data este disponibilă.

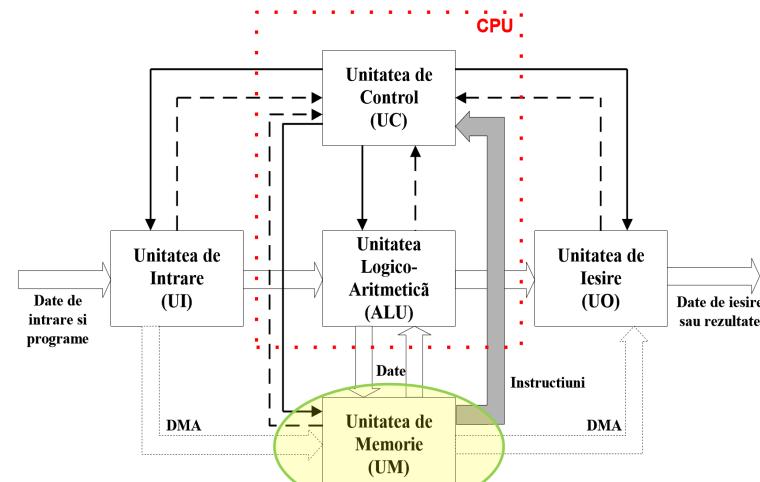
Definește viteza de lucru a memoriei



Unitatea de Memorie - UM

Caracteristicile unei memorii sunt:

- **Performanță:** timpul de acces, rata de transfer, etc.;
- **Tipul fizic** (tehnologia de realizare): semiconductoare, magnetic, optic, magneto-optic, etc.;
- **Unitatea de transfer:** octetul, cuvântul, blocul;
- **Capacitatea:** dimensiunea cuvântului, numărul de cuvinte;
- **Amplasarea:** internă (principală), externă (secundară);
- **Modul de acces:** secvențial, direct, aleator (random), asociativ;
- **Organizarea;**
- **Caracteristici fizice:** volatilă/nonvolatilă, care se poate sterge/ care nu se poate sterge, etc.



Unitatea de Memorie - UM

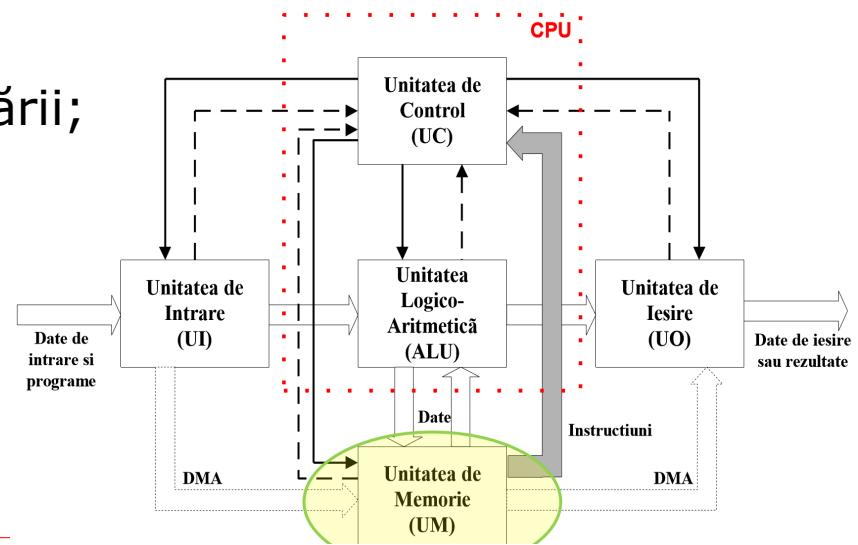
Memoria este organizată ca o structură ierarhică pe nivele determinate de apropierea de CPU;

Principalele **nivele ierarhice** de memorie sunt:

- **memoria principală** (memoria primară sau memoria internă);
- **memoria secundară** (memoria externă).

Aceste nivale se **diferențiază** prin:

- tehnologia în care sunt făcute;
- viteza de lucru;
- capacitatea de memorare;
- modul de administrare a funcționării;
- numărul de subnivele.



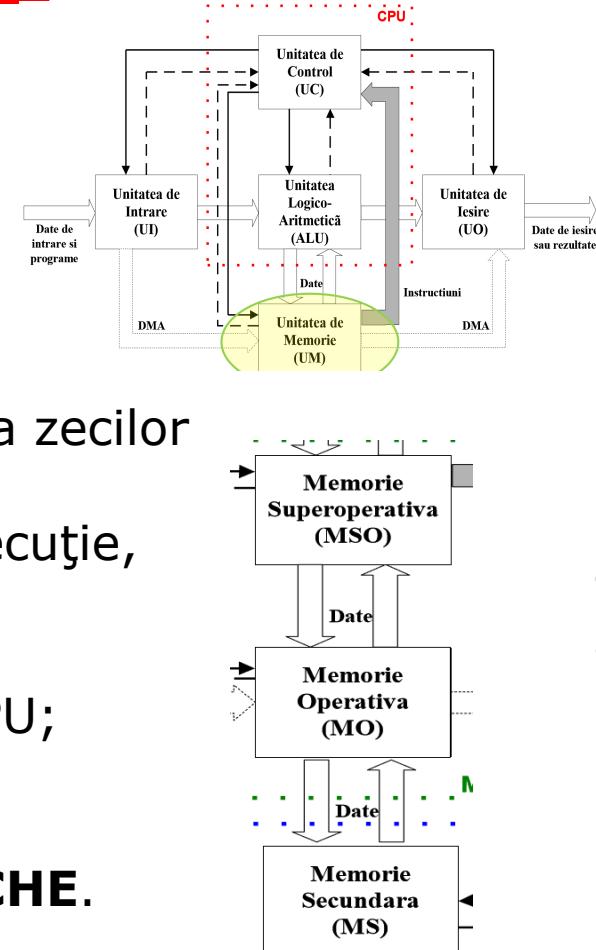
Unitatea de Memorie – Memoria Principală

Memoria internă;

- 2 nivale:

Memoria superoperativă (MSO):

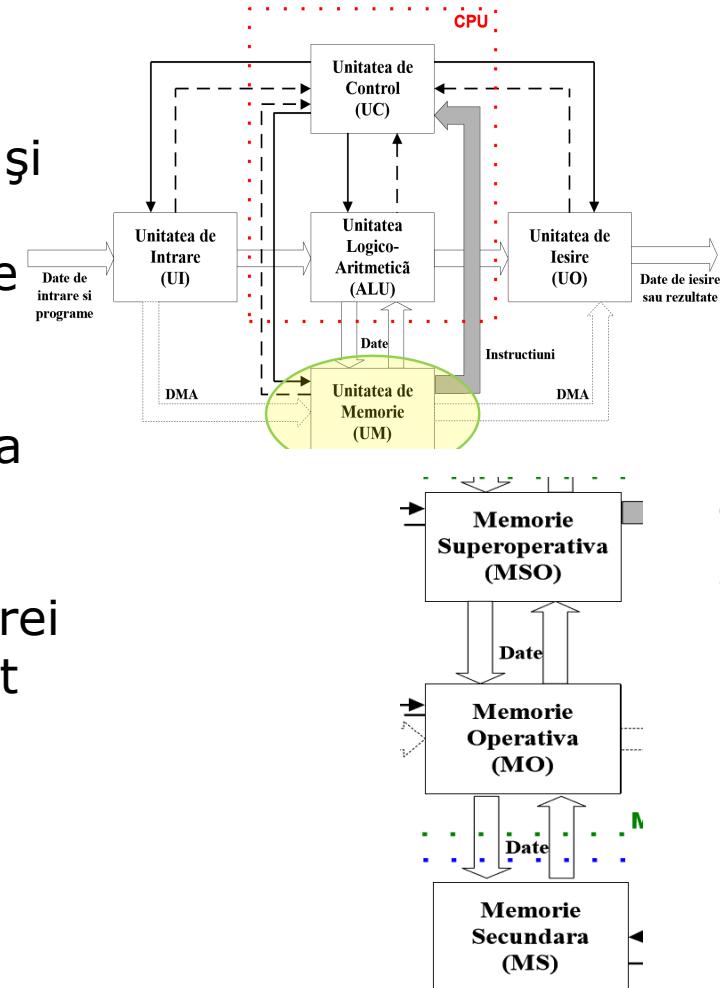
- cea mai apropiată de procesor;
- capacitate mică;
- timp de acces foarte mic (de ordinul ns sau a zecilor de ns);
- stochează instrucțiunile curente aflate în execuție, rezultatele intermediare și datele aferente programului în execuție;
- viteza de lucru este comparabilă cu cea a CPU;
- resursă scumpă a calculatorului;
- în prezent CPU încorporează MSO;
- cea mai răspândită MSO este **memoria CACHE**.



Unitatea de Memorie – Memoria Principală

Memoria operativă (MO):

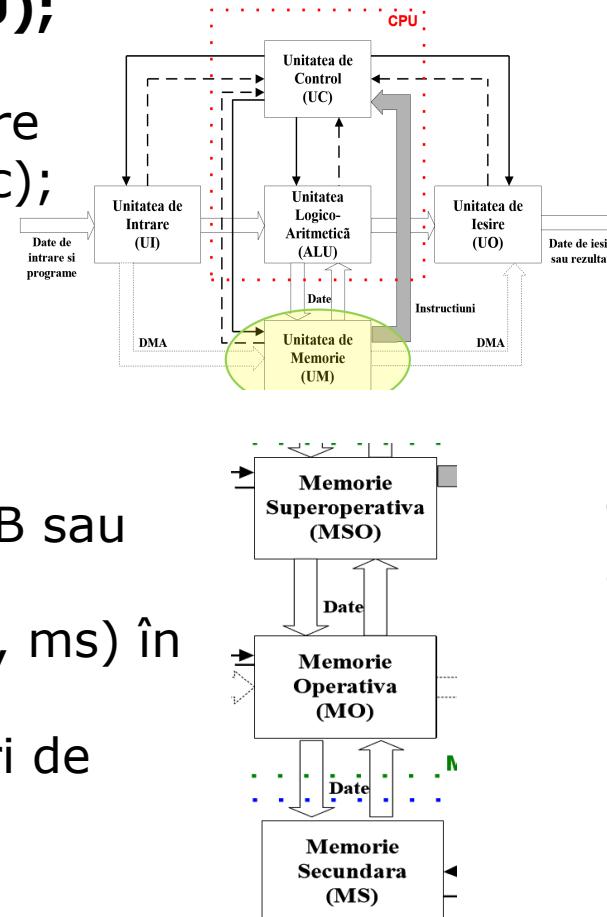
- conține programul (sau programele în cazul multiprogramării) care se execută și datele aferente;
- capacitatea MO (de ordinul MB) este mai mare în comparație cu MSO;
- timpul de acces este mai mare;
- funcționarea MO se bazează pe tehnica accesului aleator (Random Access Technique);
- timpul necesar pentru accesarea oricărei locații de memorie este identic indiferent de adresa;
- în prezent este realizată folosind tehnologia semiconductoare (memoriile RAM, SRAM, DRAM).

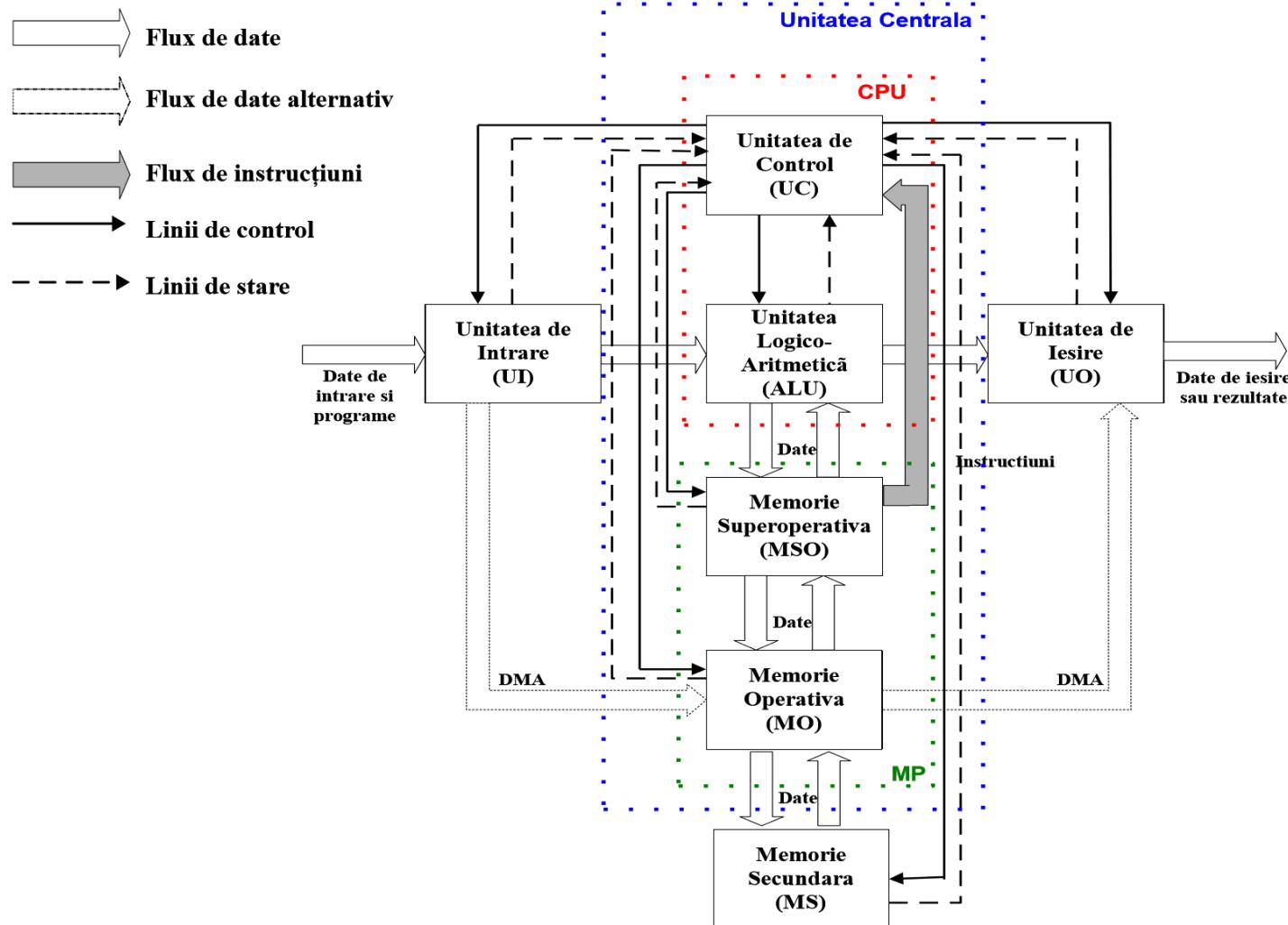


Unitatea de Memorie – Memoria Secundară

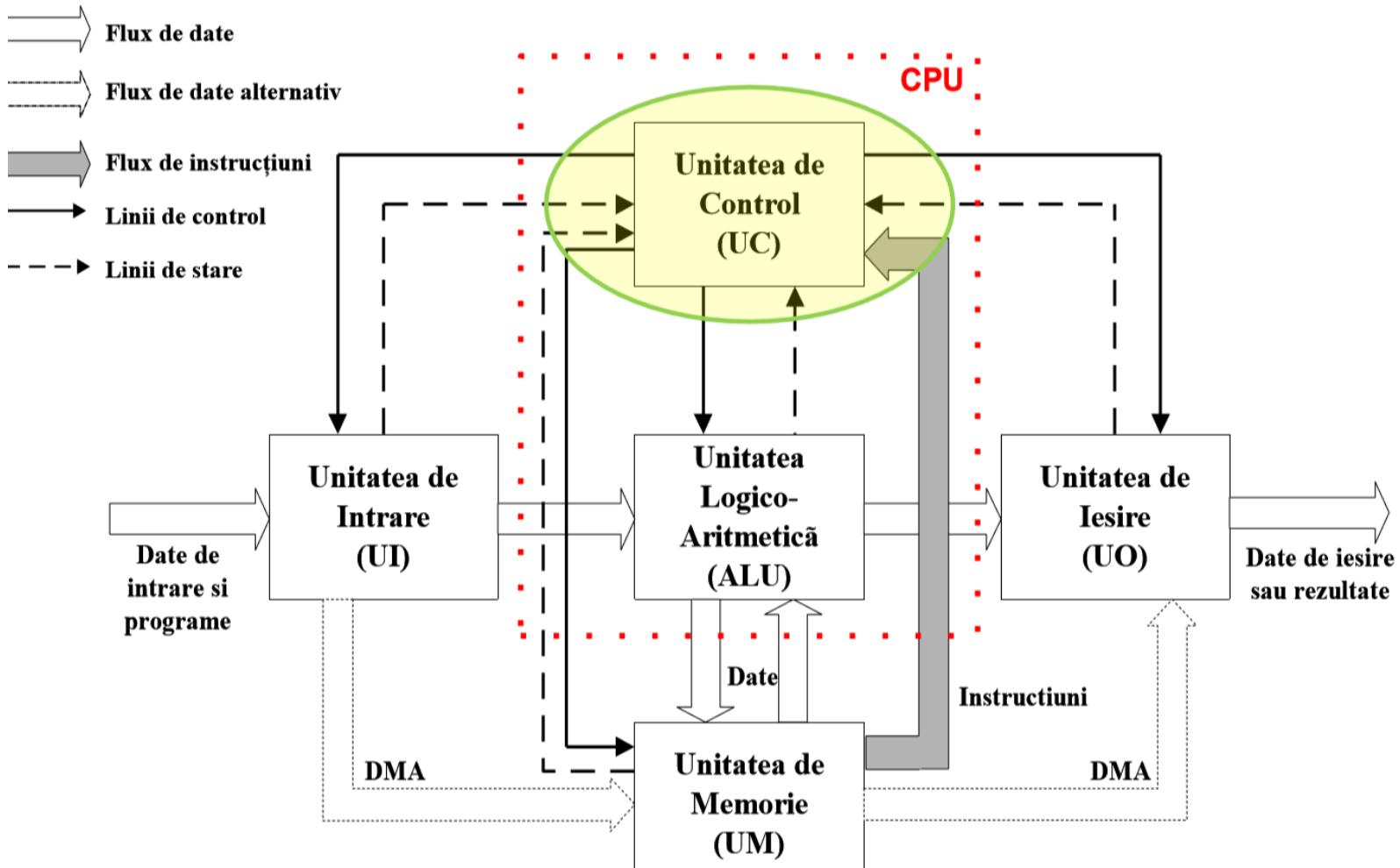
Memoria externă (nu face parte din CPU);

- Unitatea centrală = CPU + MP;
- MS este alcătuită din dispozitive de stocare externe (HDD, FDD, CD-ROM, DVD-ROM, etc);
- Aceste tipuri de MS diferă prin:
 - principiul de funcționare;
 - mediul de stocare;
 - capacitate;
 - viteză de funcționare, etc.
- Capacitate foarte mare (de ordinul MB, GB sau TB);
- Timp de acces foarte mare (de ordinul μ s, ms) în comparație cu MP;
- Destinată pentru stocarea de volume mari de programe și date.



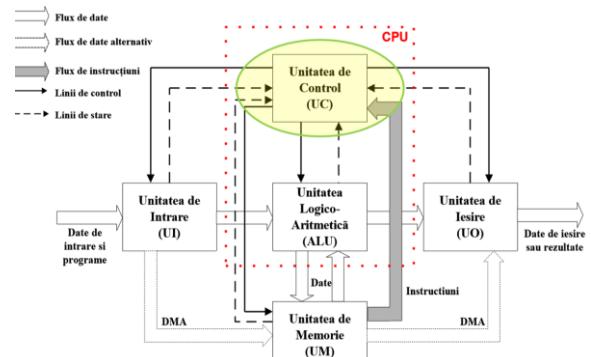


Structura de bază a calculatorului



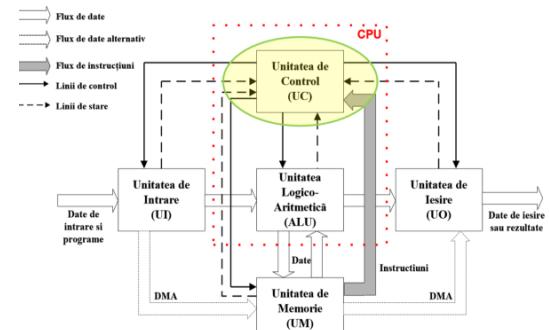
Unitatea de control

- “creierul” calculatorului;
- Rol de: – a superviza;
 - a monitoriza;
 - administra funcționarea tuturor unităților din calculator, inclusiv propria funcționare.
- UC asigură automatizarea completă a procesului de calcul;
- Parte componentă a CPU;
- Alcătuită din registre, numărătoare, divizoare de frecvență, decodificatoare, etc.;
- Din punct de vedere conceptual UC-urile sunt de două feluri: – cablate;
 - microprogramate.



Unitatea de control

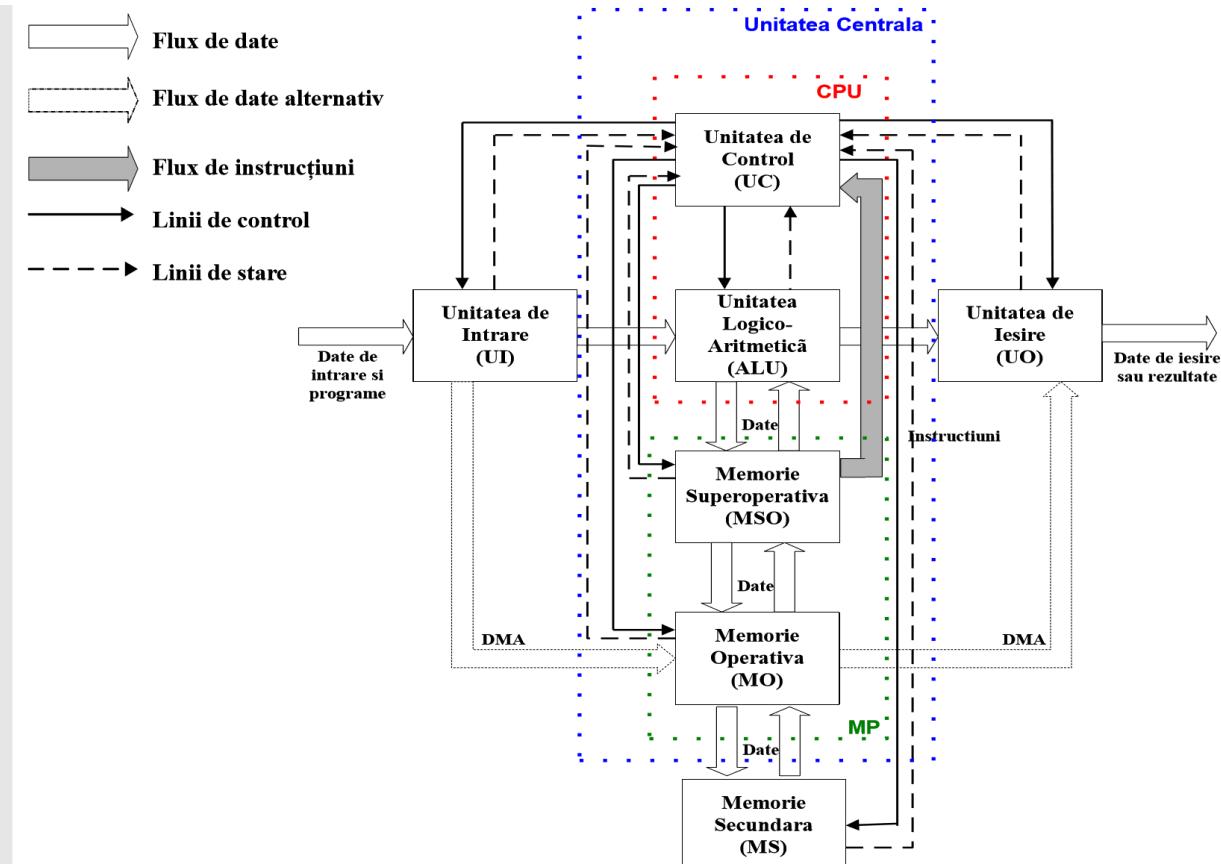
- UC stabilește funcția de bază curentă determinată din instrucțiunea curentă aflată în execuție și din informația de stare primită de la subunitățile implicate în execuție.
 - UC-ul implementează mecanismul fundamental de execuție a unei instrucțiuni:
 - citește instrucțiunea curentă din memorie (fie din MO, fie din MSO);
 - o decodifică (adică o interpretează) pentru a decide care este funcția de executat, după care;
 - activează semnalele de comandă necesare pentru realizarea funcției și apoi; – execută funcția.



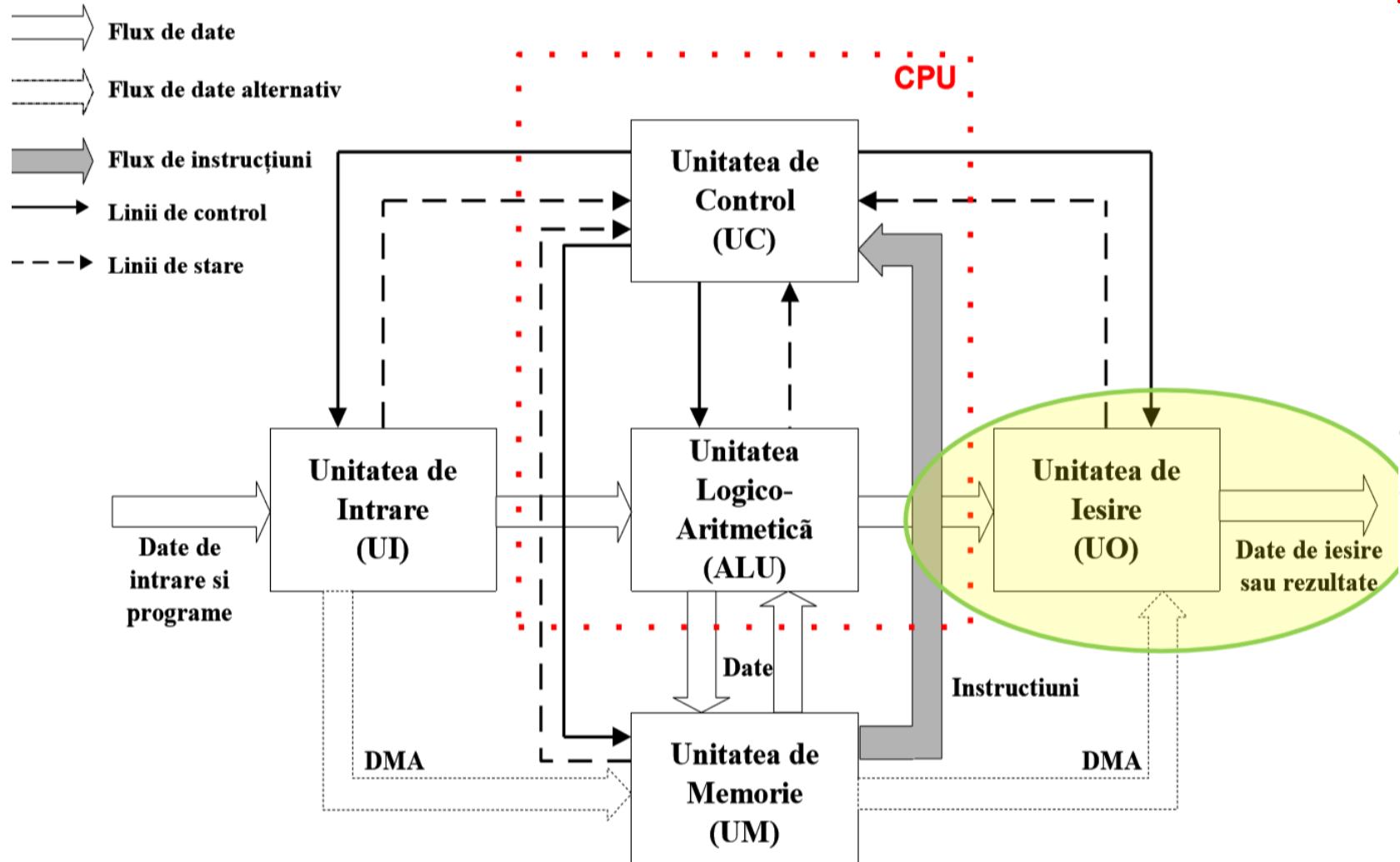
CURS 4 - continuare

Dacă un program din MS trebuie executat, acesta mai întâi se transferă în MP. Acest transfer se face controlat software printr-o componentă a software-ului de sistem numit **Sistem de Operare**

Tinând cont de ierarhizarea memoriei, structura de bază a unui calc. digital devine:

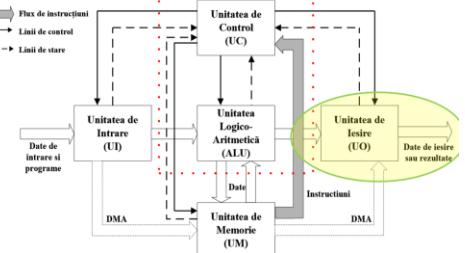


STRUCTURA DE BAZĂ A CALCULATORULUI



UNITATEA DE IEȘIRE (UO)

- Asigură comunicarea între calculator și mediul exterior;
- Extrage rezultatele din calculator;
- Două moduri de lucru:
 - **direct:** prin intermediul display-ului sau monitorului, imprimantei, plotter, etc.
 - **indirect:** atunci când se folosește un mediu intermedian cum ar fi benzile de hârtie perforată, cartelele, etc. În acest caz se vor folosi dispozitive specializate de tipul perforatoarelor de cartele sau benzi, etc.
- Unitățile de intrare și de ieșire se numesc global unități de I/O;
- Ele constituie echipamentele periferice ale unui calculator;
- Funcție de bază a unităților I/O: comunicarea la distanță care include echipamente specializate pentru constituirea de rețele (modem, routere, bridge, etc).



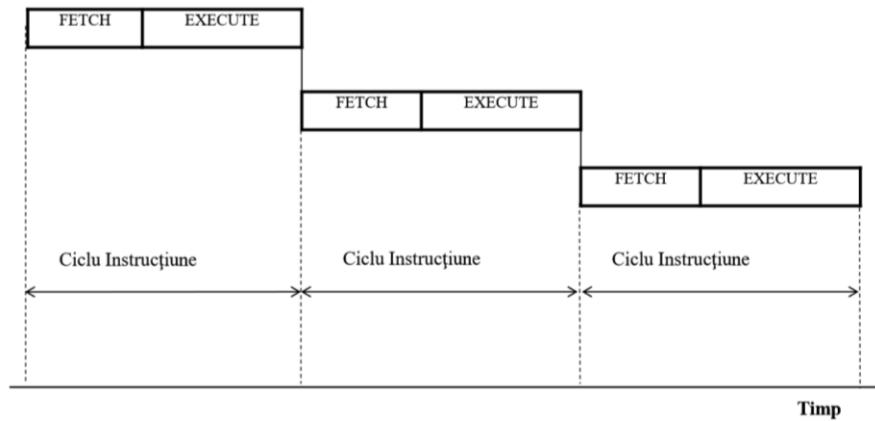
Ciclul instrucțiune

Definiție: Ansamblul de operații legate de realizarea unei instrucțiuni se numește **ciclu instrucțiune**

- Programul este stocat în memoria calculatorului;
- Programul este format dintr-un set de instrucțiuni;
- Aceste instrucțiuni sunt scrise în limbaj cod mașină, iar programul se numește program în limbaj cod mașină.
- Conform principiului von Neumann, programul este executat secvențial, instrucțiune după instrucțiune de către CPU.

Ciclul instrucțiune

- Pentru execuția unei instrucțiuni trebuie realizate următoarele faze:
 - **FETCH**: citirea instrucțiunii în limbaj cod mașină din memorie și transferarea ei în CPU;
 - **EXECUTE**: executarea instrucțiunii de către unitățile funcționale ale calculatorului (în special de ALU), adică generarea rezultatului.
- Aceste faze se execută repetat în timpul rulării unui program;
- O secvență FETCH-EXECUTE pentru o instrucțiune formează **ciclul instrucțiune**.



FAZA FETCH

□ Cuprinde următoarele subfaze:

1. CPU citește din memorie de la adresa specificată în registrul numărător de program PC instrucțiunea ce va fi executată și o încarcă într-un registru din UC numit Registrul de Instrucțiune (RI).
2. În UC instrucțiunea se împarte în două câmpuri majore:
 - câmpul operație (OPCODE), pe care îl dirijează în Registrul de Funcție (RF);
 - câmpul de adresă (ADRESA), pe care îl stochează în Registrul de Adresă (RA).

Instrucțiune:

0 1.....(L-1)	0 1.....(n-1)
OPCODE	ADRESA

FAZA FETCH

3. UC decodifică OPCODE sau conținutul Registrului de Funcție; => interpretarea OPCODE de către UC;
4. UC incrementează numărătorul de program PC pentru a specifica adresa următoarei instrucțiuni de executat;

Program Counter (PC) ⇔ Instruction Pointer (IP)

Faza FETCH este obligatorie pentru orice instrucțiune, ea asigură extragerea instrucțiunii curente din memorie și decodificare acesteia;

După generarea adresei următoarei instrucțiuni faza FETCH se încheie și urmează faza EXECUTE.

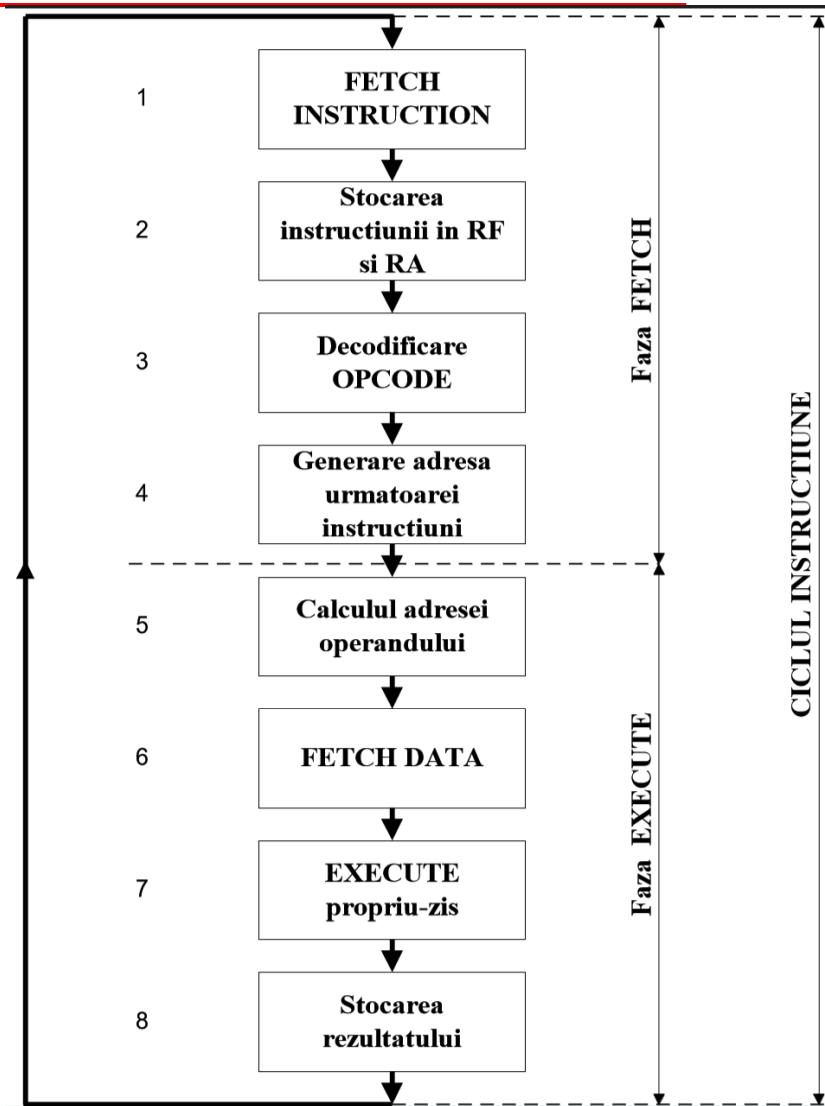


FAZA EXECUTE

Faza EXECUTE este formată din următoarele subfaze sau pași:

5. Folosind informația din RA, UC trebuie să determine adresa efectivă a operandului(sau ale operanzilor);
6. CPU citește data (datele) din memorie de la adresa (adresele) calculată(e) ca adresă(e) efectivă(e) și o (le) extrage ca fiind operand (operanze). => FETCH DATA sau FETCH OPERAND. Data este transferată în ALU.
7. UC comandă ALU să efectueze funcția sau operația specificată de OPCODE, prin procesarea operanzilor și determinarea rezultatului. => faza de EXECUTE propriu-zisă.
8. Rezultatul este memorat fie în memorie fie în registre speciale ale CPU.

CICLUL INSTRUCȚIUNE: faze și subfaze



CICLUL INSTRUCȚIUNE: faze și subfaze

Unele subfaze din ciclul instrucțiune pot lipsi pentru anumite instrucțiuni.

- Pentru instrucțiunile fără operanzi lipsesc pașii 5 și 6. În cazul instrucțiunilor cu operanzi multiplii pașii 5 și 6 se repetă.
- Pentru instrucțiunile de salt nu sunt necesari pașii 4, 5, 8.
- De asemenea în cazul instrucțiunilor de comparare, rezultatul nu trebuie memorat, deci pasul 8 lipsește.
- Întotdeauna faza FETCH este unică, în timp ce faza EXECUTE prezintă aspecte diverse, în funcție de instrucțiunea concretă ce trebuie executată. De aceea faza EXECUTE poate fi realizată în diferite aranjamente ale pașilor săi specifice clasei instrucțiune.

CICLUL INSTRUCȚIUNE: faze și subfaze

- Un calculator digital trebuie să parcurgă mii, milioane sau miliarde de ori de cicluri instrucțiune pentru a prelucra un program.
- Un ciclu instrucțiune se execută din punct de vedere al unui CPU într-un ciclu mașină. Acest ciclu se măsoară în μ s sau ns.
- Viteza de procesare se măsoară în KHz, MHz sau GHz.
- În cazul calculatoarelor moderne viteza se măsoară în: – numărul instrucțiuni pe secundă (IPS), => MIPS; – numărul de operații în virgulă mobilă (FLOP), => KFLOP sau MFLOP.

