

ARHITECTURA CALCULATORARELOR

Informatica , an I, 2021-2022

Curs 6

MICROPROCESORUL

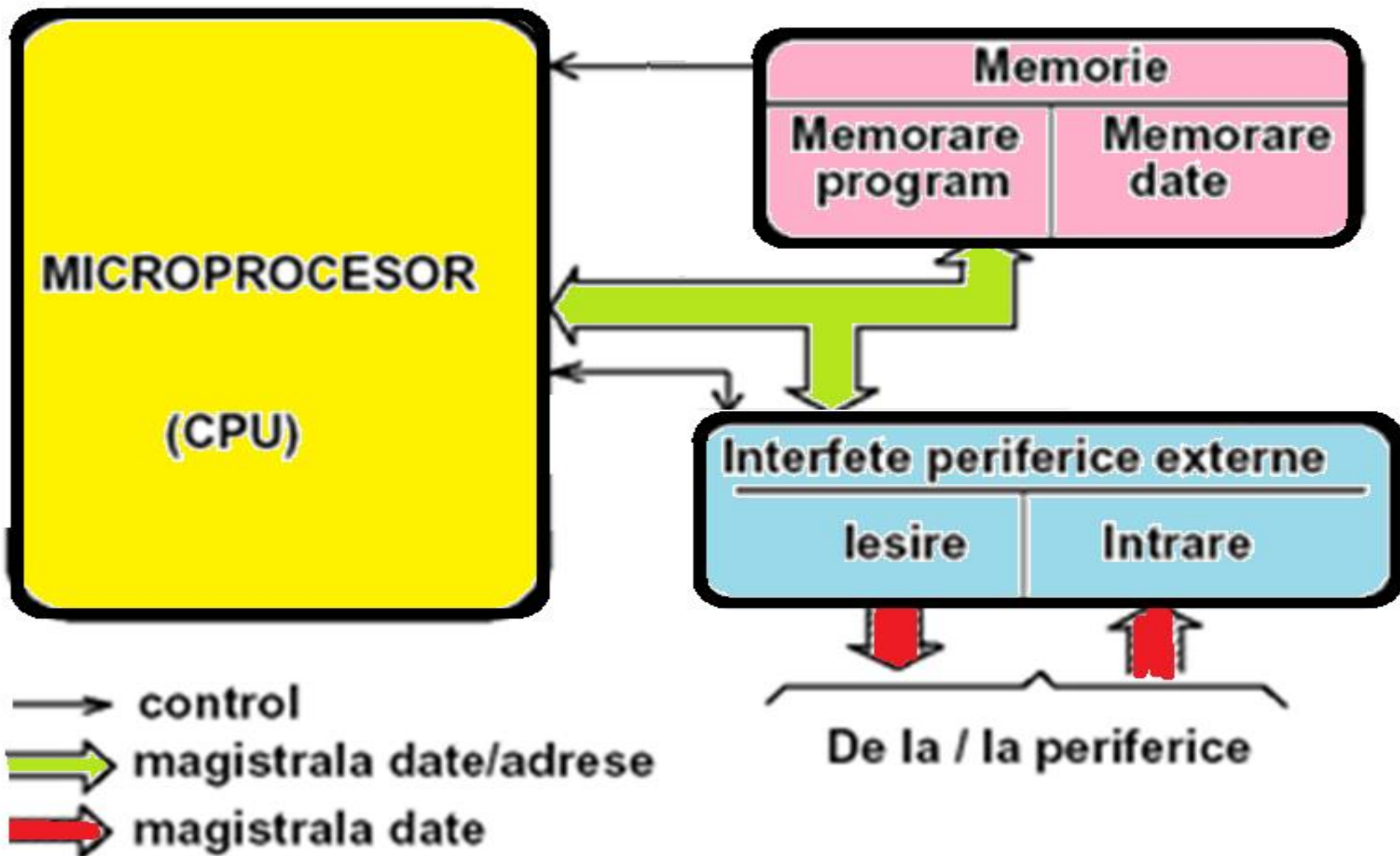
MICROPROCESOR ?

MICROPROCESÓR, *microprocesoare*, s. n. Circuit integrat de mare densitate și complexitate, specializat pentru operații de calcul, comandă și control într-un computer sau alt sistem electronic. – din engl. **microprocessor**.

Sursa: [DEX '09 \(2009\)](#)

MICROPROCESÓR (< engl.) s. n. Unitate centrală de prelucrare (CPU) a unui calculator realizată adesea pe o singură plăcuță de siliciu sau alt material semiconductor (chip), care în prezent poate avea peste un milion de tranzistoare.

Sursa: [DE \(1993-2009\)](#)

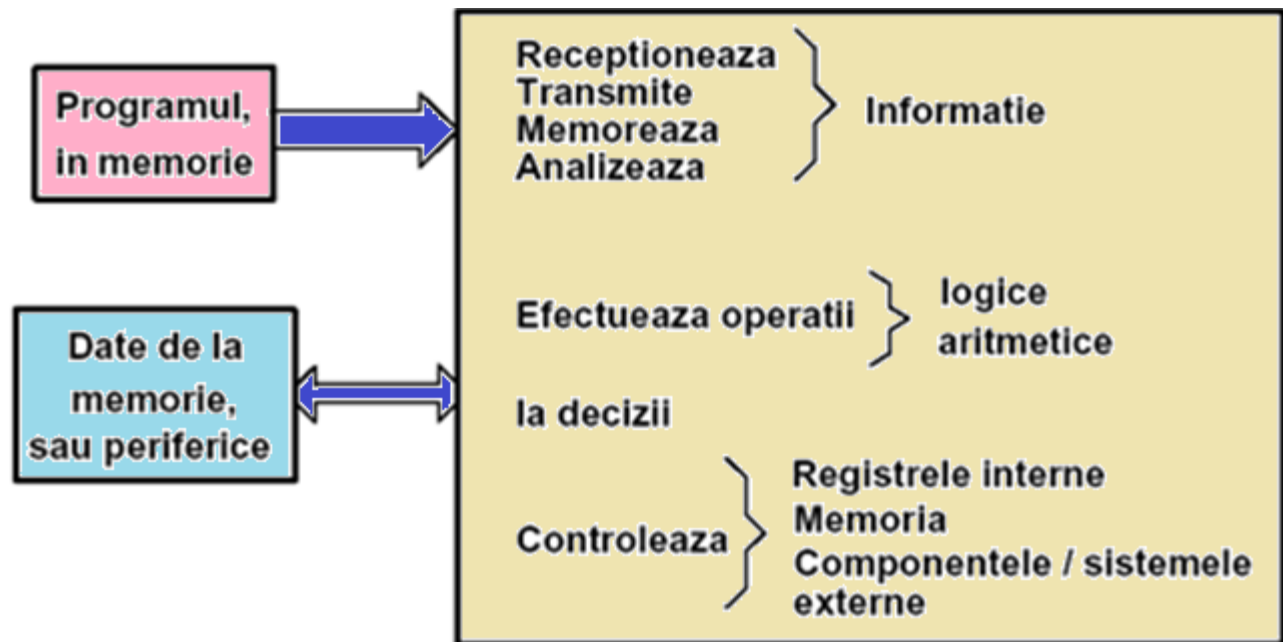


Funcțiile unui microprocesor

-in corelatie cu memoria și perifericele sistemului-

Impartire functii pe grupe funcționale:

- 1) manipulare informații (instrucțiuni, date transmise, date primite);
- 2) execuție operații de calcul;
- 3) control și supervizare:
 - componente din sistem;
 - operații efectuate.



Funcțiile de baza ale unui microprocesor

- 1) funcția de intrare (*INPUT*): interfata între lumea exterioară și sistem;
- 2) funcția de ieșire (*OUTPUT*): interfata între sistem și lumea exterioară;
- 3) funcția de memorare (*MEMORY*): păstrarea instrucțiunilor programului, a datelor de intrare, a datelor intermediare precum și a rezultatelor (date finale/de ieșire);
- 4) funcția aritmetico-logica: efectuarea operațiilor aritmetice și logice;
- 5) funcția de *CONTROL*: acțiunile de secvențializare și control ale întregii activități a sistemului de calcul

Structura funcționala și fluxul de informații dintr-un sistem microprocesor

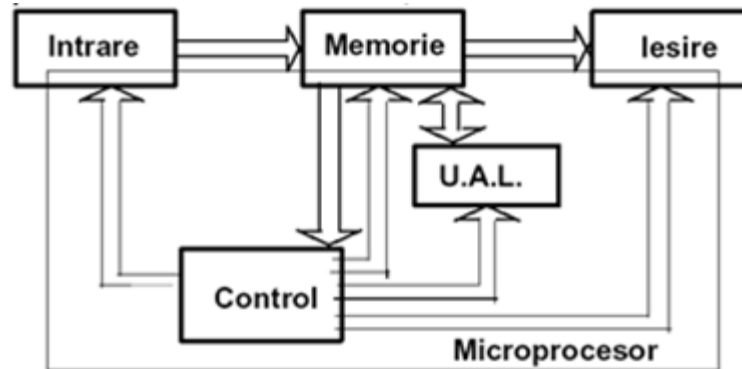
1) informația este adusă, prin funcția de intrare, (*INPUT*), sub coordonarea funcției *CONTROL*, în memoria sistemului. Informația poate fi *instrucțiuni* sau *date* pentru prelucrare (operații aritmetice sau logice).

2) Din memorie, tot sub coordonarea funcției de *CONTROL*, *informația este citită și decodificată* de către microprocesor, executându-se secvențial instrucțiunile programului. Datele și rezultatele sunt vehiculate *procesor <-> memorie*, sub comanda funcției de *CONTROL*.

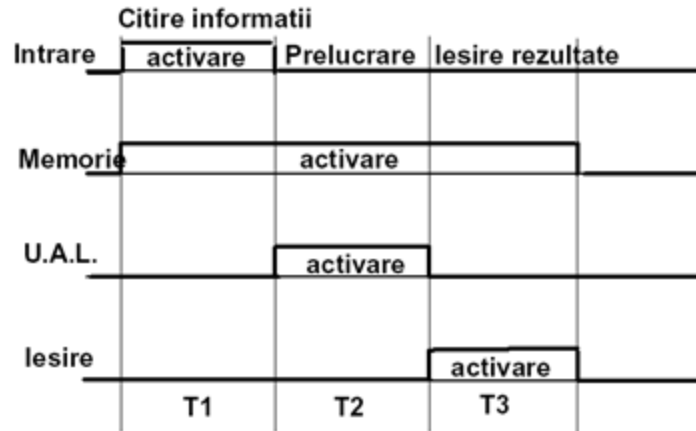
3) La final, coordonat de către funcția de *CONTROL*, se activează, pentru transferul rezultatelor spre ieșirea sistemului, funcția de ieșire (*OUTPUT*).

4) *Blocul ALU (UAL)*, efectueaza operatii aritmetice si operații logice.

4) *Secțiunea de CONTROL* generează și coordonează toate semnalele de control necesare pentru a executa operațiile de calcul din *ALU*, și pentru sincronizarea transferului de date in sistem.



Sucesiunea si durata operatiilor este controlată de către secțiunea CONTROL



- în intervalul $T1$, informația (instrucțiuni și date), se preia de la intrare și se memorează (activare secțiuni *INPUT* și *MEMORY*).
- în $T2$ instrucțiunile din memorie sunt preluate și executate cu ajutorul ALU. Rezultatele sunt duse apoi în memorie.
- În $T3$ se transferă rezultatele spre exteriorul sistemului, activându-se funcția de *OUTPUT*.

La baza sincronizării se afla semnalul unui ceas de sistem (controlat printr-un cristal de cuarț).

Reprezentarea informației

La baza reprezentarii informatiei se afla **cuvinte de instrucțiuni** respectiv **cuvinte de date** (*vezi lungimea cuvântului unui procesor*)

Fiecare *instrucțiune* din programul microprocesorului va fi formată din unul sau mai multe *cuvinte de instrucțiune*.

Cuvântul de instrucțiune **este alcatuit din:**

- **codul operației** : operația de executat;
- **operandii** : indică asupra cui se va efectua operația conținută în codul operației.



Memorarea informației

Instrucțiunile sunt *memorate în ordinea executării*, iar microprocesorul le *citește și le execută secvențial*:

Pas 1: citire un cuvânt de instrucțiune

Pas 2: decodificare adica obtinere operanzi si adresa lor

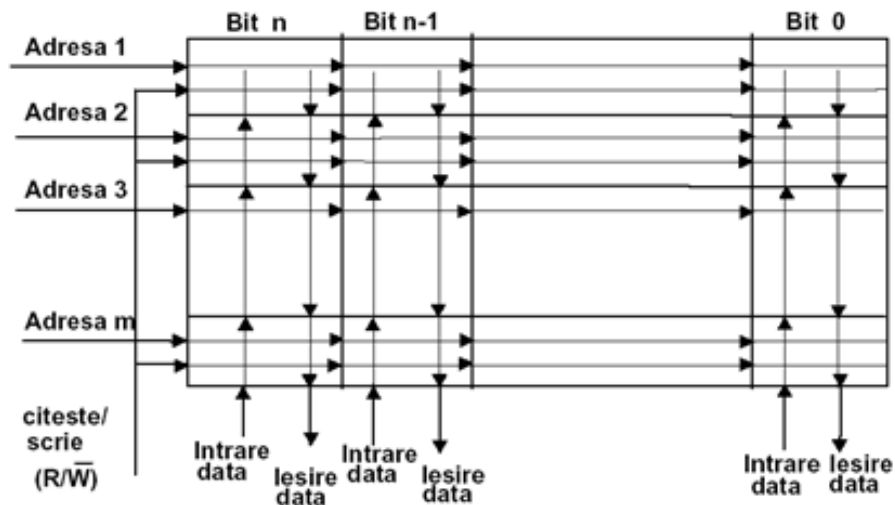
Pas 3: se *execută* operația indicată.

Pas 4: Procesul se reia de la pasul 1 cu instructiunea urmatoare

Fiecare cuvânt de instrucțiune sau date are asociat, in mod unic, un număr care reprezintă *adresa* sa.

Memoria sistemului este organizata sub forma unei matrice, cu liniile având un număr de celule (biți), egal cu cel al cuvântului de instrucțiune sau de date - fiecărei linii îi corespunde, o adresă unica -

Exemplu: organizare memorie pe cuvinte de n biți (n = numărul de biți din cuvântul de date sau instrucțiune al microprocesorului - 8, 16, 32, 64).



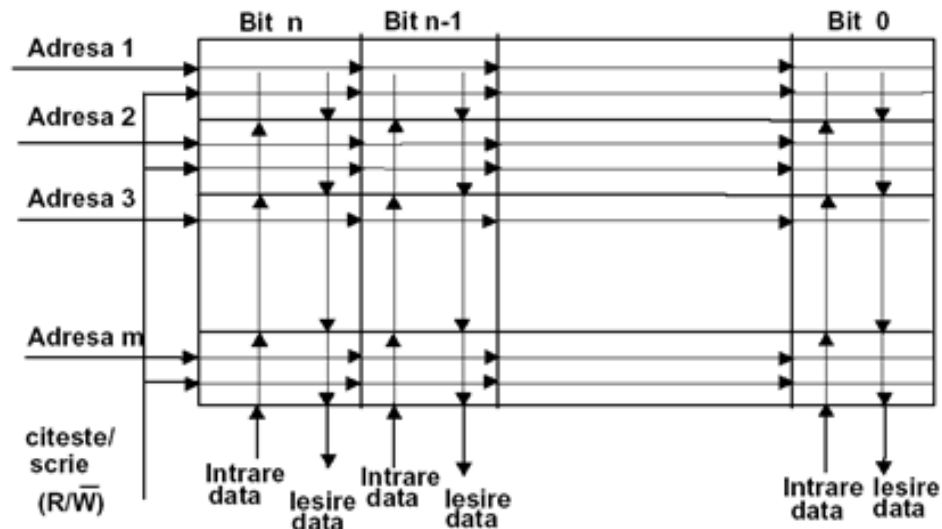
Memorarea informației –continuare-

O operație cu memoria de la o anumită adresă va determina activarea tuturor celulelor de memorie elementare de la adresa respectivă (de pe linia cu aceeași adresa).

În funcție de tipul semnalului (R/W (citire/scriere)), secțiunea de *CONTROL* va indica operația ce se execută, adică *sensul fluxului de informație (microprocesor ->memorie, pentru operație de scriere, sau memorie ->microprocesor, pentru o operație de citire)*. Transferul se va efectua pe toate cele n linii de date simultan.

Memorii **ROM** și **RAM**.

Microprocesorul poate conține pe cip memorie *ROM* (programe) și memorie *RAM* (de uz general, sau, sub denumirea de registre, celule cu destinație specială în memorarea, decodificarea și executarea instrucțiunilor, a operațiilor aritmetice și logice etc.).



Conceptul de program

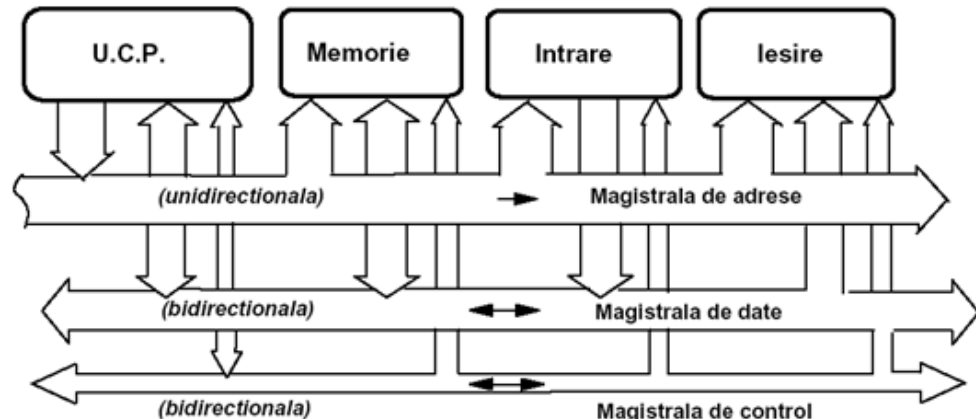
Conceptul de *program* se referă la *operarea secvențială, reprezentarea și memorarea informațiilor*.

Executarea unei secvențe de operații, de către microprocesor, este obținută prin implementarea unei succesiuni de instrucțiuni, memorate în memoria sistemului.

Fiecare operație este implementată printr-o secvență de una sau mai multe instrucțiuni.

Conceptul de magistrală

- *Magistralele* reprezintă legăturile dintre părțile funcționale ale microprocesorului.
- Tipuri de magistrale într-un microprocesor:
 - *magistrale interne* (canalele de legătura între diversele unități funcționale din unitatea centrala (CPU) a sistemului);
 - *magistrale externe* (canalele de comunicație între CPU și componentele externe acestuia).
- Fiecare grup de magistrale poate fi subdivizat în trei categorii, in functie de tipul informației transferate:
 - *magistrale de adrese*;
 - *magistrale de date*;
 - *magistrale de control*.

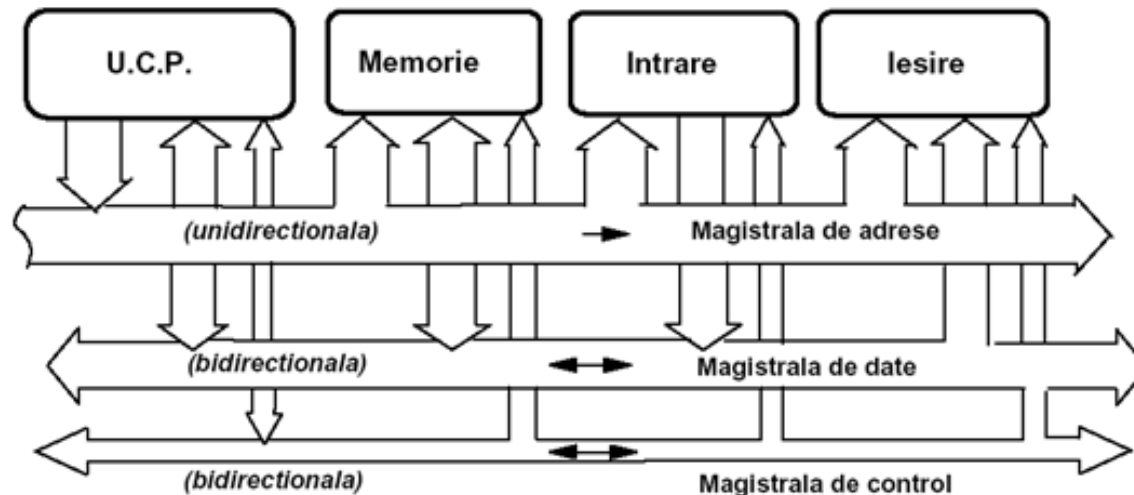


Conceptul de magistrală-continuare-

În funcție de cum se efectuează transferul unui cuvânt de informație avem:

- **Magistrale paralele**, ce permit transferul integral și simultan al unui cuvânt de informație
- **Magistrale seriale**, pentru care datele sunt transmise multiplexat în timp (un cuvânt este transmis bit cu bit, pe aceeași linie) (mai lente decât magistralele paralele)

Nota: Practic, toate magistralele unui *CPU* (interne și externe) sunt organizate ca și magistrale paralele. Magistralele seriale se utilizează pentru transmiterea datelor la / de la echipamente aflate la distanță de *CPU* (terminale, linii telefonice, etc.).



Conceptul de magistrală-continuare- *Magistrala de date*

- Magistrala de date este destinată:
 - transferului *unidirecțional* de instrucțiuni de la memorie
 - transferului *bidirecțional*, de date între memorie, unitatea centrală și interfețele de intrare/ieșire.
- Direcția transferului de informație este controlată și coordonată de către secțiunea de *CONTROL* a microprocesorului, prin generarea de semnale specifice (citire sau scriere).

Desigur va mai amintiți de cele două tipuri de arhitecturi (Neumann și Harvard)....



Conceptul de magistrală-continuare- *Magistrala de date*

Desigur va mai amintiti de cele doua tipuri de arhitecturi (Neumann si Harvard)....?

Raspunsul corect =>

- La structurile de tip *von Neuman*, există o singură magistrală de date, pe care se vehiculează, *la momente de timp diferite*, atât cuvintele de program (coduri de instrucțiune), cât și cuvintele de date.
- La structurile de tip *Harvard*, există magistrale independente pentru transferul cuvintelor de instrucțiune, respectiv a datelor.

Conceptul de magistrală-continuare-

Magistrala de adrese

- Magistrala de adrese este o magistrală unidirecțională. Prin ea circula codul binar reprezentând adresa de memorie a datei ce se va utiliza în cadrul operației ce se execută.
- **Adresabilitatea unui microprocesor** este data de numărul de biți ai magistralei de adrese. Un număr de 65536 (64 KBytes) celule de memorie necesita 16 linii de adresa pe această magistrală. ($2^{16}=65536$).

Si iarasi la arhitecturile de tip von Neumann si Harvard...

- *von Neuman* au o singura magistrală de adrese, vehiculând la momente de timp diferite, adresa de unde se citește cuvântul de instrucțiune, respectiv adresa la / de la care se efectuează transferul datelor la execuția instrucțiunii.
- *Harvard* au două magistrale de adrese:
 - adresarea memoriei de program/instructiuni;
 - adresarea memoriei de date

ADICA *Harvard* permite ca în paralel cu execuția unei instrucțiuni să se poată adresa și citi cuvântul de instrucțiune următor.

Conceptul de magistrală-continuare- *Magistrala de control*

- Magistrala de control este folosita pentru vehicularea informațiilor legate de operația in curs de executie.

Numărul de semnale de pe aceasta magistrală depinde de numărul de semnale de control necesare pentru microprocesorul utilizat.

Cele mai uzuale semnale de pe magistrala de control sunt:

- *semnalul de ceas* al sistemului (asigură funcționarea secvențiala cu o periodicitate fixă a întregului CPU);
- *semnalele de citire / scriere în memorie, citire / scriere pentru intrare / ieșire din sistem* etc.

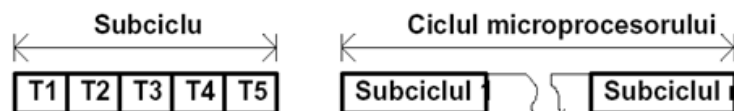
Structura functionala a unui microprocesor

Microprocesorul executa în mod secvențial instructiunile dintr-un program memorat.

- Semnalul de temporizare minimă, în raport cu care se desfășoară toate celelalte temporizări sincronizate din funcționarea microprocesorului, se numește **semnal de ceas** (clock) al sistemului.
- **Ciclul de instrucțiune** al microprocesorului = *secvența de operații efectuate pentru a extrage o instrucțiune din memorie, a o decodifica și executa.*

Ciclul microprocesorului este format din unul sau mai multe *subcicluri*, fiecare la rândul lui durând mai multe *perioade de ceas*.

Un subciclu este utilizat pentru obținerea fiecărui cuvânt de instrucțiune sau date, sau pentru fiecare operație de bază ce trebuie executată.

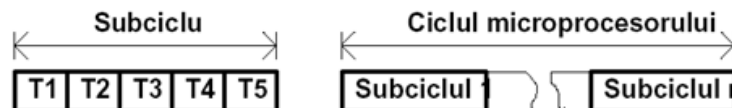


Structura ciclului unui microprocesor.

Structura functionala a unui microprocesor

Ciclul microprocesorului este format din unul sau mai multe *subcicluri*, fiecare la rândul lui durând mai multe *perioade de ceas*.

Un subciclu este utilizat pentru obținerea fiecărui cuvânt de instrucțiune sau date, sau pentru fiecare operație de bază ce trebuie executată.



Structura ciclului unui microprocesor.

registru contor de program (PC = program counter), un registru special folosit pentru a cunoaște în permanență care instrucțiune se execută (sau va fi executată la pasul următor). Acesta conține adresa următoarei instrucțiuni de program care va fi citită din memorie (un *pointer* care arată unde s-a ajuns cu execuția programului la un moment dat).

Primul subciclu al fiecărui ciclu al microprocesorului este utilizat pentru *extragerea de informație din memorie (fetch)* și durează în general trei perioade de ceas:

- T1: conținutul *PC* reprezentând adresa de unde se va citi instrucțiunea curentă, este adus pe magistrala de adrese, selectând, celula de memorie dorită;
- T2: conținutul *PC* este incrementat (pregătire pentru următoarea extragere din memorie)
- T3: conținutul memoriei adresate este adus în microprocesor și memorat temporar;
- în T4 și T5 (prezente sau nu, în funcție de operația ce se execută), se transferă informația între registre sau se efectuează operațiile aritmetice simple.

Structura functionala a unui microprocesor

Note:

- în funcție de procesor și de instrucțiunea executată, un subciclu poate dura și mai multe perioade de ceas;
- întotdeauna, în primul subciclu, procesorul citește un cuvânt considerat ca primul din instrucțiune.
- o instrucțiune poate fi formată din unul, două sau trei cuvinte.
- La execuția *subciclurilor de fetch* (extragere a cuvântului de instrucțiune din memorie) conținutul este transferat pe magistrala de adrese a sistemului. Există un registru intermediar, **registrul de adresă a memoriei (MAR)**, care conține adresa locației de memorie curentă cu care se lucrează. Registrul *MAR* este un registru intern, inaccesibil utilizatorului în mod direct prin program.

Structura functionala a unui microprocesor

Diagrama functionala a unei sectiuni de *CONTROL* :

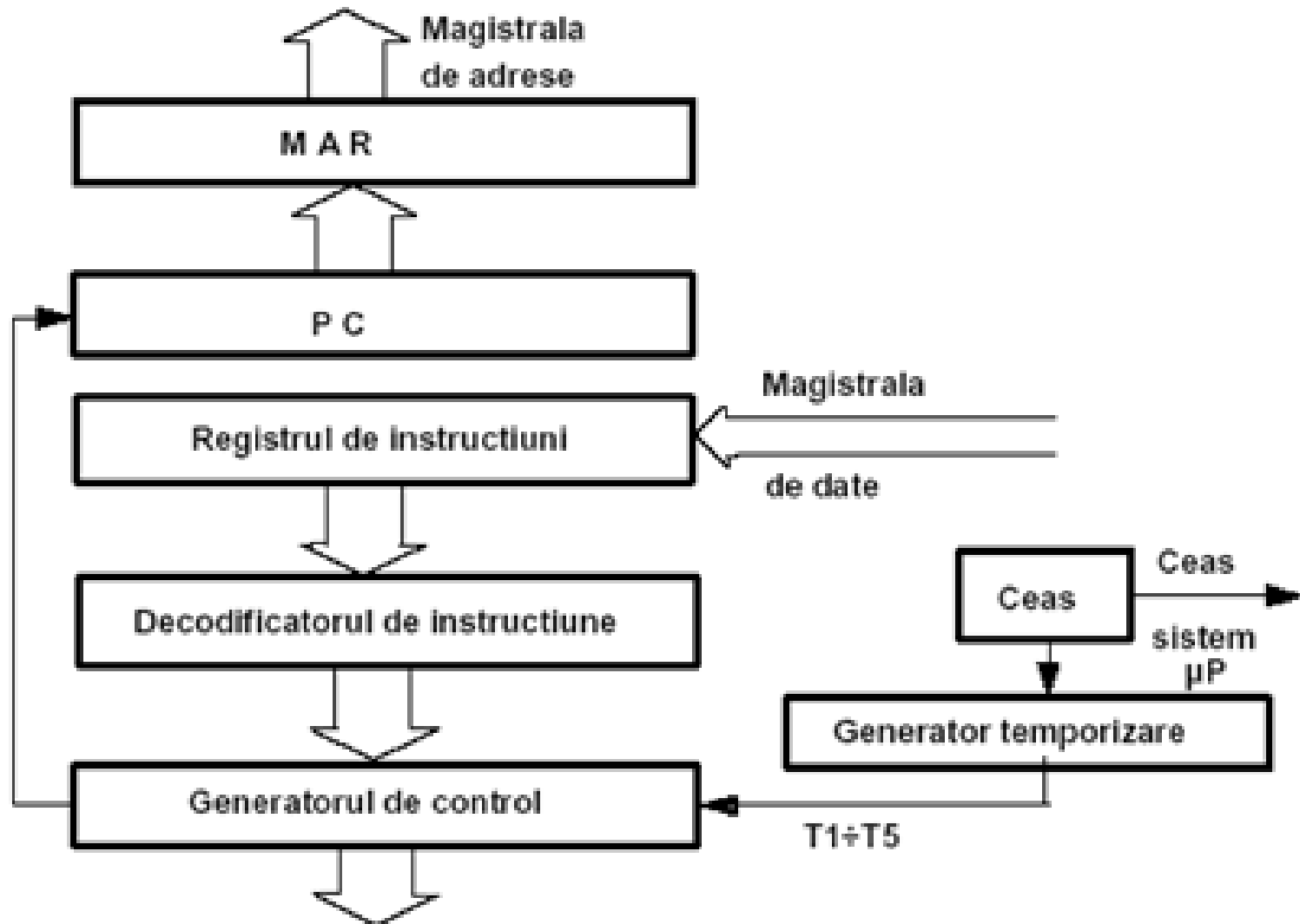


Diagrama functionala a unei sectiuni de **CONTROL**

- Conținutul celulei de memorie adresată la operația de *fetch* este transferat pe magistrala de date, și de acolo este memorat în **registrul de instrucțiune**.

Nota: Unele microprocesoare au implementata o structură numita "pipeline" (conducta), care permite citirea în avans a mai multor cuvinte din memoria program, ceea ce mărește viteza globala de execuție a microprocesorului (în timp ce o instrucțiune se execută, o alta poate fi decodificată, iar o alta poate fi citita)

- Cuvântul de instrucțiune memorat de către registrul de instrucțiune, este decodificat de către **decodificatorul de instrucțiune**. Decodificatorul de instrucțiune convertește apoi codul operației în secvența corespunzătoare a semnalelor de control și temporizare și declanșează execuția propriu-zisă a instrucțiunii (transfer de date, operații aritmetico-logice, etc.).

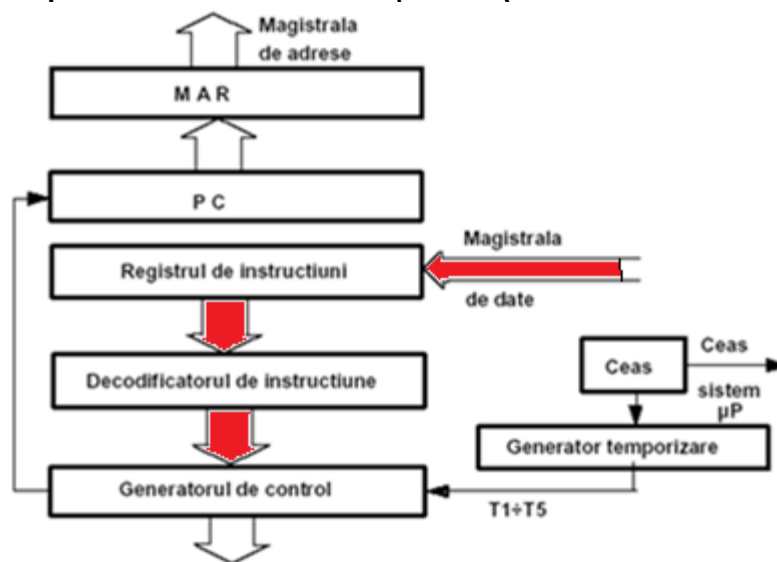
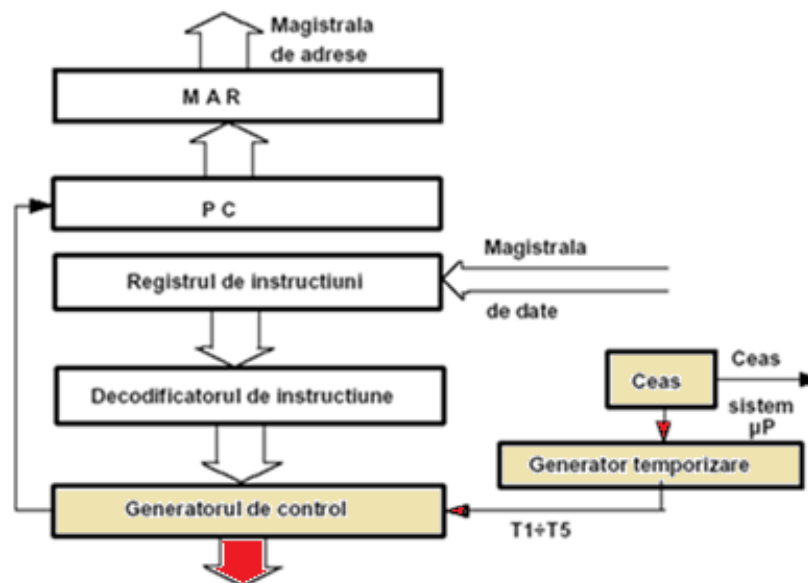


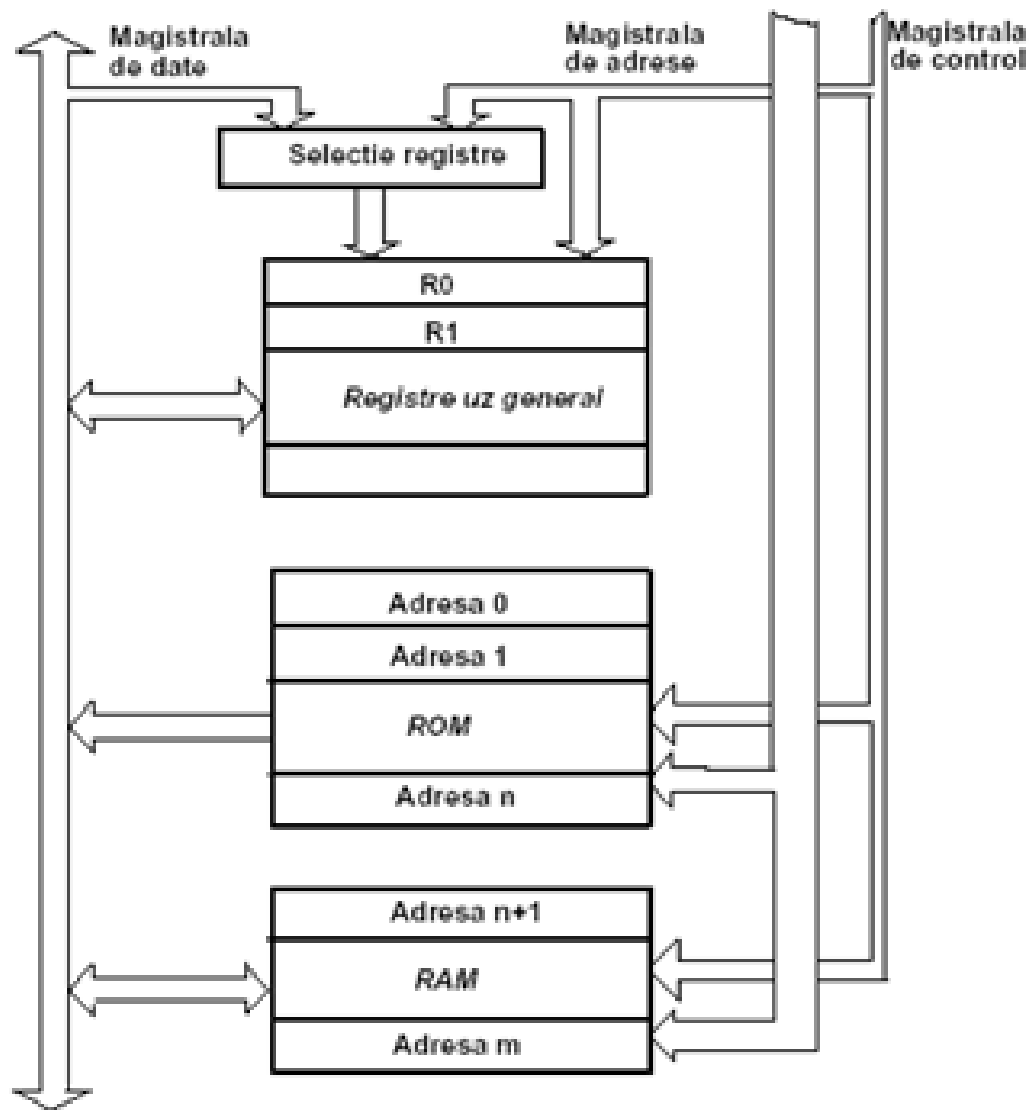
Diagrama functionala a unei sectiuni de **CONTROL**

Implementarea fizica a decodificatorului de instructiune este realizată fie prin integrare echivalentă cu o rețea de porți logice, fie cu o memorie de tip *ROM*

- Semnalele de control ale sistemului sunt obținute în **generatorul de control**. Acesta, pe baza ieșirilor *decodificatorului de instructiuni* și ale *ceasului sistemului*, furnizează semnalele de control pentru întreg sistemul microprocesor.
- Toate tipurile de microprocesoare au un pin special, denumit *RESET*, care, prin activare, aduce sistemul într-o stare inițială unică, permițând lansarea în execuție a primei instrucțiuni a unui program. Semnalul *RESET* determina și unele inițializări interne ale microprocesorului.



Structura principală a memoriei interne a unui microprocesor



- Microprocesorul poate conține:
 - memorie *ROM*
 - memorie *RAM* (cache)
 - registre (fizic tot memorie *RAM*)
- **Registrele** alcătuiesc un element mai deosebit față de memoria RAM din punctul de vedere al programării microprocesorului. Astfel, datele memorate de către registre pot fi manipulate, modificate, operate în *ALU*, prin instrucțiuni specifice de operare cu registrele.

Nota: **instrucțiunile care se execută în registre sunt cele mai rapide. (o operație cu un registru de aprox. 10 ori mai rapidă).**

Tipuri de registre:

- *registre speciale:*
- *registre de uz general.*

Structura principală a memoriei interne a unui microprocesor

Registreele de uz general sunt utilizate pentru manipularea și memorarea temporară a datelor. Acestea sunt adresabile de către utilizator/programator: pentru stocarea și manevrarea datelor programului - operanzi, date, adrese de memorie conținând date utile programului, etc.

Operatiile uzuale permise cu acestea sunt:

- încărcarea cu un cuvânt dorit de date, deplasarea informației între registre, sau între registre și memorie, rotirea conținutului acestora;
- tratarea unei perechi de registre ca reprezentând date pe cuvinte cu lungime dublă (la registre de 8 biți, perechi de 16 biți) și posibilitatea operării cu acestea.

Registreele speciale numite și registre de lucru, sau registre interne ale microprocesorului, au funcții predefinite specifice în funcționarea acestuia. Deși uneori pot fi setate la valori inițiale specificate, aceste registre nu sunt accesibile prin program, la execuția instrucțiunilor. În această categorie sunt registrele **contor de program (PC)**, **registrul de instrucțiuni**, **registrul decodificator de instrucțiuni**, etc.

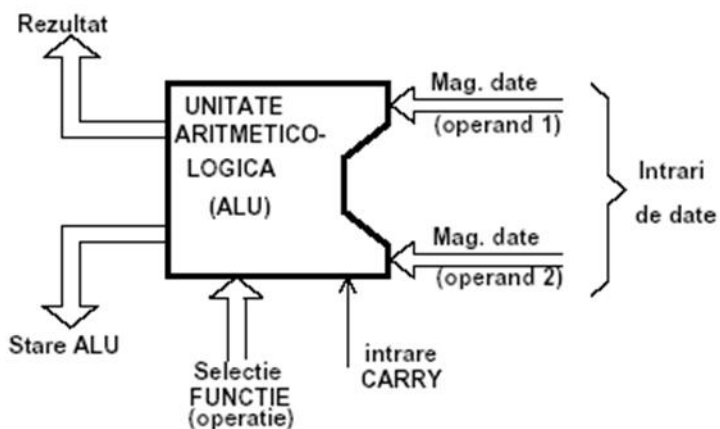
Subrutinele (subprogramele alias functiile in limbaje precum C, java) reprezintă porțiuni de program utilizate frecvent pentru a efectua anumite operații specifice. Deoarece la rândul ei, o subrutina poate apela o altă subrutină, etc., este necesară utilizarea unui mecanism specific pentru a se putea cunoaște adresa de revenire, la terminarea execuției unei subrutine, la punctul din program de unde se făcuse apelarea ei. Acest mod de a trata *evenimentele asincrone, relativ la desfășurarea curentă a programului microprocesorului, poartă numele de **lucru în întreruperi***.

O subrutină odată apelată, trebuie să permită revenirea în program, permițând continuarea acestuia din punctul de apel al subrutinei, sau de întrerupere. În acest scop există o zonă specială a memoriei, numită **stivă (stack)**, în care se stochează, pe perioada execuției unei subrutine, sau tratării unei întreruperi, adresa la care se va reveni, la terminarea acestei secțiuni speciale de program. În acest scop, microprocesorul este prevăzut cu un registru special, numit **registrul indicator de adresa al stivei programului (SP - stack pointer)**.

La terminarea codului subrutinei, programul va continua apoi din punctul de unde a fost efectuat apelul subrutinei.

Efectuarea operatiilor aritmetico-logice intr-un microprocesor

- Secțiunea din microprocesor care efectuează operațiile aritmetice și logice este compusă dintr-o **unitate aritmetico-logica (ALU)**, ce efectuează operația propriu-zisă, și **registre speciale** necesare la manipularea și memorarea temporară a operanzilor:



Structura secțiunii aritmetico – logice a microprocesorului

- Cele două *intrări de date* ale ALU vor furniza valorile celor doi (cazul operatorilor binari) operanzi ai unei operații aritmetice sau logice efectuate în ALU.
- Intrarea de *selecție a funcției* provine de la secțiunea de *CONTROL*, și specifică operația ce se va efectua.
- Intrarea *semnalului de transport (carry)*, este utilizată la efectuarea operațiilor aritmetice în dubla precizie.

Structura de bază a ALU

Posibilitățile *ALU* depind de tipul și performanțele microprocesorului: de la operații simple precum adunări și scăderi pe cuvinte de 8 biți și operații logice elementare, până la înmulțiri și împărțiri pe 16 biți etc.

În general, *ALU* poate efectua:

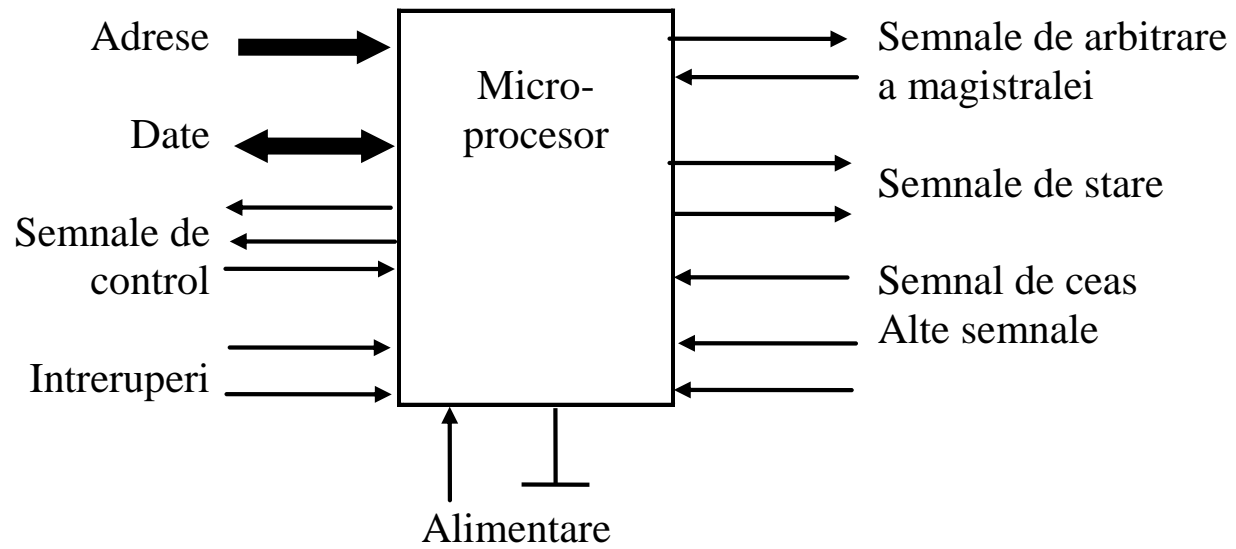
- adunări binare,
- scăderi binare
- operații logice booleene
- rotații la dreapta sau la stânga ale conținutului operanzilor.

Note:

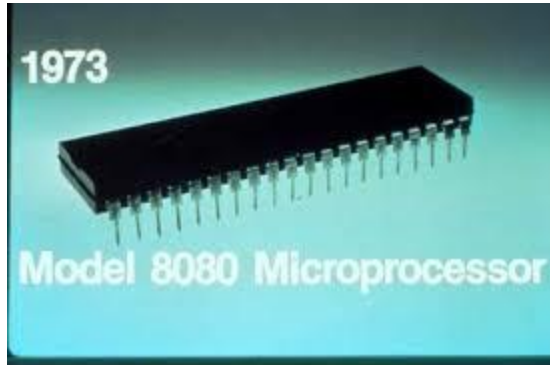
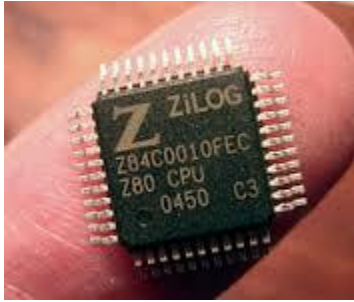
- operațiile complexe sunt implementate prin succesiuni de astfel de operații simple, printr-un software adecvat.
- microprocesoarele de 16 sau 32 biți, au implementate în setul de instrucțiuni și operațiile de înmulțire și împărțire

Semnalele unui microprocesor

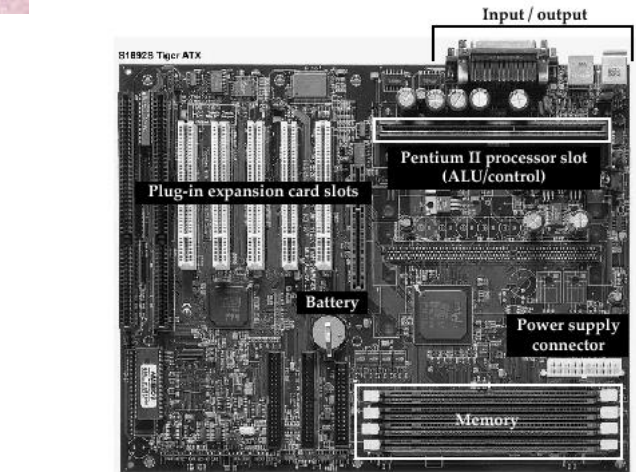
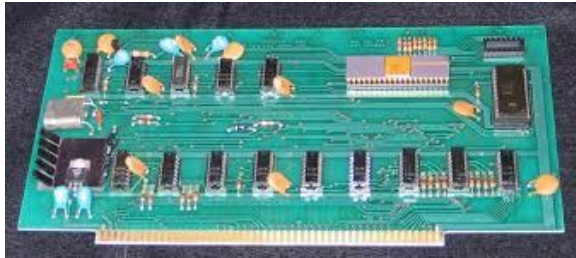
- semnale de control
- semnale de intrerupere
- semnale pentru arbitrarea magistralei
- semnale de stare
- semnale diverse.



Cum arata totusi un microprocesor ?



Unde se afla microprocesorul ?



Sa respiram un pic prin istorie....

Primul procesor

Primul microprocesor pe un singur chip, **4004** , 15 noiembrie 1971.

- procesor de 4 biti,
- 2300 de tranzistoare
- frecventa de 740 kHz.

Procesoare pe 8 biti:

Intel 8080, in anul 1974, (clona la Z80 ?)

- frecventa de 2 MHz
- 6000 de tranzistoare

Procesoare pe 16 biti, sfarsitul anilor '70

- procesoarele 8086, 8088, 80186 (XT technology), 80188 , 80282 respectiv 80286 (AT technology),
- 80282 avea 134000 tranzistoare in 1982.

Procesoare pe 32 de biti, sfarsitul anilor '80.

- 386, 486 etc.

Sa respiram un pic prin istorie.....

Procesor Pentium

Pentium I, 22 martie 1993.

- frecvente de 60 sau 66 MHz
 - 3,1 milioane tranzistoarelor integrate
- => Windows95 (windows + s.o.)!!!!!!!!!!!!!!!!!!!!

Pentium Pro (1995), 150 MHz si 200 MHz si Pentium II (1997),

Pentium III , anul 1999

Pentium IV, noiembrie 2000

Sa respiram un pic prin istorie....

Procesoare multi-core

- Pentium D , 2005 , primul procesor cu doua nuclee. In 2004 procesoarele lucrau la 3.8 GHz, avand probleme termice => intensificarea cercetarii in dezvoltarea procesoarelor mai eficiente din punct de vedere energetic.

.... au rezultat.....

- procesoare pe 64 biti, bazate pe microarhitectura NetBurst.
- 2006, Intel 64 , 64 biti, cu arhitectura “Core microarchitecture”.
- procesoare cu patru nuclee. Xeon, (destinat pentru servere si workstation-uri,) Intel Core 2 (pentru desktop-uri), Pentium Dual Core, dar si procesoare mai modeste, precum Celeron si Celeron M

Sa respiram un pic prin istorie....

Procesoare recente

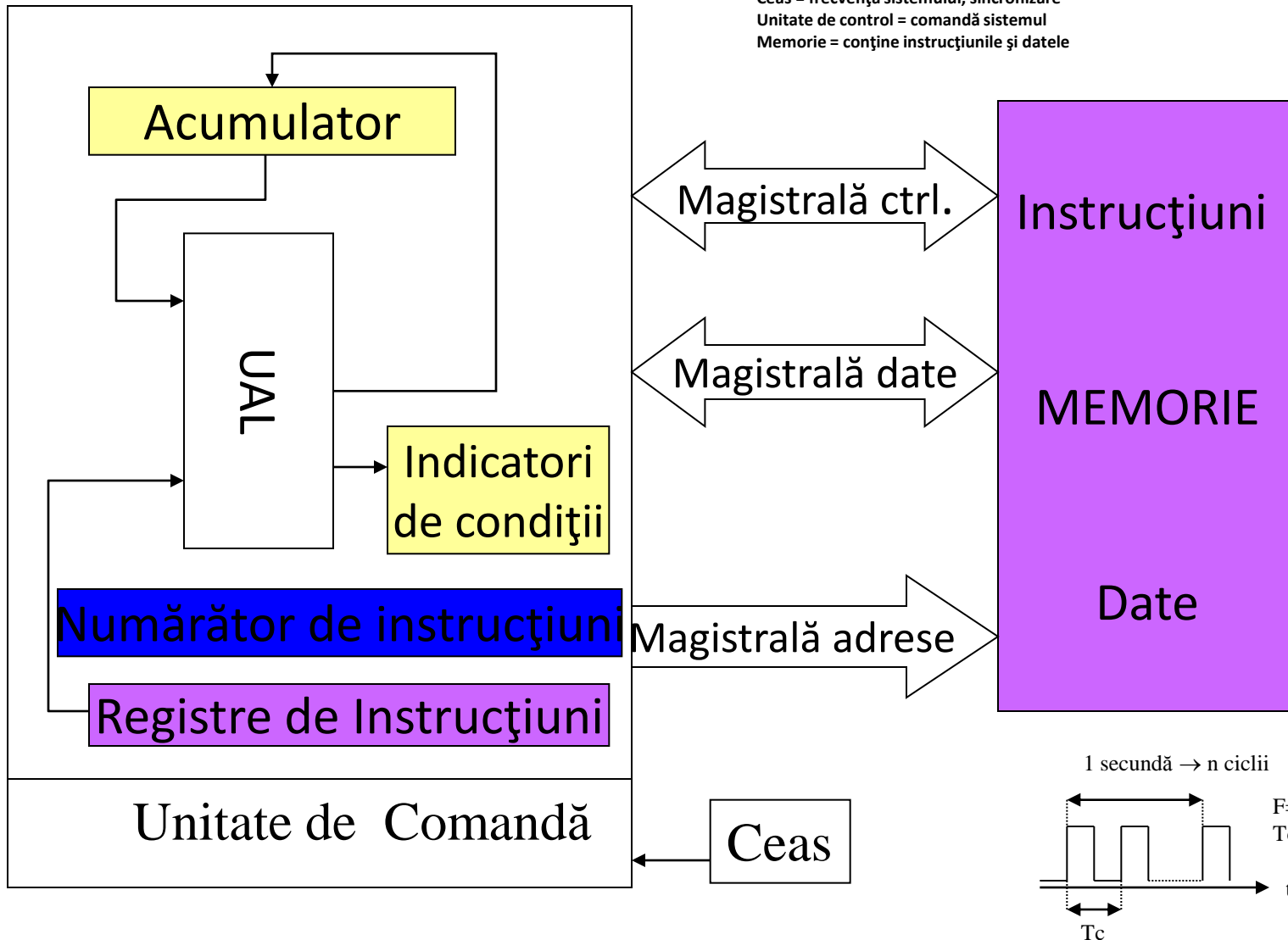
- Succesorul microarhitecturii “Core” a fost “Nehalem”.

Core i7 pentru desktop, in Noiembrie 2008.

- modele cu aceasta arhitectura: Intel Pentium, Core i3, Core i5, Core i7, respectiv Xeon.
- apar si procesoare foarte performante, cum ar fi de exemplu “Gulftown”, care are 6 nuclee fizice.
- Microarhitectura Sandy Bridge/Ivy Bridge. Procesoarele cu aceasta microarhitectura au modele precum Celeron, Pentium, Core i3, Core i5, Core i7.
- de mentionat procesorul cu 8 nuclee fizice (Core i7 Sandy Bridge-E) cu 2270 milioane de tranzistoare.

IN REZUMAT avem Schema funcțională

Acumulator = conține un operand și rezultatul
Numărător de instrucțiuni (PC) = adresa instrucțiunii următoare
Registre de instrucțiuni = codul operației sau operandul
Magistrală = ansamblu de conexiuni: adrese, date, control
Indicator de condiții = starea acumulatorului după ultima operație efectuată de UAL
Ceas = frecvența sistemului, sincronizare
Unitate de control = comandă sistemul
Memorie = conține instrucțiunile și datele



Tendinte si perspective de evolutie a sistemelor bazate pe microprocesoare

cresterea gradului de integrare

- dimensiune mai mica a elementului de comutare (tranzistor): 35nm -> ? -> pm ?
- cresterea numarului de tranzistoare la:
 - procesoare – **si mai multe miliarde de tranzistoare / cip** (vezi limitarea viteza lumina!)
 - memorii

reducerea puterii consumate

- distributie inteligenta a consumului
- controlul dinamic al consumului: energie mai mare daca este nevoie si unde este nevoie
- limitarea frecventei de lucru

arhitecturi multicore

- de la 2 core-uri/chip la 128 core-uri
- arhitecturi simetrice si asimetrice (ex: Intel v.s. Pover PC)

network-on-chip

- inlocuirea magistralelor paralele cu magistrale seriale – retele in interiorul procesorului