

9. Sisteme microprogramate exemplu de implementare

9.1 Proiectarea unității de comandă microprogramată a calculatorului didactic

Proiectarea unității de comandă microprogramată a calculatorului didactic implică rezolvarea următoarelor aspecte:

- specificarea caracteristicilor unității de comandă microprogramate pentru calculatorul didactic ;
- specificarea detaliată a structurii calculatorului didactic;
- stabilirea operațiilor elementare pentru controlul resurselor calculatorului didactic;
- stabilirea variabilelor de stare (condiții de test);
- stabilirea formatului microinstrucțiunilor;
- proiectarea microsecvențiatorului;
- descrierea microprogramului.

9.1.1 Specificarea caracteristicilor unității de comanda microprogramate pentru calculatorul didactic

Aspectele principale care trebuie considerate în vederea specificării unității de comandă microprogramată au fost prezentate în paragraful 9.1. Analizând aceste aspecte se constată că există o mare varietate de soluții și opțiuni pentru specificarea unității de comandă.

Având în vedere structura calculatorului didactic se poate proiecta unitatea sa de comandă sub forma microprogramată cu următoarele atribute generale:

- din punctul de vedere al relației pozitionale între MC și memoria principală se va alege o memorie de control separată de memoria principală;
- implementarea memoriei de control se va face cu memorii de tip PROM;
- memoria de control se va organiza astfel încât să se specifice o singură microinstrucțiune într-un cuvânt de control;
- organizarea logică a microinstrucțiunilor va fi de tip minimal (microinstrucțiunea împărțită în câmpuri);
- implementarea microinstrucțiunilor va fi de tip serie-paralelă;
- se va considera o implementare monofază a microinstrucțiunilor.

Având în vedere caracteristicile prezentate, structura generală a unității de comandă microprogramată pentru calculatorul didactic se prezintă în Fig. 9.22.

unde:

MC - memoria de control, păstrează microprogramul care implementează fazele de citire interpretare și execuție ale instrucțiunilor mașină. Este de tip PROM. Un cuvânt de control din MC, echivalent cu o microinstrucțiune, generează toate semnalele de comandă (microoperațiile) ce sunt active la un moment dat pe durata unei perioade de sincronizare.

μRI - registrul de microinstrucțiuni, păstrează microinstrucțiunea curentă ce se execută.

μCP - registrul de adrese al MC, păstrează adresa microinstrucțiunii următoare care se va citi.

μS - microsecvențiatorul, reprezintă unitatea de comandă convențională care asigură citirea, interpretarea și validarea acțiunilor microinstrucțiunilor. Pe baza tipului microinstrucțiunii curente,

a stării primitivelor funcționale ale calculatorului didactic precum și a codului operație a instrucțiunii mașină curente (ce interpretează și se execută), μS formează adresa μI următoare și validează (activează spre resursele unității de execuție) microoperațiile din μI curentă.

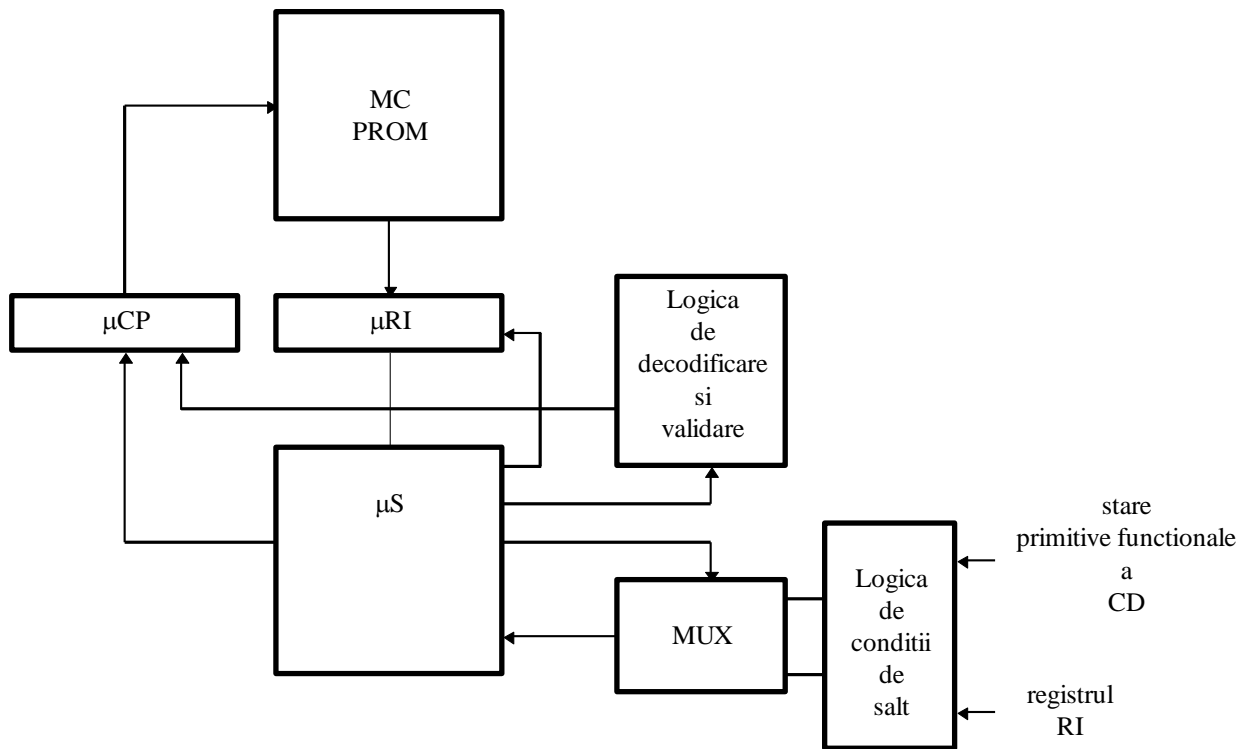
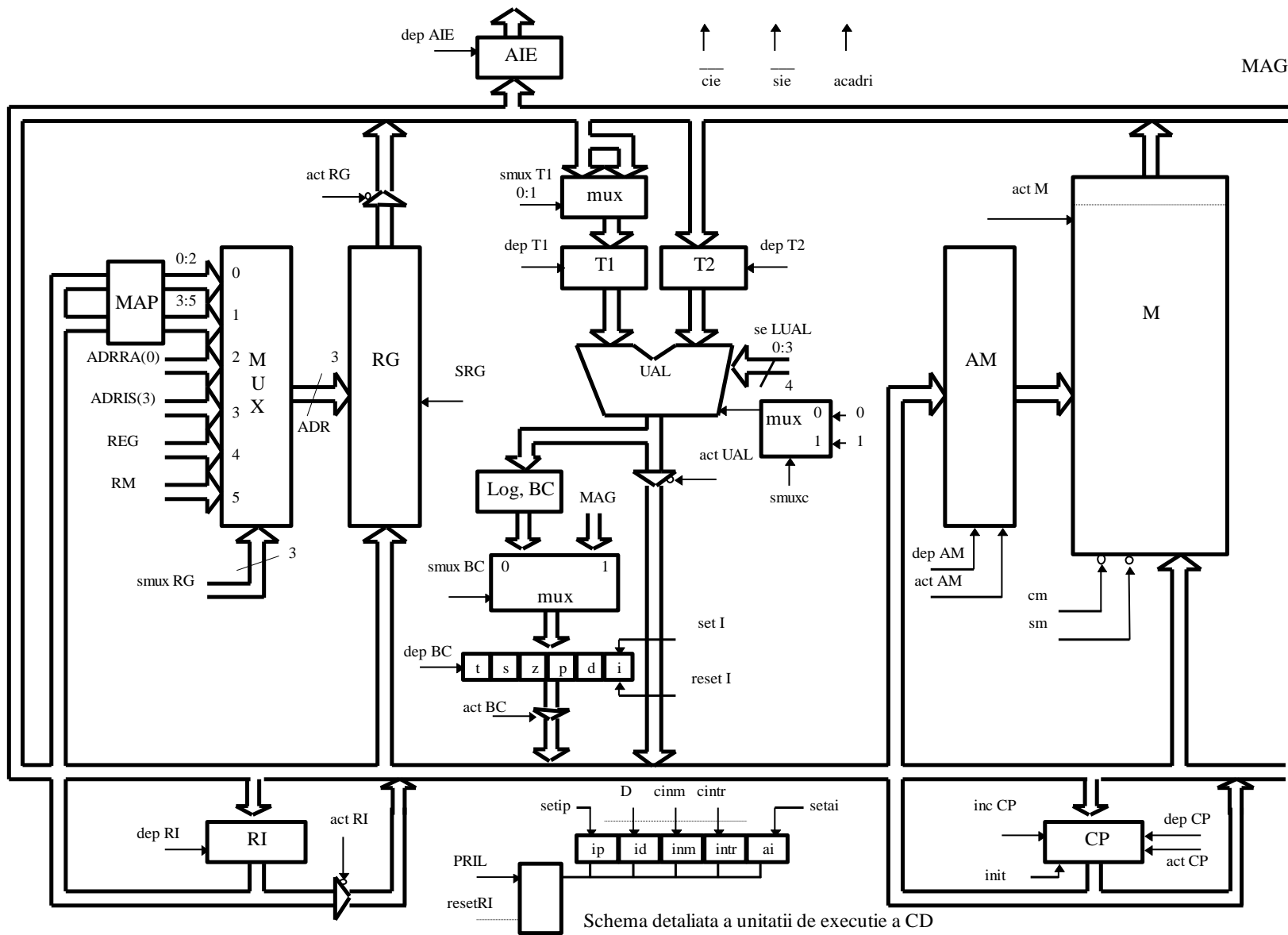


Figura 9.1

9.1.2 Specificarea detaliată a structurii calculatorului didactic

Pentru a stabili formatul microinstrucțiunilor și a proiecta microsecvențiatorul este necesar să detaliem structura părții operaționale (unității de execuție) a calculatorului didactic.

În Fig.9.23 se prezintă structura detaliată a unității de execuție a calculatorului didactic, fără a include subsistemul de intrări/ieșiri.



Schema detaliata a unitatii de executie a CD

Figura 9.2

Este necesar să precizăm că toate resursele care au acces la magistrala MAG, trebuie să fie prevăzute cu ieșiri cu trei stări. Acest deziderat se poate realiza în două moduri :

- alegem resurse cu ieșiri cu trei stări (dacă există) ;
- alegem resurse care nu au ieșiri cu trei stări însă intercalăm între ieșirile lor și magistrală circuite separatoare cu trei stări ("buffere tri state").

În schema detaliată nu se precizează tipul componentelor cu care se face implementarea ci se stabilesc numai resursele și operațiile elementare asupra lor. Se pun în evidență funcțiile logice pe care trebuie să le efectueze primitivele funcționale.

De exemplu, registrul CP indiferent de componenta cu care se implementează, trebuie să fie controlat de semnale (microoperații) care specifică :

- inițializarea (ștergerea)- initl ;
- incrementarea - incCP ;
- încărcarea cu o valoare specificată de conținutul magistralei MAG -depCP ;
- activarea pe magistrala MAG - actCP .

9.1.3 Stabilirea operațiilor elementare pentru controlul resurselor calculatorului didactic

Analizând schema detaliată a unității de execuție a calculatorului didactic, unde s-a precizat modul de utilizare a primitivelor funcționale rezultă lista semnalelor de comandă:

- 1.actAM - activează registrul de adrese al memoriei pe liniile de adrese spre memorie (s-a considerat că trebuie să fie dezactivat când intervine DMA-ul);
- 2.depAM - depune (încarcă) conținutul magistralei MAG în AM;
- 3.cm - comandă operația de citire din memorie;
- 4.sm - comandă operația de scriere în memorie;
- 5.actAM - activează conținutul celulei de memorie adresate pe magistrala MAG;
- 6.incCP - incrementează CP ;
- 7.depCP - depune în CP conținutul magistralei MAG ;
- 8.actCP - activează conținutul registrului CP pe magistrala MAG ;
- 9.depRI - depune în RI conținutul magistralei MAG ;
- 10.actRI - activează conținutul registrului RI pe magistrala MAG;
- 11.smuxRG_{0,2} - selectează adresa memoriei locale RG, specificând astfel registrul general implicat în operația curentă ;
- 12.sRG - comandă scrierea în memoria locală RG ;
- 13.actRG - activează conținutul memoriei locale RG pe magistrala MAG (conținutul registrului general selectat de smuxRG) ;
- 14.smuxT1_{0,1} - selectează informația care se înscrie în registrul T1;
- 15.depT1 - depune date (încarcă) în registrul T1 ;
- 16.depT2 - depune date (încarcă) în registrul T2 ;
- 17.selUAL_{0,3} - specifică operația ce trebuie efectuată de UAL ;
- 18.actUAL - activează rezultatul din UAL pe magistrala MAG ;
- 19.smuxC - selectează informația pentru transportul de intrare în UAL
- 20.smuxIND - selectează informația ce se înscrie în indicatorii de condiții ;
- 21.depIND - depune (încarcă) datele selectate de multiplexorul smuxIND în indicatorii de condiții IND (t,s,z,p,d,I) ;
- 22.actIND - activează pe magistrala MAG conținutul indicatorilor de condiții IND ;
- 23.depAIE -depune (încarcă) în registrul AIE informația ce se găsește pe magistrala MAG (adresa portului de intrare/ieșire implicat în transfer);

- 24.cie - activează pe magistrala MAG conținutul portului de intrare (citește port de intrare) ;
- 25.sie - comandă scrierea într-un port de ieșire ;
- 26.setI - înscrie "1" (setează) în indicatorul I ;
- 27.resetI - înscrie "0" (resetează) în indicatorul I;
- 28.setip - înscrie "1" (setează) în bistabilul ip de cerere de întrerupere prin program ;
- 29.setai - înscrie "1" în bistabilul de achitare întrerupere ai
- 30.resetbi - înscrie "0" (resetează) în bistabilul de cerere de întrerupere corespunzător cererii luate în considerare și în bistabilul de achitare întrerupere (ai) ;
- 31.actADRI - activează pe magistrala MAG adresa celulei capcană corespunzătoare nivelului de întrerupere ce urmează să fie tratat ;

9.1.4 Stabilirea variabilelor de stare (condițiilor de test)

Pentru a asigura ramificațiile în cadrul microprogramului, microsecvențiatorul (μ S) testează starea primitivelor funcționale (caracterizată prin valoarea indicatorilor de condiții) și biți din registrul de instrucțiuni (cod operație, mod de adresare).

Lista variabilelor de stare (semnalelor testate) în fazele de citire interpretare și execuție a instrucțiunilor mașină rezultă din analiza programului AHPL descris în capitolul 5 și este prezentată în cele ce urmează .

În lista semnalelor testate se precizează semnalul sau combinația de semnale analizate precum și pasul AHPL în care are loc testul.

Lista semnalelor testate (sau a funcțiilor logice testate) în programul ce descrie fazele de citire interpretare și execuție a instrucțiunilor mașină a calculatorului didactic se prezintă în tabelul 9.1.

În lista semnalelor testate s-au pus în clar toate condițiile care au apărut în programul AHPL. Această listă se poate reduce din mai multe puncte de vedere și anume:

- să se specifice numai semnalele directe iar decizia de testare pe "0" logic sau pe "1" logic să fie luată prin microprogram;
- grupurile de semnale specificate prin funcții logice pot fi testate individual, însă în mai multe microinstrucțiuni de salt ceea ce conduce la scăderea performanțelor unității de comandă microprogramată.

Tabelul 9.1

Nume semnal sau funcție logică testată	pas AHPL
1.SL(SYN(starta))	1
2.id v inm	2
3.inm \vee id \wedge (intr \wedge I)	2
4.RI ₀	4
5.RI ₈ \wedge RI ₉	5,37,39,54
6.RI ₈ \wedge RI ₉ \wedge RI ₁₃ \wedge RI ₁₄	5
7.RI ₉ \wedge RI ₁₃	7
8.RI ₉ \wedge RI ₁₃	7
9.RI ₉	10
10.RI ₈	11
11.RI ₁₅	22
12.RI ₁	26
13.RI ₃	27,60,66
14.RI ₄	28
15.RI ₅	28
16.RI ₂	29,57,60
17.RI ₇	30,59
18.RI ₇ \wedge RI ₈ \wedge RI ₉	30,59
19.RI ₇ \wedge RI ₈ \wedge RI ₉	30,59
20.RI ₇	31,58
21.RI ₇ \wedge RI ₈ \wedge RI ₉	31,58
22.RI ₇ \wedge RI ₈ \wedge RI ₉	31,58
23.RI ₈ \wedge RI ₉	37,39,54
24.RI ₆	38,53,69,72
25.RI _{4,6}	55,60
26.RI ₂	60
27.RI ₄ \wedge RI ₅	67
28.RI ₄ \wedge RI ₅	67
29.RI ₄ \wedge RI ₅	67
30.RI ₆	81
31.indicatori condiții + funcții de indicatori condiții	85
32.salt necondiționat "1"	15,16,18,32,37,42,47,53,56,65,70,71,75,78,83,88

9.1.5 Stabilirea formatului microinstrucțiunilor

Pentru unitatea de comandă microprogramată a calculatorului didactic vom defini două tipuri de microinstrucțiuni și anume:

- microinstrucțiune operațională, care specifică controlul primitivelor funcționale ale unității de execuție ;
- microinstrucțiune de ramificație, care permite testarea stării primitivelor funcționale și asigură ramificația în microprogram .

Pentru a putea stabili structura microinstrucțiunilor trebuie analizate următoarele aspecte:

- lista semnalelor care controlează primitivele funcționale ale unității de execuție ;
- programul AHPL care descrie secvența de comandă a fazelor de citire interpretare și execuție a instrucțiunilor mașină pentru a stabili setul de microoperații elementare ce trebuie să se desfășoare în paralel ;
- lista semnalelor (funcțiilor logice de semnale) care reprezintă condițiile de test în cadrul algoritmului ce se implementează în microprogram .

Listele de semnale care controlează primitivele funcționale respectiv condițiile de test au fost prezentate anterior.

Din analiza programului AHPL care descrie fazele de citire interpretare și execuție a instrucțiunilor mașină pentru calculatorul didactic (prezentat în capitolul 5) rezultă seturile de operații elementare ce trebuie să se desfășoare în paralel. În cele ce urmează se prezintă seturile de operații elementare ce se desfășoară în paralel și pașii AHPL când sunt ele active (lista este alcătuită numai pentru faza de citire interpretare pas 1:25).

Lista seturilor de microoperații elementare paralele :

Nr. crt.	Seturi de microoperații paralele	pas AHPL
1.	setai	2.1
2.	actCP;depAM	2.2;13;20
3.	actAM;cm;actM;depRI	3
4.	smuxRG=0;actRG;smuxT1=0;depT1	6
5.	smuxRG=1;actRG;depT2	8
6.	selUAL=5;actUAL;smuxT1=0;depT1	9;15
7.	incCP	12;19
8.	actAM;cm;actM;depT2	14
9.	selUAL=0BH;actUAL;smuxRG=1;sRG	16
10.	smuxRG=1;actRG;depT2	17
11.	selUAL=0CH;actUAL;smuxRG=1;sRG;depT2	18
12.	actAM;cm;actM;smuxT1=0;depT1	21;24
13.	selUAL=4;actUAL;depAM	23;25
14.	etc pentru pașii	25...

Formatul microinstrucțiunilor sunt prezentate în Fig 9.24, cele operaționale Fig.9.24 a) iar cele de ramificație în Fig.9.24 b).

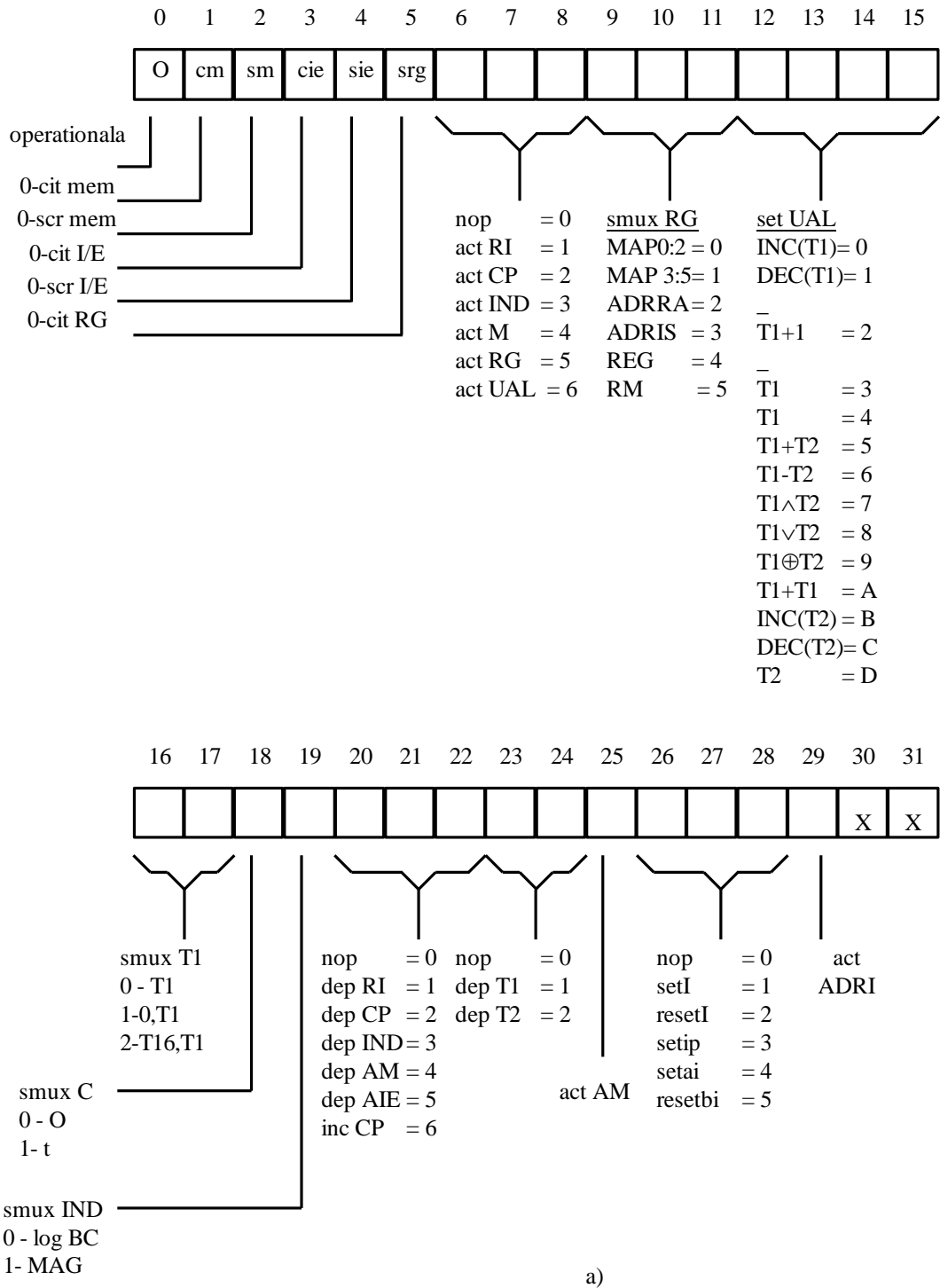
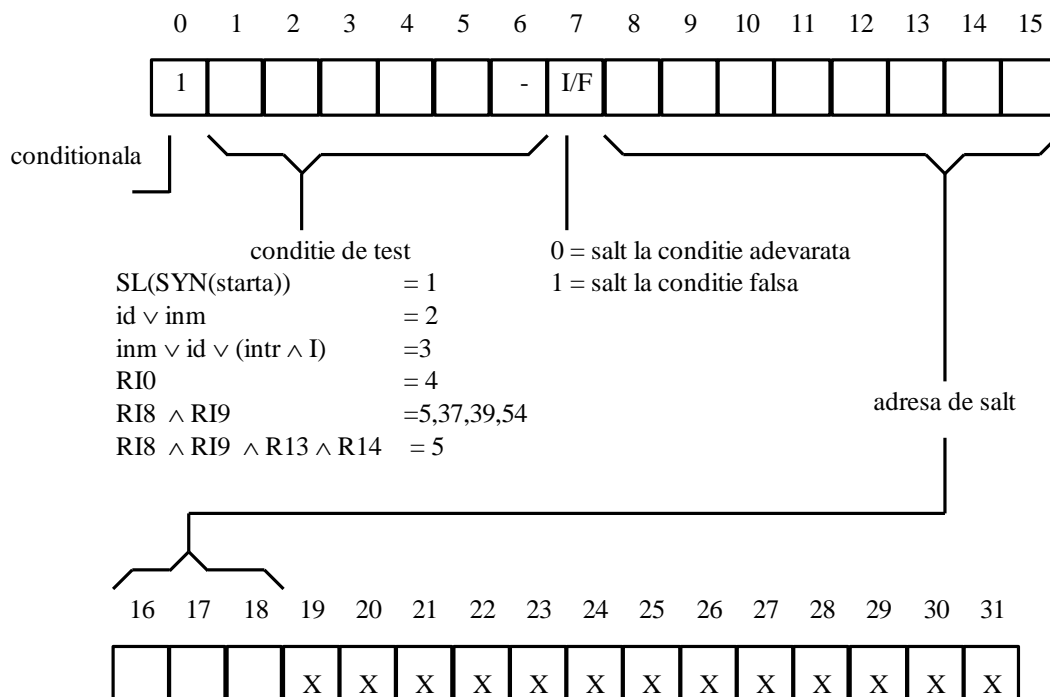


Figura 9.3 a)
Structura microinstrucțiunilor operaționale



b)

Fig 9.2 b) Structura microinstrucțiunilor condiționale

În proiectarea microinstrucțiunilor s-a ținut seama că microoperațiile paralele prezentate anterior să facă parte din câmpuri distincte și de faptul că anumite microoperații se exclud reciproc (cum ar fi activarea unor resurse pe magistrala MAG).

De asemenea s-a considerat, pentru a mări flexibilitatea în elaborarea microprogramelor, că decizia în funcție de testarea unei condiții să poată fi luată la valoare adevărată sau falsă. Pentru aceasta s-a prevăzut bitul 7 în microinstrucțiunea de ramificație care face un "sau exclusiv " cu condiția de test.

Lungimea microprogramului s-a considerat de maximum 2048 de locații, rezultând astfel un număr de 11 biți în formatul microinstrucțiunii de ramificație pentru adresa de salt (biții 8..18).

Observație: Având în vedere faptul că în general memoriile de tip PROM sunt organizate pe 4 sau 8 biți, desi lungimea exactă a microinstrucțiunii este de 30 de biți, s-a preferat să se lucreze cu o lungime de 32 de biți.

9.1.6 Proiectarea microsecvențiatorului

Având în vedere structura microinstrucțiunilor se poate descrie funcționarea microsecvențiatorului (μS) care asigură citirea interpretarea și execuția microinstrucțiunilor (activarea microoperațiilor spre unitatea de execuție a CD).

Programul AHPL de descriere a unității de comandă microprogramată a calculatoului didactic este :

MODULE: Unitatea de comandă microprogramată

MEMORY: MC [2048;32]; mRI [32]; mCP [11]

INPUTS: lista condițiilor de test

OUTPUTS: lista semnalelor de comandă

1. $\mu RI \leftarrow \text{BUSFN}(\text{MC}; \text{DCD}(\mu \text{CP}))$

/* se citește microinstrucțiunea curentă în registrul de microinstrucțiuni μRI */

2. $\rightarrow \mu RI_0 / (4)$

/* se separă microinstrucțiunile de ramificație la pas 4 și se continuă cu cele operaționale */

3. $\text{cm}, \text{sm}, \text{cie}, \text{sie}, \text{sRG} = \mu RI_{1:5}$;

$\text{actRI}, \text{actCP}, \text{actIND}, \text{actM}, \text{actRG}, \text{actUAL} = \text{DCD}_{1:6}(\mu RI_{6:8})$;

$\text{smuxRG}_{0:2} = \mu RI_{9:11}$;

$\text{selUAL}_{0:3} = \mu RI_{12:15}$;

$\text{smuxT1}_{0:1} = \mu RI_{16:17}$;

$\text{smuxC} = \mu RI_{18}$;

$\text{smuxIND} = \mu RI_{19}$;

$\text{depRI}, \text{depCP}, \text{depIND}, \text{depAM}, \text{depAIE}, \text{incCP} = \text{DCD}_{1:6}(\mu RI_{20:22})$;

$\text{depT1}, \text{depT2} = \text{DCD}_{1:2}(\mu RI_{23:24})$;

$\text{actAM} = \mu RI_{25}$;

$\text{setI}, \text{resetI}, \text{setip}, \text{setai}, \text{resetbi} = \text{DCD}_{1:5}(\mu RI_{26:28})$;

$\text{actADRI} = \mu RI_{29}$;

$\mu \text{CP} \leftarrow \text{INC}(\mu \text{CP})$;

$\rightarrow (1)$

/* se activează câmpurile de microoperații spre unitatea de execuție, se incrementează contorul de adrese microinstrucțiuni și se revine la pas 1 pentru citire microinstrucțiunii următoare

4. $\mu \text{CP} - (\text{INC}(\mu \text{CP}) ! \mu RI_{8:18}) * (g, g)$

$\rightarrow (1)$

/* se formează adresa microinstrucțiunii următoare care poate fi adresa curentă incrementată (dacă condiția de salt nu este îndeplinită) sau adresa de salt din microinstrucțiune (dacă condiția de salt este îndeplinită) după care se merge la citirea microinstrucțiunii următoare */

END SEQUENCE

S-a considerat că funcția g reprezintă condiția de test și are următoarea formă :

$$g = \mu RI_7 \oplus (SL(SYN(starta))! id \vee inm! inm \vee id (intr^I)! RI_0! RI_8^RI_9! \dots RI_6! COND! 1)*DCD_{0:31}(\mu RI_{1:6})$$

Semnalul de test COND reprezintă ieșirea unui multiplexor care are ca intrări indicatorii de condiții sau funcții logice dintre aceștia conform descrierii din tabelul 5.5, iar ca selecție biții RI_{4:7} din codul operație al instrucțiunii de salt condiționat.

9.1.6.1 Descrierea microprogramului

Descrierea microprogramului care implementează unitatea de comandă microprogramată a calculatorului didactic se poate face utilizând un limbaj de microasamblare. Pentru aprofundarea relației între cuvântul de control și semnalele de comandă ce controlează primitivele funcționale ale unității de execuție se exemplifică descrierea unei secțiuni de microprogram specificând direct biții din microinstrucțiune.

1. -> SL(SYN(starta)) / (1)

2. -> (inm v id, inm v id v (intr ^ I)) / (91,2.2)

2.1 ai <- 1 ;

-> (91)

2.2 AM <- CP

Trebuie remarcat faptul că stabilindu-se microinstrucțiuni de tip operațional și de tip condițional, pașii AHPL care specifică acțiune și salt vor fi implementați în două microinstrucțiuni distincte iar pașii AHPL care specifică salturi multiple se implementează prin mai multe microinstrucțiuni se ramificație.

Imaginea memoriei de control pentru secvența prezentată anterior este următoarea:

```
pas 00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 15 16 17 18 19 20 21 22
23 24 25 26 27 28 29 30 31
```

```
1.  1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 x x
2.  1 0 0 0 0 0 1 0 ..adresa pasului 91 ... x x x x x x x x x x x x x x
2.  1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 1 x x x x x x x x x x x x x x
2.1 0 1 1 1 1 1 0 0 0 x x x x x x x x x x x x x 0 0 0 0 0 0 1 0 0 0 x x
2.1 1 0 1 1 1 1 1 0 ..adresa pasului 91 ... x x x x x x x x x x x x x x
2.2 0 1 1 1 1 1 0 1 0 x x x x x x x x x x x x x 1 0 0 0 0 0 0 0 0 0 x x
```

lvarea se face într-un timp polinomial.