

CUPRINS

MODULUL 1: ARHITECTURA SISTEMULUI DE CALCUL	1
CAPITOLUL 1. Concepte fundamentale referitoare la calculatoare.....	1
1.1. Arhitectura sistemului de calcul	1
1.2. Descrierea sistemului de calcul	2
1.3. Elementele componente ale sistemului de calcul	3
1.4. Legi empirice.....	5
1.5. Tipuri de calculatoare	5
1.6. Reprezentarea informației	8
LISTA DE FIGURI	14
BIBLIOGRAFIE	15

MODULUL 1: ARHITECTURA SISTEMULUI DE CALCUL

CAPITOLUL 1. Concepte fundamentale referitoare la calculatoare

1.1. Arhitectura sistemului de calcul

Datorită accesibilității calculatoarelor pe de o parte, din punct de vedere al utilizatorului, al instrumentelor pe care acestea le oferă și dinamicii care există în acest domeniu (calculatoare de diverse tipuri: calculatoare personale, dispozitive mobile, echipamente cu sisteme de calcul înglobate și interconectate prin internet, sisteme Cloud, etc.), pe de altă parte, este esențială cunoașterea conceptelor fundamentale referitoare la sistemele de calcul și dobândirea de cunoștințe generale privind serviciile oferite de sistemele de calcul. Abordarea potrivită, în acest context, este cea arhitecturală. În cele ce urmează este prezentat acest concept. Calculatorul este un system de calcul

Noțiunea de sistem

Sistem (definiție) [1] = Un ansamblu de elemente inter-relaționate ce compun un întreg.

Termenul de „system” în latină și greacă înseamnă „a pune împreună, a combina”. Un subsistem este o parte a unui sistem. În mod tipic un sistem este alcătuit din componente (elemente) care sunt interconectate și interacționează pentru a facilita fluxul de informație.

Arhitectura sistemului este dispunerea și interconectarea componentelor pentru a obține funcționalitatea dorită a sistemului. [2]

Definiție. Un sistem cu intrări (I), ieșiri (E) și transformări ($F(x)$) cunoscute, dar cu conținut necunoscut se numește *black-box*.

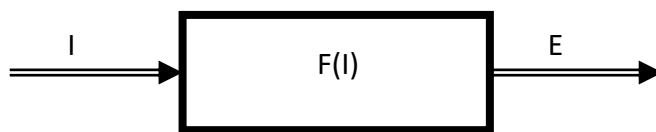


Figura 1 Black Box

Proprietatea cea mai importantă a cutiei negre este utilizabilitatea. Adică utilizarea fără a cunoaște detalii de implementare (constructive).

Pentru a facilita construcția sistemelor din module cu funcționalitate cunoscută (black box) acestea au fost standardizate.

Un *standard* cuprinde o descriere a modului de utilizare a unui modul (specificații de utilizare).

Organizațiile internaționale de standarde: ISO (International Standard Organization), IEEE (Institute of Electrical and Electronics Engineers), IETF (Internet Engineering Task Force), ș.a. au elaborat o serie de standarde, respectate de producători, în realizarea modulelor respective.

Definiție. Un *sistem de calcul* este un sistem care execută programe stocate în memorie în interacțiune cu mediul extern.

Arhitectura sistemelor de calcul sau arhitectura calculatoarelor este teoria din spatele construcției unui calculator. În același mod în care un arhitect stabilește principiile și obiectivele construirii unui proiect ca baze ale unor planuri de construcție, în același mod un arhitect de calculatoare stabilește arhitectura sistemului de calcul ca bază a specificațiilor de proiectare.

Obiectivul principal în arhitectura unui sistem de calcul îl reprezintă un raport cost/performanță cât mai bun. Componenta sistemului este o cutie neagră, adică se cunoaște utilitatea, dar nu modul de implementare.

Arhitectura sistemului este dispunerea și interconectarea componentelor pentru a obține funcționalitatea dorită a sistemului.

1.2. Descrierea sistemului de calcul

Definiție. *Sistemul de calcul* este o mașină virtuală care execută programe, scrise în limbajul mașinii respective, stocate în memorie, în interacțiune cu mediul extern [2].

Un *program* este o soluție algoritmică a unei probleme, scrisă într-un limbaj, numit *limbaj de programare*. Un *algoritm* este o soluție secvențială a unei probleme.

Limbajul mașină este limbajul executat de mașina. Limbajul de programare nu este limbajul mașină. Limbajul de programare este translatat în limbaj mașină pentru execuție.

Program este succesiunea de instructiuni ce implementeaza un algoritm.

Există două tehnici de executie a programului:

- compilare si executie
- interpretare (masina virtuala care interpreteaza si executa programul).

Compilerul este translatează programul din limbaj de programare în program în limbaj mașină (executabil), iar interpretorul translatează o instrucțiune în secvență de instrucțiuni în limbaj mașină pentru execuție (este o mașină virtuală care execută programul).

1.3. Elementele componente ale sistemului de calcul

Din definiția sistemului de calcul, prin identificarea funcțiilor de realizat (acțiuni reprezentate de verbe), se pot distinge următoarele componente funcționale conceptuale: [2]

- de *executare (procesare)*, reprezentată de (*micro*)procesor
- de *memorare*, reprezentată de *memorie*
- de *comunicație* cu mediul extern, reprezentată de *unitățile de intrare/ieșire (dispozitivele periferice)*

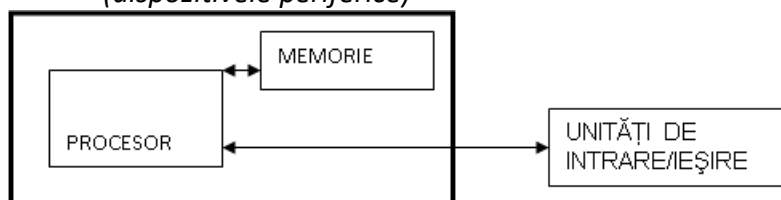


Figura 2 Componentele sistemului de calcul

Din punct de vedere tangibilității, sistemul de calcul este împartit în două mari parti:

- partea *hardware* – reprezentata de circuitele electronice, placi, cabluri, memorii, etc. care reprezinta echipamentul propriu-zis de calcul și care sunt tangibile.
- partea *software* – reprezentata prin programe care implementează algoritmi și reprezintă idei abstracte.

Diferența dintre hardware și software până nu demult a fost evidentă, cu timpul însă ele au devenit logic echivalente. Ambele putând realiza aceleași funcții, iar alegerea implementării făcându-se după criterii preț/performanță.

Pentru a înțelege funcționarea calculatorului vom introduce noțiunea de informație care, furnizată de utilizator sau mediu, este convertită în format binar, intern, prelucrat de sistemul de calcul (date).

Adoptarea reprezentării binare a fost impusă de utilizarea în construcția calculatoarelor a dispozitivelor cu două stări stabile, notate conventional cu 0 și 1. Unitatea de măsură pentru numerele binare este *bit*-ul (Binary digIT).

Vom face în continuare o prezentare succintă a componentelor sistemelor de calcul cu referire directă la calculatoare. Numeroasele componente ale unui sistem de calcul pot fi grupate în unități având funcții mai complexe bine precizate. În figura următoare denumirea

fiecarei unități indică funcția ei, iar săgețile de legatură arată modul de transmitere a informației de la una la alta [2].

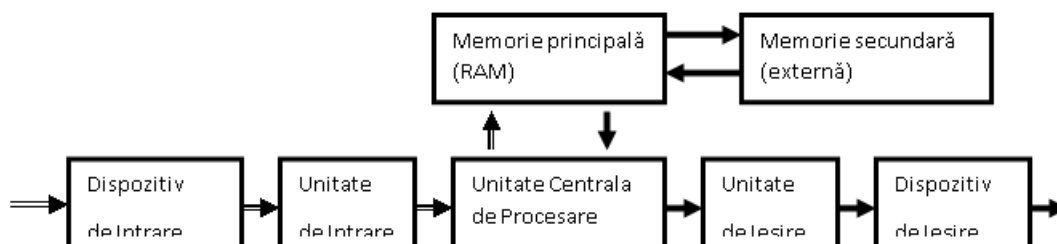


Figura 3 Fluxul informațional

Informația, furnizată de mediul extern (utilizator), este preluată de dispozitivul de intrare, codificată (convertită în format binar) și transmisă unității de intrare care realizează interfața cu unitatea de procesare (unitatea centrală de procesare = UCP) .

Exemple de dispozitive de intrare: tastatura, mouse, scanner, MODEM, etc. Astfel, la tastatura, apăsarea unei taste produce codului binar corespunzător al tastei apăsate. Scannerul preia o imagine și o transformă într-o succesiune de coduri binare. MODEM-ul preia datele transmise de la distanță. Unitatea de intrare realizează interfața cu UCP a.î. dispozitivele de intrare pot fi diferite.

Informația este înregistrată și pastrată în memoria principală. De aici ea poate fi transmisă ulterior altor unități funcționale. Informația este supusă prelucrării în UCP. UCP este formată din unitatea de calcul și unitatea de comandă. Unitatea de calcul efectuează operații simple, aritmetice și logice, asupra unor operanți din memorie, înregistrând rezultatele tot în memorie. Unitatea de comandă are ca rol coordonarea funcționării celorlalte unități, pe baza unor instrucțiuni sau comenzi.

Informația care nu este prelucrată la un moment dat, poate fi pastrată în unități de memorie externă, mai lentă decât memoria principală (operativă), dar cu o capacitate mai mare. Informația poate fi transmisă, dacă este cazul, de la o memorie la alta.

Rezultatele prelucrărilor sunt transmise utilizatorului prin unitatea de ieșire către dispozitivul de ieșire. Dispozitivul de ieșire realizează conversia datelor din format binar în formatul necesar reprezentării informației. Exemple de dispozitive de ieșire : monitor, imprimantă, MODEM, plotter, etc. De exemplu, o imprimantă convertește codurile binare ale caracterelor în formatul tipărit. Similar, un monitor (display) transformă reprezentările binare ale informației în formatul afișat.

Părți hardware principale componente :

- *UCP (Unitatea Centrală de Procesare) sau CPU (Central Processing Unit) în engleză, este înglobată într-un circuit integrat numit microprocesor*
- *Memoria*

- *Subsistemul de I/E* care include *dispozitivele de intrare/ieșire* numite și *dispozitive periferice*

UCP are rolul de a prelucra programul alcătuit din instrucțiuni și de a controla activitatea întregului sistem.

Memoria este cea în care se stochează informația în format binar.

Subsistemul de I/E realizează interfața cu mediul exterior.

Suportul de comunicație reprezintă infrastructura de comunicație necesară transmiterii datelor.

1.4. Legi empirice

Legea lui Moore (legea hardware-ului)

În anul 1965, Gordon Moore, fondator al companiei Intel, observând că numărul de tranzistoare creștea constant, a prezis că numărul acestora se va dubla anual [3]. Aceasta a devenit legea lui Moore, exprimată ca dublarea numărului de tranzistoare la fiecare 18 luni. Evident acest progres tehnologic a dus la creșterea performanțelor sistemelor și la scăderea prețurilor.

Legea software-ului

Enunțată de Nathan Myhrvold [4] spune că “software-ul este ca un gaz, crescându-și volumul astfel încât să ocupe tot spațiul pe care îl are la dispoziție”. Această lege indică faptul că resursele hard disponibile sunt imediat consumate de către soft, chiar mai mult existând o cerere permanentă de resurse hardware.

1.5. Tipuri de calculatoare

Există două direcții importante în dezvoltarea calculatoarelor:

- *CISC (Complex Instruction Set Computers)* corespunzătoare calculatoarelor realizate cu microprocesoare cu arhitectură CISC.
- *RISC (Reduced Instruction Set Computers)* corespunzătoare calculatoarelor realizate cu microprocesoare RISC, reprezentativ fiind microprocesorul SPARC realizat de firma Sun.

În paralel sunt dezvoltate direcții alternative:

- *Calculatoare paralele*. Exemplu: reprezentativ este MIPS (Millions of Instruction Per Second) realizat la Universitatea Stanford USA, cu arhitectură mai specială, paralelă.
- *Calculatoare orientate către limbaj*: direcție nouă de dezvoltare o constituie cipurile JVM (Java Virtual Machine).

Exemple de tipuri de calculatoare sunt:

- *Calculatoare personale* – ele se referă la calculatoarele de birou și la laptopuri. Ele sunt monoprosesor și se numesc PC-ri (dacă microprocesorul este CISC) sau stații de lucru (dacă procesorul este RISC). Puterea lor de calcul crește pe măsura evoluției tehnologice. Pot fi echipate cu MODEM-uri pentru transmisia la distanță.



Figura 4 Calculator Desktop



Figura 5 Calculator Laptop

- *Server-e* – Ele se referă la calculatoarele cu putere mai mare din rețea pe care se află instalat software-ul corespunzător, deservind stațiile de lucru.



Figura 6 Calculator Server

- *Mulțime de stații de lucru* – numite și Networks of Workstations (NOW), sau Clusters of Workstations (COW) – sunt alcătuite din mai multe stații de lucru legate prin rețele de mare viteză și având un software distribuit pentru soluționarea împreună a unor probleme specifice unui domeniu.
- *Sistemele Cloud* - specifice sistemelor foarte mari cu capacitate foarte mare de stocare și de procesare, care partajează resursele de calcul
- *Supercalculatoarele* - cu UCP foarte rapide, resurse mari (memorie) și interconectări rapide folosite pentru calcule foarte complicate științifice
- *Dispozitivele mobile* – calculatoare de dimensiune mică, portabile, dar cu resurse mari de calcul (tablete, telefoane smart)



Figura 7 Tableta

- *IoT(Internet of Things)* – obiecte cu resurse de calcul înglobate care comunică pe Internet



Figura 8 Internet of Things

- console de jocuri, Game Consoles

Toate aceste calculatoare au unitatea centrală de prelucrare (CPU = Central Processing Unit) integrată pe un chip, numit *microprocesor*.

1.6. Reprezentarea informației

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

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 Y(Yotta)= 10^{24}

Multiplii pentru măsurarea memoriei sunt:

1 Ki = 2^{10}

1 Mi = 2^{20}

1 Gi = 2^{30}

1 Ti= 2^{40}

1 Pi= 2^{50}

1 Ei = 2^{60}

1 Zi= 2^{70}

1 Yi= 2^{80}

Conceptul de program

Un program este o soluție algoritmică a unei probleme scrisă într-un limbaj, numit limbaj de programare.

Calculatorul este mașină virtuală care execută programe în limbajul mașinii.

Limbajul de programare oferă instrumente pentru descrierea soluției algoritmice a unei probleme:

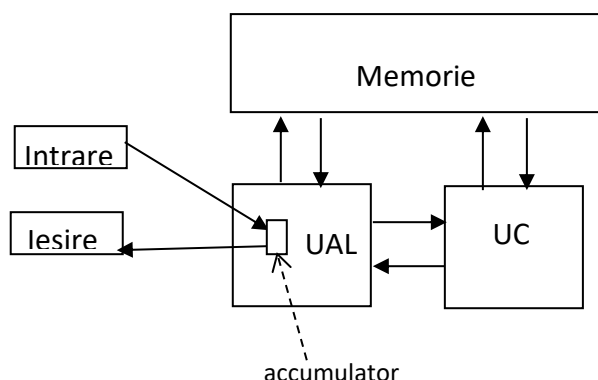
- tipuri de date
- set operatori
- instrucțiuni = operații (de control al fluxului)
- biblioteci de funcții, proceduri, obiecte

Arhitectura VON NEUMANN

Modelul von Neumann a fost specificat de von Neumann în 1946 și cuprinde elementele principale componente ale unui sistem de calcul și interacțiunea între ele.

El are la bază patru componente: *memoria, unitatea aritmetică și logică* cu un registru intern special numit acumulator, *unitatea de comandă* și *echipamentele de intrare și de ieșire*.

Arhitectura masinii von Neumann este cea din figura urmatoare:



În această arhitectură programele sunt memorate și ea stă la baza calculatoarelor moderne cu un procesor.

Unitățile componente se numesc și **unități funcționale** deoarece fiecare are o funcționalitate specifică în sistem.

Unitatea centrală (UC) are rolul de a prelucra programul alcătuit din instrucțiuni și de a controla activitatea întregului sistem. Instrucțiunea este memorată ca secvența de biți în memorie.

UC este alcătuită din:

unitatea de comandă

unitatea aritmetică și logică (UAL) sau unitatea de calcul

regiștrii

Unitatea de comandă decodifică instrucțiunile, le interpretează și comandă operațiile de executat.

Unitatea de calcul execută operațiile aritmetice și logice.

Regiștrii sunt folosiți pentru stocarea temporară a datelor de prelucrat. În modelul von Neumann este specificat doar registrul acumulator. Ulterior setul de regiștri a fost îmbogățit cu mai multe tipuri, în funcție de datele stocate (adrese, date, stare, etc.).

Memoria este cea în care se stochează informația în format binar. De aceea suportul de memorie trebuie să asigure două stări stabile distincte. Ea este compusă dintr-un șir de **locații de memorie**, iar accesul la ele se face prin **adrese**. Locația de memorie are dimensiunea de un **octet**. Un octet fiind format din 8 biți. Un **bit** putând stoca o valoare

binară (0 sau 1). Numărul locațiilor de memorie formează **capacitatea de stocare** a memoriei. În memorie sunt stocate atât date cât și programe.

Operațiile care se execută cu memoria sunt: de citire și de scriere

Într-un sistem de calcul există mai multe tipuri de memorie. Ele sunt dispuse ierarhic în sistem în funcție de timpul de acces și de prețul acestora. Astfel pe nivelul inferior se află **memoriile de masă** reprezentate prin discuri magnetice, CD-uri, având timp mare de acces, preț mic și corespunzător capacitate mare. În memoriile de masă se stochează informația persistentă în timp. Pe urmatorul nivel se află **memoria primară (RAM=Random Acces Memory)**. Această memorie este indispensabilă în sistemul de calcul, fiind cea cu care se execută programele. Timpul de acces este mic, pretul mai mare și capacitatea de stocare mult mai mică decât cea a memoriei de masă. Caracteristica principală memoriei primare este **volatilitatea**, adică pierderea informației la căderea tensiunii. Cu această memorie se efectuează atât operații de scriere cât și operații de citire.

Următorul nivel îl constituie cel al memoriilor de tip **cache**. Acestea sunt memorii foarte rapide, mai scumpe, folosite pentru accelerarea vitezei de lucru a calculatorului.

Pe ultimul nivel se află **regiștrii** care sunt cei mai rapizi aflându-se direct conectați la unitățile de procesare. Capacitatea lor de stocare este foarte mică, iar numărul lor este limitat de dimensiunea microprocesorului.

Alte tipuri de memorii aflate în sistem sunt:

memoria **ROM (Read Only Memory)**, care stochează programele furnizate de fabricantul calculatorului. Singura operație efectuată cu acest tip de memorie este citirea ei.

memoria **CMOS** (numele este dat de tehnologia de realizare a acesteia). În această memorie sunt păstrate informații de configurare a calculatorului. Cu ea se realizează operații de citire și scriere și este alimentată de la o baterie.

Unitatea de intrare este componenta care asigură funcția de preluare a informațiilor de intrare. Acestea sunt **citite** de la un **dispozitiv de intrare**.

Unitatea de ieșire este componenta care asigură funcția de furnizare a informațiilor la ieșire. Acestea sunt **scrise** și transmise unui **dispozitiv de ieșire**.

Pentru sincronizarea unităților funcționale componente, există între ele **interfețe** în care se află **buffer** (**elemente temporare de memorare**).

Informația este transmisă în sistem pe căi electrice numite **magistrale**. În funcție de tipul de informație care circulă prin ele, ele se clasifică în: **magistrala de adrese**, **magistrala de date**,

magistrala de control. Pentru a se obține flexibilitate în interconectarea diverselor componente ale sistemului de calcul, magistralele sunt standardizate.

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 **translatate** 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 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.).

LISTA DE FIGURI

Figura 1 Black Box	2
Figura 2 Componentele sistemului de calcul	3
Figura 3 Fluxul informațional	4
Figura 4 Calculator Desktop	6
Figura 5 Calculator Laptop	6
Figura 6 Calculator Server	7
Figura 7 Tableta.....	7
Figura 8 Internet of Things.....	8

BIBLIOGRAFIE

- [1] dexonline, "www.dexonline.ro".
- [2] O. Chelai, Arhitectura sistemelor de calcul, Constanta: Universitatea Ovidius din Constanta, 2005.
- [3] Moore, "Moore's Law," <https://www.intel.com/content/www/us/en/silicon-innovations/moores-law-technology.html>, 1965.
- [4] N. Myhrvold, "Nathan's Law," <https://www.wired.com/1995/09/myhrvold/>, 1995.
- [5] Logitech, "<https://www.logitech.com/assets/45920/6/hd-pro-webcam-c920-quick-start-guide.pdf>".
- [6] Euramis, "Euramis," office@euramis.ro, 2019.
- [7] M. V. S. Andrew Tanenbaum, Distributed Systems: Principles and Paradigms, Prentice-Hall, Inc., 2007.
- [8] Microsoft, "support.microsoft.com".
- [9] Wikipedia, "Wikipedia.org".
- [10] S.C. SAN Software SRL, Suport curs Inițiere în „Operare, Introducere, Validare și Prelucrare Date”, Contract nr. POSDRU/105/5.1/G/76944, 2013.
- [11] Ministerul Muncii si Protectiei Sociale, "LEGE Nr. 319," *Monitorul Oficial al României*, 14 iulie 2006.
- [12] AEISSM, "Securitatea și sănătatea în muncă. În beneficiul companiilor," <https://osha.europa.eu/ro>, 2019.
- [13] Cannon, "www.cannon.ro," www.cannon.ro, 2019.
- [14] IRIScan, "www.irislink.com".
- [15] Brother_pt-e550wsp, "support.brother.com".
- [16] D. QuickScan, "Datalogic.com".
- [17] W. Intuos, "www.wacom.com".

[18] A. Router, "www.asus.com".