

Figura 7.10. Structura generală a unei unități de comandă și control care utilizează un decodificator.

Semnalul C_5 va fi activat în a doua unitate de timp a subciclului de extragere și a celui de indirectare, deci se poate defini prin ecuația:

$$C_5 = \overline{P} \cdot \overline{Q} \cdot T_2 + \overline{P} \cdot Q \cdot T_2$$

Acest semnal trebuie însă activat și în subciclul de execuție. Pentru acest exemplu, se presupune că există numai trei instrucțiuni care execută citirea memoriei: LDA, ADD și AND. Se notează semnalele activate la ieșirea decodificatorului cu numele instrucțiunii respective. Atunci, C_5 se poate defini ca:

$$C_5 = \overline{P} \cdot \overline{Q} \cdot T_2 + \overline{P} \cdot Q \cdot T_2 + P \cdot \overline{Q} \cdot (LDA + ADD + AND) \cdot T_2$$

În același mod se scriu ecuațiile tuturor semnalelor de comandă. Rezultatul va fi un set de ecuații booleene care definește funcționarea UCC.

UCC trebuie să controleze de asemenea starea ciclului de instrucțiune. La sfârșitul fiecărui subciclu, trebuie să se actualizeze semnalele P și Q pentru a indica următorul subciclu care se va executa.

7.4.3. Unități de comandă microprogramate

7.4.3.1. Principiul UCC microprogramate

Semnalele de comandă pot fi reunite sub forma unei succesiuni de cifre binare, formând un cuvânt numit *cuvânt de comandă*. Fiecare micro-operație se caracterizează printr-un cuvânt specific de comandă, iar succesiunea cuvintelor de comandă prin care se indică secvența corectă a micro-operațiilor pentru fiecare operație poate fi memorată într-o *memorie de comandă*.

O unitate de comandă în care succesiunea cuvintelor de comandă, adică a valorilor semnalelor de comandă, este memorată, este numită *unitate de comandă microprogramată*. Fiecare cuvânt de comandă memorat în memoria de comandă formează o *microinstrucțiune*, iar secvența de microinstrucțiuni formează un *microprogram*.

Deosebirea principală între o unitate de comandă microprogramată și una cablată constă în modul în care circuitul trece dintr-o stare în alta pentru a genera semnalele de comandă.

- În cazul cablat, o stare corespunde unei *faze*, caracterizată prin activarea unui semnal de fază. Într-o fază sunt generate anumite semnale de comandă necesare execuției unei funcții.
- În cazul microprogramat, o stare corespunde unei *microinstrucțiuni*, care codifică micro-operațiile care trebuie executate în timpul aceluiași semnal de ceas.

O unitate de comandă microprogramată are două funcții principale:

- *Funcția de control propriu-zis*, prin care se definesc micro-operațiile care trebuie executate. Această definiție cuprinde de obicei selecția operanzilor, a operației de executat, selecția destinației rezultatului etc.
- *Funcția de secvențiere*, prin care se definește adresa microinstrucțiunii următoare. Această definiție se referă la identificarea sursei pentru adresa următoare, la controlul condițiilor de test sau la generarea directă a valorii adresei.

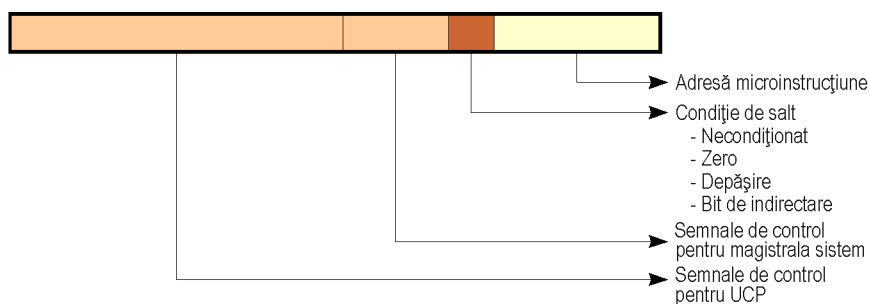


Figura 7.11. Format tipic de microinstrucțiune orizontală.

Conform acestor funcții, o microinstrucțiune este formată din următoarele câmpuri principale:

- Câmpul semnalelor de comandă generate pentru controlul circuitelor;
- Câmpul de adresă, care conține adresa următoarei microinstrucțiuni care se va executa, dacă o anumită condiție este adevărată (de exemplu, dacă bitul de indirectare din codul instrucțiunii este 1). Dacă această condiție este falsă, se va executa următoarea microinstrucțiune din memoria de comandă.

- Câmpul de condiții, care indică următoarea microinstrucțiune care va fi executată.

Acest tip de microinstrucțiune se numește *microinstrucțiune orizontală* (Figura 7.11). Pe lângă acestea, există și *microinstrucțiuni verticale*.

Există câte un bit pentru fiecare semnal de comandă intern al UCP și un bit pentru fiecare semnal de control al magistralei sistem. Câmpul de condiție arată condiția în care se execută un salt în microprogram, iar câmpul de adresă conține adresa la care se efectuează saltul. O asemenea microinstrucțiune este interpretată în modul descris mai jos.

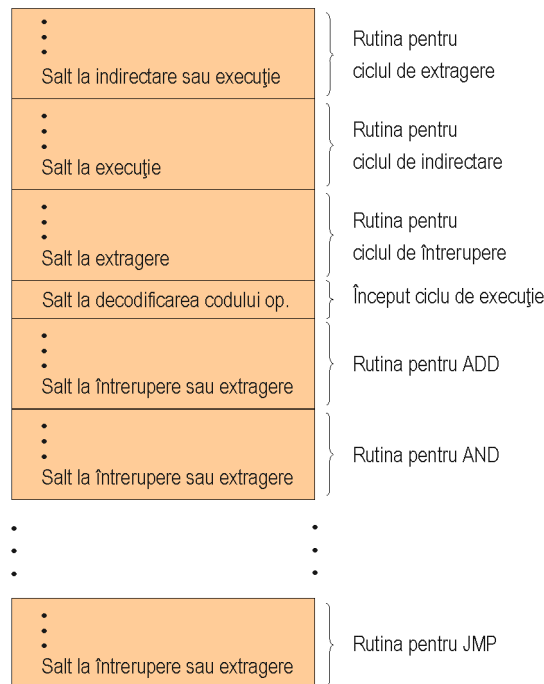


Figura 7.12. Amplasarea microinstrucțiunilor în memoria de comandă.

1. Pentru execuția microinstrucțiunii, se activează toate semnalele de comandă cărora le corespunde un bit de 1 în câmpul semnalelor de comandă, și se dezactivează cele cărora le corespunde un bit de 0. Semnalele de comandă care vor fi activate vor determina execuția uneia sau a mai multor micro-operații.
2. În cazul în care condiția indicată de câmpul de condiție este falsă, se execută următoarea microinstrucțiune din microprogram.
3. În cazul în care condiția indicată de câmpul de condiție este adevărată, se execută microinstrucțiunea indicată de câmpul de adresă.

În Figura 7.12 se prezintă amplasarea microinstrucțiunilor în memoria de comandă.

Microinstrucțiunile din fiecare rutină sunt executate secvențial. Fiecare rutină se termină cu o instrucțiune de salt care indică următoarea rutină care va fi executată. Există o rutină specială pentru începutul unui subciclu de execuție, care indică rutina care va fi executată pentru diferitele instrucțiuni (ADD, AND, ..., JMP), în funcție de codul instrucțiunii.

Memoria de comandă reprezintă o descriere completă a funcționării unității de comandă, deoarece definește secvența de micro-operații care trebuie executate în timpul fiecărui subciclu (de extragere, de indirectare, de execuție, de întrerupere), și specifică secvențierea acestor subcicluri.

7.4.3.2. Structura unei UCC microprogramate

În Figura 7.13 se prezintă elementele principale ale unei unități de comandă microprogramate.

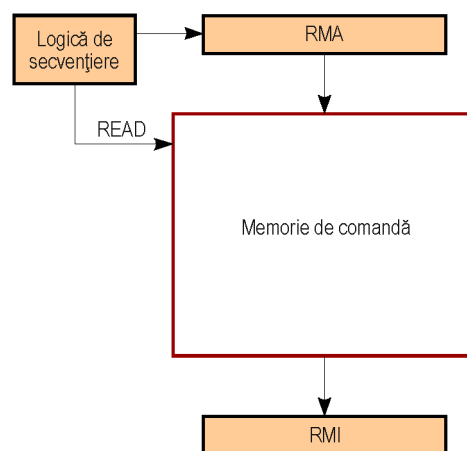


Figura 7.13. Elementele principale ale unei unități de comandă microprogramate.

Registrul de microadrese (RMA) conține adresa următoarei microinstrucțiuni care se va citi din memoria de comandă. După citire, microinstrucțiunea este transferată în *registrul de microinstrucțiuni (RMI)*. Prin acest registru se activează semnalele de comandă. Citirea unei microinstrucțiuni din memoria de comandă este echivalentă cu execuția microinstrucțiunii respective. *Logica de secvențiere* încarcă registrul de microinstrucțiuni și activează comanda de citire.

Structura mai detaliată a unității de comandă este prezentată în Figura 7.14.

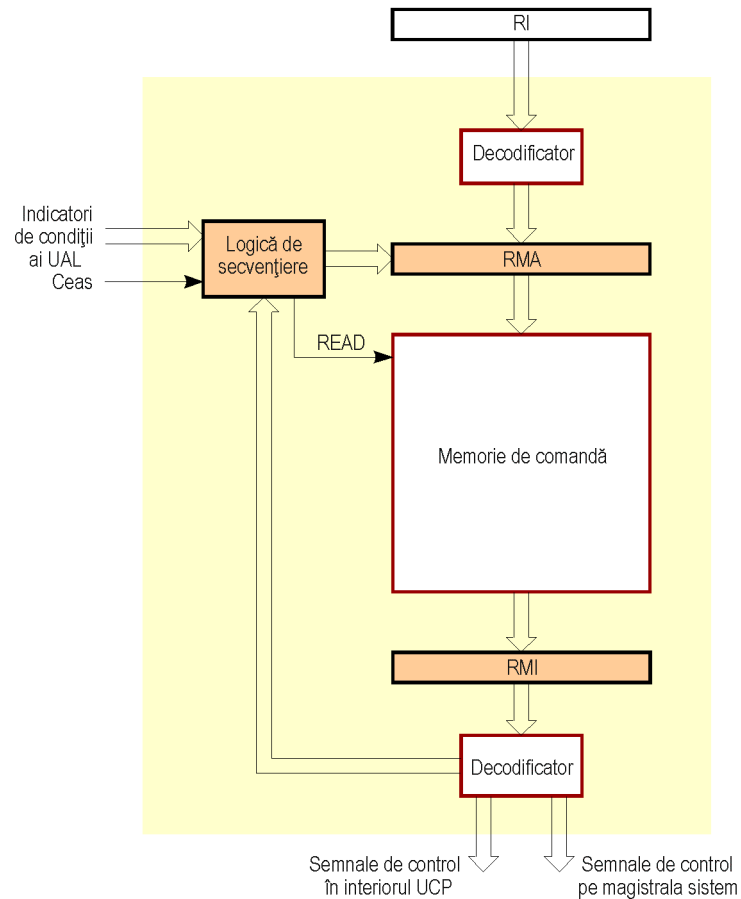


Figura 7.14. Structura detaliată a unei unități de comandă microprogramate.

Comparativ cu o unitate de comandă cablată, unitatea microprogramată are aceleași intrări (RI, indicatori de condiții ai UAL, ceas) și ieșiri (semnale de comandă). Unitatea de comandă funcționează astfel:

1. Pentru execuția unei instrucțiuni, logica de secvențiere activează un semnal de citire a memoriei de comandă.
2. Cuvântul a cărei adresă se specifică în registrul de microadrese este citit în registrul de microinstrucțiuni.
3. Conținutul registrului de microinstrucțiuni activează semnalele de comandă și generează informații despre adresa următoare pentru logica de secvențiere.

4. Logica de secvențiere încarcă o nouă adresă în registrul de microadrese, pe baza informațiilor despre adresa următoare de la registrul de microinstrucțiuni și a indicatorilor de condiții ai UAL.

În Figura 7.14 există două decodificatoare. Primul decodificator translatează codul operației din registrul de instrucțiuni RI într-o adresă a memoriei de comandă. Al doilea decodificator nu este utilizat pentru microinstrucțiunile orizontale, ci pentru cele verticale. Într-o microinstrucțiune orizontală, fiecărui bit al câmpului de comandă îi corespunde un semnal de comandă. Într-o microinstrucțiune verticală, se utilizează un cod pentru fiecare operație care trebuie executată, de exemplu $RA \leftarrow PC$, iar decodificatorul translatează acest cod în semnale de comandă individuale.

Avantajul microinstrucțiunilor verticale este că ele sunt mai compacte, utilizând un număr mai mic de biți, cu prețul unei logici suplimentare.

Avantajul principal al utilizării microprogramării pentru implementarea unităților de comandă este simplificarea proiectării acestora. O unitate de comandă cablată trebuie să conțină o logică complexă pentru secvențierea micro-operațiilor din ciclul de instrucțiune. În schimb, decodificatoarele și logica de secvențiere dintr-o unitate de comandă microprogramată sunt simple.

Dezavantajul principal al unităților microprogramate este că sunt mai lente decât cele cablate realizate într-o tehnologie comparabilă. Cu toate acestea, microprogramarea este tehnica cea mai utilizată pentru implementarea unităților de comandă.

7.4.3.3. Secvențierea microinstrucțiunilor

Pentru alegerea unei tehnici de secvențiere, trebuie să se țină cont de două aspecte: dimensiunea microinstrucțiunii și timpul în care se poate genera adresa. Primul aspect este legat de costul memoriei de comandă, iar al doilea de viteza de execuție a microinstrucțiunilor.

La execuția unui microprogram, adresa următoarei microinstrucțiuni poate fi:

- Determinată de registrul de instrucțiuni;
- Adresa imediat următoare;
- Destinația unui salt.

Adresa următoarei microinstrucțiuni este generată pe baza microinstrucțiunii curente, a indicatorilor de condiții și a conținutului registrului de instrucțiuni. Se pot utiliza diferite tehnici, acestea fiind clasificate pe baza formatului informației de adresă din cadrul microinstrucțiunii:

- Cu două câmpuri de adresă;
- Cu un singur câmp de adresă;
- Format variabil.

Soluția cea mai simplă este cea *cu două câmpuri de adresă* în fiecare microinstrucțiune (Figura 7.15).

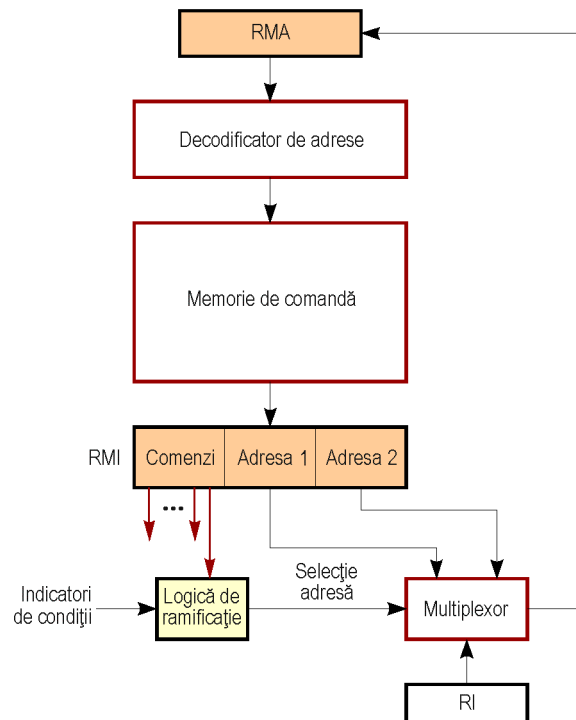


Figura 7.15. Logică de secvențiere cu două câmpuri de adresă.

Există un multiplexor ale cărei intrări sunt cele două câmpuri de adresă și registrul de instrucțiuni. Pe baza unei intrări de selecție a adresei, multiplexorul transmite fie codul operației din RI, fie una din cele două adrese la registrul de microadrese (RMA). Conținutul registrului RMA este decodificat pentru a obține adresa microinstrucțiunii următoare. Semnalele de selecție a adresei sunt generate de o logică de ramificație ale cărei intrări sunt indicatorii de condiții și biții din partea de comandă a microinstrucțiunii.

Deși soluția cu două adrese este simplă, necesită o lungime mai mare a microinstrucțiunii decât alte tehnici. Cu o logică suplimentară, se poate reduce lungimea microinstrucțiunii. O soluție este de a utiliza *un singur câmp de adresă* (Figura 7.16). Adresa următoare poate fi dată de:

- Câmpul de adresă;
- Codul din registrul de instrucțiuni;
- Adresa imediat următoare.

Semnalele de selecție a adresei determină opțiunea selectată. Prin această metodă se reduce numărul câmpurilor de adresă la unul singur, dar de multe ori câmpul de adresă nu va fi utilizat.

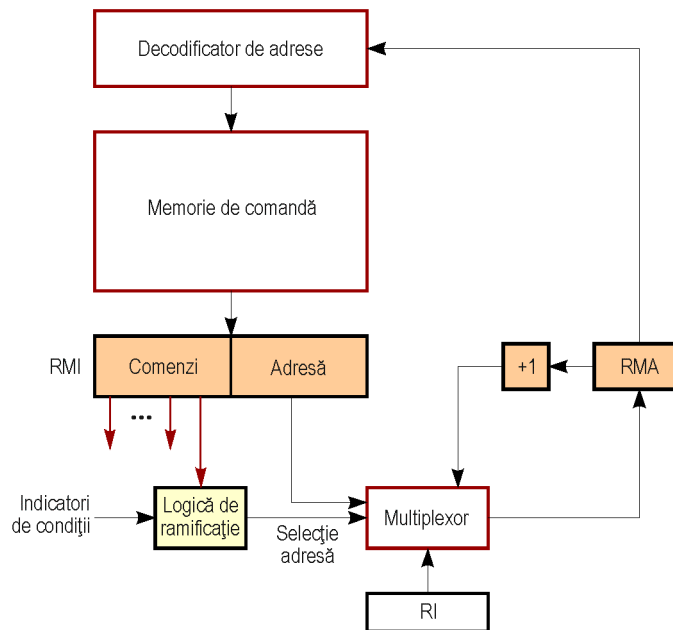


Figura 7.16. Logică de secvențiere cu un singur câmp de adresă.

O altă soluție este de a prevedea *două formate diferite* ale microinstrucțiunii, un bit indicând formatul care va fi utilizat (Figura 7.17).

În cazul primului format, toți biții sunt utilizați pentru activarea unor semnale de comandă. Următoarea adresă este fie adresa imediat următoare, fie o adresă determinată din registrul de instrucțiuni. În cazul formatului al doilea, o parte a biților sunt utilizați pentru controlul logicii de ramificație, iar ceilalți biți conțin adresa. Acest format specifică un salt condiționat sau necondiționat.

Un dezavantaj al acestei metode este că se consumă un ciclu întreg pentru fiecare microinstrucțiune de salt. În cazul celorlalte metode, generarea adresei are loc ca o parte a aceluiași ciclu în care se generează semnalele de control, reducându-se astfel accesurile la memoria de control.

Există mai multe tehnici pentru generarea sau calculul adresei următoare. Acestea se pot împărți în:

- Tehnici *explicite*, pentru care adresa este disponibilă în mod explicit în cadrul microinstrucțiunii;
- Tehnici *implicite*, care necesită o logică suplimentară pentru generarea adresei.

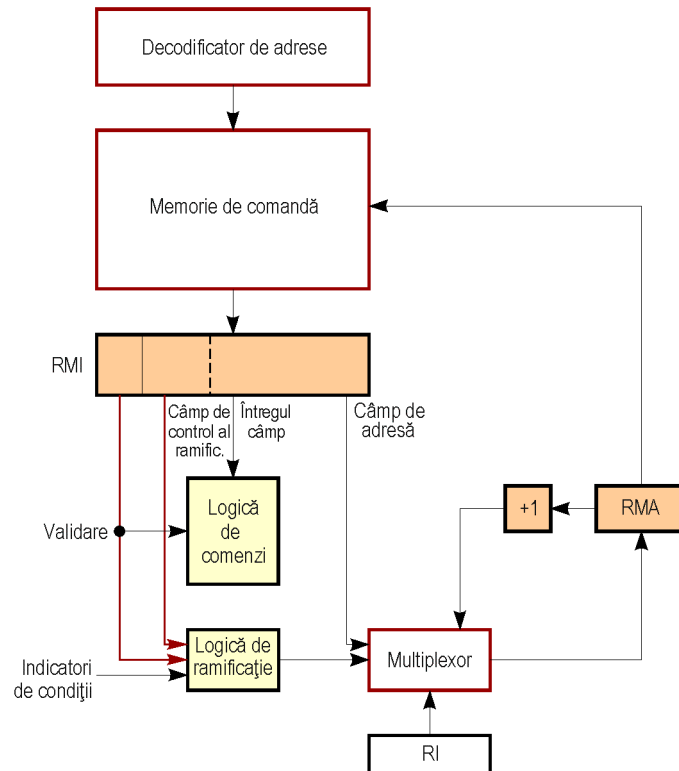


Figura 7.17. Logică de secvențiere cu două formate de microinstrucțiuni.

În Tabelul 7.2 se prezintă diferitele tehnici pentru generarea adresei microinstrucțiunii următoare.

Tabelul 7.2. Tehnici pentru generarea adresei microinstrucțiunii următoare.

Tehnici explicite	Tehnici implicite
Două câmpuri	Translatare
Salt necondiționat	Adunare
Salt condiționat	Control rezidual

În cazul microinstrucțiunilor cu două adrese, sunt disponibile două adrese în cadrul fiecărei microinstrucțiuni. Dacă se utilizează un format cu o singură adresă sau un format variabil, se pot implementa diferite instrucțiuni de salt. O instrucțiune de salt condiționat depinde de următoarele tipuri de informații:

- Indicatorii de condiții ai UAL;

-
- O parte a câmpului de cod al operației sau al modului de adresare din cadrul instrucțiunii;
 - O parte a unui registru, ca de exemplu bitul de semn;
 - Biții de stare ai unității de comandă.

Prin *translatare*, partea de cod a operației din cadrul instrucțiunii trebuie transformată într-o adresă a unei microinstrucțiuni. O altă metodă este de a *combina sau aduna* două porțiuni ale unei adrese pentru a forma adresa completă. *Controlul rezidual* implică utilizarea adresei microinstrucțiuni care a fost salvată în prealabil într-un registru al unității de comandă. De exemplu, în cadrul microprogramului pot exista subrutine. Se utilizează un registru intern sau o stivă de registre pentru a păstra adresele de revenire.