Arhitectura Calculatoarelor

T.3 –Tipuri de arhitecturi ale calculatoarelor numerice

Tipuri de arhitecturi ale calculatoarelor numerice. Arhitectura von Neumann, Maşina Turing. Clasificarea Calculatoarelor. Taxonomia Flynn, Taxonomia lui Wang, Clasificare comercială.

Scopul Lecţiei: De a face cunoştinţă cu tipurile de arhitecturi a sistemelor de calcul şi clasificarea sistemelor de calcul după diferite elemente specifice.

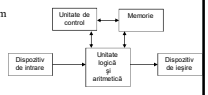
Studentul trebuie *să cunoască:*
- ***Arhitectura von Neumann; Maşina Turing;***
- ***Clasificarea calculatoarelor după Flynn, Wang. Clasificarea Comercială***
- ***Trendul în arhitectura calculatoarelor***

Conf. Univ. Dr. Creţu Vasilii

Arhitectura von Neumann

Într-un articol publicat în 1947, John von Neumann a expus nişte principii care stau la baza calculatoarelor moderne. Acestea sunt:
- Existenţa unui mediu de intrare prin intermediul căruia să poată fi introdus un număr practic nelimitat de date şi instrucţiuni.
- Existenţa unei memorii în care să fie depuşi operanzii şi instrucţiunile şi de unde să fie preluate rezultatele în ordinea dorită.
- Existenţa unei secţiuni de calcul care să fie capabilă să execute operaţii aritmetice şi logice asupra datelor din memorie.
- Existenţa unui mediu de ieşire prin intermediul căruia să poată fi comunicat utilizatorului un număr nelimitat de instrucţiuni.
- Existenţa unei unităţi de comandă capabilă să interpreteze instrucţiunile citite în memorie şi, pe baza informaţiilor citite în memorie şi a informaţiilor furnizate de secţiunea de calcul, să fie capabilă să decidă între mai multe variante de desfăşurare a operaţiilor.
- Datele şi instrucţiunile trebuie să fie stocate în memorie sub aceeaşi formă.

Programul stocat este cel mai important bloc al modelului von Neumann. Un program este stocat în memoria calculatorului împreună cu datele ce sunt procesate. Înainte de apariţia calculatoarelor cu program stocat, programele erau stocate pe medii externe cum ar fi cartele perforate. În calculatorul cu program stocat acesta poate fi manipulat ca şi cum ar reprezenta date. Aceasta a dus la apariţia compilatoarelor şi sistemelor de operare şi face posibilă marea versatilitate a calculatoarelor moderne.



Ca urmare, caracterizarea arhitecturii von Neuman se face prin :
- utilizarea memoriei interne pentru a stoca secvenţe de control pentru îndeplinirea unei anumite sarcini – secvenţe de programe;
- datele, cât şi instrucţiunile sunt reprezentate ca siruri de biti şi sunt stocate într-o memorie readwrite;
- conţinutul memoriei se poate accesa în funcţie de locatie (adresa), indiferent de tipul informatiei conţinute;
- execuţia unui set de instrucţiuni se efectueaza secvenţial, prin citirea de instrucţiuni consecutive din memorie.

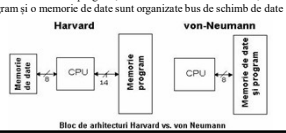
Arhitectura Harvard şi avantajele sale

Atunci când se implementează microprocesoare, se folosesc în mod tradiţional două abordări pentru construirea unei arhitecturi:
- Arhitectura lui Von Neumann
- Arhitectura Harvard

Arhitectura von Neumann utilizează o memorie omogenă a microprocesorului. Pot fi scrise diferite programe în această memorie. În acest caz, un program special de încărcare funcţionează împreună cu ele ca date. Apoi, controlul poate fi transferat la aceste programe şi ei încep deja să-şi execute algoritmul. Cu această abordare a controlului cu microprocesor, este posibil să se obţină o flexibilitate maximă a sistemului microprocesor.

Dezavantajul arhitecturii von Neumann este posibilitatea unei încălcări neintenţionată a performanţei sistemului (erori de soft) şi distrugerea intenţionată a funcţionării sale (atac viral). În arhitectura Harvard două tipuri de memorie microprocesor sunt fundamental diferite:
- Memorie program (pentru stocarea instrucţiunilor microprocesorului)
- Memorie de date (pentru stocarea temporară şi prelucrarea variabilelor)

Arhitectura Harvard este fundamental imposibil de a efectua o operaţie de scriere în memoria de program, care elimină posibilitatea distrugerii accidentale a programului de control în cazul erorilor de program, atunci când se lucrează cu date, sau atacuri de către terţe părţi. Mai mult, pentru o memorie de program şi o memorie de date sunt organizate bus de schimb de date individuale (magistrala de sistem).



Bloc de arhitecturi Harvard vs. von Neumann

Aceste caracteristici au determinat domeniul arhitecturii Harvard. Arhitectura Harvard este folosită în microcontrolere și în procesoare de semnale. unde este necesară asigurarea unei fiabilități ridicate a echipamentului. Semnalul de procesoare arhitectura de la Harvard completat folosind trei-bus unitate de microprocesor de operare. Arhitectura trei bus permite unității de operare pentru a combina operațiunile de citire două operanzi cu rezultate record ale comenzii în memoria microprocesorului. Acest lucru sporește foarte mult performanța microprocesorului de semnal fără a crește frecvența ceasului.
Structura modificată Harvard este utilizată în jetoanele moderne ale procesoarelor de semnal. Chiar și în continuare pe calea reducerii costului cristalului prin reducerea zonei ocupate de magistralele de sistem a mers la un singur cristal producătorii de calculatoare - microcontrolere. Aceste cipuri folosesc un sistem bus pentru transmiterea de comenzi și date (arhitectura Harvard modificata) si in interiorul cristalului.
Procesoarele de semnalizare necesită deseori mai multe buse interne pentru a implementa algoritmi precum transformarea rapidă Fourier și filtrarea digitală. De obicei, sunt utilizate două magistrale de date, o magistrală de scriere a datelor și o magistrală de citire a instrucțiunilor. O astfel de structură a microprocesorului a fost numită arhitectura extinsă a lui Harvard. Această abordare practică producătorii de semnal procesoare - firma Analog Devices (familie de procesoare de semnal Blackfin și Tiger Shark), Texas Instrunents (familia C5000 ™ DSP-uri de procesoare de semnal și C6000 ™ DSP-uri), Freescale (familia DSP56K MSC8251 și procesoare de semnal).

Cum funcționează arhitectura Harvard?

Arhitectura Harvard are diferite zone de adresă de memorie pentru program și pentru date.
Acest lucru are ca rezultat capacitatea de a proiecta un circuit în așa fel încât o magistrală și un circuit de control pot fi utilizate pentru a gestiona fluxul de informații din memoria programului și unul separat pentru a gestiona fluxul de informații către memoria de date.
Utilizarea de magistrale separate înseamnă că este posibil ca un program să fie recuperat și executat fără a fi întrerupt de transferul ocazional de date în memoria de date.
De exemplu, într-o versiune simplă a acestei arhitecturi, unitatea de recuperare a programului ar putea fi ocupată cu recuperarea următoarei instrucțiuni din secvența programului și în paralel efectuarea unei operații de transfer de date care ar fi putut face parte din instrucțiunea programului anterior. .
La acest nivel, arhitectura Harvard are o limitare, deoarece în general nu este posibil să plasați codul programului în memoria de date și să îl executați de acolo.

Adăugări în arhitectură

Multe variante existente mai complicate pot fi adăugate la forma simplă a arhitecturii Harvard.
O adăugare obișnuită este adăugarea unui cache de instrucțiuni la magistrala de date a programului, care permite unității de execuție a instrucțiunilor un acces mai rapid la următorul pas din program, fără a fi nevoie să mergeți la o memorie mai lentă pentru a ajunge la pas. programului ori de câte ori este necesar.

Adrese de memorie

Un computer Harvard are zone de adresă de date și instrucțiuni diferite: adresa de instrucțiuni una nu este aceeași zonă cu adresa de date una.
Adresa de instrucțiuni 1 ar putea conține o valoare de douăzeci și patru de biți, în timp ce adresa de date una ar putea indica un octet de 8 biți, care nu face parte din acea valoare de douăzeci și patru de biți.

Sistem de memorie

Deoarece există o zonă de memorie separată pentru instrucțiuni și date, separând atât semnalele, cât și memoria de stocare a codului și a datelor, acest lucru face posibilă accesarea simultană a fiecărui sistem de memorie.

Avantaj

- Există mai puține șanse de corupție în transmisie, deoarece datele și instrucțiunile sunt transferate prin diferite autobuze.
- Datele și instrucțiunile sunt accesate în același mod.
- Permite diferite medii de stocare pentru instrucțiuni și date. De exemplu, puteți pune instrucțiunile într-un ROM ieftin și datele în RAM scump.
- Cele două amintiri pot utiliza dimensiuni de celule diferite, făcând astfel o utilizare eficientă a resurselor.
- Are o lățime de bandă de memorie mai mare, care este mai previzibilă având memorii separate pentru instrucțiuni și date.

Nivel de protecție

Pe sistemele care nu au o unitate de gestionare a memoriei, oferă un nivel suplimentar de protecție, deoarece datele nu pot fi pornite ca cod, expunând sistemul la multe probleme, cum ar fi depășirea bufferului.
De aceea este popular cu sistemele încorporate mici, cum ar fi un cuptor cu microunde sau un ceas.

Viteză mai mare

Arhitectura Harvard poate citi o instrucțiune și, de asemenea, accesa memoria de date simultan cu o viteză rapidă.
Oferă performanțe mai mari, deoarece permite obținerea simultană a datelor și instrucțiunilor pentru a fi stocate în amintiri separate și a călători prin diferite autobuze.
O arhitectură Harvard va ajuta, în general, un computer cu un anumit nivel de complexitate să funcționeze mai repede decât o arhitectură Von Neumann, atâta timp cât nu este necesar să partajați resursele între cod și memoriile de date.
Dacă limitările pinului sau alți factori forțează utilizarea unei singure magistrale pentru a accesa ambele spații de memorie, este posibil ca aceste beneficii să fie anulate în mare măsură.

Dezavantaje

Complexitate și costuri mai mari

Problema arhitecturii Harvard este complexitatea și costul său mare, deoarece în loc de o magistrală de date, acum sunt necesare două.
Producerea unui computer cu două autobuze este mult mai scumpă și consumă mult timp. Este nevoie de o unitate de control pentru două autobuze, care este mai complicată și consumă mult timp și este costisitoare de dezvoltat.
Aceasta înseamnă o implementare mai complexă pentru producători. Necesită mai mulți pini pe CPU, o placă de bază mai complexă și trebuie să duplice cipurile RAM, precum și un design cache mai complex.

Utilizare redusă

Arhitectura Harvard nu este utilizată pe scară largă, ceea ce face mai dificilă implementarea. Acesta este motivul pentru care este rar folosit în afara procesorului.
Cu toate acestea, aceastăarhitectură este uneori utilizată în CPU pentru a-și gestiona cache-urile.

Utilizarea greșită a spațiului de memorie

Când există spațiu liber în memoria de date, acesta nu poate fi folosit pentru a stoca instrucțiuni și invers.
Prin urmare, amintirile speciale dedicate fiecăruia dintre ele trebuie să fie atent echilibrate în fabricarea lor.

Maşina Turing

În 1936, matematicianul englez Allan Turing a creat un automat abstract care să opereze cu numere calculabile. Un număr calculabil este un număr a cărei parte zecimală poate fi determinată cu un număr finit de iteraţii.

Automatul a fost sintetizat pe baza următoarelor ipoteze:

a)Automatul are un număr **n** finit de stări.

b)Automatul se află în orice moment într-o stare **i, 1≤i≥n,** urmând ca în momentul imediat următor să se afle în starea **j, 1≤j≥n.**

c)Fiecare din cele **n** stări se caracterizează prin:

-valoarea caracteristică e_i, care este o valoare curentă a numărului ce se calculează;

-funcţia f_j care aplicată stării e_i permite obţinerea următoarei stări e_j .

-deplasamentul d_{ij} care va trebui aplicat numărului pentru a se realiza din starea **i** în starea **j** , adică $j=i+d_{ij}$

-*Sistemul reprezentativ (SR)* sau memoria maşinii este construit dintr-o bandă magnetică de lungime practic infinită, împărţită în segmente de lungime egală, fiecare segment putând stoca un număr finit de semne.

-*Procesorul* (P) este un circuit secvenţial cu un număr finit de stări, care poate executa următoarele instrucţiuni, (setul de instrucţiuni al maşinii):

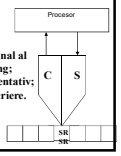
-schimbă segmentul de pe bandă de la poziţia curentă;

-poziţionează capul de citire (C) cu o poziţie la dreapta;

-poziţionează capul de citire cu o poziţie la stânga;

-opreşte sistemul.

Pentru a realiza un calcul cu această maşină, se înscriu datele într-un mod convenabil şi se descompune algoritmul de calcul, în funcţie de modul de reprezentare a datelor, într-o secvenţă de instrucţiuni ale maşinii. Ultima instrucţiune este cea de oprire a maşinii.



Modelul funcţional al maşinii Turing; SR-sistem reprezentativ; C/S-cip citire/scriere.

Maşina Turing este compusă din următoarele piese:

• o bandă infinită de hîrtie cu pătrăţele; în fiecare pătrăţel se poate scrie exact un caracter din alfabetul nostru; banda este iniţial plină cu ``spaţii", mai puţin o parte (să-i spunem de început) unde este scris şirul cu datele de intrare;

• un cap de citire-scriere, care se poate mişca deasupra benzii, la stînga sau la dreapta;

• o unitate de control, care conţine un număr finit de reguli care indică maşinii ce să facă la fiecare mişcare în funcţie de litera curentă de pe bandă şi starea în care maşina se află. Unitatea de control se află, în fiecare moment dat, într-o stare; stările posibile sunt fixate dinainte, şi sunt în număr finit.

Fiecare regulă are forma următoare:

Dacă

• sunt în starea s1;

• sub capul de citire este litera a;

atunci:

• trec în starea s2;

• scriu pe bandă litera b;

• mut capul de citire/scriere în direcţia D.

Şi asta-i tot! Orice algoritm de calcul poate descris de o astfel de maşină, prin toate stările posibile, şi toate aceste reguli, numite reguli de tranziţie, care indică cum se trece de la o stare la alta. În pofida simplităţii, maşina Turing poate deci calcula orice se poate calcula cu cele mai performante supercomputere

Clasificarea Calculatoarelor

Este destul de dificil să se clasifice tipurile de calculatoare din cauza multitudinii lor. Totuşi, anumite taxonomii s-au impus. Vom prezenta trei clasificări după arhitectură şi una strict comercială.

Taxonomia Flynn

A fost publicată în 1966 de către M. J. Flynn care avea în vedere existenţa într-un sistem de calcul a două fluxuri:

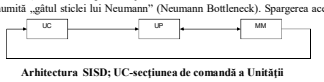
-fluxul de instrucţiuni, care reprezintă programul;

-fluxul de date, care reprezintă datele de intrare sau rezultatele parţiale.

Clasificarea lui Flynn ia în consideraţie gradul de multiplicitate ale celor două fluxuri şi identifică patru tipuri de arhitecturi:

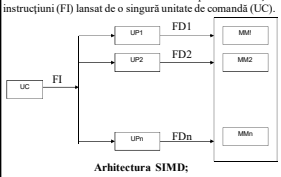
SISD (Single Instruction Single Data)
SIMD (Single Instruction Multiple Data)
MISD (Multiple Instructions Single Data)
MIMD (Multiple Instructions Multiple Data)

SISD realizează o execuţie sequenţială a instrucţiunilor. Principalul neajuns al acestei arhitecturi este viteza de execuţie care , la un moment dat, este plafonată, situaţie denumită „gâtul sticlei lui Neumann" (Neumann Bottleneck). Spargerea acestei limitări este realizată prin arhitectura paralelă.



Arhitectura SISD; UC-secţiunea de comandă a Unităţii centrale; UP-secţiunea de prelucrare a Unităţii centrale; MM-modul de memorie; FD-flux de date; FI-flux de instrucţiuni.

Arhitectura **SIMD** se caracterizează prin n unităţi de procesare (UP) care operează sub controlul unui flux unic de instrucţiuni (FI) lansat de o singură unitate de comandă (UC).



Arhitectura SIMD;
UC-secţiunea de comandă a Unităţii centrale; UP-secţiuni de prelucrare a Unităţii centrale; MM-moduri de memorie; FD-flux de date; FI-flux de instrucţiuni.

Cele mai cunoscute maşini SIMD sunt calculatoarele vectoriale. Acestea transformă instrucţiuni care se execută în n paşi într-o maşină SIMD, în instrucţiuni care se execută într-un singur pas. De exemplu, suma a doi vectori: c[i]=a[i]+b[i] pentru i=1…n se face într-un pas, fiecare Unitate de Prelucrare calculând o componentă a vectorului sumă.

MISD nu are nici un sens şi de aceea nu este utilizată.

MIMD cuprinde două feluri de maşini:
-multiprocesoare şi
-multicalculatoare.

Multiprocesoarele se caracterizează prin existenţa memoriei comune la care au acces n procesoare. Schimbul de informaţie dintre procesoare se face prin variabilele partajate din memoria comună la care au acces toate procesoarele, însă accesul trebuie făcut prin excludere mutuală pentru a realiza ceea ce se numeşte consistenţa memoriei.

Multicalculartoarele se caracterizează prin existenţa unui număr foarte mare de calculatoare (de la ordinul sutelor în sus) care sunt legate printr-o reţea topologică. Fiecare procesor are memoria lui locală, văzută doar de el, iar comunicarea între procesoare se face prin mesaje.

Taxonomia lui Wang (Feng) (Цзе-юнь Фэн)

Această clasificare presupune o organizare matricială a datelor. O matrice de dimensiunea m x n are m cuvinte, fiecare cuvânt are o lungime de n biţi. Criteriul de clasificare este gradul de paralelism în procesarea datelor organizate matricial. Conform acestui criteriu există patru tipuri de arhitecturi şi anume:

-**WSBS** (Word Serial-Bit Serial) în care se lucrează pe un singur cuvânt, fiecare cuvânt fiind prelucrat serial bit cu bit, ns1, ms1….

-**WSBP** (Word Serial-Bit Paralel) în care se lucrează pe un singur cuvânt , biţii fiecărui cuvânt fiind prelucraţi simultan, n>1, ms1.

WPBS (Word Paralel-Bit Serial) în care se lucrează pe un singur bit la toate cuvintele simultan, ns1, m>1.

-**WPBP** (Word Paralel-Bit Paralel) în care se lucrează simultan pe toate cuvintele şi pe toţi biţii, n>1, m>1.

WSBS nu are elemente de paralelism.

WSPB şi WPBS sunt parţial paralele, fiind orientate pe prelucrarea vectorilor.

WPBP este complet paralel.

Taxonomia lui Shore

Spre deosebire de Flynn, Shore şi-a bazat clasificarea pe modul în care este organizat calculatorul din părţile sale componente. Din acest punct de vedere, au fost identificate şase tipuri de maşini, fiecăreia atribuindu-se o cifră romană.

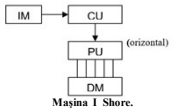
Maşina I are o arhitectură convenţională von Neumann, cu următoarea structură:
-unitate de comandă (CU)
-unitate de procesare (PU)
-memorie pentru instrucţiuni (IM)
-memorie pentru date (DM)

O citire a DM produce toţi biţii unui cuvânt, care sunt prelucraţi în paralel de PU, PU putând conţine mai multe unităţi funcţionale.

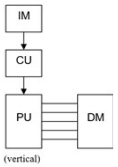
Această clasă include calculatoare vectoriale, de exemplu CRAY 1.

Maşina II este similară maşinii I, cu deosebirea că, în timp ce maşina I citeşte slice-uri orizontale, maşina II citeşte un slice vertical.

Exemple de calculatoare de tip maşina II Shore : ICL, DAP, STARAN etc.



Maşina I Shore.



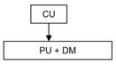
Maşina II Shore

Maşina III este o combinaţie a maşinilor I şi II. Un exemplu de astfel de maşină este calculatorul ortogonal Shooman(1970)

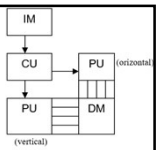
Masina IV se obţine prin multiplicarea unităţilor PU şi DM din maşina I şi prin trimiterea acestui ansamblu de la o singură unitate de control UC. Exemplu: PEPE.

Maşina V este exact maşina IV cu facultatea suplimentară că unităţile PU sunt aşezate pe o linie şi se asigură conexiuni între vecinii cei mai apropiaţi; fiecare PU poate adresa informaţii din memoria sa dar şi din cea a vecinilor săi imediaţi. Este un masiv conectat. Exemplu: ILIA CIV.

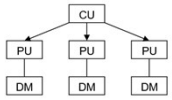
Maşina VI este denumită maşina cu logica în memorie. Este o abordare alternativă a distribuirii comenzii în memorie.
Exemplu: calculatoare cu memorii asociative.



Maşina VI Shore

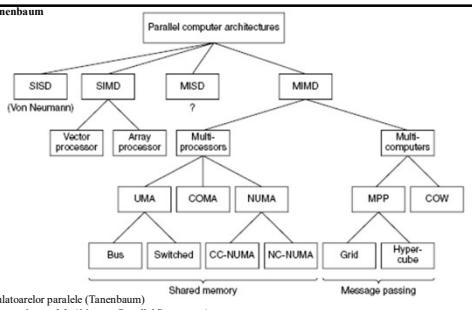


Maşina III Shore



Maşina IV Shore

Clasificarea Tanenbaum



Clasificarea calculatoarelor paralele (Tanenbaum)
MPP = procesoare masiv paralele (*Massive Parallel Processors*)
COW = cluster de staţii de lucru (*Cluster Of Workstations*)

Clasificare comercială

Dacă primele trei clasificări erau strict legate de arhitectură, clasificarea următoare are ca punct de vedere piața de calculatoare. Sistemele de calcul pot fi:

-calculatoare personale;
-servere;
-sisteme dedicate.

Calculatoarele personale sunt cele mai populare. Au cel mai mic cost și în ultimii ani s-au produs 150-200 milioane pe an. Prețul lor nu depășește suma de 10000 dolari.

Serverele sunt destinate să ofere servicii tot mai sofisticate de rețea. Costul lor este de 10.000 – 10.000.000 dolari. În ultimii ani s-au produs aproximativ 4 milioane servere pe an.

Sistemele dedicate sunt construite pentru anumite aplicații speciale. Costul lor este de 10.000-100.000 dolari.

CLASIFICAREA CALCULATOARELOR

CLASIFICAREA PE GENERAȚII

PERIOADA	mijlocul anilor 1940 – anii '50	anii 1960	anii 1970	anii 1980	
GENERAȚIA	PRIMA	A DOUA	A TREIA	A PATRA	INTELIGENȚA ARTIFICIALĂ
BAZA DE ELEMENTE	TUBURI ELECTRONICE	TRANZISTORI	CIRCUITE INTEGRATE	CIRCUITE INTEGRATE PE SCARĂ LARGĂ ȘI FOARTE LARGĂ	
VITEZA DE OPERARE	ZECI DE MII DE OPERAȚII PE SECUNDĂ	SUTE DE MII DE OPERAȚII PE SECUNDĂ	MAI MULT DE UN MILION DE OPERAȚII PE SECUNDĂ	ZECI ȘI SUTE DE MILOANE DE OPERAȚII PE SECUNDĂ	
SOFTWARE	LIMBAJE COD MAȘINĂ	LIMBAJE DE PROGRAMARE DE NIVEL ÎNALT	PROGRAME APLICATIVE ȘI DE REȚEA	INTERFEȚE GRAFICE	

CLASIFICAREA DUPĂ PERFORMANȚE

CLASE	MICROCALCULATOARE	MINICALCULATOARE	CALCULATOARE MARI (macrocalculatoare)	SUPERCALCULATOARE
CARACTERISTICI GENERALE	• Procesorul central este un microprocesor realizat pe un cip; • Exploatare autonomă; • Viteza de operare: sute de Mips (milioane de instrucțiuni pe sec.); • Memorie RAM: până la câțiva Gb.	• Până la zeci de procesoare; • Deservesc de la zeci până la sute de utilizatori; • Viteza de operare: până la zeci de Mips (milioane de instrucțiuni pe sec.); • Memoria RAM: până la câțiva Gb.	• Până la sute de procesoare; • Deservesc o sute sau mii de utilizatori; • Viteza de operare: de ordinul sutelor sau miilor de Mips (milioane de instrucțiuni pe sec.); • Memoria RAM: până la zeci de Gb.	• Structură multiprocesor (până la zeci de mii de procesoare) permite procesarea paralelă; • Viteza de operare: până la zeci de Tflops (fiop – operații în virgulă mobilă pe secundă); • Memoria RAM: până la zeci de Tb.
DOMENII DE UTILIZARE	Sunt incorporate în diverse aparate casnice, tehnică de birou, sisteme automate etc., dar mai cunoscute sunt în calitate de calculatoare personale de diverse modele și forme: de birou, laptop, notebook, palmtop, pen-based.	Inițial proiectate ca sisteme specializate pentru gestionarea proceselor industriale, a telecomunicațiilor. Mai târziu au devenit sisteme universale utilizate în producția și proiectarea asistată de calculator. Modele noi sunt introduse drept servere ale rețelelor de calculatoare.	Grație faptului că oferă stabilitate, toleranță la erori, flexibilitate, securitate sporită, permit prelucrarea sincronă a milioane de tranzacții și date, fiind utilizate în cadrul companiilor aeriene, băncilor internaționale, companiilor petroliere etc.	Sunt utilizate în aplicații care solicită un volum foarte mare de calcule matematice, precum: previziuni meteorologice, cercetări în domeniul energiei nucleare, explorări petroliere, grafică și animație.
EXEMPLE DE MODELE	Laptop Toshiba	Sistemul de calcul PDP 7	Sistemul de calcul IBM 360	Sistemul de calcul Cray Q2

Caracteristica generală a unui calculator include următoarele date:
- viteza de operare;
- capacitatea memoriei interne;
- componența, capacitatea și timpul de acces ale unităților de memorie externă;
- componența și parametrii tehnici respectivi ai echipamentelor periferice;
- parametrii de bază și gabarit;
- costul.

În funcție de aceste date, calculatoarele moderne se clasifică în 4 categorii:
supercalculatoare;
calculatoare mari (macrocalculatoare);
minicalculatoare;

Supercalculatoarele pot executa peste 10 bilioane de operații pe secundă, iar prețul lor depășește 20 de milioane de dolari. Cercetări și proiectări în industria supercalculatoarelor se realizează în SUA și Japonia de firmele *Gray Reseach, Fujitsu EAT Systems, Sutherland* etc,
Supercalculatoarele se utilizează în prelucrări extrem de complexe ale datelor în aeronautică, fizica nucleară, astronautică, seismologie, prognoza meteo etc

Calculatoarele mari pot executa 1 bilion de operații pe secundă, prețul variind între 20 de mii și câteva milioane de dolari. Calculatoarele mari includ zeci de unități de disc magnetic și imprimante, sute de console aflate la diferite distanțe de unitatea centrală. Aceste calculatoare se utilizează în cadrul unor mari centre de calcul și funcționează în regim non-stop. Pricipalele firme producătoare de calculatoare mari sînt *IBM, UNYSIS, HONEYWELL* etc.

Minicalcultoarele pot efctua sute de milioane de operaţii pe secundă, iar preţul lor nu depăşeşte 200-300 de mii de dolari. Echipamentele periferice ale unui minicalculator includ cîteva discuri magnetice, una sau două imprimante, mai multe console. Minicalculatoarele sînt mai uşor de utilizat şi operat decît calculatoarele mari şi se utilizează în proiectarea asisată de calculator, în automatizări industriale, pentru prelucrarea datelor în experimentele ştiinţifice etc. Dintre firmele producătoare de minicalculatoare vom remarca *IBM, Wang, Texas Instruments, Data General, DEC, Hewlett-Packard* etc.

Microcalculatoarele, denumite şi calculatore personale, sînt realizate la preţuri scăzute - între 100 şi 15000 de dolari şi asigură o viteză de calcul de ordinul milioanelor de operaţii pe secundă. Echipamentele periferice ale unui microcalculator includ o unitate de disc rigid, una sau două unităţi de disc flexibil, o imprimantă şi o consolă. Structura modulară şi gruparea tuturor echipamentelor în jurul unei magistrale permite configurarea microcalculatorului în funcţie de necesităţile individuale ale fiecărui utilizator. Corporaţii care produc microcalculatoare există în foarte multe ţări, însă lideri mondiali, unanim recunoscuţi, sînt firmele *IBM, DEC, Hewlett-Packard, Apple, Olivetti* etc.

TRENDUL ÎN ARHITECTURA CALCULATOARELOR

Din punct de vedere tehnologic cele mai importante tendinţe sunt:

-gradul de integrare al tranzistorilor pe cip creşte cu cca. 55% pe an; tehnologia de integrare a microprocesoarelor a evoluat de la 10 microni (1971) la 0,18 microni 2001.

-frecvenţa ceasului creşte şi ea cu 50% pe an;

-pentru memoriile DRAM, densitatea de integrare creşte cu cca 40-50% pe an, iar timpul de acces aferent scade cu 3 % pe an.

-tehnologia şi performanţele reţelelor se îmbunătăţesc semnificativ.

Se poate spune că aceste tendinţe respectă legea lui Gordon Moore, cofondator împreună cu Obert Noyce a societăţii INTEL. Acesta, în 1965, enunţă celebra sa lege: *„numărul de tranzistori din circuitele integrate se va dubla la fiecare doi ani”* . Aceasta înseamnă că, la fiecare 10 ani se schimbă prefixul de măsurare, adică totul creşte de 1000 de ori. Într-adevăr, dacă hard discurile din anii 90 aveau 100 MB, în 2000 ele au 100 GB. Frecvenţa ceasului era în 1990 de 8 MHz iar în 2000 era de 1 GHz etc.

Între cele mai evidente tendinţe de evoluţie în arhitectură amintim:

-exploatarea paralelismului la nivelul instrucţiunilor şi firelor de execuţie, atât prin tehnici statice (soft) cât şi dinamice(hard); există şi tehnici hibride cum ar fi cazul procesorului Intel Ithamium IA-64;

-structuri tot mai performante de ierarhizare a sistemului de memorie prin utilizarea arhitecturilor evoluate de memorie cache;

-reducerea latenţei ……critice de program, prin tehnici de predicţie;

-utilizarea microprocesoarelor Shered memory în special în cadrul arhitecturii serverelor şi staţiilor grafice.