Curs nr.2

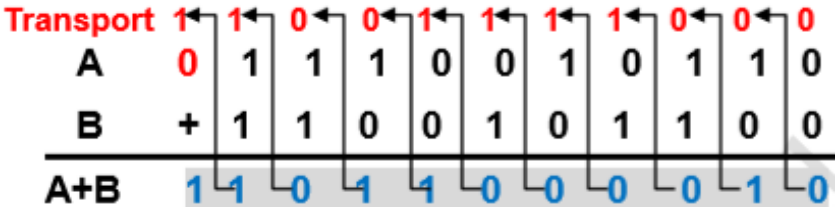
OPERAŢII CU NUMERE BINARE

A. ADUNAREA NUMERELOR BINARE

Reguli de bază:

- 0 + 0 = 0 transport 0;
- 0 + 1 = 1 transport 0;
- 1 + 0 = 1 transport 0;
- 1 + 1 = 0 transport 1.

Pentru a aduna două numere binare se adună între ei biţii numerelor (începând de la dreapta la stânga) iar la acest rezultat se adaugă transportul (care poate fi 0 sau 1) conform regulilor de mai sus.



Algoritmul de realizare a adunării de mai sus:

- Se adună biţii de pe prima coloană din dreapta. Rezultatul se trece sub coloană iar transportul deasupra celei de-a doua coloane din dreapta;
- Se adună biţii de pe a doua coloană din dreapta. Rezultatul se adună cu transportul de deasupra coloanei apoi se trece rezultatul obţinut sub coloană iar transportul se trece deasupra celei de-a treia coloane din dreapta;
- Se continuă adunarea după acest algoritm până se ajunge la prima coloana din stânga.

  0 + 0 = 0 transport 0;
  1 + 0 + 0 = 1 transport 0
  1 + 1 + 0 = 0 transport 1
  0 + 1 + 1 = 0 transport 1
  1 + 0 + 1 = 0 transport 1
  0 + 1 + 1 = 0 transport 1
  0 + 0 + 1 = 1 transport 0
  1 + 0 + 0 = 1 transport 0
  1 + 1 + 0 = 0 transport 1
  1 + 1 + 1 = 1 transport 1
  0 + 1 = 1

Rezultă numărul binar 11011000010

Exemple de adunări cu numere binare:

Transport	0	0	0	0	0	0	0	0
A	0	1	0	0	1	1	0	0
B	1	0	0	1	0	0	0	1
A+B	1	1	0	1	1	1	0	0

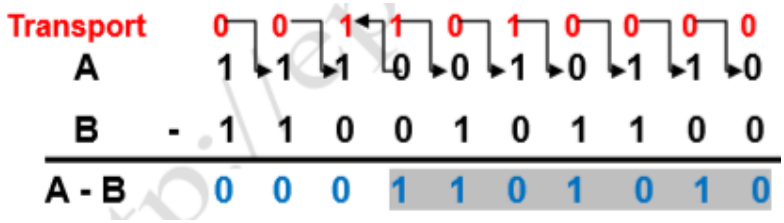
Transport	1	1	1	1	1	1	1	0
A	0	1	1	1	1	1	1	1
B	0	0	1	1	1	1	1	1
A+B	1	0	1	1	1	1	1	0

B. SCĂDEREA NUMERELOR BINARE

Reguli de bază:

- 0 + 0 = 0 transport 0;
- 1 - 0 = 1 împrumut 0;
- 1 - 1 = 0 împrumut 0;
- 0 - 1 = 1 împrumut 1.

Exemplu:



Pentru a scadea două numere binare se scad între ei biţii numerelor (începând de la dreapta la stânga) iar din acest rezultat se scade împrumutul (care poate fi 0 sau 1) conform regulilor de mai sus.

- Se scad din biţii numărului A biţii numărului B de pe prima coloană din dreapta. Rezultatul se trece sub coloană iar împrumutul deasupra celei de-a doua coloane din dreapta.
- Se scad biţii de pe a doua coloană din dreapta. Din rezultat se scade

împrumutul de deasupra coloanei apoi se trece rezultatul obţinut sub coloană iar împrumutul se trece deasupra celei de-a treia coloane din dreapta.

- Se continuă scăderea după acest algoritm până se ajunge la prima coloana din stânga.

Exemple de scăderi cu numere binare:

Imprumut	0	1	1	0	0	1	1	0
A	1	1	0	0	1	1	0	0
B	1	0	0	1	0	0	0	1
A-B	0	0	1	1	1	0	1	1

Imprumut	0	0	0	0	0	1	0	0
A	1	0	1	1	1	1	0	1
B	1	0	0	0	0	0	1	1
A-B	0	0	1	1	1	0	1	0

Imprumut	1	0	1	0	1	0	1	0
A	1	0	1	0	1	0	1	0
B	0	1	0	1	0	1	0	1
A-B	0	1	0	1	0	1	0	1

Imprumut	1	1	0	0	1	1	0	0
A	1	0	0	1	1	0	0	1
B	0	1	1	0	0	1	1	0
A-B	0	0	1	1	0	0	1	1

C. ÎNMULȚIREA NUMERELOR BINARE

Reguli de bază:

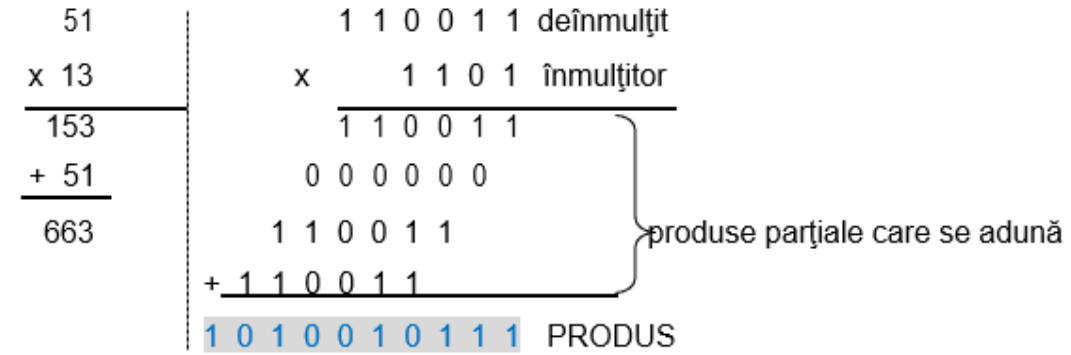
- 0 x 0 = 0;
- 1 x 0 = 0;
- 0 x 1 = 0;
- 1 x 1 = 1.

Pentru a înmulţii două numere binare A (deînmulţit) şi B(înmulţitor) se procedează exact ca la înmulţirea a două numere zecimale:

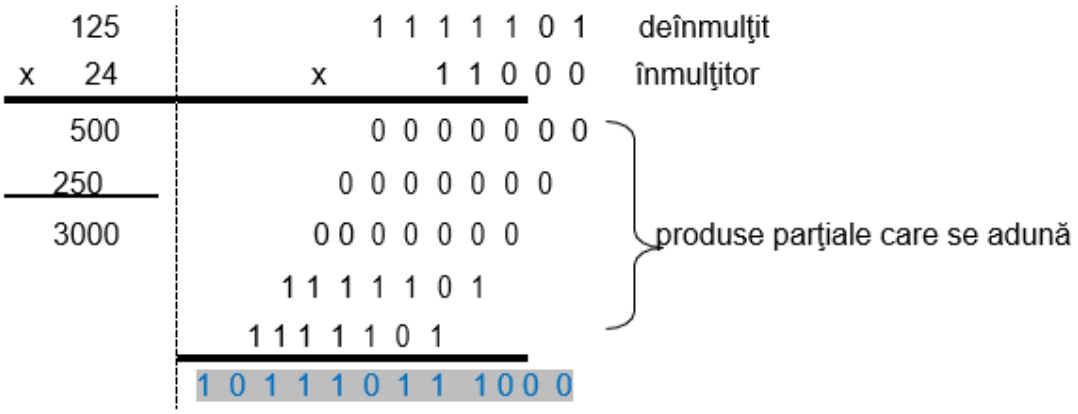
- Se înmulţeşte pe rând fiecare cifră a înmulţitorului cu cifrele deînmulţitului;
- Se scriu rezultatele obţinute unul sub altul decalându-le cu o unitate spre stânga;
- Se adună pe verticală cifrele rezultatelor fiecărei înmulţiri respectând regulile de adunare a numerelor binare.

Exemple de înmulţiri a numerelor binare:

Exemplul 1.



Exemplul 2.



D. ÎMPĂRŢIREA NUMERELOR BINARE

Algoritmul de împărţire a două numere binare are la bază metoda împărţirii a două numere întregi. Fiind dat deîmpărţitul D şi împărţitorul Î, pentru operaţia de împărţire trebuie să se determine câtul C şi restul R, astfel încât să fie satisfăcută relaţie:

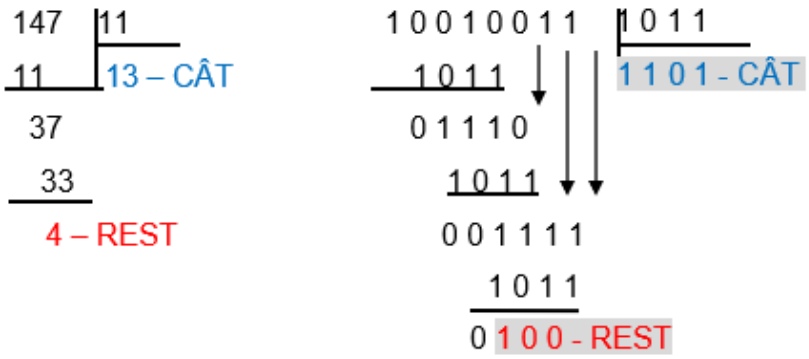
D = Î x C + R

Operaţia de împărţire în cazul numerelor binare, se va reduce la o serie de scăderi ale împărţitorului din restul parţial ţinând cont de următoarele reguli:

- Dacă restul este mai mare decât împărţitorul câtul este 1;
- Dacă restul este mai mic decât împărţitorul câtul este 0.

La efectuarea scăderilor se respectă regulile de scăderea a numerelor binare. Exemple de împărţire a numerelor binare

Exemplul 1.



Algoritmul împărţirii deîmpărţitului 10010011 la împărţitorul 1011:

Deoarece împărţitorul 1011 este mai mare decât primii 4 biţi ai deîmpărţitului 1001 împărţitorul se va împărţii la primi 5 biţi ai deîmpărţitului 10010;

Deoarece 10010 este mai mare decât 1011.

Deci *primul bit al câtului este 1;*

- Înmulţim câtul 1 cu împărţitorul 1011 şi trecem rezultatul în stânga sub primi 5 biţi ai deîmpărţitului;
- Scădem 1011 din 10010 (respectând regulile scăderii în binar) şi obţinem restul 111;

- Coborâm bitul deîmpărţitului, care este 0 (vezi săgeata) şi obţinem restul 1110;
- Deoarece restul 1110 este mai mare decât împărţitorul 1011

  câtul este 1.

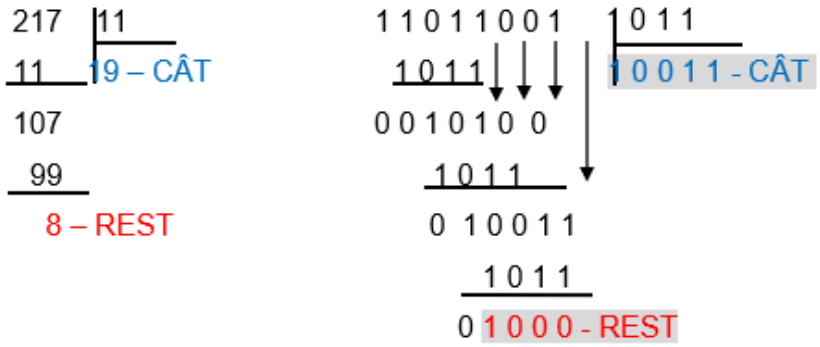
  Deci *al doilea bit al câtului este 1*;
- Înmulţim câtul 1 cu împărţitorul 1011 şi trecem rezultatul în stânga sub restul 1110;
- Scădem 1011 din 1110 (respectând regulile scăderii în binar) şi obţinem restul 11;
- Coborâm bitul deîmpărţitului, care este 1 (vezi săgeata) şi obţinem restul 111;
- Deoarece restul 111 este mai mic decât împărţitorul 1011

  câtul este 0.

  Deci *al treilea bit al câtului este 0;*
- Coborâm bitul deîmpărţitului, care este 1 (vezi săgeata) şi obţinem restul 1111;
- Deoarece restul 1111 este mai mare decât împărţitorul 1011 câtul este 1.

  Deci *al patrulea bit al câtului este 1;*
- Înmulţim câtul 1 cu împărţitorul 1011 şi trecem rezultatul în stânga sub restul 1111;
- Scădem 1011 din 1111 (respectând regulile scăderii în binar) şi obţinem restul 100.

Exemplul 2.



Algoritmul împărţirii deîmpărţitului 1101100 la împărţitorul 1011:

- Deoarece numărul format din primi 4 biţi ai deîmpărţitului 1101 este mai mare decât 1011.

  Deci *primul bit al câtului este 1*;

- Înmulţim câtul 1 cu împărţitorul 1011 şi trecem rezultatul în stânga sub primi 4 biţi ai deîmpărţitului;
- Scădem 1011 din 1101 (respectând regulile scăderii în binar) şi obţinem restul 10;
- Coborâm bitul deîmpărţitului, care este 1 (vezi săgeata) şi obţinem restul 101;
- Deoarece restul 101 este mai mic decât împărţitorul 1011 câtul este 0.
  Deci *al doilea bit al câtului este 0*;
- Coborâm bitul deîmpărţitului, care este 0 (vezi săgeata) şi obţinem restul 1010;
- Deoarece restul 1010 este mai mic decât împărţitorul 1011 câtul este 0.
  Deci *al treilea bit al câtului este 0*;
- Coborâm bitul deîmpărţitului, care este 0 (vezi săgeata) şi obţinem restul 10100;
- Deoarece restul 10100 este mai mare decât împărţitorul 1011 câtul este 1.
  Deci *al patrulea bit al câtului este 1*;
- Înmulţim câtul 1 cu împărţitorul 1011 şi trecem rezultatul în stânga sub restul 10100;
- Scădem 1011 din 10100 (respectând regulile scăderii) şi obţinem restul 1001;
- Coborâm bitul deîmpărţitului, care este 1 (vezi săgeata) şi obţinem restul 10011;
- Deoarece restul 10011 este mai mare decât împărţitorul 1011 câtul este 1.
  Deci *al cincilea bit al câtului este 1*;
- Înmulţim câtul 1 cu împărţitorul 1011 şi trecem rezultatul în stânga sub restul 10011;
- Scădem 1011 din 10011 (respectând regulile scăderii) şi obţinem restul 1000.

Operaţii simple cu numere scrise în diferite baze

În continuare vor fi prezentate operaţiile de adunare şi scădere a numerelor scrise în binar, octal şi hexazecimal a numerelor întregi fără semn.

Adunarea

Adunarea se face după aceleaşi reguli ca în zecimal, cu observaţia că cifra cea mai mare dintr-o baza “b” va fi b-1 (adică 9 in zecimal, 7 in octal, 1 in binar şi F în hexazecimal). Deci dacă prin adunarea a două cifre de rang “i” se va obţine un rezultat mai mare decât b-1, va apare acel transport spre cifra de rang următor “i”+1, iar pe poziţia de rang “i” va rămâne restul împărţirii rezultatului adunării cifrelor de rang “i” la bază. Transportul spre cifra de rang “i”+1 va deveni o nouă unitate la suma cifrelor de rang “i”+1, adică se va mai aduna acel transport 1.

Exemple:

1111 1	11	11
01010110(B)+	1364(Q)+	6D8A32(H)+
10110101(B)	3721(Q)	33E4C8(H)
100001011(B)	5305(Q)	A16EFA(H)

S-a marcat transportul de o unitate la cifra de rang superior prin scrierea unui 1 deasupra cifrei de rang superior la care s-a făcut transportul. Operaţia de adunare în binar este utilă la reprezentarea numerelor în complement faţă de 2 când se alege varianta adunării valorii 1 la reprezentarea din complement faţă de 1 (vezi lucrarea cu reprezentarea datelor).

Exemple:

- Să se adune cele 2 numere întregi 347(D) şi 438(D) convertite mai sus în lucrare în bazele 16 si 8 si să se verifice rezultatul prin conversia lui in baza 10

347(D)+ 438(D) = 785(D)
15B(H) + 1B6(H) = 311(H). Verificare: 311(H) = 3*256+1*16+1 = 785
533(Q) + 666(Q) = 1421(Q). Verificare:1421(Q) = 1*512+4*64+2*8+1 = 785

Scăderea

Si pentru scădere sunt valabile regulile de la scăderea din zecimal şi anume: dacă nu se pot scădea două cifre re rang “i” (adică cifra descăzutului este mai mică decât a scăzătorului) se face “împrumut” o unitate din cifra de rang următor (“i”+1). În cazul în care cifra din care se doreşte realizarea “împrumutului” este 0, se face “împrumutul” mai departe la cifra de rang următor.

Exemple:

01011010(B) –	A3D4(H) –
01001100(B)	751B(H)
00001110(B)	2EB9(H)

- Să se scadă cele două numere întregi 347(D) şi 438(D) convertite mai sus în lucrare în bazele de numeraţie 16 şi 8 şi să se verifice rezultatul prin conversia lui în zecimal.

438 – 347 = 91(D)
1B6(H) – 15B(H) = 5B(H). Verificare: 5B(H) = 5*16+11 = 91
666(Q) – 533(Q) = 133(Q). Verificare 133(Q) = 1*64+3*8+3 = 91

Operaţia de scădere este utilă când se doreşte reprezentarea numerelor în complement faţă de 2 şi se efectuează scăderea din 2nr_biti_reprez + 1 a numărului reprezentat în modul.

Se vor realiza conversiile din exemplele prezentate:

- conversia numerelor din baza 10 in baza 2,8 şi 16
- conversia unui număr intre bazele 2,8 şi 16
- conversii din bazele 2,8 şi 16 în baza 10
- conversia numerelor zecimale din baza 10 în baza 2 şi 16
- operaţii de adunare şi scădere numere în bazele 2 şi 16

Anexa 1

Hexazecimal baza 16	Zecimal baza 10	Calcul
0	0	-
1	1	-
2	2	-
3	3	-
4	4	-
5	5	-
6	6	-
7	7	-
8	8	-
9	9	-
A	10	-
B	11	-
C	12	-
D	13	-
E	14	-
F	15	-
10	16	$1\times16^1+0\times16^0 = 16$

11	17	$1\times16^1+1\times16^0 = 17$
12	18	$1\times16^1+2\times16^0 = 18$
13	19	$1\times16^1+3\times16^0 = 19$
14	20	$1\times16^1+4\times16^0 = 20$
15	21	$1\times16^1+5\times16^0 = 21$
16	22	$1\times16^1+6\times16^0 = 22$
17	23	$1\times16^1+7\times16^0 = 23$
18	24	$1\times16^1+8\times16^0 = 24$
19	25	$1\times16^1+9\times16^0 = 25$
1A	26	$1\times16^1+10\times16^0 = 26$
1B	27	$1\times16^1+11\times16^0 = 27$
1C	28	$1\times16^1+12\times16^0 = 28$
1D	29	$1\times16^1+13\times16^0 = 29$
1E	30	$1\times16^1+14\times16^0 = 30$
1F	31	$1\times16^1+15\times16^0 = 31$
20	32	$2\times16^1+0\times16^0 = 32$
30	48	$3\times16^1+0\times16^0 = 48$
40	64	$4\times16^1+0\times16^0 = 64$
50	80	$5\times16^1+0\times16^0 = 80$
60	96	$6\times16^1+0\times16^0 = 96$
70	112	$7\times16^1+0\times16^0 = 112$
80	128	$8\times16^1+0\times16^0 = 128$
90	144	$9\times16^1+0\times16^0 = 144$
A0	160	$10\times16^1+0\times16^0 = 160$
B0	176	$11\times16^1+0\times16^0 = 176$
C0	192	$12\times16^1+0\times16^0 = 192$
D0	208	$13\times16^1+0\times16^0 = 208$

E0	224	$14\times16^1+0\times16^0 = 224$
F0	240	$15\times16^1+0\times16^0 = 240$
100	256	$1\times16^2+0\times16^1+0\times16^0 = 256$
200	512	$2\times16^2+0\times16^1+0\times16^0 = 512$
300	768	$3\times16^2+0\times16^1+0\times16^0 = 768$
400	1024	$4\times16^2+0\times16^1+0\times16^0 = 1024$

DESCRIEREA GENERALĂ A UNUI SISTEM DE CALCUL

1. CONCEPTE GENERALE

Noţiunea de sistem

Sistem (definiţie) = Un ansamblu de elemente inter-relaţionate ce compun un întreg. Termenul de „system” in latină şi 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 şi interacţionează pentru a facilita fluxul de informaţie. În funcţie de tipul sistemului acesta se poate diferenţia de elemente, maşini, procese.

Tipuri de sisteme:
• **Sisteme deschise** – Sisteme care pot fi influenţate de evenimentele din afara graniţelor lor.
• **Sisteme închise** - Sisteme care nu sunt influenţate de evenimentele din afara graniţelor lor.
• **Sisteme dinamice** – Sisteme cu componente sau fluxuri care se schimbă în timp.
O distincţie care trebuie specificată este între sistemele fizice şi cele conceptuale. Cele conceptuale sunt abstracte şi au la bază idei. ajutând la modelarea sistemelor fizice.

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

2. SISTEMUL DE CALCUL

2.1. Definiţie

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

Componentele sistemului de calcul sunt:

- hardware – echipamente
- software - programe

2.2. Evoluţia sistemelor de calcul

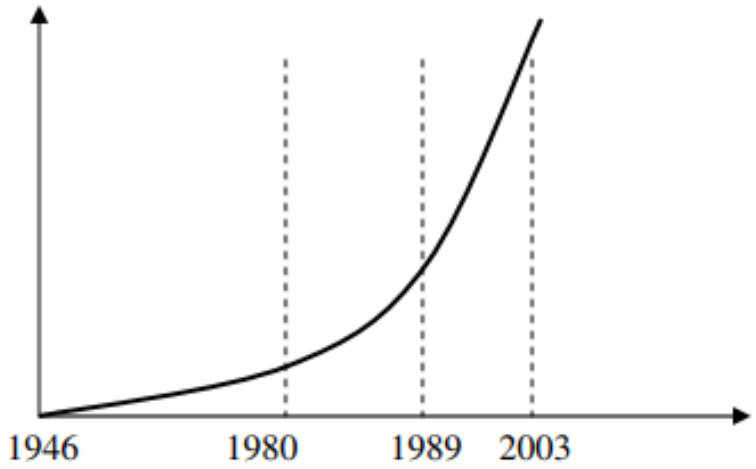
În anul 1946 John von Neumann a lansat ideea programului înregistrat, pentru care o maşină de calcul trebuie să fie dotată cu un dispozitiv de memorare a datelor şi comenzilor, care trebuie să lucreze cu o viteză mare şi să permită înregistrarea simplă şi rapidă a informaţiilor. Astfel, au apărut noţiunile de algoritm de rezolvare a unei probleme şi programul de prelucrare a algoritmului, a secvenţelor de comenzi şi memorare date.

John von Neumann a recomandat constructorilor de calculatoare 3 principii care să fie utilizate la realizarea unor calculatoare complexe şi rapide:

- programele şi datele trebuie să fie codificate sub formă binară;
- programele şi datele trebuie păstrate în memoria calculatorului, concept care stă la baza calculatoarelor moderne;
- trebuie să existe o unitate centrală de prelucrare care trebuie să poată extrage, decodifica şi executa instrucţiunile programului.

Evoluţie: se observă o crestere accentuata in ultimele doua decenii

- dupa1989 exploziva
- apar o concepte noi o domenii noi



Din punct de vedere al utilizatorului obişnuit progresul a fost realizat datorită:

- accesabilităţii
  - disponibilitatea calculatoarelor prin preţul mic al acestora
  - uşurinţa în utilizare - interfaţa grafică prietenoasă, sugestivă
- funcţionalităţii crescute
  - prin creşterea resurselor de calcul şi memorare (viteză de execuţie şi memorie)
  - înglobarea a mai multor funcţii complexe
- suportului de comunicare oferit
  - prin folosirea infrastructurii existente de comunicaţie
  - posibilitatea de comunica diferite tipuri de informaţii: poştă, multimedia, etc.

2.3. Arhitectura sistemelor de calcul

Arhitectura sistemelor de calcul sau **arhitectura calculatoarelor** este teoria din spatele construcţiei unui calculator.
În acelaşi mod în care un arhitect stabileşte principiile şi obiectivele construirii unui proiect ca baze ale unor planuri de construcţie, în acelaşi mod un arhitect de calculatoare stabileşte arhitectura sistemului de calcul ca bază a specificaţiilor de proiectare.
Obiectivul principal în arhitectura unui sistem de calcul îl reprezintă un **raport cost/performanţă cât mai bun.**

Arhitectura sistemului de calcul = dispunerea si interconectarea componentelor pentru a obtine functionalitatea dorita a sistemului de calcul.

Arhitecturi generale:

1. **Arhitectura multistrat** – niveluri ierarhice. Un nivel inferior ofera suport nivelului superior pentru executia functiilor sale

2. **Decompozitia functionala** – descompunere a componentelor dupa functiile realizate

3. **Decompozitia conceptuala** – descompunere a sistemului dupa entitatile identificate (ce inglobeaza toata functionalitatea obiectului).

2.4. Descrierea sistemului de calcul

Definiţie. Masina de calcul care executa secvential programe scrise in limbajul masinii respective, stocate in memorie, in interactiune cu mediul extern.

Un **program** = solutie algoritmica a unei probleme scrisa intr-un limbaj, numit **limbaj de programare.**

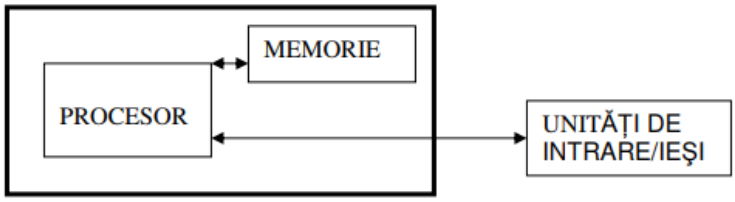
Program = succesiune de instructiuni ce implementeaza un algoritm.

Un **algoritm**=solutie secventiala a unei probleme.

Limbajul de programare≠limbajul masina.

Limbajul masina este limbajul executat de masina.

Limbajul de programare este translatat in limbaj masina pentru executie.
Sunt doua forme de executie:
- compilare si executie
- interpretare (masina virtuala care interpreteaza si executa programul)



Elemente componente:
- de procesare (prelucrare)
- de memorare
- de comunicatie
  - cu mediul extern
  - suport de comunicaţie

Din punct de vedere perceptiei, sistemul de calcul este impartit in doua mari parti:

• **partea hardware** – reprezentata de circuitele electronice, placi, cabluri, memorii, etc. care reprezinta echipamentul propriu-zis de calcul si care sunt tangibile.

• **partea software** – reprezentata prin programe care implementeaza algoritmi si reprezinta idei abstracte.

Diferenta dintre hardware si software pina nu demult a fost evidenta, cu timpul insa ele au devenit logic echivalente. Ambele putind realiza aceleasi functii, iar alegerea implementarii facindu-se dupa criterii pret/performanta.

2.5. Partile componente ale sistemului de calcul

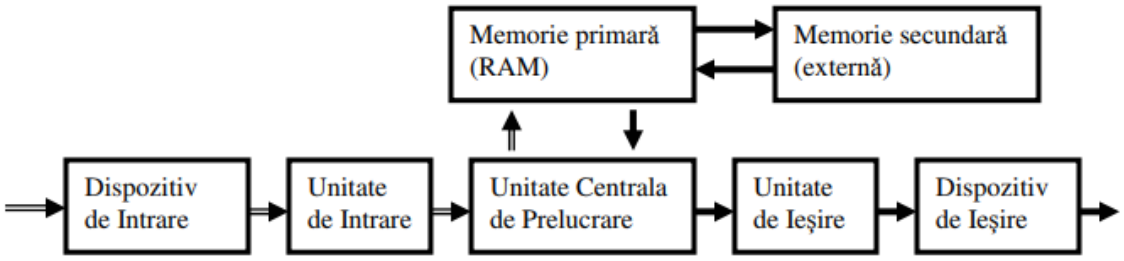
Fluxul informaţiei

Pentru a intelege functionarea calculatorului vom introduce notiunea de informatie care, furnizata de utilizator sau mediu, este convertita în format binar, intern, prelucrat de sistemul de calcul (date).

Adoptarea reprezentarii binare a fost impusa de utilizarea in constructia calculatoarelor a dispozitivelor cu doua stari stabile, notate conventional cu 0 si 1. Unitatea de masura pentru numerele binare este bit-ul (binary digit).

În continuare sunt prezentare succint componentele sistemelor de calcul cu referire directa la calculatoare. Numeroasele componente ale unui sistem de calcul pot fi grupate în unități având funcții mai complexe bine precizate.

În figura urmatoare denumirea fiecărei unități indică funcția ei, iar săgețile de legatură arată modul de transmitere a informației de la una la alta.



Informatia, furnizata de mediul extern (utilizator), este preluată de dispozitivul de intrare, codificată (convertită în format binar) şi transmisă unităţii de intrare care realizează interfaţa cu unitatea de procesare (unitatea centrală de procesare = UCP) Exemple de dispozitive de intrare: tastatura, mouse, scanner, MODEM, etc. Astfel, la tastatura, apasarea unei taste produce codului binar corespunzator al tastei apăsate. Scannerul preia o imagine şi o transformă într-o succesiune de coduri binare. MODEM-ul preia datele transmise de la distanţă.

Unitatea de intrare realizează interfaţa cu UCP a.î. dispozitivele de intrare pot fi diferite.

Informatia este înregistrată și păstrată în memoria primară. De aici ea poate fi transmisa ulterior altor unități funcționale. Informația este supusa prelucrarii in UCP. UCP este formată din unitatea de calcul şi unitatea de comandă. Unitatea de calcul efectueaza operatii simple, aritmetice si logice, asupra unor operanzi din memorie, inregistrind rezultatele tot in 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 păstrată în unități de memorie externă (de obicei discuri magnetice), mai lente decât memoria internă (operativă) dar cu o capacitate mai mare. Informatia poate fi transmisa, daca este cazul, de la o memorie la alta.
Din punct de vedere al localizării, memoria sistemului de calcul este împărţită în:
- memorie internă
- memorie externă
Memoria internă este formată din:

- **RAM (Read Only Memory**) – volatila (îşi pierde conţinutul dacă nu este alimentată cu curent electric) = memoria primară a sistemului de calcul (indispensabilă)

- PROM (**Programmable Read Only Memory**) - nevolatila = memorie care poate fi doar citită şi care cuprinde programe ale fabricanţilor sistemului de calcul

- **CMOS (Complementar Metal Oxid Siliciu)** – memorie asemănătoare memoriei RAM, de capacitate foarte mică, alimentată permanent de la o baterie, care păstrează setările de configurare a sistemului de calcul.

Memoria externa - **discuri magnetice şi optice.**

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

1.2.6. Părţi hardware principale componente:

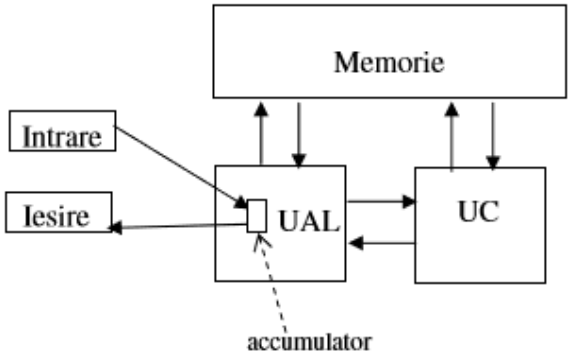
- UCP (Unitatea Centrală de Procesare)
- Memoria
- Subsistemul de I/E
- Suportul de comunicaţie

UCP are rolul de a prelucra programul alcătuit din instrucţiuni şi de a controla activitatea întregului sistem.
Memoria este cea în care se stochează informaţia în format binar.
Subsistemul de I/E realizează interfaţa cu mediul exterior.
Suportul de comunicaţie reprezintă infrastructura de comunicaţie necesară transmiterii datelor.

3. MODELUL VON NEUMANN

Modelul von Neumann a fost specificat de von Neumann în 1946 şi cuprinde elementele pricipale componente ale unui sistem de calcul şi interacţiunea între ele. El are la baza patru componente: memoria, unitatea aritmetica si logica cu un registru intern special numit accumulator, unitatea de comandă și echipamentele de intrare si de iesire.

Arhitectura masinii von Neumann este cea din figura urmatoare:



Această arhitectură stă la baza calculatoarelor moderne cu un procesor.

Unităţile componente se numesc şi **unităţi funcţionale** deoarece fiecare are o funcţionalitate specifică în sistem.

Unitatea centrală (UC) are rolul de a prelucra programul alcătuit din instrucţiuni şi de a controla activitatea întregului sistem. Instrucţiunea este memorată ca secventa de biţi în memorie.

UC este alcătuită din:

- unitatea de comandă
- unitatea aritmetică şi logică (UAL) sau unitatea de calcul
- regiştrii.

Unitatea de comandă decodifică instrucţiunile, le interpretează şi comandă operaţiile de executat.

Unitatea de calcul execută operaţiile aritmetice şi logice.

Regiştrii sunt folosiţi pentru stocarea temporară a datelor de prelucrat. În modelul von Neumann este specificat doar registrul acumulator. Ulterior setul de regiştri a fost îmbogăţit cu mai multe tipuri, în functie de datele stocate (adrese, date, stare, etc.).

Memoria este cea în care se stochează informaţia în format binar. De aceea suportul de memorie trebuie să asigure două stări stabile distincte.
Ea este compusă dintr-un şir de **locaţii de memorie**, iar accesul la ele se face prin **adrese**. Locaţia de memorie are dimensiunea de un **octet**. Numărul locaţiilor de memorie formează **capacitatea de stocare** a memoriei. În memorie sunt stocate atât date cât şi programe.
Operaţiile care se execută cu memoria sunt: de citire şi de scriere

Într-un sistem de calcul există mai multe tipuri de memorie, dispuse ierarhic în sistem, în funcţie de timpul de acces şi de preţul acestora.
Astfel, pe nivelul inferior se află **memoriile de masă** reprezentate prin discuri magnetice, CD-uri, având timp mare de acces, preţ mic şi corespunzător capacitate mare. În memoriile de masă se stochează informaţia 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 şi capacitatea de stocare mult mai mică decât cea a memoriei de masă. Caracteristica principală memoriei primare este volatilitatea, adică pierderea informaţilei la căderea tensiunii. Cu această memorie se efectuează atât operaţii de scriere cât şi operaţii 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 alfă regiştrii care sunt cei mai rapizi aflându-se direct conectaţi la unităţile 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 stocheaza programele furnizate de fabricantul calculatorului. Singura operaţie efectuată cu acest tip de memoruie este citirea ei.
- **memoria CMOS** (numele este dat de tehnologia de realizare a acesteia). În această memorie sunt păstrate informaţii de configurare a calculatorului. Cu ea se realizează operaţii de citire şi scriere şi este alimentată de la o baterie.

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

Unitatea de ieşire este componenta care asigură funcţia de furnizare a informaţiilor la ieşire. Acestea sunt **scrise** şi transmise unui **dispozitiv de ieşire**.

Pentru sincronizarea unităţilor funcţionale componente, există între ele **interfeţe** în care se află **buffere** (elemente temporare de memorare).
Informatia este transmisă în sistem pe căi electrice numite **magistrale**.
În funcţie de tipul de informaţie care circulă prin ele, ele se clasifică în: **magistrale de adrese**, **magistrale de date**, **magistrale de control**.
Pentru a se obţine flexibilitate în interconectarea diverselor componente ale sistemului de calcul, magistralele sunt standardizate.