PROIECTAREA SISTEMELOR NUMERICE
UNITATE DE COMANDĂ MICROPROGRAMATĂ

Dispozitiv de împărțire binară. Unitate de comandă UC microprogramată.

Realizarea microprogramată a unității de comandă a unui sistem numeric se bazează pe utilizarea unei memorii PROM. Cuvintele acestei memorii reprezintă fiecare o microinstrucțiune a programului. Ele sunt compuse în principal din cuvinte de test, de adresă şi de comandă.

Se defineşte o metodă de sinteză pentru UC a sistemelor numerice, după care se aplică diferitele etape ale acestei metode la sinteza UC a sistemului numeric care realizează împărțirea a 2 numere binare.

1) Metoda

O UC microprogramată este de fapt o unitate de tratare a adreselor din memoria de microprogram. Metoda de sinteză a UC se aseamănă cu cea a UE în privința realizării şi se încheie printr-o etapă de programare.

Realizarea microprogramată a UC a unui sistem numeric care realizează un algoritm dat este caracterizată de repertoriul său de microinstrucțiuni (microinstrucțiuni de test şi comandă, microinstrucțiuni de test, microinstrucțiuni de comandă, microinstrucțiuni de apel de subprogram, microinstrucțiuni de retur dintr-un subprogram etc.). Ea se efectuează plecând de la organigrama originară prin parcurgerea următoarelor etape:

1. Adaptarea eventuală a organigramei la repertoriul de microinstrucțiuni al UC alese;
2. Definirea variabilelor de comandă ale UE;
3. Redactarea programului originar şi determinarea formatului şi a câmpurilor microinstrucțiunilor;
4. Declararea regiştrilor şi resurselor UC şi descrierea funcțională a acesteia cu ajutorul unei organigrame;
5. Construirea schemei UC şi declararea eventuală a regiştrilor şi resurselor adiționale;
6. Realizarea UC cu ajutorul unor componente combinaționale şi secvențiale disponibile pe piață;
7. Adaptarea programului originar şi programarea memoriei.

2) UC cu 2 microinstrucțiuni cu registru de adrese

2.1. Caiet de sarcini

Repertoriul de microinstrucțiuni constituie caietul de sarcini al UC. El permite determinarea formatului şi a câmpurilor microinstrucțiunilor specificând totodată asocierea lor cu elementele organigramei.

Repertoriul propus în fig. 4 asociază o microinstrucțiune de test binar fiecărui romb şi o microinstrucțiune de comandă fiecărui dreptunghi. El se aplică direct organigramei originare a sistemului numeric care realizează împărțirea a 2 numere binare (fig. 2 – unitate de execuție).

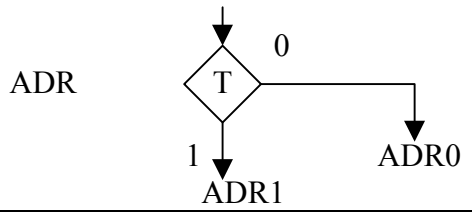
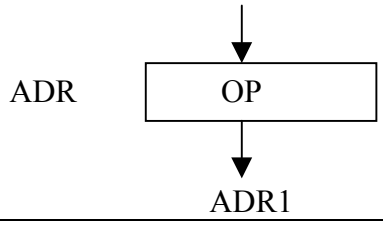
IF T = 0 THEN GOTO ADR0 ELSE GOTO ADR1	
DO OP AND GOTO ADR1	

Figura 4.

2.2. Variabilele de comandă a UE

Fiecare dreptunghi al organigramei originare (fig. 2 – unitate de execuție) conține o operație a UE. Ținând cont de resursele şi de regiştrii aleşi pentru realizarea acestei unități de execuție, se pune problema definirii variabilelor sale de comandă, pentru ca ea să execute ansamblul operațiilor organigramei. Valorile care trebuie să li se atribuie sunt precizate în tabela operațiilor UE (fig. 5). Ele rezultă din tabelele operațiilor proprii resurselor şi regiştrilor utilizați.

Operație	Descriere
OP0	NOP
OP1	$F \leftarrow 1$
OP2	$A \leftarrow 0$; $B \leftarrow X$; $C \leftarrow Y$; $F \leftarrow 0$; $I \leftarrow 0$
OP3	$(A,B) \leftarrow (A,B) * 2$; $B_0 \leftarrow 0$
OP4	$B_0 \leftarrow 1$
OP5	$A \leftarrow A + C$
OP6	$A \leftarrow A - C$
OP7	$A \leftarrow A + C$; $I \leftarrow I + 1$
OP8	$A \leftarrow A - C$; $I \leftarrow I + 1$

Figura 5.

Op	A_A	SH_A	LD_A	G_B	S_B	SH_B	LD_B	WS_C	J_F	K_F	CLR_I	LD_I	P_I	A_{AU}
OP0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
OP1	∅	0	0	0	∅	0	0	0	1	0	0	0	0	∅
OP2	0	0	1	0	0	0	1	1	0	1	1	∅	∅	∅
OP3	∅	1	0	∅	∅	1	∅	0	0	0	0	0	0	∅
OP4	∅	0	0	0	1	0	1	0	0	0	0	0	0	∅
OP5	1	0	1	0	∅	0	0	0	0	0	0	0	0	0
OP6	1	0	1	0	∅	0	0	0	0	0	0	0	0	1
OP7	1	0	1	0	∅	0	0	0	0	0	0	0	1	0
OP8	1	0	1	0	∅	0	0	0	0	0	0	0	1	1

Figura 6.

Fiecare romb al organigramei corespunde unei instrucțiuni de test. În cursul execuției unei asemenea instrucțiuni, UE trebuie să rămână inactivă. Ea efectuează

atunci o instrucțiune neutră NOP. Pentru a simplifica mai mult concepția UC, admitem în plus că toate variabilele sale de ieşire (adică toate variabilele de comandă ale UE) sunt egale cu 0 în timpul unei instrucțiuni de test. Tabela operațiilor UE (fig. 5 şi 6) conține deci în mod obligatoriu operația NOP pe lângă celelalte operații deduse din organigramă.

Determinarea ansamblului minimal al variabilelor distincte C care trebuie să fie generate de UC se face cu ajutorul unui graf de compatibilități.

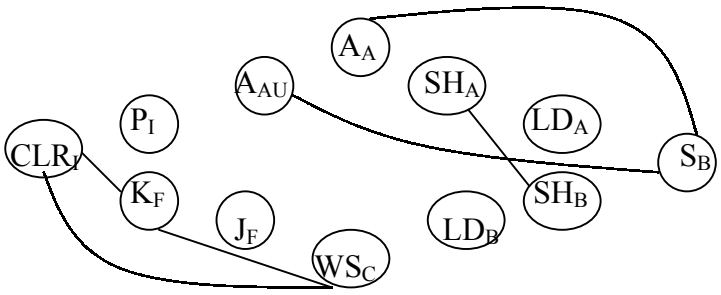


Figura 7.

Procedăm după cum urmează:

a. Căutăm variabilele de comandă care pot rămâne constante. Atribuirea unor valori particulare condițiilor indiferente din tabela operațiilor UE (fig. 5 şi 6) reduce 2 variabile la starea de constante:

$G_B = 0$

$LD_I = 0$ $\qquad \overline{LD_I} = 1$

b. Fiecare variabilă care rămâne va constitui un nod în graf.

c. ***Compatibilitatea*** a 2 variabile (faptul că au valori egale pentru fiecare operație în care sunt ambele specificate) e indicată printr-un arc neorientat care uneşte cele 2 noduri.

d. Un ***poligon complet*** = un ansamblu de noduri care sunt toate conectate 2 câte 2.

e. Ansamblul minimal al poligoanelor complete corespunde ansamblului minimal al variabilelor distincte C. Cele 2 ansambluri minimale de poligoane complete din graful de compatibilități din fig. 7 determină 2 ansambluri minimale de 8 variabile distincte.

$C0 = A_{AU}$	$C0 = S_B = A_{AU}$
$C1 = P_I$	$C1 = P_I$
$C2 = J_F$	$C2 = J_F$
$C3 = WS_C = K_F = CLR_I$	$C3 = WS_C = K_F = CLR_I$
$C4 = LD_B$	$C4 = LD_B$
$C5 = LD_A$	$C5 = LD_A$
$C6 = SH_A = SH_B$	$C6 = SH_A = SH_B$
$C7 = A_A = S_B$	$C7 = A_A$

Alegem prima soluție. În tabelul de mai jos sunt date valorile hexazecimale corespunzătoare cazului când toate condițiile indiferente sunt alese 0.

Operație	C7:0
OP0	00
OP1	04
OP2	38
OP3	40
OP4	90
OP5	A0
OP6	A1
OP7	A2
OP8	A3

Figura 8.

2.3. Programul originar şi definirea microinstrucțiunilor

Redactarea acestui program se face transcriind organigrama originară a UC (fig. 2 – unitate de execuție) cu ajutorul repertoriului de microinstrucțiuni. Programul este cel ce va fi înscris în PROM.

Fiecare din etichetele NEXT specifică adresa microinstrucțiunii care ocupă linia următoare.

```
WAIT:     DO OP1 AND GOTO NEXT
          IF E = 0 THEN GOTO WAIT
                   ELSE GOTO EXECUTE
EXECUTE:  DO OP2 AND GOTO NEXT
          DO OP3 AND GOTO NEXT
          DO OP6 AND GOTO LOOP
LOOP:     IF A7 = 0 THEN GOTO SUBC
                    ELSE GOTO ADDC
ADDC:     DO OP3 AND GOTO NEXT
          DO OP7 AND GOTO TEST
TEST:     IF I7 = 0 THEN GOTO LOOP
                   ELSE GOTO NEXT
          IF A7 = 0 THEN GOTO INCB
                   ELSE GOTO NEXT
          DO OP5 AND GOTO WAIT
SUBC:     DO OP4 AND GOTO NEXT
          DO OP3 AND GOTO NEXT
          DO OP8 AND GOTO TEST
INCB:     DO OP4 AND GOTO WAIT
```

Formatul şi câmpurile celor 2 microinstrucțiuni din fig. 9 este determinat de:

- numărul microinstrucțiunilor din repertoriu;
- numărul variabilelor de test ale organigramei;
- ansamblul minimal de variabile de comandă;
- dimensiunea programului originar.

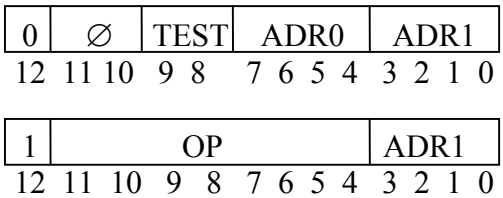


Figura 9. Microinstrucțiunile

Microinstrucțiunea de test binar este definită de un bit de cod MC = 0, un câmp TEST pentru selecția variabilei de test conținută în romb şi 2 câmpuri ADR0 şi ADR1 care dau, respectiv, adresa instrucțiunii următoare pentru valoarea 0 şi pentru valoarea 1 a variabilei de test.

Microinstrucțiunea de comandă este definită de un bit de cod MC = 1, un câmp OP care dă starea variabilelor de comandă a UE pentru efectuarea operației conținute în dreptunghi şi un câmp ADR1 care dă adresa microinstrucțiunii următoare.

2.4. Declararea şi descrierea funcțională

Pentru a executa cele 2 microinstrucțiuni din fig. 9, UC face apel la:

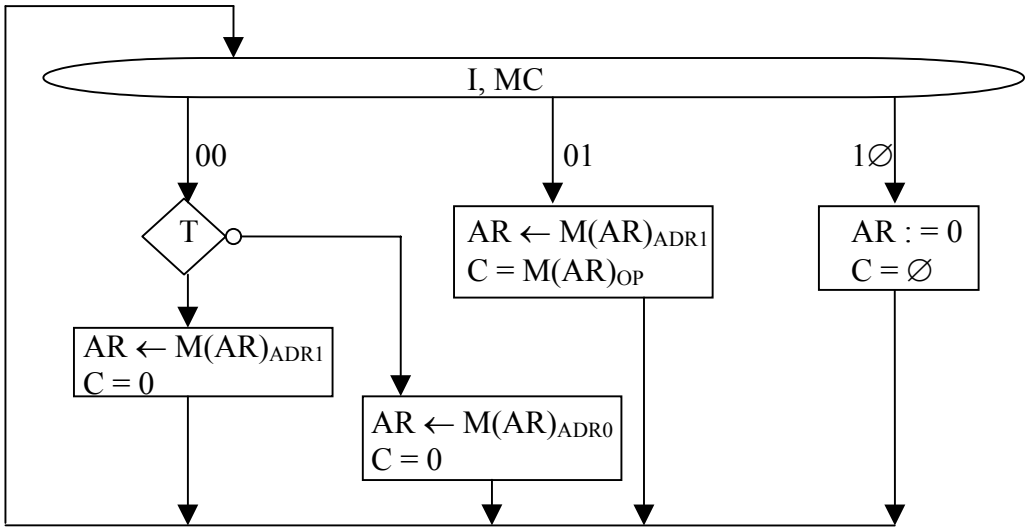
registru: AR_{1x4} – ca registru de adresă (Address Register) al PROM-ului

resurse: M_{15x13} – memorie de microprogram;

T_{1x1} – multiplexor de test;

C_{1x8} – registru pentru variabilele de comandă.

Organigrama din fig. 10 descrie funcționarea UC pentru fiecare din microinstrucțiuni, în funcție de codul lor MC şi pentru inițializarea efectuată atunci când variabila de intrare a dispozitivului de împărțire I = 1.



$M(AR)_{ADR1}$ – Conținutul locației din memoria M de la adresa dată de AR, partea ADR1 din această locație

Figura 10.

2.5. Schemă şi declarație adițională

Pe lângă elementele anterioare schema UC mai are nevoie de:

AM_{1x4} - MUX de adresă, pentru a selecta una sau cealaltă dintre cele 2 adrese furnizate de memorie (fig. 11).

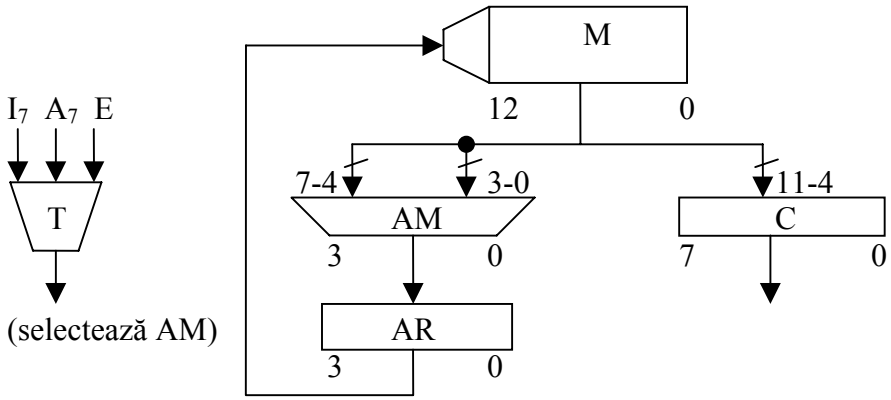


Figura 11.

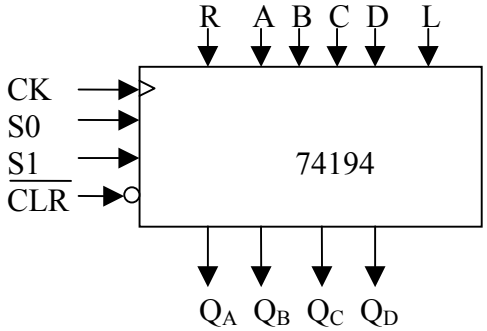
2.6. Realizarea

Tabelul din figura 12 exprimă operațiile descrise în organigrama UC (fig. 10) ținând cont de multiplexorul de adrese AM. Realizarea UC se reduce la alegerea unui ansamblu de regiştri şi resurse capabile să execute aceste operații.

Operație	Descriere
OP1	AR ← AM; AM = $M(AR)_{ADR1}$; C = 0
OP2	AR ← AM; AM = $M(AR)_{ADR0}$; C = 0
OP3	AR ← AM; AM = $M(AR)_{ADR1}$; C = $M(AR)_{OP}$
OP4	AR : = 0; AM = ∅; C = ∅

Figura 12.

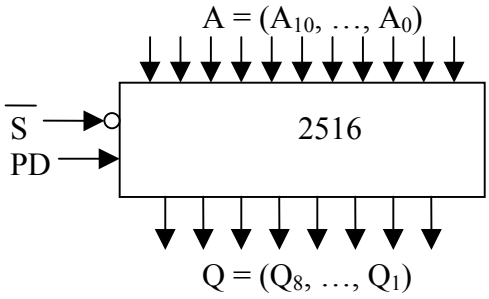
AR – se foloseşte registru de deplasare bidirecțional (74194).



Operație	Descriere	$\overline{CLR}$	S1	S0
CLEAR	$(Q_A, Q_B, Q_C, Q_D) \leftarrow (0,0,0,0)$	0	∅	∅
HOLD	$(Q_A, Q_B, Q_C, Q_D) \leftarrow (Q_A, Q_B, Q_C, Q_D)$	1	0	0
SHIFT RIGHT	$(Q_A, Q_B, Q_C, Q_D) \leftarrow (R, Q_A, Q_B, Q_C)$	1	0	1
SHIFT LEFT	$(Q_A, Q_B, Q_C, Q_D) \leftarrow (Q_B, Q_C, Q_D, L)$	1	1	0
LOAD	$(Q_A, Q_B, Q_C, Q_D) \leftarrow (A, B, C, D)$	1	1	1

AM şi **C** – se foloseşte MUX 2:1 pe 4 biți (74157) pentru **AM** şi 2 circuite ŞI cu 2 intrări (7408) pentru **C**.

M – se folosesc 2 EPROM de 2048x8 biți (2516).



Operație	Descriere	PD	$\overline{S}$
READ	Q = EPROM(A)	0	0
3-STATE	Q = $\bigtriangledown$	0	1
3-STATE POWER DOWN	Q = $\bigtriangledown$	1	∅

Conform figurilor 10 şi 12 şi ținând cont de registrul şi resursele alese vom defini starea variabilelor de comandă pentru execuția fiecăreia dintre operații (figura 13).

Operație	$\overline{CLR_{AR}}$	$S1_{AR}$	$S0_{AR}$	G_{AM}	S_{AM}	A_C	I	MC	T
OP1	1	1	1	0	0	0	0	0	1
OP2	1	1	1	0	1	0	0	0	0
OP3	1	1	1	0	0	1	0	1	∅
OP4	0	∅	∅	∅	∅	∅	1	∅	∅

Figura 13.

Valorile corespunzătoare variabilei de inițializare I, bitului de cod MC şi variabilei de test T conduc la următoarele relații logice:

$\overline{CLR_{AR}} = I$

$S1_{AR} = S0_{AR} = 1$

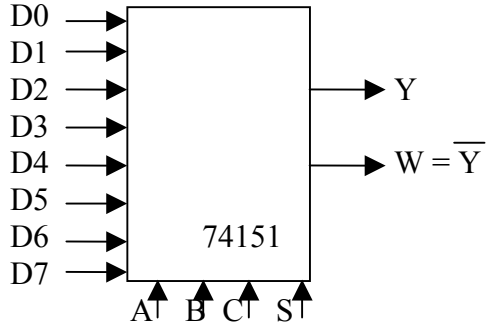
$G_{AM} = 0$

$S_{AM} = \overline{MC + T}$

$A_C = MC$

toate fiind funcții de I, MC şi T.

Multiplexorul de test T selectează variabilele E, A_7 şi I_7 relative la microinstrucțiunile de test binar ale programului. Aceste operații apar în detaliu în tabelul din figura 14, care precizează starea variabilelor de comandă ale MUX 8:1 de tip (74151), utilizat pentru efectuarea lor.



După cum se vede din codificarea utilizată în tabel se pune în corespondență starea câmpului TEST a microinstrucțiunii cu cea a variabilelor A_T şi B_T ale MUX.

Operație	Descriere	S_T	C_T	B_T	A_T	TEST
SELECT E	T = E	0	0	0	0	00
SELECT A_7	T = A_7	0	0	0	1	01
SELECT I_7	T = I_7	0	0	1	0	10

Figura 14.

2.7. Adaptarea şi programarea

Adaptarea se reduce la adresare, adică la numerotarea în hexazecimal a microinstrucțiunilor plecând de la 0. Atribuind adresa 0 microinstrucțiunii inițiale a programului originar şi numerotând microinstrucțiunile în ordine, obținem un program adaptat posibil pentru sistemul numeric dat.

0 DO OP1 AND GOTO 1
1 IF E = 0 THEN GOTO 0
ELSE GOTO 2
2 DO OP2 AND GOTO 3
3 DO OP3 AND GOTO 4
4 DO OP6 AND GOTO 5
5 IF A_7 = 0 THEN GOTO B
ELSE GOTO 6
6 DO OP3 AND GOTO 7
7 DO OP7 AND GOTO 8
8 IF I_7 = 0 THEN GOTO 5
ELSE GOTO 9
9 IF A_7 = 0 THEN GOTO E
ELSE GOTO A
A DO OP5 AND GOTO 0
B DO OP4 AND GOTO C
C DO OP3 AND GOTO D
D DO OP8 AND GOTO 8
E DO OP4 AND GOTO 0

În figura 15 se prezintă programul în hexazecimal al memoriei M a UC pentru situația în care câmpul nedefinit (∅) al microinstrucțiunii de test binar este ales egal cu 0. Redactarea acestui program se realizează pe baza programului adaptat al sistemului numeric, ținând cont de formatul microinstrucțiunilor (fig. 9), codificarea variabilelor de test (fig. 14) şi codificarea variabilelor de comandă (fig. 8).

Dimensiunea programului este de 15x13 biți = 195 biți.

Adresă (în hexa)	Conținut (în hexa)
0	1041
1	0002
2	1383
3	1404
4	1A15
5	01B6
6	1407
7	1A28
8	0259
9	01EA
A	1A00
B	190C
C	140D
D	1A38
E	1900

Figura 15.

TEMĂ: Să se deseneze schema unității de comandă UC cu 2 microinstrucțiuni cu registru de adresă.