

1.4. ARHITECTURA STRATIFICATĂ A SISTEMULUI DE CALCUL

Calculatorul numeric este o masina care poate rezolva probleme prin executia unor instructiuni. Solutia problemei, descrisa prin secventa de instructiuni, se numeste **program**. Instructiunile sunt scrise dupa un anumit limbaj. Exista mai multe tipuri de limbaje care pot fi folosite pentru programarea calculatorului si deasemenea mai multe niveluri la care se poate privi sistemul de calcul.

In cartea sa "Organizarea structurata a calculatoarelor", Andrew S. Tanenbaum descrie sistemul de calcul ca o ierarhie de niveluri abstracte, fiecare abstractizare bazandu-se pe cea inferioara in realizarea functiilor sale fara a cunoaste insa in detaliu continutul acesteia. Calculatorul privit la nivelul respectiv este considerat o **masina virtuala**, M_i , iar limbajul de programare pentru respectivul nivel **limbaj masina**, L_i . La nivelul cel mai inferior se afla calculatorul real care executa propriu-zis instructiunile sale masina, L_0 . Nivelul superior comunica instructiunile sale nivelului inferior, care trebuie să le execute (evident prin instructiunile de nivel inferior). Instructiunile de nivel superior sunt ori **translate** in instructiuni de nivel inferior (o instructiune L_1 in mai multe instructiuni L_0) formindu-se astfel un program in L_0 si executat apoi de M_0 , ori **interpretate** prin translatarea si executia pe rind a instructiunilor L_1 .

NIVELUL FRAMEWORK-ULUI
NIVELUL LIMBAJULUI DE NIVEL INALT
NIVELUL LIMBAJULUI DE ASAMBLARE
NIVELUL SISTEMULUI DE OPERARE
NIVELUL ARHITECTURII SETULUI DE INSTRUCTIUNI
NIVELUL MICROARHITECTURII
NIVELUL LOGIC DIGITAL

Calculatoarele moderne sunt formate din doua sau mai multe niveluri. Exista masini cu sase niveluri. In fig. Nivelul 0 corespunde structurii hardware a masinii, nivelul 1 corespunde microarhitecturii, nivelul 2 arhitecturii setului de instructiuni, nivelul 3 sistemului de operare, nivelul 4 limbajului de asamblare, nivelul 5 limbajului orientat pe problema.

Nivelul 0 numit si **nivel logic digital** cuprinde si **nivelul echipamentelor** (care se afla la cel mai scazut nivel si in care utilizatorul poate vedea tranzistoarele). La nivelul digital se afla obiectele numite **porti(gates)** care sunt construite din tranzistoare, dar au intrari si iesiri digitale (semnale ce reprezinta "0" sau "1"). Portile pot fi combinate pentru a forma o memorie de un bit care poate stoca "0" sau "1", iar memoriile de 1 bit pot fi combinate pentru a forma grupuri de 16, 32 sau 64 biti, numite **registre**.

La **nivelul microarhitecturii** se afla registre, o memorie locala si un circuit numit **UAL (Arithmetic and Logic Unit)** care poate executa operatii simple aritmetice si logice. UAL este o unitate combinationala cu doua intrari si o iesire. Aici se poate distinge o cale de date, prin care circula datele, una de adrese si una de control si stare. La unele calculatoare operatiile caii de date sunt controlate prin *microprogram*, iar la altele direct prin hardware. Microprogramul este un interpretor al instructiunilor nivelului superior.

Nivelul setului de instructiuni, numit si **ISA (Instruction Set Architecture)** reprezinta primul nivel programabil de catre utilizator. Aici gasim instructiunile masina furnizate de producator, care la rindul lor sunt interpretate sau executate prin hardware-ul de la nivelul inferior.

Nivelul urmator, al **sistemului de operare** este hibrid, adica se intilnesc si instructiuni ISA si instructiuni noi ale sistemului de operare care sunt interpretate de sistemul de operare sau direct de microprogram.

Primele trei nivele nu sunt utilizate de programatorul obisnuit, ci de **programatorii de sistem** care realizeaza sau intretin interpretoarele sau translatoarele pentru masina virtuala. De la nivelul 4 in sus masina virtuala este folosita de programatorii de aplicatie. Nivelurile 2 si 3 sunt de obicei interpretate, iar de la 4 in sus translatate, desi exista si exceptii. Alta deosebire este ca limbajele primelor trei niveluri sunt numerice (i.e. sunt alcatuite din siruri de numere). La nivelul 4 limbajul de programare se numeste **de asamblare** si reprezinta o scriere simbolica pentru nivelul inferior. Programul care interpreteaza limbajul de asamblare se numeste **asamblor**.

Limbajele de la nivelul 5 se numesc **limbaje de nivel inalt (HLL = High Level Language)** si sunt realizate pentru programatorul de aplicatii. Aici gasim: C, C++, Pascal, Prolog, Java, LISP, etc. Aceste limbaje sunt traduse prin translatoare numite **compilatoare** (exista si exceptii: Java este interpretat). Tot la acest nivel se afla si interpretoarele pentru aplicatii particulare anumitor domenii (matematica, chimie, etc.).

Nivelul **Framework-ului** reprezintă cadrul de dezvoltare aplicații, care oferă instrumente pentru dezvoltarea aplicațiilor complexe care necesită mai multe tehnologii și mai multe limbaje de programare. De exemplu aplicațiile internet, pentru a căror dezvoltare sunt necesare mai multe limbaje de programare și tehnologii (un framework este .NET de la Microsoft).

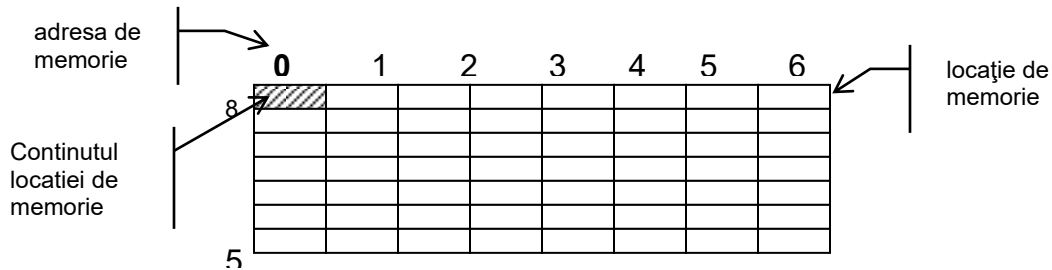
2. MEMORIA SISTEMULUI DE CALCUL

2.1.Organizarea memoriei

Memoria stochează informația în format binar.

Ea este compusă din *locații de memorie* care păstrează datele. Fiecare locație este referită printr-o *adresă*. Se poate imagina memoria ca un fișet cu multe sertare. Fiecare sertar reprezentând o locație de memorie cu o adresă unică și care conține diferite obiecte. Conținutul sertarului se poate modifica prin obiectele care le conține. La fel și conținutul locației de memorie.

Dimensiunea locației de memorie este de un octet.



2.2. Unități de măsură pentru memorie

Deoarece orice sistem de calcul lucrează cu informația în sistem binar, adresele și datele stocate sunt reprezentate prin simple numere binare.

Unități de măsură a informației sunt:

- bitul = o cifră binară (0, 1)
- octetul (byte) = 8 biți
- cuvântul (word) = 16 biți
- dublucuvântul (doubleword) = 32 biți
- dubludublucuvântul (doubledoubleword) = 64 biți

Multiplii unităților de măsură sunt:

$$1 \text{ K (Kilo)} = 10^3$$

$$1 \text{ M (Mega)} = 10^6$$

$$1 \text{ G (Giga)} = 10^9$$

$$1 \text{ T (Tera)} = 10^{12}$$

$$1 \text{ P (Penta)} = 10^{15}$$

$$1 \text{ E (Exa)} = 10^{18}$$

$$1 \text{ Z (Zetta)} = 10^{21}$$

$$1 \text{ Y (Yotta)} = 10^{24}$$

Multiplii pentru măsurarea memoriei sunt:

$$1 \text{ Ki} = 2^{10}$$

$$1 \text{ Mi} = 2^{20}$$

$$1 \text{ Gi} = 2^{30}$$

$$1 \text{ Ti} = 2^{40}$$

$$1 \text{ Pi} = 2^{50}$$

$$1 \text{ Ei} = 2^{60}$$

$$1 \text{ Zi} = 2^{70}$$

$$1 \text{ Yi} = 2^{80}$$

2.3. Caracteristicile memoriei:

1. *capacitatea memoriei* = numărul de locații de memorie
2. *timpul de acces* = timp de obținere a valorii stocate în locația de memorie de la accesarea ei.
3. *rata de transfer* = număr de locații transferate în unitatea de timp
4. *operațiile efectuate asupra ei*:
 - citire = redarea datelor stocate în memorie

- scriere = stocarea datelor în memorie.

2.4. Tipuri de memorie

Din punct de vedere al localizării, există două tipuri de memorie în sistem:

- memoria internă – accesată direct de microprocesor
- memoria externă – stochează permanent datele care sunt transferate în memoria RAM pentru procesare

2.5. Memoria internă

Memoria internă este compusă din:

- memoria *RAM (Random Access Memory)* – cu acces aleator la date, care realizează ambele tipuri de operații (citire, scriere)
- memoria *ROM (Read Only Memory)* – realizează numai operația de citire (conținutul nu poate fi alterat).
- Memorie CMOS – alimentată permanent de la o baterie și care stochează informații de configurare (setup) ale sistemului de calcul.

Ierarhia memoriei

.....

2.5.1. Memoria RAM

Există două tipuri de bază (în funcție de implementarea ei):

- memorie *statică (SRAM)*

Memorie realizată din CBB (circuite basculante bistabile) care pentru anumite semnale date la intrare și în funcția de starea anterioară a circuitului, furnizează o ieșire de 0 sau 1.

Are timp mic de acces, deci este foarte rapidă.

Păstrează informația cât timp este alimentată cu tensiune.

- memorie *dinamică (DRAM)*

Realizată din condensatori, încărcăți cu o sarcină electrică sau descărcați, corespunzător lui 1 respectiv 0 logic.

Deoarece condensatorii se descarcă în timp trebuie reîncărcați periodic. Operația de reîncărcare se numește *refresh de memorie*.

Timpul de acces este mai mare decât la SRAM.

Pentru memoria RAM mai există următoarele implementări:

- EDO RAM – Extended Data Out RAM – care realizează simultan citirea și scrierea memoriei
- ERAM – Enhanced RAM – memorii DRAM care implementează și paginarea memoriei = FPM (?) Fast Page Mode – memorii cu paginare rapidă.
- S-RAM – Synchronous RAM - sincronizate separat pe baza unui semnal de tact și cu un cache integrat
- CDRAM – Cached RAM – idem S-RAM, dar cu cache-ul mai mare
- R-RAM – RAMBus RAM – memorie cu magistrale interne rapide.

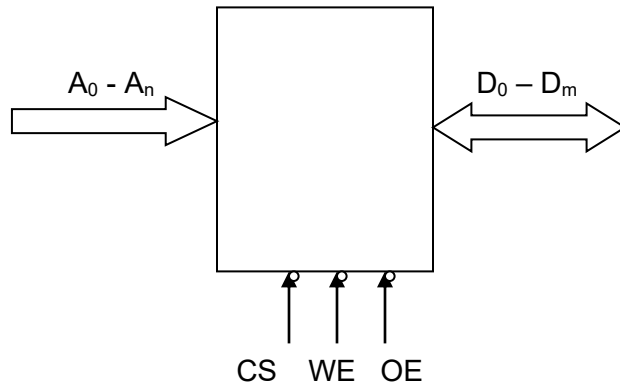
Cipuri de memorie

Pentru a obține o dimensiune de memorie se pot folosi diferite cipuri de memorie (de 1, 4, 8, 16, 32 biți). Cipurile grupate formează plăcuțe de tip:

- SIPP (Single Inline Pin Package) – cu conector în formă de pini
- SIMM (Single Inline Memory Modules) – cu conector în formă de placă
- DIMM (Dual Inline Memory Modules)

Acestea se introduc în soclul de memorie aflat pe placa de bază.

Un cip de memorie este un circuit integrat cu pini de intrare, ieșire și intrare/ieșire.



Unde:

- $A_0 - A_n$ reprezintă biții de adrese
- $D_0 - D_m$ reprezintă biții de date
- CS este pinul de selecție a circuitului (ChipSelect)
- WE este pinul de activare a scrierii (WriteEnable)
- OE este pinul de activare a ieșirii (OutputEnable)

2.5.2. Memoria cache

Memoria de tip RAM este mai lentă decât viteza de lucru a microprocesorului. În consecință datele solicitate de microprocesor vin mai lent, iar microprocesorul trebuie să introducă stări de așteptare (wait states) pentru sincronizarea operațiilor. Memoria cache este introdusă ca o memorie tampon, mai rapidă și mai aproape de microprocesor (timpul necesar parcurgerii traseului până la memorie fiind mult mai scurt).

Memoria cache este divizată fizic în două nivele:

- *memoria cache primară* sau *interna* sau memoria cache pe circuit, plasată în chiar chipul microprocesorului, dar de dimensiuni mai mici
- *memoria cache secundară* sau *externă* sau memoria cache pe placă, plasată în exteriorul microprocesorului și de dimensiuni mai mari

Activitatea memoriei cache este controlată de un *controller de memorie cache* care gestionează adresele și operațiile de citire/scriere efectuate de memorie.

La nivelul memoriei cache interne se realizează o distincție între tipurile de date stocate în memorie. În consecință implementările sunt diferite pentru memoria cache pentru cod, memoria cache pentru date și memoria cache pentru gestiunea memoriei. Memoria cache pentru cod are o implementare de conducă deoarece programele se execută secvențial. În conducă este încărcată o secvență de instrucțiuni care este cel mai probabil să se execute. Conducă se golește la apariția instrucțiunilor de salt, dar se reumple cu instrucțiunile de la adresa de salt. Celelalte memorii cache au implementări asociative deoarece informația nu este atât de predictibilă ca a codului. Microprocesoarele mai noi dispun de două nivele de cache intern.

Memoria cache externa se afla pe placa de baza a sistemului si este formata din cipuri de memorie foarte rapide (de tip SDRAM) cu timpi de acces foarte mici. La acest nivel nu se realizeaza o distinctie intre tipurile de date manipulate, iar implementarea pentru aceasta memorie este asociativa.

Modul de lucru pentru memoria cache este urmatorul: microprocesorul cauta datele necesare in cache-ul intern, daca nu le gaseste (cache miss) le cauta in cache-ul extern, iar daca nu sunt nici in acesta le cauta in memoria RAM.

2.6. Memoria externă

Memoria externă reprezintă suportul pentru stocarea permanentă a datelor.

Caracteristici:

- caracteristicile generale ale memoriei
- *densitatea de înregistrare a datelor* = număr de biți/unitate de stocare.

Medii de stocare a datelor

1. Magnetice:

- Harddiscul (HD)
- Floppy discul (FD)
- Harddiscul portabil (HPD)
- Banda magnetică (streamer)

2. Optice:

- Discul CD-ROM

3. Magneto-optice:

- Discul magneto-optic

4. Memoria Flash

Banda magnetică

- informația stocată în format binar similar procesului Hi/Fi
- există două tipuri de benzi magnetice:
 - o cu înregistrare analogică (capacități de 50MB – 1GB)
 - o cu înregistrare digitală (DAT – Digital Audio Tape) – capacități de până la 5GB.și densitate mare de înregistrare a datelor și mult mai performante beneficiând de avantajele prelucrărilor digitate. Sunt stocate și informații de control pentru securizarea datelor.
- accesul la informație se face *secvențial*
- informația este organizată în cadre, înregistrări, piste.
- preț mic => capacitate mare => ideală pentru stocări masive de date
- timp mare de acces

Mai utilizată este banda *Streamer*.

Discul flexibil (Floppy disk)

- Construcție: - disc flexibil din plastic pe care este depus un strat subțire de material magnetic (oxid de crom). Dispune de capete de citire/scriere pentru fiecare față a discului.
- înregistrarea informației se face prin magnetizare/demagnetizare.
- Capacitate mică de stocare a datelor
- Timp mare de acces.
- Rată mică de transfer.
- Unitatea de disc cuprinde un controller de disc care dă funcționalitate prin implementarea interfeței cu unitățile componente și a operațiilor asupra discului.

- Organizarea informației pe disc în sectoare, piste (numerotate începând cu 0 din centrul discului).
- Pentru păstrarea informației pe disc trebuie realizată operația de formatare.
- Acces direct la blocurile de memorie
- Acces simultan la mai multe fișiere

Harddiscul

- Numit și disc *dur,amovibil, fix* sau *Winchester*, este cel mai utilizat mediu de stocare în sistemele actuale.
- Construcție: mai multe discuri magnetice suprapuse, prinse pe același ax, înregistrate pe ambele fețe, capete de citire/scriere pentru fiecare față.
În funcție de mobilitatea capetelor există: HD cu *capete fixe* și HD cu *capete mobile*.
- Capacitate mare de stocare a datelor
- Timp de acces mai bun decât la floppy
- Rată mare de transfer a datelor
- Densitate mare înregistrare
- Organizarea informației pe disc:
 - o *piste* – zone circulare concentrice ale unui disc. Numerotarea pistelor prin *interșesere (interleaving)*
 - o *Sectoare* – reprezintă unități de transfer a informației pe disc și de diviziune a pistelor (o pistă formată din mai multe sectoare). Un sector este format dintr-un număr de octeți (putere a lui 2). La DOS, 512 octeți.
 - o *Clustere* – organizare logică a mai multor sectoare și formând unități de alocare.
 - o *Cilindri* – reprezintă piste active. Un cilindru este format din piste cu aceeași rază.
- Acces direct la blocurile de memorie: pentru a ajunge la o înregistrare trebuie indicat cilindrul, pista, sectorul și adresa de început în sector.
- Acces simultan la mai multe fișiere
- Discul trebuie pregătit pentru păstrarea informației prin formatare. Aceasta are două etape:
 - *Formatarea primară* (de nivel scăzut-low level) – stabilirea structurilor de piste circulare, concentrice și a sectoarelor pe disc.
 - *Formatarea logică* – ce cuprinde partiționarea discului corespunzător căreia discul este împărțit în mai multe unități logice numite *partiții*.

Discul optic

- Densitate foarte mare de înregistrare pe suprafață mică
- Construcție: material policarbonat, foarte dur și rezistent pe care se depune un strat fin de aluminiu reflectorizant.
- Tehnologie laser pentru înregistrare/redare. Scrierea se face printr-o rază foarte dură care arde adâncituri (pits) pe spirală. Zonele nealterate ale spiralei sunt numite lands. Un 1 logic este reprezentat printr-o succesiune pit/land, iar 0 logic prin succesiuni pit-pit sau land-land.
- Două tipuri în funcție de caracterul permanent al datelor stocate:
 - o Discuri reversibile R/W
 - o Discuri ireversibile RO
- Organizarea informației pe disc – o pistă în spirală cu parcurgere din interior la exterior. Pista este împărțită în sectoare cu dimensiunea de 2048 octeți.
- Capacitate mare de stocare
- Timp bun de acces
- Densitate mare de înregistrare.

- Exemplu: discurile CD-ROM

Discul magneto-optic

- operațiile de citire/scriere se fac prin procedee magnetice și optice.

Memoria Flash

Memorie asemanatoare celei RAM cu consum redus de energie si care se cupleaza foarte rapid. Are forma unei cartele de credit (extraplata). Este accesata printr-o interata speciala.

Iscrierea se face ca la EPROM-uri.

Timp de acces foarte bun.

Alimentate de la o baterie.

Capacitate mica de stocare.

Pret foarte mare. Utilizate la laptopuri.

Codificarea informației

Informația, pentru a fi procesată, trebuie transformată în sistem binar. Acest lucru se realizează prin *codificare*.

1. Reprezentarea numerelor:

C_{n-1}	C_{n-2}	C_1	C_0
-----------	-----------	-------	-------	-------

- **Numerele întregi** sunt reprezentate binar în mărime și semn.

- Reprezentarea în **marime**

$$N = \sum_{i=0}^{n-1} c_i 2^i$$

- Reprezentarea în **marime și semn**

$$N = (-1)^s \sum_{i=0}^{n-1} c_i 2^i$$

- Reprezentarea în **complement fata de 2**

Numerele negative sunt reprezentate în complement față de doi

$$N = (-1)^s \left(\sum_{i=0}^{n-1} \bar{c}_i 2^i + 1 \right)$$

- **Numerele cu zecimale|reale** sunt reprezentate în **virgulă mobilă (Floating Point)**

S	M	E
---	---	---

Unde **S=bit de semn**, **M=mantisa**, **E=exponentul** fiecare pe un număr de biți. IEEE a elaborat **standardul IEEE754** pentru numere în virgulă mobilă cu 4 formate: 2 pe 32 biți simple și 2 extinse pe 64 biți.

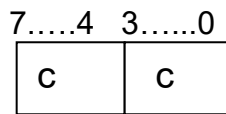
- **Reprezentarea numerelor zecimale**

- **BCD (Binary Coded Decimal)** – cod pe 4/8 biți pentru reprezentarea cifrelor zecimale

- **BCD neimpachetat** – o cifră se codifică pe un octet în digitul mai puțin semnificativ

7.....4	3.....0
	C

- **BCD impachetat** – două cifre se codifică pe un octet

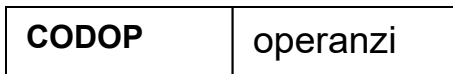


2. Reprezentarea caracterelor

Există mai multe standarde de codificare a caracterelor, dintre care cele mai utilizate sunt:

- **ASCII** – pentru caractere – cod pe 7/8 biți pentru reprezentarea caracterelor
- **UNICODE** – cod pe 16 biți (2 octeți pentru reprezentarea caracterelor)

3. Reprezentarea instructiunilor

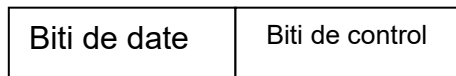


● Ordinea de stocare a datelor în memorie

Pentru date mai mari de un octet există două posibilități de ordonare:

- ***big endian*** – în care octetul mai semnificativ este memorat întâi (la adresă mai mică)
- ***little endian*** – în care octetul mai semnificativ este memorat la adresă mai mare.

4. Coduri detectoare/corectoare de erori



Utilizeaza biti **suplimentari (biti de control)**, atasati bitilor de date, cu care se poate verifica corectitudinea datelor stocate sau transmise.