4. UNITATEA CENTRALĂ DE PRELUCRARE

4.1. Structura UCP

Pentru a examina structura unității centrale de prelucrare (UCP), se prezintă mai întâi funcțiile principale realizate de această unitate:

- *Extragerea instrucțiunilor*: UCP trebuie să citească instrucțiunile din memorie.
- *Interpretarea instrucțiunilor*: Fiecare instrucțiune trebuie decodificată pentru a determina operația care trebuie executată.
- *Citirea datelor*: Pentru execuția unor instrucțiuni este necesară citirea datelor din memorie sau de la un dispozitiv de I/E.
- *Prelucrarea datelor*: Execuția unei instrucțiuni poate necesita execuția unei operații aritmetice sau logice asupra datelor.
- *Scrierea datelor*: Rezultatele execuției trebuie scrise în memorie sau la un dispozitiv de I/E.

Pentru a putea executa aceste funcții, UCP necesită memorarea temporară a unor date într-o memorie internă proprie. De exemplu, trebuie memorată adresa ultimei instrucțiuni, pentru a se cunoaşte adresa de la care se va extrage instrucțiunea următoare. De asemenea, instrucțiunile şi datele trebuie memorate temporar pe durata execuției unei instrucțiuni.

Figura 4.1 prezintă structura generală a UCP, cu indicarea conexiunilor cu restul sistemului prin magistrala sistem.

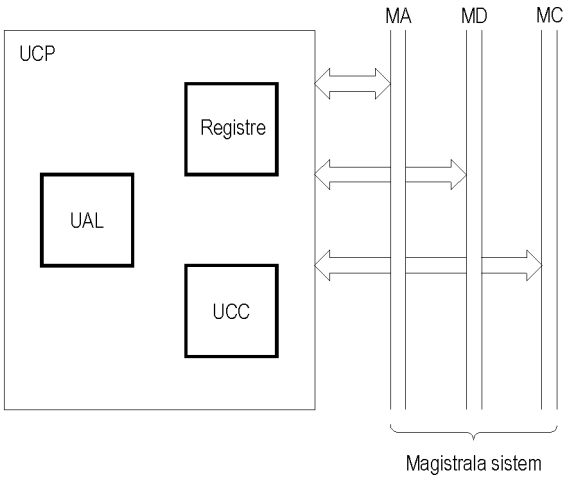

Figura 4.1. Structura generală a UCP împreună cu magistrala sistem.

Figura 4.2 prezintă o imagine mai detaliată a UCP. Se pune în evidență *magistrala internă a* UCP, care este necesară pentru transferul datelor între diferite registre ale UAL, deoarece UAL prelucrează doar datele din memoria internă a UCP. Figura indică de asemenea elemente componente tipice ale UAL.

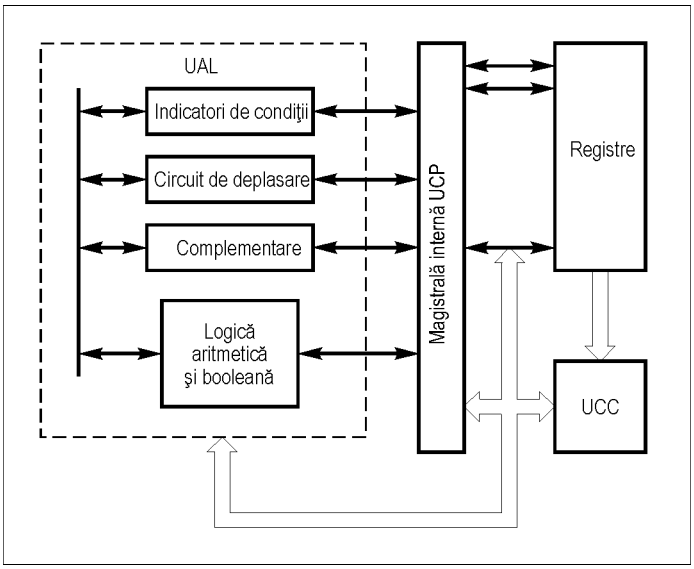

Figura 4.2. Structura mai detaliată a UCP.

4.2. Registre

Registrele din cadrul UCP sunt de două tipuri:

- *Registre utilizator*: Acestea permit reducerea numărului de accesuri la memoria principală, şi deci creşterea vitezei de execuţie a programelor.
- *Registre de control şi de stare*: Acestea se utilizează de unitatea de control pentru controlul funcţionării UCP, şi de programele sistemului de operare pentru controlul execuţiei programelor.

4.2.1. Registre utilizator

Aceste registre sunt accesibile prin instrucţiunile limbajului maşină. Se deosebesc următoarele *categorii* de registre utilizator:

- Registre generale;
- Registre de date;
- Registre de adrese;
- Registre ale indicatorilor de condiţii.

Registrele generale pot fi utilizate în scopuri variate de către programe. În unele cazuri, utilizarea lor în cadrul setului de instrucţiuni este *ortogonală*. Aceasta înseamnă că oricare registru general poate conţine operandul oricărei instrucţiuni. În multe cazuri există însă restricţii, de exemplu pot exista registre dedicate pentru anumite operaţii aritmetice. Uneori, registrele generale se pot utiliza pentru adresarea operanzilor, prin diferite moduri de adresare.

Registrele de date se pot utiliza numai pentru păstrarea datelor şi nu se pot utiliza pentru calculul adresei operanzilor.

Registrele de adrese, utilizate pentru adresarea instrucţiunilor sau a datelor, pot avea o utilizare generală, sau pot fi dedicate unui anumit mod de adresare.

Registrele indicatorilor de condiţii sunt vizibile parţial de către programator. *Indicatorii de condiţii* sunt biţi poziţionaţi de către UCP în funcţie de rezultatele diferitelor operaţii. De exemplu, o operaţie aritmetică poate produce un rezultat pozitiv, negativ, zero, sau poate rezulta o depăşire. Pe lângă rezultatul care se memorează într-un registru sau în memorie, se poziţionează şi unii indicatori de condiţii. Aceşti indicatori pot fi testaţi ulterior, de obicei de instrucţiunile de salt condiţionat.

4.2.2. Registre de control şi de stare

În general, aceste registre nu sunt vizibile programatorilor, sau unele din ele pot fi vizibile pentru instrucţiunile maşină executate într-un mod special al UCP, numit *mod privilegiat*. Diferitele calculatoare au diferite organizări ale registrelor, şi utilizează terminologii diferite. Totuşi, se pot deosebi următoarele registre care sunt esenţiale pentru execuţia instrucţiunilor:

- *Contorul de program* (PC): Conține adresa următoarei instrucțiuni care se va extrage din memorie.
- *Registrul de instrucțiuni* (RI): Conține ultima instrucțiune extrasă din memorie.
- *Registrul de adrese al memoriei* (RA): Conține adresa unei locații de memorie.
- *Registrul de date al memoriei* (RD): Conține cuvântul de date care se va înscrie în memorie sau cuvântul citit recent.

Contorul de program conține adresa instrucțiunii următoare, deoarece de obicei acest registru este actualizat de UCP imediat după fiecare extragere a unei instrucțiuni. O instrucțiune de salt va modifica de asemenea contorul de program. Instrucțiunea extrasă din memorie este depusă în *registrul de instrucțiuni*, unde ea este analizată (decodificată). Transferul datelor cu memoria are loc prin utilizarea *registrului de adrese şi de date al memoriei* (RA, respectiv RD). Registrul RA este conectat la magistrala de adrese, iar RD la magistrala de date.

Registrele menționate sunt utilizate pentru transferul datelor între UCP şi memorie. În cadrul UCP, datele trebuie aplicate la intrările UAL pentru prelucrare. UAL poate avea acces direct la registrul RD şi la registrele generale. O altă variantă este cea în care există registre suplimentare la intrările UAL şi ieşirea acestuia, prin intermediul cărora are loc comunicația cu registrele generale şi registrul de date al memoriei.

UCP conține un registru numit cuvânt de stare al programului (PSW-*Program Status Word*). Acest registru păstrează indicatori de condiții şi alte informații de stare. Câmpurile principale ale acestui registru sunt:

- *Semn*: Conține bitul de semn al rezultatului ultimei operații aritmetice.
- *Zero*: Setat atunci când rezultatul este 0.
- *Transport* (*Carry*): Setat dacă a rezultat un transport sau un împrumut la o operație aritmetică.
- *Egal*: Setat dacă rezultă o egalitate în urma unei operații de comparare.
- *Depăşire* (*Overflow*): Utilizat pentru a indica o depăşire la o operație aritmetică.
- *Validare întreruperi* (*Interrupt Enable/Disable*): Utilizat pentru validarea sau invalidarea întreruperilor.
- *Supervizor* (*Supervisor*): Arată dacă UCP execută un program în mod supervizor sau în mod utilizator. Anumite instrucțiuni privilegiate pot fi executate numai în mod supervizor, iar anumite zone de memorie sunt accesibile numai în acest mod.

Pot exista şi alte registre din această categorie, de exemplu pentru controlul operațiilor de I/E.

4.3. Memoria stivă

O *stivă* este un set ordonat de elemente, dintre care numai unul poate fi accesat la un moment dat. Punctul de acces este numit *vârful stivei*. Un nou element poate fi adăugat numai în vârful stivei, iar un element poate fi şters numai din vârful stivei. De aceea, o stivă se mai numeşte listă de tip LIFO (*Last In, First Out*).

Accesul la vârful stivei se realizează printr-un registru numit *indicator de stivă* (SP - *Stack Pointer*).

Figura 4.3 prezintă principalele operaţii cu stiva.

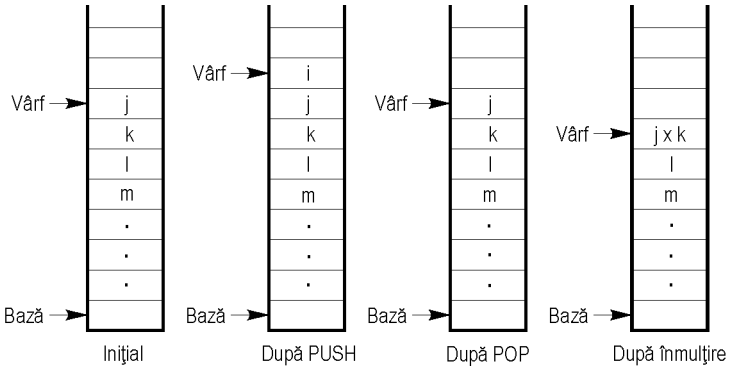

Figura 4.3. Operaţiile principale cu stiva.

O operaţie PUSH adaugă un nou element în vârful stivei. O operaţie POP elimină elementul din vârful stivei. În ambele cazuri, vârful stivei este deplasat în mod corespunzător, prin actualizarea indicatorului de stivă. Anumite *operaţii binare*, care necesită doi operanzi, pot utiliza elementele din vârful stivei ca operanzi. După operaţie, operanzii sunt eliminaţi din stivă, iar rezultatul este depus în vârful stivei. Operaţiile *unare*, care necesită un singur operand, pot utiliza elementul din vârful stivei ca operand.

4.4. Execuţia instrucţiunilor

Funcţia principală a unui calculator este execuţia programelor. UCP realizează această funcţie prin execuţia instrucţiunilor specificate de programul aflat în memorie. UCP citeşte (extrage) câte o instrucţiune din memorie, o execută şi apoi trece la următoarea instrucţiune. Prelucrarea necesară pentru o singură instrucţiune reprezintă un *ciclu de instrucţiune* (Figura 4.4).

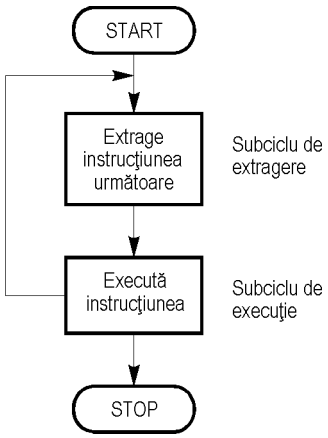

Figura 4.4. Ciclul de instrucțiune sub formă simplificată.

Fiecare ciclu de instrucțiune constă din două părți principale: *subciclul de extragere* şi *subciclul de execuție*. Pe lângă aceste subcicluri, mai poate exista un *subciclu de întrerupere* şi un *subciclu de indirectare*. Procesul de execuție al instrucțiunilor se opreşte numai la oprirea calculatorului, la apariția unei erori sau la întâlnirea unei instrucțiuni de oprire a execuției programului.

4.4.1. Subciclul de extragere

În acest subciclu, se citeşte din memorie o nouă instrucțiune. UCP păstrează evidența ultimei instrucțiuni extrase din memorie şi, în mod obişnuit, extrage următoarea instrucțiune, deci cea de la adresa imediat următoare. Această secvență obişnuită poate fi modificată prin *instrucțiuni de salt*.

Figura 4.5 prezintă transferurile de date din acest subciclu.

Registrul PC conține adresa următoarei instrucțiuni. Adresa este depusă în RA şi este plasată pe magistrala de adrese. Unitatea de comandă solicită o operație de citire a memoriei, iar rezultatul este plasat pe magistrala de date şi copiat în RD, de unde este apoi transferat în RI. În acelaşi timp, PC este incrementat cu 1, fiind pregătit pentru următorul subciclu de extragere.

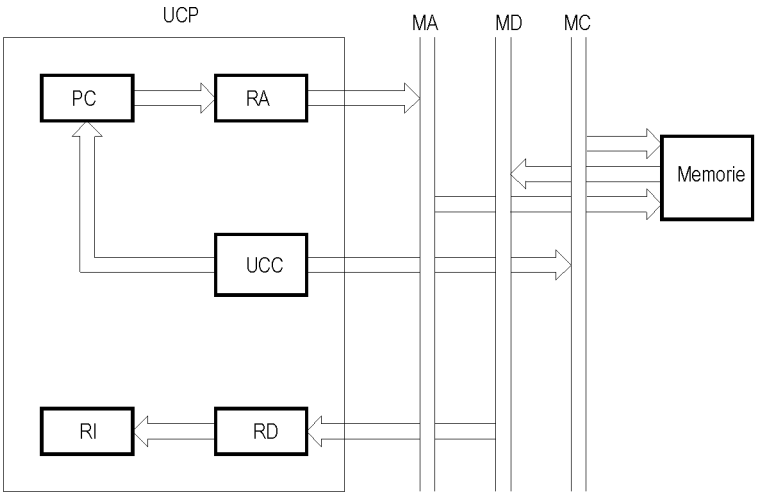

Figura 4.5. Transferuri de date în subciclul de extragere.

4.4.2. Subciclul de execuție

Instrucțiunea citită din memorie este sub forma unui cod binar care este interpretat de UCP, executând acțiunea specificată de acest cod. Această acțiune se poate încadra într-una din următoarele categorii:

- Transfer între UCP şi memorie;
- Transfer între UCP şi dispozitivele de I/E;
- Operație aritmetică sau logică;
- Control, de exemplu modificarea secvenței normale de execuție a instrucțiunilor.

În funcție de tipul instrucțiunii, deci de acțiunea executată, subciclul de execuție diferă.

Se consideră exemplul unui calculator ipotetic, cu structura instrucțiunilor şi a datelor din Figura 4.6. Lungimea instrucțiunilor şi a datelor întregi este de 16 biți. Memoria este organizată pe locații de 16 biți sau cuvinte. Din structura instrucțiunilor rezultă că pot exista $2^4 = 16$ coduri de operații (coduri de instrucțiuni), şi se pot adresa până la $2^{12} = 4096$ cuvinte de memorie.

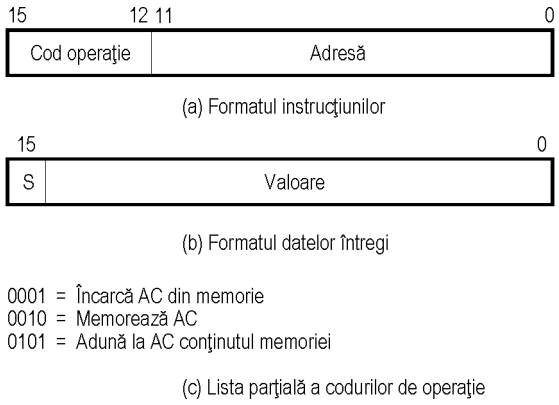

Figura 4.6. Structura instrucțiunilor şi a datelor pentru un calculator ipotetic.

Se consideră un fragment de program pentru adunarea conținutului a două cuvinte succesive de memorie, cu adresele `9B0h` şi `9B1h`, şi memorarea rezultatului în locul cuvântului al doilea. Fragmentul de program se află în memorie, începând de la adresa `100h`. Se notează cu PC contorul de program, cu AC registrul acumulator, şi cu RI registrul de instrucțiuni.

În Figura 4.7 se prezintă execuția acestui fragment de program, indicând conținutul relevant al memoriei şi al registrelor UCP. Notația utilizată este cea hexazecimală.

Etapele de execuție ale acestui fragment de program sunt următoarele:

1. Contorul de program PC conține valoarea `100h`, adresa primei instrucțiuni. Această instrucțiune se depune în registrul de instrucțiuni RI.
2. Cei 4 biți mai semnificativi din RI specifică încărcarea registrului AC. Ceilalți biți specifică adresa locației de memorie `9B0h`.
3. Contorul de program PC este incrementat, şi se extrage următoarea instrucțiune.
4. Se adună vechiul conținut al AC cu conținutul locației `9B0h`, şi rezultatul se depune în AC.
5. Contorul de program PC este incrementat, şi se extrage următoarea instrucțiune.
6. Conținutul AC este memorat la adresa `9B1h`.

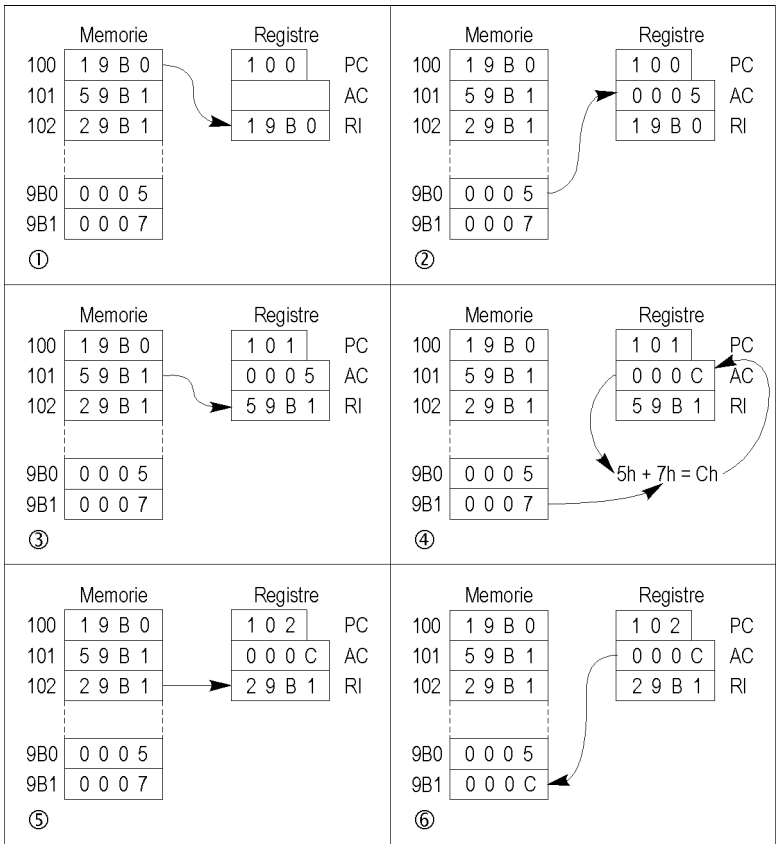

Figura 4.7. Execuția unui fragment de program.

În acest exemplu, sunt necesare trei cicluri de instrucțiune, fiecare constând dintr-un subciclu de extragere şi unul de execuție. În cazul unui set de instrucțiuni mai complex, sunt necesare mai puține cicluri. De exemplu, pot exista instrucțiuni care conțin mai multe adrese, care adună conținutul a două locații de memorie şi memorează rezultatul într-una din aceste locații. Astfel, subciclul de execuție pentru o instrucțiune poate necesita mai multe referiri la memorie. De asemenea, în locul unei referiri la memorie, o instrucțiune poate specifica o operație de I/E.

Cu aceste considerații, structura mai detaliată a unui ciclu de instrucțiuni este prezentată în Figura 4.8, sub forma unei diagrame de stare. Pentru anumite instrucțiuni, unele stări nu vor fi executate, iar pentru altele vor fi executate de mai multe ori.

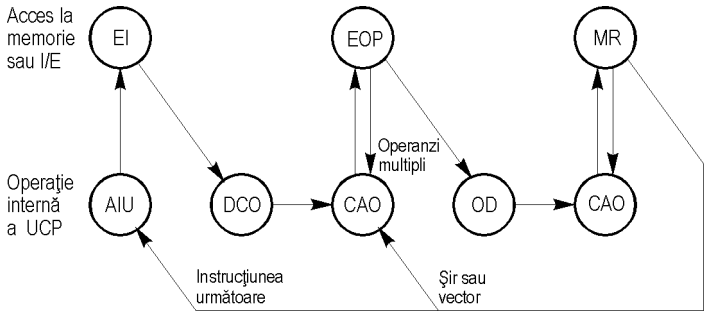

Figura 4.8. Diagrama de stări a unui ciclu de instrucțiuni.

Stările pot fi descrise astfel:

- *Determinarea adresei instrucțiunii următoare* (*AIU*): cel mai frecvent, aceasta implică incrementarea contorului de program PC.
- *Extragerea instrucțiunii* (*EI*): se citeşte instrucțiunea din memorie în RI.
- *Decodificarea codului instrucțiunii* (*DCO*): se analizează instrucțiunea pentru a determina tipul operației de executat şi operanzii care trebuie utilizați.
- *Calculul adresei operandului* (*CAO*): Dacă operația implică utilizarea unui operand din memorie sau care se poate obține printr-o operație de I/E, se determină adresa acestui operand.
- *Extragerea operandului* (*EOP*): se citeşte operandul din memorie sau de la un dispozitiv de I/E.
- *Execuția operației cu datele* (*OD*): se execută operația indicată.
- *Memorarea rezultatului* (*MR*): se înscrie rezultatul în memorie sau se transmite la un dispozitiv de I/E.

Stările din partea superioară a diagramei implică un transfer între UCP şi memorie, sau UCP şi un dispozitiv de I/E. Stările din partea inferioară implică numai operații interne UCP. Starea *CAO* apare de două ori, deoarece o instrucțiune poate implica o operație de citire, de scriere, sau ambele.

La anumite calculatoare, o singură instrucțiune poate specifica o operație asupra unui vector de numere (tablou unidimensional) sau un şir de caractere. Aceasta se reflectă în diagramă.

4.4.3. Subciclul de întrerupere

Calculatoarele dispun de un mecanism prin care diferite module pot întrerupe operațiile executate de UCP. Întreruperile sunt prevăzute în primul rând pentru a creşte

eficiența UCP. De exemplu, cele mai multe periferice sunt lente comparativ cu UCP, şi aceasta trebuie să rămână inactivă până la terminarea unei operații de transfer. Prin existența întreruperilor, UCP poate executa alte instrucțiuni în timpul unei operații de I/E. Atunci când un periferic este pregătit pentru a accepta alte date de la UCP, sau pentru a transmite date la UCP, modulul de I/E al perifericului transmite o *cerere de întrerupere* către UCP. UCP răspunde prin suspendarea execuției programului curent, executând un *program de servire* (de tratare) *a întreruperii*, după care va relua execuția programului.

Pentru recunoaşterea întreruperilor, se adaugă ciclului de instrucțiune un *subciclu de întrerupere* (Figura 4.9). În acest subciclu, UCP testează dacă a apărut o cerere de întrerupere.

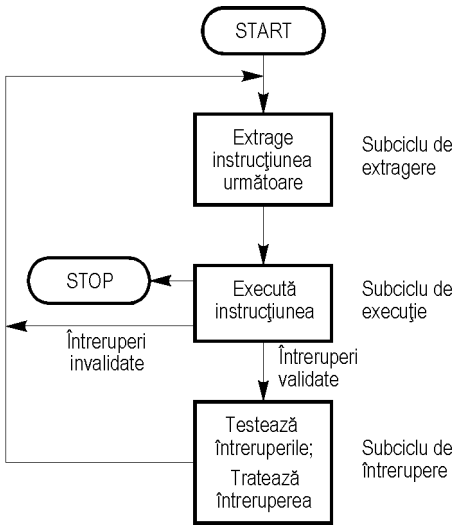

Figura 4.9. Ciclul de instrucțiune completat cu subciclul de întrerupere.

Dacă semnalul de cerere a întreruperii nu este activat, UCP continuă cu subciclul de extragere. Dacă a apărut o întrerupere, UCP execută următoarele:

1. Salvează contextul programului curent. Aceasta înseamnă salvarea adresei următoarei instrucțiuni de executat şi a altor date din registre, care sunt semnificative pentru programul respectiv.
2. Încarcă în contorul de program adresa programului de servire a întreruperii.

UCP va executa în continuare instrucțiunile din rutina de servire, după care va reveni la programul întrerupt.

UCP poate dezactiva (invalida) toate sau o parte a semnalelor de întrerupere. O *întrerupere dezactivată* înseamnă că UCP va ignora cererea respectivă de întrerupere. De exemplu, în general este de dorit să se termine servirea unei întreruperi înaintea ac-

ceptării alteia. De aceea, întreruperile sunt de obicei dezactivate în timpul servirii unei întreruperi. Dacă apar întreruperi în acest timp, ele vor fi servite după validarea întreruperilor.

Transferurile de date din subciclul de întrerupere se prezintă în Figura 4.10.

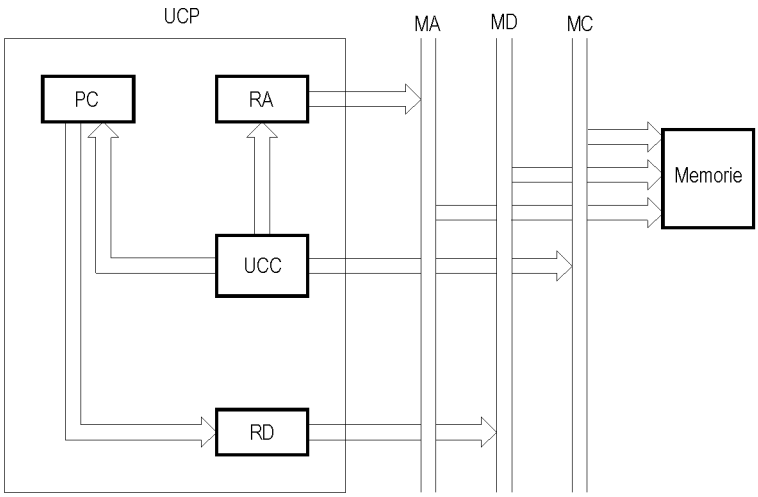

Figura 4.10. Transferuri de date în subciclul de întrerupere.

Conținutul curent al PC trebuie salvat astfel încât UCP să poată relua programul întrerupt. Adresa unei locații speciale de memorie rezervată pentru memorarea PC este depusă în RA de unitatea de comandă. PC poate fi salvat în memoria stivă, caz în care în RA se transferă indicatorul de stivă. Conținutul PC este transferat în RD pentru a fi înscris în memorie. Adresa rutinei de tratare a întreruperii se transferă apoi în PC. Ca urmare, următorul ciclu de instrucțiune va începe prin extragerea primei instrucțiuni din rutina de tratare.

4.4.4. Operații de I/E

Perifericele sunt conectate la calculator prin intermediul unor *module de I/E*. Aceste module pot transfera date direct cu UCP. Aşa cum UCP poate iniția o operație de citire sau scriere cu memoria, indicând adresa unei locații de memorie, UCP poate citi sau scrie date de la sau la un modul de I/E. În ultimul caz, UCP identifică un anumit periferic controlat de un modul de I/E.

În multe cazuri, este mai eficientă realizarea transferurilor direct între periferic şi memorie. În aceste cazuri, UCP acordă unui modul de I/E permisiunea de citire din sau scriere în memorie, astfel că transferurile pot avea loc fără intervenției UCP. Această operație se numeşte *acces direct la memorie* (DMA).