

### CALCULATOARE NUMERICE

*Calculatorul numeric* reprezintă un sistem digital, format din dispozitive fizice conectate în vederea prelucrării informațiilor numerice. Un calculator numeric (sistem de calcul) cuprinde două categorii de componente:

- Componente fizice (echipamente), ansamblul lor fiind cunoscut sub numele de *hardware*;
- Componente logice (programe), prin intermediul cărora sunt utilizate echipamentele, ansamblul componentelor logice fiind cunoscut sub numele de *software*.

Sistemul de calcul reprezintă deci un ansamblu de componente hardware și software în interacțiune, destinat prelucrării datelor. Un asemenea sistem se caracterizează prin următoarele:

- Componentele funcționale sunt realizate cu ajutorul circuitelor electronice, majoritatea fiind circuite integrate, asigurând o viteză ridicată în efectuarea operațiilor aritmetice și fiabilitate în funcționare.
- Funcționează pe baza unui program memorat, format dintr-o succesiune de instrucțiuni introduse în memoria calculatorului, instrucțiuni care sunt extrase din memorie, interpretate și executate.
- Informația memorată și prelucrată este una discretă, fiind codificată astfel încât mărimile asupra căreia operează pot lua numai două valori distincte (0 și 1). O astfel de informație se numește *informație binară*.

### PROGRAMAREA CALCULATOARELOR NUMERICE

Pentru rezolvarea unei probleme cu ajutorul calculatorului, este necesară descrierea acesteia printr-o succesiune de operații, care constituie *algoritmul* de rezolvare. Descrierea se poate realiza, la început, în limbaj natural, transpunându-se apoi într-un limbaj artificial, numit *limbaj de programare*.

*Limbajul de programare* reprezintă un set de instrucțiuni, împreună cu regulile de organizare ale acestora într-un program. Totalitatea simbolurilor și cuvintelor folosite în limbajul respectiv constituie *vocabularul* limbajului. Totalitatea regulilor de scriere ale instrucțiunilor reprezintă *sintaxa* limbajului, iar totalitatea regulilor prin care se asociază o semnificație instrucțiunilor reprezintă *semantica* limbajului.

Calculatorul poate executa numai instrucțiuni exprimate intern sub forma unor șiruri de cifre binare. Programele în care instrucțiunile sunt scrise sub această formă se numesc programe în *limbaj mașină*. Limbajul mașină este caracteristic fiecărui tip de calculator. Scrierea și introducerea programelor sub această formă este dificilă. Pentru simplificarea scrierii programelor, au fost create limbaje în care fiecărui cod de instrucțiune i s-a atașat un *nume mnemonic*. Acest mod de reprezentare a instrucțiunilor mașină se numește *limbaj simbolic* sau *limbaj de asamblare*. Acest limbaj permite utilizatorului să realizeze codificări simbolice ale instrucțiunilor și ale adreselor de memorie.

Pentru a se executa un program în limbaj de asamblare, acesta trebuie translatat în prealabil în limbaj mașină, printr-un proces numit *asamblare*. Programul care se translatează se numește *program sursă*, iar cel rezultat în urma translatării se numește *program obiect*. Operația de translatare a programului sursă în program obiect este executată de un program special numit *asamblor*.

O instrucțiune în limbaj de asamblare corespunde unei instrucțiuni în limbaj mașină, deosebirea dintre ele constând numai în modul de reprezentare. Deci, fiecare instrucțiune în limbaj de asamblare este translatată într-o singură instrucțiune în limbaj mașină.

Un limbaj de asamblare este specific unui calculator, astfel încât trebuie să se cunoască instrucțiunile și organizarea internă a acelui calculator. Din acest motiv, s-au elaborat limbaje pentru programe care se pot executa pe orice calculator. Acestea sunt limbaje orientate pe probleme sau *limbaje de nivel înalt*, spre deosebire de limbajele de asamblare, care sunt limbaje orientate pe calculator sau *limbaje de nivel scăzut*. Asemenea limbaje de nivel înalt sunt *Pascal*, *C*, *BASIC*, *FORTRAN*, *LISP*, *COBOL*, *PROLOG* etc.

Unei instrucțiuni într-un limbaj de nivel înalt îi corespunde o succesiune de instrucțiuni în limbaj mașină. Translatarea în limbajul mașină se poate realiza cu ajutorul unui *compiler*, care generează din programul sursă un *program executabil*, acesta fiind executat după ce întregul program a fost compilat.

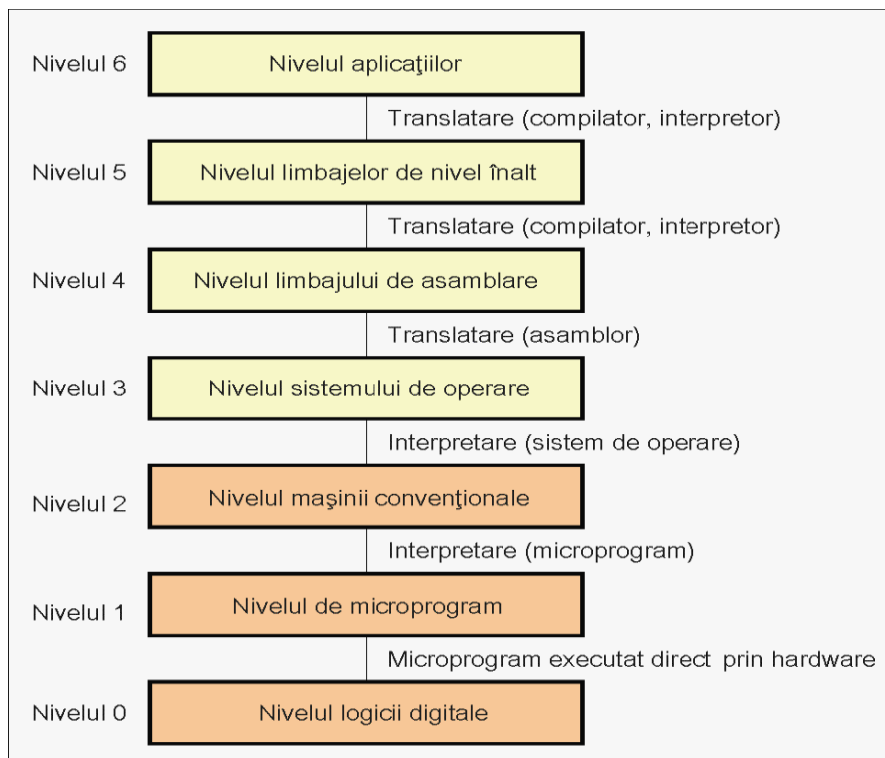
O altă posibilitate este utilizarea unui *interpretor*, care translatează fiecare instrucțiune în limbajul de nivel înalt într-o succesiune de instrucțiuni mașină, acestea fiind executate imediat. În acest caz nu se generează un program executabil. Viteza de execuție este însă redusă, deoarece fiecare instrucțiune trebuie interpretată chiar dacă ea este executată în mod repetat.

Uneori este mai convenabil să se considere că există un calculator ipotetic sau o *mașină virtuală*, a cărei limbaj mașină este un anumit limbaj de nivel înalt. Un asemenea calculator ar executa direct instrucțiunile limbajului de nivel înalt, fără a fi necesară utilizarea unui translator

(compiler) sau interpretor. Chiar dacă implementarea unei mașini virtuale care să lucreze direct cu un limbaj de nivel înalt ar fi prea costisitoare, se pot scrie programe pentru această mașină, deoarece aceste programe pot fi traduse sau interpretate cu un program care poate fi executat direct de calculatorul existent.

#### MODELUL UNUI CALCULATOR NUMERIC

Un model posibil al unui calculator numeric modern reprezintă o ierarhie de mașini virtuale pe mai multe nivele (Figura 1).



**Figura 1.** Ierarhia de niveluri a unui calculator modern.

Nivelul 0, numit *nivelul logicii digitale*, este reprezentat de componentele hardware ale calculatorului (mașina fizică). Circuitele acestui nivel execută instrucțiunile mașinii ale nivelului 1. Elementele de bază ale acestor circuite sunt *porțile logice*, fiecare poartă fiind formată la rândul ei dintr-un număr de tranzistoare. O poartă logică are una sau mai multe intrări digitale (semnale reprezentând 0 logic sau 1 logic), și are ca ieșire o funcție simplă a acestor intrări, de exemplu ȘI logic, SAU logic.

Nivelul 1, numit *nivelul de microprogram*, interpretează instrucțiunile nivelului 2, pentru fiecare instrucțiune a acestui nivel existând câte un microprogram. Fiecare microprogram definește în mod implicit un limbaj de nivel 2 și o mașină virtuală. De obicei există multe similarități între mașinile virtuale de nivel 2, chiar și în cazul calculatoarelor diferiților producători. Acest nivel se

numește *nivelul mașinii convenționale*. Atunci când se descrie setul de instrucțiuni al unui calculator, se descrie de fapt mașina virtuală de nivel 2, și nu mașina reală de nivel 1. Sunt descrise deci instrucțiunile interpretate de către microprogram, și nu instrucțiunile executate direct prin hardware.

De menționat că la anumite calculatoare nivelul de microprogram lipsește. La aceste calculatoare, instrucțiunile mașinii convenționale sunt executate direct de circuitele electronice ale nivelului 0.

Nivelul 3 este de obicei un nivel hibrid, deoarece multe din instrucțiunile limbajului său sunt prezente în cadrul instrucțiunilor nivelului 2. Există în plus un set de noi instrucțiuni, o organizare diferită a memoriei, posibilitatea de execuție a mai multor programe în paralel și alte facilități. Noile facilități adăugate la nivelul 3 sunt realizate cu ajutorul unui interpretor, numit *sistem de operare*, iar instrucțiunile identice cu cele ale nivelului 2 sunt executate direct prin microprogram. Nivelul 3 este numit *nivelul sistemului de operare*.

Un sistem de operare reprezintă un ansamblu de programe care asigură exploatarea optimă a resurselor hardware și software ale unui sistem de calcul. Sistemele de operare au fost create pentru simplificarea activității de programare, utilizarea optimă a posibilităților de lucru ale echipamentelor și reducerea intervenției utilizatorilor în cursul execuției programelor.

Nivelele 0-3 nu sunt destinate utilizării directe de către programatorii obișnuiți, ci pentru rularea translatoarelor și interpretoarelor scrise de programatorii de sistem.

Nivelul 4 și nivelele superioare sunt destinate programatorilor de aplicații. Nivelul 4 este *nivelul limbajului de asamblare*. Programele scrise în limbaj de asamblare sunt translate în limbajul nivelului 1, 2 sau 3 și apoi interpretate de către mașina virtuală corespunzătoare.

Nivelul 5 constă din limbajele destinate programatorilor de aplicație, fiind numit *nivelul limbajelor de nivel înalt*. Programele scrise în aceste limbaje sunt translate în limbajele nivelului 3 sau 4 cu ajutorul compilatoarelor sau interpretoarelor.

Nivelul 6 reprezintă *nivelul aplicațiilor*. Constă din colecții de programe destinate unor domenii specializate, de exemplu pentru administrație, economie, proiectare asistată de calculator, grafică etc.

Fiecare nivel reprezintă o abstractizare distinctă, cu diferite obiecte și operații. Setul tipurilor de date, a operațiilor și facilităților fiecărui nivel reprezintă arhitectura nivelului respectiv. Arhitectura tratează acele aspecte care sunt vizibile utilizatorului nivelului respectiv, ca de exemplu dimensiunea memoriei disponibile. Aspectele de implementare, ca de exemplu tehnologia utilizată

pentru implementarea memoriei, nu fac parte din arhitectură. *Arhitectura calculatorului* reprezintă studiul proiectării acelor părți ale unui sistem de calcul care sunt vizibile pentru programatori.