

# CAPITOLUL 1

## SISTEME DE CALCUL

### 1.1. Introducere

Din cele mai vechi timpuri oamenii au căutat să simplifice anumite activități, cum este și aceea de numărare. Astfel a luat naștere abacul, care poate fi considerat unul dintre cele mai primitive instrumente menite să automatizeze procesul de numărare.

Mult timp după inventarea abacului, matematicieni și alți oameni de știință au pus bazele științei calculatoarelor de astăzi, dintre care amintim pe: Blaise Pascal, Gotfried Wilhelm Leibniz, Charles Babbage, George Boole și Claude Shannon. Unul dintre pionierii în domeniu este considerat matematicianul francez Blaise Pascal (1623-1662), care a inventat în anul 1642 prima mașină de calcul operațională, denumită „Mașina aritmetică”. Această „mașină aritmetică” realiza operațiile de adunare și de scădere; interesant este faptul că scăderea se realiza utilizându-se tehnici complementare, foarte asemănătoare cu cele existente în calculatoarele din zilele noastre. Operațiile de înmulțire și împărțire erau implementate prin serii de adunări, respectiv scăderi repetate.

Matematicianul britanic Charles Babbage (1791-1871) a inventat primul dispozitiv ce poate fi considerat un computer în sensul modern al cuvântului. Acest prim dispozitiv modern calcula tabele de funcții logaritmice și trigonometrice prin intermediul unor oameni ce se numeau „computers”. Cea dintâi mașină concepută în 1822 de Charles Babbage se numea „Difference Engine”, fiind construită parțial, iar cea de-a doua mașină a fost denumită „Analytical Engine” (1830), dar nu a fost construită.

„Difference Engine” a fost construită ulterior conform desenelor originale de către o echipă la Muzeul Științei din Londra. Această mașină de calcul avea următoarele caracteristici:

- 4000 componente
- Cântărea 3 tone,
- Aproximativ 3 metri lățime și 2 metri și jumătate lungime.

Acest echipament a efectuat prima secvență de calcule la începutul anilor 1990, obținându-se rezultate cu o precizie de 31 de cifre zecimale.

O mare parte din activitatea lui Babbage a fost dedicată realizării de calculatoare analogice, dar tehnologia existentă la acea dată nu i-a permis realizarea de mecanisme de mare precizie. Sistemul de calcul analogic nu avea sistem de operare; simțind nevoia de software pentru calculatorul său, Babbage a angajat o tânără pe nume Ada Lovelace (fiica poetului britanic Lord Byron), nume după care a fost denumit limbajul de programare ADA. Astfel, **Ada Lovelace** este cunoscută **drept primul programator din lume**.

O contribuție deosebită în domeniul pionieratului calculatoarelor au avut-o și George Boole și Claude Shannon. În jurul anului 1850 matematicianul englez George Boole a inventat Algebra booleană (Algebra Boole), care a rămas relativ necunoscută și neutilizată până în anul 1938, când teza de masterat a lui Claude Elwood Shannon a demonstrat cum conceptele lui Boole „TRUE” și „FALSE” pot fi utilizate pentru a reprezenta funcționalitatea comutatoarelor din circuitele electronice.

În ultimii șaizeci de ani calculatoarele au evoluat continuu, specialiștii în domeniu împărțind această perioadă în cinci etape, denumite *generații* de calculatoare. Fiecare generație de calculatoare este caracterizată de o dezvoltare tehnologică majoră ce a schimbat fundamental modul în care calculatoarele operează, având ca rezultat echipamente mai mici, mai puternice, mai ieftine, mai eficiente și mai fiabile. De asemenea, fiecare generație se întinde pe o perioadă de aproximativ zece ani, în prezent aflându-ne în generația a cincea de calculatoare. Criteriile care au stat la baza acestei clasificări includ:

- Arhitectura sistemelor;
- Tehnologia de construcție a componentelor;
- Modalitățile de procesare a programelor;
- Caracteristicile sistemelor de operare;
- Limbajele de programare folosite.

## 1.2. Generații de calculatoare

**Prima generație** în evoluția calculatoarelor (1938-1953) a fost caracterizată de utilizarea tuburilor electronice iar calculatoarele erau destinate calculelor științifice și comerciale. În această perioadă se încadrează calculatoarele Harvard Mark 1 și ENIAC.

**Harvard Mark 1** (*IBM Automatic Controlled Calculator*) a fost construit între anii 1938 și 1944, fiind format din mai multe calculatoare ce lucrau asupra unor părți ale aceleiași probleme sub supravegherea unei singure unități de control. Această mașină de calcul era construită din comutatoare, relee și alte dispozitive mecanice, conținând 750.000 de componente, având 16 metri lungime, 2 metri și jumătate înălțime și cântărind 5 tone. Numerele folosite în calcule erau de 23 de cifre, o înmulțire dintre două numere dura 4 secunde iar o împărțire dura 10 secunde.

**ENIAC** - *Electronic Numerical Integrator And Computer* a fost realizat prin contribuția inginerilor William Mauchly și J. Presper Eckert de la Universitatea din Pennsylvania între anii 1943 și 1946. Acest calculator avea 3 metri înălțime, ocupa un spațiu de 30 mp și cântărea 30 tone. În construcția sa erau folosite 18.000 tuburi cu vacuum, având nevoie de o putere de 150 kW (suficient pentru a ilumina un mic oraș). Principala problemă cu acest tip de calculator era fiabilitatea: în fiecare zi trebuiau să fie înlocuite aproximativ 50 de tuburi cu vacuum. Prin realizarea acestui calculator, în anul 1943 Eckert și Mauchly au inițiat conceptul de creare a unui program stocat în calculator pentru care era folosită o memorie internă utilizată pentru a stoca atât instrucțiunile cât și date.

Succesorii lui ENIAC au fost:

- *EDVAC* - *Electronic Discrete Variable Automatic Computer* (4000 de tuburi);
- *EDSAC* - *Electronic Delay Storage Automatic Calculator* (1949 - 3000 de tuburi);
- *UNIVAC I* - *Universal Automatic Computer* (1951 – primul computer comercial);
- *ILLIAC I* (1949) - construit la Universitatea Illinois, primul computer deținut de o instituție academică.

**Generația a doua** (1954 - 1963) a fost caracterizată prin folosirea tranzistorului pentru realizarea circuitelor logice. Cercetarea în domeniul semiconductorilor a început în anul 1945 la faimosul centru de cercetare Bell Laboratories din SUA. Cei trei ingineri ce lucrau la acest proiect, William Shockley, Walter Brattain și John Bardeen au creat primul tranzistor pe 23 decembrie 1947. După această realizare au luat o pauză pentru sărbătorile de Crăciun înainte de a publica evenimentul; de aceea cărțile de referință indică faptul că primul tranzistor a fost creat în 1948.

Calculatoarele din cea de-a doua generație aveau în jur de 100 de instrucțiuni complexe, memorie de tip magnetic și dispozitive periferice. Prelucrarea se făcea secvențial prin intermediul unui sistem de operare simplu (FMS și IBSYS pentru mașina de calcul IBM 7094), folosindu-se atât limbajul de asamblare cât și limbaje evolute. Limbajul de programare Fortran a apărut în anul 1956 iar limbajele Algol și Cobol (Common Business Oriented Language - limbaj orientat spre aplicațiile de afaceri) după anul 1960. Calculatoarele erau folosite în mare parte pentru calcule științifice, rezolvarea sistemelor de ecuații liniare și neliniare, ecuații diferențiale etc. În anul 1955 a devenit operațional primul calculator cu tranzistori, denumit TRADIC (Transistor Digital Computer).

Această perioadă în evoluția calculatoarelor se caracterizează prin distincția creată între proiectanți, constructori, programatori și personalul de întreținere. Calculatoarele aveau nevoie de încăperi speciale, climatizate, pentru a se asigura parametrii de funcționare, aceste

mașini de calcul fiind întreținute și exploatate de către operatori calificați. Programele erau scrise în limbajul Fortran sau în limbaj de asamblare și erau imprimate pe cartele perforate. Programatorii duceau programul scris în camera de intrare și îl înmânau operatorilor. După ce programul era executat, operatorul aducea rezultatele de la imprimantă în așa numita cameră de ieșire, pentru a fi ridicate de către programator. În acest moment existau două variante: fie programul era corect și programatorul obținea rezultatele dorite, fie trebuia depanat programul și reluat acest proces.

Primele calculatoare din această a doua generație au fost construite pentru a fi utilizate în industria energiei atomice.

**Generația a treia de calculatoare** (1964 - 1980) se caracterizează prin utilizarea circuitelor integrate pe *scară mică* (SSI - *Small Scale Integration*) și *medie* (MSI - *Medium Scale Integration*). Cercetările în domeniul circuitelor integrate au debutat cu succes în cadrul firmei Texas Instruments, unde în anul 1958 inginerul Jack Kilby a reușit să combine mai multe componente pe o singură bucată de semiconductor, realizând astfel primul circuit integrat. La numai trei ani după această realizare, în anul 1961, Fairchild (cercetător la Texas Instruments) și Texas Instruments au realizat primele circuite integrate comerciale ce conțineau funcții logice de bază, două porți logice fiind implementate prin intermediul a 4 tranzistori bipolari și 4 rezistoare. Tot Fairchild introducea în anul 1970 pentru prima oară memoria de 256 biți static RAM (Random Access Memory).

Această perioadă de dezvoltare a calculatoarelor poartă denumirea de anii „*big iron*”, în care calculatoarele de tip mainframe ale firmei IBM dețineau supremația.

În anul 1970 compania japoneză Busicom de calculatoare a cerut firmei Intel un set de 12 circuite integrate pentru a le utiliza într-un nou calculator. T. Hoff, proiectant la Intel, inspirat de această cerere, a creat primul microprocesor, denumit **Intel 4004**. Acest procesor avea 2300 de tranzistoare și putea executa 60.000 operații pe secundă. Primul microprocesor de uz general, denumit **8080**, a fost introdus de Intel în 1974, fiind un procesor pe 8-biți, cu 4500 tranzistori și putând efectua 200.000 operații pe secundă. Alte procesoare din această vreme au fost: Motorola 6800, MOS Technology 6502, Zilog Z80.

Treptat începe deschiderea spre era calculatoarelor personale; astfel, în 1974 Ed Roberts lansează pe piață calculatorul Altair 8800 ce avea următoarele caracteristici:

- Era bazat pe microprocesorul 8080.
- Avea prețul de 375\$.
- Fără tastatură, ecran, capacitate de sto
- Avea 4KB memorie,

În anul 1975 Bill Gates și Paul Allen fondează firma Microsoft și implementează limbajul de programare **BASIC 2.0** pe Altair 8800, care devine **primul limbaj de nivel înalt** disponibil pe un **calculator personal**.

Alte exemple de calculatoare din această perioadă sunt:

- S. Wozniak și S. Jobs produc Apple I în 1976 și Apple II în 19

Caracteristici:

- 16k ROM, 4k de RAM, tastatură și display color;
- preț de 1300\$, în 1977 afacere de 700.000 \$ iar în 1978 afacere de 7 milioane de dolari.

- TRS-80 (bazat pe microprocesorul Z80) de la Radio Shack în 1977. Caracteristici:
  - 4k ROM, 4k RAM, tastatură și drive de tip casetă;
  - preț de 600\$.

**Generația a patra de calculatoare** (1981 - prezent) se caracterizează prin construcția de calculatoare ce utilizează circuite integrate pe *scară mare (LSI - Large Scale Integration)* și *foarte mare (VLSI - Very Large Scale Integration)*. Folosirea microprocesorului și a microprogramării a oferit calculatoarelor posibilitatea utilizării unui set complex de instrucțiuni și asigură un grad sporit de flexibilitate.

În această perioadă a fost scos pe piață primul calculator personal (PC- Personal Computer) de către firma IBM în anul 1981. Toate calculatoarele personale ce au fost construite ulterior și au păstrat arhitectura originală IBM au fost denumite calculatoare compatibile IBM-PC. Primul IBM-PC avea următoarele caracteristici:

- Microprocesor pe 16-biți 8088
- Memorie ROM BASIC;
- Floppy-disc de 360KB ca
- Sistem de operare DOS 1.0
- Preț de 1365 \$

Cele mai importante momente din evoluția ulterioară a calculatoarelor sunt:

- În 1983 calculatorul IBM-XT are hard-disc (10MB de memorie costau 3000\$).
- În 1985 Intel introduce microprocesorul 80386 (primul membru pe 32-biți din familia 80x86).
- În 1986 firma Compaq introduce primul sistem bazat pe 80386.
- În 1989 Intel introduce microprocesorul 80486, ce includea coprocesor matematic.
- În 1992 apar procesoarele Intel Pentium (cu 64-biți pentru magistrala de memorie), AMD și Cyrix (*procesoare compatibile - „clone”*).
- În 1996 apare Intel Pentium Pro.
- În 1998 apare Intel Pentium II.
- În 2000 apare Intel Pentium IV la 1.5 GHz.
- În 2003 este depășită granița de 3 Ghz de funcționare a procesoarelor comerciale.

În prezent, cele mai moderne calculatoare sunt cele care au și gradul de miniaturizare cel mai mare, aici incluzând calculatoarele portabile de tipul *laptop* sau *palmtop* (denumit și *PDA - Personal Digital Assistant* sau *PocketPC*).

**Calculatoarele din cea de-a cincea generație** (prezent - viitorii ani), sunt încă în faza de dezvoltare. Ele se bazează pe utilizarea inteligenței artificiale, a circuitelor integrate specializate și a procesării paralele. Există unele aplicații ale celei de-a cincea generații de calculatoare care sunt deja utilizate astăzi, cum ar fi recunoașterea vorbirii. Utilizarea procesării paralele și a superconductorilor face viabilă inteligența artificială.

Procesarea moleculară și cuantică precum și nanotehnologiile se pare că vor schimba fața calculatoarelor în următorii ani. Scopul principal al celei de-a cincea generații de calculatoare este acela de a dezvolta echipamente capabile să răspundă limbajului natural uman și să fie capabile de învățare și organizare proprie.

Procesarea cuantică reprezintă un termen introdus încă din anii '70 și se bazează pe fizica cuantică, folosindu-se anumite proprietăți ale atomilor și nucleelor ce le permite să lucreze împreună drept biți cuantici (denumiți *qubits*) pentru a fi utilizați de procesor și memorie. Prin interacțiunea dintre ei, izolați de mediul exterior, qubiții pot realiza anumite calcule matematice mult mai rapid decât calculatoarele convenționale. Qubiții nu se bazează pe natura binară tradițională a procesării. Spre deosebire de codificarea tradițională a numerelor ce folosește doar 0 și 1, calculatoarele cuantice codifică informația ca o serie de stări mecanico-cuantice precum mișcarea de rotație a electronilor sau orientarea polarizării ce pot reprezenta un 0 sau un 1, o combinație a celor două sau pot reprezenta un număr ce semnifică faptul că starea unui qubit este undeva între 0 și 1. Este important faptul că, utilizând această idee, un calculator cuantic monoprosesor poate executa o sumedenie de operații în paralel. Prelucrarea cuantică nu este cea mai bună soluție pentru activități de procesare de texte sau poștă electronică, dar este ideală pentru aplicații criptografice sau de modelare și indexare a bazelor de date de dimensiuni foarte mari.

### 1.3. Clasificarea sistemelor de calcul

Marea varietate a calculatoarelor a impus clasificarea lor după diferite criterii: cost, capacitate, complexitate, aplicații. În general, tipul calculatorului se determină după:

- *tipul unității centrale de prelucrare (UCP) sau al microprocesorului* – cele mai mari calculatoare tind să utilizeze unități centrale de prelucrare constituite separat, de mare viteză, cu componente complexe;
- *cantitatea de memorie principală pe care microprocesorul o poate utiliza* – un calculator echipat cu o memorie principală de mare capacitate poate memora programe mai complexe și chiar mai multe programe diferite în același timp;
- *capacitatea de stocare a memoriei auxiliare* – sistemele de calcul tind să fie echipate cu dispozitive periferice de memorare de mare capacitate;
- *viteza perifericelor de ieșire este o altă caracteristică* – cele mai mari calculatoare sunt dotate cu dispozitive de ieșire rapide, a căror viteză se măsoară, de exemplu, în sute de mii de linii care pot fi tipărite pe minut;
- *viteza de prelucrare* exprimată în milioane de instrucțiuni pe secundă (MIPS - Millions of Instructions Per Second) variază de la 3 - 4 MIPS la cele mai mici calculatoare, până la mai mult de 200 MIPS pentru supercalculatoare;
- *numărul utilizatorilor* care pot avea acces la calculator în același timp – calculatoarele personale admit numai un singur utilizator, alte tipuri acceptă mai mult de doi sau trei utilizatori în același timp, iar cele mai mari calculatoare suportă sute de utilizatori simultan;
- *costul sistemului* poate varia foarte mult.

Având în vedere totalitatea criteriilor enumerate mai sus, calculatoarele sunt, în general, grupate în patru categorii de bază: *microcalculatoare*, *minicalculatoare*, *calculatoare mainframe* și *supercalculatoare*. Este dificil să se asocieze o definiție fiecărei categorii, ținând seama de progresele tehnologice și de rapiditatea cu care se pot schimba parametrii de mai sus. Totuși următoarele definiții ar putea fi suficiente.

*Microcalculatorul*, numit adesea calculator personal (PC – Personal Computer), reprezintă tipul de calculator care utilizează un microprocesor ca unitate centrală de prelucrare (UCP) și care poate fi folosit numai de o singură persoană la un moment dat. Există un mare număr de variante, în ceea ce privește dimensiunea, de la calculatoare personale portabile (*laptop*) la puternice stații de lucru (*desktop workstations*) care sunt utilizate pentru calcule ingineresti și științifice.

Stațiile de lucru utilizează sistemele de operare UNIX sau Windows NT/2000/XP și sunt echipate cu procesoare RISC puternice (cum ar fi Digital Alpha, PowerPC sau MIPS) sau cu procesoare Intel Pentium (și compatibile). Calculatoarele personale lucrează folosind sistemul de operare Windows sau un alt sistem de operare similar, fiind folosite pentru aplicații standard. Microcalculatoarele pot fi folosite cu ușurință de neprogramatori datorită numărului mare de pachete de programe de aplicații disponibile.

În ultimii ani PC-urile au devenit un lucru comun în instituții, școli, universități și locuințe. PC-urile au schimbat modul în care se gândește, se învață și se lucrează. Astăzi PC-ul este un ajutor de neprețuit pentru cei care vor să-și îmbunătățească performanțele și calitatea muncii. Piața explozivă a PC-urilor a condus la creșterea numărului utilizatorilor, la accelerarea dezvoltării domeniilor de aplicații, de la programele de prelucrare de texte până la tehnologii care au permis oamenilor să lucreze acasă, să se joace, să învețe sau să facă cercetări avansate.

*Minicalculatorul* este cunoscut ca un calculator de mărime medie, ce nu este portabil. El suportă până la 50 de utilizatori simultan și are o memorie principală de capacitate mare. În mod normal minicalculatorul deservește o rețea de terminale simple. IBM AS/400 sau DEC Vax/750 sunt exemple de minicalculatoare.

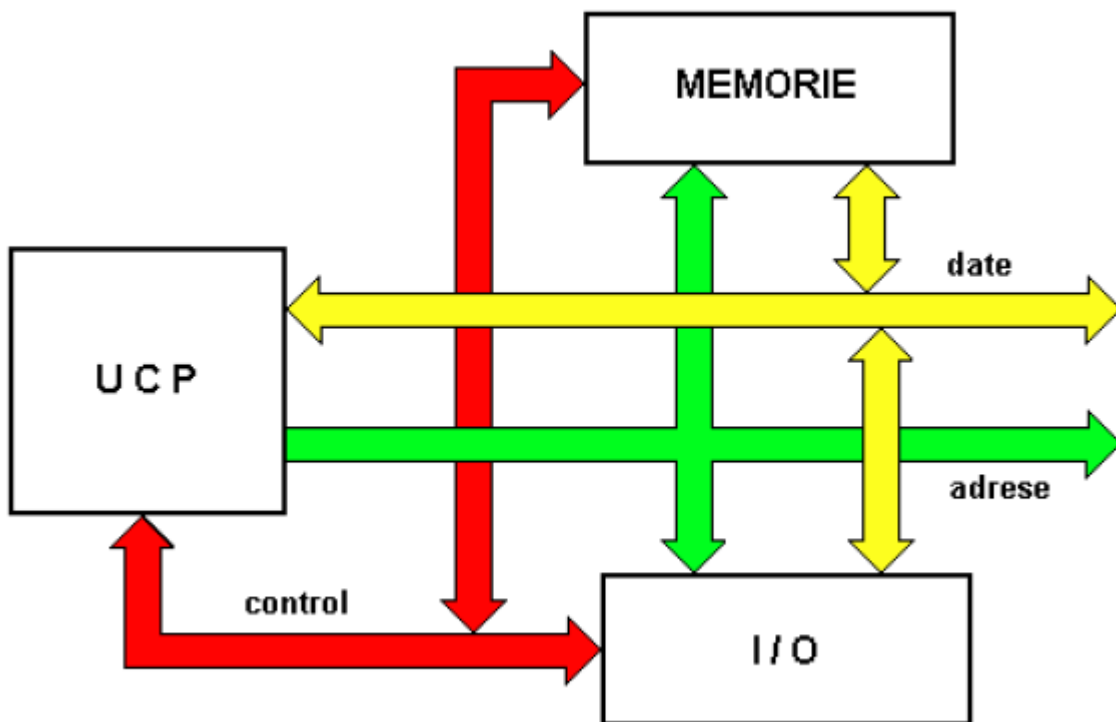
*Calculatorul mainframe* reprezintă un calculator de mari dimensiuni și foarte puternic care este amplasat într-un cadru care poate fi controlat. Un astfel de calculator suportă prelucrări cerute de sute, chiar mii de utilizatori precum și calcule specializate. Este solicitat de companiile care vehiculează și prelucrează un volum mare de informație. Ca exemplu este modelul 390 al IBM.

*Supercalculatorul* posedă resurse hardware și software deosebite. Se utilizează în industria de apărare, în lumea cercetării științifice, în câteva universități, în industria aeronautică și spațială. Un supercalculator poate executa peste 1,8 miliarde de operații pe secundă.

## 1.4. Elemente de arhitectura și structura unui sistem de calcul

### 1.4.1 Arhitectura unui sistem de calcul

Sistemele de calcul existente cunosc un mare număr de variante arhitecturale. Principiul de bază al proiectării calculatoarelor clasice a fost ordonarea secvențială a operațiilor elementare. Acesta se rezumă prin două aspecte majore: execuția unui ansamblu de funcții de către un procesor unic și descrierea prelucrărilor în conformitate cu algoritmi secvențiali. Modelul care sta la baza arhitecturii unui sistem de calcul este modelul microcalculatorului pe baza arhitecturii VON NEUMANN.



Modelul Von Neumann

Un sistem de calcul este un ansamblu de două componente:

- **HARDWARE** – este un termen care acoperă totalitatea componentelor electronice și mecanice ale sistemului de calcul (partea fizică - **unitatea centrală de prelucrare - UCP, memoria M, dispozitivele periferice de intrare/ieșire I/E**);
- **SOFTWARE** – este un termen care acoperă totalitatea programelor utilizate într-un sistem de calcul (partea logică - **sistem de operare, programe utilitare**) care interacționează între ele în vederea satisfacerii cerințelor utilizatorilor.

**Sistemul de operare (SO)** reprezintă acea componentă software care asigură interconectarea tuturor componentelor sistemului de calcul, transformându-le într-o entitate – calculatorul și care asigură și interconectarea acestuia cu mediul exterior.

Calculatoarele personale moderne își au rădăcinile în SUA începând cu anul 1940, deși pe piață au apărut în 1981. Dintre oamenii de știință ce și-au adus contribuția în acest domeniu, trebuie amintit John von Neumann (1903-1957), matematician născut în Ungaria. El a fost primul care a proiectat un calculator cu memorie de lucru (memoria RAM de astăzi). Modelul unui calculator personal al lui Von Neumann include: UCP, intrarea, ieșirea, memoria de lucru și memoria permanentă.

Dintre elementele care au determinat ca PC-urile să se impună pe piața sistemelor de calcul enumerăm:

- au reprezentat un început pentru standardizare, având o *arhitectură deschisă*;
- fiind bine documentate au oferit posibilități de extindere;
- au fost ieftine, simple și robuste.

Primele calculatoare personale, realizate de către firma IBM, aveau la bază microprocesorul Intel 8088 iar ca sistem de operare MS-DOS de la Microsoft. Numele original de *calculator personal* sau *PC (Personal Computer)* provine deci de la IBM iar toate calculatoarele construite după aceea, având aceeași arhitectură de bază, poartă denumirea de calculator *compatibil IBM-PC*, tocmai datorită faptului că păstrează arhitectura funcțională de bază a primului calculator personal scos pe piață de către firma IBM. PC-urile existente astăzi sunt la fel de puternice ca minicalculatoarele și calculatoarele mainframe de acum câțiva ani.

#### 1.4.2 Procesorul

Componenta cea mai importantă a unui sistem de calcul, în particular a unui calculator personal, este *procesorul* sau *unitatea centrală de prelucrare (UCP)*. Denumirea de unitate centrală de prelucrare provine din următoarele considerații:

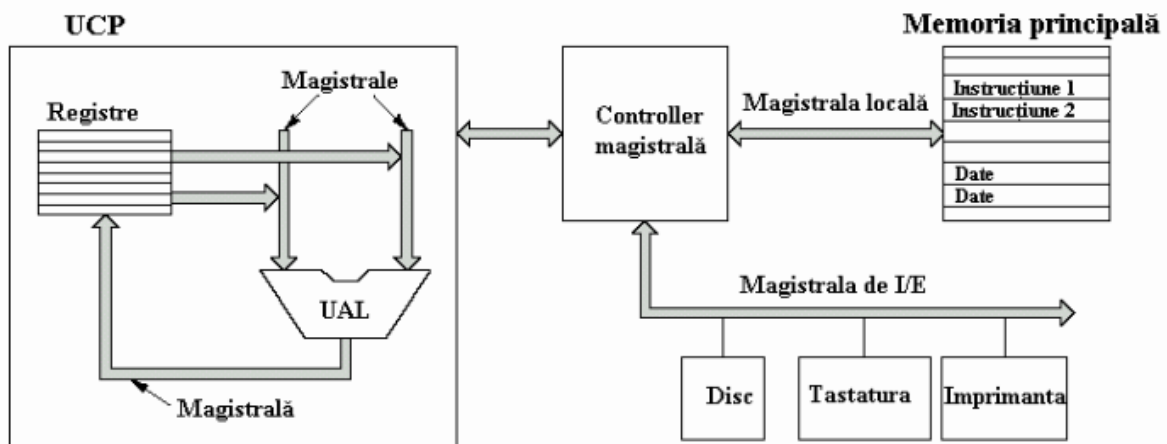
- *procesor*, deoarece prelucrează datele;
- *central*, deoarece este centrul de prelucrare a datelor din sistem;
- *unitate*, deoarece, de cele mai multe ori, este un circuit integrat care conține zeci sau sute de milioane de tranzistoare – un *microprocesor*.

Microprocesorul conține:

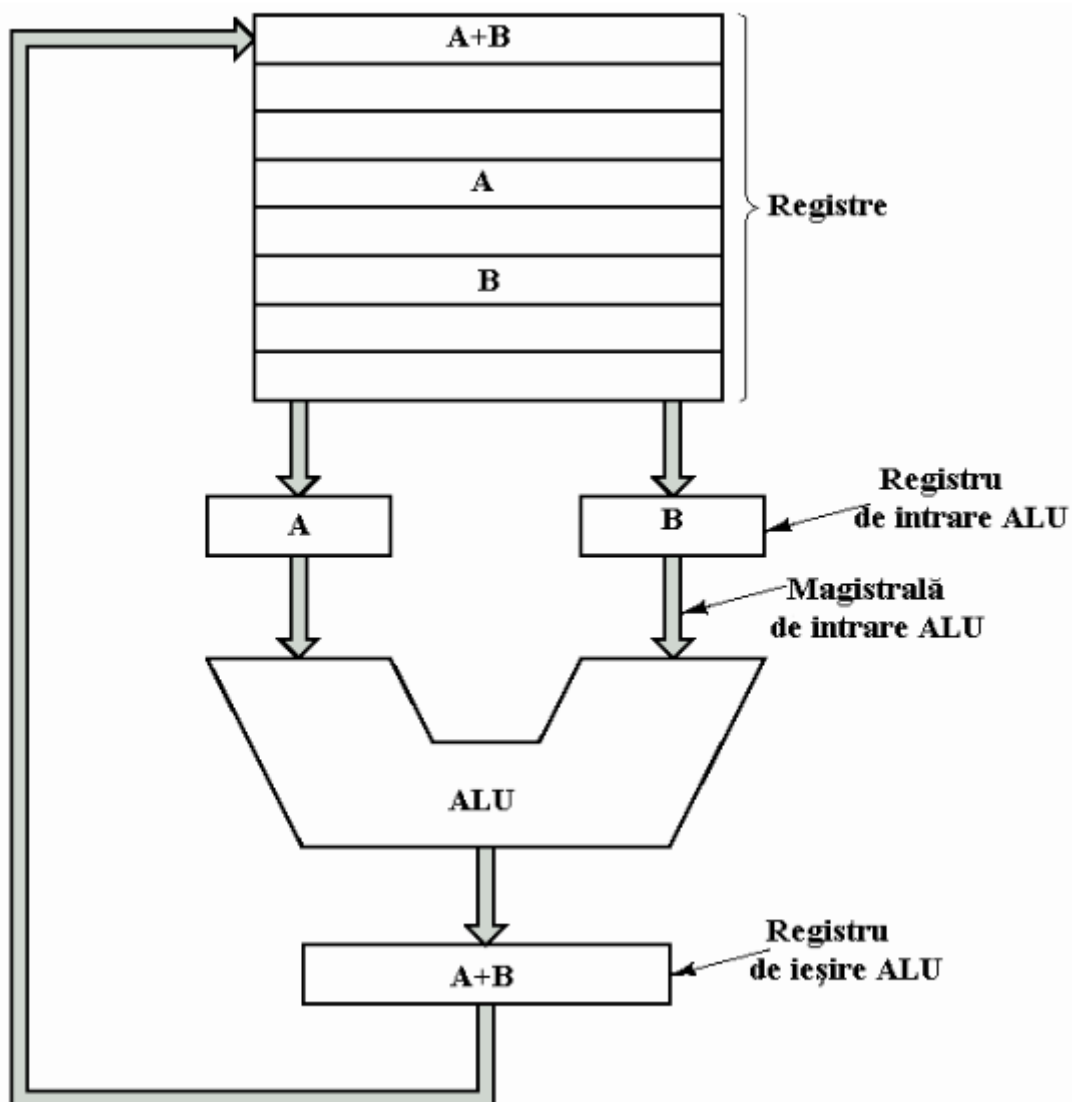
- unitatea aritmetico-logică (UAL);
- unitatea de comandă și control (UCC);
- un ansamblu de registre (R).

Unitatea aritmetico-logică execută operații aritmetice și funcții logice. Unitatea centrală de procesare coordonează activitatea din calculator în vederea prelucrării datelor. Pentru aceasta, ea primește și transmite date în mod continuu. Datele pentru prelucrare vin de la memorie și diversele dispozitive periferice din calculator (tastatură, hard-disc etc.) iar după procesare vor fi trimise către memorie sau alte unități. Transmiterea se face prin intermediul magistrelor.

Unitatea centrală de procesare primește *instrucțiunile* care vor fi executate. Fiecare instrucțiune reprezintă o comandă pentru prelucrarea datelor. Activitatea procesorului constă din *calculare și transportul datelor*.



Organizarea unui sistem de calcul



Prelucrarea datelor într-un procesor von Neumann

La ora actuală, unitatea centrală de prelucrare poate utiliza una din tehnologiile:

- **Tehnologia CISC** (*Complex Instruction Set Computer*) care a fost adoptată încă de la primele arhitecturi de calculatoare. Procesoarele sunt capabile să interpreteze mai mult de 400 instrucțiuni și execuția se realizează în mai multe cicluri mașină. Numărul de cicluri mașină necesare în execuția unei instrucțiuni variază de la o instrucțiune la alta.
- **Tehnologia RISC** (*Reduced Instruction Set Computer*) prin care se implementează numai instrucțiunile de bază (cele mai folosite). Fiecare instrucțiune se execută într-un singur ciclu mașină, având consecințe pozitive asupra performanțelor. Această tehnologie permite construcția unor mașini foarte puternice, capabile să asigure prelucrarea paralelă a mai multor aplicații.
- **Tehnologia VLIW** (*Very Long Instruction Word*) prin care procesorul utilizează instrucțiunea într-un format lung. Ideea este de a reuni mai multe instrucțiuni în una singură. Astfel procesorul poate pregăti mai multe instrucțiuni printr-o singură operație și va fi mult mai eficient. În mod normal procesoarele care nu sunt de tip VLIW recepționează o instrucțiune pe cuvânt. Un cuvânt este o cantitate de date transmisă la procesor, iar procesoarele VLIW recepționează mai multe instrucțiuni în fiecare cuvânt. Pentru reordonarea instrucțiunilor se utilizează un compilator. În acest caz unitatea centrală de prelucrare proiectată în tehnologie VLIW devine foarte complexă. Procesorul Itanium pe 64 de biți al firmei Intel a fost proiectat în tehnologie VLIW. O altă companie care utilizează VLIW este TransMeta cu procesorul Crusoe.

Firma Sun Microsystems a definitivat între anii 1984 și 1987 *arhitectura SPARC* (*Scalable Processor ARChitecture*) ce se bazează pe tehnologia RISC.

Începând din 1990, IBM a lansat pe piață sistemul RISC/6000, prin stații de lucru și servere, având la bază *arhitectura POWER* (*Performance Optimization With Enhanced RISC*). Acest sistem reprezintă rezultatul muncii susținute de IBM, în vederea obținerii unor performanțe sporite la un preț de cost cât mai redus.

**Procesoarele RISC**, apărute pentru prima oară în anii '80, păreau la început predestinate să domine industria computerelor în anii '90 și să facă uitate vechile arhitecturi de calculatoare. Practic toți producătorii importanți din industria calculatoarelor ofereau (și oferă în continuare) sisteme gen RISC; giganții IBM și Hewlett Packard și-au dezvoltat propriile procesoare RISC, în timp ce alți producători, ca DEC sau Siemens, au preferat să cumpere licențe ale unor arhitecturi deja existente pentru a ține pasul cu concurența acerbă din domeniu.

Conceptul de arhitectură „RISC” este deseori greșit utilizat sau definit, de aceea, pentru a fi definit și înțeles mai bine, trebuie să facem o întoarcere în timp și să vedem, de asemenea, diferențele esențiale între cele două noțiuni: CISC și RISC. Era recunoscut de prin anii '50 faptul că se putea sacrifica din eficiența memoriei la codificarea unui set de instrucțiuni pentru a câștiga în performanță. Instrucțiunile simple, de lungime fixă, erau ușor de implementat și se executau mai rapid. Această tehnică era folosită pentru implementarea setului de instrucțiuni al calculatorului IBM 360 de tip mainframe din anii '50-'60. Acest set de instrucțiuni se baza pe o arhitectură clasică CISC, dar mecanismul de microcod ce executa de fapt instrucțiunile era un procesor RISC mai simplu.

Microcodul este de fapt, software-ul de nivel jos care conduce execuția setului de instrucțiuni, iar procesoarele RISC se numeau *controlere orizontale de microcod*. Cu toate că erau cunoscute avantajele arhitecturilor RISC, costurile ridicate ale memoriei determinau folosirea

în continuare a arhitecturilor CISC, mai eficiente din punct de vedere al capacității de stocare și care păreau că reprezintă cea mai bună soluție în acel moment (se utilizau instrucțiuni capabile să facă mai multe lucruri).

Primele inovații față de vechile arhitecturi de microprocesor au apărut în cadrul firmei IBM, în cadrul unui proiect început în 1975 și care acum se consideră a fi pionieratul în domeniul arhitecturii RISC. John Cocke, un inginer de la IBM, a observat că doar o mică parte (aproximativ 10%) a mulțimii de instrucțiuni a calculatorului IBM 360 era folosită în majoritatea timpului, iar această submulțime concentra cel mai mare procent din timpul de execuție (90%). Membrii echipei IBM și-au propus astfel să simplifice mulțimea de instrucțiuni pentru a obține o medie de o execuție pe ciclu de ceas. Acest obiectiv era realizabil doar dacă mulțimea de instrucțiuni era structurată în conductă, salvându-se în acest mod timpul pierdut pentru aducerea și decodarea instrucțiunilor din memorie.

Două noi proiecte ce au pornit câțiva ani mai târziu au adus conceptele RISC în centrul atenției arhitecturilor de calculatoare. Primul dintre acestea a fost condus de David Patterson de la Universitatea din Berkeley și a culminat cu definiția procesoarelor RISC I și RISC II la începutul anilor '80. Patterson a conturat, de asemenea, conceptul RISC. Potrivit acestuia, procesoarele RISC au inaugurat o nouă mulțime de principii arhitecturale. Din această cauză, noțiunea de RISC a fost considerată mai degrabă o filozofie decât o rețetă arhitecturală diferită. Punctele relevante ale filozofiei proiect menționate de Patterson sunt:

- mulțimea de instrucțiuni trebuie să fie simplă;
- instrucțiunile trebuie să ruleze la cea mai mare rată posibilă;
- noțiunea de „pipelining” este mai importantă decât mărimea programului;
- tehnologia compilatorului este un element critic într-un proiect RISC: optimizarea compilatoarelor trebuie să translateze cât mai mult posibil din complexitatea hardware-ului către faza de compilare.

Rezultatele acestor cercetări au dat naștere unei arhitecturi mai simple, caracterizată de instrucțiuni mai puține, multe registre, acces simplificat pentru încărcarea și depozitarea datelor în memoria principală și posibilitatea execuției instrucțiunilor într-o singură perioadă de ceas.

Procesorul respectiv era mai mic, cu performanțe mai mari, dar se folosea cu 20-25% mai multă memorie și erau necesare memorii cache scumpe pentru a ține „ocupat” microprocesorul RISC. Din această cauză, costurile ridicate ale arhitecturilor RISC au împiedicat răspândirea acestora pe piața consumatorilor medii și a aplicațiilor comerciale. Procesoarele RISC erau însă folosite uzual în stațiile de lucru foarte puternice pentru aplicații științifice, tehnice și militare, unde se justificau prețurile mari pentru performanțe înalte.

O dată cu evoluția microprocesoarelor RISC, s-a descoperit că avantajul acestora nu costă în micșorarea setului de instrucțiuni, ci în simplitatea acestora. În zilele de astăzi majoritatea microprocesoarelor RISC au cam același număr de instrucțiuni ca și cele CISC. Datorită modurilor mai simple de adresare ale instrucțiunilor RISC, având nevoie de un singur acces la memoria principală și putând fi executate într-un singur ciclu de ceas, execuția lor a putut fi foarte ușor implementată în structuri de tip pipeline și structuri superscalare ce permit execuția simultană a mai multor instrucțiuni.

Tot evidențiind avantajele tehnologiei RISC, se pune în mod evident întrebarea: De ce să mai folosim procesoare CISC, când cele RISC sunt în mod clar mai performante?

Pentru a răspunde la această întrebare, să evidențiem câteva aspecte ale problemei. În primul rând, diferențele dintre microprocesoarele CISC și cele RISC nu mai sunt așa de mari odată cu implementările RISC făcute în ultimii ani de către microprocesoarele compatibile Intel. Avantajul major care apare prin folosirea procesoarelor CISC este acela al compatibilității soft; astfel, cantitatea de software ce rulează în prezent pe microprocesoare CISC este imensă și deocamdată nu se poate renunța la ea.

Succesul mai vechi al variantelor de Microsoft Windows (ce rulează pe procesoare CISC) și faptul că noile versiuni de Windows, rulează tot pe platforme cu procesoare CISC (Intel și compatibile cu Intel), face să se mențină încă utilizarea cu precădere a acestor tipuri de microprocesoare. Dacă cineva dorește să achiziționeze un calculator bazat pe un microprocesor RISC, nu va putea beneficia de programele scrise pentru calculatoarele PC, deoarece majoritatea acestui soft este special proiectat pentru calculatoare PC.

Dacă softul pentru PC va dori să ruleze pe un procesor RISC, vor apărea numeroase probleme, printre care:

- aplicațiile au fost compilate pentru a lucra doar cu setul de instrucțiuni x86;
- software-ul se așteaptă să sesizeze un mediu de operare Microsoft sub care să lucreze;
- multe aplicații și jocuri DOS mai vechi trebuie să găsească mediul hard al calculatorului PC, lucrând de multe ori direct cu resursele hard ale calculatorului.

Pentru prima problemă ar putea exista soluția recompilării aplicației astfel încât să poată opera cu setul de instrucțiuni al microprocesorului RISC. La ora actuală, multe aplicații sunt disponibile în mai multe versiuni, fiind compilate pentru platforme RISC, dar numărul acestora este totuși destul de redus și există rețineri în privința firmelor de a elabora mai multe versiuni (din acest punct de vedere) ale aceleiași aplicații. Reținerile acestor firme sunt întemeiate, deoarece piața este prea mică pentru ca ele să-și permită o asemenea dezvoltare și, în al doilea rând, este greu de menținut un nivel apropiat pentru două sau mai multe versiuni de program.

În concluzie, apare o problemă cu dublu sens: nu există multe calculatoare RISC pentru că nu există soft pentru ele și nu există soft pentru că nu există calculatoare RISC!

Pentru a doua problemă, se părea că firma Microsoft a rezolvat problema o dată cu apariția sistemului de operare NT care oferă portabilitate pentru procesoarele MIPS și ALPHA de tip RISC. Inexistența unui sistem de operare de la Microsoft este o piedică esențială în calea pătrunderii procesoarelor RISC pe piața calculatoarelor personale.

Dacă pentru primele două probleme prezentate mai sus există soluții de rezolvare, pentru cea de-a treia nu există o soluție tehnică generală, din moment ce aplicațiile scrise pentru un mediu hardware specific unui PC nu vor putea rula pe procesoare RISC. În acest sens, singura soluție viabilă este practic rescrierea aplicației pentru noua platformă, soluție care nu se impune din aceleași considerente prezentate mai înainte: numărul de stații de lucru ce folosesc microprocesoare RISC este încă redus.

### ***Tipuri de procesoare RISC***

Cele mai importante arhitecturi ce conțin procesoare RISC sunt:

- MIPS, folosite în stații de lucru Silicon Graphics;
- SPARC, folosite în stații de lucru Sun;
- PA-RISC, folosite în stații de lucru Hewlett-Packard;
- PowerPC, folosite în calculatoare IBM PC și Apple Macintosh;
- Alpha, folosite în stații de lucru DEC (Digital Equipment Corporation).

Competiția pe piața microprocesoarelor RISC este foarte mare; în mod continuu apar pe piață noi versiuni de procesoare. Toate aceste arhitecturi evoluează însă în aceleași direcții:

- implementare pe 64 de biți;
- unități performante de execuție;
- noi instrucțiuni pentru aplicații multimedia și DSP (Digital Signal Processing);
- frecvențe de ceas intern foarte mari, superioare procesoarelor CISC;
- implementări superscalare, putând să execute mai multe instrucțiuni simultan;
- unități de operare în virgulă mobilă foarte puternice;
- memoria cache integrată de dimensiuni mari.

### ***Procesoarele din seria MIPS (Silicon Graphics)***

La originea acestor microprocesoare se află niște proiecte experimentale inițiate la Universitatea din Stanford la începutul anilor '80. Traducerea termenului „MIPS” ne oferă o imagine relevantă asupra filozofiei proiectului respectiv: *MIPS* provine de la *Microprocessor without Interlocking Pipeline Stages* (*Microprocesor fără stadii în conductă blocate*).

Obiectivul proiectanților MIPS a fost acela de a produce un procesor RISC cu funcționare în conductă și inter-blocare pipeline controlate software. Dacă o instrucțiune necesită două cicluri de ceas pentru a fi executată, este de datoria compilatorului să programeze o instrucțiune de tipul NOP (No OPeration) următoare. În acest mod singura modalitate prin care se întrerupe funcționarea normală în timpul execuției sunt aceste instrucțiuni NOP controlate software (de compilator), în timp ce partea hardware nu va bloca de fiecare dată execuția pipeline. Această caracteristică reduce cantitatea de componente hardware necesare pentru manufacturarea procesorului.

Un produs MIPS din anul 1995 a fost MIPS T5 (redenumit apoi R1000), cu o arhitectură superscalară pe 64 de biți nouă, compatibilă cu cipurile mai vechi Rxxx. Arhitectura scalară dispunea de 5 canale, 64 de registre interne și o memorie cache internă de 32 KB, utilizându-se o tehnologie de fabricație de 0,35 de microni.

Unele concepte deosebit de interesante cu privire la acest aspect au fost studiate la Universitatea Stanford cu MIPS-X, un produs derivat al arhitecturii MIPS ce avea o serie de caracteristici în plus. Multe dintre acestea au fost mai târziu introduse în procesorul comercial MIPS. Microprocesorul MIPS R2000 este un procesor pe 32 de biți cu o memorie cache de nivel 2, diferențiată pentru instrucțiuni și date. O memorie tampon de scriere ajută la manipularea tuturor datelor stocate în memorie. Produsul R2000 folosește o magistrală comună pentru memoria cache externă – o arhitectură non Harvard (arhitectura Harvard presupune utilizarea de magistrale diferite pentru instrucțiuni și pentru date).

Construcția acestui procesor înglobează o arhitectură radicală de coprocesor. Unitatea de control a întregilor din UCP este separată de așa numitul „Coprocesor de control al sistemului” (System Control Coprocessor), care este, de fapt, un controlor de memorie cache integrat direct pe cip UCP și unitatea de calcul în virgulă mobilă comunică prin intermediul memoriei. Microprocesorul înglobează 32 de regiștri generali și 16 regiștri (pe 64 de biți) separați pentru calcule în virgulă mobilă. Coprocesorul pentru calculul în virgulă mobilă conține o unitate pentru adunare, una pentru împărțire și una pentru înmulțire. Nu există biți de testare a condițiilor (indicatori de stare, sau *flags*, cum sunt denumiți la Intel). Programarea regiștrilor este controlată software.

### ***Procesoarele din seria SPARC (Sun Microsystems)***

Procesorul *SPARC (Scalable Processor ARChitecture)* se poate „lăuda” ca fiind descendentul unei familii ilustre de microprocesoare, aceea a procesoarelor RISC-I și RISC-II dezvoltate la Universitatea din Berkeley în anii 80. Această arhitectură a fost definită de firma Sun Microsystems și actualizată în permanență. Firma Texas Instruments a fost unul dintre principalii furnizori de cipuri ca urmare a unui contract cu firma Sun, unul dintre produsele anului 1995 fiind UltraSparc, cu o arhitectură pe 64 de biți și o implementare superscalară cu 4 canale.

O caracteristică importantă a arhitecturii este adăugarea de noi instrucțiuni pentru accelerarea graficii și a prelucrărilor video; astfel pot fi prelucrați până la 8 pixeli într-o singură instrucțiune sau ciclu de ceas. Dacă, în general, arhitectura acestui procesor este o arhitectură de tip RISC, există două “curiozități” ale acesteia, care îl disting în familia procesoarelor RISC.

În primul rând, SPARC utilizează conceptul de „ferestre de registre” (*register windows*) în scopul eliminării operațiilor de încărcare și stocare în stivă ce apar la apelurile de proceduri. Acest lucru putea fi însă obținut și prin programarea regiștrilor în momentul compilării. Echipa de la Berkeley a utilizat însă aceste ferestre de registre deoarece nu avea la momentul respectiv expertiza (pentru crearea compilatorului) necesară pentru a implementa alocarea interprocedurală a regiștrilor cu ajutorul software-ului (compilatorului).

În al doilea rând, o altă curiozitate a arhitecturii SPARC o reprezintă existența instrucțiunilor „etichetate” (*tagged instructions*). Se știe că limbaje de programare declarative de genul Lisp sau Prolog folosesc tipuri de date etichetate. Arhitectura SPARC utilizează instrucțiuni ce pot manipula cu ușurință o etichetă (în engleză *tag*) pe 2 biți în fiecare cuvânt de memorie. Această caracteristică putea mări viteza de execuție a unui program Lisp cu câteva procente.

Procesoarele UltraSPARC IV sunt procesoare ce suportă două fire de execuție (*chip multithreading*) pe procesor, bazate pe două stadii pipeline UltraSPARC III. Alte caracteristici:

- 66 milioane de tranzistori pe cip;
- pipeline cu 14 stadii;
- frecvența de ceas între 1.05-1.2 GHz;
- L1 cache de 64KB pentru date și 32 KB pentru instrucțiuni, 2KB Write, 2KB Pre-fetch;
- L2 cache de 16 MB;
- scalabilitate multiprocesor cu suport arhitectural până la 1000 de procesoare pe un singur sistem;
- controller-ul de memorie este capabil să adreseze până la 16 GB de memorie principală la o viteză de 2,4 GB/s.

Sun Microsystems Inc. este cel mai titrat producător de procesoare ce utilizează mulțimea de instrucțiuni Sparc, dar nu este singurul producător. Alt producător important este Fujitsu, ale cărui noi procesoare Sparc64 VI cu nume de cod Olympus vor veni pe piață în 2005, la viteze de peste 2,4 GHz, manufacturați în tehnologie de 90 de nm (nanometri). Performanța estimată a acestui procesor este de 4 ori mai mare decât a generației actuale de la Fujitsu, Sparc64 V, ce rulează la 1,35 GHz.

### ***Procesorul PA-RISC (Hewlett Packard)***

Arhitectura *PA-RISC (Precision Architecture)* a firmei Hewlett-Packard este destinată stațiilor de lucru performante, adoptând o linie nouă și modernă. Performanțele de operare în virgulă mobilă ale acesteia sunt excelente față de majoritatea competitorilor. S-au inclus noi instrucțiuni pentru funcții de accelerare a graficii și a procesărilor video, similare celor de la *SPARC*. Numărul de formate de instrucțiuni este mai mare decât la orice alt procesor RISC: sunt prezente nu mai puțin de 12 combinații diferite de *opcode* (coduri de operație) și regiștri sau câmpuri pentru constante într-un singur cuvânt (spre comparație, procesoarele *SPARC* și *MIPS* pot utiliza doar 4 combinații diferite).

Există în mod normal două moduri diferite de adresare, precum și încă două moduri adiționale ce oferă suport pentru operațiile ce au loc înainte sau după modificarea unui registru index. Acest lucru oferă posibilitatea utilizării în total a patru modalități de adresare.

Arhitectura *PA (Precision Architecture)* posedă coduri de operații (*opcode*) pe 6 biți. Acest lucru reduce numărul de instrucțiuni posibile la mai puțin de 64 (26), deși anumite instrucțiuni au mai multe variante, folosind biți speciali în cadrul formatului instrucțiunii. Numărul de regiștri generali este de 32, completați cu încă 32 de regiștri cu caracter special, utilizați pentru administrarea întreruperilor, a nivelurilor de protecție, etc. Caracteristica atipică a acestui procesor este aceea că implementarea execuției în *pipeline* se face pe doar 3 nivele, iar funcționarea optimă a conductei necesită programare software.

### ***Procesoarele PowerPC (IBM și Motorola)***

Suținut de firme puternice, ca IBM, Apple și Motorola, PowerPC este concurentul principal al microprocesoarelor bazate pe arhitectura x86. Principalul avantaj constă în posibilitatea rulării software-ului Apple, PC și Unix. Folosind tehnicile de recompilare binară, integrând și un emulator rapid pentru x86, procesorul PowerPC este capabil să utilizeze majoritatea sistemelor de operare și a software-ului într-un singur sistem.

### ***Procesoarele Alpha – DEC (Digital)***

Aceste procesoare se deosebesc de celelalte procesoare RISC prin frecvențele foarte mari ale ceasului intern, arhitectura modernă pe 64 de biți a acestora fiind una dintre cele mai performante de pe piață.

### ***Concluzii și viitorul procesoarelor RISC***

Fără îndoială, de la apariție și până în prezent, microprocesoarele RISC au avut performanțe superioare celor bazate pe arhitectura CISC. În ultimii ani însă, prin apariția microprocesoarelor Intel (și a produselor clonă) ce au preluat multe dintre conceptele tehnologiei clasice RISC, diferența dintre performanțele celor două tipuri arhitecturale s-a micșorat vizibil, ajungându-se ca cele mai noi procesoare Pentium III și Pentium 4 să concureze cu succes procesoarele RISC.

Se presupune că nu se va renunța foarte ușor în viitor la vechea arhitectură CISC (care are însă are și va prelua în continuare dintre beneficiile RISC), ajungându-se poate la situația în care vom putea cu greu să spunem care sunt deosebirile dintre cele două tipuri arhitecturale ce erau atât de diferite în trecut. Următoarele microprocesoare x86, precum cele bazate pe nucleele *Mustang și Sledgehammer* de la AMD sau *Pentium 4, Foster și Itanium* (ultimul pe 64 de biți) de la Intel vor reduce din ce în ce mai mult „gaura” ce desparte cele două variante tehnologice.

### 1.4.3 Memoria principală

Memoria principală are funcția de a stoca instrucțiunile programelor și datele asupra cărora acționează aceste instrucțiuni. De remarcat, un program pentru a se executa trebuie să se găsească în memoria principală. Fazele execuției unui program sunt:

- se selectează o instrucțiune din memoria principală;
- se încarcă instrucțiunea în microprocesor, are loc decodificarea ei pentru stabilirea operației care se va executa și identificarea operanzilor;
- se execută instrucțiunea;
- se repetă aceste etape în mod secvențial, pentru fiecare instrucțiune a programului.

Memoria are la bază un ansamblu de registre de aceeași mărime (o matrice de regiștri). Succesiunea de poziții binare ce este recunoscută și adresată individual se numește *locație de memorie* sau *cuvânt de memorie*.

Lungimea unui cuvânt de memorie este, în general, un multiplu de 8 biți (1 byte), putând avea: 8 biți, 16 biți, 32 biți, 64 biți, în funcție de modelul calculatorului.

Fiecare locație de memorie este identificată prin *adresa* sa. Procesorul selectează o anumită locație din memorie prin plasarea adresei sale pe *magistrala de adrese*.

Numărul total al locațiilor adresabile care pot fi accesate de procesor reprezintă *capacitatea memoriei* sau *spațiul de adresare fizică*.

### 1.4.4 Echipamente periferice

*Echipamentele periferice* sunt dispozitivele cuplate sau având posibilitatea de a fi cuplate la un calculator. După *funcția de bază*, acestea se pot clasifica în:

- ***Echipamente de intrare*** - având rolul de a capta și colecta informațiile, în forma lor uzuală, pentru a fi prelucrate în calculator. Din această categorie amintim: tastatura, mouse-ul, creionul optic, tableta digitală, scanner-ul, cititorul de coduri de bare (un scanner foarte simplu), cititorul de cartele magnetice/perforate, captatorul de sunete (microfonul), terminalul inteligent (terminalul pentru tranzacții financiare) etc.
- ***Echipamentele de ieșire*** - au rolul de a genera informația prelucrată în calculator într-o formă direct utilizabilă. Redarea în exterior a informației se poate obține în mod vizual, prin echipamentele periferice: ecran (de tip CRT - Cathode Ray Tube sau LCD - Liquid Crystal Display, monitoare monocrom sau color; CGA - Color Graphic Adapters, EGA - Enhanced Graphics Adapters; VGA - Video Graphic Adapter; XGA, XGA/HDA; VESA etc.), imprimantă (matricială, cu jet de cerneală, termică, laser), trasor sau *sonor*.
- ***Echipamente de memorare*** sau ***echipamente de intrare/ieșire*** - au rolul de a stoca o mare cantitate de informație, pentru un timp nedeterminat, în vederea utilizării ulterioare. Din această grupă de periferice fac parte unitățile de: discuri/benzi magnetice și discuri optice (Compact Disk-Read Only Memory, CD-ROM; Write Once, Read Many, WORM).
- ***Echipamente de transport la distanță*** au rolul de a oferi o formă adecvată informației ce se emite/recepționează prin suportul fizic (cablu electric, unde electro-magnetice, fibră optică). În această categorie este inclus modemul.

### 1.4.5 Magistrale

Una dintre componentele esențiale din structura unui sistem de calcul, alături de unitatea centrală de prelucrare, este *magistrala* („bus” în engleză). O magistrală sau un set de magistrale are rolul de a conecta procesorul la memoria principală și la echipamentele periferice. Este bine știut că magistrala are o importanță majoră în obținerea unor performanțe optime, ea asigurând viteza de lucru a sistemului de calcul. O magistrală bine aleasă permite calculatorului să lucreze la parametrii procesorului.

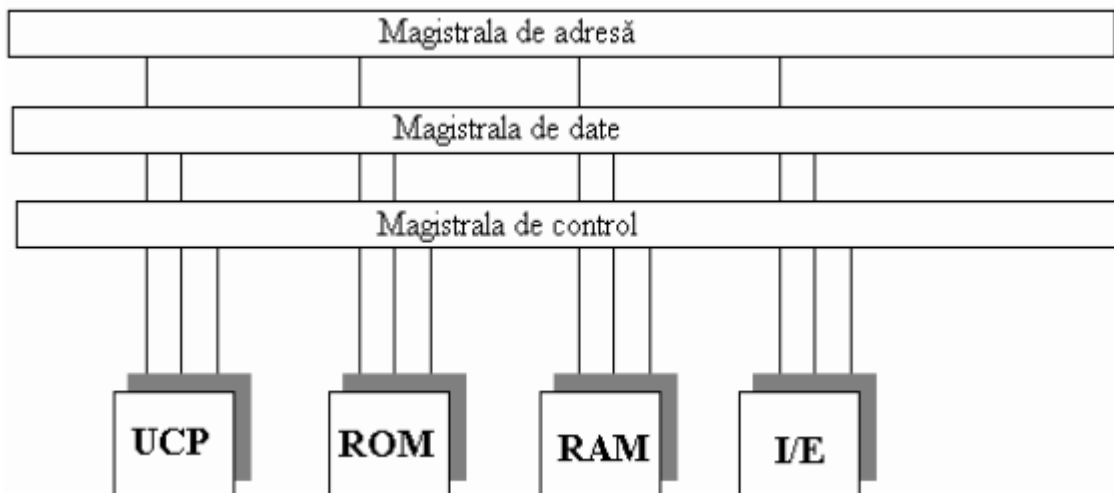
Primele PC-uri aveau numai o magistrală, care era comună pentru unitatea centrală de prelucrare (UCP), memoria RAM (Random Access Memory) și componentele de I/E (Intrare/Ieșire).

În anul 1987, firma Compaq a demonstrat pentru prima dată cum se poate separa magistrala de sistem de magistrala de I/E, astfel încât acestea să poată fi utilizate la parametri diferiți. Această arhitectură multi-magistrală a devenit standard industrial.

Magistralele unui sistem se pot diviza în:

- *magistrala de sistem* sau *magistrala locală* care conectează unitatea centrală de prelucrare (UCP) cu memoria RAM;
- *magistralele de I/E* care conectează UCP cu celelalte componente.

Fizic, o magistrală este reprezentată printr-un ansamblu de trasee de pe placa de circuit imprimat. Aceste trasee sunt utilizate pentru transmisia datelor, adreselor de memorie sau a unor semnale de control. Din aceste considerente, în literatura de specialitate se regăsesc denumirile de *magistrala de date*, *magistrala de adrese* și *magistrala de control*.



**Magistralele de date, adrese și control**

*Magistrala de adrese* este utilizată de procesor pentru a selecta o locație de memorie sau un anumit periferic.

*Magistrala de date* este utilizată pentru transferul datelor între procesor și unitatea de memorie sau un dispozitiv periferic.

*Magistrala de control* oferă semnalele pentru sincronizarea fluxului de date între procesor și unitatea de memorie sau un dispozitiv periferic.

O *arhitectură de magistrală* reprezintă modul în care componentele unui sistem de calcul, în particular ale unui PC, sunt interconectate. Principalele caracteristici care trebuie luate în calcul la alegerea unei arhitecturi de magistrală sunt:

- să asigure performanțe maxime microprocesorului;
- să fie operațională pe întreaga durată de viață a sistemului;
- să permită eventuala modernizare a microprocesorului;
- să permită includerea pe sistem a celor mai noi aplicații: multimedia, transfer de informație etc.

*Magistrala locală* asigură comunicarea între UCP și memoria RAM, eventual printr-o memorie de tip *cache*. Ea se află pe placa de bază și este proiectată în așa fel încât să corespundă specificațiilor microprocesorului. Tehnologia microprocesorului determină caracteristicile magistralei de sistem. În modelul inițial de PC, magistrala de sistem funcționa pe 8 biți și putea transfera aproximativ 106 octeți/secundă. Acum, pentru a putea face față unui procesor Pentium, este necesară o magistrală pe 64 biți, cu o viteză de transfer a datelor de  $5 \cdot 10^8$  octeți/secundă. În tabelele 1.1 și 1.2 sunt prezentate diferite microprocesoare și magistralele lor de sistem.

Caracteristici ale magistralei de sistem

UCP mai vechi	Lățimea magistralei de sistem	Viteza pe magistrala de sistem
8088	8 biți	4.77 MHz
8086	16 biți	8 MHz
80286-12	16 biți	12 MHz
80386SX-16	16 biți	16 MHz
80486SX-25	32 biți	25 MHz
80486DX-33	32 biți	33 MHz
80486DX2-50	32 biți	25 MHz
80486DX-50	32 biți	50 MHz
80486DX2-66	32 biți	33 MHz
80486DX4-120	32 biți	40 MHz
5X86-133	32 biți	33 MHz
Intel P60	64 biți	60 MHz
Intel P100	64 biți	66 MHz
Cyrix 6X86 P133+	64 biți	55 MHz
AMD K5-133	64 biți	66 MHz
Intel P150	64 biți	60 MHz
Intel P166	64 biți	66 MHz
Cyrix 6X86 P166+	64 biți	66 MHz
Pentium Pro 200	64 biți	66 MHz
Cyrix 6X86 P200+	64 biți	75 MHz
Pentium II	64 biți	66 MHz

*Magistralele de I/E* conectează UCP la toate celelalte componente ale sistemului și reprezintă extensii ale magistralei locale. Principalele tipuri de magistrale de I/E sunt: *PC AT, ISA, EISA, IBM Micro Channel, VESA Local Bus, PCI, SCSI, USB*.

Caracteristici ale magistralei de sistem

Microprocesor	Tip chipset	Viteza pe magistrala sistem	Viteza UCP
Intel Pentium II	82440BX82440GX	100 MHz	350, 400, 450 MHz
AMD K6-2	Via MVP3ALi Aladdin V	100 MHz	250, 300, 400 MHz
Intel Pentium Xeon	82450NX	100 MHz	450, 500 MHz
Intel Pentium III	82440JX	133 MHz	533, 665 MHz
AMD K7		200 MHz	600, 800 MHz

*Primul PC* produs de firma IBM (proiect demarat în 1980) folosea un procesor Intel 8088. Arhitectura magistralei de sistem, cât și cea de extensie reprezentau o continuitate a magistralei locale a procesorului. Pentru realizarea transferurilor de acces direct la memorie (*DMA – Direct Memory Access*), întreruperilor și funcțiilor de ceas/numărător, erau prevăzute cipuri speciale. Arhitectura era simplă și prezenta o mare disponibilitate de extensie. Drept urmare, plăcile adaptoare proiectate pentru primul PC pot fi utilizate pe calculatoarele actuale, dotate cu procesor Pentium și magistrală PCI. Sistemul suporta 1 sau 2 unități de dischete de 160 K și posibilitatea de a conecta încă două unități de dischete externe. Magistrala lucra pe 8 biți.

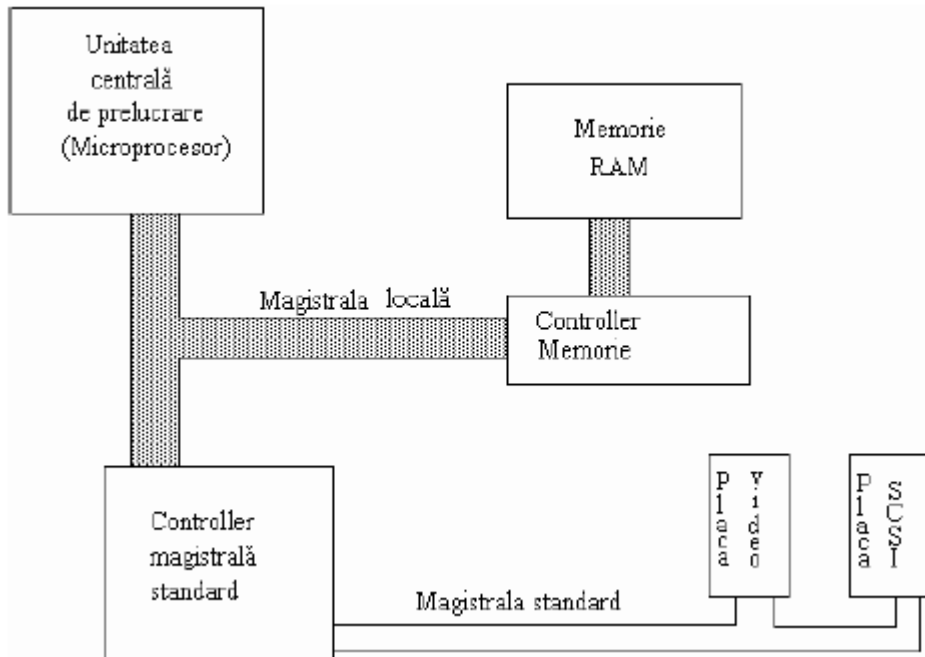
În 1984 IBM lansează pe piață *sistemul PC-AT*, având caracteristicile:

- procesor Intel 80286 (pe 16 biți);
- adresarea pe 24 biți;
- 16 MB memorie RAM;
- posibilitatea de lucru în mod protejat;
- frecvența ceasului de 6 MHz, cu posibilitatea de creștere la 8 MHz.

Noua magistrală pe 16 biți, cu adresare pe 24 biți, permitea utilizarea vechilor adaptoare fără modificări hardware sau software, datorită păstrării magistralei pe 8 biți și a conectorilor de extensie. Varianta PC-AT oferea trei canale DMA suplimentare și încă 7 nivele de întrerupere. În modelul original PC-AT, magistrala utiliza aceeași frecvență de ceas ca a microprocesorului. Ulterior, în multe sisteme care au apărut pe piață, magistralele de extensie lucrau la 10 sau 12 MHz. Pentru a rezolva problemele determinate de diferența de viteză între procesor și magistrală au fost create circuite logice ce asigură o funcționare asincronă (prin care se acceptă ca viteza procesorului să fie independentă de cea a magistralei).

*Arhitectura ISA (Industry Standard Architecture)* reprezintă un standard de „facto”, publicat de Intel, cu specificațiile magistralei PC-AT. Standardizarea nu a fost facilitată, deoarece IBM nu a făcut cunoscute caracteristicile magistralei sale. Arhitectura standard a microcalculatoarelor realizate până la sfârșitul anului 1992 cuprindea o magistrală locală capabilă să asigure o rată a transferului de până la 132 Mocteți/s (MBps). Magistrala locală conecta unitatea centrală de componentele care aveau rolul de control și nu se substituia magistralelor existente.

## Arhitectura standard a unui PC



Concurenții firmei IBM, *grupul celor nouă*: AST, Compaq, Epson, HP, Olivetti, NEC, Tandy, Zenith și Wyse, și-au unit forțele pentru a prezenta o arhitectură diferită: *EISA (Extended Industry Standard Architecture)* ce are avantajul de a rămâne compatibilă cu arhitectura ISA pe 16 biți. Magistrala EISA are un format pe 32 biți, atât pentru adrese cât și pentru date, permite accesul la 4 GB de memorie internă, iar rata transferului este de 33 MB/sec.

În anul 1987 IBM încearcă să recucerească controlul asupra arhitecturii sistemelor PC, pierdut odată cu publicarea detaliilor tehnice ISA și lansează pe piață produsele sale din familia PS/2 (Personal System/2) ce au la bază *arhitectura MCA (Micro Channel Architecture)*.

Arhitectura MCA este complexă, deoarece exploatează puterea microprocesoarelor pe 32 biți (80386, 80486 Intel). Magistrala de adrese este de asemenea pe 32 biți. Arhitectura este optimizată prin prezența a 8 canale DMA și posibilitatea de a conecta până la 16 echipamente periferice simultan. Magistrala MCA este asincronă și conține un program de identificare a plăcilor de extensie. Rata transferului de date 160 MB/sec. Există o magistrală MCA în format pe 16 biți pentru calculatoarele dotate cu microprocesoare 80286 Intel și se mai găsesc plăci video pe 24 biți adaptate la această arhitectură.

O dată cu crearea de către Microsoft a interfeței grafice Windows, este necesară o suprafață mai mare a ecranului precum și o rezoluție mai bună. Adaptoarele VGA atașate magistralei ISA erau depășite. Soluția a constat în conectarea adaptorului video și a memoriei asociate pe magistrala locală a sistemului. VESA a încercat să standardizeze monitoarele PC (diagonala ecranului, rata de înprospătare a imaginilor sau timpul de afișare pentru o imagine) magistrala și conectorii prin care se atașează echipamentele la magistrala locală a procesorului. Standardul VESA a ales ca etalon magistrala locală a microprocesorului Intel 80486.

*Magistrala VESA (Video Electronics Standard Association)* acceptă echipamente pe 32 și 64 biți, putând lucra la frecvențe mai mari 33 MHz în cazul formatului pe 32 biți.

*Magistrala PCI (Peripheral Component Interconnect)* a fost dezvoltată de Intel în 1993, la concurență cu standardul VESA. Versiunea 1.0 acceptă 32 biți la 33 Mhz, ceea ce permite să se atingă o rată de transfer de 132 MB/s, ca și în cazul magistralei VESA.

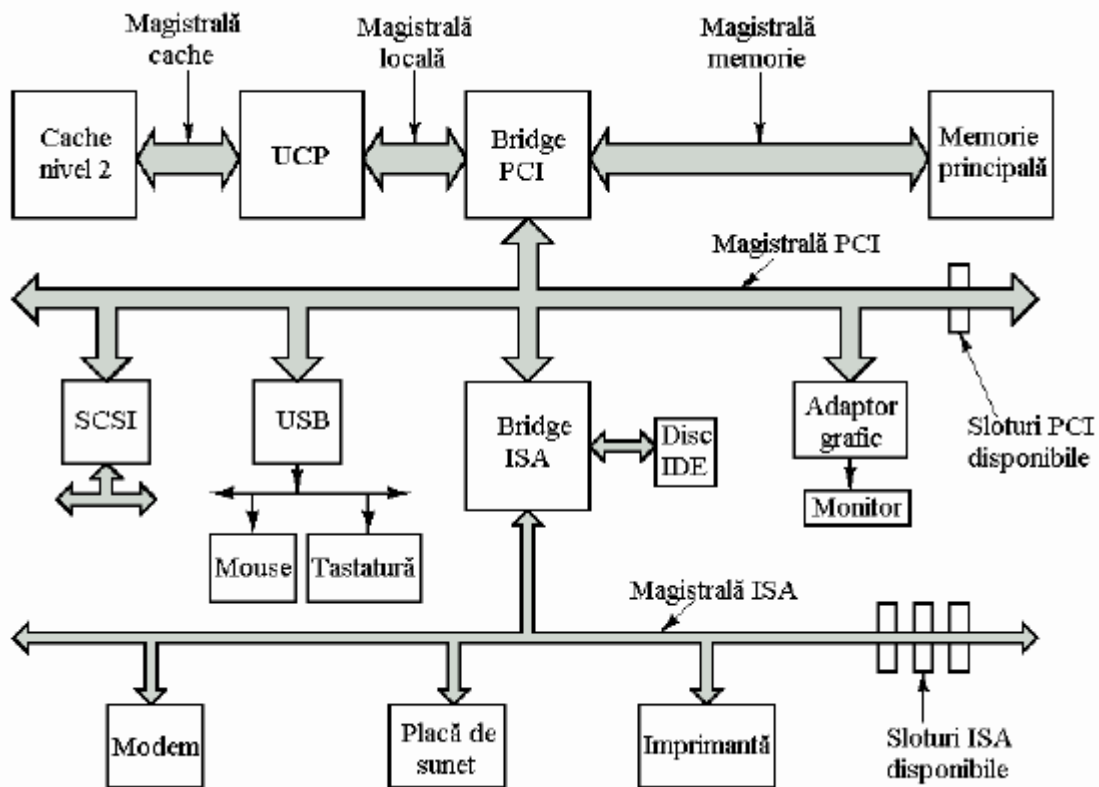
Magistrala PCI prezintă marele avantaj de a fi total independentă de procesor și dispune de propria memorie tampon. Arhitectura PCI poate fi combinată cu o altă arhitectură de magistrală, cum ar fi ISA sau EISA. PCI este autoconfigurabilă, plăcile conectate fiind automat detectate și utilizate în mod optim (Plug and Play). În specificația 2.0 PCI permite accesul pe 64 de biți, pentru a se putea utiliza cu microprocesorul Intel Pentium.

*Magistrala SCSI (Small Computer System Interface)* suportă diverse periferice. Viteza de transfer variază de la 4 Mocteți/s la 80 Mocteți/s.

*USB (Universal Serial Bus)* este o magistrală serie apărută în 1996 și care a devenit succesoarea magistrelor tradiționale. Ea permite utilizarea a 127 de periferice conectate la un singur canal. În plus, permite recunoașterea automată a perifericelor conectate pe canal și determinarea driver-ului necesar în funcționare. Pe o astfel de magistrală, informațiile codificate în *NRZI (Non Return to Zero Inverted)* pot circula la un debit adaptat perifericului (variază de la 1.5 la 12 Mocteți/s pe un cablu torsadat). USB utilizează principiile de funcționare similare celor din rețelele locale. În tabelul următor sunt ilustrate caracteristicile diferitelor tipuri de magistrale de I/E.

Magistrala de I/E	An apariție	Lățime magistrală	Viteza magistrală	Debit maxim (teoretic)
PC și XT	1980-82	8 biți	Sincronă cu UCP: 4.77 și 6 MHz	4-6 MBps
ISA (AT) Magistrală simplă	1984	16 biți	Asincronă: 8 MHz	8 MBps
MCA Magistrală inteligentă realizată de IBM	1987	32 biți	Asincronă: 10 MHz	40 MBps
EISA Magistrală inteligentă pentru servere.	1988	32 biți	Asincronă: 8 MHz	32 MBps
VL Magistrală de mare viteză, utilizată în 486s	1993	32 biți	Sincronă cu UCP: 33, 40, 50 MHz	100-160 MBps
PCI Magistrală inteligentă de mare viteză	1993	32 biți	Asincronă: 33 MHz	132 MBps
USB Magistrală inteligentă, simplă și modernă	1997		Serială	1.2 MBps
FireWire (IEEE1394) Magistrală inteligentă de mare viteză pentru video, memorare etc.	1999		Serială	80 MBps

## Arhitectura tipică PCI a unui sistem cu procesor Pentium II



### 1.4.6 Arhitecturi paralele

Viitorul arhitecturilor de calculator este în strânsă concordanță cu noțiunea de paralelism. Acest principiu arhitectural permite tratarea în paralel a informației ce presupune execuția evenimentelor concurente. Aceste evenimente se pot regăsi la nivel de program, de procedură, instrucțiune sau în interiorul unei instrucțiuni. Ca o definiție foarte simplă, un calculator paralel reprezintă o colecție de procesoare interconectate între ele pentru a permite coordonarea activităților acestora și schimbul de date. Calculatoarele paralele necesită algoritmi paraleli, adică algoritmi ce pot fi implementați pe calculatoare paralele.

La nivelul cel mai înalt, tratarea paralelă permite execuția simultană a mai multor programe independente. Se utilizează în sistemele mari de tip mainframe și se tratează la nivelul sistemului de operare (multiprograme, timp partajat, multiprelucrare).

Tratarea paralelă a instrucțiunilor independente utilizează tehnica de vectorizare. Se tratează la nivel de sistem (vectorial), de limbaj de programare (Fortran vectorial) sau la nivel de algoritm.

Tehnica denumită *pipeline* permite introducerea paralelismului la nivel de instrucțiune. În acest sens, o instrucțiune este împărțită în mai multe etape succesive și se execută în același timp etape diferite ale mai multor instrucțiuni.

Arhitecturile monoprocesor au, în general, o structură de bază comună: o memorie principală, procesorul central și un ansamblu de echipamente periferice. Multe calculatoare monoprocesor fac apel la tehnica tratării paralele, dar există limitări.

Paralelismul se poate realiza:

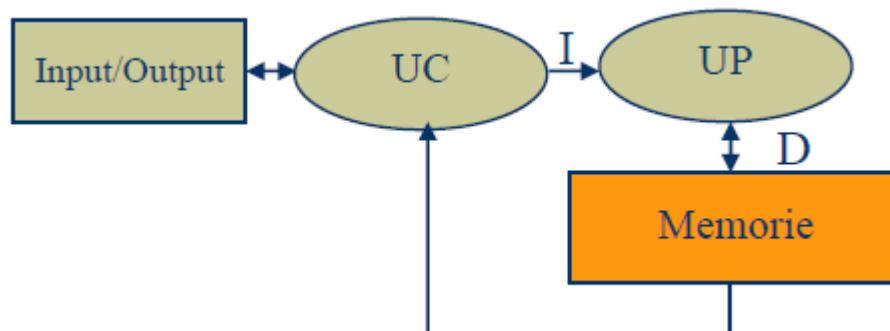
- între mai multe unități funcționale;
- între UAL paralele;
- prin multiprogramare și timp partajat.

Condițiile paralelismului trebuie să țină seama de *restricțiile* inerente în tratarea informației (algoritmare) și de *posibilitățile* oferite de componentele fizice (multiplicarea structurilor de execuție). Creșterea numărului de procesoare în același calculator, modifică structura de bază a acestuia. Problemele de acces la memorie devin foarte importante, pentru a putea transmite datele în ritmul de tratare al procesoarelor. De asemenea, problemele de comunicare între procesoare sunt esențiale.

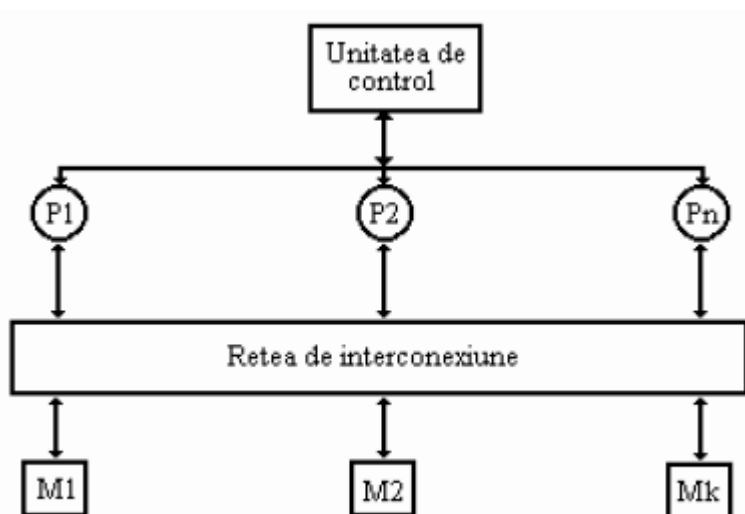
Deoarece procesul principal într-un calculator constă din execuția unei succesiuni de instrucțiuni asupra unui ansamblu de date, arhitecturile paralele pot fi clasificate după fluxurile de instrucțiuni și date (după Flynn) astfel:

- **SISD** - Single Instruction (Stream), Single Data (Stream);
- **SIMD** - Single Instruction (Stream), Multiple Data (Stream);
- **MISD** - Multiple Instruction (Stream), Single Data (Stream);
- **MIMD** - Multiple Instruction (Stream), Multiple Data (Stream).

Arhitectura SISD



Arhitectura SIMD



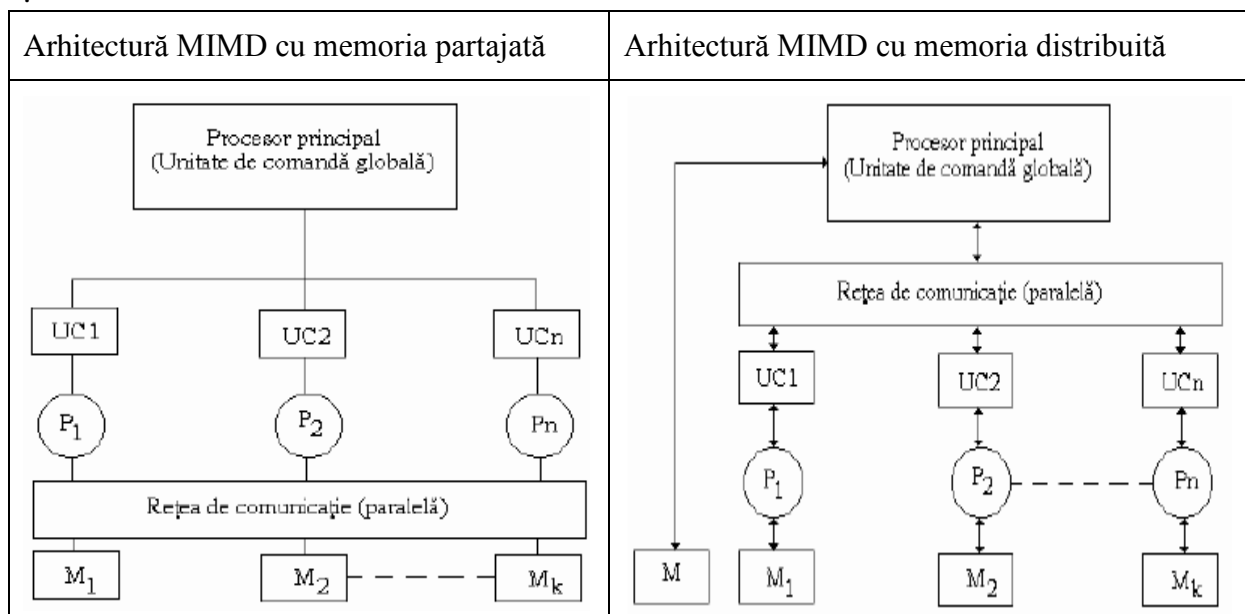
**SISD** (fluxuri unice pentru instrucțiuni și date) reprezintă structura de bază a unei mașini secvențiale (numită mașină Von Neumann). Exemple: IBM 370, DEC VAX, SUN, IBM PC, Macintosh.

**SIMD** (flux unic de instrucțiuni, flux multiplu de date) conține mai multe unități de prelucrare ce sunt supravegheate de o singură unitate de control. Toate unitățile de tratare primesc aceeași instrucțiune (sau același program în cazul arhitecturii **SPMD** (Single Program, Multiple Data) transmisă prin unitatea de control, dar operează pe ansambluri de date distincte.

Fiecare unitate de prelucrare  $P_i$ , execută aceeași instrucțiune în același moment, obținându-se o funcționare sincronă a procesoarelor. Memoria poate fi împărțită în mai multe module,  $M_j$ . În această situație, accesul unităților de prelucrare la diferitele module de memorie se face printr-o rețea de interconexiune. Această arhitectură include și *procesoarele vectoriale* și *rețelele sistolice*. Exemple: CRAY-1, DAP CM-1, WARP, CM-2, ILLIAC IV.

**MIMD** (fluxuri multiple de instrucțiuni, fluxuri multiple de date), permite realizarea unui paralelism explicit. Exemple de astfel de calculatoare sunt: transputerele, Supernode, DADO, N-cube, Ultracomputer, Butterfly, Alliant, Sequent Balance, CRAY X-MP.

După cum memoria centrală este comună pentru mai multe procesoare sau mai multe unități centrale (procesor + memorie centrală proprie) ce comunică între ele prin mesaje, se poate spune că structura este puternic cuplată, *MIMD cu memorie partajată* sau slab cuplată, *MIMD cu memorie distribuită*



În primul model, procesoarele execută sarcinile prin partajarea datelor din memoria comună, ce este divizată în module ( $M_i$ ). În al doilea model, procesoarele execută sarcinile schimbând mesaje. În ambele modele fiecare procesor ( $P_i$ ) posedă propria unitate de control ( $UC_i$ ).

În viitor se prevede o revenire a constructorilor la arhitecturi mai puțin complexe, având la bază procesoare care integrează o mai mare putere de calcul și facilități de comunicație. Alternativa „distribuită” constituie varianta cea mai potrivită din punct de vedere economic.