

CURS 8

UNITATEA CENTRALĂ

Setul de instrucțiuni. Activitatea generală a unui sistem de calcul constă în stocarea și prelucrarea informațiilor. Întregul proces de tratare a informației în cadrul unui sistem de calcul are loc conform unui *algoritm de prelucrare*. Acest algoritm poate fi modelat prin seturi de instrucțiuni specifice unui limbaj de programare. Instrucțiunile alcătuiesc un program. Indiferent de limbajul de programare utilizat, instrucțiunile unui program trebuie aduse, printr-un set de transformări succesive, până la nivelul unui set de comenzi elementare care pot fi executate de sistemul de calcul și care formează setul de instrucțiuni al calculatorului. Aceste instrucțiuni aparțin unui limbaj intern, propriu fiecărui sistem de calcul. Limbajul se numește *limbaj-mașină*. Limbajul-mașină cuprinde un set restrâns de instrucțiuni elementare de forma unor coduri binare de lungime dată.

Componența sistemului de calcul care realizează executarea instrucțiunilor este *unitatea centrală de prelucrare*. (UCP). Pentru aceasta, unitatea centrală de prelucrare trebuie să poată efectua următoarele funcții:

- citirea și scrierea informațiilor în memoria internă (MI);
- recunoașterea și executarea instrucțiunilor-mașină;
- transmiterea de comenzi celorlalte componente ale sistemului de calcul, realizând astfel coordonarea funcționării sistemului de calcul.

Formatul instrucțiunilor specifică numărul de cuvinte de memorie utilizate pentru codificarea fiecărui tip de instrucțiune și semnificația câmpurilor care formează instrucțiunea.

Formatul folosit se stabilește în faza de proiectare a mașinii, atunci când se proiectează setul de instrucțiuni-mașină. În general o instrucțiune-mașină trebuie să cuprindă două câmpuri:

- codul operației, adică operația ce trebuie efectuată (operație aritmetică, logică);
- zona de adrese, care trebuie să cuprindă la rândul ei un număr de adrese pentru operanzi, o adresă pentru rezultat și adresa instrucțiunii care urmează.

Pentru reducerea lungimii instrucțiunilor-mașină s-au stabilit convenții de reprezentare a unei instrucțiuni.

1. Instrucțiunea poate opera cu cel mult doi operanzi.
2. Adresa următoarei instrucțiuni ce trebuie executată se citește întotdeauna dintr-un registru specializat, numit *registru-contor de adrese*. La executarea fiecărei instrucțiuni, acest registru este mărit (incrementat) cu lungimea instrucțiunii pe care a executat-o.
3. Într-un program încărcat în memorie, pentru a fi executat, instrucțiunile succesive se află la adrese consecutive de memorie.
4. Pentru realizarea unei ramificații în execuția unui program se folosesc instrucțiuni speciale de salt, care modifică valoarea memorată în registrul contor de adresă, încărcând în acesta valoarea adresei noii instrucțiuni la care se va face saltul.
5. Adresa de destinație a unei operații este identică cu una din adresele-sursă, caz în care acest operand-sursă este distrus în urma executării instrucțiunii.
6. Cel puțin una dintre adresele celor doi operanzi trebuie să fie adresa unuia dintre regiștrii unității centrale de prelucrare. Un calculator care respectă această regulă se numește calculator cu o singură adresă.

Executarea unei instrucțiuni. Sub controlul unui program și cu datele memorate în memoria internă, unitatea centrală de prelucrare execută fiecare instrucțiune în două etape, ca o secvență de pași sincronizați în timp.

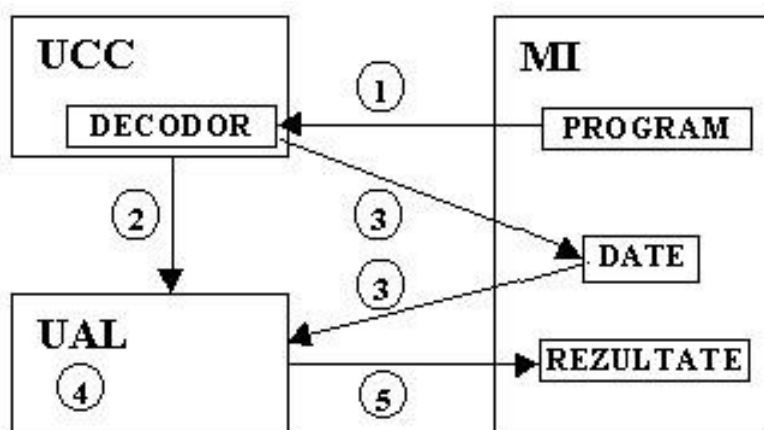


Figura 1. Execuția unei instrucțiuni mașină.

Funcțiile principale realizate de această unitate (UCP):

1. *Extragerea instrucțiunilor:* UCP trebuie să citească instrucțiunile din memorie. Instrucțiunea se încarcă de UCC

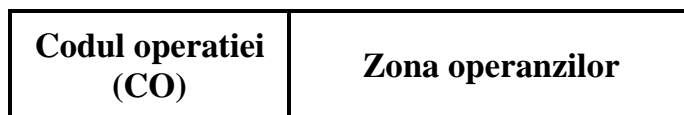
2. *Interpretarea instrucțiunilor*: Fiecare instrucțiune trebuie decodificată pentru a determina operația care trebuie executată; decodificare instrucțiune și emitere ordin către UAL
 3. *Citirea datelor*: Pentru execuția unor instrucțiuni este necesară citirea datelor din memorie sau de la un dispozitiv de I/E.
 4. *Prelucrarea datelor*: Execuția unei instrucțiuni poate necesita execuția unei operații aritmetice sau logice asupra datelor.
 5. *Scrierea datelor*: Rezultatele execuției trebuie scrise în memorie sau transmise la un dispozitiv de I/E.
- Citirea instrucțiunii și pregătirea operanzilor care reprezintă interpretarea de fapt a instrucțiunii. În această etapă se transferă adresa instrucțiunii de executat din registrul contor-program în registrul de adresă al memoriei; de aici se transferă în registrul de instrucțiuni instrucțiunea citită din memoria internă. Pentru instrucțiunile mai lungi secvența se repetă. Se modifică valoarea memorată în contorul de adrese pentru a puncta următoarea instrucțiune.
 - Executarea instrucțiunii presupune executarea unui set de comenzi succesive corespunzătoare codului operației descrise în instrucțiunea ce se execută.
 - O comandă se concretizează într-un cuvânt de comandă care determină o acțiune a unității aritmetice logice, a memoriei sau a dispozitivelor de intrare-ieșire. Aceste comenzi succesive se mai numesc și *micro-operații*.
 - O micro-operație se desfășoară pe parcursul unui *ciclu-mașină*.
 - Ciclul-mașină reprezintă timpul total necesar executării comenzii, măsurat în *impulsuri de tact*.
 - Cu cât o instrucțiune se compune din mai multe micro-operații, cu atât durata ei de execuție crește.
 - Timpul total pentru execuția instrucțiunii se numește *ciclu-instrucțiune* și se măsoară tot în impulsuri de tact.

Pentru a reduce timpul de execuție al unei instrucțiuni se poate recurge la executarea concurentă a mai multor micro-operații. Ciclul-instrucțiune al unui sistem de calcul este cu atât mai bun cu cât frecvența impulsurilor de tact este mai mare, cu cât micro-operațiile sunt mai complexe sau când micro-operațiile se pot executa concurent.

Structura unei instrucțiuni-mașină.

În general o instrucțiune-mașină trebuie să cuprindă două câmpuri:

- * codul operației, adică operația ce trebuie efectuată;
- * zona de adrese, conține adrese operanți.



Formatul unei instrucțiuni-mașina

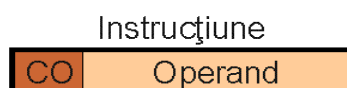
Codul operației reprezintă ceea ce trebuie să facă instrucțiunea respectivă, și acest cod trebuie să se regăsească în setul de coduri admise de unitatea centrală de prelucrare. Zona operanzilor diferă de la o instrucțiune la alta, dar în principal cuprinde următoarele elemente:

- * modul de adresare.
- * adrese operanți.

Modul de adresare al unui operand definește algoritmul de localizare în memorie a operandului. Există mai multe moduri de adresare:

- * adresare imediată.
- * adresare prin regiștri.
- * adresare prin locații de memorie.

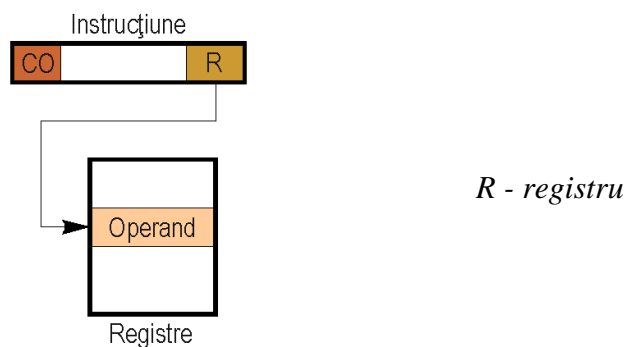
Adresarea este **imediată** atunci când în instrucțiune se specifică valoarea operandului și nu adresa lui.



Adresare imediata

Adresarea prin registre este atunci când în zona de adresă se specifică un registru de memorie. În acest caz este necesar ca, în prealabil, registrul respectiv să fie încărcat cu valoarea dorită. Acest mod de adresare are avantajul unui număr redus de biți (lungime redusă a instrucțiunii), întrucât sunt puțini regiștri în care se pot găsi operanți. Accesul la regiștri este mult mai rapid decât accesul la memorie. Respectă condiția ca un operand să fie preluat din registru în cazul sistemului de calcul cu o singură adresă. Ca dezavantaj, are loc un schimb permanent de date între memorie și regiștri, orice operație între cei doi

operanzi din memorie executându-se prin cel puțin două instrucțiuni-mașină: una de transfer memorie-registru și alta de operație efectivă.

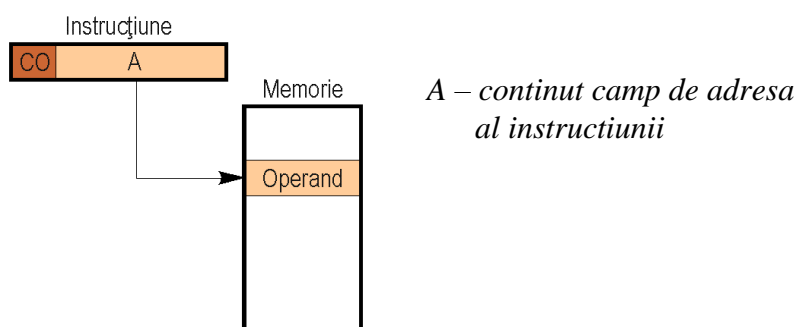


Adresare prin registre

Adresarea prin locații de memorie este atunci când informația folosită pentru calculul adresei operandului se preia din memorie. Acest mod de adresare necesită un algoritm de calcul pentru localizarea operandului. Din punct de vedere al algoritmului de calcul avem:

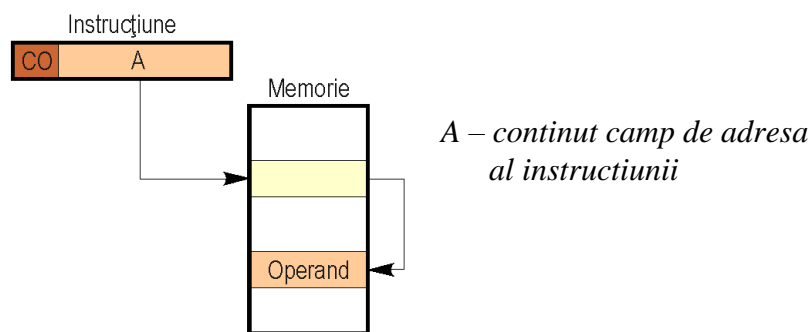
- * adresare directă.
- * adresare indirectă.
- * adresare indexată.

Adresarea directă – informația din instrucțiune reprezintă chiar locația unde se memorează operandul.



Adresare directă

Adresarea indirectă – informația din instrucțiune reprezintă adresa adresei operandului. Această modalitate necesită un ciclu suplimentar de calcul a adresei.



Adresare indirecta

Adresarea indexată – valoarea din instrucțiune se adună algebric cu valoarea unui registru pentru a determina adresa operandului.

$$AE = A + (R)$$

AE - Adresa efectivă a locației sau a registrului care conține operandul la care se face referire

Tipuri de instrucțiuni-mașină. După operația pe care o generează, avem:

- * *instrucțiuni de transfer de date între memorie, regiștri sau stivă.*
- * *instrucțiuni aritmetice.* Orice procesor execută:
 - * *operații elementare* cu numere întregi sau cu numere reale, ori operații complexe, implementate hardware sau software;
 - * *operații logice și de deplasare*, care lucrează la nivel de bit, folosite pentru determinarea biților de control din cuvintele de stare sau pentru realizarea rapidă a unor operații aritmetice (împărțirea la 2, înmulțirea cu 2).
- * *instrucțiuni de comparație și de salt.* Aceste instrucțiuni generează continuarea execuției programului de la o altă adresă decât cea imediat următoare.
- * *instrucțiuni de repetare*, folosite pentru execuția repetată a unei secvențe de instrucțiuni, folosind o variabilă-contor. Această variabilă memorează numărul de cicluri care trebuie efectuate.
- * *instrucțiuni de apel procedură*, în care, pentru executarea unei proceduri se memorează pe stivă adresa de revenire utilizată după execuția procedurii apelate.
- * *instrucțiuni de intrare-ieșire*, folosite pentru transferul de date, cu dispozitivele periferice. Conțin adresa unui dispozitiv sau a unui registru de interfață cu dispozitivul de intrare-ieșire.

După modul de reprezentare a operanzilor, instrucțiunile se împart în:

- * instrucțiuni în virgulă fixă.
- * instrucțiuni în virgulă mobilă.
- * instrucțiuni zecimale.
- * instrucțiuni pe șiruri de caractere.
- * instrucțiuni matriciale.
- * instrucțiuni care operează asupra structurilor de date tabelare.

Orice sistem de calcul cuprinde setul standard de instrucțiuni, adică instrucțiunile în virgulă fixă.

După forma instrucțiunii, structura și lungimea ei, instrucțiunile se împart în:

- * *instrucțiuni cu format fix* - instrucțiunile au aceeași lungime și implică o structură simplă a procesorului, dar o utilizare inefficientă a memoriei;
- * *instrucțiuni cu format variabil*, în care lungimea depinde de tipul operanzilor, de numărul de operanzi și de modul de adresare. Pentru executarea unor astfel de instrucțiuni, procesorul trebuie să determine, la citirea instrucțiunilor din memorie, lungimea instrucțiunii și să citească, în cicluri succesive, toate cuvintele de memorie aferente instrucțiunii.

După modul de adresare al operanzilor, instrucțiunile pot fi:

- * cu nici o adresă - operanzii sunt căutați rapid în stiva sistemului de calcul;
- * cu o singură adresă - codifică de regulă operații unare;
- * cu două adrese - pentru operații binare în care una din adrese este un registru iar cealaltă o adresă de memorie.

Tipuri de procesoare. Procesoarele sunt împărțite de regulă după setul de instrucțiuni:

- * Procesoarele CISC (Complex Instructions Set Computer). Sunt procesoare cu set complet de instrucțiuni, cu format variabil, care permit un număr mare de moduri de adresare. Executarea unei instrucțiuni presupune efectuarea mai multor operații în mai multe cicluri-mașină. Procesoarele rezultate sunt complexe, cu un număr mare de cablaje care realizează implementarea setului de instrucțiuni. Aceste procesoare utilizează eficient memoria internă. Un exemplu este tipul MMX de la procesorul Pentium care are un set extins de instrucțiuni pentru multimedia.

- * **Procesoarele RISC (Reduce Instructions Set Computer).** Sunt procesoare cu set redus de instrucțiuni, care au instrucțiuni elementare, majoritatea putând fi executate într-o singură perioadă de ceas. Instrucțiunile au lungime fixă, folosind un singur acces la memorie. Au un număr minim de moduri de adresare și, în compensație, un număr mare de regiști în care se depun operanzii. Instrucțiunile lucrează cu operanzii de preferat în regiștri, operațiile fiind elementare. Datorită simplității operațiilor elementare, pot fi construite unități de prelucrare paralele rezultând execuția simultană a mai multor instrucțiuni. Cu aceste procesoare se implementează tehnica PIPE-LINE de executare a instrucțiunilor, în care, în fiecare perioadă de ceas se preia câte o instrucțiune. La un moment dat sunt în execuție mai multe instrucțiuni, în diverse stadii. Acest mecanism a fost preluat de INTEL începând cu seria 486.