

Sanda Maican

# Sisteme numerice cu circuite integrate

Culegere de probleme

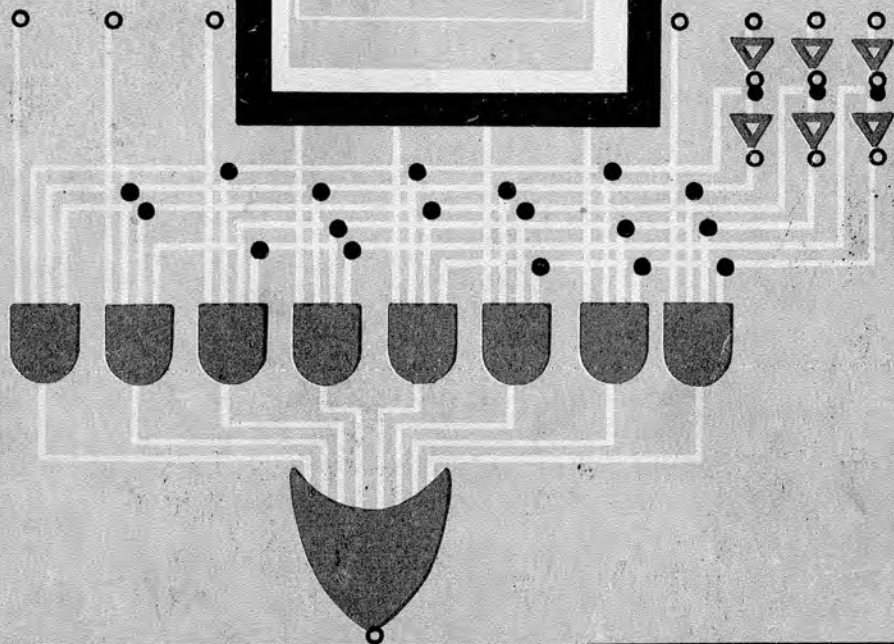
AUTOMATICĂ

ELECTRONICĂ

INFORMATICĂ

MANAGEMENT

SERIA PRACTICĂ



AUTOMATICA  
I N F O R M A T I C A  
E L E C T R O N I C A  
S E R I A P R A C T I C A  
M A N A G E M E N T

## SERIA PRACTICĂ

- Automatică
- Informatică
- Electronică
- Management

### Cielul, MĂSURAREA PARAMETRIILOR\*

### Cielul, STUDIUL MUNCII\*

- E. Samal. Tehnica reglării. Manual practic*  
*S. Bajureanu, A. Băldăcescu, M. Epure, I. Burlacu. Elemente și sisteme pneumatice*  
*I. Flores. Practica programării calculatoarelor*  
*F. G. Shinsky. Practica sistemelor de reglare automată*  
*E. J. McCarty ș.a. Sisteme integrate de prelucrare a datelor în conducerea activității economice*  
*M. Dumitrescu, P. Isac, P. Turcu, M. Ene (coordonatori). Organizarea conducerii, producției și muncii*  
*C. Simbotin, Cl. Tanasieciuc. Comutația statică în automatizări*  
*M. K. Starr. Conducerea producției. Sisteme și sinteze*  
*V. Crăciunoiu ș.a. Elemente de execuție*  
*A. Vlădescu ș.a. Radioreceptoare*  
*M. Mayer. Tiristoare în practică. Mutatoarele comutație forțată*  
*G. Moltgen. Tiristoare în practică. Mutatoare cu comutație de la rețea*  
*L. Zamfirescu, I. Opșescu. Automatizarea cuptoarelor industriale*  
*I. Popadache. Automatizări aplicată, ediția I-a și a II-a*  
*Șt. Alexandru. Automatizarea proceselor tehnologice în industria lemnului*  
*V. H. Iisickin. Prognoza tehnico-științifică în ramurile industriei*  
*G. Raymond. Tehnica televiziunii în culori*  
*J. J. Samuelly, J. Pignaret, A. Sarazin. Instrumentația electronică în fizica nucleară*  
*T. Homoș. Capacitatea de producție în construcții de mașini*  
*S. Radu, D. Filoti. Centrale telefonice automate. Sisteme de comutație*  
*R. Stere ș.a. Tranzistoare cu efect de câmp*  
*D. N. Sapiro. Proiectarea radioreceptoarelor*  
*V. Antonescu, M. Popovici. Ghid pentru controlul statistic al calității producției*  
*N. Stanciu ș.a. Tehnica imaginii în cinematografie și televiziune*  
*P. Vezeanu, Șt. Pătrașcu. Măsurarea temperaturii în tehnică*  
*T. Penescu, V. Petrescu. Măsurarea presiunii în tehnică*  
*P. Popescu, P. Mihoardea. Măsurarea debitului în tehnică*  
*P. Vezeanu. Măsurarea nivelului în tehnică*  
*C. Hidoș, P. Isac (coordonatori). Studiul muncii, vol. I-VIII*  
*V. Baltac ș.a. Calculatorul FELIX C-256. Structură și programare*  
*G. Sonea, M. Silefchi. Creșterea planificată a productivității muncii*  
*R. L. Morris. Proiectarea cu circuite integrate TTL*  
*A. Brilliantov. Calculul și construcția televizoarelor portabile cu tranzistoare*  
*I. Stăncioiu. Eficiența economică a asimilării de utilaje noi*  
*Ishikawa Kaoru. Controlul de calitate pentru maștri și șefi de echipe*  
*Magnus Radke 222 măsuri pentru reducerea costurilor*  
*A. M. Buhtiarov ș.a. Culegere de probleme de programare*  
*P. Constantinescu, C. V. Negoită. Sisteme informatice, modele ale conducerii și sistemelor conduse*  
*E. S. Bufla. Conducerea modernă a producției, vol. I și II*  
*A. Vătășescu ș.a. Dispozitive semiconductoare. Manual de utilizare*  
*A. Nadolo. Măsurarea volumului și cantității lichidelor în industrie*  
*Ch. Jones Design. Metode și aplicații*  
*Gh. Pisău ș.a. Elaborarea și introducerea sistemelor informatice*  
*C. Hidoș. Analiza și proiectarea circuitelor informaționale în unitățile economice*  
*A. Vătășescu ș.a. Circuite integrate liniare Manual de utilizare*  
*M. Silățeanu ș.a. Scheme de televizoare, magnetofone, picupuri*  
*vol. I și 2 ed. a II-a*  
*D. W. Davies, D. L. Barber. Rețele de interconectarea (cablurilor)*  
*V. Pescaru ș.a. Fișiere, baze și bănci de date*  
*D. Patriche. Marketing industrial*  
*Gh. Baștiurea ș.a. Comanda numerică a mașinilor-unelte*  
*N. Sprinceană, R. Dobrescu, Th. Borangiu. Automatizări discrete în industrie. Culegere de probleme*  
*M. Florescu, E. Niculescu-Mizil ș.a. Cibernetică, automatizări, informatică în industria chimică*  
*S. Călin, M. Tertîșco, I. Dumitrușco, ș.a. Optimizări în automatizări industriale*

Control științific: dr. ing. VASILE BALTAC  
Redactor: ing. SMARANDA DIMITRIU  
Tehnoredactor: MARIA TRĂSNEA  
Coperta seriei: CONSTANTIN GULUȚĂ

Bun de tipar: 02.07.1980. Coli de  
tipar: 20,25. C.Z. 681.142.642 : 261.382.

---

*Intreprinderea poligrafică „Oltenia“  
Str. Mihai Viteazul nr. 4 Craiova  
Republica Socialistă România*

## Cuvînt înainte

● *Lucrarea este de fapt teoretică : majoritatea celor 485 probleme atrag atenția asupra aspectelor teoretice legate de analiza și proiectarea cu CI, aspecte prezentate sub forma rezolvărilor, soluțiilor sau indicațiilor.*

*Am considerat că această formă de prezentare a teoriei în aplicații este mai atractivă și mai eficientă, solicitînd mai intens participarea cititorului.*

● *Lucrarea are și un caracter practic : încearcă să furnizeze cititorului cunoștințe care să-i permită să-și realizeze concret sistemele proiectate. Se demonstrează că metodele matematice (Capitolul 1) sînt foarte importante în proiectare, dar insuficiente. Sînt completate cu date privind funcționarea logică (Capitolele 2 și 3) și parametrii tehnologici ai CI (Capitolul 4). Trebuie să se țină cont de asemenea de modul de conectare între CI (Capitolul 5). Toate aceste cunoștințe îmbinate, crează baza pentru înțelegerea și proiectarea sistemelor logice microprogramate (Capitolul 6).*

● *Lucrarea are un caracter unilar : sînt prezentate atît elementele (diverse tipuri de CI) cît și metodele de structurare a acestora în sistemele numerice.*

● *Lucrarea are un caracter continuu : aspectele teoretice se întrepătrund, sînt înălțuite și prezentate gradat, în majoritatea cazurilor dificultatea crescînd treptat în cadrul capitolelor. Se recomandă deci parcurgerea în ordine a problemelor.*

● *Lucrarea are un caracter didactic : elaborată pe parcursul a doi ani, conține în majoritate probleme a căror formulare și utilitate pentru studiul sistemelor numerice a fost validată în activitatea de seminar și laborator desfășurată de autoare cu studenții Facultății de Electronică și TC — București.*

● *Punctele anterioare exprimă de fapt părerea autoarei despre cum ar trebui să arate o astfel de carte. Gradul în care a reușit să realizeze aceste intenții va rezulta în urma confruntării cu cititorii.*

● La începutul fiecărui capitol se face o prezentare a conținutului, atrăgându-se atenția asupra categoriilor de probleme parcurse. Anexa cuprinde principalele notații și date de catalog sumare ale circuitelor utilizate în probleme și care nu apar în catalogul IPRS. S-au indicat în lucrare codurile seriei 74XXX deoarece sînt descrise și circuite care nu se fabrică în țară. În anexă, unde este cazul, se indică în paranteză codurile IPRS.

● Autoarea mulțumește inginerilor : Cristian Mihail Băleanu, Bogdan Cocora, Ion Maican, Vlad Maican, Gheorghe M. Ștefan, Dan Tomescu pentru sprijinul primit pe parcursul diferitelor etape de elaborare a acestei lucrări.

● Autoarea mulțumește tov. prof. dr. doc. ing. Mihai Drăgănescu și tov. dr. ing. Vasile Baltac pentru sugestiile de îmbunătățire a lucrării și ajutorul acordat la publicarea acesteia.

● Autoarea mulțumește Editurii Tehnice pentru publicarea acestei lucrări.

SANDA MAICAN

# Cuprins

	Cuvînt înainte . . . . .	5
<i>Capitolul 1</i>	Metode matematice în teoria sistemelor și circuitelor numerice . . . . .	9
<i>Capitolul 2</i>	Circuite și sisteme logice combinaționale. Analiză și proiectare . . . . .	45
<i>Capitolul 3</i>	Circuite și sisteme logice secvențiale. Analiză și proiectare . . . . .	89
<i>Capitolul 4</i>	Metode practice de proiectare și realizare. Relația dintre implementare și tehnologie .	199
<i>Capitolul 5</i>	Probleme de cuplare a CI . . . . .	245
<i>Capitolul 6</i>	Sisteme logice microprogramate. Microprocesoare	261
<i>Anexa :</i>	Notății . . . . .	286
	Catalog sumar cu circuitele utilizate .	290
	<i>Bibliografie</i> . . . . .	322





**Prezentarea conținutului**

Întreagă această culegere are la bază considerarea problematicii sistemelor și circuitelor integrate digitale din perspectiva teoriei generale a sistemelor cu stări finite, SSF [1], [2]. Prin particularizarea acesteia, în capitolul 1 se obține o modelare matematică utilă în descrierea sistemelor electronice (vezi problema 1.1). Sistemul logic secvențial pe baza căruia se pot obține celelalte tipuri de sisteme numerice (vezi problema 1.2) poate fi modelat ca în figura 1.

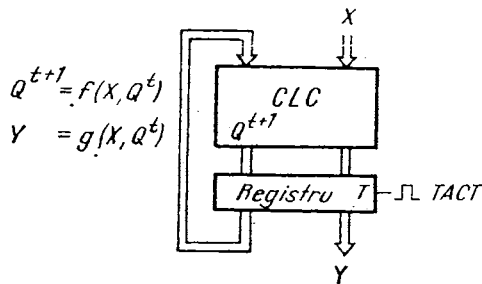


Fig. 1.

— Comportarea sistemului logic secvențial este descrisă cu ajutorul funcțiilor caracteristice  $f$  și  $g$  care arată cum evoluează ieșirile ( $Y$ ) și stările ( $Q$ ) la un anumit moment de timp ( $t + 1$ ) în raport cu stările de la momentul de timp anterior ( $t$ ) și intrările ( $X$ ).

În problemele 1.5—1.9 sînt prezentate modalitățile de explicitare matematică a funcțiilor caracteristice : graf, tabel de tranziție, organigramă, forme de undă și translații între acestea.

Se explică apariția hazardului combinațional și modul de înlăturare al acestuia (pr. 1.2 și 1.3—1.9).

— Comportarea circuitului logic combinațional ,CLC, este independentă de timp și este descrisă cu ajutorul funcțiilor booleene. Problemele 1.13—1.27 familiarizează cititorul cu postulatele și teoremele algebrei booleene.

Modalitățile de explicitare a funcțiilor booleene (tabel de adevăr, forme canonice, diagrame Veitch-Karnaugh ș.a.) sînt prezentate în problemele 1.28—1.34.

O altă problemă importantă tratată în acest capitol este cea a minimizării sistemelor și constă în găsirea sistemului minim echivalent cu cel dat. Aceasta are implicații directe în proiectare, la reducerea numărului circuitelor integrate cu care se realizează practic sistemul.

Minimizarea se poate obține prin următoarele metode :

— codificarea în spațiile de intrare, ieșire și stare (vezi problemele 1.3—1.9). Acest lucru nu este posibil în toate cazurile, uneori intrările și ieșirile fiind impuse de sistemele exterioare.

— reducerea spațiului de stare prin partiția acestuia în clase de echivalență (vezi problemele 1.11 și 1.12).

— minimizarea circuitului logic combinațional asociat sistemului prin metoda diagramelor Veitch-Karnaugh și Quinne Mc Cluskey (problemele 1.35—1.47).

Întregul aparat matematic prezentat în acest capitol este utilizat în restul lucrării fiind presupus cunoscut în capitolele următoare.

Modelarea matematică a sistemelor logice pornind de la teoria generală a SSF conduce la rezultate interesante constînd în :

● Clasificarea sistemelor digitale (vezi problema 1.2).

● Descrierea matematică a comportării sistemelor digitale (vezi capitolul 1).

● Stabilirea unei metode generale de proiectare a sistemelor logice secvențiale sincrone (vezi problema 3.112) ș.a.

**Problema 1.1.** Să se prezinte cele două modalități de definire matematică a unui sistem cu stări finite, SSF.

*Rezolvare*

**A.** Un SSF poate fi descris matematic cu ajutorul unui cvintuplu :

$$S = (X, Y, Q, f, g)$$

unde :

$X\{x_1, x_2, \dots, x_p\}$  este mulțimea variabilelor de intrare în sistem ;  
 $Y\{y_1, y_2, \dots, y_m\}$  — mulțimea variabilelor de ieșire ;  
 $Q\{q_1, q_2, \dots, q_n\}$  — mulțimea stărilor sistemului ;  
 $f; X \times Q \rightarrow Q$  — funcția de tranziție a stărilor care definește procesul de modificare a stărilor ca fiind dependent de intrare și de starea anterioară ;  
 $g: X \times Q \rightarrow Y$  — (modelul Mealy) este funcția de tranziție a ieșirilor. În modelul Mealy,  $g$  arată că ieșirile se modifică în funcție de intrări și de starea anterioară. În modelul Moore,  $g: Q \rightarrow Y$ , ieșirile depind doar de stări anterioare. Se poate arăta că cele două modele sînt echivalente.

Sistemul astfel descris se numește cu stări finite deoarece mulțimile  $X, Y, Q$  sînt finite (deci discrete). De asemenea spațiul timpului, care nu apare explicit în definiție este discret și format din mulțimea numerelor întregi  $T\{0, 1, 2, \dots, N\}$ .

Funcțiile  $f$  și  $g$  se numesc funcții caracteristice și definesc evoluția sistemului în timp. Ele pot fi reprezentate prin :

a) tabele de tranziții (fig. 1.1).

Se reprezintă pe coloane variabilele de intrare și pe linii componentele spațiului stărilor. În intersecțiile rezultate se notează starea și ieșirea următoare. De exemplu, în tabel se vede că, dacă sistemul  $S$  este în starea  $q_1$  și se aplică la intrare  $X_1$ ,  $S$  trece în starea  $q_n$  și are la ieșire  $Y_3$ . Deci, intersecțiile definesc funcțiile  $f$  și  $g$ .

b) grafuri de tranziție (fig. 1.2).

$Q \backslash X$	$X_1$	$X_2$	.....	$X_p$
$q_1$	$q_n, Y_3$			
$q_2$		$f, g$		
$\vdots$				
$q_n$				

Fig. 1.1.

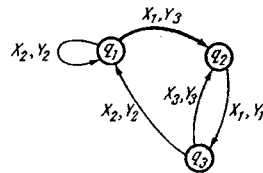


Fig. 1.2.

În noduri se reprezintă stările sistemului, iar laturile conțin determinări de intrare și ieșire. Un exemplu de interpretare a grafului (pentru latura îngroșată din figură) este : dacă sistemul  $S$  se află în starea  $q_1$  și se aplică la intrare  $X_1$ ,  $S$  trece în starea  $q_2$  și va livra la ieșire  $Y_3$ .

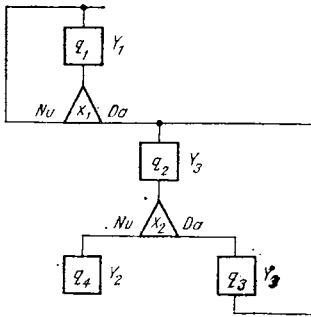


Fig. 1.3.

c) organigrame (fig. 1.3).

În dreptunghiuri se notează stările sistemului și alăturat, ieșirile.  $X_1$  și  $X_2$  sînt condițiile de intrare, testate în vederea stabilirii evoluției sistemului. De exemplu, dacă sistemul este în starea  $q_1$  și are la intrare  $X_1$ , trece în starea  $q_2$  și va avea la ieșire  $Y_3$ . Dacă în urma testării,  $X_1$  nu a apărut la intrare, sistemul rămîne în starea  $q_1$ .

**B.** Un SSF poate fi descris matematic și prin tripletul :

$$S = (X, Y, M)$$

unde :

$X$  este spațiul de intrare ;

$Y$  este spațiul de ieșire ;

$M : X \rightarrow Y$  este funcția de intrare-ieșire.

Această definiție a comportării intrare-ieșire este foarte utilă în experimente, în diagnoză de sistem, în timp ce definiția prin cvintuplu este utilă în analiza și sinteza sistemelor.

În cazul sistemelor electronice, funcția  $M$  este explicitată cu ajutorul formelor de undă. Acestea se pot obține cu ușurință pe ecranul unui osciloscop și constituie semnalele la ieșirile sistemului în funcție de intrări, măsurate la diferite momente de timp. Amănunte despre acest mod de descriere a comportării sistemelor se dau în problemele 1.3 și 1.6—1.9.

**Problema 1.2.** Să se particularizeze definiția unui SSF ca cvintuplu pentru cazul circuitelor și sistemelor logice combinaționale, secvențiale și microprogramate. Ce diferențe apar între modelul matematic și comportarea reală a sistemelor pusă în evidență experimental ?

*Rezolvare*

**CIRCUITUL LOGIC COMBINAȚIONAL, CLC,** este un caz particular de SSF la care spațiul stărilor este mulțimea vidă :

$$Q \equiv \emptyset$$

Se observă că în acest caz :

a)  $f : \emptyset \times X \rightarrow \emptyset$ , deci funcția  $f$  nu există ;

b)  $g : \emptyset \times X \rightarrow Y$ , deci  $g : X \rightarrow Y$  este identică cu funcția  $M$  de intrare-ieșire, domeniul de definiție al funcției  $g$  reducîndu-se la mulțimea de intrare.

Deci pentru  $\bar{C}\bar{L}\bar{C}$ , cvintuplul de definire se reduce la tripletul:

$$S_c = (X, Y, g \equiv M)$$

punîndu-se astfel în evidență faptul cunoscut că la ieșirea CLC, variabilele nu depind de timp sau stare, ci doar de vectorul de intrare (fig. 1.4).

În modelarea matematică,  $g$  se exprimă cu ajutorul funcțiilor booleene (vezi problemele 1.13—1.26).

Experimental  $g$  se determină prin formule de undă care arată cum se modifică ieșirile în funcție de intrări. Astfel, în figura 1.5, trecerea lui  $X_1$  din 0 în 1, va determina comutarea lui  $Y_2$  din 1 în 0. Experimental se constată că modificarea ieșirii nu este simultană cu cea a intrării, ci se face cu o anumită întârziere, după un timp numit *timp de propagare* ( $t_p$ ). Timpii de propagare depind de funcția logică realizată de fiecare ieșire și de vectorul de intrare.

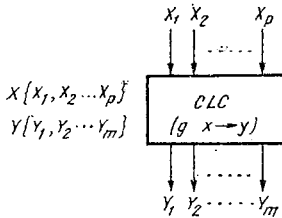


Fig. 1.4.

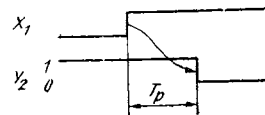


Fig. 1.5.

Notăm timpii de propagare pentru vectorul de intrare

$$V_1(X_1 = X_2 = \dots = X_p = 0) \text{ cu:}$$

$$t_{PY_1}^{V_1}, t_{PY_2}^{V_1}, \dots, t_{PY_m}^{V_1},$$

pentru vectorul de intrare  $V_2(X_1 = X_2 = \dots = X_{p-1} = 0, X_p = 1)$  cu:

$$t_{PY_1}^{V_2}, t_{PY_2}^{V_2}, \dots, t_{PY_m}^{V_2}$$

Deci pentru vectorul de intrare  $V_k$  vom avea timpii de propagare

$$t_{PY_1}^{V_k}, t_{PY_2}^{V_k}, \dots, t_{PY_j}^{V_k}, \dots, t_{PY_m}^{V_k}$$

Răspunsul circuitului va fi corect la ieșiri numai după ce efectul unei modificări la intrare s-a propagat la *toate* ieșirile. Și aceasta trebuie să se întîmple pentru *toate* modificările posibile la intrare. Deci, după

un timp ( $t_p$ ) egal cu cel mai mare dintre toți timpii de propagare sîntem siguri de răspunsul sistemului la apariția vectorului de intrare respectiv :

$$t_p = \text{Max}\{t_{PY_j}^{V_K}\} \quad \begin{matrix} K = 1, 2, \dots, 2^P \\ j = 1, 2, \dots, n \end{matrix}$$

În intervalul de timp  $[0 - t_p]$  pot apare mai mulți vectori de ieșire, diferiți de vectorul de răspuns corect, care apare la momentul  $t_p$  și se menține pînă la o nouă schimbare a vectorului de intrare. Acest fenomen care se petrece în intervalul  $[0 - t_p]$  și poartă numele de *hazard combinațional* nu poate fi pus în evidență prin metodele matematice de exprimare a funcțiilor  $g$  cu ajutorul algebrei booleene, dar trebuie să se țină cont de el în proiectarea și analiza sistemelor.

SISTEMUL LOGIC COMBINAȚIONAL, SLC, este un CLC la care se elimină apariția hazardului prin adăugarea unei memorii care preia datele de la ieșirea CLC doar la momente discrete de timp, mai mari decît  $t_p$  (fig. 1.6). Aceasta se realizează prin comanda memoriei cu impulsuri de tact (ceas-clock),  $T$ . Prin cuplarea în cascade a unor astfel de CLC, la momentul  $t$  se schimbă vectorul de intrare, iar la  $t + 1$ , vectorul de ieșire corect este transmis la ieșirea memoriei.

SISTEMUL LOGIC SECVENȚIAL, SLC, este format de un SSF căruia i se pot aplica definițiile generale date în problema 1.1. Trebuie observat că un SLS se poate reprezenta ca un SLC cu reacție. La ieșirile SLS din figura 1.7 apare hazard combinațional, în timp ce la ieși-

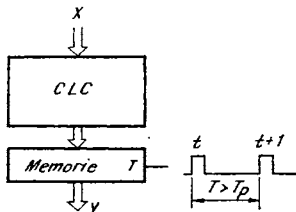


Fig. 1.6.

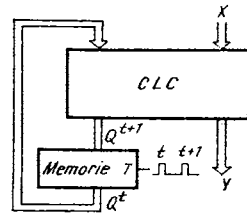


Fig. 1.7.

rile SLS din figura 1.8 hazardul este eliminat ; sistemul se numește sincron. Pentru ambele figuri se observă că atât ieșirea  $Y$  cit și starea  $Q$  depind de intrare și de starea la momentul de timp anterior.

Notarea tactului cu  $T$  sau  $\bar{T}$  arată că sistemul comută pe frontul pozitiv, respectiv negativ al impulsurilor.

Cazuri particulare de SLS mai des întîlnite sînt :

- sisteme autonome, la care mulțimea de intrare nu are elemente.

Acestea sînt sisteme ce-și modifică stările după o lege proprie.

Exemplu : generatoare de tact, divizare de frecvență ;

● sisteme la care  $Y = Q$ .

Exemplu : bistabili, numărătoare.

SISTEMUL LOGIC MICROPROGRAMAT, SLM, este un SSF cu spațiul stărilor structurat, în care  $Q$  nu va fi o mulțime ci un sistem definit prin cvintuplul :

$$Q = (X^{10}, Y^{10}, Q^{10}, f^1, g^1).$$

Deci la SLM reacția este aplicată intrării prin intermediul unui alt sistem ca în figura 1.9.

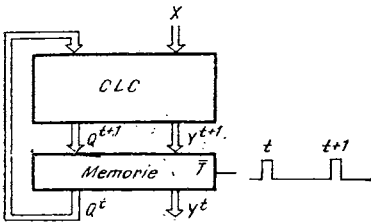


Fig. 1.8.



Fig. 1.9.

În funcție de proprietățile lui  $Q$  și ale funcției  $g$  deosebim cazurile în care SL de reacție este : a) registru, b) SLC, c) SLS, d) SLM.

**Problema 1.3.** Se dă sistemul definit de următorul tabel de tranziție :

$Q \backslash X$	0	1
A	B, 1	D, 0
B	C, 1	A, 0
C	D, 0	B, 1
D	A, 0	C, 1

Să se stabilească :

- organigrama sistemului ;
- graful de tranziții al sistemului ;
- particularizarea modelului Mealy pentru acest sistem cu eliminarea hazardului combinațional ;
- formele de undă în funcție de impulsurile de ceas ale sistemului.

*Rezolvare*

a) Pentru stabilirea organigramei se pornește dintr-una din stări (de exemplu din A). Se observă că, în funcție de  $X$  se ajunge în stări

distincte (în  $B$  sau  $D$ ). Apoi se consideră aceste noi stări și se pornește din ele urmărindu-se în continuare traiectoria sistemului. Organigrama este cea din figura 1.10.

b) Graful sistemului este dat în figura 1.11.

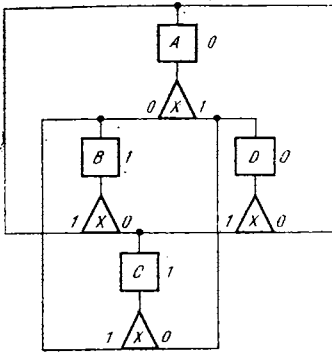


Fig. 1.10.

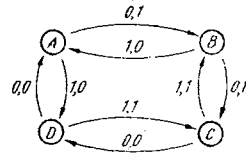


Fig. 1.11.

Așa cum s-a indicat în problema 1.1, în nodurile grafului se notează stările, iar pe laturi se notează intrările, ieșirile. Sensul laturii indică pe cel de evoluție a sistemului.

c) Particularizarea modelului Mealy presupune stabilirea numărului de intrări, ieșiri și căi de reacție. Pentru reducerea acestui număr se face o codificare binară. Astfel pentru  $2^n$  intrări (respectiv ieșiri sau stări) sînt necesari  $n$  biți pentru codificare. În cazul acestei probleme există o singură intrare în sistem,  $X$ , luînd valoarea 0 sau 1. Există o singură ieșire  $Y$ , cu valoarea 0 sau 1.

Cele 4 stări se pot codifica cu 2 biți ca în tabelul următor :

starea \ codul	codul	
	$q_1$	$q_2$
A	0	0
B	0	1
C	1	0
D	1	1

Codificarea, prin reducerea numărului de intrări, ieșiri și stări, conduce în majoritatea cazurilor la minimizarea sistemului, cînd acesta trebuie proiectat.



Modelul Mealy pentru sistemul particular al acestei probleme este prezentat în figura 1.12. În figura 1.13 este dată o schemă echivalentă în care în dreptul intrărilor, ieșirilor și stărilor se indică numărul acestora. Memorarea se realizează cu un registru,  $R$ .

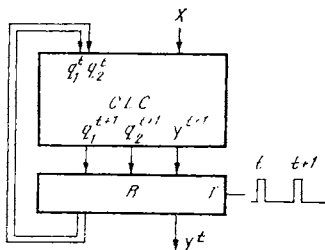


Fig. 1.12.

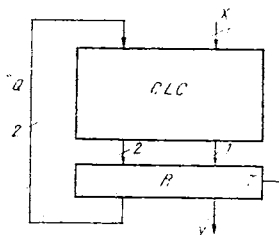


Fig. 1.13.

d) Formele de undă (fig. 1.14) se stabilesc pornind de la tabel, graf de tranziții sau organigramă, considerînd că sistemul evoluează dintr-o stare inițială arbitrară (fie aceasta  $A$ ). Pentru reprezentarea stărilor s-a ținut cont de codificarea de la punctul c. S-a considerat că sistemul comută cu frontul pozitiv al impulsurilor de tact.

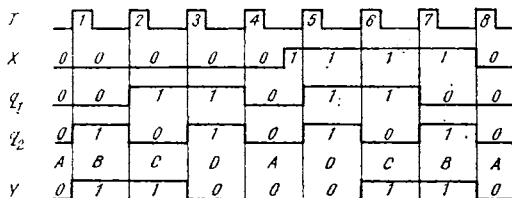


Fig. 1.14.

e) Traiectoriile ce descriu comportarea sistemului sînt două, identice cu formele de undă pentru  $X = 0$ , respectiv pentru  $X = 1$ .

**Problema 1.4.** Să se reia problema 1.3 pentru sistemul descris de următorul tabel de tranziții :

$Q \backslash X$	$X_1$	$X_2$
$A$	$B, 1$	$C, 0$
$B$	$C, 0$	$A, 1$
$C$	$A, 1$	$B, 1$

*Indicație.* Urmăriți rezolvarea problemei 1.3.

c) Mulțimea de intrare are două elemente,  $X_1$  și  $X_2$ . Codificarea va fi  $X_1 = 0$ ,  $X_2 = 1$ , deci există o singură cale de intrare.

Mulțimea de stare are trei elemente deci se codifică cu 2 biți, una din combinații rămânând neutilizată.

**Problema 1.5.** Pentru sistemul definit de organigramă din figura 1.15 să se stabilească :

- tabelul de tranziții ;
- graful de tranziții ;
- modelul Mealy al sistemului.

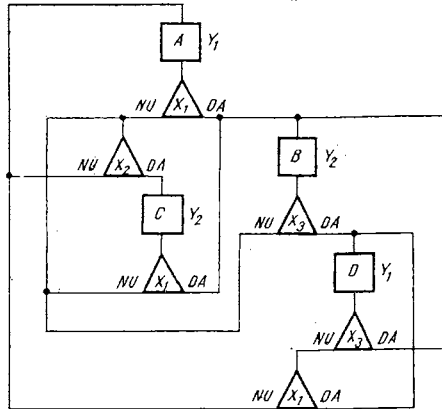


Fig. 1.15.

*Soluție*

- Tabelul de tranziții este dat în figura 1.16.
- Graful de tranziții este cel din figura 1.17.

$X \backslash Q$	$X_1$	$X_2$	$X_3$
A	$B, Y_2$	$C, Y_2$	$A, Y_1$
B	$A, Y_1$	$C, Y_2$	$D, Y_1$
C	$B, Y_2$	$C, Y_2$	$A, Y_1$
D	$D, Y_1$	$A, Y_1$	$B, Y_2$

Fig. 1.16.

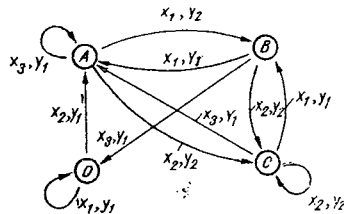


Fig. 1.17.

c) În figura 1.18 este prezentat modelul Mealy al sistemului cu hazard combinațional, iar în figura 1.19 cel fără hazard combinațional.

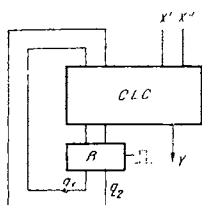


Fig. 1.18.

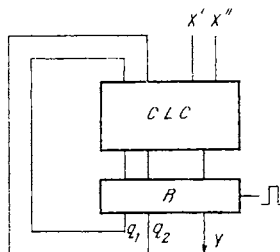


Fig. 1.19.

Am presupus următoarea codificare :

stare \ cod	$q_1$	$q_2$
A	0	0
B	0	1
C	1	0
D	1	1

intrare \ cod	$X'$	$X''$
$X_1$	0	1
$X_2$	1	0
$X_3$	1	1

ieșire \ cod	Y
$Y_1$	0
$Y_2$	1

**Problema 1.6.** Pentru sistemul definit de graful din figura 1.20 să se deducă :

- tabelul de tranziții ;
- organigrama ;
- modelul Mealy fără hazard combinațional ;
- formele de undă.

*Rezolvare*

a) Tabelul de tranziții este cel din figura 1.21.

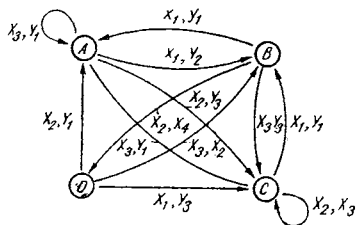


Fig. 1.20.

Q \ X	$X_1$	$X_2$	$X_3$
A	$B, Y_2$	$C, Y_3$	$A, X_1$
B	$A, Y_1$	$D, Y_4$	$C, Y_3$
C	$B, Y_2$	$C, Y_3$	$A, Y_1$
D	$C, Y_3$	$A, Y_1$	$B, Y_2$

Fig. 1.21.

b) O organigramă posibilă obținută prin testarea  $X_1$  și  $X_2$  este cea din figura 1.22.

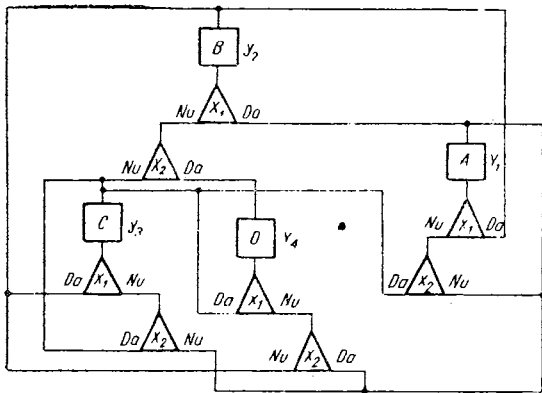


Fig. 1.22.

c) Sistemul are :

- 3 variabile de intrare codificabile cu 2 biți  $X'$  și  $X''$
- 4 variabile de stare codificabile cu 2 biți  $q_1$  și  $q_2$
- 4 variabile de ieșire codificabile cu 2 biți  $Y'$  și  $Y''$ . Deoarece pentru fiecare stare corespunde o configurație de ieșire distinctă se pot asocia aceleași coduri pentru B și  $Y_2$ , pentru A și  $Y_1$ , pentru C și  $Y_3$ , respectiv pentru D și  $Y_4$ . Deci spațiul ieșirilor este identic cu cel al stărilor. Modelul Mealy fără hazard combinațional este dat în figura 1.23.

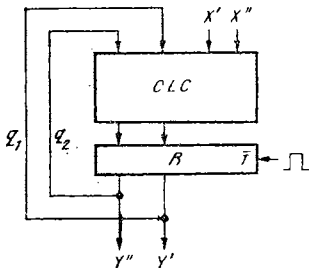


Fig. 1.23.

d) Cu următoarea codificare a intrărilor și ieșirilor (stărilor) și modelul Mealy din figura 1.23, pentru a determina formele de undă

se consideră toate cele 3 situații de intrare și, menținându-se parametrii  $X$  neschimbați, se dau impulsuri de tact sistemului, analizându-se răspunsul la ieșire.

	cod	$X'$	$X''$
intrare			
$X_1$		0	0
$X_2$		0	1
$X_3$		1	0

	cod	$q_1$	$q_2$
stare			
$A$		0	0
$B$		0	1
$C$		1	0
$D$		1	1

Astfel pentru  $X_1 = 00$ , formele de undă se dau în figura 1.24.

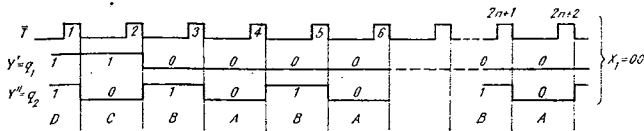


Fig. 1.24.

Am considerat că sistemul comută pe frontul negativ al impulsurilor de tact (acesta este sensul notației  $\bar{T}$  din fig. 1.23). Se observă că, după maximum două tacturi, sistemul intră în ciclul de repetiție al stărilor  $BA$ .

În figura 1.25 se dau formele de undă pentru  $X_2 = 01$  din care rezultă că, după maximum 3 impulsuri de tact, sistemul se blochează în starea  $C$ .

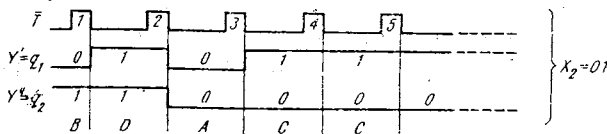


Fig. 1.25.

Pentru  $X_3 = 10$ , formele de undă din figura 1.26 ilustrează blocarea sistemului în starea  $A$  peste 0, 1, 2 sau 3 impulsuri de tact, după cum starea inițială este respectiv  $A$ ,  $C$ ,  $B$  sau  $D$ .

Formele de undă se stabilesc cel mai ușor cu ajutorul grafului de tranziții și se utilizează frecvent în descrierea experimentală a sistemelor deoarece se pot obține cu ajutorul osciloscopului.

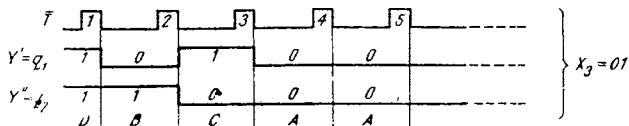


Fig. 1.26.

**Problema 1.7.** Se dă sistemul logic cu următoarea evoluție experimentală cu comutare pe frontul pozitiv al tactului.

Intrare	Ișire (stare)	Cod		Cod				
		Intrare	X'	X''	Stare	q <sub>1</sub>	q <sub>2</sub>	q <sub>3</sub>
X <sub>1</sub>	DABCEBCE .....	X <sub>1</sub>	0	0	A	0	0	1
X <sub>2</sub>	ECBADAD .....	X <sub>2</sub>	0	1	B	0	1	0
X <sub>3</sub>	ABCDEE .....	X <sub>3</sub>	1	0	C	1	0	0
X <sub>4</sub>	BECDADA .....	X <sub>4</sub>	1	1	D	0	0	0
					E	1	1	1

Să se determine :

- Modelul Mealy al sistemului, fără eliminarea hazardului combinațional ;
- graful de tranziții ;
- organigrama ;
- tabelul de tranziții.

*Indicație*

Pentru simplificare, evoluția sistemului nu a fost dată prin desen. Trecerea de la descrierea alfanumerică la forme de undă este imediată (vezi fig. 1.27) și acestea din urmă sînt m i utile în rezolvarea problemei.

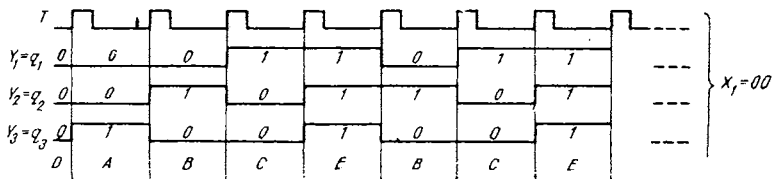


Fig. 1.27.

**Problema 1.3.** Pentru sistemul definit de formele de undă din fig. 1.28, să se determine :

- modelul Mealy (fără hazard combinațional) ;
- graful de tranziții ;
- organigrama.

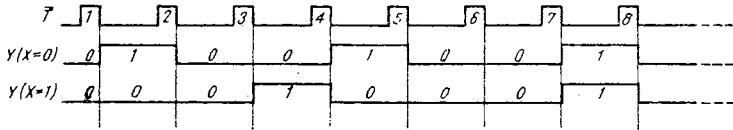
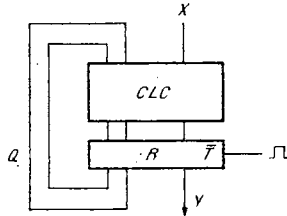


Fig. 1.28.

*Rezolvare*

a) Pentru  $X = 0$ , sistemul repetă la ieșire configurația 010, deci are 3 stări interne. Pentru  $X = 1$ , se repetă 0001, deci 4 stări. Numărul maxim de stări fiind 4, codificarea se face pe 2 biți. Modelul Mealy este dat în figura 1.29. Se observă că spațiul de stare este distinct de cel de ieșire.

Fig. 1.29.



b) În acest caz graful de tranziții nu este univoc definit deoarece cunoaștem doar numărul stărilor nu și succesiunea acestora. Două din soluțiile posibile sînt date în figurile 1.30 și 1.31.

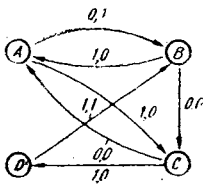


Fig. 1.30.

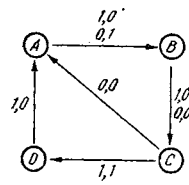


Fig. 1.31.

*Indicație*

c) Organigrama este de asemenea neunivoc definită și se construiește pornind de la unul din grafurile posibile.

**Problema 1.9.** Să se stabilească modelul Mealy și graful de tranziții pentru sistemul definit în figura 1.32.

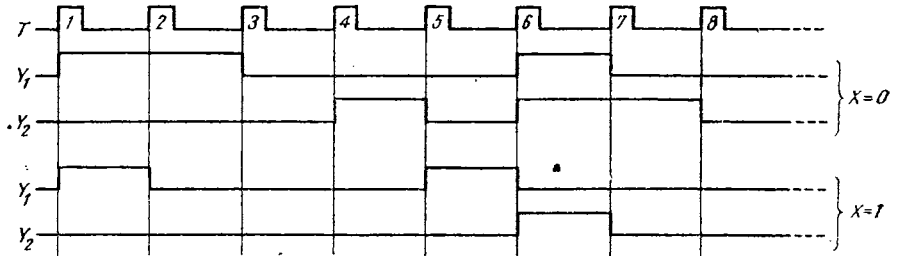


Fig. 1.32.

*Indicație*

Problema se rezolvă similar cu 1.8. Sistemul are 8 stări distincte, 1 intrare și 2 ieșiri și comută cu frontul pozitiv.

**Problema 1.10.** Să se demonstreze că relația din spațiul stărilor definită de proprietatea „pentru orice intrare  $X_K$  aplicabilă în stările  $q_1$  și  $q_2$ ,  $q_1$  conduce la aceeași valoare a funcției de ieșire ca și  $q_2$ , adică :

$$g(X_K, q_1) = g(X_K, q_2)''$$

este o relație de echivalență.

*Indicație*

Pentru a face demonstrația e necesar să cunoaștem următoarele definiții :

*Produsul a 2 mulțimi*,  $E = E_1 \times E_2$  este o mulțime ce conține toate perechile ordonate  $(e_{1k}, e_{2j})$  formate din elementele acelor mulțimi.

*Relația binară.* Proprietatea  $P$  definește o relație binară în mulțimea produs a 2 mulțimi dacă realizează o partiție a acesteia în 2 submulțimi  $G$  și  $\bar{G}$  astfel ca :

$$G \cap \bar{G} = \emptyset \text{ și } G \cup \bar{G} = E$$

*Relația de echivalență* este o relație binară cu proprietățile :

● simetric :

dacă  $(x, y) \in G$ , atunci și  $(y, x) \in G$



● reflexivitate :

dacă  $X \in E$ , rezultă că  $(x, x) \in G$

● tranzitivitate :

dacă  $(x, y) \in G$  și  $(y, z) \in G$ , atunci și  $(x, z) \in G$ .

În cazul problemei proprietatea  $P$  este cea specificată de textul subliniat. Trebuie demonstrat că este simetrică, reflexivă și tranzitivă.

De exemplu simetria rezultă din corectitudinea propoziției :

„pentru orice intrare  $X_K$  aplicabilă în stările  $q_2$  și  $q_1$ ,  $q_2$  conduce la aceeași valoare a funcției de ieșire ca și  $q_1$ , adică :

$$g(X_K, q_2) = g(X_K, q_1)“.$$

**Problema 1.11.** Se dă sistemul  $S$  definit prin următorul tabel de tranziție :

$Q \backslash X$	0	1	2
1	2/1	2/0	3/0
2	1/0	4/1	4/1
3	2/1	2/0	5/0
4	3/0	2/1	2/1
5	6/1	4/0	3/0
6	8/0	9/1	6/1
7	6/1	2/0	8/0
8	4/1	4/0	7/0
9	7/0	9/1	7/1

a) Să se stabilească graful de tranziție al sistemului.

b) Să se reducă spațiul stărilor sistemului  $S$ .

c) Să se determine tabelul de tranziție al sistemului  $S_e$ , sistemul minim echivalent cu  $S$ .

*Rezolvare*

b) Se determină partiția în clase de echivalență prin următorul algoritm simplu :

— Se determină partiția în clase de 1 — echivalență,  $C_1$  (adică stările echivalente când se aplică la intrare secvențe de lungime 1). Pentru aceasta :

1. se determină întâi perechile de stări echivalente (stările care, la aplicarea intrărilor  $X = 0, 1, 2$  conduc la aceleași valori ale funcției de ieșire). Pentru o determinare mai simplă a acestor stări și pentru pregătirea etapei următoare se construiește tabelul din figura 1.33.

Linile și coloanele reprezintă stările 2—9 și 1—8. Cu  $x$  sînt notate perechile de stări neechivalente. În cazul stărilor echivalente, în intersecție sînt notate perechile de stări care rezultă din aplicarea secvenței

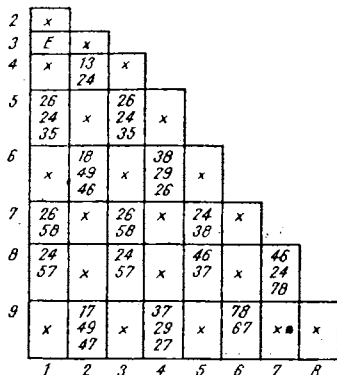


Fig. 1.33.

de intrare. De exemplu, stările (1, 2) nu sînt echivalente deoarece pentru secvențele de intrare 0, 1, 2 se obțin secvențele de ieșire (1, 0, 0)<sub>1</sub> și (0, 1, 1)<sub>2</sub> care nu sînt identice.

Stările 1 și 5 sînt echivalente deoarece conduc la secvențe de ieșire identice (1, 0, 0), iar stările succesive vor fi 2, 6 pentru  $X = 0$ , 2, 4 pentru  $X = 1$  și 3, 5 pentru  $X = 2$ .

Stările 1 și 3 se notează cu  $E$  deoarece conduc nu numai la ieșiri identice dar și la stări identice.

2. se construiesc clasele de echivalență inspectînd perechile echivalente pe coloane, de la dreapta la stînga. Găsim astfel perechile 7, 8, apoi 6, 9, apoi 5, 8 și 5, 7. Se observă că datorită tranzitivității relației de echivalență, cu perechile 7, 8 și 5, 8 și 5, 7 se poate forma clasa de stări echivalente 5, 7, 8. Se continuă procedeul într-un mod similar. Rezultatele sînt date în tabelul următor.

Clasele maxime rezultate  $C_1'$  și  $C_1''$  alcătuiesc pe

$$C_1 = \{1, 3, 5, 7, 8\}, \{2, 4, 6, 9\}$$

care reprezintă partiția de 1 — echivalență.

Perechi echivalente	Clase de echivalență rezultate
78	
69	
58, 57	5, 7, 8
49, 47	4, 6, 9
38, 37, 35	3, 5, 7, 8
29, 26, 24	2, 4, 6, 9 — $C_1''$
18, 17, 15, 13	1, 3, 5, 7, 8 — $C_1'$

— Se determină partiția în clase de 2 — echivalență,  $C_2$  (clase ce conțin stările echivalente pentru toate secvențele de intrare de lungime 2). Aceasta se face examinând tabelul și eliminând perechile de stări ce nu conduc la stări 1 — echivalente, aparținând lui  $C'_1$  sau  $C''_1$ . De exemplu, 1 și 8 sînt 2 — echivalente deoarece conduc la perechile de stări 2, 4  $C'_1$  și 5, 7  $C'_1$ . Dar 2, 9 și 4, 9 și 6, 9 nu sînt 2 — echivalente deoarece conduc la 4, 7<sub>X=1</sub>, 2, 9<sub>X=2</sub> respectiv 6, 7<sub>X=2</sub> care nu sînt echivalente. Deci la partiția după secvența de lungime 2, starea 9 nu mai este echivalentă cu 2, 6 și 4. Rezultă :

$$C_2 = \{1, 3, 5, 7, 8\} \{2, 4, 6\} \{9\}$$

— Se determină partiția în clase de 3 — echivalență eliminînd din tabel perechile ce conduc la 2, 9, 4, 9, și 6, 9 și care vor fi discernabile (neechivalente) la aplicarea secvențelor de lungime 3. Rezultă :

$$C_3 = \{1, 3, 5, 7, 8\} \{2, 4\} \{6\} \{9\}$$

— Se determină partiția în clase de 4 — echivalență :

$$C_4 = \{1, 3, 8\} \{5, 7\} \{2, 4\} \{6\} \{9\}$$

— Se determină partiția în clase de 5 — echivalență. Întrucît nu mai apar elemente discernabile rezultă :

$$C_4 = C_5 = C$$

unde  $C$  este partiția sistemului  $S$  în clase de echivalență.

Am redus astfel spațiul  $Q$  al sistemului  $S$  de la 9 la 5 stări cu corespondențele :

$$\begin{aligned} 1, 3, 5 &\rightarrow Q_1 \\ 5, 7 &\rightarrow Q_2 \\ 2, 4 &\rightarrow Q_3 \\ 6 &\rightarrow Q_4 \\ 9 &\rightarrow Q_5 \end{aligned}$$

c) Sistemul minim  $S_e$  echivalent cu  $S$  va avea deci 5 stări  $Q_1-Q_5$  și are tabelul de tranziție definit astfel :

		$X$			
		0	1	2	
$Q_e$	1, 3, 8	$Q_1$	24/1	24/0	57/0
	2, 4	$Q_2$	13/0	24/1	24/0
	5, 7	$Q_3$	6/1	24/0	38/0
	6	$Q_4$	8/0	9/1	6/1
	9	$Q_5$	7/0	9/1	7/1

Înlocuind în tabel perechile de stări cu clasele de echivalență cărora le aparțin rezultă sistemul echivalent cu  $S$  definit de:

$Q_i \backslash X$	0	1	2
$Q_1$	$Q_2/1$	$Q_3/0$	$Q_3/0$
$Q_2$	$Q_1/0$	$Q_2/0$	$Q_2/1$
$Q_3$	$Q_4/1$	$Q_2/0$	$Q_1/0$
$Q_4$	$Q_1/0$	$Q_5/1$	$Q_4/1$
$Q_5$	$Q_3/0$	$Q_5/1$	$Q_3/1$

**Problema 1.12.** Se dă sistemul  $S$  definit de graful din figura 1.34.

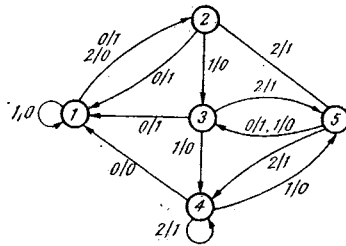


Fig. 1.34.

- Să se determine tabelul de tranziții;
- Să se reducă spațiul stărilor;
- Să se construiască tabelul de tranziții și graful de tranziții al sistemului minim echivalent cu  $S$ .

**Problema 1.13.** Să se demonstreze că algebra booleană este o latică distributivă și complementată (cu element unitar).

*Rezolvare*

În demonstrație se pornește de la postulatele algebrei booleene în care sînt definite două legi de compoziție internă (suma și produsul) pe mulțimea variabilelor de intrare cu proprietățile:

- asociativitate  $A + (B + C) = (A + B) + C$  și  $A \cdot (BC) = (AB) \cdot C$
- comutativitate  $A + B = B + A$  și  $A \cdot B = B \cdot A$
- absorbție  $A + AB = A$  și  $A \cdot (A + B) = A$
- idempotență  $A + A = A$  și  $AA = A$

Aceste postulate definesc algebra booleană ca o latică. Aceasta este distributivă deoarece există și postulatele de:

- distributivitate la stînga  $(A + B)C = AC + BC$
- distributivitate la dreapta  $A + BC = (A + B) \cdot (A + C)$

Algebra booleană este de asemenea o latice cu element unitar, fapt ce rezultă din :

- principiul terțiului exclus  $A + \bar{A} = 1$
- principiul contradicției  $A\bar{A} = 0$ .

Deci elementul unitar este 1 pentru sumă și 0 pentru produs.

**Problema 1.14.** Să se demonstreze că algebra booleană este un grup abelian față de operația de sumă și produs.

*Indicație*

Știind că grupul este o structură algebrică în care se definește o lege de compoziție internă (\*) cu proprietățile :

- asociativitate  $a*(b*c) = (a*b)*c$
- element neutru  $e : a*e = e*a = a$
- element invers  $a*\bar{a} = e$

și că grupul este abelian (comutativ) dacă  $a*b = b*a$ , se caută printre postulatele algebrei booleene acelea ce-i conferă structură de grup față de operațiile de sumă și produs.

Se observă că cele două grupuri au elemente neutre diferite : 1 pentru legea de compoziție internă sumă și 0 pentru legea de compoziție internă de tip produs.

Trebuie remarcată dualitatea celor două grupuri ; făcând substituțiile  $(+) \leftrightarrow (\cdot) \quad A \leftrightarrow \bar{A} \quad 0 \leftrightarrow 1$ , se pot face translații între relațiile scrise în cadrul fiecărui grup.

**Problema 1.15.** Pornind de la postulatele algebrei booleene să se demonstreze teoremele lui De Morgan :

$$A + B = \overline{\bar{A} \cdot \bar{B}}$$

$$A \cdot B = \overline{\bar{A} + \bar{B}}$$

**Problema 1.16.** Să se demonstreze că :

$$A + \bar{A}B = A + B$$

$$\bar{A} + AB = \bar{A} + B$$

$$\bar{A} + A\bar{B} = \bar{A} + \bar{B}$$

$$A + \bar{A}\bar{B} = A + \bar{B}$$

*Rezolvare*

Se consideră prima relație. O metodă de demonstrare este aceea de a construi tabelul de adevăr pentru ambii membri ai identității observând că pentru o anumită configurație a variabilelor de intrare  $A$  și  $B$  se obțin aceleași valori.

A	B	$\overline{AB}$	$A + \overline{AB}$	$A + B$
0	0	0	0	0
0	1	1	1	1
1	0	0	1	1
1	1	0	1	1

Celelalte relații se obțin din prima prin permutări.

Un alt mod de rezolvare, utilizând postulatul de distributivitate la dreapta este următorul :

$$A + \overline{A}B = (A + \overline{A})(A + B) = A + B$$

**Problema 1.17.** Să se verifice identitățile :

$$AB + AC + B\overline{C} = AC + B\overline{C}$$

$$AB + \overline{B}C + A\overline{C} = AB + \overline{B}C$$

utilizând algebra booleană.

*Rezolvare*

$$\begin{aligned} AB(\overline{C} + C) + AC + B\overline{C} &= AB\overline{C} + (ABC + AC) + B\overline{C} = \\ &= B\overline{C}(A + 1) + AC(B + 1) = AC + B\overline{C} \end{aligned}$$

În cazul celei de a doua identități se multiplică un termen cu  $B + \overline{B} = 1$ .

**Problema 1.18.** Să se verifice identitățile :

$$\overline{\overline{A + B} + \overline{A} + \overline{B}} = A$$

$$\overline{AB + BC + CA} = \overline{A}B + \overline{B}C + \overline{C}A$$

$$(A + B)(B + C)(C + A) = AB + BC + CA$$

$$(A + B)(\overline{A} + C) = AC + \overline{A}B$$

utilizând algebra booleană și metoda tabelului de adevăr.

**Problema 1.19.** Să se demonstreze pornind de la postulatele algebrei booleene comutativitatea funcției SAU EXCLUSIV (XOR).

*Rezolvare*

$$\text{Se știe că } A \oplus B = A\overline{B} + \overline{A}B$$

$$\text{Deci } B \oplus A = B\overline{A} + \overline{B}A \quad \text{a) } \overline{A}B + A\overline{B} \quad \text{b) } A\overline{B} + \overline{A}B \text{ q.e.d.}$$

a) conform postulatalui de comutativitate a funcției ȘI (produs);

b) se aplică proprietatea de comutativitate a funcției SAU (suma).

**Problema 1.20.** Să se verifice identitatea (asociativitatea funcției SAU — EXCLUSIV (XOR)) :

$$Y = (A \oplus B) \oplus C = A \oplus (B \oplus C)$$

Să se verifice că  $Y = 1$  (0) dacă un număr impar (par) de variabile este egal cu 1. (Acest rezultat este valabil pentru orice număr de intrări și este folosit la construirea detectorului de paritate).

**Problema 1.21.** Să se demonstreze că :

$$\begin{aligned} A \oplus B &= \overline{A \oplus \overline{B}} = \overline{\overline{A} \oplus B} \\ A\overline{B} + \overline{A}B &= AB + \overline{A}\overline{B} \\ A \oplus 1 &= \overline{A}; A \oplus 0 = A \\ A \oplus \overline{A} &= 1; A \oplus A = 0 \\ A \oplus B &= B \oplus A = \overline{A} \oplus \overline{B} \\ (A \oplus B) + C &\neq (A + C) \oplus (B + C) \\ A \cdot (B \oplus C) &\neq AB \oplus AC \end{aligned}$$

**Problema 1.22.** Să se demonstreze că :

$$(A + B)(A + C) = A + BC$$

*Indicație*

Se aplică postulatele distributivității, absorbției și idempotenței.

**Problema 1.23.** Complementați funcția următoare :

$$f = A \cdot [(B\overline{C} + D) + E]$$

*Rezolvare*

Se aplică teoremele lui De Morgan :

$$\begin{aligned} \overline{f} &= \overline{A(B\overline{C} + D) + E} = \overline{A} + \overline{(B\overline{C} + D)} + \overline{E} = \overline{A} + \overline{B\overline{C} + D} \cdot \overline{E} = \\ &= \overline{A} + \overline{B\overline{C}} \cdot \overline{D} \cdot \overline{E} = \overline{A} + (\overline{B} + C) \cdot \overline{D} \cdot \overline{E} \end{aligned}$$

**Problema 1.24.** Să se exprime :

$$f = AB + A\overline{C} + \overline{A}\overline{B}C$$

cu ajutorul funcției ȘI NU (NAND)

*Rezolvare*

Se aplică teorema lui De Morgan :

$$f = \overline{\overline{AB} \cdot \overline{A\overline{C}} \cdot \overline{\overline{A}\overline{B}C}}$$

Se observă că s-au operat trei modificări: s-a înlocuit operația + cu ·, s-au negat variabilele  $AB$ ,  $A\bar{C}$ ,  $\bar{A}\bar{B}C$  și s-a negat întreaga expresie.

**Problema 1.25.** Să se exprime :

$$f = \overline{AB + \bar{A}\bar{B}C + \bar{B}(A + C)}$$

cu ajutorul funcției SAU NU (NOR).

**Problema 1.26.** Să se exprime :

$$f = A(\bar{B} + \bar{C}) + (\bar{B} + \bar{D})(\bar{A} + \bar{C}) + \bar{D}$$

cu ajutorul funcției ȘI NU (NAND).

**Problema 1.27.** Să se demonstreze utilizând algebra booleană că :

$$\bar{A}BC + A\bar{B}C + ABC = (A + B)C$$

**Problema 1.28.** Să se prezinte formele de exprimare a unei funcții booleene.

*Rezolvare*

Se consideră spre exemplificare familia de funcții booleene de 3 variabile ( $A$ ,  $B$ ,  $C$ ).

I. Reprezentarea cu ajutorul *tabelului de adevăr*.

	A	B	C	f
0	0	0	0	$c_0$
1	0	0	1	$c_1$
2	0	1	0	$c_2$
3	0	1	1	$c_3$
4	1	0	0	$c_4$
5	1	0	1	$c_5$
6	1	1	0	$c_6$
7	1	1	1	$c_7$

În stînga acestuia sînt întabulate ordonat toate configurațiile de intrare posibile, iar în dreapta este notat vectorul  $a$

$$a = \{a_0, a_1, \dots, a_7\}$$

care particularizează funcția  $f$ . În acest caz există  $2^8$  funcții posibil de obținut cu 3 variabile de intrare. În general numărul funcțiilor este  $2^{2^N}$  unde  $N$  este numărul variabilelor de intrare.



II. *Forma canonică normală disjunctivă, fcnđ* (sumă de termeni de tip produs,  $P$ ):

$$f = a_0P_0 + a_1P_1 + \dots + a_7P_7$$

unde:  $P_0 = \bar{A}\bar{B}\bar{C} = 0$ ,  $P_1 = \bar{A}\bar{B}C = 1$ ,  $P_2 = \bar{A}B\bar{C} = 2 \dots P_7 = ABC = 7$  reprezintă produsele ordonate ale variabilelor de intrare. Pentru comoditate uneori se reprezintă termenii canonici prin numerele zecimale asociate ( $A-2^2$ ,  $B-2^1$ ,  $C-2^0$ ).

III. *Forma canonică normală conjunctivă, fcnđ* (produs de termeni de tip sumă,  $S$ ):

$$f = \bar{a}_0S_0 \cdot \bar{a}_1S_1 \cdot \bar{a}_2S_2 \dots \bar{a}_7S_7$$

unde:  $S_0 = A + B + C$ ,  $S_1 = A + B + \bar{C}$ ,  $S_2 = A + \bar{B} + C, \dots S_7 = \bar{A} + \bar{B} + \bar{C}$

Se observă că spre deosebire de *fcnđ* unde expresia funcției conține termenii canonici pentru care coeficienții vectoriali  $a_k$  sînt 1, la *fcnc* intervin în expresia funcției acei termeni  $S_k$  pentru care  $a_k$  sînt 0.

IV. *Forma elementară*. La formele canonice, termenii conțineau toate variabilele de intrare, negate sau nenegate. Termenii formei elementare nu conțin toate variabilele de intrare. Această formă se obține din cele canonice prin minimizare.

V. *Forma neelementară*. Conține negări de mai multe variabile, incluziuni de paranteze. Se obține din celelalte forme prin aplicarea teoremelor algebrei booleene, fiind utilizată în implementarea funcțiilor logice.

VI. *Diagrama Veitch-Karnaugh*. Constituie o reprezentare grafică a formelor canonice. Elementele mulțimii de intrare sînt reprezentate prin suprafețe dreptunghiulare din intersectarea cărora rezultă termenii canonici.

În figura 1.35 sînt reprezentate diagramele pentru funcții de 2, 3 și 4 variabile.

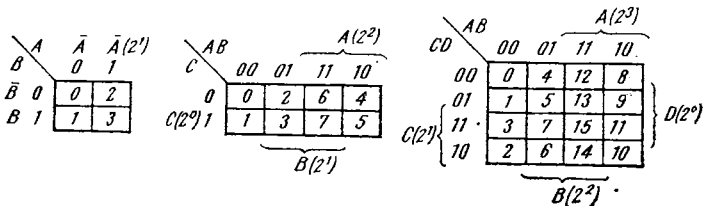


Fig. 1.35.

Diagrama Veitch-Karnaugh pentru funcții de 5 variabile (fig. 1.36) este formată din două diagrame pentru 4 variabile, una pentru  $A$  și alta pentru  $\bar{A}$ .

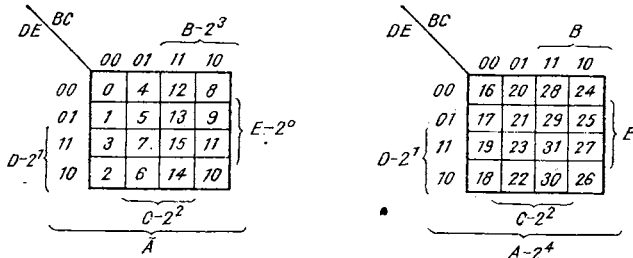


Fig. 1.36.

Pentru 6 variabile reprezentarea va conține 4 diagrame de 4 variabile ( $CDEF$ ) câte una pentru fiecare suprafață  $\bar{A}\bar{B}$ ,  $\bar{A}B$ ,  $A\bar{B}$  și respectiv  $AB$  (fig. 1.37).

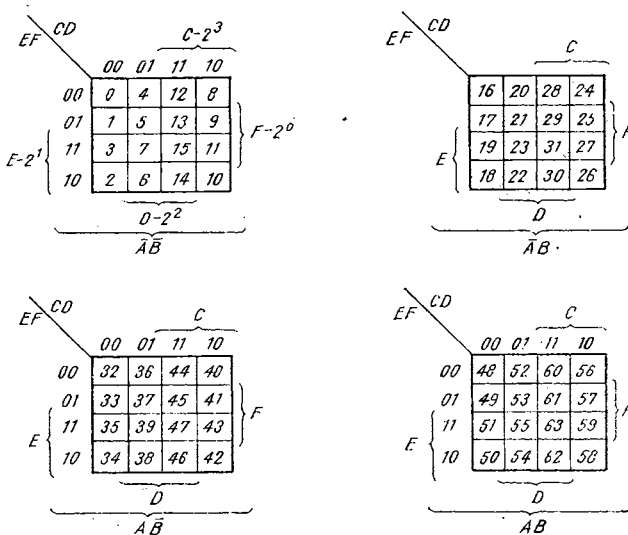


Fig. 1.37.

Particularizarea funcțiilor se face prin notarea în diagramă a coeficienților  $a_k$  corespunzători fiecărui termen canonic.

**Problema 1.29.** Să se exprime funcția  $f$  dată prin tabel de adevăr, cu  $fcnd$ ,  $fcnc$  și diagrame Veitch.

$\bullet$	$A$	$B$	$C$	$D$	$f$
0	0	0	0	0	0
1	0	0	0	1	1
2	0	0	1	0	1
3	0	0	1	1	0
4	0	1	0	0	1
5	0	1	0	1	1
6	0	1	1	0	0
7	0	1	1	1	1
8	1	0	0	0	1
9	1	0	0	1	1
10	1	0	1	0	1
11	1	0	1	1	0
12	1	1	0	0	0
13	1	1	0	1	0
14	1	1	1	0	0
15	1	1	1	1	1

*Rezolvare*

$$\begin{aligned}
 fcnd : f &= P_1 + P_2 + P_4 + P_5 + P_7 + P_8 + P_9 + P_{10} + P_{15} = \\
 &= \bar{A}\bar{B}CD + \bar{A}\bar{B}C\bar{D} + \bar{A}\bar{B}C\bar{D} + \bar{A}\bar{B}CD + \bar{A}BCD + \bar{A}\bar{B}C\bar{D} + \\
 &\quad + \bar{A}\bar{B}CD + \bar{A}\bar{B}C\bar{D} + ABCD
 \end{aligned}$$

$$\begin{aligned}
 fcnc : f &= S_0 \cdot S_3 \cdot S_8 \cdot S_{11} \cdot S_{12} \cdot S_{13} \cdot S_{14} = (A + B + C + D)(A + B + \\
 &+ \bar{C} + \bar{D})(A + \bar{B} + \bar{C} + D)(\bar{A} + B + \bar{C} + \bar{D})(\bar{A} + \bar{B} + C + D) \cdot \\
 &\cdot (\bar{A} + \bar{B} + C + \bar{D})(\bar{A} + \bar{B} + \bar{C} + D).
 \end{aligned}$$

Diagramele Veitch sînt date în figura 1.38.

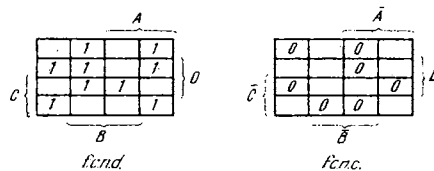


Fig. 1.38.

**Problema 1.30.** Să se exprime :

$$\begin{aligned}
 f_1 &= P_0 + P_1 + P_5 + P_7 \\
 f_2 &= P_3 + P_{21} + P_{24} + P_{29} + P_{30}
 \end{aligned}$$

prin tabel de adevăr,  $fcnc$  și diagrame Veitch.

**Problema 1.31.** Să se exprime funcția  $f$  dată în formă elementară cu  $fcnd$ ,  $fcnc$  și tabel de adevăr :

$$f = AC + B\bar{C}$$

*Indicație*

Se dezvoltă expresia funcției pentru a o aduce la  $fcnd$ .

$$f = AC(B) + (AC\bar{B}) + (A)B\bar{C} + (\bar{A}B\bar{C}) = ABC + A\bar{B}C + AB\bar{C} + \bar{A}B\bar{C} = P_7 + P_5 + P_6 + P_2$$

Se observă că se adaugă variabila care lipsește, respectiv  $B$  la termenul  $AC$  și încă un termen cu aceeași variabilă negată ( $AC\bar{B}$ ).

**Problema 1.32.** Să se exprime funcția :

$$f = AC + BD$$

prin  $fcnd$ , tabel de adevăr, diagrame Veitch-Karnaugh.

*Indicație*

Funcția este dată în formă elementară. Trebuie dezvoltată pentru a o aduce la  $fcnd$ .

$$f = AC(D) + (AC\bar{D}) + (A)BD + (\bar{A}BD) = A(B)CD + (A\bar{B}CD) + A(B)C\bar{D} + (A\bar{B}C\bar{D}) + AB(C)D + (A\bar{B}C\bar{D}) + \bar{A}B(C)D + (\bar{A}B\bar{C}D) = P_{15} + P_{11} + P_{14} + P_{10} + P_{13} + P_7 + P_5$$

În paranteze sînt trecute variabilele sau termenii care se adaugă expresiei dinainte de semnul egal.

**Problema 1.33.** Să se exprime funcția :

$$f = B\bar{C} + AD + \bar{A}C$$

prin  $fcnd$ , tabel de adevăr, diagrame Veitch-Karnaugh și  $fcnc$ .

**Problema 1.34.** Să se exprime funcția :

$$f = \overline{AB \cdot C}$$

prin  $fcnd$ , tabel de adevăr,  $fcnc$  și diagrame Veitch-Karnaugh.

*Indicație*

Funcția este dată în forma neelementară.

Se aplică teorema lui De Morgan și se dezvoltă pentru a obține termenii canonici disjunctivi

$$f = AB + \bar{C} = AB(C) + (AB\bar{C}) + (AB)\bar{C} + (\bar{A}\bar{B})\bar{C} + (\bar{A}B)\bar{C} + (A\bar{B})\bar{C}$$

**Problema 1.35.** Să se minimizeze funcția :

$$f = P_3 + P_7 + P_8 + P_9 + P_{12} + P_{13} + P_{15}$$

## Rezolvare

Minimizarea constă în trecerea de la forma canonică la o formă elementară de exprimare a funcției, deci la eliminarea unor variabile de intrare din termenii funcției.

Pentru a obține forma minimă se consideră diagramele Veitch pentru funcție și negata ei, exprimate prin *f*cn.d și *f*cn.d. Deci se vor lua în considerare patru diagrame.

Pentru minimizare se aleg suprafețele maxime formate din constituenții 1-rilor, respectiv 0-urilor, suprafețe avînd ca dimensiune un număr de pătrate egal întotdeauna cu puteri ale lui 2. Aceste suprafețe corespund termenilor elementari (implicanți primi), iar reprezentarea geometrică este de fapt identică cu aplicarea teoremei:

$$AB + A\bar{B} = A$$

Dintre cele 6 funcții minime găsite (fig.1.39 și 1.40) se alege cea care la implementarea cu circuitele disponibile, conduce la numărul cel mai mic de CI.

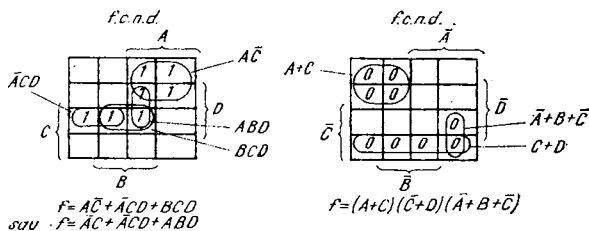


Fig. 1.39.

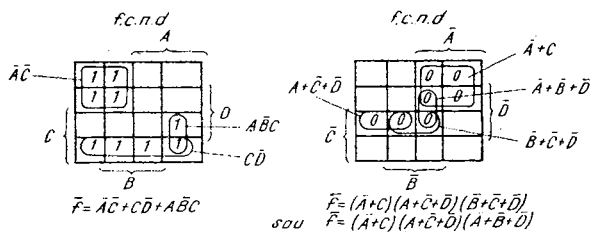


Fig. 1.40.

**Problema 1.36.** Să se minimizeze funcția:

$$f = \bar{A}\bar{B}\bar{D} + A\bar{B}\bar{C}\bar{D} + ABC\bar{D}$$

- cu diagrame Veitch-Karnaugh;
- utilizînd teoremele algebrei booleene.

*Rezolvare*

a) Funcția trebuie adusă la forma canonică.

$$f = \overline{A}B(C)\overline{D} + (\overline{A}B\overline{C}\overline{D}) + A\overline{B}\overline{C}\overline{D} + ABC\overline{D} = P_3 + P_4 + P_{12} + P_{14}$$

Diagrama Veitch-Karnaugh pentru  $f$  este dată în figura 1.41.

Suprafața maximă ce acoperă 1-urile este  $B\overline{D}$ , cu dimensiunea  $2 \times 2$  pătrate deoarece lateralele diagramei sînt adiacente. Într-adevăr redesenînd diagrama pentru  $f$  cu o altă dispunere a suprafețelor ca în figura 1.42 se observă că suprafața  $B\overline{D}$  reprezentînd funcția minimă este plasată în centru.

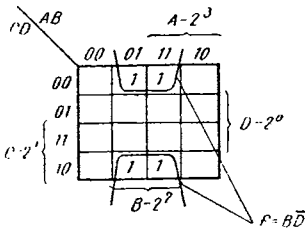


Fig. 1.41.

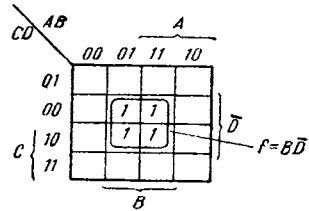


Fig. 1.42.

b) În ultimii doi termeni se poate ca factor  $A\overline{B}\overline{D}$  și apoi,  $B\overline{D}$  în expresia care rezultă :

$$f = \overline{A}B\overline{D} + A\overline{B}\overline{D}(\overline{C} + C) = (A + \overline{A})B\overline{D} = B\overline{D}$$

Se observă că funcția  $f$  se minimizează mai ușor utilizînd algebra booleană. Dar pentru majoritatea funcțiilor minimizarea cu diagrame Veitch-Karnaugh este mai simplă.

**Problema 1.37.** Să se minimizeze funcția :

$$f = P_0 + P_2 + P_3 + P_{10}$$

utilizînd diagrama Veitch-Karnaugh pentru  $f$ .

*Rezolvare*

Construind diagrama Veitch-Karnaugh (fig. 1.43) pentru  $f$  în  $f$  se constată că termenii canonici sînt plasați în cele 4 colțuri, și pentru minimizare pot fi grupați într-o singură suprafață  $\overline{B}\overline{D}$ , cu dimensiunea  $2 \times 2$  pătrate. Aceasta datorită adiacenței lateralelor diagramei (vezi și problema anterioară).

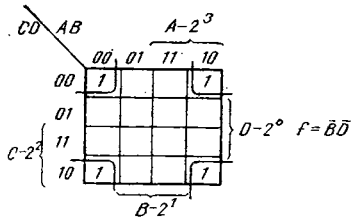


Fig. 1.43.

**Problema 1.38.** Să se minimizeze cu diagrama Veitch-Karnauch pentru  $f$  funcția de 5 variabile

$$f = P_0 + P_2 + P_3 + P_4 + P_6 + P_{14} + P_{16} + P_{18} + P_{19} + \\ + P_{20} + P_{22} + P_{24} + P_{26} + P_{30}$$

*Rezolvare*

Se reprezintă  $f$  prin diagramele Veitch-Karnaugh (fig. 1.44) și se observă că în dreapta se obțin suprafețele maxime  $ADE$  (linia de jos) și  $ACE$  (cele 4 colțuri).

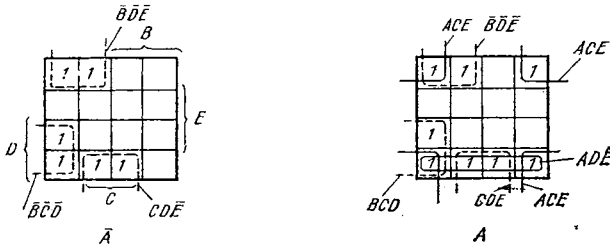


Fig. 1.44.

În rest suprafețe maxime se pot obține doar ținând cont de adiacența celor două diagrame:  $BDE$ ,  $BCD$ ,  $CDE$ . Deci :

$$f = ADE + ACE + BDE + BCD + CDE$$

**Problema 1.39.** Să se minimizeze funcțiile logice asociate unui decodificator  $BCD/7$  segmente utilizând termenii canonici redondanți.

*Indicație*

Cu cele 7 segmente  $a \div g$  se pot sintetiza toate cifrele între 0 și 9 ca în figura 1.45.



Fig. 1.45.

Deci tabelul de adevăr pentru funcțiile  $a-g$  va fi :

	$a$	$b$	$c$	$d$	$e$	$f$	$g$
0	1	1	1	1	1	1	0
1	0	1	1	0	0	0	0
2	1	1	0	1	1	0	1
3	1	1	1	1	0	0	1
4	0	1	1	0	0	1	1
5	1	0	1	1	0	1	1
6	0	0	1	1	1	1	1
7	1	1	1	0	0	0	0
8	1	1	1	1	1	1	1
9	1	1	1	0	0	1	1
10	x	x	x	x	x	x	x
11	x	x	x	x	x	x	x
12	x	x	x	x	x	x	x
13	x	x	x	x	x	x	x
14	x	x	x	x	x	x	x
15	x	x	x	x	x	x	x

Diagrama de minimizare pentru funcția  $a$  va fi cea din figura 1.46.

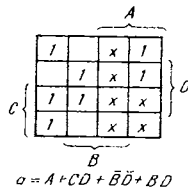


Fig. 1.46.

Termenii canonici redondanți sînt  $10 \div 15$  care nefiind semnificativi pentru sistem pot fi atît 0 cît și 1, deci pot contribui la completarea suprafețelor de minimizare.

În mod similar se procedează și pentru celelalte segmente.

Rezultatele sînt :

$$b = \bar{C}\bar{D} + CD + \bar{B}$$

$$c = B + \bar{C} + D$$



$$d = C\bar{D} + \bar{B}C + B\bar{C}D + \bar{B}\bar{D}$$

$$e = \bar{B}\bar{D} + C\bar{D}$$

$$f = \bar{C}\bar{D} + B\bar{C} + A + B\bar{D}$$

$$g = B\bar{D} + B\bar{C} + \bar{B}C + A$$

**Problema 1.40.** Să se minimizeze și să se exprime cu ajutorul funcției ȘI NU (NAND) următoarele :

$$f_1 = P_1 + P_2 + P_8 + P_{13}$$

$$f_2 = P_0 + P_1 + P_2 + P_6 + P_8$$

$$f_3 = P_5 + P_7 + P_9 + P_{10} + P_{14}$$

$$f_4 = P_0 + P_2 + P_4$$

**Problema 1.41.** Să se minimizeze și să se exprime cu ajutorul funcției ȘI NU (NAND) următoarele funcții reprezentînd circuitul logic combinațional al unui numărător sincron realizat cu bistabili JK :

$$f_1 = P_7, \text{ termeni redondanți } P_5 \div P_8$$

$$f_2 = P_7, \text{ termeni redondanți } P_0 \div P_6$$

$$f_3 = P_3 + P_{11}, \text{ termeni redondanți } P_4 \div P_7 \text{ și } P_{12} \div P_{15}$$

$$f_4 = P_7 + P_{15}, \text{ termeni redondanți } P_0 \div P_3 \text{ și } P_8 \div P_{11}$$

$$f_5 = P_1 + P_5 + P_9 + P_{13}, \text{ t.r. } P_2, P_3, P_6, P_7, P_{10}, P_{11}, P_{14}, P_{15}$$

$$f_6 = P_3 + P_7 + P_{11} + P_{15}, \text{ t.r. } P_1, P_5, P_9, P_{13}$$

$$f_7 = P_0 + P_2 + P_4 + P_6 + P_8 + P_{10} + P_{12} + P_{14}, \text{ t.r. } P_1, P_3, P_5, P_7, P_9, P_{11}, P_{13}, P_{15}$$

$$f_8 = P_1 + P_3 + P_5 + P_7 + P_9 + P_{11} + P_{13} + P_{15}, \text{ t.r. } P_0, P_2, P_4, P_6, P_8, P_{10}, P_{12}, P_{14}$$

**Problema 1.42.** Să se minimizeze și să se exprime cu ajutorul funcției ȘI NU (NAND) următoarele funcții reprezentînd circuitul logic combinațional al unui numărător sincron realizat cu bistabili D.

$$f_1 = P_8 + P_9 + P_{12} + P_{13}$$

$$f_2 = P_3 + P_4 + P_5 + P_6 + P_{12} + P_{13} + P_{14}$$

$$f_3 = P_1 + P_2 + P_5 + P_6 + P_9 + P_{10} + P_{13} + P_{14}$$

$$f_4 = P_0 + P_2 + P_4 + P_6 + P_8 + P_{10} + P_{12} + P_{14}$$

**Problema 1.43.** Să se minimizeze cu metoda Quine-Mc Cluskey următoarea funcție :

$$f = P_0 + P_2 + P_3 + P_5 + P_7 + P_8 + P_{10} + P_{11} + P_{13} + P_{15}$$

### Rezolvare

Metoda Quine-Mc Cluskey este mai laborioasă decât cea a diagramei Veitch-Karnaugh; se utilizează în special în cazul funcțiilor cu mai mult de 6 variabile. Are avantajul că algoritmul metodei este ușor programabil pe calculator. Minimizarea cu această metodă conține următoarele etape:

I. Se grupează termenii canonici astfel încât termenii din fiecare grupă să conțină același număr de 1-uri, respectiv 0-uri.

	A	B	C	D
0	0	0	0	0
2	0	0	1	0
8	1	0	0	0
3	0	0	1	1
5	0	1	0	1
10	1	0	1	0
7	0	1	1	1
11	1	0	1	1
13	1	1	0	1
15	1	1	1	1

II. Se compară fiecare termen dintr-o grupă cu toți cei din grupa următoare, aplicând relația de reducere  $AB + A\bar{B} = A$ .

Com- parare între	Rezultatul comparării			
	A	B	C	D
0, 2	0	0	—	0
0, 8	—	0	0	0
2, 3	0	0	1	—
2, 10	—	0	1	0
8, 10	1	0	—	0
3, 7	0	—	1	1
3, 11	—	0	1	1
5, 7	0	1	—	1
5, 13	—	1	0	1
10, 11	1	0	1	—
7, 15	—	1	1	1
11, 15	1	—	1	1
13, 15	1	1	—	1

În continuare se repetă procedeul pînă cînd nu mai este posibilă nici o reducere.

Comparare între	Rezultatul comparării			
	A	B	C	D
0, 2, 8, 10	—	0	—	0
2, 3, 10, 11		0	1	—
3, 7, 11, 15	—	—	1	1
5, 7, 13, 15	—	1	—	1

Termenii rezultați (0, 2, 8, 10), (2, 3, 10, 11), (3, 7, 11, 15) și (5, 7, 13, 15) se numesc implicați primi, *IP*.

III. Se aleg acei implicați primi care asigură acoperirea minimală a termenilor canonici, *TC*. Aceasta se face cu ajutorul unui tabel de acoperire (fig. 1.47); în care pe colțane se notează *TC* iar pe linii *IP*.

<i>IP</i> \ <i>TC</i>	0	2	3	5	7	8	10	11	13	15
0, 2, 8, 10	x	x				x	x			
2, 3, 10, 11		x	x				x	x		
3, 7, 11, 15			x		x			x		x
5, 7, 13, 15				x	x				x	x

Fig. 1.47.

În intersecții se notează acei *TC* acoperiți de fiecare *IP*. De exemplu *IP* (0, 2, 8, 10), avînd forma  $-0-0$  (adică  $\bar{B}\bar{D}$ ) acoperă *TC* 0 și 8. Într-adevăr pentru demonstrație se extinde  $\bar{B}\bar{D}$  folosind teoremele algebrei booleene :

$$\begin{aligned} \bar{B}\bar{D} &= (A + \bar{A})\bar{B}(C + \bar{C})\bar{D} = A\bar{B}C\bar{D} + \bar{A}\bar{B}C\bar{D} + \bar{A}\bar{B}\bar{C}\bar{D} + A\bar{B}\bar{C}\bar{D} = \\ &= 10 + 2 + 0 + 8 \end{aligned}$$

Se observă că un *TC* poate fi acoperit de mai multe linii. Se caută întii acei *TC* care sînt acoperiți de o singură linie. Evident *IP* respectiv va aparține soluției finale de minimizare. Astfel pentru asigurarea acoperirii pentru 5 și 13, *IP* (5, 7, 13, 15) va fi ales în soluție. La fel (0, 2, 8, 10) pentru a asigura acoperirea *TC* 0 și 8. Se observă că mai rămîn neacoperite coloanele 3 și 11. Pentru acoperire poate fi ales oricare din cei doi *IP* rămași. Deci sînt două soluții de minimizare :

$$\begin{aligned} f &= (0, 2, 8, 10) + (5, 7, 13, 15) + (2, 3, 10, 11) = \bar{B}\bar{D} + BD + \bar{B}C \\ f &= (0, 2, 8, 10) + (5, 7, 13, 15) + (3, 7, 11, 15) = \bar{B}\bar{D} + BD + CD \end{aligned}$$

**Problema 1.44.** Utilizând metoda Quine-Mc Cluskey să se minimizeze :

$$f_1 = P_0 + P_2 + P_4 + P_6$$

$$f_2 = P_0 + P_1 + P_4 + P_5 + P_{12} + P_{13}$$

$$f_3 = P_0 + P_4 + P_{18} + P_{19} + P_{22} + P_{23} + P_{25} + P_{29}$$

**Problema 1.45.** Să se reducă cu metoda Quine-Mc Cluskey funcția :

$$f = P_0 + P_2 + P_4 + P_6 + P_7 + P_8 + P_{10} + P_{11} + P_{12} + P_{13} + P_{14} + P_{16} + P_{18} + P_{19} + P_{29} + P_{30}$$

*Soluție*

$$f = \bar{A}\bar{E} + \bar{B}\bar{C}\bar{E} + BC\bar{D}\bar{E} + BC\bar{D}E + \bar{A}\bar{B}\bar{C}D + \bar{A}\bar{B}CD \quad (E - 2^\circ)$$

**Problema 1.46.** Să se reducă utilizând metoda Quine-Mc Cluskey :

$$f = P_0 + P_1 + P_2 + P_3 + P_7 + P_{14} + P_{15} + P_{22} + P_{23} + P_{29} + P_{31}$$

*Indicație*

Cînd trebuie ales între doi *IP* pentru a obține acoperirea totală a *TC* se alege acel *IP* care acoperă cît mai mulți *TC*, deoarece astfel se asigură soluția cu număr minim de variabile de intrare.

*Soluție*

$$f = \bar{A}\bar{B}\bar{C} + CDE + ABCE + \bar{A}\bar{B}CD + \bar{A}BCD \quad (E \rightarrow 2^\circ)$$

**Problema 1.47.** Să se minimizeze utilizînd o metodă la alegere următoarele funcții :

$$f_1 = P_5 + P_{10} + P_{11} + P_{30} + P_{31}$$

$$f_2 = P_1 + P_2 + P_5 + P_7 + P_{15}$$

### Prezentarea conținutului

În acest capitol se demonstrează că metodele matematice prezentate anterior sînt insuficiente pentru proiectarea cu circuite integrate. Unul din scopurile importante ale acesteia este obținerea unor scheme cu număr minim de circuite integrate și cu un cost cît mai scăzut.

Abordînd din acest punct de vedere problema implementării se impune cu necesitate parcurgerea a două etape :

- cunoașterea tipurilor de circuite integrate combinaționale ;
- stăpînirea metodelor specifice de proiectare cu aceste circuite și a criteriilor de alegere a acelor care conduc la o schemă minimă.

Există mai multe metode de implementare a CLC, diferențiate după nivelul de complexitate al CI utilizate :

1. **Implementarea cu CI SSI** (*Small Scale Integration* — Integrare pe scară redusă, cu sub 50 de tranzistoare integrate). Din această categorie fac parte diferitele tipuri de porți : ȘI NU (NAND), sau NU (NOR), SAU EXCLUSIV (XOR), ȘI SAU NU (AOI).

Pentru implementarea unei singure funcții logice este necesară minimizarea matematică a acesteia. În problema 2.1 se demonstrează că, prin realizarea unei funcții, dată în formă elementară, cu diverse tipuri de porți, se obține un număr variabil de circuite integrate. Înseamnă că, ulterior minimizării matematice este necesară alegerea celui tip de poartă care conduce la schema cea mai redusă ca dimen-

siuni. Această etapă în proiectare nu mai este formalizată matematic și pentru parcurgerea ei cât mai rapidă este necesară formarea unor deprinderi, a unei experiențe.

În problemele de la începutul capitolului se prezintă modalitățile de implementare a unei funcții logice, exprimată într-una din formele descrise în capitolul 1 (vezi problemele 1.28—1.34) :

— cu porți ȘI NU (NAND) — problemele 2.1—2.8, 2.16—2.17 și 2.67—2.70 ;

— cu porți SAU NU (NOR) — problemele 2.1 și 2.9—2.10 ;

— cu porți SAU EXCLUSIV (XOR) — problemele 2.1 și 2.14 ;

— cu porți ȘI SAU NU (AOI) — problemele 2.1 și 2.15—2.20.

Se propun de asemenea și probleme de analiză : pornind de la schema concretă să se deducă funcțiile logice (vezi 2.11—2.13, 2.16 și 2.25—2.30).

În problemele 2.21—2.24 se implementează cu porți CLC-uri descrierea matematică și cea a implementării este și mai ambiguă. Se arată că se poate obține o schemă mai mică nereducându-se total funcțiile, ci punind în evidență termeni comuni.

**2. Implementarea cu CI MSI** (*Medium Scale Integration* — Integrare pe scară medie — cu 50—500 tranzistoare integrate), utilizează circuite universale ca : decodificatorul și multiplexorul care generează în interior toți termenii canonici pentru 2, 3 respectiv 4 variabile de intrare.

DCD transmite acești termeni în exterior și sînt adunați în porți corespunzător cu funcțiile ce trebuie realizate (vezi problemele 2.31—2.52 și 2.67—2.70). MUX îi adună în interior, avînd intrări de validare ce permit numai transmiterea acelor care intervin în expresia funcției (vezi problemele 2.53—2.70).

În aceste cazuri, deoarece se pornește direct de la termenii canonici, minimizarea matematică a funcțiilor logice nu mai este necesară. Există metode specifice de reducere a schemelor, la MSI minimizarea implementării presupunînd pe lîngă obținerea unui număr mic de CI și utilizarea unor CI cu un grad mai scăzut de complexitate (vezi problemele 2.43 și 2.57—2.59). Din aceleași considerente, pentru implementarea unui CLC descris de un ansamblu de funcții logice se preferă soluția DCD + porți, deoarece, cea cu MUX conține un circuit MSI pentru fiecare funcție, fiind mai costisitoare chiar la același număr de CI.

În continuarea capitolului se studiază unele tipuri de CI MSI specializate, cum sînt :

— comparatoare (problemele 2.71—2.72) ;

— codificatoare cu priorități (problemele 2.73—2.79) ;

— sumatoare (problemele 2.92—2.93 și 2.97—2.99) ;

— unități logico-aritmetice (ALU) — (problemele 2.100—2.101).

De asemenea sînt tratate probleme de aritmetică binară și zecimală (vezi 2.87—2.102).

3. **Implementarea cu CI LSI** (*Large Scale Integration* — Integrare pe scară largă, cu peste 500 tranzistoare integrate) conține la rîndul său două metode specifice, diferențiate după tipul circuitelor utilizate :

**A. Implementarea cu ROM-uri** (2.103—2.127) se face conform celor arătate în rezolvarea problemei 2.115. În această secțiune sînt tratate aspecte importante legate de utilizarea memoriilor ROM ca :

- definirea memoriilor ROM, PROM, REPR0M (2.103—2.104) ;
- organizare internă, parametri ce definesc capacitatea și structura ROM-urilor (problemele 2.103 și 2.105—2.106) ;
- extinderea sau reducerea dimensiunilor memoriilor (2.107—2.114) ;
- exemple de proiectare utilizînd ROM-uri (2.115—2.127) ;
- minimizarea implementării cu ROM.

Referitor la acest ultim aspect trebuie remarcat faptul că, deoarece memoria ROM poate fi privită ca circuit universal de implementare cu DCD și porți, nu mai este necesară minimizarea matematică a funcțiilor logice. Minimizarea implementării cu ROM este independentă de aceasta și se realizează după următoarele criterii :

- ocuparea cît mai completă a memoriei (2.115—2.117) ;
- reducerea capacității ROM-ului prin codificare la intrare (2.123—2.125) sau decodificare la ieșire (2.126, 2.127). Este evident că reducerea ROM-ului operînd în spațiul de intrare este considerabil mai mare : se face cu puteri ale lui 2.

**B. Implementarea cu PLA-uri** (2.128—2.133) are particularitățile sale. În acest caz se revine la minimizarea matematică a funcțiilor logice, implementarea fiind similară cu cea a sistemelor de funcții combinaționale cu porți. PLA-ul este de fapt o structură universală, extinsă, de implementare cu 2 nivele de porți.

În concluzie trebuie subliniat că cele 2 metode de implementare cu SSI și MSI—DCD au fost generalizate, ajungîndu-se la structurile LSI corespondente, PLA-ul și respectiv ROM-ul.

**Problema 2.1.** Să se implementeze funcția :

$$f = AB + \bar{A}\bar{B}$$

- a) cu porți ȘI (AND), SAU (OR) și INVERSOARE ;
- b) cu porți ȘI NU (NAND) ;
- c) cu porți SAU NU (NOR) ;
- d) cu porți SAU EXCLUSIV (XOR) ;
- e) cu porți SI—SAU—NU (AOI) și INVERSOARE.

Care este relația între minimizarea matematică a funcției  $f$  și minimizarea implementării ?

*Rezolvare*

Prin implementarea unui sistem sau a unei funcții booleene se înțelege realizarea acestora cu circuite integrate, adică, proiectarea și desenaarea schemei cu simbolurile logice asociate circuitelor și calcularea numărului de circuite integrate necesare.

a) Implementarea cu porți ȘI (AND), SAU (OR) și INVERSOARE este dată în figura 2.1. Se desenează circuitele prin simboluri logice și, în dreptul fiecăruia se notează numărul asociat în cadrul seriei 74XXX. Frația dinaintea numărului arată cât reprezintă poarta respectivă în cadrul circuitului integrat. Se utilizează 1x7404, 1x7408 și 1x7432 deci, în total 3 integrate din care rămân însă disponibile 4 INVERSOARE, 2 ȘI-uri și 3 SAU-uri.

Se observă că acest tip de implementare este dezavantajos : se utilizează trei tipuri de CI și, în plus nu se poate aplica la funcții cu mai mulți termeni canonici deoarece nu există circuite integrate cu SAU-uri cu mai mult de 2 intrări.

b) Implementarea cu porți ȘI NU(NAND) are față de a) avantajul că, folosind un singur tip de CI, atinge un grad mai mare de utilizare a circuitelor integrate, conducând astfel la reducerea numărului acestora.

Pentru implementare se transformă expresia funcției  $f$  cu ajutorul teoremei lui De Morgan (vezi problemele 1.15, 1.24).

$$f = \overline{\overline{A} \cdot \overline{B}}$$

Schema este dată în figura 2.2. Se utilizează 2x7400 din care rămân libere 3 porți ȘI NU(NAND). O soluție mai simplă cu numai 1x.400 este prezentată în figura 2.8.

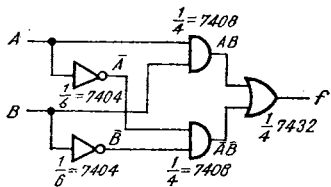


Fig. 2.1.

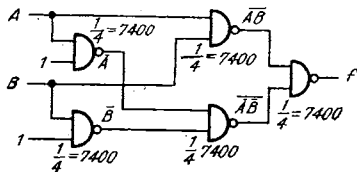


Fig. 2.2.

c) Implementarea cu SAU NU(NOR) asigură, de asemenea datorită uniformității, un număr redus de circuite integrate față de cazul a).



Expresia funcției  $f$  se transformă cu ajutorul teoremei lui De Morgan, înlocuind funcția ȘI :

$$f = \overline{\overline{A + B} + \overline{A + B}}$$

Implementarea este cea din figura 2.3 : se utilizează două circuite integrate,  $2 \times 7402$ , rămânând libere două porți.

d) Se observă că funcția  $f$  este chiar SAU EXCLUSIV negat. Într-adevăr :

$$\overline{A \oplus B} = \overline{A\overline{B} + \overline{A}B} = \overline{A\overline{B}} \cdot \overline{\overline{A}B} = (\overline{A} + B)(A + \overline{B}) = AB + \overline{A}\overline{B} = f$$

Deci implementarea (fig. 2.4) se va face cu un singur circuit integrat, două porți XOR rămânând neutilizate.

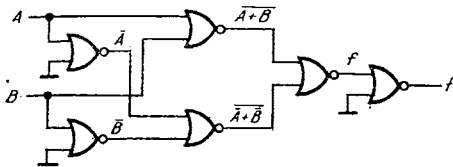


Fig. 2.3.

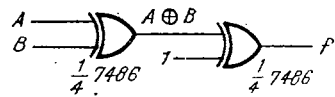


Fig. 2.4.

e) Sînt necesare două circuite integrate  $1 \times 7451$ ,  $1 \times 7404$  (fig. 2.5).

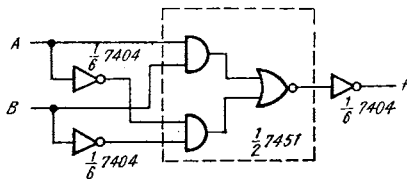


Fig. 2.5.

Se observă că deși funcția  $f$  este dată în formă elementară (deci minimizată), nu toate implementările a-:-e conduc la un număr minim de circuite integrate. Cu SAU EXCLUSIV implementarea se face cu o singur CI, în timp ce cu celelalte metode sînt necesare 2—3 circuite integrate.

Pentru *minimizarea implementării* unei funcții logice trebuie ca :

- să se utilizeze circuitele integrate cît mai puțin dispersate ca tipuri ;
- să se aleagă tipul de CI cel mai potrivit pentru implementare.

**Problema 2.2.** Să se implementeze cu porți ȘI NU (NAND) funcția logică :

$$f = P_0 + P_2 + P_4 + P_6 + P_9 + P_{11} + P_{13} + P_{15}$$

a) în *fncl* ;

b) în formă minimizată.

Să se compare cele două soluții ca număr de circuite integrate.

*Rezolvare*

a) Se aplică teorema lui De Morgan pentru a înlocui funcția SAU :

$$f = \overline{\overline{P_0} \cdot \overline{P_2} \cdot \overline{P_4} \cdot \overline{P_6} \cdot \overline{P_9} \cdot \overline{P_{11}} \cdot \overline{P_{13}} \cdot \overline{P_{15}}}$$

Termenii canonici negați se realizează cu ajutorul unor ȘI NU-uri (NAND) cu 4 intrări, iar  $f$  se obține prin cuplarea acestora la un ȘI NU(NAND) cu 8 intrări. Sînt necesare inversoare pentru obținerea variabilelor de intrare negate. Schema este dată în figura 2.6. Sînt necesare 5 circuite integrate, și anume  $1 \times 7430$ ,  $3 \times 7420$ ,  $1 \times 7404$ . În cazul în care inversarea se face tot cu ȘI NU-uri se înlocuiește 7404 cu 7400.

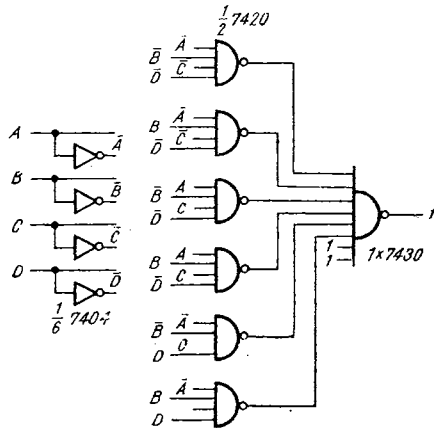


Fig. 2.6.

b) Prin minimizare (vezi problemele 1.35—1.37) funcția  $f$  devine:

$$f = \overline{\overline{\overline{A}B\overline{D}} + \overline{\overline{A}B\overline{D}} + \overline{\overline{A}B\overline{D}}} = \overline{\overline{\overline{A}B\overline{D}} \cdot \overline{\overline{A}B\overline{D}} \cdot \overline{\overline{A}B\overline{D}}}$$

Implementarea se va face doar cu 3 circuite integrate:  $2 \times 7410$   $1 \times 7404$ , din care rămân nefolosite 2 ȘI NU-uri și 3 inversoare (fig. 2.7).

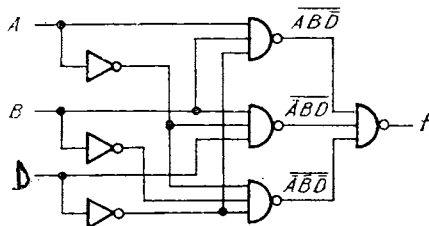


Fig. 2.7.

În concluzie implementarea funcției minimizele conduce la un număr mai redus de circuite integrate deoarece se reduce numărul variabilelor de intrare ( $C$  nu mai intervine), deci și numărul intrărilor în porți, iar porțile cu mai puține intrări sînt mai multe într-o capsulă.

**Problema 2.3.** Să se implementeze cu minimum de porți ȘI NU funcția

$$f = P_3 + P_7 + P_8 + P_9 + P_{12} + P_{13} + P_{15}$$

*Indicație*

Funcția este minimizată în problema 1.35. Din cele 6 funcții minime găsite :

$$f = A\bar{C} + \bar{A}CD + BCD \text{ sau } f = A\bar{C} + \bar{A}CD + ABD$$

se implementează minim cu  $1 \times 7410$  și  $1 \times 7400$ , rămînînd un ȘI NU cu două intrări nefolosit.

**Problema 2.4.** Să se implementeze cu minimum de porți ȘI NU funcția din problema 1.38.

*Soluție*

$$1 \times 7410, 1 \times 7420, 1 \times 7404$$

**Problema 2.5.** Să se implementeze cu minimum de porți ȘI NU (NAND) :

$$f = ABC\bar{C} + C\bar{D} + \bar{A}B + \bar{B}D + A\bar{B}\bar{C}$$

*Soluție*

$$1 \times 7400, 1 \times 7420, 1 \times 7404$$

**Problema 2.6.** Să se implementeze cu minimum de porți ȘI NU (NAND) un sumator binar complet de 2 biți.

*Indicație*

Tabelul de adevăr al sumatorului binar complet este :

$C^0$	A	B	S	$C^1$
0	0	0	0	0
1	0	0	1	0
2	0	1	0	1
3	0	1	1	0
4	1	0	0	1
5	1	0	1	0
6	1	1	0	1
7	1	1	1	0

$S$  — suma ;  
 $C^0$  — transportul de la rangul anterior ;  
 $C^1$  — transportul către rangul următor.

**Problema 2.7.** Să se minimizeze și să se implementeze cu porți ȘI NU (NAND) ;

$$f = P_3 + P_5 + P_7$$

*Soluție*

Se utilizează trei porți ȘI NU cu două intrări ( $3/4 \times 7400$ ).

**Problema 2.8.** Să se implementeze cu porți ȘI NU (NAND) un circuit de anticoincidență cu trei intrări (circuit care dă 1 la ieșire când cele trei variabile de intrare nu sînt identice, toate 0 sau toate 1).

*Soluție*

$$3/4 \times \text{CDB 400E} + 2/3 \times \text{CDB 410E}.$$

**Problema 2.9.** Să se implementeze cu porți SAU NU (NOR) un circuit cu „vot majoritar“ cu trei intrări (circuit care furnizează la ieșire valoarea logică a majorității variabilelor de intrare).

*Soluție*

$$5/4 \times 74.2$$

**Problema 2.10.** Să se implementeze cu porți SAU NU (NOR) :

$$f = P_0 + P_3 + P_4 + P_5 + P_6 + P_7$$

*Soluție*

$$1 \times 7427 + 1/4 \times 7402.$$

**Problema 2.11.** Să se determine funcția logică realizată de circuitul din figura 2.8.

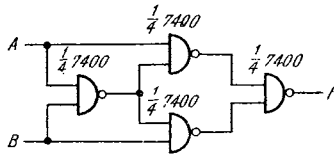


Fig. 2.8.

**Problema 2.12.** Ce operație logică realizează circuitul din figura 2.9 ?

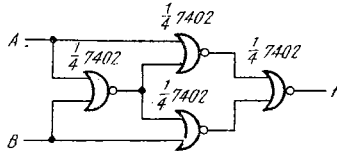


Fig. 2.9.

**Problema 2.13.** Se dă schema logică din figura 2.10 cu ajutorul căreia se comandă aprinderea diodei electroluminiscente (light emitting diode, LED).

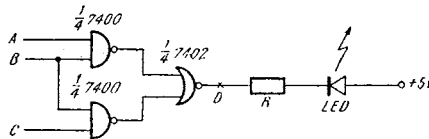


Fig. 2.10.

Se dau nivelele de ieșire ale TTL,  $U_L = 0,4V$ ,  $U_H = 3,5V$  și caracteristica diodei electroluminiscente (fig. 2.11).

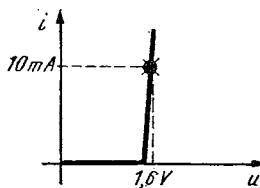


Fig. 2.11.

- a) Să se determine valoarea rezistenței  $R$  astfel ca dioda deschisă să funcționeze în punctul static  $A$  ;
- b) Să se determine tabelul de adevăr  $D(ABC)$  indicându-se stările pentru care dioda se aprinde (este deschisă) ;
- c) Să se înlocuiască circuitul logic care comandă LED-ul printr-o singură poartă integrată.

*Soluție*

a)  $R = 300\Omega$ .

b) Dioda este aprinsă pentru toate combinațiile variabilelor de intrare exceptând-o pe  $ABC = 111$ .

c)  $1/3 \times 7411$ .

**Problema 2.14.** Să se implementeze cu SAU EXCLUSIV (XOR) funcții :

$$f = P_1 + P_2 + P_4 + P_7 + P_8 + P_{11}$$

*Indicație*

Se reduce la forma  $f = A \oplus B \oplus C \oplus D$ .

**Problema 2.15.** Realizați funcția NU cu ajutorul circuitului ȘI SAU NU (AND OR INVERT, AOI) CDB 451E.

*Rezolvare*

Dezavantajul schemei din figura 2.12 constă în aceea că încărcă cu două intrări circuitul precedent.

O soluție mai convenabilă, în care încărcarea este numai o intrare, este cea din figura 2.13.

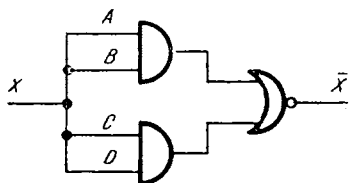


Fig. 2.12.

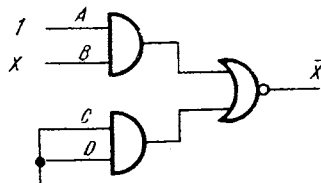


Fig. 2.13.

**Problema 2.16.** Stabiliți tabelul de adevăr al funcției realizate de rețeaua de porți din figura 2.14.

- a) Implementați funcția utilizând numai porți ȘI SAU NU (AOI) ;
- b) Implementați funcția utilizând numai porți ȘI NU (NAND) ;
- c) Comparați cele două soluții ca număr de circuite integrate.

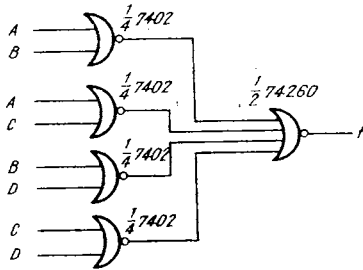


Fig. 2.14.

Soluție

$$a) f = \overline{\overline{A + B + A + C + B + D + C + D}} = \overline{\overline{\overline{A} \overline{B} + \overline{A} \overline{C} + \overline{B} \overline{D} + \overline{C} \overline{D}}} = (\overline{A} + \overline{D})(\overline{B} + \overline{C}) = AD + BC$$

Implementarea se face cu  $1/2 \times 7454$  pentru a obține  $\overline{\overline{AD + BC}}$  și cu  $1/2 \times 7454$  pentru a realiza inversarea expresiei (vezi problema anterioară).

b)  $3/4 \times 7400$ .

c) Ambele implementări conduc la utilizarea unui singur circuit integrat, dar soluția b) este mai economică deoarece mai rămâne o poartă liberă care ar putea fi folosită la realizarea altor funcții.

**Problema 2.17.** Să se realizeze funcția SAU EXCLUSIV (XOR):

a) cu porți ȘI NU (NAND), CDB 400E;

b) cu porți ȘI SAU NU (AOI), CDB 451E;

c) Să se compare cele două soluții ca număr de circuite integrate.

Rezolvare

a) Implementarea cu porți ȘI NU este dată în figura 2.15.

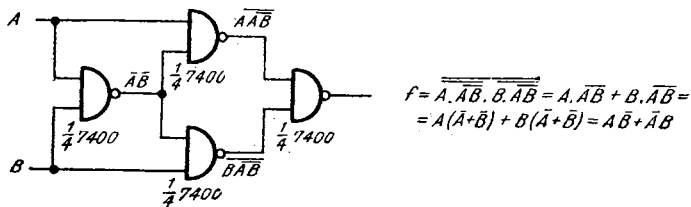


Fig. 2.15.

Deci se utilizează  $1 \times$  CDB 400E (7400).

b) Implementarea cu porți AOI este dată în figura 2.16.

Se utilizează  $1 \times$  CDB 451E (7451).

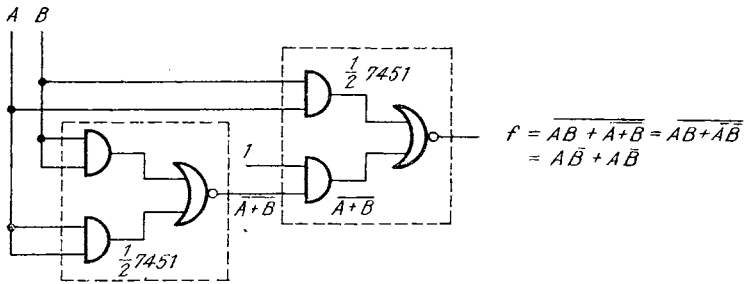


Fig. 2.16.

**Problema 2.18.** Utilizând porți ȘI ȘAU NU (AOI) să se realizeze funcția :

$f = P_0 + P_2 + P_3 + P_4 + P_8 + P_9 + P_{10} + P_{14}$ , cu termeni redundanți  $P_7$  și  $P_{13}$ , considerind că dispunem atât de variabile, cât și de variabile negate.

*Rezolvare*

Rezultă  $f = \overline{A} \overline{C} \overline{D} + \overline{A} \overline{B} C + A \overline{B} \overline{C} + A C \overline{D}$

Termenii redundanți nu contribuie la minimizare, deci vor fi aleși 0.

Implementarea se poate face cu  $1 \times 7455$  (AOI cu două porți AND cu 4 intrări cu expander) și  $1 \times 7460$  (2 expandere cu 4 intrări) ca în figura 2.17.

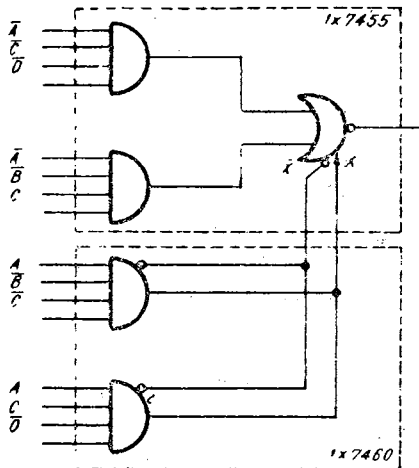


Fig. 2.17.



**Problema 2.19.** Să se implementeze utilizând porți ȘI SAU NU (AOI) funcția :

$$f = ABC + \overline{A}DE + \overline{A}\overline{B}CD + B\overline{C}\overline{D}E + \overline{A}\overline{B}\overline{C} + C\overline{D}\overline{E}$$

**Problema 2.20.** Să se implementeze utilizând porți ȘI SAU NU (AOI) funcția :

$$f = \overline{A}\overline{B} + \overline{C}\overline{D} + \overline{A}\overline{B}\overline{C} + \overline{B}\overline{C}\overline{D}$$

**Problema 2.21.** Să se implementeze cu porți ȘI NU (NAND) circuitul logic combinațional definit de următoarele funcții :

$$f_1 = P_0 + P_1 + P_2 + P_3 + P_{10} + P_{11} + P_{14} + P_{15}$$

$$f_2 = P_0 + P_1 + P_5$$

$$f_3 = P_1 + P_2 + P_3 + P_5$$

- minimizând fiecare funcție independent ;
- minimizând ansamblul funcțiilor.

*Rezolvare*

a) Diagrama Veitch-Karnaugh pentru funcția  $f_1$  este dată în figura 2.18. Suprafețele maxime care acoperă 1-urile sînt  $\overline{A}\overline{B}$  și  $AC$ . Deci :

$$f_1 = \overline{A}\overline{B} + AC = \overline{\overline{A}\overline{B} \cdot \overline{AC}}$$

și se implementează cu 3 porți ȘI NU (NAND) cu 2 intrări și 2 inversoare.

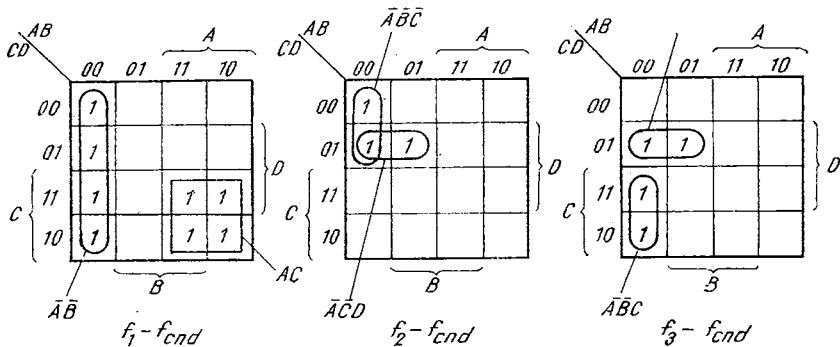


Fig. 2.18.

Din diagrama de minimizare pentru funcția  $f_2$  (figura 2.18) rezultă :

$$f_2 = \bar{A}\bar{B}\bar{C} + \bar{A}\bar{C}D = \overline{\overline{\bar{A}\bar{B}\bar{C}} \cdot \overline{\bar{A}\bar{C}D}}$$

și sînt necesare 2 porți ȘI NU (NAND) cu 3 intrări, 1 poartă cu 2 intrări și încă un inversor : pentru  $\bar{C}$  ( $A$  și  $B$  au fost deja negate pentru obținerea funcției  $f_1$ ).

De asemenea  $f_3$  minimizată (figura 2.18) este :

$$f_3 = \bar{A}\bar{B}C + \bar{A}\bar{C}D = \overline{\overline{\bar{A}\bar{B}C} \cdot \overline{\bar{A}\bar{C}D}}$$

și se implementează cu 2 porți ȘI NU (NAND) cu 3 intrări și 1 poartă ȘI NU cu 2 intrări.

Deci în total sînt necesare 5 circuite integrate  $2 \times 7400$ ,  $2 \times 7410$ ,  $1 \times 7404$ .

b) Se observă că dacă funcția  $f_1$  nu se minimizează total, suprafața  $\bar{A}\bar{B}$  poate fi considerată ca fiind formată din  $\bar{A}\bar{B}C$  și  $\bar{A}\bar{B}\bar{C}$  care sînt comune cu funcțiile  $f_2$ , respectiv  $f_3$ . Astfel implementarea ansamblului de funcții :

$$f_1 = \bar{A}\bar{B}C + \bar{A}\bar{B}\bar{C} + AC$$

$$f_2 = \bar{A}\bar{B}\bar{C} + \bar{A}\bar{C}D$$

$$f_3 = \bar{A}\bar{B}C + \bar{A}\bar{C}D$$

în care termenii comuni se obțin cu ajutorul unei singure porți se face cu numai 3 circuite integrate (fig. 2.19), deci mai economic ca în cazul în care  $f_1$  este în formă minimă.

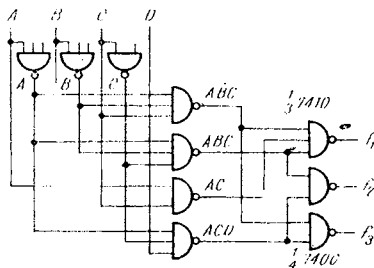


Fig. 2.19.

Deci, pentru un CLC definit de un ansamblu de funcții logice, implementarea se obține, nu neapărat prin minimizarea funcțiilor, ci punînd în evidență termenii comuni cîl mai numeroși.

- Problema 2.22.** Să se implementeze un decodificator BCD/7 segmente
- cu porți ȘI (AND) și SAU (OR);
  - cu porți ȘI NU (NAND);
  - cu porți ȘI SAU NU (AOI).

*Indicație*

Tabelul de adevăr și minimizarea funcțiilor de segment au fost prezentate în problema 1.39.

În acest caz trebuie implementat un sistem de 7 funcții  $a, b, \dots, g$  de 4 variabile  $A, B, C, D$ . Se observă din expresiile minimizate ale funcțiilor, că acestea conțin termeni elementari comuni. *Deci soluția minimă se obține luându-se în considerare ansamblul funcțiilor, nu implementând fiecare funcție independent.*

**Problema 2.23.** Implementați sistemul cu ieșiri multiple definit de funcțiile :

$$f_1 = P_2 + P_3 + P_{10} + P_{13} + P_{15}$$

$$f_2 = P_4 + P_5 + P_{10} + P_{13} + P_{15}$$

$$f_3 = P_8 + P_9 + P_{10}$$

Comparați rezultatul cu implementarea obținută în cazul în care funcțiile sînt minimizate și realizate fiecare în parte.

**Problema 2.24.** Implementați sistemul cu ieșiri multiple :

$$f_1 = \bar{A}BC + C\bar{D} + \bar{A}\bar{B}\bar{C}D + B\bar{D}$$

$$f_2 = A\bar{C} + \bar{B}\bar{C}D + \bar{A}BCD$$

$$f_3 = \bar{A}\bar{B}\bar{C} + AB\bar{C}D + \bar{C}\bar{D} + \bar{B}\bar{D}$$

**Problema 2.25.** Să se stabilească tabelul de adevăr,  $f_{nd}$ ,  $f_{nc}$  și diagramele Veitch-Karnaugh pentru funcția logică implementată în figura 2.20.

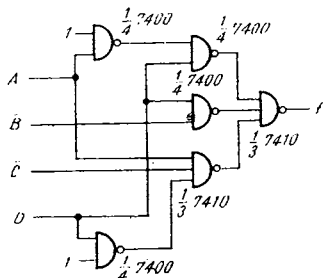


Fig. 2.20.

*Indicație*

În formă elementară  $f = \bar{A}D + \bar{B}D + A\bar{C}\bar{D}$ . Se exprimă în celelalte forme ca în problema 1.28.

**Problema 2.26.** Cum se modifică funcția  $f$  din problema anterioară dacă circuitele ȘI NU din figura 2.21 devin SAU NU-uri ?

**Problema 2.27.** Care este  $f_{cnd}$  a funcției implementată în figura 2.21 ?

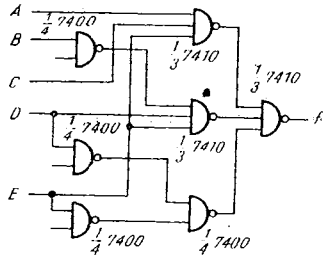


Fig. 2.21.

*Soluție*

$$f = P_0 + P_2 + P_4 + P_6 + P_8 + P_{12} + P_{16} + P_{18} + P_{20} + P_{22} + P_{24} + P_{28}$$

unde

$$A - 2^4, B - 2^3, C - 2^2, D - 2^1, E - 2^0.$$

**Problema 2.28.**

- Să se deducă pentru funcția implementată în figura 2.22 diagrama Veitch-Karnaugh, tabelul de adevăr,  $f_{cnd}$  și  $f_{cnc}$ .
- Ce proprietate are funcția  $f$  ?

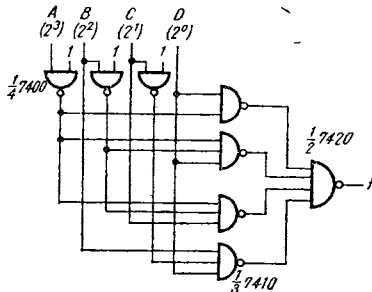


Fig. 2.22.

*Soluție*

b) Funcția  $f$  semnalizează apariția numerelor prime în intervalul  $[0-15]$ .

**Problema 2.29.** Să se reia problema anterioară pentru funcția implementată în figura 2.23.

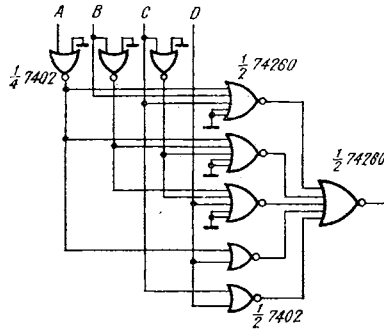


Fig. 2.23.

*Soluție*

Este aceeași funcție ca în problema precedentă.

**Problema 2.30.** Să se exprime funcțiile implementate în figura 2.24 prin  $f_{cnd}$  și tabel de adevăr.

Implementarea din figură este minimă ?

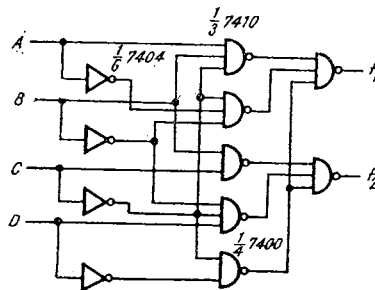


Fig. 2.24.

**Problema 2.31.** Să se proiecteze un circuit combinațional cu 3 variabile de intrare care să semnalizeze la ieșire apariția tuturor termenilor canonici de tip  $P$  (decodificator).

Să se arate care sînt deosebirile între decodificator (DCD) și demultiplexor (DMUX).

*Rezolvare*

Pentru 3 variabile de intrare există  $2^3 = 8$  termeni canonici distincți, deci circuitul va avea 8 ieșiri. Soluția este prezentată în figura 2.25, considerînd intrarea  $E$  și conexiunea punctată inexistentă.

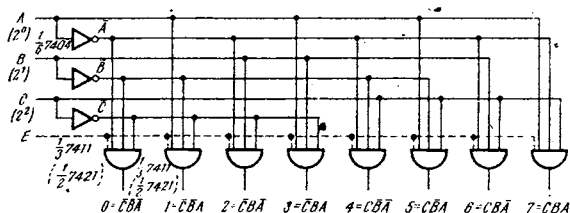


Fig. 2.25.

Se observă că pentru obținerea termenilor canonici sînt necesare atît variabilele de intrare  $A, B, C$  cît și negatele acestora.

Un astfel de circuit se numește decodificator (DCD).

Demultiplexorul (DMUX) este un DCD care are o intrare suplimentară ( $E$  — conexiunea punctată din figura 2.25) de condiționare a funcționării. Astfel pentru  $E = 1$ , DMUX lucrează ca DCD, iar pentru  $E = 0$  este inhibată funcționarea, toate ieșirile fiind pe 0.

O altă interpretare a schemei de DMUX este următoarea : DMUX distribuie semnalul (0 sau 1) de pe o cale ( $E$ ), pe mai multe căi (0—7), în funcție de adresa ( $A, B, C$ ) acelor căi.

**Problema 2.32.** În figura 2.26 se dă structura demultiplexorului 74155.

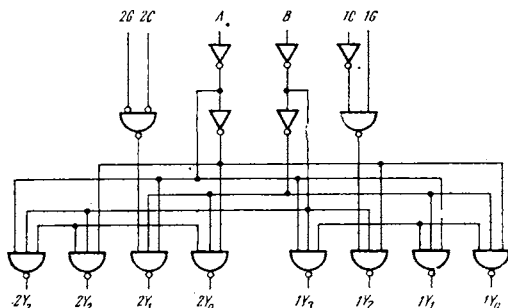


Fig. 2.26.

a) Să se determine tabelul de adevăr și să se descrie funcționarea circuitului ;

b) Să se repete punctul a, în cazul în care se conectează împreună 1C cu 2C și 1G cu 2G.

$$(1C = 2C = C, 1G = 2G = G)$$

c) Ce diferențe sînt între DMUX 74155 utilizat ca la punctul b și DMUX implementat cu porți din figura 2.25 ?

*Rezolvare*

c) Ambele sînt demultiplexoare cu 8 ieșiri, dar cel integrat (74155) lucrează pe activ zero față de cel din figura 2.25 : avînd ȘI NU-uri pe ieșiri, semnalizează prin 0 apariția adreselor la intrare, funcționarea fiind permisă cînd  $G = 0$ . Spunem că DMUX integrat este activ pe 0 în timp ce DMUX implementat cu porți integrate este activ pe 1.

**Problema 2.33.** Să se deducă tabelul de adevăr și structura demultiplexorului 74154, știind că decodifică 4 variabile, are 2 intrări de condiționare  $G_1, G_2$  care permit funcționarea circuitului doar cînd sînt ambele pe 0 și că se semnalizează prin 0 la ieșire apariția la intrare a configurațiilor asociate termenilor canonici.

**Problema 2.34.** Să se implementeze cu decodificator (DCD) și porți funcția :

$$f = P_0 + P_2 + P_5 + P_6$$

*Rezolvare*

Deoarece funcția are 3 variabile este necesar un decodificator cu 3 intrări. Acesta va furniza informație despre apariția celor 8 termeni canonici la intrare. Cu porți se selectează acei termeni canonici care intervin în expresia funcției.

*Soluția I.* Cu poartă ȘI NU (fig. 2.27).

*Soluția II.* Cu poartă ȘI (fig. 2.28).

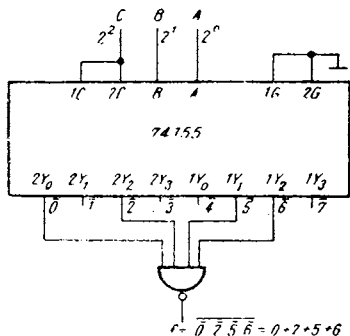
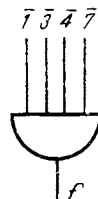


Fig. 2.27.



$$\begin{aligned} \bar{f} &= 1 + \bar{3} + 4 = \overline{1 \cdot \bar{3} \cdot 4 \cdot \bar{7}} \\ \Rightarrow \overline{1 \cdot \bar{3} \cdot 4 \cdot \bar{7}} &= f \end{aligned}$$

Fig. 2.28.

În loc de poarta ȘI NU se poate utiliza o poartă ȘI la care se conectează terminalele asociate termenilor canonici ce nu intervin în expresia funcției. Aceștia sînt toți 1 cînd la intrarea decodicatorului apar configurațiile asociate lui 0, 2, 5 sau 6 și deci valorile logice obținute pentru  $f$  la ieșirea ȘI-ului sînt corecte.

Deci la implementarea cu DCD (circuite MSI) nu mai este necesară minimizarea, se pornește de la fcd a funcției.

**Problema 2.35.** Să se implementeze cu DCD și porți un sumator binar complet de 2 biți.

**Problema 2.36.** Să se implementeze cu DCD și porți funcția  $f(A, B, C)$  care ia valoarea logică 1 cînd un număr par de variabile de intrare este 1. Care dintre implementările, cu poartă ȘI sau cu poartă ȘI NU este mai convenabilă și de ce ?

**Problema 2.37.** Să se implementeze cu DCD și porți funcția  $f(A, B, C)$  care ia valoarea logică 1 cînd un număr impar de variabile este 1.

**Problema 2.38.** Să se implementeze cu DCD și porți o funcție care semnalizează apariția numerelor prime cuprinse între 0 și 15.

**Problema 2.39.** Să se implementeze cu DCD și porți circuitul logic combinațional cu ieșiri multiple definit de :

$$f_1 = P_0 + P_4 + P_5 + P_9$$

$$f_2 = P_1 + P_3 + P_7 + P_8 + P_{10} + P_{11} + P_{12}$$

$$f_3 = P_0 + P_3 + P_4 + P_5$$

$$f_4 = P_7 + P_9 + P_{11} + P_{13} + P_{15}$$

*Indicație*

Se utilizează  $1 \times 74154$ ,  $1 \times 7420$ ,  $2 \times 7430$ .

Pentru implementarea unui circuit logic combinațional cu ieșiri multiple este necesar un singur DCD și se adaugă cîte o poartă pentru fiecare funcție din sistem.

**Problema 2.40.** Realizați cu demultiplexoare un decodicator cu 64 de căi ( $2^6$  stări) — extinderea demultiplexării.

*Rezolvare* (fig. 2.29).

Interpretări ale funcționării schemei :

a) Se decodifică cele 64 de stări de intrare atunci cînd  $E$  (enable) este pe 0 ;

b) Se transmit datele, 0 sau 1, de la intrarea  $E$  pe oricare din cele 64 de căi selectate prin adresă.



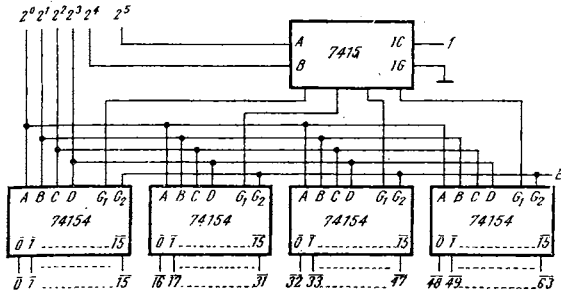


Fig. 2.29.

**Problema 2.41.** Extindeți demultiplexarea a 8 căi la 64 de căi de semnal.

*Indicație*

Se utilizează  $9 \times 74155$ .

**Problema 2.42.** Realizați cu 2 demultiplexoare (DMUX) cu 8 căi un DMUX cu 16 căi.

**Problema 2.43.** Implementați funcția :

$$f = P_0 + P_2 + P_3 + P_4 + P_7 + P_8 + P_9$$

- a) cu DMUX de 16 căi, 74154;
- b) cu DMUX de 8 căi, 74155;
- c) cu DCD de 10 căi, 7442.

**Problema 2.44.** Să se realizeze cu un DCD de  $2^{n-1}$  căi o funcție cu  $n$  variabile de intrare.

*Rezolvare*

Se va considera exemplul concret al funcției de 3 variabile ce semnalizează apariția unui număr impar de 1-uri la intrare. Tabelul de adevăr al funcției  $f$  este :

	A	B	C	f
0	0	0	0	0
1	0	0	1	1
2	0	1	0	1
3	0	1	1	0
4	1	0	0	0
5	1	0	1	0
6	1	1	0	1
7	1	1	1	1

Conform celor arătate în problema 56,  $f$ , avind 3 variabile se poate implementa cu un DCD cu  $2^3 = 8$  căi (74155 sau 7442). Dar problema ne cere să realizăm funcția cu un DCD de  $2^2 = 4$  căi. Soluția constă în a ține cont de cea de a 3-a variabilă (fie aceasta  $A$ ), nu la intrarea DCD, ci la intrarea porților ce selectează termenii canonici din expresia funcției (fig. 2.30).

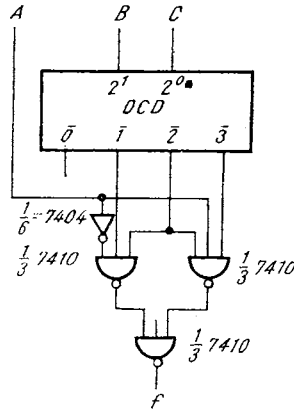


Fig. 2.30.

Pentru a demonstra corectitudinea schemei se scrie expresia funcției  $f$ , se separă variabilele  $BC$  și se aplică teoremele lui De Morgan

$$\begin{aligned}
 f &= \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC = \bar{A}(\bar{B}C) + B\bar{C} + A(BC) = \\
 &= \bar{A}P_1 + P_2 + AP_3 = \overline{\bar{A}P_1} \cdot \overline{P_2} \cdot \overline{AP_3} = (A + \bar{P}_1) \cdot \bar{P}_2 \cdot (\bar{A} + \bar{P}_3) = \\
 &= \bar{P}_2(A\bar{P}_3 + \bar{A}\bar{P}_1) = A\bar{P}_2\bar{P}_3 + \bar{A}\bar{P}_1\bar{P}_2
 \end{aligned}$$

Produsul  $\bar{P}_1\bar{P}_3$  este 0 deoarece ieșirile DCD,  $\bar{P}_1$  și  $\bar{P}_3$  nu sînt simultan pe 1.

**Problema 2.45.** Să se realizeze funcțiile care selectează numerele prime, numerele multiplu de 3 și numerele multiplu de 4, cuprinse între 0 și 31 :

- cu 2 DMUX de 16 căi și porți ;
- cu 1 DMUX de 16 căi și porți ;
- cu porți.

d. Să se compare soluțiile  $a$ ,  $b$ ,  $c$  ca număr de circuite integrate și simplitate de proiectare.

❖ **Problema 2.46.** Să se implementeze cu DCD 7442 și porți un translator de cod BCD/7 segmente :

- pentru aprinderea unor LED-uri cu catod comun ;
- pentru aprinderea unor LED-uri cu anod comun.

**Problema 2.47.** Să se realizeze cu DMUX un sistem de transmitere pe cite 8 căi a datelor de pe 2 căi de informație,  $D_1$  și  $D_2$ .

*Soluție* (fig. 2.31).

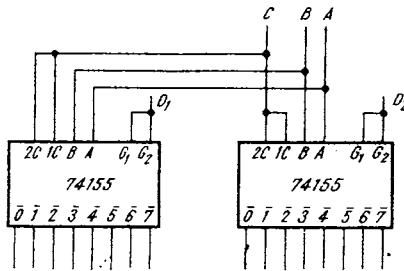


Fig. 2.31.

**Problema 2.48.** Să se realizeze cu DMUX de 8 căi și porți extinderea decodificării la 10 căi.

*Rezolvare*

Schema din figura 2.32 realizează decodificarea primelor 10 stări ce se obțin cu 4 biți. Se poate utiliza, de exemplu, pentru comanda aprinderii tuburilor NIXIE de afișaj. Fiecare ieșire se conectează la unul din anozii (catozii) tubului, selectând cifra ce trebuie aprinsă.

Se observă la această schemă că stările 10—15 sînt interpretate tot ca 8 și respectiv 9. În aplicațiile în care ne deranjează apariția acestor erori, se poate utiliza o schemă în care se decodifică complet cele două configurații.

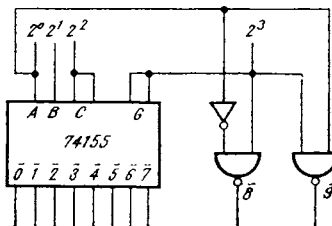


Fig. 2.32.

**Problema 2.49.** Să se implementeze funcția :

$$f = P_1 + P_3 + P_5 + P_7 + P_9 + P_{11} + P_{13} + P_{15}$$

utilizând un DCD și porți. Să se compare cu soluția utilizând numai circuite integrate SSI.

**Problema 2.50.** Implementați cu DCD sistemul cu ieșiri multiple :

$$f_1 = \bar{A}BC + C\bar{D} + \bar{A}\bar{B}\bar{C}D + B\bar{D}$$

$$f_2 = A\bar{C} + \bar{B}\bar{C}D + \bar{A}BCD$$

$$f_3 = \bar{A}\bar{B}\bar{C} + AB\bar{C}D + \bar{C}\bar{D} + \bar{B}\bar{D}$$

Comparați cu rezultatele obținute la problema 2.23.

*Indicație*

Se aduc expresiile funcțiilor la *fend*. De exemplu :

$$f_2 = A\bar{C}\bar{B}D + A\bar{C}B\bar{D} + A\bar{C}B\bar{D} + A\bar{C}\bar{B}\bar{D} + A\bar{B}\bar{C}D + \bar{A}\bar{B}\bar{C}D + \\ + \bar{A}BCD = P_{13} + P_9 + P_{12} + P_8 + P_9 + P_1 + P_7 = P_1 + P_7 + \\ + P_8 + P_9 + P_{12} + P_{13}$$

**Problema 2.51.** Implementați cu DCD și porți sistemul cu ieșiri multiple definit în problema 2.21. Comparați cu rezultatul obținut la implementarea cu SSI.

**Problema 2.52.** Utilizați un DCD 7442 ca un DMUX de 8 căi.

*Soluție*

Vezi figura 2.33.

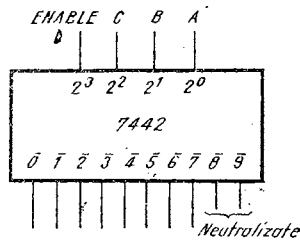


Fig. 2.33.

**Problema 2.53.** Implementați funcția :

$$f = P_0 + P_2 + P_3 + P_9$$

cu DCD 7442 și porți.

**Problema 2.54.** Să se proiecteze o schemă logică cu porți ȘI, SAU și inversoare, cu ajutorul căreia să se poată realiza orice funcție cu 3 intrări.

*Rezolvare*

În cazul general cele  $2^8$  funcții cu 3 intrări se pot exprima prin :

$$f = X_0 \bar{C}\bar{B}\bar{A} + X_1 \bar{C}\bar{B}A + X_2 \bar{C}B\bar{A} + X_3 \bar{C}BA + X_4 C\bar{B}\bar{A} + X_5 C\bar{B}A + X_6 CBA + X_7 CBA$$

particularitățile obținându-se prin vectorul  $X(X_0, X_1, \dots, X_7)$ , unde  $X_{i(0-7)} \in \{0, 1\}$ . De exemplu, funcția

$$f_1 = P_0 + P_2 + P_3 + P_5$$

are vectorul asociat  $X(1, 0, 1, 1, 0, 1, 0, 0)$ .

Implementarea funcției  $f$  se poate face ca în figura 2.34.

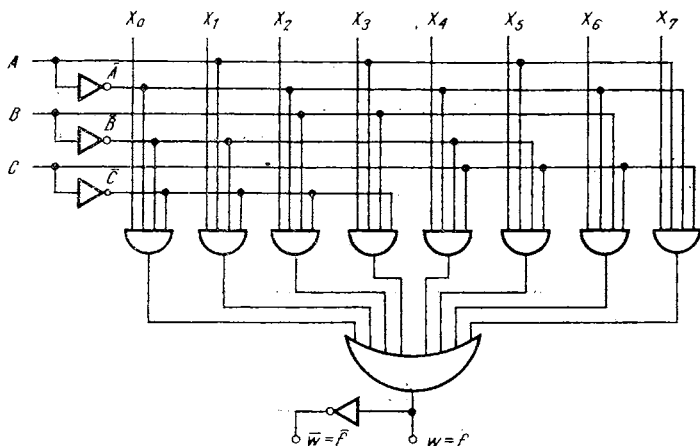


Fig. 2.34.

Această structură se numește multiplexor (MUX) de 8 căi și se găsește integrată în CI MSI, 74151, cu unele mici deosebiri de schemă internă (vezi anexa), care păstrează însă aceeași funcționare.

Interpretări posibile pentru funcționare sînt :

**A.** MUX realizează orice funcție logică cu 3 intrări,  $A, B, C$ , prin conectarea corespunzătoare a intrărilor  $X_i$ .

**B.** MUX transferă datele de la una din căile  $X_i$ , selectabilă prin adresare ( $A, B, C$ ), la ieșirea  $W$ .

Deci MUX adună datele de pe mai multe căi, pe o singură cale deosebire de DMUX care distribuie datele de pe o cale pe mai căi.

C. MUX realizează conversia biților din paralel în serie da baleiază succesiv adresele sale.

În cazul particular al funcției  $f_1$  implementarea este dată în ra 2.35.

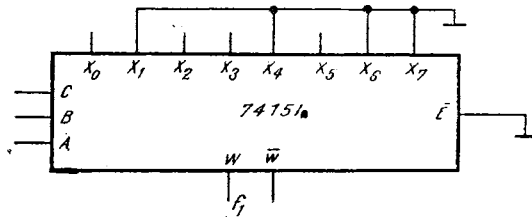


Fig. 2.35.

Deci la implementarea cu MUX nu e necesară minimizarea.

**Problema 2.55.** Să se implementeze funcția  $f_1$  din problema rioară cu ajutorul MUX 74152 (care nu are decât ieșire  $\bar{W}$ ).

*Soluție*

Vezi figura 2.36.

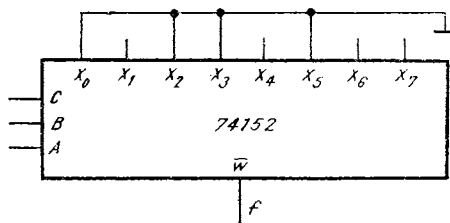


Fig. 2.36.

**Problema 2.56.** Să se implementeze funcția de 3 variabile :

$$f = P_0 + P_1 + P_2 + P_6 + P_7$$

cu ajutorul unui MUX de 8 căi.

**Problema 2.57.** Să se implementeze funcția de 4 variabile :

$$f = P_1 + P_3 + P_5 + P_7 + P_8 + P_{10} + P_{12} + P_{14}$$

cu ajutorul unui MUX de 16 căi 74150.

**Problema 2.58.** Să se implementeze o funcție de  $n$  variabile de intrare cu ajutorul unui MUX de  $2^{n-1}$  căi.

*Rezolvare*

Fie aceeași funcție  $f_1$  din problema 2.54 care, în acest caz, va trebui implementată cu ajutorul unui MUX de  $2^2 = 4$  căi. Pornind de la tabelul de adevăr al lui  $f_1$ :

	C	B	A	$f_1$
0	0	0	0	1
1	0	0	1	0
2	0	1	0	1
3	0	1	1	1
4	1	0	0	0
5	1	0	1	1
6	1	1	0	0
7	1	1	1	0

se separă două dintre variabile,  $B$  și  $C$ , care vor constitui intrările de selecție ale MUX. Variabila  $A$  se compară cu  $f_1$  jucînd rolul intrării  $X$  de validare: pentru  $CB = 00$ ,  $\bar{A} = f_1$ , pentru  $CB = 01$ ,  $f_1$  este 1, deci nu depinde de  $A$ , ș.a.m.d. Schema (figura 2.37) se poate determina și pornind de la *fncl* a funcției prin separarea uneia din variabile:

$$f = (\bar{C}\bar{B})A + (\bar{C}B)\bar{A} + (\bar{C}B)A + (C\bar{B})A = (\bar{C}\bar{B})A + (\bar{C}B) \cdot 1 + (C\bar{B})A = P_0 \cdot \bar{A} + P_1 \cdot 1 + P_2 \cdot A + P_3 \cdot 0$$

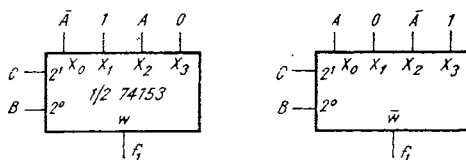


Fig. 2.37.

Acastă implementare a funcțiilor cu MUX este mai economică decît cea din problemele 2.54—2.57 și se recomandă a fi utilizată în toate aplicațiile.

**Problema 2.59.** Să se implementeze funcția:

$$f = P_0 + P_4 + P_8 + P_{12} + P_{15}$$

cu ajutorul unui MUX de 8 căi.

**Problema 2.60.** Să se implementeze funcțiile din problemele 2.5 și 2.57 cu MUX de 4 căi, respectiv 8 căi.

**Problema 2.61.** Să se implementeze funcția :

$$f = P_1 + P_3 + P_5 + P_7 + P_{11} + P_{13} + P_{17} + P_{19} + P_{21} + P_{23}$$

cu ajutorul unui MUX de 16 căi.

**Problema 2.62.** a) Să se implementeze un sumator complet de 4 biți cu MUX de 4 căi.

b) Să se compare cu soluția cu DCD.

*Rezolvare*

b) Implementarea cu MUX folosește tot 2 circuite integrate, dar acestea sînt amîndouă MSI. Soluția cu DCD utilizează 1 MSI și 1 C) de porți, deci este mai economică.

Pentru implementarea cu MSI a sistemelor logice combinaționale cu mai multe ieșiri se preferă soluția cu DCD și porți deoarece se adaugă cîte o poartă pentru fiecare ieșire. La soluția cu MUX se adaugă cîte un MUX pentru fiecare ieșire, ceea ce este neconvenabil.

**Problema 2.63.** Să se realizeze o extindere a multiplexării de la 8 la 16 căi.

*Soluție*

Vezi figura 2.38.

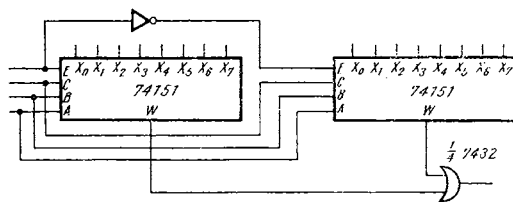


Fig. 2.38.

**Problema 2.64.** Să se realizeze o multiplexare de 64 de căi :

a) cu MUX de 8 căi ;

b) cu MUX de 16 căi.

**Problema 2.65.** Să se realizeze cu MUX de 8 căi și porți o multiplexare de 10 căi.

**Problema 2.66.** Să se realizeze funcția \$f\$ de 5 variabile cu 2 etaje de multiplexoare :

$$f = P_0 + P_2 + P_4 + P_5 + P_{11} + P_{20} + P_{24} + P_{25} + P_{31}$$



*Rezolvare*

Tabelul de adevăr al funcției este următorul :

	<i>E</i>	<i>D</i>	<i>C</i>	<i>B</i>	<i>A</i>	<i>f</i>
0	0	0	0	0	0	1
1			0	0	1	0
2			0	1	0	1
3			0	1	1	0
4			1	0	0	1
5			1	0	1	1
6			1	1	0	0
7			1	1	1	0
8	0	1	0	0	0	0
9			0	0	1	0
10			0	1	0	0
11			0	1	1	1
12			1	0	0	0
13			1	0	1	0
14			1	1	0	0
15			1	1	1	0
16	1	0	0	0	0	0
17			0	0	1	0
18			0	1	0	0
19			0	1	1	0
20			1	0	0	1
21			1	0	1	0
22			1	1	0	0
23			1	1	1	0
24	1	1	0	0	0	1
25			0	0	1	1
26			0	1	0	0
27			0	1	1	0
28			1	0	0	0
29			1	0	1	0
30			1	1	0	0
31			1	1	1	1

Se separă variabilele *ED* de *CBA* și se implementează pe rând cu 4 MUX de 4 căi grupele  $f(CBA)$  definite de fiecare configurație *ED*. Acestea se adună apoi cu ajutorul altui MUX de 4 căi.

Desigur funcția se implementează mai economic cu un MUX de 16 căi. S-a ales acest exemplu doar pentru ilustrarea simplificată a metodei care este aplicabilă în cazul funcțiilor cu 6 sau mai multe variabile.

Schema de implementare este dată în figura 2.39.

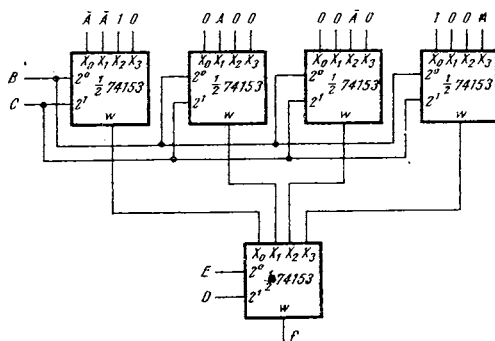


Fig. 2.39.

**Problema 2.67.** Să se implementeze funcția de 6 variabile care semnalizează apariția multiplilor lui 5 în intervalul 0—63, cu 2 etaje de MUX.

**Problema 2.68.** Să se implementeze cu 2 etaje de MUX o funcție care ia valoarea logică 1 pentru numerele prime cuprinse în intervalul 0—63. Cum se modifică schema, dacă dorim semnal 1 doar pentru numerele prime cuprinse între 32—63 ?

**Problema 2.69.** Implementați convertorul din cod BCD în cod Gray definit de tabelul următor :

A	B	C	D	$f_1$	$f_2$	$f_3$	$f_4$
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	0
0	1	0	0	0	1	1	0
0	1	0	1	0	1	1	1
0	1	1	0	0	1	0	1
0	1	1	1	0	1	0	0
1	0	0	0	1	1	0	0
1	0	0	1	1	1	0	1

- cu DCD7442 și porți ;
- cu DMUX de 8 căi și porți ;
- cu DMUX de 16 căi și porți ;
- cu MUX de 8 căi ;

- e) cu porți ȘI NU (NAND).
- f) Comparați ca număr de circuite integrate toate soluțiile de implementare anterioare.

**Problema 2.70.** Să se implementeze un decodor BCD exces 3/7 segmente :

- a) cu DCD7442 și porți ;
- b) cu DMUX de 8 căi și porți ;
- c) cu MUX de 8 căi ;
- d) cu porți ȘI NU (NAND).
- e) Comparați soluțiile ca număr de circuite integrate.

**Problema 2.71.** Să se implementeze minim un convertor de cod Gray/BCD :

- a) cu DCD ;
- b) cu MUX ;
- c) cu porți ȘI NU (NAND).
- d) Comparați soluțiile ca număr de circuite integrate.

**Problema 2.72.** Să se implementeze un circuit logic combinațional definit în figura 2.40, unde :

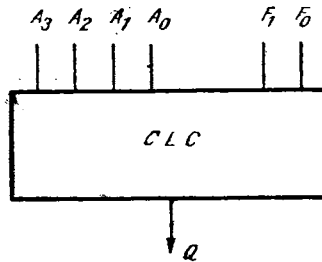


Fig. 2.40.

$F_0$	$F_1$	
0	0	$Q = 1$ dacă $A_3A_2A_1A_0$ es'e multiplu de 2 ;
0	1	$Q = 1$ dacă $A_3A_2A_1A_0$ es'e multiplu de 4 ;
1	0	$Q = 1$ dacă $A_3A_2A_1A_0 = 13$ ;
1	1	$Q = 1$ dacă $A_3 = 1$ .

Implementarea se va face cu :

- a) porți ȘI NU (NAND) ;
- b) MUX ;
- c) DCD.

Se vor compara soluțiile ca număr de circuite integrate.

**Problema 2.73.** Folosind rezultatele din problema 1.20 să se explice funcționarea circuitului din figura 2.41.

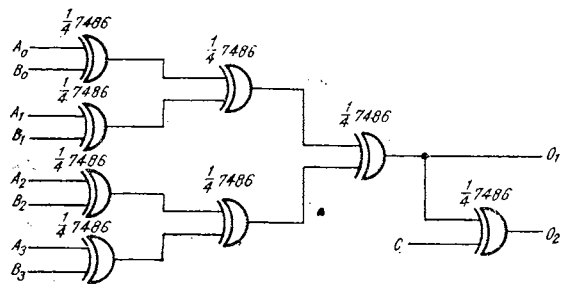


Fig. 2.41.

*Indicație*

Circuitul este un detector de paritate și imparitate.

**Problema 2.74.** Să se explice funcționarea circuitului 74180 a cărei schemă logică este dată în figura 2.42.

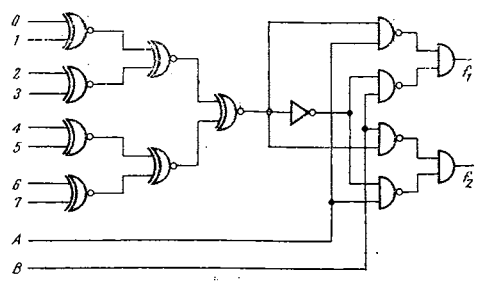


Fig. 2.42.

*Indicație*

Circuitul este un detector de paritate și imparitate.

**Problema 2.75.** Cunoscând funcționarea codicatorului prioritar de 8 intrări 74148, să se explice funcționarea schemei din figura 2.43.

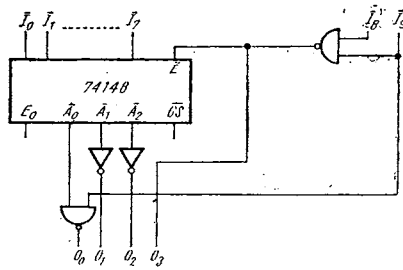


Fig. 2.43.

*Soluție*

Schema realizează codificarea în BCD a numerelor zecimale 0–9, în ordinea priorității. De exemplu, dacă apare la intrare vectorul 101111101, cu 0 pe intrările 1 și 8, la ieșire se va transmite codul lui 8(0001).

**Problema 2.76.** Realizați cu  $2 \times 74148$ ,  $1 \times$  MUX de 2 căi și porți un codificator prioritar din zecimal (20 căi) în BCD.

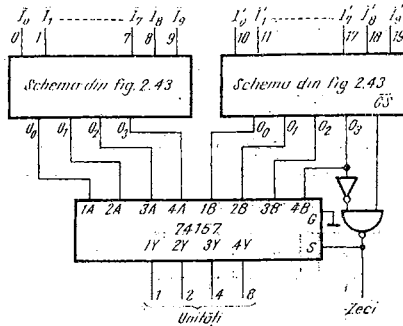


Fig. 2.44.

*Soluție*

Vezi figura 2.44.

**Problema 2.77.** Să se explice funcționarea schemei din figura 2.45.

*Soluție*

În cazul în care apar mai multe semnale pe intrări, schema transmite doar pe cel mai prioritar.

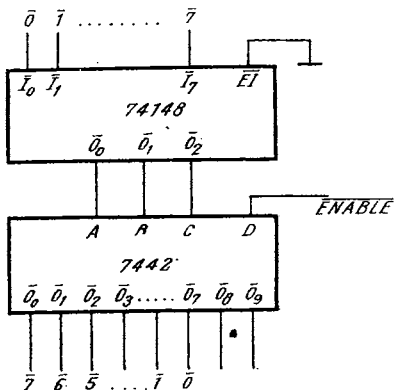


Fig. 2.45.

**Problema 2.78.** Se dă schema din figura 2.46 care realizează conversia unui cod de 5 biți în alt cod de 5 biți.

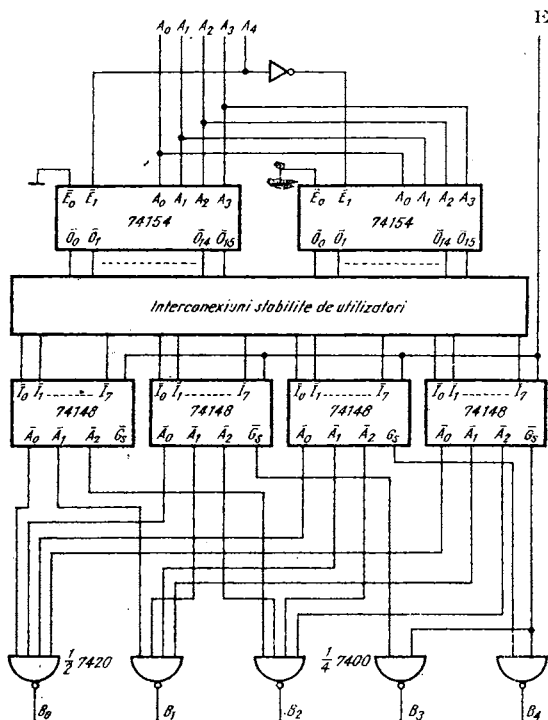


Fig. 2.46.

Să se stabilească la alegere o anumită conversie de cod și să se implementeze sistemul de interconexiuni.

**Problema 2.79.** Să se descrie funcționarea circuitului din figura 2.47 destinat să detecteze dacă un singur 0 se aplică pe intrări.

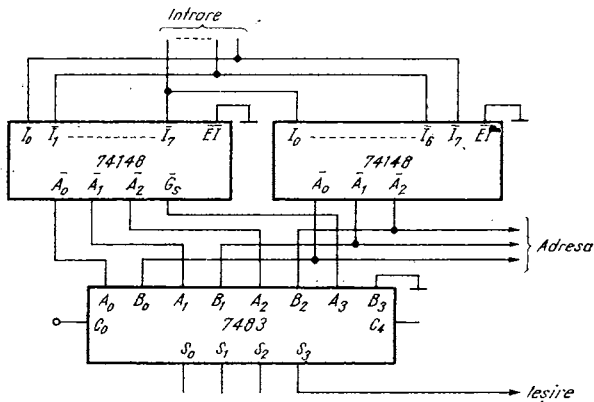


Fig. 2.47.

**Problema 2.80.** Să se realizeze utilizând  $2 \times 74148$  un codificator prioritar de 16 biți.

*Soluție*

Vezi figura 2.48.

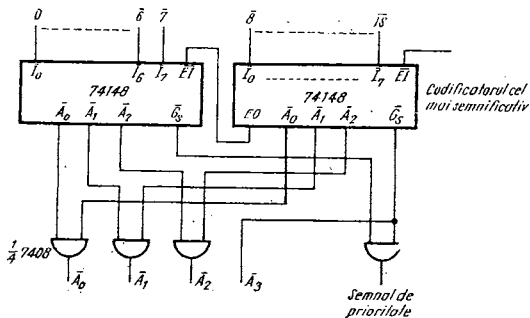


Fig. 2.48.

**Problema 2.81.** Să se implementeze un codificator prioritar de 64 de biți :

- a) utilizând  $9 \times 74148$  și porți ;  
 b) utilizând  $9 \times 74148$  și  $3 \times \text{MUX}$  de 8 căi.

**Soluție**

a) Vezi figura 2.49.

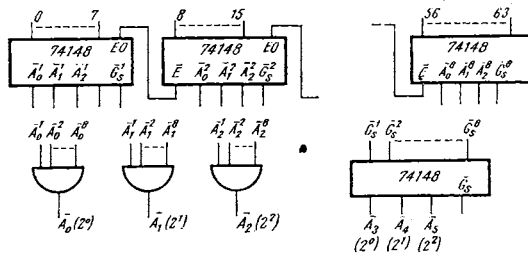


Fig. 2.49.

b) Se înlocuiesc porțile SI cu 8 intrări cu MUX 74152 ca în figura 2.50.

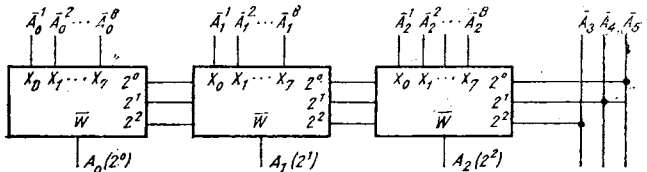


Fig. 2.50.

**Problema 2.82.** Să se găsească tipurile de porți simple cu ajutorul cărora se poate realiza compararea a 2 biți.

**Soluție**

Vezi figura 2.51.

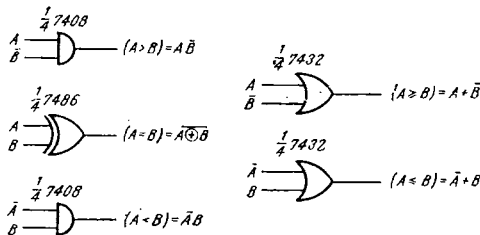


Fig. 2.51.



**Problema 2.33.** Să se realizeze o schemă de semnalizare a identității a 2 cuvinte de câte 4 biți :

- a) cu porți ;
- b) cu un MUX și un DCD de 16 căi.

*Soluție*

b) vezi figura 2.52.

**Problema 2.84.** Să se realizeze un comparator de 2 cuvinte de câte 2 biți,  $A(A_1A_0)$  și  $B(B_1B_0)$  definit ca în figura 2.53.

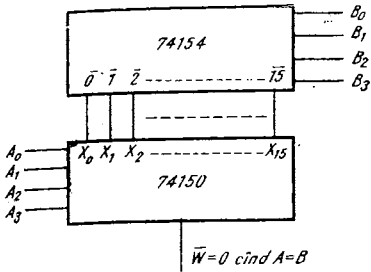


Fig. 2.52.

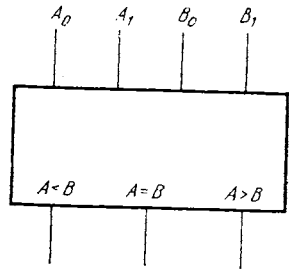


Fig. 2.53.

**Problema 2.85.** Să se implementeze cu porți un circuit care să dea semnalizează prin 1 când cuvântul  $A(A_2A_1A_0)$  este mai mare decât  $B(B_2B_1B_0)$ .

*Soluție*

I. cu transport succesiv (figura 2.54).

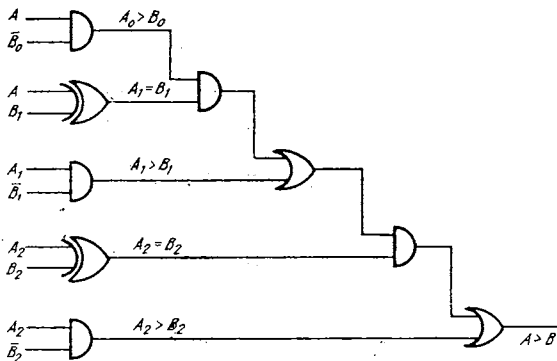


Fig. 2.54.

II. cu transport simultan (fig. 2.55).

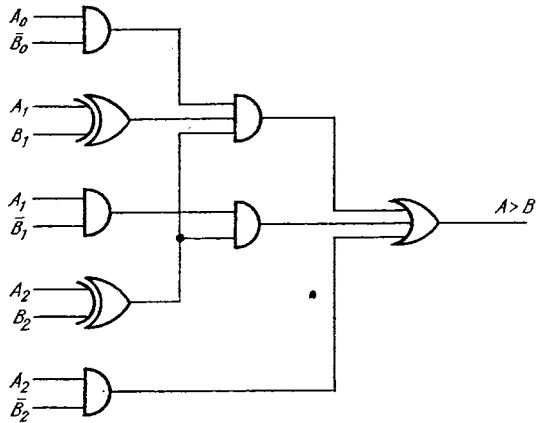


Fig. 2.55.

**Problema 2.86.** Să se realizeze circuitul anterior în cazul com a 2 cuvinte de 4 biți :

- a) cu transport succesiv ;
- b) cu transport simultan.

**Problema 2.87.** Să se realizeze un comparator de cuvinte de utilizând  $2 \times 7485$ .

*Soluție*

Vezi figura 2.56.

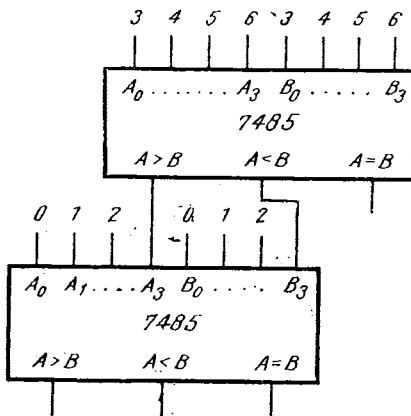


Fig. 2.56.

Se distribuie biții cei mai semnificativi pe primul nivel, rezultatul comparării acestora aplicându-se celui de al doilea comparator în pozițiile cele mai semnificative.

**Problema 2.88.** Să se realizeze compararea a 2 cuvinte de 8 biți utilizând  $2 \times 7485$ .

*Soluție*

Se utilizează și intrările  $A < B$ ,  $A > B$ ,  $A = B$  ale comparatorului (fig. 2.57).

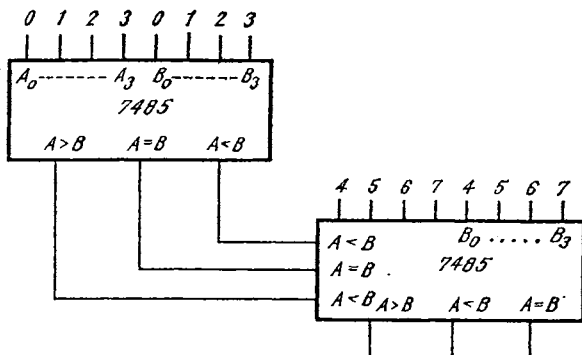


Fig. 2.57.

**Problema 2.89.** Să se realizeze cu porți ȘI, SAU, EXCLUSIV un sumator complet de 2 biți.

*Soluție*

Vezi figura 2.58.

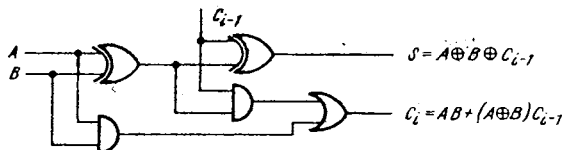


Fig. 2.58.

**Problema 2.90.** Să se implementeze cu modulul realizat anterior un sumator de 2 cuvinte de 4 biți.

**Problema 2.91.** a) Utilizând sumatoare de 2 biți să se realizeze un circuit de adunare binară a  $n$  biți, cu propagarea transportului (carry) de la un rang binar la altul.

b) Să se realizeze o schemă cu propagare rapidă a transportului.

*Rezolvare*

a) Se conectează câte un sumator de 2 biți pentru fiecare rang binar ca în figura 2.59. La adunarea numerelor  $A(A_0, A_1, \dots, A_{n-1})$  cu  $B(B_0, B_1, \dots, B_{n-1})$  rezultatul este format din numărul binar  $S(S_0, S_1, \dots, S_{n-1})$  și transportul final,  $C = C_n$ . Dezavantajul unei astfel de scheme constă în faptul că transportul se obține după o întârziere mare, egală cu  $nT_p$  ( $T_p$  este timpul de propagare a transportului printr-un sumator).

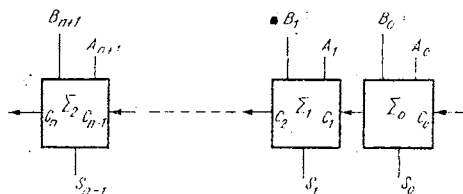


Fig. 2.59.

b) Pentru a înlătura dezavantajul amintit se poate realiza un circuit care propagă rapid transportul (carry lookahead).

Se știe că :

Transportul pentru primul rang binar,  $\Sigma_0$ , este  $C_0$ .

Transportul pentru poziția 1 este  $C_1 = A_0B_0 + C_0(A_0 + B_0)$ .

Transportul pentru poziția 2 este  $C_2 = A_1B_1 + C_1(A_1 + B_1)$ .

Generalizînd se obține  $C_{K+1} = A_KB_K + C_K(A_K + B_K)$ .

Notînd cu  $P_K = A_KB_K$  și  $S_K = A_K + B_K$ , se poate exprima transportul în funcție de  $C_0$  astfel :

$$C_{K+1} = P_K + (S_K P_{K-1}) + S_K S_{K-1} P_{K-2} + (S_K S_{K-1} S_{K-2} P_{K-3}) + \dots + (S_K S_{K-1} \dots S_1 P_0) + (S_K S_{K-1} \dots S_0 C_0)$$

Se observă că propagarea rapidă a transportului se poate face realizînd funcțiile ȘI ( $P_K = A_K B_K$ ) și SAU ( $S_K = A_K + B_K$ ) între variabilele de intrare ale fiecărui rang binar și apoi grupîndu-le cu funcția ȘI-SA U, conform expresiei anterioare.

Funcția  $P_K = A_K B_K$  generează transport (carry generate) independent de transportul anterior.

Funcția  $S_K = A_K + B_K$  (carry propagate) permite trecerea transportului anterior.

În figura 2.60 este dată implementarea unității de program rapidă a transportului pentru un sumator de 4 biți. În acest fel, întârzierea pentru fiecare rang al transportului este dată de 3 porți, deci suma se obține după întârzierea a 4 porți (utilizând SAU EXCLUSIV).

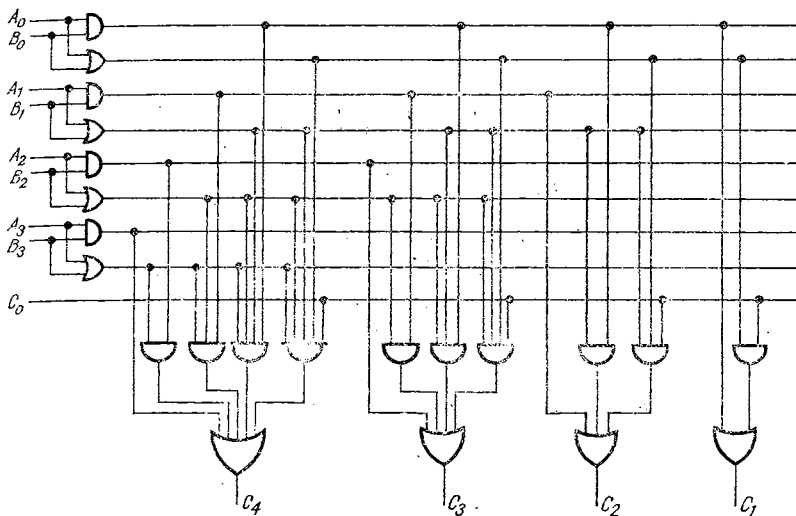


Fig. 2.60.

**Problema 2.92.** Să se realizeze cu porți AND, OR, XOR un scăzător complet de 1 bit definit de tabelul următor :

$C_{i-1}$	A	B	$D = A - B$	$C_i$
0	0	0	0	0
0	0	1	1	1
0	1	0	1	0
0	1	1	0	0
1	0	0	1	1
1	0	1	0	1
1	1	0	0	0
1	1	1	1	1

*Indicație*

Rezultă  $D = A \oplus B \oplus C_{i-1}$ ,  $C_i = \bar{A}B + (\bar{A} \oplus B)C_{i-1}$

**Problema 2.93.** Să se implementeze cu porți un circuit care să realizeze comandat suma și diferența a 2 biți.

### Indicație

Tabelul de adevăr pentru sumatorul binar complet este dat în problema 2.6, iar pentru scăzător în problema 2.92.

**Problema 2.94.** Să se realizeze cu sumatorul de 4 biți 7483 un sumator de 3 biți.

### Soluție

Vezi figura 2.61.

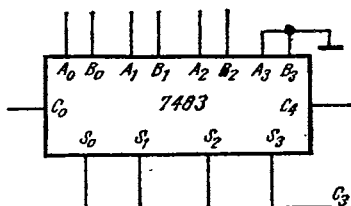


Fig. 2.61.

**Problema 2.95.** Realizați cu 7483, 2 sumatoare independente: unul de 2 biți și unul de 1 bit.

### Soluție

Vezi figura 2.62.

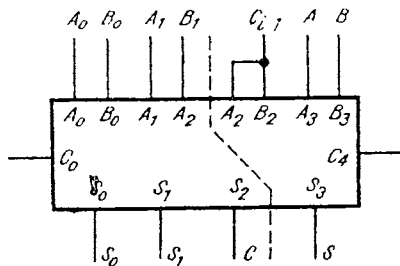


Fig. 2.62.

**Problema 2.96.** Să se realizeze cu două sumatoare de 2 biți și porți un sumator scăzător de 2 biți. Ce se întâmplă la ieșire dacă la intrare se aplică  $A, B, C_{i-1}$  complementate?

### Soluție

Vezi figura 2.63.

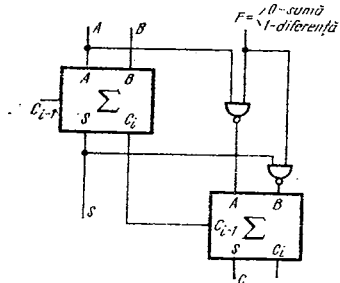


Fig. 2.63.

**Problema 2.97.** Să se găsească algoritmul de corecție zecimală a operației de adunare binară pe 4 biți.

*Soluție*

Vezi figura 2.64.

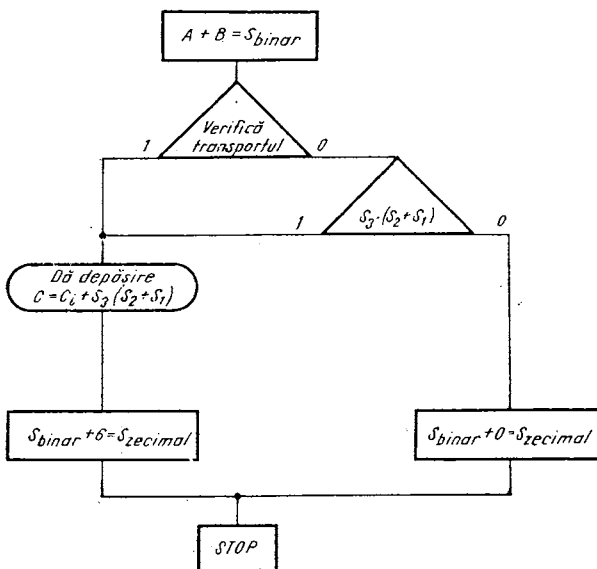


Fig. 2.64.

**Problema 2.98.** Să se găsească algoritmul de corecție zecimală a operației de scădere binară.

*Indicație*

Scăderea se realizează prin adunarea cu complementul, depășirea fiind dată de  $\bar{C}_4$  (transportul de la rangul cel mai semnificativ). Corecția se face prin sumarea cu numărul 10 (1010). Pentru obținerea complementului se utilizează circuite XOR.

**Problema 2.99.** Să se implementeze un sumator binar zecimal utilizând în principal două sumatoare binare 7483.

*Soluție*

Vezi figura 2.65.

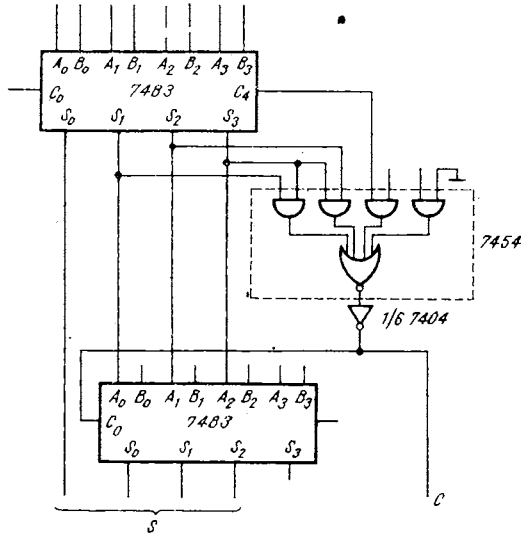


Fig. 2.65.

**Problema 2.100.** Să se realizeze un scăzător binar zecimal utilizând  $2 \times 7483$  și porți.

**Problema 2.101.** Să se realizeze un sumator-scăzător binar zecimal utilizând 7483 și porți.

**Problema 2.102.** Să se proiecteze un sistem logic combinațional care are 2 intrări de date  $A$  și  $B$  și 3 intrări de control  $C_0, C_1, C_2$ . Sistemul va produce comandat următoarele funcții booleene, depinzând de cele 8 combinații posibile ale intrărilor de control :

$$A + B, \overline{A + B}, A \cdot B, \overline{A \cdot B}, A \oplus B, \overline{A \oplus B}, 1, 0$$



**Problema 2.103.** Utilizând o unitate logico-aritmetică (ULA) 74181 să se realizeze un sistem care să efectueze următoarele operații :  $A$  plus  $B$ ,  $A$  minus  $B$ ,  $A \oplus B$ ,  $A + B$ ,  $A \cdot B$ ,  $\bar{A}$ , NU OPEREAZĂ, NU OPEREAZĂ. Comanda se va realiza cu ajutorul unui cod de 3 biți.

*Indicație*

Vezi figura 2.66.

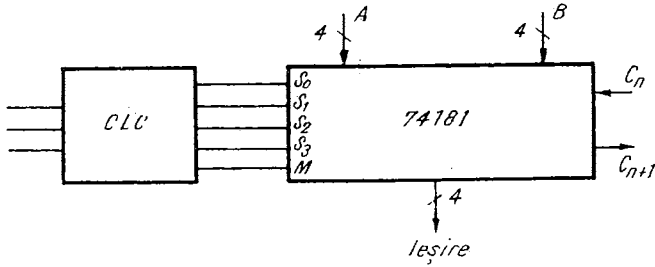


Fig. 2.66.

CLC este un circuit logic combinațional care transformă codul asociat celor 8 operații indicate în enunț, în cod de 5 biți asociat fiecăreia din cele 32 de operații efectuate de 74181. CLC se poate realiza cu 7442 (sau 74155) și cu porți.

**Problema 2.104.** Să se realizeze un circuit de multiplicare a 4 biți cu 3 biți utilizând  $2 \times 7483$  și porți.

*Rezolvare*

Fie cei 4 biți  $M_3M_2M_1M_0$  și cei 3 biți  $N_2N_1N_0$ . Regula de multiplicare binară este dată în exemplul :

$$\begin{array}{r}
 M_3M_2M_1M_0 \times N_2N_1N_0 \\
 1\ 1\ 0\ 1 \times 1\ 0\ 1 \\
 \hline
 1\ 1\ 0\ 1 \\
 0\ 0\ 0\ 0 \\
 1\ 1\ 0\ 1 \\
 \hline
 1\ 0\ 0\ 0\ 0\ 1
 \end{array}$$

Deci este vorba de înmulțiri cu 1 și cu 0 și adunarea rezultatelor deplasate.

Soluția este dată în figura 2.67. Întâi se multiplică  $M_3M_2M_1M_0$  cu  $N_0$  cu ajutorul porților SI, rezultatul fiind adunat cu produsul parțial  $M_3M_2M_1M_0 \times N_1$ , convenabil poziționat. Rezultatul se adună cu produsul parțial cu  $N_2$ .

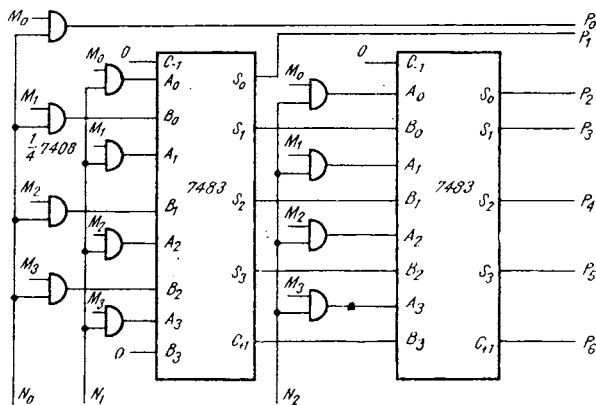


Fig. 2.67.

**Problema 2.105.** Să se explice ce este o memorie ROM și la ce se utilizează.

*Rezolvare*

Memoria ROM (Read Only Memory — „memorie numai cu citire”) numită și memorie fixă sau permanentă, este formată (figura 2.68) din două nivele de porți: ȘI-uri (în decodificator) și SAU-NU-uri (în matricea de memorie). DCD primește codurile de intrare în binar, (fie  $n$  numărul intrărilor) și activează pentru fiecare cod cite o ieșire din cele  $2^n$ . Aceste ieșiri se conectează sau nu la circuitul SAU-NU, realizându-se astfel memorarea lui 0, respectiv 1. Astfel pentru ieșirea  $\bar{O}_0$ , în figura 2.68 s-a explicat schema logică (cea electrică fiind mai complicată) a circuitului SAU-NU. Deci, în fiecare intersecție a matricei

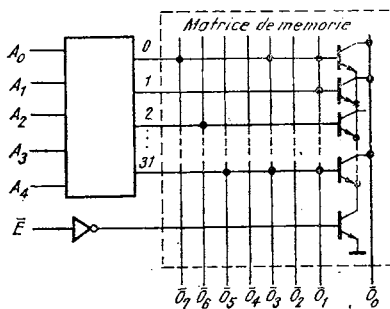


Fig. 2.68.

este plasat câte un tranzistor. La ROM-urile TTL, pentru memorarea lui 0 aceste tranzistoare se conectează ( $T_0, T_1, T_{31}$ ) la firul de ieșire și pentru memorarea lui 1 nu se conectează ( $T_2$ ). Pentru simplificare la ieșirile  $\bar{O}_1 - \bar{O}_7$  s-au reprezentat numai punctele de conectare. De exemplu, când se activează ieșirea 0 a DCD va apare la  $\bar{O}_7 - \bar{O}_0$  vectorul 01110100, numit și cuvânt. Se spune că memoria din figura 2.68 are memorate 32 cuvinte a câte 8 biți fiecare, deci 256 biți.

Vectorii de intrare se numesc adrese și reprezintă codurile în binar ale numerelor asociate fiecărui cuvânt.

Fie :  $n$  — numărul de biți ai vectorului de intrare (adresa) ;

$c$  — numărul de cuvinte memorate în ROM ;

$b$  — numărul de biți din fiecare cuvânt (este 4 sau 8 la majoritatea tipurilor de ROM realizate până acum).

Între  $n$  și  $c$  există relația  $c = 2^n$ . Deci vom întâlni încapsulate numai memorii ROM la care numărul cuvintelor este putere a lui 2.

Modul de organizare al ROM-ului este specificat prin produsul  $c \times b$ .

Capacitatea memoriei ROM se exprimă prin numărul total de biți memorati. Unitatea de măsură pentru aceștia este kilobitul (1 Kbit = = 1 024 biți).

La memoriile MOS, în fiecare intersecție din matrice sînt tranzistoare MOS cu oxid subțire, pentru stocare lui 0, sau cu oxid gros, pentru 1 logic.

Intrarea  $\bar{E}$  (Enable) permite ( $\bar{E} = 0$ ) sau inhibă ( $\bar{E} = 1$ ) funcționarea ROM-ului. În caz că memoria e dezactivată ( $\bar{E} = 1$ ), indiferent de adresare, la toate ieșirile se obține semnal logic 1.

Ieșirile  $\bar{O}_0 - \bar{O}_7$  pot fi „open collector“ sau „three state“, ceea ce permite punerea lor în paralel cu ieșirile altor memorii.

Memoriile ROM se utilizează în implementarea CLC, cu număr mare de intrări și ieșiri, care altfel s-ar realiza cu foarte multe porți.

Această aplicație este posibilă observînd că structura ROM constituie o generalizare a formei de implementare cu DCD și porți (vezi problema 2.34) : se decodifică pe nivelul de ȘI-uri toți termenii canonici (fiecare fir de cuvânt de la intrarea matricei reprezintă de fapt un termen canonic format din variabilele de intrare) și cei care intervin în expresia fiecărei funcții,  $\bar{O}_0 - \bar{O}_7$ , sînt adunați prin nivelul de SAU NU-uri. Deci lista de cuvinte a ROM-ului este chiar tabelul de adevăr al CLC.

#### Observație

La implementarea cu ROM nu este necesară minimizarea, deoarece sînt memorati în DCD toți termenii canonici și sînt incluse toate posibilitățile de apariție a acestora în funcția de ieșire.

Datorită faptului că pot memora tabele de adevăr foarte mari, de la  $32 \times 8$  la  $8192 \times 8$  biți, ROM-urile se utilizează în microprogramare, conversii de cod, generatoare de caractere ș.a.

**Problema 2.106.** Să se explice ce sînt memoriile PROM și REEPROM.  
*Rezolvare*

Memoriile PROM (Programmable ROM) sînt ROM-uri care se pot programa de către utilizatori. În general programarea se face prin selectarea fiecărei celule a matricei introducînd semnale cu anumiți parametri pe intrări și ieșiri.

Memoria REEPROM (REProgrammable ROM) poate fi ștearsă și reprogramată după utilizare.

Acste tipuri sînt mai avantajoase în experimentări față de memoriile ROM care sînt livrate de firme cu conținutul cerut de utilizator, conținut ce odată programat nu mai poate fi modificat.

**Problema 2.107.** Să se exprime în Kilobiți capacitatea unor memorii ROM organizate :

- a.  $256 \times 4$ ; b.  $512 \times 4$ ; c.  $512 \times 8$ ; d.  $1024 \times 4$ ; e.  $1024 \times 8$ ;  
 f.  $2048 \times 8$ .

Care este numărul biților de adresă în fiecare din aceste cazuri ?

*Soluții*

- a. 1 Kbiți, 8; b. 2 Kbiți, 9; c. 4 Kbiți, 9; d. 4 Kbiți, 10; e. 8 Kbiți, 10; f. 16 Kbiți, 11.

**Problema 2.108.** Să se imagineze o organizare internă de ROM de  $256$  cuvinte  $\times 4$  biți, la care matricea de memorie este de  $32 \times 32$  biți.

*Soluție*

Vezi figura 2.69

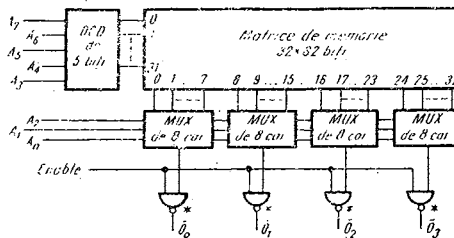


Fig. 2.69.

**Problema 2.109.** Cu ajutorul unor circuite integrate ROM de 256 biți, organizat în  $32$  cuvinte  $\times 8$  biți (vezi fig. 2.68) să se realizeze o memorie de  $128 \times 8$  biți (extinderea numărului de cuvinte).

### Rezolvare

Memoria de  $128 \times 8$  are de 4 ori mai multe cuvinte decît circuitul integrat. Deci vor fi necesare 4 CI pentru a o implementa.

Extinderea numărului de cuvinte ale memoriei se face conectînd cele 4 circuite integrate ca în figura 2.70. Fiecare circuit integrat se adresează cu 5 biți ( $A_0 - A_4$ ), conectați în paralel la toate ROM-urile

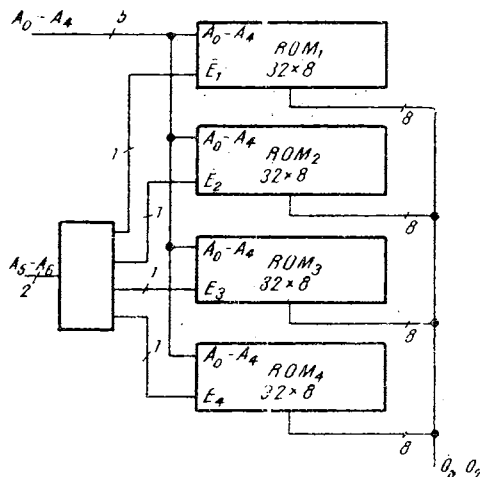


Fig. 2.70.

Pentru adresarea memoriei de  $128 \times 8$  mai sînt necesari încă 2 biți  $A_5$ ,  $A_6$  care, decodificați, activează doar cite unul din circuitele integrate ROM. Astfel la ieșirile cuplate în paralel citim pentru  $A_5A_6 = 00$ , conținutul  $ROM_1$ , pentru  $A_5A_6 = 01$ , conținutul  $ROM_2$  etc., celelalte trei circuite integrate fiind în acest timp inhibate.

În general pentru extinderea numărului de cuvinte al ROM :

- se calculează numărul circuitelor integrate necesare ;
- se conectează în paralel adresele și ieșirile circuitelor
- biții suplimentari de adresă se decodifică și selectează circuitele pe intrările  $\bar{E}$ .

**Problema 2.110.** Cu ajutorul unor circuite integrate de memorie ROM de 256 biți ( $32$  cuvinte  $\times 8$  biți) să se realizeze o extindere la  $64 \times 16$  biți (extinderea numărului de cuvinte și a numărului de biți pe cuvînt).

### Rezolvare

Extinderea numărului de cuvinte se face cu ajutorul intrării  $\bar{E}$ , activînd pentru  $A_5 = 0$ ,  $ROM_1$  și  $ROM_2$ , și pentru  $A_5 = 1$  celelalte

2 memorii (fig. 2.71). În acest caz, DCD (vezi problema 2.109) s-a redus la un inversor.

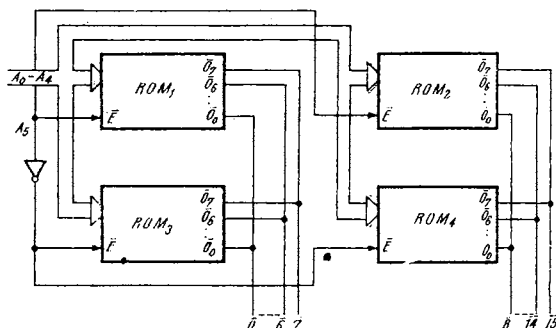


Fig. 2.71.

Extinderea lungimii cuvântului se face prin adăugarea ROM<sub>3</sub> și ROM<sub>4</sub> conectate în paralel cu ROM<sub>1</sub>, respectiv ROM<sub>2</sub> pe adrese și  $\bar{E}$ . Ieșirile ROM<sub>3</sub> și ROM<sub>4</sub> sînt conectate în paralel cu cele ale ROM<sub>1</sub> respectiv pentru a obține lungimea cuvântului de 16 biți.

**Problema 2.111.** Să se realizeze o memorie cu organizare 512 cuvinte  $\times$  4 biți :

- a) cu ROM-uri  $256 \times 4$  (extinderea numărului de cuvinte) ;
- b) cu ROM-uri  $32 \times 8$  (reducerea numărului de biți pe cuvînt).

*Indicație b)* Se poate reduce lățimea cuvântului concomitent cu mărirea numărului de adrese prin diverse procedee din care unul este prezentat în figura 2.72. Cu ajutorul MUX 74157 pentru  $A_5 = 0$  se transferă la ieșire  $\bar{O}_0 - \bar{O}_3$ , iar pentru  $A_5 = 1$ ,  $\bar{O}'_0 - \bar{O}'_3$ .

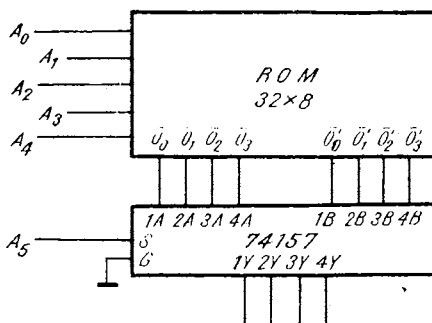


Fig. 2.72.

**Problema 2.112.** Să se realizeze o memorie fixă de  $2048 \times 4$  cu circuite de ROM organizate  $256 \times 4$ .

*Rezolvare*

Trebuie extins numărul de cuvinte al memoriei: pentru a trece de la 256 la 2048 cuvinte sînt necesare 8 circuite integrate conectate ca în figura 2.73. Ieșirile memoriilor,  $\bar{O}_0 - \bar{O}_3$  sînt legate în paralel. La fel și adresele  $A_0 - A_8$ . Dar memoria de  $2048 \times 4$  se adresează cu 11 biți. Restul de 3 biți  $\downarrow$ ,  $A_9 - A_{11} \downarrow$ , se decodifică. Ieșirile DCD 7442

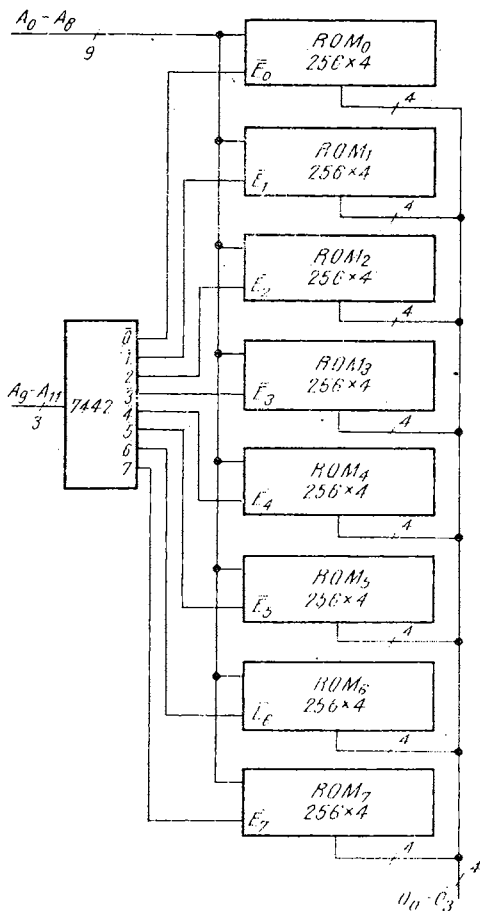


Fig. 2.73.

( $\bar{E}_0 \dots \bar{E}_7$ ) comandă activarea pe rind a circuitelor integrate astfel încât pe primele 256 adrese se citește ROM<sub>0</sub>, pe adresele 256–512 se citește ROM<sub>1</sub> ș.a.m.d.

**Problema 2.113.** Se consideră memoriile ROM din figura 2.74 și 2.75. Citind în ordine conținutul adreselor se constată că memorarea nu este corectă la adresa 641 bitul  $\bar{O}_6$ . Care circuit integrat trebuie înlocuit în fiecare din cele 2 memorii ?

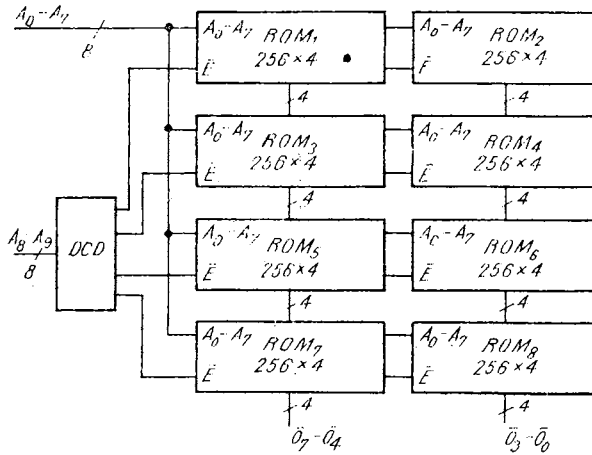


Fig. 2.74.

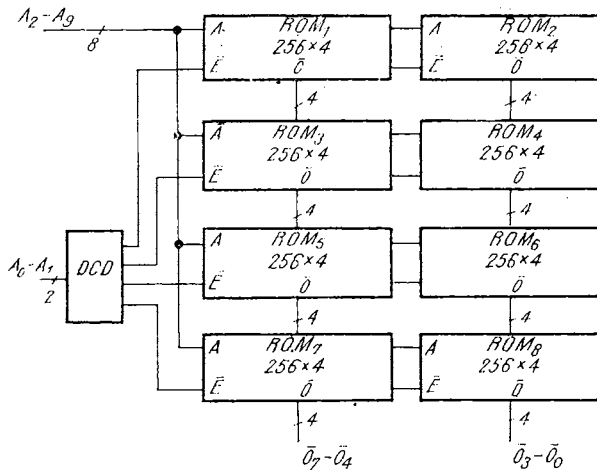


Fig. 2.75.



### Rezolvare

La memoria din figura 2.74 se decodifică la adresare biții cei mai semnificativi, deci prin baleierea în ordine crescătoare a adreselor, se citesc întâi ROM<sub>1</sub> și ROM<sub>2</sub>, apoi, între 256—512 se citesc simultan ROM<sub>3</sub> și ROM<sub>4</sub>, între 512 și 768, ROM<sub>5</sub> și ROM<sub>6</sub> și între 768—1024, ROM<sub>7</sub> și ROM<sub>8</sub>. Deci adresa la care se semnalează defectul, fiind între 512 și 768, indică drept defecte circuitele ROM<sub>5</sub> sau ROM<sub>6</sub>. Pentru a restrînge sfera de localizare a defectului ne orientăm și după ieșire:  $\bar{O}_6$  eronat indică pe ROM<sub>5</sub> drept circuit ce trebuie înlocuită. ROM<sub>6</sub> este bun deoarece biții  $\bar{O}_0 - \bar{O}_3$  care se citesc din el, apar corect.

Pentru memoria din figura 2.75, decodificarea se face pentru biții cei mai puțin semnificativi. Deci se vor citi adresele alternativ din fiecare circuit integrat: 0 (în general  $4n$ ) din ROM<sub>1</sub> și ROM<sub>2</sub>,  $1(4n+1)$  din ROM<sub>3</sub> și ROM<sub>4</sub>,  $2(4n+2)$  din ROM<sub>5</sub> și ROM<sub>6</sub>,  $3(4n+3)$  din ROM<sub>7</sub> și ROM<sub>8</sub>. Deci  $641(4n+1)_{n=160}$  se citește din ROM<sub>3</sub> și ROM<sub>4</sub>, iar bitul  $\bar{O}_6$  indică ROM<sub>3</sub> drept defect.

**Problema 2.114.** Să se reia problema anterioară pentru adresele:

- a) 512, bitul  $\bar{O}_1$ ;
- b) 975, bitul  $\bar{O}_4$ ;
- c) 273, biții  $\bar{O}_3$  și  $\bar{O}_6$ .

**Problema 2.115.** Să se realizeze un ROM de 32 Kbiți cu organizare de  $2048 \times 16$ :

- a) cu ROM de 16 Kb ( $2048 \times 8$ )
- b) cu ROM de 8 Kb ( $2048 \times 4$ )
- c) cu ROM de 8 Kb ( $1024 \times 8$ )
- d) cu ROM de 4 Kb ( $1024 \times 4$ )
- e) cu ROM de 4 Kb ( $512 \times 8$ )
- f) cu ROM de 2 Kb ( $512 \times 4$ ).

Să se compare soluțiile ca număr de circuite integrate și din punct de vedere al încărcării adreselor și al numărului de ieșiri cuplate în paralel.

**Problema 2.116.** Să se utilizeze un ROM de  $256 \times 4$  ca o memorie cu organizarea  $128 \times 8$ .

### Rezolvare

Este necesar să se citească memoria astfel încît să se obțină cuvinte mai puține, dar duble ca lungime. Pentru aceasta trebuie ca, la o adresare să se citească cite două cuvinte din ROM. Într-adevăr în schema din figura 2.76, ceasul cu 2 faze (T2 $\emptyset$ ) realizează pe faza întâi ( $\emptyset_1$ ) adresarea pentru  $A_0$ . Pentru  $\emptyset_1 = 1$  se citesc din ROM cuvintele de la adresele impare și, pe frontul negativ al  $\emptyset_1$  se înscriu

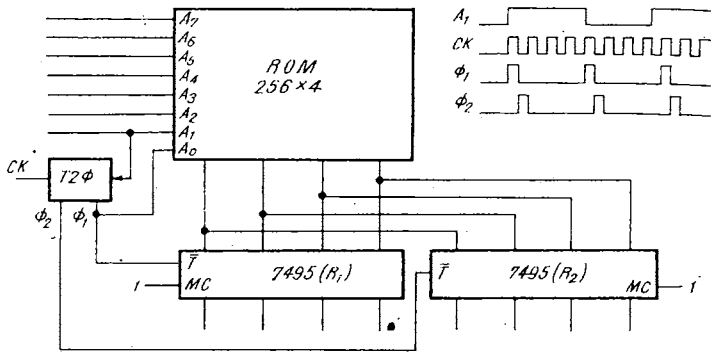


Fig. 2.76.

în registrul  $R_1$ . Pentru  $\Phi_1 = 0$  apar la ieșirile ROM cuvintele de la adresele pare, care se înscriu în  $R_2$  pe frontul negativ al  $\Phi_2$ . Astfel, la sfârșitul fazei  $\Phi_2$ , în registrele  $R_1$  și  $R_2$  se găsește înscris un cuvânt de 8 biți, citit de la adresa  $A_1 - A_8$ . Deci în exterior, ROM-ul apare ca avînd dimensiunea  $128 \times 8$ .

**Problema 2.117.** Să se implementeze cu ROM un CLC care realizează funcțiile :

$$\begin{aligned}
 f_1 &= P_0 + P_2 + P_4 + P_6 + P_{20} + P_{22} + P_{26} \\
 f_2 &= P_{13} + P_{14} + P_{15} + P_{30} \\
 f_3 &= P_1 + P_5 + P_{11} + P_{19} + P_{20} \\
 f_4 &= P_2 + P_7 + P_9 + P_{17} + P_{23} + P_{28} \\
 f_5 &= P_{25} + P_{26} + P_{27} + P_{28} + P_{29} \\
 f_6 &= P_0 + P_5 + P_{10} + P_{15} + P_{20} + P_{25} + P_{28} \\
 f_7 &= P_0 + P_3 + P_6 + P_9 + P_{12} + P_{15} + P_{18} + P_{21} + P_{24} + \\
 &+ P_{27} + P_{29} \\
 f_8 &= P_4 + P_8 + P_{12} + P_{16} + P_{20} + P_{24} + P_{26} + P_{29}
 \end{aligned}$$

*Rezolvare*

Implementarea cu ROM presupune următoarele etape :

- stabilirea dimensiunii memoriei necesare pentru aplicația respectivă ;
- alegerea tipurilor de circuite ROM cele mai potrivite : cu dimensiuni identice sau cît mai apropiate de cele stabilite anterior ;
- dacă nu există memorii ROM cu dimensiuni identice cu cele dorite se fac transformări de dimensiuni cu metodele prezentate în problemele 2.109—2.112 și 2.115—2.116 ;
- stabilirea tabelului de adevăr al ROM ;

— reducerea dimensiunii ROM, atunci cînd este posibil, utilizînd metode ca în problemele 2.125—2.129.

Implementarea cu ROM se face urmărind două scopuri :

- utilizarea unui număr minim de circuite integrate ;
- folosirea integrală a capacității memoriei.

Uneori aceste deziderate pot fi atinse simultan (vezi problema 2.117), alteori trebuie realizat un compromis în favoarea unuia dintre ele (vezi problemele 2.119, 2.121).

În cazul particular al problemei 2.117 se observă că termenul canonic de rang cel mai mare este  $P_{30}$ . Deci sînt necesare 5 intrări în memoria ROM, pentru a obține din DCD primii 31 termeni canonici. Trebuie implementate 8 funcții : înseamnă că ROM-ul va avea organizarea  $32 \text{ cuvinte} \times 8 \text{ biți}$ . Există astfel de circuite integrate și este necesar unul singur pentru a rezolva problema.

Tablelul de adevăr al ROM se stabilește pornind de la formele canonice ale funcțiilor negate ca în modelul următor :

	$A$ ( $2^4$ )	$B$ ( $2^3$ )	$C$ ( $2^2$ )	$D$ ( $2^1$ )	$E$ ( $2^0$ )	$\bar{f}_1$	$\bar{f}_2$	$\bar{f}_3$	$\bar{f}_4$	$\bar{f}_5$	$\bar{f}_6$	$\bar{f}_7$	$\bar{f}_8$
0	0	0	0	0	0	0	1	1	1	1	0	0	1
1	0	0	0	0	1	1	1	0	1	1	1	1	1
2	0	0	0	1	0	0	1	1	0	1	1	1	1
3	0	0	0	1	1	1	1	1	1	1	1	0	1
						$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
30	1	1	1	1	0	1	0	1	1	1	1	1	1
31	1	1	1	1	1	1	1	1	1	1	1	1	1

Implementarea CLC este dată în figura 2.77 în care ROM-ul are programat în interior acest tabel de adevăr. Deoarece ieșirile ROM sînt negate se obțin în exterior funcțiile

$$\bar{f}_K = f_K (K = 1 \div 8)$$

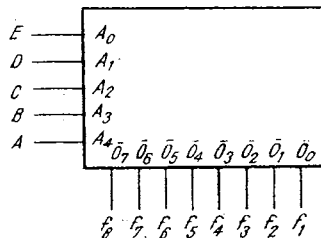


Fig. 2.77.

**Problema 2.118.** Să se realizeze cu ajutorul unui ROM un decodificator BCD/7 segmente cu caracterele ca în figura 2.78.

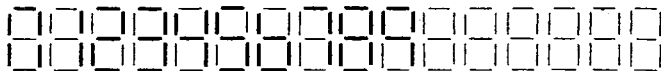


Fig. 2.78.

*Indicație*

ROM-ul trebuie să aibă 4 intrări și 7 ieșiri. Se utilizează cel mai mic ROM existent, cu dimensiunea de  $32 \times 8$  biți care va fi folosit la mai puțin de jumătate din capacitate. Implementarea cu ROM în acest caz este neeconomică.

**Problema 2.119.** Să se definească structura și conținutul unui ROM ce comandă afișarea prin puncte ( $5 \times 7$ ) a caracterelor alfabetice.

*Indicație*

Pe o matrice de  $5 \times 7$  puncte se pot realiza toate caracterele alfabetice ca în exemplele din figura 2.79.

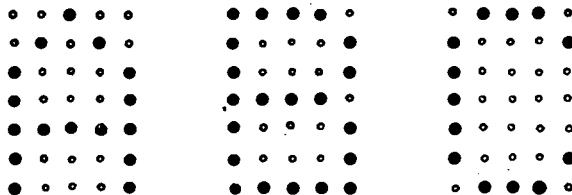


Fig. 2.79.

Sînt 26 caractere alfabetice care trebuie memorate, fiecare avînd 7 rînduri de puncte: sînt deci necesare 182 cuvinte de cîte 5 biți fiecare.

Implementarea se poate face cu :

- a) 3 ROM de  $64 \times 8$  biți ; b) 1 ROM de  $256 \times 8$  biți etc.

Soluțiile a) și b) indică cele două extreme posibile în implementare :

- obținerea umplerii ROM-urilor cît mai aproape de capacitatea lor (pentru a) ;
- obținerea numărului minim de circuite integrate (pentru b).

**Problema 2.120.** Să se determine organizarea unui ROM care generează toate produsele a 2 numere de 4 biți (deci care realizează multiplicarea binară).

**Problema 2.121.** Să se implementeze cu ROM un translator de cod din binar (8 biți) în BCD.

*Soluție*

Dimensiunea ROM-ului va fi  $256 \times 10$  biți. Implementarea cu utilizarea maximă a capacității de memorie se face cu 3 ROM de  $256 \times 4$ , iar cea cu număr minim de circuite integrate constă în 2 ROM de  $256 \times 8$  biți.

**Problema 2.122.** Să se implementeze cu ROM un circuit care realizează înmulțirea  $M \times N = P$  unde  $M, N, P$  sînt exprimate în BCD și  $M, N$  sînt cuprinse între 0 și 9.

*Rezolvare*

Memoria necesară are 8 intrări și 8 ieșiri (numărul 81 fiind cel mai mare ce poate apare ca rezultat). Implementarea se poate face cu un ROM de  $256 \times 8$  biți, ca în figura 2.80. Registrul de adrese este format din 2 secțiuni: una conținînd pe  $M$ , cealaltă pe  $N$ .

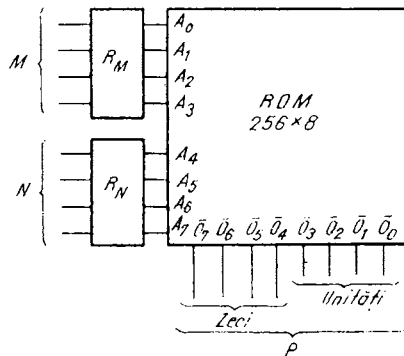


Fig. 2.80.

**Problema 2.123.** Să se implementeze cu ROM un generator al funcției  $\sin x$ ,  $x$  variînd între 0 și  $\pi/2$  cu pași de  $\pi/512$ . La ieșire trebuie obținute 4 zecimale exacte în cod BCD.

*Indicație*

Dimensiunea ROM este  $256 \times 16$  biți.

**Problema 2.124.** Să se determine organizarea unui ROM generator de 64 caractere alfanumerice de  $5 \times 7$  puncte (vezi figura 2.79) care furnizează la ieșire:

- liniile de puncte orizontale (cîte 5 puncte);
- coloanele de puncte verticale (cîte 7 puncte).

*Soluție.*

- a)  $64 \times 7$  cuvinte a câte 5 biți pe cuvint ;
- b)  $64 \times 5$  cuvinte a câte 7 biți pe cuvint.

**Problema 2.125.** Un CLC primește pe intrări numere de 8 biți, unele în cod Gray, altele în BCD cu exces 3 și trebuie să furnizeze la ieșire numai numere în cod binar. Să se implementeze cu ROM acest CLC știind că nu pot apare simultan mai multe numere pe intrări.

*Rezolvare*

Organizarea ROM-ului este dată în figura 2.81. Pentru implementare sînt necesare 16 ROM-uri de  $8192 \times 8$  biți și un DCD.

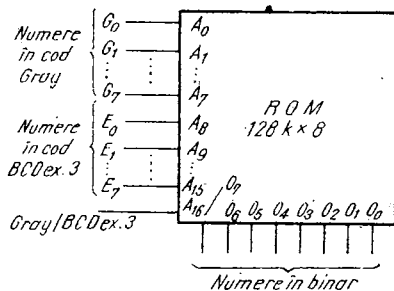


Fig. 2.81.

Deoarece nu pot apare simultan pe intrări numere din cele 2 grupe se poate reduce dimensiunea ROM-ului cu schema din figura 2.82. Bitul cuplat pe A<sub>8</sub> care indică în ce cod este numărul sosit pe intrări comandă trecerea acestuia prin MUX către memorie. În acest fel, prin adăugarea doar a 2 MUX s-a redus foarte mult capacitatea ROM-ului necesar : de  $2^8 = 256$  ori.

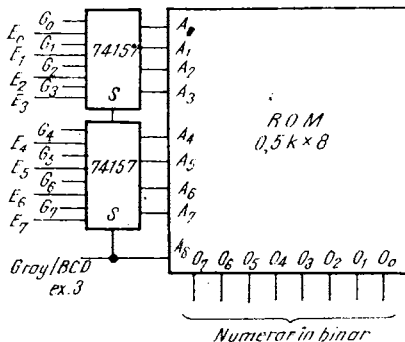


Fig. 2.82.

**Problema 2.126.** Să se realizeze cu ROM minim un convertor al numerelor 0—9 din cod zecimal în cod Gray.

*Indicație*

ROM-ul trebuie să aibă 10 intrări și 4 ieșiri, deci  $1024 \times 4$  biți.

Pentru reducere se poate realiza întâi o codificare din zecimal în binar cu  $1 \times 74148$  și  $1 \times 7400$  (vezi schema din figura 2.43) și apoi se utilizează un ROM de  $16 \times 4$  biți pentru codificarea binar-Gray. Reducerea ROM-ului este atât de însemnată încît utilizarea unui circuit organizat  $32 \times 8$  este neeconomică deoarece se pierde 75% din capacitatea acestuia în mod inutil. Este de preferat o implementare cu CI SSI sau MSI (pr. 2.69).

**Problema 2.127.** Să se explice funcționarea schemei din figura 2.83. De cîte ori s-a redus dimensiunea ROM-ului prin cuplarea MUX la intrare ?

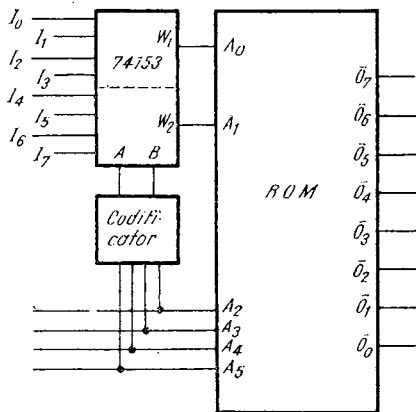


Fig. 2.83.

*Rezolvare*

Adresele  $A_2$ — $A_5$  codificate comandă multiplexarea căilor  $I_0$ — $I_7$  pe  $A_0$  și  $A_1$ . ROM-ul are dimensiunea  $64 \times 8$  biți. Dacă nu era cuplat MUX la intrare ar fi fost necesar un ROM de  $4096 \times 8$  deci de  $\varnothing 64$  de ori mai mare.

**Problema 2.128.** Să se implementeze cu ROM un convertor de cod din binar (8 biți) în zecimal.

*Rezolvare*

ROM-ul are 8 intrări și 22 de ieșiri (10 pentru unități, 10 pentru zeci și 2 pentru sute) deci  $256 \times 22$  biți și se poate implementa cu 3 ROM-uri de  $256 \times 8$  biți.

Cu schema din figura 2.84 se realizează cu ROM conversia din binar în BCD și apoi cu ajutorul DCD 7442, conversia BCD — zecimal. Se reduce astfel capacitatea ROM-ului la  $256 \times 10$  biți, deci de 2,2 ori. Implementarea se poate face cu 3 ROM-uri de  $256 \times 4$  biți sau 2 ROM-uri de  $256 \times 8$  biți.

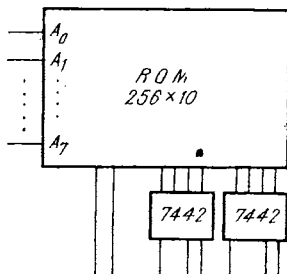


Fig. 2.84.

**Problema 2.129.** Cu cât s-a redus dimensiunea ROM-ului în schema din figura 2.85, datorită cimpurilor codificate ce apar la ieșire ?

*Soluție*

$$52/16 = 3,2.$$

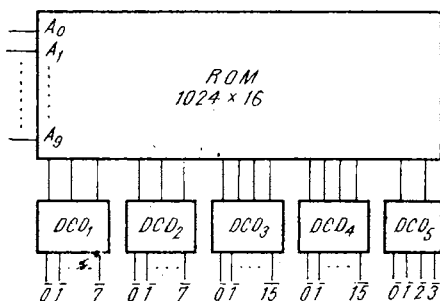


Fig. 2.85.

**Problema 2.130.** Să se explice ce este un PLA (Programmable Logic Array). Dar un FPLA ?

*Rezolvări.*

PLA este o structură LSI conținând o matrice de ȘI-uri și o matrice de SAU-uri. Conexiunile la ȘI-uri sau SAU-uri se realizează pr-



gramabil în procesul de fabricație, sau de către utilizator conform aplicației concrete. PLA este o structură mobilă și se utilizează eficient pentru sisteme cu mai mult de 8 variabile de intrare. În figura 2.86 dăm o reprezentare schematică a unui PLA cu  $N$  intrări,  $M$  ieșiri și  $L$  termeni elementari realizabili. (Ex. : 82S100 este un PLA cu  $N = 16$ ,  $L = 48$ ,  $M = 8$ ).

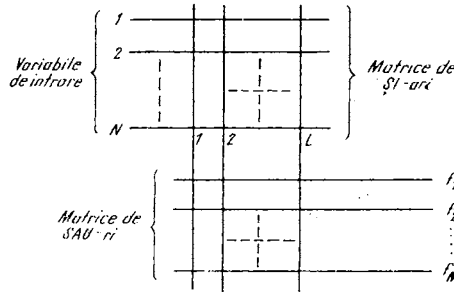


Fig. 2.86.

Spre deosebire de ROM la care matricea de ȘI-uri are o configurație fixă, este un decodificator al tuturor cuvintelor de intrare, fiind programabil doar nivelul de SAU-uri, la PLA este programabil și primul nivel. În altă interpretare spunem că nivelul de ȘI-uri la ROM furnizează toți termenii canonici posibil de construit ca variabilele de intrare, iar prin nivelul de SAU-uri programabile se selectează acei termeni ce intervin în expresia fiecărei funcții.

La PLA este programabil și nivelul de ȘI-uri, deci este posibilă și necesară o implementare pornind nu de la termenii canonici, ci de la cei elementari, constituienți ai formei minimizezate.

Tehnologic se impune ca ambele niveluri să fie realizate cu circuite de tip ȘI-NU. De asemenea poate fi programată și complementarea sau necomplementarea ieșirii.

FPLA (Field PLA) este un PLA care poate fi programat de către utilizator.

**Problema 2.131.** Să se explice cum se programează un FPLA cu care se implementează sistemul de funcții :

$$\begin{aligned}
 f_1 &= A_0 \bar{A}_1 A_{14} + \bar{A}_2 \bar{A}_{13} \bar{A}_{14} A_{15} \\
 f_2 &= A_1 A_2 + \bar{A}_1 \bar{A}_{14} + A_0 \bar{A}_{15} \\
 &\dots \\
 f_3 &= A_0 \bar{A}_1 A_{14} + A_{14} \bar{A}_{15}
 \end{aligned}$$

### Rezolvare

Se observă că funcțiile sînt date în forma elementară. Este necesar 1 FPLA 82S100 reprezentat în figura 2.87. Punctele pe primul nivel indică variabilele  $A_K$ , respectiv  $\bar{A}_K$  care se cuplează la ȘI-uri. Aceste ȘI-uri implementează de fapt termenii elementari ce intervin în expresiile funcțiilor. Punctele de pe al II-lea nivel arată care din termenii elementari se cuplează la SAU-ul respectiv. Se observă că termenul  $A_0\bar{A}_1A_{14}$  intervine și în  $f_1$  și în  $f_8$  deci acesta se poate forma doar o dată, la ȘI-ul  $L_0$ .

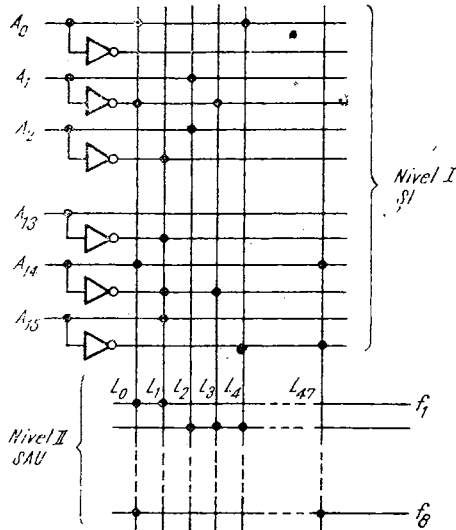


Fig. 2.87.

### Observație

La implementarea cu PLA există o restricție datorată limitării numărului de termeni  $L$  distincți ai grupului de funcții. Minimizarea, în acest caz, urmărește obținerea cît mai multor termeni comuni per ansamblu și nu este întotdeauna identică cu minimizarea matematică prezentată în capitolul 1.

**Problema 2.132.** Utilizînd circuite PLA cu  $N$  intrări,  $L$  termeni interni și  $M$  ieșiri să se realizeze scheme de:

- extinderea numărului de termeni interni, păstrînd același numărul de intrări și ieșiri;
- extinderea numărului de ieșiri,  $N$  rămînînd același.

*Rezolvare*

a) Cu schema din figura 2.88 în care PLA-urile sînt cuplate în paralel pe intrări și ieșiri se obțin de  $n$  ori mai mulți termeni interni.

b) Dacă se decuplează ieșirile din paralel în schema din figura 2.88 se obține un PLA cu  $n \times M$  ieșiri și  $n \times L$  termeni interni.

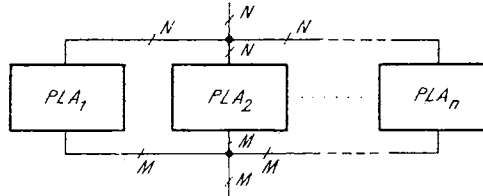


Fig. 2.88.

**Problema 2.133.** Să se implementeze cu PLA sistemul logic combinațional definit de funcțiile :

$$f_1 = (FA + \bar{F}\bar{A})(GB + \bar{G}\bar{B})(HC + \bar{H}\bar{C})(ID + \bar{I}\bar{D})$$

$$f_2 = \bar{E}I + E$$

$$f_3 = \bar{E}\bar{I}$$

$$f_4 = \bar{E}HI + E$$

$$f_5 = \bar{E}\bar{H}I$$

$$f_6 = \bar{E}GHI + E$$

$$f_7 = \bar{E}\bar{G}HI$$

$$f_8 = EFGHI + E$$

Să se calculeze ROM-ul echivalent cu acest PLA.

*Indicație*

Implementarea se simplifică dacă se va realiza funcția :  $f_1 = \bar{F}A + F\bar{A} + \bar{G}B + G\bar{B} + \bar{H}C + H\bar{C} + \bar{I}D + I\bar{D}$  (vezi problema 1.21).

PLA-ul trebuie să aibă 9 intrări ( $A-I$ ) și 8 ieșiri ( $f_1-f_8$ ). Cu 82 S100 implementarea se poate face direct cu formele funcțiilor date, utilizînd 16 termeni din cei 48 posibili. Se va desena configurația internă a legăturilor.

ROM-ul cu care s-ar putea realiza SLC-ul definit de funcțiile  $f_1-f_8$  are  $512 \times 8$  biți.

**Problema 2.134.** Să se implementeze cu PLA un SLC care realizează sumarea și scăderea a două numere în BCD (cuprinse între 0—9).

*Indicație*

PLA-ul va avea 9 intrări și 5 ieșiri.

**Problema 2.135.** Să se implementeze cu PLA sistemul logic combinațional definit în problema 2.102,  $A$  și  $B$  fiind numere binare de câte 8 biți fiecare.

*Indicație*

SLC realizează următoarele funcții de 2 biți,  $A$  și  $B$ :

$C_2$	$C_1$	$C_0$	funcție
0	0	0	$A + B$
0	0	1	$\overline{A + B} = \overline{A} \overline{B}$
0	1	0	$A \cdot B$
0	1	1	$\overline{A \cdot B} = \overline{A} + \overline{B}$
1	0	0	$A \oplus B = \overline{A} B + A \overline{B}$
1	0	1	$\overline{A \oplus B} = \overline{A} B + A \overline{B}$
1	1	0	1
1	1	1	0

Deoarece  $A(A_0 A_1 \dots A_7)$  și  $B(B_0, B_1 \dots B_7)$  sînt numere binare de 8 biți, PLA-ul va avea 16 intrări și 8 ieșiri. Fiecare ieșire va fi definită de funcția:

$$f_K = \overline{C_2} \overline{C_1} \overline{C_0} A_K + \overline{C_2} \overline{C_1} \overline{C_0} B_K + \overline{C_2} \overline{C_1} C_0 \overline{A}_K \overline{B}_K + \overline{C_2} C_1 \overline{C_0} A_K B_K + \\ + \overline{C_2} C_1 C_0 \overline{A}_K + \overline{C_2} C_1 C_0 \overline{B}_K + C_2 \overline{C_1} \overline{C_0} A_K \overline{B}_K + C_2 \overline{C_1} \overline{C_0} \overline{A}_K B_K + \\ + C_2 \overline{C_1} C_0 A_K B_K + C_2 \overline{C_1} C_0 \overline{A}_K \overline{B}_K + C_2 C_1 \overline{C_0}$$

în care  $K = 0, 1, \dots, 7$ . Fiecare funcție introduce 11 termeni distincți. Este deci necesar un PLA cu 16 intrări, 8 ieșiri și 88 de termeni interni.

Se vor cupla  $2 \times 82S100$  ca în figura 2.88 în vederea dublării numărului de termeni interni.

### Prezentarea conținutului

În prima sa parte, capitolul își propune să prezinte detaliat amănunte privind funcționarea logică a tipurilor uzuale de circuite logice secvențiale.

Problemele 3.1—3.12 și 3.22—3.24 se referă la diversele categorii de *bistabili* (RS, RSC, JKMS, T, D, tampon-„latch“ ș.a.) atrăgînd atenția asupra particularităților și diferențelor specifice ale acestora.

*Monostabilii* recomutabili (retriggerable) și nerecomutabili ca și unele aplicații ale lor sînt prezentați în problemele 3.13—3.21.

O atenție mai mare este acordată *numărătoarelor* (3.25—3.59) sintetizate din bistabili (3.25—3.28) sau MSI. Sînt reprezentate numărătoare zecimale cu aplicații în afișaje cu LED-uri și conversie BCD-binar.

Numărătorul binar 74193 este analizat în privința implicațiilor pe care le are faptul că încărcarea paralelă (LD) are loc pe palierul impulsului, deci nesincron față de numărare. Sînt prezentate următoarele aplicații :

- reducerea stărilor numărătorului prin reacție directă BR-LD, prin reacție cu bistabil RS sau prin reacție sincronă (3.31—3.37) ;
- ceasuri multifazice (problemele 3.39—3.40) ;
- conectarea în cascadă a numărătoarelor.

Se insistă asupra deosebirilor între numărătoarele sincrone și asincrone. Utilizînd diverse tipuri de numărătoare se dau scheme realizînd :

- divizarea frecvenței ;
- generarea unor forme de undă ;
- conversia D/A ;
- înmulțirea și împărțirea secvențială a două numere ș.a.

*Registrele* sînt reprezentate în special prin circuitul 7495 a cîrui funcționare este analizată independent sau în diverse scheme ca numărătoare în inel, Johnson ș.a. Se dau exemple de proiectare a unor scheme cu autoinițializare și autocorecție a erorilor (problemele 3.65—3.70) indicîndu-se metode de transformare a unor circuite care, în funcție de starea inițială pot evolua în cicluri repetitive diferite, în unele cu graf unic. Sînt prezentate de asemenea și următoarele aplicații cu registrul 7495 :

- deplasare stînga ;
- generatoare de diverse forme de undă (problemele 3.75—3.76) ;
- divizoare de impulsuri ;
- generatoare de coduri corectoare de erori și succesiuni pseudoaleatoare (problemele 3.79—3.83) ;
- convertoare : binar-BCD, serie-paralel, paralel-serie ;
- înmulțirea și împărțirea cu un polinom dat (scheme utilizabile de obicei în prelucrarea serială a datelor, la testarea erorilor).

*Memoriile RWM (RAM)* sînt analizate în funcție de diversele tehnologii de realizare : TTL, MOS statice și dinamice.

Este descrisă structura internă pentru memorii cu diferite organizări.

Sînt indicate metode de extindere a capacității memoriei : a numărului de cuvinte și a numărului de biți pe cuvînt (problemele 3.98—3.103).

Se insistă asupra diferențelor între memoriile MOS statice și dinamice și între diversele tipuri de memorii dinamice.

Sînt prezentate mai detaliat memoriile MOS dinamice de  $16K \times 1$  biți (3.106—3.111).

Partea a doua a capitolului se referă la *metodele de proiectare* a SLSS (problemele 3.112—3.140).

Pentru a înțelege și aplica aceste metode, cititorul trebuie să stăpînească perfect aspectele teoretice parcurse de la începutul lucrării pînă în acest punct ; este necesar să cunoască modelarea matematică a SLSS (capitolul 1), proiectarea componentelor acestora : CLC (capitolul 2) și bistabili, registre (capitolul 3).

Este exemplificată proiectarea cu implementarea CLC cu diferite tipuri de circuite SSI, MSI și LSI (ROM, PLA), iar cu cea a registrului, cu bistabili D, JKMS sau 7495. De asemenea se ilustrează implementări de sisteme definite prin cele mai diverse modalități : graf, organigrame, forme de undă, tabele de tranziții etc.

Este atinsă și problematica legată de testări înlănțuite ale condițiilor de intrare, reducerea CLC asociat sistemului, realizabilitatea fizică a sistemelor, eliminarea hazardului..

**Problema 3.1.** Să se descrie prin diagrame de timp funcționarea bistabilului RS din figura 3.1.

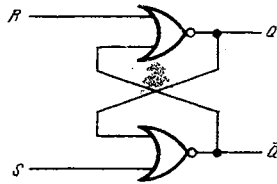


Fig. 3.1.

*Rezolvare*

În figura 3.2 se dau formele de undă ce caracterizează intrările

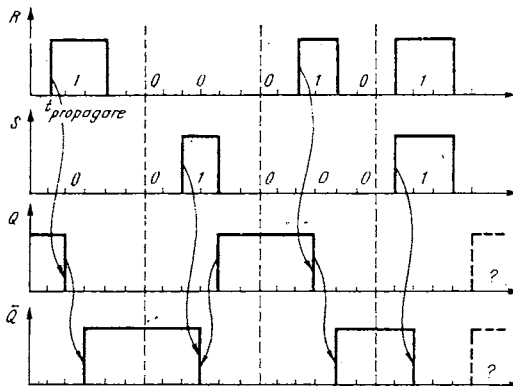


Fig. 3.2.

și ieșirile circuitului. Se observă că  $R = 1$  (când  $S = 0$ ) determină trecerea lui  $Q$  în 0 ( $\bar{Q} = 1$ ) și  $S = 1$  produce comutarea lui  $Q$  în 1 ( $\bar{Q} = 0$ ). Spunem că  $R$  (Reset) comandă aducerea la 0 a lui  $Q$ ,  $S$  (Set) pune  $Q$  în 1 și  $R, S$  sînt active pe 1.

Dacă  $R$  și  $S$  sînt simultan pe 1, ambele ieșiri trec în 0. Apoi, comutarea simultană a intrărilor din 1 în 0, determină o incertitudine la ieșiri: sînt comandate ambele porți SAU NU să treacă în 1 și răspunsul în  $Q$  și  $\bar{Q}$  va fi dependent de timpii de propagare ai porților. SAU NU-ul cu timp mai mic de propagare ajunge primul la nivel logic 1, forțîndu-l pe celălalt, prin reacție să treacă în 0.

Configurația de intrare  $RS = 11$  determină intrarea bistabilului într-o stare de incertitudine, numită astfel din două motive:

1. Pentru  $RS = 11$ , circuitul nu mai respectă proprietatea bistabililor de a avea valori logice inversate la ieșiri.

2. Vectorul de intrare  $RS = 11$  nu poate fi succedat de  $RS = 00$ , deoarece, în acest caz nu se poate preciza starea în care va comuta bistabilul ( $Q\bar{Q} = 10$  sau  $Q\bar{Q} = 01$ ).

**Problema 3.2.** Să se descrie prin diagrame de timp și tabel de adevăr funcționarea schemei din figura 3.3. Care este relația ce descrie funcționarea circuitului?

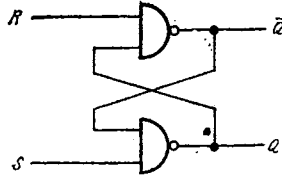


Fig. 3.3.

*Indicație:*  $Q = \bar{S} + RQ$ .

Bistabilul RS realizat cu porți ȘI NU funcționează asemănător cu cel cu porți SAU NU:  $R$  și  $S$  au aceleași semnificații comandând trecerea lui  $Q$  în 0, respectiv în 1 (dar  $Q$  și  $\bar{Q}$  apar inversat față de figura 3.1). Deosebirea constă în faptul că  $R$  și  $S$  sînt active pe 0 și starea nedeterminată apare pentru  $RS = 00$ .

**Problema 3.3.** Stabiliți tabelul de adevăr și diagramele de timp pentru bistabilul RS cu ceas (clock), RSC din figura 3.4.

**Problema 3.4.** Să se descrie funcționarea bistabilului D, obținut din RSC prin cuplarea intrărilor  $\bar{R} = S = D$ .

**Problema 3.5.** Să se deseneze diagramele de timp ce descriu funcționarea bistabilului MS (Master Slave) format din 2 bistabili RSC ca în figura 3.5.

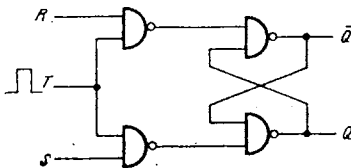


Fig. 3.4.

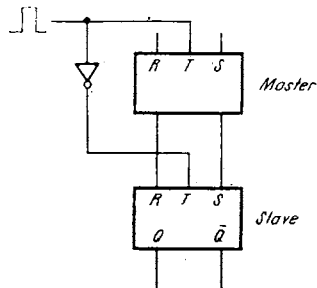


Fig. 3.5.



Să se explice de ce la acest tip de bistabil, spre deosebire de cele anterior descrise, modificarea ieșirii  $Q$  se face pe front, nu pe palierul impulsului de ceas.

**Problema 3.6.** a) Să se stabilească tabelul de tranziții al bistabilului JK din figura 3.6.

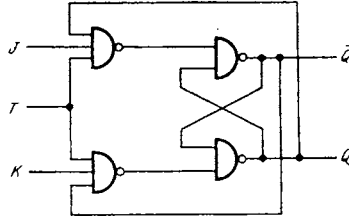


Fig. 3.6.

b) Să se scrie funcția ce definește tranziția stărilor.

c) Să se explice de ce se înlătură cu această schemă starea nedeterminată a bistabilului RS.

*Soluție*

$$b) Q_{t+1} = \bar{Q}_t J + Q_t \bar{K}$$

**Problema 3.7.** Să se deseneze schema bistabilului JKMS realizat cu porți ȘI NU (NAND). Câte circuite integrate sînt folosite ?

**Problema 3.8.** Să se realizeze un bistabil de tip T :

$$Q_{t+1} = \bar{T}Q + T\bar{Q}$$

a) cu JKMS 7473

b) cu un bistabil D 7474 folosind ieșirea  $\bar{Q}$

c) cu un bistabil D la care  $\bar{Q}$  nu este accesibilă.

*Soluție*

a) Conectînd  $J$  și  $K$  la 1 logic, bistabilul se transformă în bistabil de tip T (își schimbă starea la fiecare impuls de tact, pe frontul negativ).

b) Se conectează  $\bar{Q}$  la D și bistabilul devine T, comutînd pe frontul pozitiv.

**Problema 3.9.** Realizați un bistabil de tip D

a) cu bistabili RS

b) cu un bistabil JKMS, 7473

c) cu un bistabil T

*Soluție*

a) vezi problema 3.4.

b) Intrarea D va fi J, cuplat cu K printr-un inversor ( $D = J = \bar{K}$ ).

**Problema 3.10.** Să se explice de ce se utilizează bistabilul RS ca tampon pentru comutatoare bipolare (figura 3.7).

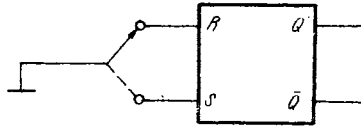


Fig. 3.7.

*Rezolvare*

La închiderea contactelor mecanice se obțin numeroase impulsuri pe durate de ordinul milisecundelor. Bistabilul RS oprește aceste impulsuri să se transmită mai departe, generând un singur impuls la ieșire.

**Problema 3.11.** Să se proiecteze cu bistabili D un bistabil multiplu cu un singur stăpîn (master) și mai mulți sclavi (slaves). Circuitul va avea o intrare și mai multe ieșiri adresabile.

*Soluție.* Considerăm cazul a 8 ieșiri adresabile  $0_0 \dots 0_7$ , prezentat în figura 3.8.

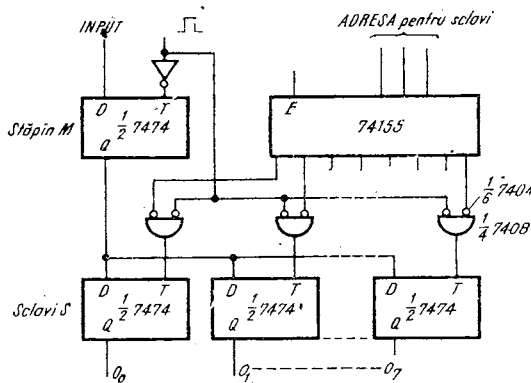


Fig. 3.8.

S-a ținut cont că bistabilii D (7474) comută pe frontul pozitiv.

**Problema 3.12.** Pentru schema din figura 3.9 să se deducă formele de undă la ieșire pentru 8 impulsuri de tact, dacă inițial :

a)  $Q_1 = 1, Q_2 = 1$

b)  $Q_1 = 1, Q_2 = 0$

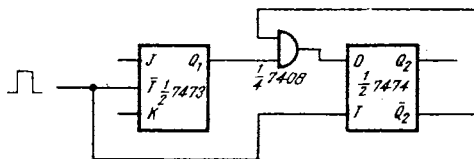


Fig. 3.9.

*Rezolvare*

a) Formele de undă se dau în figura 3.10. Se observă că,  $Q_1$  își inversează valoarea logică pe fiecare front negativ ( $\bar{T}$ ) al impulsurilor

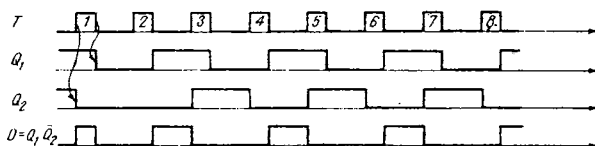


Fig. 3.10.

de tact. Bistabilul  $D$  înscrie în  $Q_2$ , pe fiecare front pozitiv al tactului ( $T$ ), ceea ce găsește la ieșire circuitului ȘI.

Se observă că circuitul este un numărător de 2 biți în cod Gray, formele de undă repetându-se din 4 în 4 impulsuri de tact.

b) Formele de undă apar deplasate: originea axelor este în pauza dinaintea impulsului 3.

**Problema 3.13.** În figura 3.11 se dă schema logică internă a monostabilului 74121.

a) Să se descrie funcționarea prin diagrama de stări în punctele indicate pe schemă;

b) Să se stabilească ecuația ce definește condiția de comutare;

c) Știind că durata stării cvasistabile a monostabilului este  $T = 0,69 R_t C_t$ , să se dimensioneze  $R_t$  și  $C_t$  pentru  $T = 50 \mu s$ ;

d) Ce se întâmplă dacă monostabilul primește pe intrare mai multe semnale de comandă în timpul stării cvasistabile?

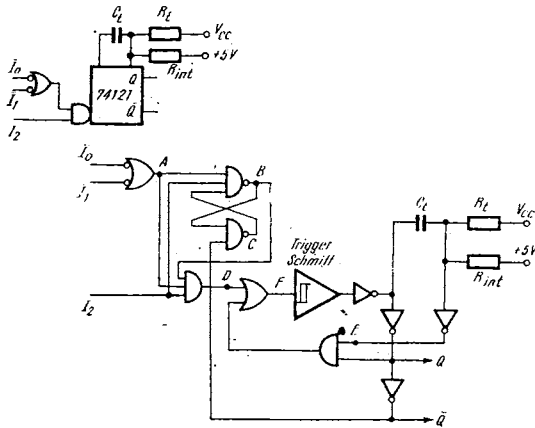


Fig. 3.11.

Rezolvare

a) Vezi figura 3.12.

$I_2$	A	B	C	D	E	F	Q	$\bar{Q}$
L	L	H	L	L	L	L	L	H
H	L	H	L	L	L	L	L	H
L	H	H	L	L	L	L	L	H
H	H	H	L	L	L	L	L	H
H	H	H	L	L	L	L	L	H
H	H	H	L	L	L	L	L	H
H	H	H	L	L	L	L	L	H
H	H	H	L	L	L	L	L	H
H	H	H	L	L	L	L	L	H
H	H	L	L	L	L	L	L	H
H	H	L	L	L	L	L	L	H
H	H	L	L	L	L	L	L	H
H	H	L	L	L	L	L	L	H
L	H	L	L	L	L	L	L	H
L	H	L	L	L	L	L	L	H
L	H	L	L	L	L	L	L	H
L	H	L	L	L	L	L	L	H
L	H	L	L	L	L	L	L	H
L	H	L	L	L	L	L	L	H
L	H	L	L	L	L	L	L	H
L	H	L	L	L	L	L	L	H
L	H	L	L	L	L	L	L	H
L	H	L	L	L	L	L	L	H

- 1 } monostabilul este în starea stabilă pregătit pentru
- 2 } operare
- 3 }

- $I_2$  comută în H (front pozitiv)
- comută și D cu o întârziere dată de  $t_p$

-- circuitul se blochează pentru a nu fi influențat de intrări pe durata stării cvazistabile

Încărcarea condensatorului  $C_1$  prin  $R_1$  (durata monostabilului)

-- E devine L. ( $C_1$  s-a încărcat la  $U_{p,ag}$ )

-- sfârșitul ciclului  
-  $I_2$  trece în L

- gata pentru a accepta o nouă comutare

Fig. 3.12.

b) Comutarea se produce cu frontul negativ pe intrările  $\bar{I}_0$  și  $\bar{I}_1$  și cu frontul pozitiv pe  $I_2$ . Monostabilul comută pentru

$$(I_0 + I_1) \cdot I_2 \cdot \bar{Q}$$

c) Alegând  $C_t = 3,3 \text{ nF}$  rezultă  $R_t = 21,7 \text{ K}\Omega$ .

d) Monostabilul nu ține cont de comenzile primite pe intrare în timpul stării cvazistabile și, datorită blocării porții D, acestea nici nu îi perturbă funcționarea. De aceea 74121 este numit monostabil nere-comutabil (nonretriggerable).

**Problema 3.14.** Desenați forma de undă la ieșirea unui monostabil 74121 cu  $T = 20 \text{ sec.}$  care primește la intrarea  $I_2$  semnalul din figura 3.13.

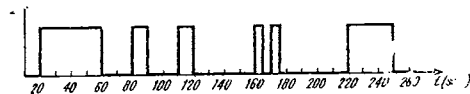


Fig. 3.13.

Cum se modifică răspunsul dacă acest semnal este aplicat pe intrarea  $\bar{I}_0$  ?

**Problema 3.15.** Să se realizeze un circuit basculant astabil cuplînd în reacție două monostabile 74121.

**Problema 3.16.** Să se realizeze cu monostabile 74121 o schemă de generator de impulsuri cu frecvență și durată variabilă.

**Problema 3.17.** Să se proiecteze cu 74121 un oscilator comandat în tensiune (variații între 2–4 V) avînd frecvența centrală de 6 MHz și deviația de frecvență  $f = 0,25 \text{ MHz}$ .

*Indicație*

Se utilizează schema din figura 3.14.

Tensiunea pe  $C_T$  variază

$$U_c = U_\infty + (U_0 - U_\infty)e^{-\frac{t}{\tau}}$$

și  $U_0 = -4,2 \text{ V}$ ,  $U_\infty = 0,8 \text{ V}$  unde  $\tau = R_L C_E$

Dacă apare și un curent constant  $I$

$$U_c = U_\infty + (U_0 - U_\infty)e^{-\frac{t}{\tau}} - \frac{I}{C_E} t$$

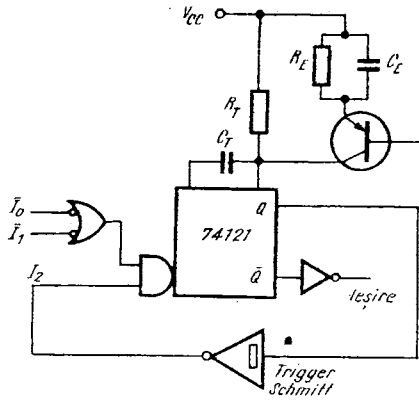


Fig. 3.14.

$$\text{De aici } I = \frac{C_E}{t} \left( 4,2 - 8,4 e^{-\frac{t}{\tau}} \right)$$

Se alege  $C_E = 20 \text{ pF}$ .

Duratele corespunzătoare intervalului de frecvență sînt 120 ns, 133 ns. Rezultă :

$$I = 160 \mu\text{A}$$

$$\Rightarrow R_E = 10 \text{ K}\Omega$$

$C_E$  are rolul de a elimina influența paraziților și se alege în funcție de frecvența semnalului de comandă.

**Problema 3.18.** Monostabilul recomutabil (retriggerable) 74122 (sau seria 9600 Fairchild) are proprietatea de a reiniția ciclul evasistabil la fiecare impuls aplicat la intrare, chiar în timpul cît se găsește comutat. Funcționarea acestuia este descrisă de diagramele din figura 3.15.

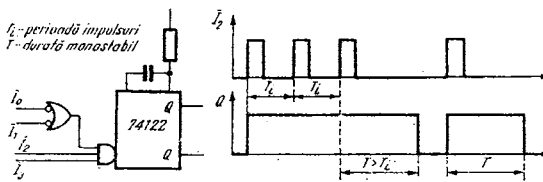


Fig. 3.15.

Să se realizeze cu 74122 un discriminator de impulsuri de frecvență mai mare decât 10 KHz.

**Problema 3.19.** Transformați un monostabil recomutabil într-unul nerecomutabil.

*Soluție*

Vezi figura 3.16.

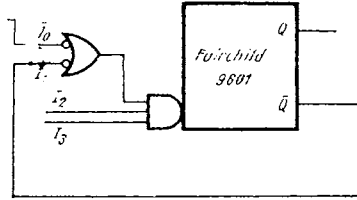


Fig. 3.16.

**Problema 3.20.** Explicați funcționarea circuitului de modificare a factorului de umplere a impulsurilor prezentat în figura 3.17. Considerați că  $C_t R_t$  produc impulsuri cu factor de umplere de  $f = 66\%$  relativ la ceasul de comandă.

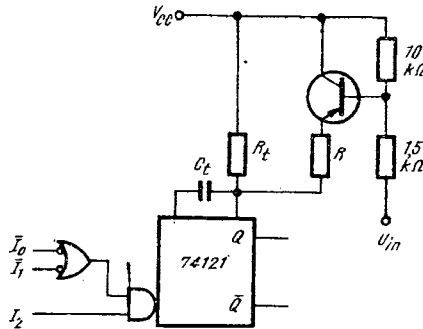


Fig. 3.17.

*Rezolvare*

Pentru 1 la intrare, tranzistorul e blocat,  $f = 66\%$ . Pentru 0 la intrare, tranzistorul se deschide,  $T = 0,69 C_t \frac{R_t R}{R_t + R}$  și pentru  $R_t = R$ ,  $f$  scade la 33%. Tranzistorul este pnp cu  $\beta$  mare și funcționează ca un comutator analog în regiunea de saturație inversă.

**Problema 3.21.** Proiectați o schemă care să formeze din impulsurile de comandă impulsuri întârziate cu  $20 \mu\text{s}$  față de frontul negativ și având durată de  $10 \text{ ms}$ .

**Problema 3.22.** Să se realizeze cu registre de bistabili (latch-uri) o demultiplexare secvențială, transmițându-se datele de pe 2 căi pe  $8 \times 2$  căi selectabile prin adresare.

*Soluție*

Vezi figura 3.18.

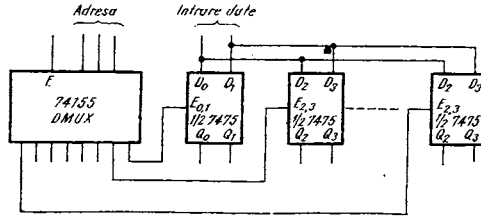


Fig. 3.18.

Se utilizează  $1 \times 74155$  și  $4 \times 7475$ .

**Problema 3.23.** Să se realizeze cu latch-uri 7475 o schemă de demultiplexare secvențială de la 4 căi la  $16 \times 4$  căi adresabile.

**Problema 3.24.** Să se realizeze cu registre latch 7475, 4 bistabili MS.

**Problema 3.25.** a) Să se deseneze formele de undă la ieșirile circuitului din figura 3.19 pentru 9 impulsuri de tact.

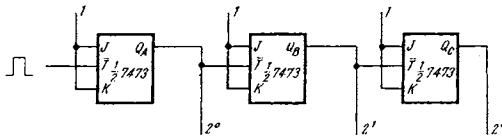


Fig. 3.19.

b) Asociind un număr zecimal fiecărei configurații binare de ieșire să se construiască graful de tranziții al circuitului.

*Soluție b)*

$$0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 7$$

↑ \_\_\_\_\_ |

Circuitul este un numărător pînă la 7.



**Problema 3.26.** Cum se poate modifica schema anterioară astfel încât graful de tranziții să fie parcurs în sens invers (numărător înapoi) ?

**Problema 3.27.** a) Determinați succesiunea stărilor și ciclul de numărare pentru numărătorul din figura 3.20.

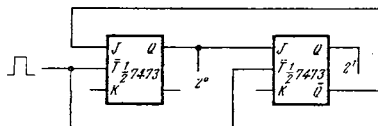


Fig. 3.20.

b) De ce acest numărător  $N$ , este sincron ?

c) Transformați numărătorul asincron din problema 3.25 într-un numărător cu aceeași lungime de ciclu cu  $N$ .

**Problema 3.28.** Determinați graful de tranziții pentru circuitul din figura 3.21.

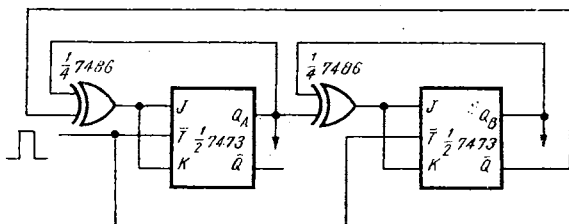


Fig. 3.21.

*Soluție* — Numărător sincron în cod Gray 00—01—11—10.

**Problema 3.29.** Proiectați un sistem de numărare, memorare și afișare cu 4 digiți.

*Soluție*

Schema este dată în figura 3.22. Numărătoarele 74192 conectate în cascadă numără în binar-zecimal impulsurile de tact.

La apariția semnalului de ceas  $E$  al registrelor 7475, acestea memorează datele din numărător și le afișează pe LED-uri cu ajutorul decodificatoarelor BCD/7 segmente.

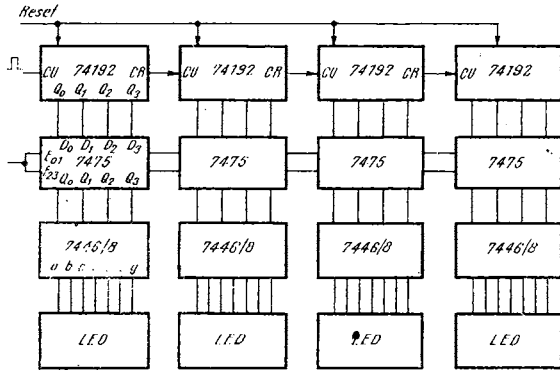


Fig. 3.22.

**Problema 3.30.** Realizați sistemul anterior cu afișaj multiplexat.  
*Rezolvare*

Schema este dată în figura 3.23. Cu ajutorul MUX se aduc pe rând cifrele la intrările 7446/8 și aprinderea LED-ului corespunzător fiecărui rang zecimal se face prin deschiderea cu DCD-ul a tranzistorului (npn) cheie respectiv. Dacă LED-urile sînt cu anod comun se utilizează 7448 și colectorii tranzistoarelor se conectează la 5 V. Pentru LED-uri cu catod comun amănuntele de schemă sînt notate în paranteze.

Fiecare LED va fi aprins un timp  $T_A$ , dintr-o perioadă de  $4 T_A$ . Deci pentru a lumina cu aceeași intensitate ca în schema 3.22, curentul trebuie să fie de 4 ori mai mare.

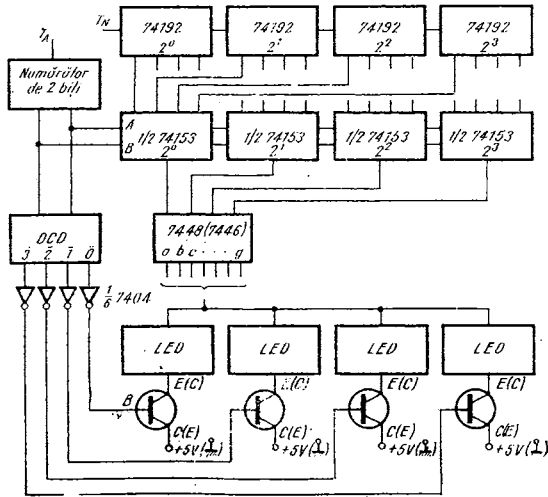


Fig. 3.23.

Pentru ca afișajul să nu pilpiie trebuie ca frecvența de aprindere a unei cifre să fie minimum 50 Hz, deci  $\frac{1}{4T_A} = 50 \text{ Hz}$ .

În general pentru  $n$  digiți, curentul prin fiecare LED trebuie să fie de  $n$  ori mai mare și frecvența impulsurilor de aprindere :

$$f_A = \frac{1}{T_A} \geq n \times 50 \text{ Hz}$$

Numărul  $n$  al cifrelor afișajului multiplexat nu poate fi prea mare deoarece se depășește curentul maxim admisibil al LED-urilor.

Pentru funcționarea corectă a schemei,  $f_A$  trebuie aleasă mult mai mare decât frecvența de numărare ( $1/T_N$ ).

*Observație.* Sistemul devine mai economic față de cel din problema 3.29 pentru mai mult de 4 digiți deoarece se reduce substanțial numărul de circuite integrate.

**Problema 3.31.** Să se realizeze cu numărătoare o schemă de conversie BCD-Binar pentru 5 digiți.

*Indicație*

Codul de convertit e încărcat într-un numărător binar zecimal,  $N_1$ . Acesta este apoi comandat să numere invers simultan cu numărătorul binar  $N_2$  care numără în sens direct. Când  $N_1$  ajunge pe 0 se crește numărarea și în  $N_2$  se găsește codul binar corespunzător.

**Problema 3.32.** Particularizați modelul Mealy pentru SLS (vezi problema 1.2) în cazul numărătorului integrat 74193.

*Rezolvare*

Modelul Mealy nu este valabil în cazul circuitului 74193 deoarece acesta nu este sincron decât în regim de numărare. Încărcarea are loc asincron pe palierul semnalului LD. Numărătorul 74193 poate fi reprezentat ca în figura 3.24. Se observă că vectorul de ieșire  $Y$  îl include

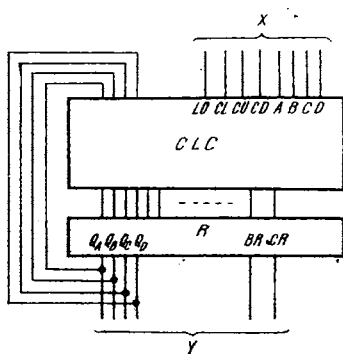


Fig. 3.24.

și pe cel de stare,  $Q$ . Ieșirile  $BR$  (BoRrow — împrumut) și  $CR$  (CaRry — transport) indică trecerea prin 0 la numărarea înapoi, respectiv prin 15 ( $Q_D Q_C Q_B Q_A = 1111$ ), înainte.

Circuitul are 8 intrări :

- $LD$  (LoaD) este intrarea de încărcare : cînd este în 0 permite memorarea în registru a numărului DCBA.

- $CL$  (CLear) este intrarea de ștergere : cînd este în 1 determină trecerea registrului în starea 0 ( $Q_D, Q_C, Q_B, Q_A = 0, BR = 0$  doar dacă  $CD = 0$  și  $LD = 1$ ).

- $CU$  (Count Up) comandă, pe frontul pozitiv al semnalului, numărarea înainte pe  $Q_A - Q_D$ .

- $CD$  (Count Down) comandă, pe frontul negativ al semnalului, numărarea înapoi.

- $D, C, B, A$  sînt intrările pe care apar datele ce trebuie înscrise în numărător.

**Problema 3.33.** a) Să se conecteze un numărător 74193 pentru a număra înainte de la 0 la 15.

b) Cum se face inițializarea funcționării numărătorului ?

c) Pe ce front comută numărătorul ?

d) Să se deseneze formele de undă pentru impulsurile de tact 15—17.

*Rezolvare*

a) Se conectează generatorul de impulsuri la intrarea  $CU$ .

b) Se aduce numărătorul în starea 0 ( $Q_D Q_C Q_B Q_A = 0000$ ) cu un impuls pe  $CL$ .

c) Numărătorul comută cu frontul pozitiv al impulsurilor de pe  $CU$ .

d) Formele de undă pentru impulsurile 15—17 se dau în figura 3.25. Pe frontul pozitiv al impulsului 15, numărătorul trece în starea 1111(15), iar pe cel al impulsului 16, revine în starea 0000, fiind pregătit pentru a relua numărarea.

Semnalul  $CR$  apare (activ pe 0) după terminarea impulsului 15.

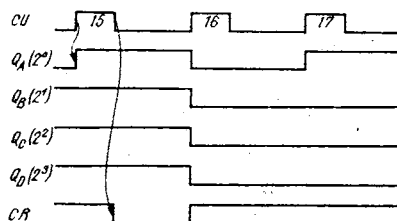


Fig. 3.25.

**Problema 3.34.** a) Să se conecteze un numărător 74193 pentru a număra înapoi de la 14 la 0.

b) Să se deseneze formele de undă pentru impulsurile 14–16. (Primul impuls este impulsul 0).

c) Cum se face inițializarea funcționării numărătorului ?

*Rezolvare*

a) Schema este dată în figura 3.26. Impulsurile de numărare se aplică pe *CD*, *CL* fiind conectat la 0 logic.

b) Formele de undă se dau în figura 3.27. La apariția frontului pozitiv al impulsului 14, numărătorul trece în 0 și, cu frontul negativ,

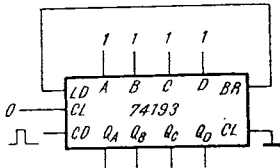


Fig. 3.26.

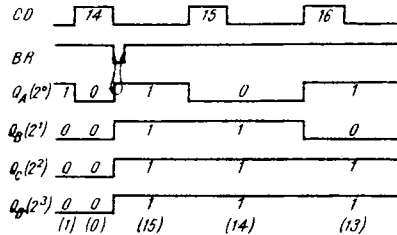


Fig. 3.27.

*BR* trece în 0, comandînd (cu o anumită întîrziere) înscrierea în paralel a 1111(15). Apoi *BR* dispare (devine 1) datorită încărcării cu un număr diferit de 0. Următorul front pozitiv realizează comutarea din 1111(15) în 1110(14) ș.a.m.d. Deci, circuitul are un ciclu repetitiv de numărare a 15 fronturi pozitive. Starea 15 apare parazit pe frontul negativ al celui de-al 14 impuls de tact.

c) Inițializarea este mai complicată ca la numărarea înainte, deoarece 74193 trebuie adus în starea 1111. Formele de undă se dau în figura 3.28. Impulsul pe *CL* ( $CL = 1$ ) aduce numărătorul pe 0. Un

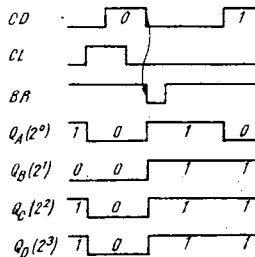


Fig. 3.28.

eventual front pozitiv (al impulsului 0) sosit în acest timp pe  $CD$ , nu este luat în considerare, comanda  $CL$  fiind prioritară față de  $CD$  (vezi în schema bloc a 74193 din anexă :  $CL$  se transmite pe intrarea  $R$ , iar  $CD$  pe tactul bistabililor). La terminare, impulsului pe  $CD$ ,  $BR$  trece pe 0, comandând înscrierea lui 1111(15) și apoi revenind în 1.

**Problema 3.35.** Să se explice cu ajutorul formelor de undă de ce cu schema numărătorului de la 14 la 0 din figura 3.29 se elimină starea parazită 15 care apărea la numărătorul din problema precedentă.

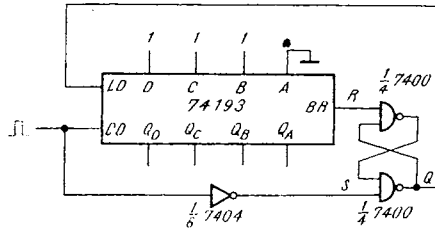


Fig. 3.29.

### Rezolvare

Formele de undă se dau în figura 3.30. Bistabilul  $RS$ , adăugat în schemă (față de figura 3.26) împiedică revenirea imediată a lui  $LD$  în 1. Deci circuitul se încarcă cu 14 când apare  $BR$ , dar nu mai numără pe următorul front pozitiv, deoarece este ținut la 0 pe  $LD$  (semnalul se transmite pe  $LD$  întârziat prin 2 porți față de  $CD$ ). Se observă că, starea 14 se menține la ieșiri o durată mai mare decât celelalte stări : de la sfârșitul impulsului 13 pînă la începutul lui 15. Deci nu apare sincron cu frontul pozitiv al tactului.

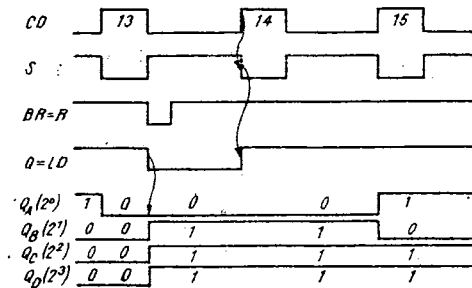


Fig. 3.30.

**Problema 3.36.** Să se proiecteze cu 74193 un numărător înapoi de la 15 la 0.

*Indicație*

Schema din figura 3.29 cu deosebirea că numărătorul se încarcă în paralel cu 1111(15).

**Problema 3.37.** Să se proiecteze cu 7193 un numărător înapoi de la 14 la 0 care își modifică toate stările cu frontul pozitiv al ceasului.

*Rezolvare*

Trebuie ca impulsul de încărcare  $LD$  să apară determinat de frontul ceasului (vezi figura 3.31).  $BR$  trece în 0 pe frontul negativ al im-

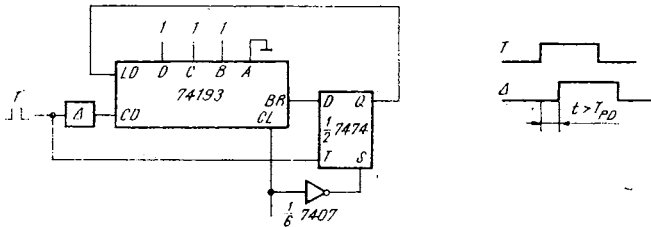


Fig. 3.31.

pulsului  $\Delta$ , dar nu se transmite la  $LD$  decât pe următorul front pozitiv. Circuitul  $\Delta$  întârzie transmiterea impulsului de tact la  $CD$ , cu un timp mai mare decât cel de propagare prin bistabilul  $D$ . Aceasta deoarece, pe frontul pozitiv al impulsului 14 trebuie să apară întâi semnalul  $LD$  și apoi  $CD$  care nu va fi luat în considerație. Formele de undă pentru impulsurile 13—15 sînt prezentate în figura 3.32.

Inițializarea schemei se face prin ștergerea numărătorului și setarea ( $S = 0$ ) bistabilului. Formele de undă sînt date în figura 3.33. Aducerea în starea inițială se face pe primul front pozitiv al tactului.

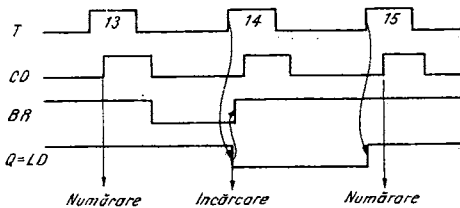


Fig. 3.32.

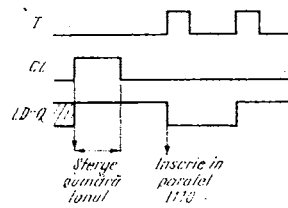


Fig. 3.33.

**Problema 3.38.** Se dă schema din figura 3.34. Dacă se comandă ștergerea ( $CL = 1$ ),  $BR$  va trece în 0, comandînd în același timp ( $LD = 0$ ) înserierea 1-urilor. Bistabilii interni din numărător (vezi schema bloc din anexă) trec în starea nedeterminată deoarece primesc comenzi atît pe  $R$  cît și pe  $S$ .

Este posibilă o schemă ca în figura 3.34 ?

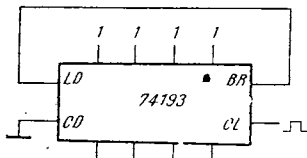


Fig. 3.34.

*Rezolvare*

Pentru  $CL = 1$ , bistabilii se șterg ( $Q = 0, \bar{Q} = 1$ ). În starea de nedeterminare,  $R = S = 0$  ieșirile vor fi  $Q = 1, \bar{Q} = 1$ . Dar  $BR$  care adună semnalele din  $\bar{Q}_A, \bar{Q}_B, \bar{Q}_C, \bar{Q}_D = 1$  va rămîne neschimbat, deci 0. La dispariția impulsului pe  $CL$  ( $CL = 0$ ),  $BR$  fiind în 0, se face încărcarea cu 1111 în numărător. Deci, schema funcționează corect, deși bistabilii interni trec prin starea nedeterminată.

**Problema 3.39.** Să se analizeze funcționarea schemei din figura 3.35.

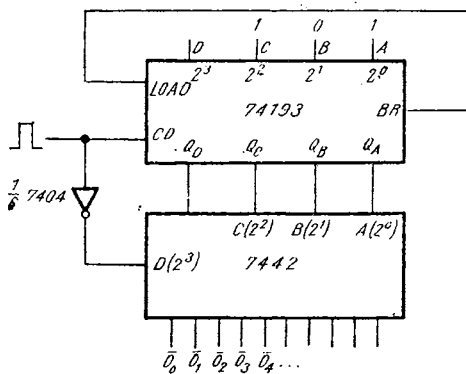


Fig. 3.35.



### Rezolvare

Este un ceas cu 5 faze, așa cum rezultă din formele de undă din figura 3.36. DCD 7442 este folosit ca DMUX fiind comandat de impulsurile de ceas pe intrarea  $D$ , pentru a-și activa ieșirile doar pe durata impulsurilor.

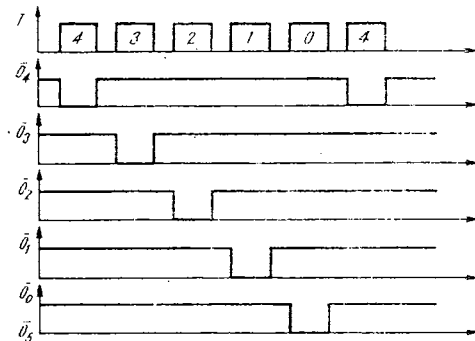


Fig. 3.36.

Numărătorul 74193 este încărcat cu numărul 5 și numără înapoi ( $CD$  — Count Down) pe frontul pozitiv între 4 și 0 (vezi problema 3.34).

**Problema 3.40.** Să se realizeze un ceas cu 5 faze la care impulsurile pe fiecare fază au duratele delimitate de fronturile pozitive ale impulsurilor de tact.

#### Indicație

Se transformă 74193 cu schema din figura 3.31 și DCD nu mai este condiționat pe  $D$  de impulsurile de ceas.

**Problema 3.41.** a) Să se proiecteze un numărător înapoi, între 250 și 0 cu  $2 \times 74193$  conectate în cascadă.

b) Care din schemele, cu reacție directă  $BR-LD$  (problema 3.34) sau cu reacție  $BR$  — bistabil  $RS-LD$ , este indicată pentru această conectare ?

c) Care este dezavantajul conectării în cascadă a numărătoarelor ?

#### Rezolvare

a) Schema cu reacție directă  $BR-LD$  este dată în figura 3.37 în care numărătoarele se încarcă cu 251(11111011).  $N_1$  este numărătorul cel mai puțin semnificativ, la fiecare apariție a  $BR_1$ , numărînd  $N_2$ .

b) Schema cu reacție directă nu este sigură în funcționare: cele două numărătoare nu au timpi de încărcare egali și, dacă  $N_2$  se încarcă

primul, dispare  $BR_2$ , iar  $N_1$  nu se mai poate încărca. Deci o astfel de schemă ar funcționa corect doar cu circuite selectate.

c) Numărătorul obținut prin conectarea în cascadă  $N_1-N_2$  nu este sincron.  $N_2$  numără cu tactul întârziat cu timpul de propagare de la  $CD$  la  $BR$ . Deci, un  $DCD$  conectat pe ieșiri ar da semnale parazite.

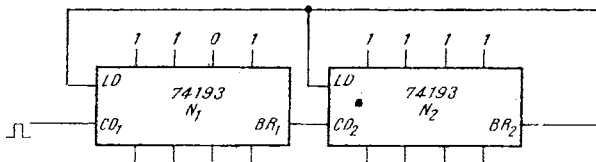


Fig. 3.37.

**Problema 3.42.** Să se reprojeteze numărătorul din problema anterioară astfel încît  $N_1$  și  $N_2$  să comute sincron.

*Indicație*

În figura 3.38,  $N_1$  numără la fiecare front pozitiv al tactului. În acest timp,  $BR_1$  fiind pe 1, comandă încărcarea lui  $N_2$  chiar cu numărul pe care îl conține, împiedicînd astfel impulsurile de pe  $CD_2$  să îl

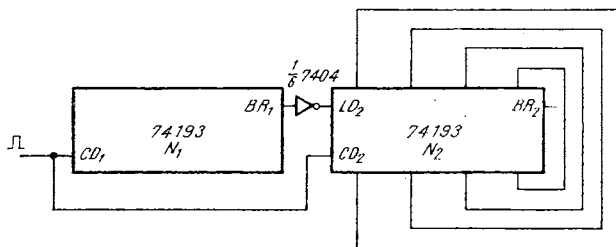


Fig. 3.38.

influențeze pe  $N_2$ . Cînd  $N_1$  ajunge pe 0,  $BR_1$  trece în 0 și este oprită încărcarea ( $LD_2 = 1$ ). Astfel că următorul impuls de ceas care sosește produce numărarea simultană, atât a lui  $N_1$ , cît și a lui  $N_2$ .

Pentru rezolvarea problemei schema trebuie completată cu reacția  $BR_2 -$  bistabil  $RS-LD_2$ ,  $LD_1$  necesară înscrierii numărului 250 cînd se ajunge în 0 (problema 3.41) și cu circuitele de inițializare.

**Problema 3.43.** Cu ajutorul numărătorului 74193 să se realizeze un numărător decadic :

- a) înainte ;
- b) înapoi.

*Soluție*

Doă scheme posibile sînt prezentate în figura 3.39. Încărcarea numărătoarelor se face ținînd cont de faptul că, la acest tip de conexiune ( $BR = LD$ , respectiv  $CR = LD$ ) se pierde o stare în timpul numărării (vezi problema 3.34).

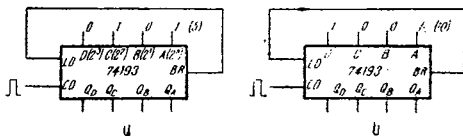


Fig. 3.39.

**Problema 3.44.** Să se realizeze cu 74193 un divizor de frecvență programabil, între 0 și 30. Programarea se face în cod binar cu ajutorul unor comutatoare bipolare.

*Soluție*

Se utilizează 4 circuite integrate (figura 3.40). În programare se ține cont de faptul că schema cu reacție directă  $CR-LD$  pierde o stare

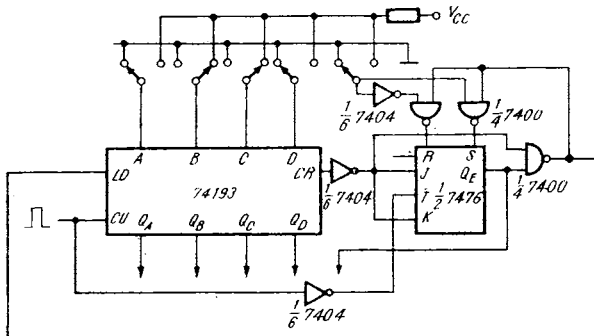


Fig. 3.40.

la numărare. Deci, dacă dorim o divizare cu  $N$ , numărul care se programează cu ajutorul comutatoarelor este  $N + 1$ .

**Problema 3.45.** Să se realizeze un numărător modulo 64:

a) cu  $2 \times 74193$

b) cu  $1 \times 74193$  și 2 bistabili  $JK$ .

**Problema 3.46.** Folosind un bistabil  $D$  cu  $S$  (Set) și  $R$  (Reset) 7474 și porți să se extindă 74193 la un numărător binar modulo 32 cu încărcare paralelă și numărare înainte ( $U_p$ ) și înapoi ( $Down$ ).

*Rezolvare*

Se adaugă o celulă de numărare modulo 2 realizată în condițiile impuse de funcționarea numărătorului 74193, ca în figura 3.41.  $CR$  și  $BR$  sînt ieșirile circuitului 74193.

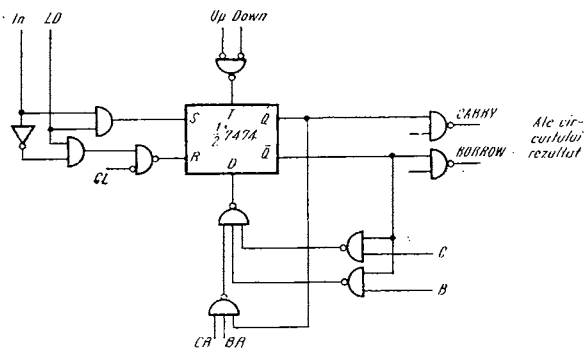


Fig. 3.41.

**Problema 3.47.** Care este capacitatea maximă de numărare pentru un circuit format din  $3 \times 74193$  conectate în cascadă?

a) cu reacția directă de la  $BR$  la  $LD$ ;

b) cu reacție de încărcare printr-un bistabil  $RS$ ;

c) cu reacție de încărcare prin bistabil  $D$ .

*Soluții*

a) 4 095; b) 4 096; c) 4 096 impulsuri.

**Problema 3.48.** Să se determine graful de tranziții și forma de unde la ieșire pentru circuitele din figurile 3.42 și 3.43 știind că numărătoarele sînt complet sincrone numărarea și încărcarea,  $LD$ , se face pe frontul pozitiv al ceasului — vezi problema 3.37 și figura 3.31.

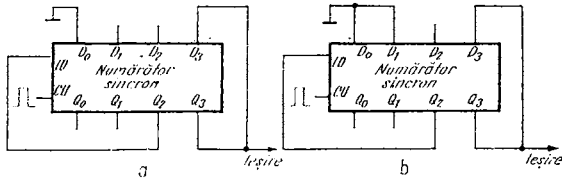


Fig. 3.42.

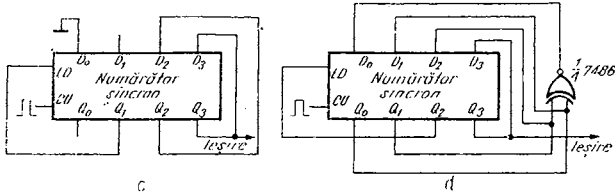


Fig. 3.43.

Soluție

a) Graful circuitului este prezentat în figura 3.44.

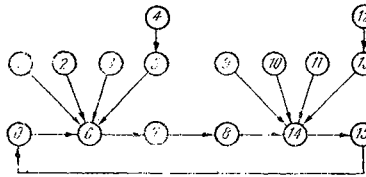


Fig. 3.44.

Ciclul repetitiv conține 6 stări. Din stările care nu aparțin acestuia se intră în ciclul după maximum 2 impulsuri de tact. Numerele zecimale sînt asociate stărilor după corespondențele :

$$Q_0 - 2^0, Q_1 - 2^1, Q_2 - 2^2, Q_3 - 2^3$$

Din formele de undă (figura 3.45) rezultă că circuitul este un divizor cu 6 avînd factorul de umplere al semnalului de ieșire :

$$f = \frac{\text{durata impulsului}}{\text{perioada impulsului}} = 50\%$$

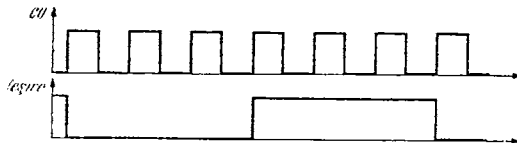
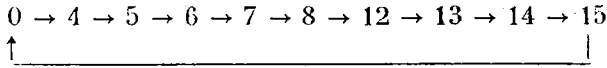


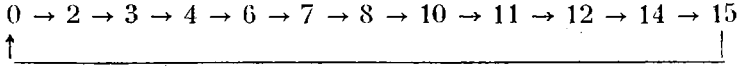
Fig. 3.45.

b) Ciclul repetitiv al circuitului este:



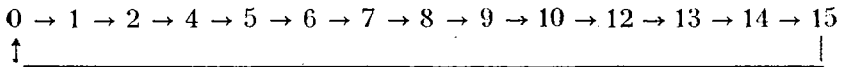
Deci este un divizor cu 10, cu factor de umplere al formei de undă de la ieșire,  $f = 50\%$ .

c) Ciclul repetitiv este:



Numărător modulo 12 cu  $f = 50\%$ .

d) Ciclul repetitiv este:



Divizor cu 14 cu  $f = 50\%$ .

**Problema 3.49.** Circuitul din figura 3.46, trebuie să realizeze forma de undă din figura 3.47,  $x$  și  $y$  fiind două numere oarecare.

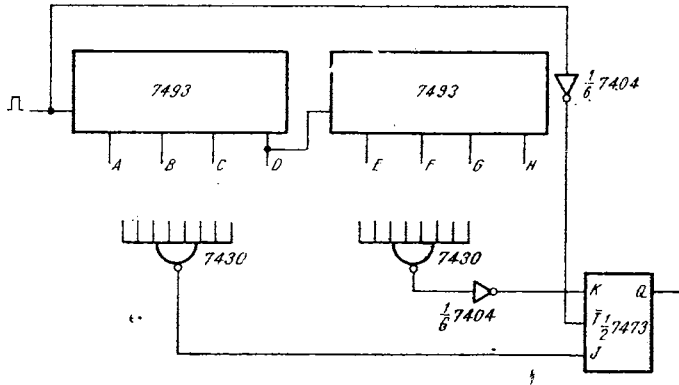


Fig. 3.46.



Fig. 3.47.

Definiți conexiunile porților pentru obținerea unei forme de undă la alegere.

*Indicație*

Pentru evitarea comutărilor parazite datorate decodificării nestrobate se recomandă decodificarea stării  $x-1$ ,  $y-1$  și folosirea aceleiași impuls de ceas.

**Problema 3.50.** Să se realizeze un circuit care să rezolve coincidența a două sau mai multe comutatoare, furnizând în cazul unui asemenea mod de operare și codul unui comutator acționat. Presupunem o tastatură cu 16 butoane.

Se vor utiliza  $2 \times 74193$  și 1 MUX 74150.

*Soluție*

vezi figura 3.48.

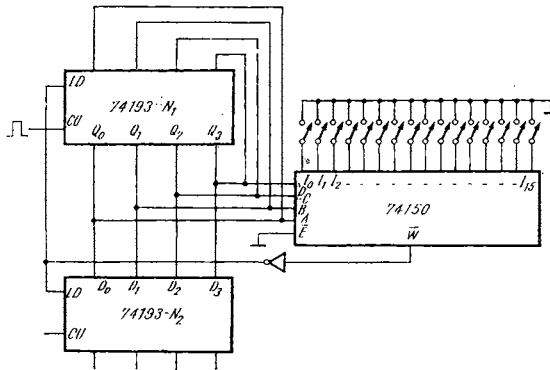


Fig. 3.48.

$N_1$  numără continuu controlînd MUX. Cînd se apasă pe o tastă, codul respectiv se înscrie în  $N_2$ , cuplat numai pe încărcare paralel și  $N_1$  se blochează în starea respectivă pînă cînd tasta revine în poziția inițială. În cazul în care sînt activate mai multe clape simultan numai cea care va coincide prima cu codul numărătorului va fi luată în considerare.

**Problema 3.51.** Să se proiecteze cu numărătoare și decodificatoare un sincrogenerator TV urmînd a livra la ieșire semnalele din figura 3.49.

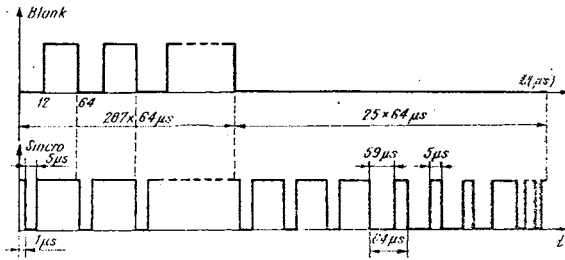


Fig. 3.49.

### Indicație

Se utilizează un numărător divizor prin 64 urmat de altul modulo 312 (reprezentat în figura 3.50 de cele 3 decade din dreapta).

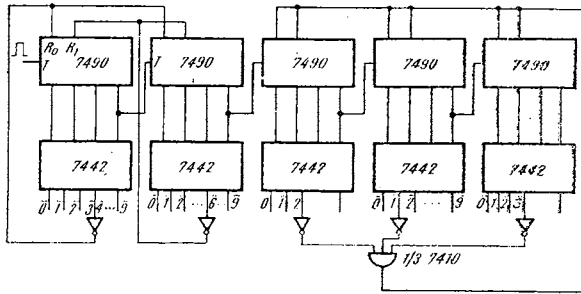


Fig. 3.50.

Folosind apoi ieșirile decodificărilor se vor acționa  $R$  și  $S$  la bistabilul ce va da semnalul „Blank” și la cel ce va da semnalul „Sincro”. Numărul ce trebuie să acționeze asupra bistabililor se obține împărțind forma de undă în intervale egale cu cel minim,  $1/\mu s$ .

**Problema 3.52.** Să se explice funcționarea schemei de conversie digital/analogică din figura 3.51.

Să se compare convertoarele cu următoarele conexiuni între numărător și codificator :

- $\bar{I}_0 = 2^0$  ;  $\bar{I}_1 = 2^1$  ; ...  $\bar{I}_7 = 2^7$
- $\bar{I}_0 = 2^7$  ;  $\bar{I}_1 = 2^6$  ; ...  $\bar{I}_7 = 2^0$ .

### Soluție

Numărătorul are capacitatea de 256 impulsuri. Schema din figura 3.51 ilustrează situația  $a$ . Pe primele 128 impulsuri intrarea  $\bar{I}_i$  a codificatorului este pe 0, deci, acesta va furniza la ieșiri codul  $\bar{A}_0\bar{A}_1\bar{A}_2 = 000$ .



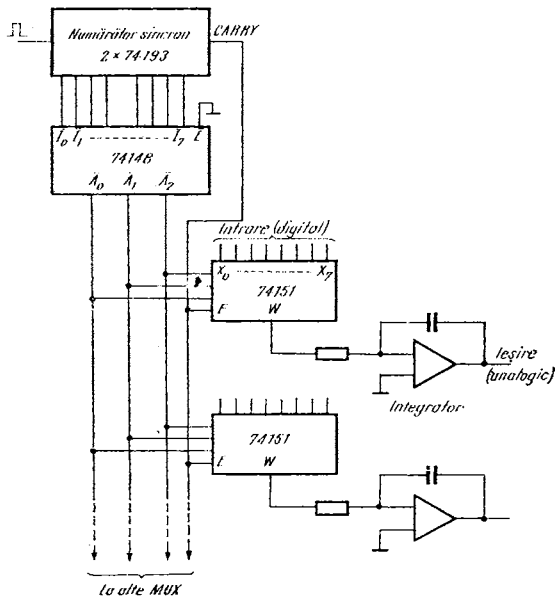


Fig. 3.51.

În acest timp, MUX-urile cu  $X_0 = 1$  vor da la ieșire semnal 1 (celelalte fiind pe 0). Deci la intrarea integratoarelor respective, forma de undă va avea un factor de umplere de 50%.

Pe următoarele 128 de impulsuri,  $I_7 = 1$  cedează prioritatea bitului  $I_6$  care va rămâne în 0 timp de 64 de impulsuri. Forma de undă la ieșirea MUX-urilor cu  $X_1 = 1$  va avea un factor de umplere de 25% ș.a.m.d.

Prin integrarea formelor de undă de la ieșirea multiplexoarelor având pe intrări doar câte un 1 și 7 de 0, se pot obține 8 trepte de tensiune, fiecare cu valoarea, jumătate din cea a treptei anterioare. Dacă cuvântul de comandă al intrărilor conține mai multe 1-uri, se obțin la ieșirea integratoarelor valori compuse cu ajutorul acestor 8 trepte. În total sînt 255 nivele de tensiune.

În cazul conexiunilor  $b$ , ieșirea de rang cel mai puțin semnificativ a număratorului are prioritatea cea mai mare. Astfel, semnalul la ieșirea MUX cu  $X_0 = 1$ , în loc să fie 1 timp de 128 impulsuri și 0 în rest, este alternativ în 1 și 0: evident tensiunea la ieșirea integratorului va fi aceeași. Și pentru celelalte configurații de intrare ale MUX-urilor se păstrează factorul de umplere de la conexiunea  $a$ ,

dar impulsurile apar strobate. Aceasta are avantajul că, față de *a*, reduce constanta de timp a integratoarelor. Dezavantajul fiind că, datorită numărului mare de comutări (256 față de 8 în cazul *a*) pe ciclul de conversie, schema este mai sensibilă la întârzieri, fronturi; scade linearitatea integratoarelor.

*La viteze mari de conversie, semnalele parazite la ieșirile MUX-urilor datorate timpilor diferiți de propagare introduc erori. Pentru a evita erorile, se cuplează bistabili D la ieșirile MUX-urilor care primesc sincron impulsurile.*

**Problema 3.53.** Explicați funcționarea convertorului analog/digital din figura 3.52.

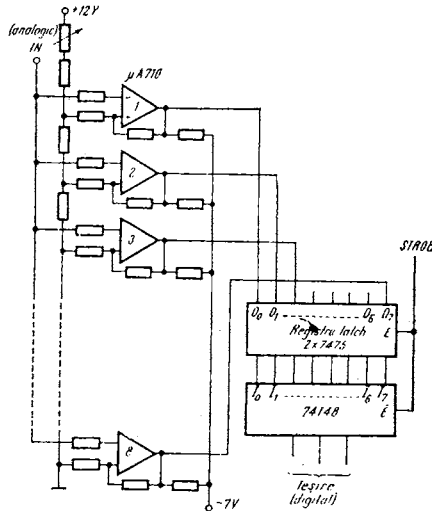


Fig. 3.52.

### Rezolvare

Tensiunea pentru conversie este aplicată simultan comparatoarelor, pe una din intrări și acestea o compară cu tensiunea de referință stabilită la cealaltă intrare cu ajutorul divizoarelor rezistive. Toate comparatoarele la care tensiunea de intrare este mai mare decât cea de referință dau 1 pe ieșire. Cuvântul rezultat din comparare este memorat și inversat în registrele „latch“. Apoi se transmite codicatorului care dă la ieșire codul negat al primului comparator cu 0 pe ieșire; acesta corespunde celei mai scăzute tensiuni de referință care se menține deasupra celei de intrare.

Registrul are rolul de a înlătura semnalele parazite la ieșirea codificatorului în timpul strobării.

Convertorul din figura 3.52 este foarte rapid datorită comandării simultane a comparatoarelor.

**Problema 3.54.** Să se proiecteze un convertor D/A de 16 biți.

*Indicație*

Trebuie extinsă schema din figura 3.51. Sînt necesare : un numărător de 16 biți, un codificator cu priorități de 16 biți, 16 MUX de 16 biți, 16 integratoare.

**Problema 3.55.** Să se proiecteze un convertor A/D de 4 biți.

*Indicație*

Schema convertorului A/D de 3 biți din figura 3.52 se extinde astfel: 16 comparatoare, un „latch“ de 16 biți și un codificator cu priorități de 16 biți.

**Problema 3.56.** Să se explice funcționarea schemei de înmulțire și împărțire secvențială utilizînd numărătoare, din figura 3.53.

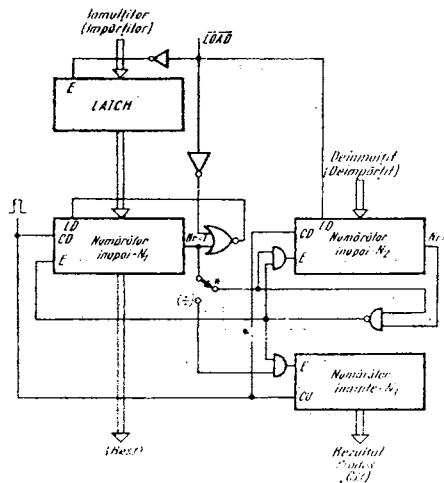


Fig. 3.53.

*Rezolvare*

Înmulțirea se face astfel :

- Se memorează înmulțitorul în „latch“.
- Cu semnalul de LOAD se încarcă înmulțitorul în  $N_1$  și deînmulțitorul în  $N_2$ .

- Se dau impulsuri de numărare pentru  $N_1$  și  $N_3$ .
- Când  $N_1$  ajunge pe 1 apare semnal  $Nr = 1$  care comandă reîncărcarea lui  $N_1$  și decrementarea lui  $N_2$ .
- Se repetă numărarea pînă cînd  $N_1$  și  $N_2$  ajung amîndouă pe 1. Circuitul ȘI NU dă semnal oprind numărătoarele. În  $N_3$  se găsește rezultatul înmulțirii.

Împărțirea se face în următoarele etape :

- Se memorează împărțitorul în „latch“.
- Cu semnalul de LOAD se încarcă împărțitorul în  $N_1$  și deîmpărțitul în  $N_2$ .
- Se dau impulsuri de numărare pentru  $N_1$  și  $N_2$ .
- Când  $N_1$  ajunge pe 1, apare semnal  $Nr = 1$  care comandă reîncărcarea împărțitorului și numărarea lui  $N_3$ .
- Se repetă numărarea pînă cînd  $N_2$  ajunge pe 1. Atunci se opresc numărătoarele :  $N_3$  va conține citul și  $N_1$ , restul împărțirii.

Se observă că, în cazul utilizării numărătoarelor 74193, acestora trebuie să li se adauge circuite de condiționare a numărării (intrarea  $E$ ) pentru a fi transformate conform cerințelor schemei din figura 3.53.

**Problema 3.57.** Să se realizeze utilizînd numărător, decodor și multiplexor un sistem automat de transmitere a unor comenzi și recepționare a semnalelor în cazul a 8 (opt) dispozitive identice sau diferite.

*Rezolvare*

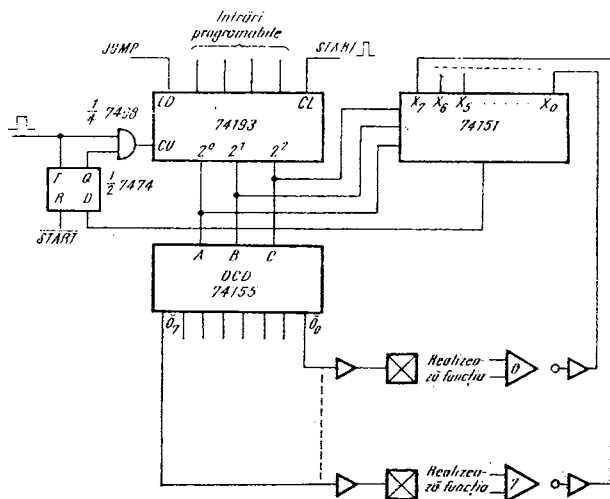


Fig. 3.54.

În figura 3.54 s-au făcut următoarele notații :

- ☒ dispozitive comandate (motoare, valve, semnalizatoare etc.);
- ▷ senzori (microswitch, fotocelulă, traductor de presiune ș.a.).

Cu START se resetează numărătorul comandându-se realizarea funcției 0. Când dispozitivul cuplat pe calea 0 termină de realizat funcția, MUX dă semnal permițând accesul unui impuls de ceas în numărător. Acesta își modifică poziția comandînd efectuarea funcției 1 ș.a.m.d. În modul de funcționare descris, dispozitivele cuplate sînt comandate în ordine de la 0 la 7.

Pe intrarea de LOAD se pot comanda și schimbări ale acestei ordini (JUMP-salt).

Se observă că 74193 a fost transformat cu un bistabil D și o poartă, în numărător cu condiționarea incrementării de către MUX.

**Problema 3.58.** Pentru schema din figura 3.55 starea inițială a numărătorului este 0000, iar cea a ieșirilor este ABCD = 1000.

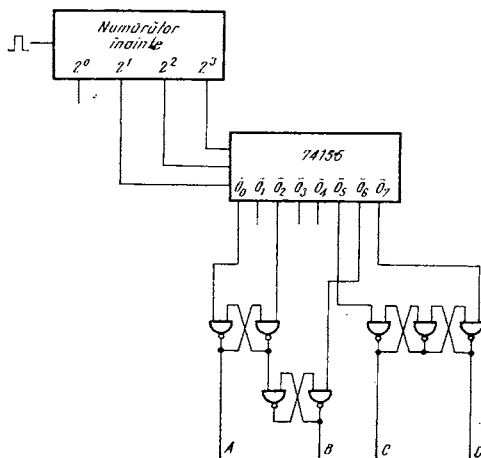


Fig. 3.55.

Se cere starea ABCD :

- a) după 6 impulsuri de tact ;
- b) după 10 impulsuri de tact ;
- c) după 13 impulsuri de tact ;
- d) după 14 impulsuri de tact.

*Soluție*

- a) 0000 ; b) 0011 ; c) 0111 ; d) 0111.

**Problema 3.59.** Să se explice funcționarea circuitului din figura 3.56 presupunând că numărarea se face înainte, de la 0 la 3.

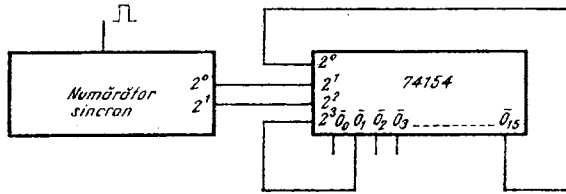
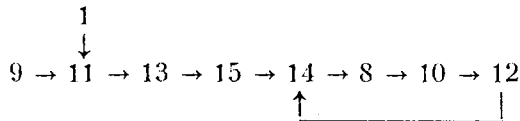


Fig. 3.56.

**Rezolvare**

Graful de tranziții la intrările DCD va fi



Trecerea din 15 în 14 se produce rapid datorită reacției  $\bar{O}_{15} - 2^0$ . Stările 0, 2, 4, 6 nu pot să apară deoarece  $\bar{O}_1$  și  $\bar{O}_{15}$  nu pot fi simultan 0. De asemenea, stările 3, 5, 7 nu sînt stabile deoarece  $O_1$  este diferit de 0 la decodificarea numerelor respective.

**Problema 3.60.** Cum se modifică starea registrului 7495 (figura 3.57), dacă inițial  $A = C = 1$ ;  $B = D = 0$ ;  $Q_A = Q_B = 1$ ;  $Q_C = Q_D = 0$ ;

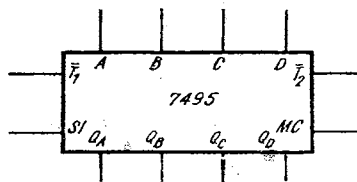


Fig. 3.57.

SI = 0 și apare din următoarele situații:

- a)  $\bar{T}_1 = 0$ ;  $\bar{T}_2 = 1$  iar MC trece din 1 în 0
- b)  $T_1 = 0$ ;  $\bar{T}_2 = 1$  „ 0 în 1
- c)  $T_1 = 1$ ;  $\bar{T}_2 = 0$  „ 1 în 0
- d)  $T_1 = 1$ ;  $\bar{T}_2 = 0$  „ 0 în 1

- e)  $\bar{T}_1 = 0$ ;  $\bar{T}_2 = 0$  iar  $MC$  trece din 1 în 0  
 f)  $\bar{T}_1 = 1$ ;  $\bar{T}_2 = 1$  „ 0 în 1  
 g)  $\bar{T}_1 = 1$ ;  $\bar{T}_2 = 1$  „ 1 în 0

### Rezolvare

Trebuie analizată schema internă a registrului (vezi anexa).  $\bar{T}_1$  este tactul de deplasare serie, iar  $\bar{T}_2$  este cel de înscriere în paralel. Problema evidențiază o subtilitate în funcționarea registrului 7495; pentru unele combinații logice pe intrările de tact, circuitul poate fi comutat cu ajutorul  $MC$  datorită apariției unui front negativ la ieșirea  $\overline{MC} \cdot \bar{T}_1 + MC \cdot \bar{T}_2$ .

a) Bistabilii comută datorită variației lui  $MC$  prin poarta predeschisă de  $\bar{T}_2$ , acționind pe modul paralel. Aceasta deoarece „masterul” din fiecare bistabil a fost încărcat de la intrările  $A, B, C, D$  ( $MC = 1$ ). Rezultă că starea de la intrări apare la ieșire. Deci  $Q_A = Q_C = 1$ ,  $Q_B = Q_D = 0$ .

b) Starea registrului nu se modifică deoarece bistabilii comută cu frontul negativ.

c) Nici o modificare.

d) Comută serie  $Q_A = 0$ ;  $Q_B = 1$ ;  $Q_C = 1$ ;  $Q_D = 0$ .

e) Nici o modificare.

f) Nici o modificare. Ar comuta serie dacă  $t_{PLH}$  pentru  $(MC \cdot \bar{T}_2)$  ar fi mai mare decât  $t_{PHL}$  (INVERTOR) +  $t_{PHL}(MC \cdot \bar{T}_1)$ , deoarece ar apare un impuls parazit pe ceasul bistabililor. Dar în realitate circuitul ȘI( $MC \cdot \bar{T}_2$ ) nu poate avea un timp de propagare așa de mare.

g) Poate încărca paralel dacă diferența  $t_{PHL}(\overline{MC} \cdot \bar{T}_2) - [t_{PLH}(\text{INV.}) + t_{PLH}(\overline{MC} \cdot \bar{T}_1)]$  este suficient de mare pentru a comuta bistabilii.

**Problema 3.61.** a) Care este starea unui registru 7495 la momentul  $t_1$ , dacă  $A = 1$ ,  $B = C = D = 0$ ,  $SI = 0$ , dacă se conectează  $\bar{T}_2$  cu  $MC$  și se aplică formele de undă din figura 3.58.

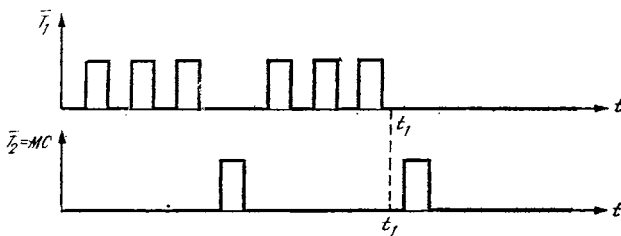


Fig. 3.58.

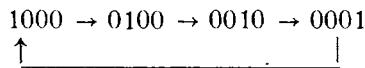
b) Care este starea registrului pentru care nu se amorsează funcționarea ?

*Rezolvare*

a) Pe primele 3 tacturi  $\bar{T}_1$ ,  $MC$  fiind 0, registrul deplasează către dreapta conținutul bistabililor. Pe frontul negativ al primului impuls  $\bar{T}_2 = MC$  se înregistrează în paralel în registrul  $Q_A Q_B Q_C Q_D = 1000$ . Pe următoarele 3 fronturi negative ale  $\bar{T}_1$  se deplasează 1000 către dreapta înscriindu-se 0 pe intrarea serie ( $\bar{SI}$ ), astfel :

- după primul tact 0100 ;
- după al doilea tact 0010 ;
- după al treilea tact (la momentul  $t_1$ ), 0001.

Circuitul este un numărător în inel cu graful



b) Funcționarea poate fi amorsată din orice stare : după maximum 3 impulsuri de tact se intră în ciclul repetitiv datorită încărcării paralele,

**Problema 3.62.** Se dă schema din figura 3.59.

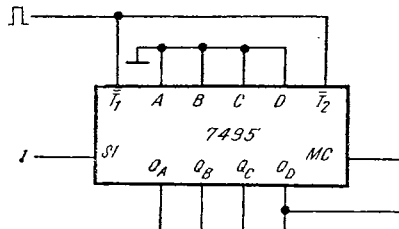


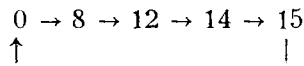
Fig. 3.59.

a) Să se deducă graful de succesiune al stărilor sistemului.

b) Care este starea registrului pentru care nu se amorsează funcționarea ?

*Rezolvare*

a) Pentru  $Q_D = 1$ , impulsul de tact încarcă paralel registrul. Deci rezultă următorul graf de tranziții, presupunând starea inițială 0.



Am considerat  $Q_A = 2^3$ ,  $Q_B = 2^2$ ,  $Q_C = 2^1$ ,  $Q_D = 2^0$



Trebuie baleiate toate cele 16 stări posibile și completat graful astfel încât să fie incluse toate aceste stări (figura 3.60).

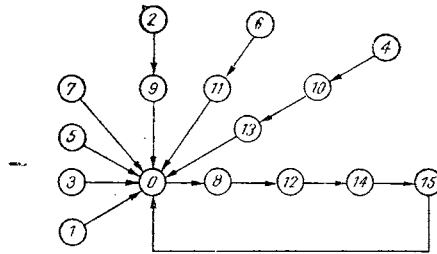


Fig. 3.60.

b) Se observă că toate stările produc amorsarea sistemului.

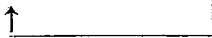
**Problema 3.63.** Schema din figura 3.59 se modifică astfel:  $D = 1$ ,  $Q_c = MC$ .

a) Să se determine graful ciclului repetitiv și numărul de tacturi al acestuia.

b) Există vreo stare care nu amorsează sistemul?

*Soluție*

a)  $1 \rightarrow 8 \rightarrow 12 \rightarrow 14$ , deci 4 tacturi



b) Construind graful complet se constată că toate stările amorsează sistemul.

**Problema 3.64.** Să se deducă graful de tranziții pentru schema din figura 3.61. Există vreo stare care nu amorsează circuitul?

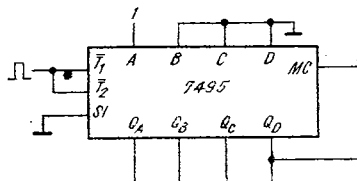


Fig. 3.61.

*Rezolvare*

Graful de tranziții este dat în figura 3.62. Se observă că este un numărător în inel. Starea 0000 nu apare în graf: este o stare de blocare a circuitului.

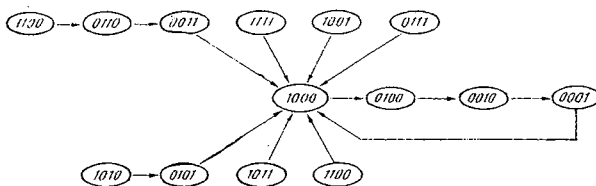


Fig. 3.62. •

**Problema 3.65.** Să se realizeze un numărător în inel cu 7495 la care amorsarea funcționării să se facă din toate stările ce nu aparțin ciclului respectiv.

*Rezolvare*

La schema din figura 3.61 se adaugă o poartă SAU NU (NOR) care, recunoscând configurația 0000 înscrie 1 prin SI (fig. 3.63). Pentru orice stare diferită de aceasta, SI este egal cu 0. Graful este cel din figura 3:62 la care se adaugă tranziția 0000 → 1000.

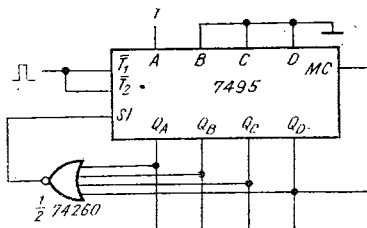


Fig. 3.63.

**Problema 3.66.** Să se implementeze un numărător în inel de 8 biți cu autoamorsare.

*Indicație*

Sînt necesare  $2 \times 7495$ :  $Q_D$  de la primul este conectat la SI de la al doilea, taturile și MC-urile fiind legate în paralel. Pentru evitarea blocării în starea 0 este necesar un SAU NU cu 8 intrări. Deoarece nu există un astfel de circuit integrat se implementează cu  $1 \times 7430$  și 9 inversoare  $\left(\frac{1}{6} 7404\right)$ . Numărul care se înscrie în paralel în registre este 10000000.

**Problema 3.67.** Să se studieze cu ajutorul grafului de tranziție comportarea schemei din figura 3.64.

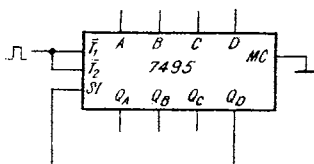


Fig. 3.64.

*Rezolvare*

Presupunând starea inițială 0001 vom determina următorul ciclu repetitiv :

0001 → 1000 → 0100 → 0010  
 ↑

Considerînd starea inițială 0011 obținem :

0011 → 1001 → 1100 → 0110  
 ↑

Pornind din starea 1110 rezultă :

1110 → 0111 → 1011 → 1101  
 ↑

Alt ciclu repetitiv va fi :

0101 → 1010  
 ↑

Observăm că circuitul poate urma unul din cele 4 cicluri repetitive, în funcție de starea inițială în care se vor afla bistabilii din interiorul registrului după aplicarea tensiunii de alimentare,  $V_{CC}$ .

0000 și 1111 sînt stări de blocare.

O astfel de schemă, avînd o comportare aleatoare nu este utilă decît dacă i se adaugă circuite pentru inițializarea registrului într-unul din ciclurile de funcționare.

**Problema 3.68.** Să se inițializeze registrul 7495 conectat ca în figura 3.64 astfel încît să funcționeze ca numărător în inel.

*Rezolvare*

Se conectează un MUX ca în figura 3.65 care, pentru toate configurațiile de intrare diferite de cele ale ciclului de numărător în inel

determină  $MC = 1$ , deci înscrierea în paralel a 1000. Se ajunge astfel la un graf identic cu al circuitului din figura 3.63.

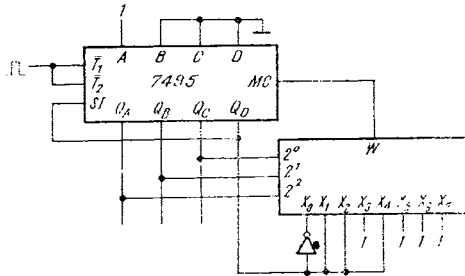


Fig. 3.65.

**Problema 3.69.** a) Să se determine grafurile de tranziție pentru circuitul din figura 3.66.

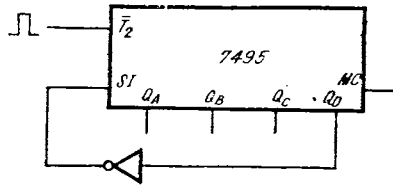
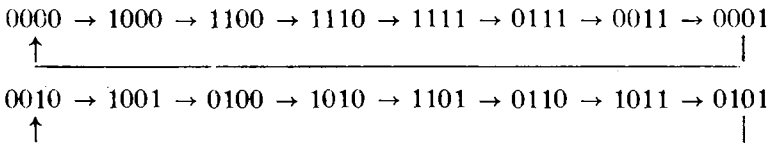


Fig. 3.66.

b) Să se transforme circuitul într-un numărător Johnson cu auto-amorsare.

*Rezolvare*

a) Comportarea circuitului poate fi descrisă cu ajutorul a două grafuri independente :



b) Primul din cele două cicluri repetitive este cel al unui numărător Johnson. Pentru autoamorsare este necesar să cuplăm circuitele adiționale care, în caz că circuitul evoluează în cel de-al doilea ciclu.

să îl aducă în primul ciclu. Dacă alegem starea 13(1101) pentru a face tranziția între cicluri avem avantajul că se pot utiliza doar ultimii 3 biți 101, deoarece nu se regăsește în nici una din stările numărătorului Johnson. Schema este prezentată în figura 3.67. Se decodifică  $Q_B \bar{Q}_C Q_D$  comandând prin  $MC = 1$  încărcarea în paralel a unui număr aparținând primului ciclu (de exemplu 1111).

**Problema 3.70.** Utilizând 2 registre 7495 să se realizeze un numărător Johnson cu autoamorsare.

**Problema 3.71.** Se dă schema din figura 3.68.

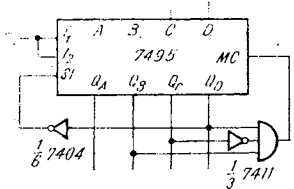


Fig. 3.67.

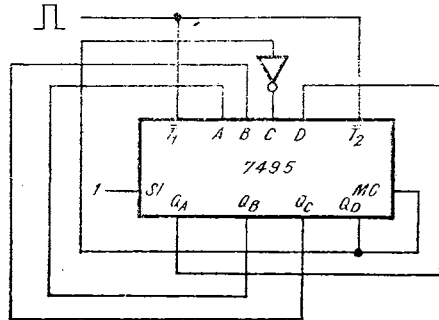


Fig. 3.68.

a) Care este graful de tranziție dacă registrul are starea inițială 0000 ?

b) Există o stare de blocare a funcționării ?

*Soluție*

a) Vezi figura 3.69.

b) Nu există.

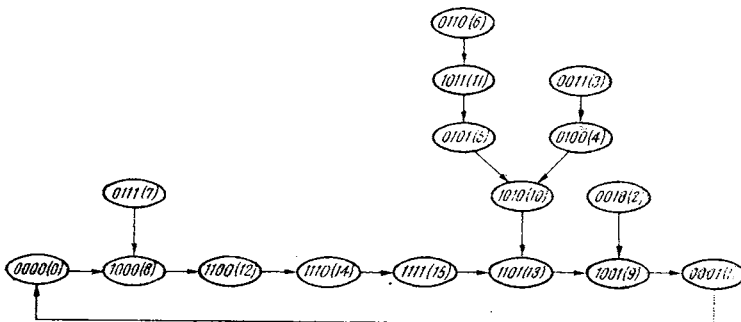


Fig. 3.69.

**Problema 3.72.** Să se conecteze un registru 7495 astfel încît să deplaseze serie către stînga.

*Soluție*

Vezi figura 3.70. Datele intră prin  $D$  și sînt deplasate  $Q_D \rightarrow Q_C \rightarrow Q_B \rightarrow Q_A$ .

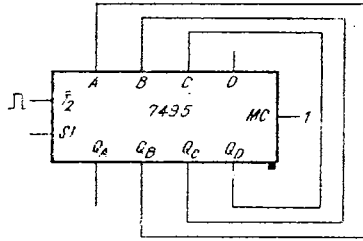


Fig. 3.70.

**Problema 3.73.** Să se realizeze cu registre 7495 un registru de deplasare de 8 biți care să deplaseze comandat, spre stînga și spre dreapta. Nu se vor utiliza circuite adiționale.

*Indicație*

Se utilizează  $2 \times 7495$  conectate, fiecare ca în figura 3.70 și legate în cascadă pentru deplasare stînga. Pentru deplasare dreapta se conectează  $Q_D$  de la primul registru la intrarea serie (SI) a celui de-al doilea.

Comanda de deplasare stînga sau dreapta se realizează cu  $MC = 1$ , respectiv 0.

**Problema 3.74.** Utilizînd în principal 2 registre 7495 și un DCD să se realizeze un generator programabil de trenuri de  $N$  impulsuri, ca în exemplul din figura 3.71 (în care  $N = 3$ ). Se va considera  $N$  maximum 7



Fig. 3.71.

Pauza din trenurile de impulsuri va fi de o perioadă a impulsului de tact.

*Rezolvare*

Numărul de impulsuri al trenului,  $N$ , este decodificat, trecînd în 0 ieșirea corespunzătoare a DCD (fig. 3.72). Impulsurile de tact se trans-

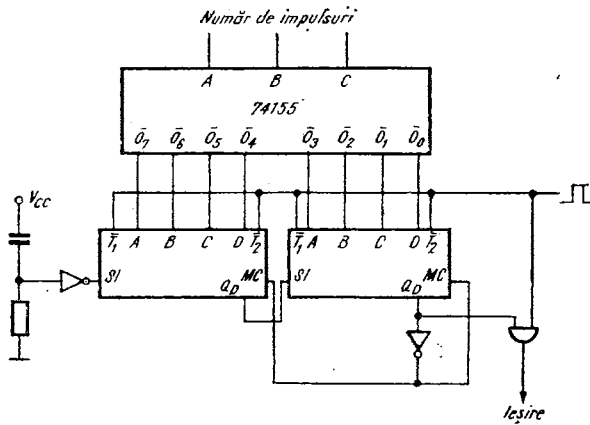


Fig. 3.72.

mit la ieșire pînă cînd acest 0 ajunge, prin deplasare serie, în  $Q_D$ . Semnalul de ieșire rămîne pe 0 și, următorul impuls de tact reînscris în paralel în registre. De exemplu, pentru  $N = 3$  se înscris în registru cuvîntul 11110111. Deci, primele 3 tacturi trec prin poarta SI, al 4-lea este oprit și, la al 5-lea se reia transmiterea impulsurilor.

Circuitul de integrare și inversorul au rolul de a amorsa corect schema la conectarea sursei de alimentare.

**Problema 3.75.** Să se reia problema anterioară utilizînd în loc de un DCD un MUX de 8 căi.

#### Rezolvare

Inițial se încarcă în registre (figura 3.73) configurația 01111111 care este deplasată către dreapta la fiecare front pozitiv al impulsurilor

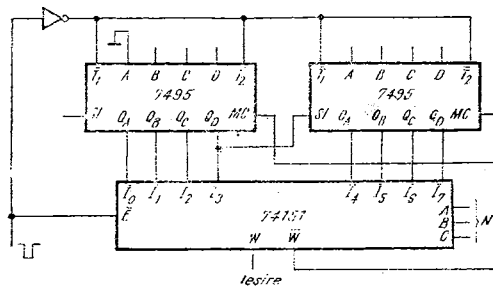


Fig. 3.73.

de tact. Acestea se transmit la ieșirea  $W$  a MUX pînă cînd 0 ajunge în registre în poziția indicată de numărul  $N$ . În acest moment,  $W$  trece în 0 și  $\overline{W} = 1$  comandă reincărcarea paralelă a registrelor.

**Problema 3.76.** Să se rezolve problemele 3.74 și 3.75 în cazul unor trenuri de  $N = 15$  impulsuri.

*Indicație*

Se utilizează — în principal  $4 \times 7495$  și  $1 \times 74154$ , respectiv  $1 \times 74150$ .

**Problema 3.77.** Utilizînd un circuit 7495 să se realizeze un divizor cu 3.

**Problema 3.78.** Utilizînd un circuit 7495 și porți logice adiționale să se implementeze un divizor cu 5.

**Problema 3.79:** Fie generorul de cod corector din figura 3.74.

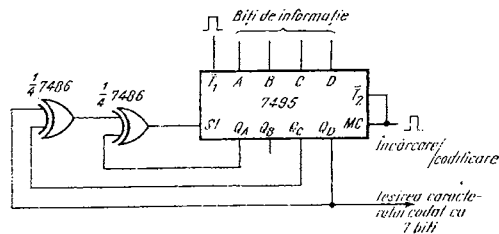


Fig. 3.74.

După încărcarea paralel a biților de informație se aplică 7 impulsuri de tact la intrarea  $T_1$ ,  $MC$  trecînd în 0.

Se cere :

- Să se găsească expresia biților de corecție  $C_1$ ,  $C_2$ ,  $C_3$ .
- Să se deducă tabela de stări arătînd cuvîntul codificat de 7 biți corespunzător fiecăreia din cele 16 combinații de intrare.

*Soluție*

$$a) C_1 = Q_A \oplus Q_C \oplus Q_D$$

$$C_2 = Q_C \oplus Q_B \oplus C_1$$

$$C_3 = Q_A \oplus Q_B \oplus C_2$$



b)	$C_3$	$C_1$	$C_2$	$Q_A$	$Q_B$	$Q_C$	$Q_D$
	1	1	1	0	0	0	1
	0	0	1	0	0	1	0
	1	1	0	0	0	1	1
	0	1	0	0	1	0	0
	1	0	1	0	1	0	1
	0	1	1	0	1	1	0
	1	0	0	0	1	1	1
	0	1	1	1	0	0	0
	1	0	0	1	0	0	1
	0	1	0	1	0	1	0
	1	0	1	1	0	1	1
	0	0	1	1	1	0	0
	1	1	0	1	1	0	1
	0	0	0	1	1	1	0
	1	1	1	1	1	1	1

Starea 0000 produce blocarea circuitului.

**Problema 3.80.** Se dă generatorul de succesiuni pseudoaleatoare din figura 3.75.

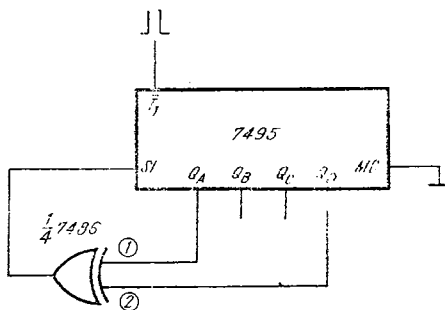


Fig. 3.75.

a) Să se determine succesiunea de stări  $Q_A Q_B Q_C Q_D$  și să se explice de ce starea 0000 nu aparține ciclului de evoluție a circuitului.

b) Să se repete punctul a) legind ieșirea  $Q_C$  la intrarea 1 a circuitului SAU EXCLUSIV.

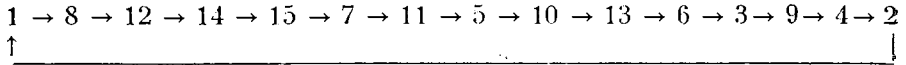
c) Se conectează  $Q_B$  la intrarea 1 a circuitului SAU EXCLUSIV. Se vor deduce ciclurile de funcționare ale circuitului.

d) Se va explica de ce la punctele a) și b) se obțin cicluri de lungime maximă.

e) Se vor repeta punctele a), b) și c) înlocuind pearta cu SAU EXCLUSIV NEGAT.

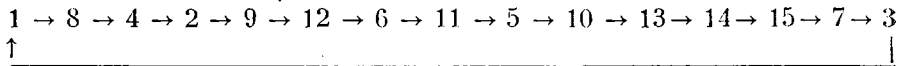
*Rezolvare*

a) Pentru  $Q_A = 2^3$ ,  $Q_B = 2^2$ ,  $Q_C = 2^1$  și  $Q_D = 2^0$  rezultă următorul graf :

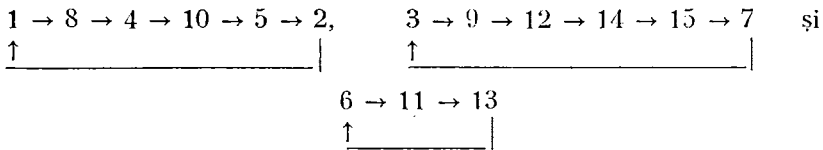


Este un generator de succesiuni pseudoaleatoare de lungime maximă. La apariția stării 0000, generatorul se blochează deoarece circuitul XOR înscris continuu 0 pe ȘI. Pentru funcționarea corectă trebuie completată schema cu circuit de deblocare.

b) Rezultă tot o succesiune pseudoaleatoare de lungime maximă :



c) Se obțin trei cicluri pseudoaleatoare :



**Problema 3.81.** Să se determine succesiunile pseudoaleatoare de lungime maximă pentru un generator cu 3 celule  $Q_A$ ,  $Q_B$ ,  $Q_C$ .

**Problema 3.82.** a) Să se descrie prin grafuri de tranziție funcționarea sistemului secvențial din figura 3.76.

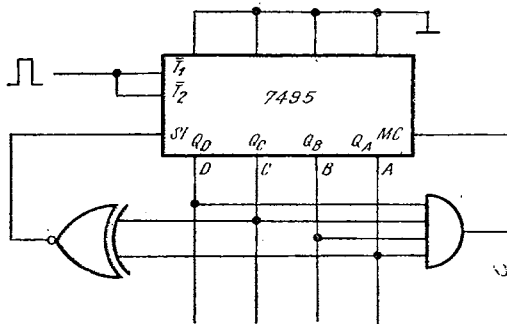


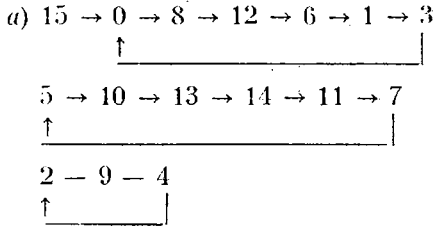
Fig. 3.76.

b) Să se explice rolul circuitului ȘI.

c) Ce se întâmplă dacă se înlocuiește circuitul ȘI cu  $\overline{A + B \cdot C + D}$ .

*Soluție*

Pentru  $A = 2^3$ ,  $B = 2^2$ ,  $C = 2^1$ ,  $D = 2^0$ , rezultă următoarele grafuri :



Deci circuitul este un generator de succesiuni pseudoaleatoare fără stare de blocare.

- b) Circuitul ȘI împiedică blocarea schemei în starea 15.
- c) Nu e corect : mai este necesară o negare.

**Problema 3.83.** Să se descrie prin grafuri de tranziție evoluția sistemului secvențial din figura 3.77 pentru cele 4 configurații  $AB$ .

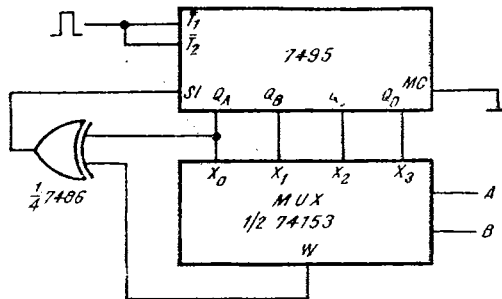


Fig. 3.77.

*Indicație*

Funcționarea este următoarea :

$AB$

00 Blocare

01 Generator SPA de lungime maximă ( $Q_A \oplus Q_B$ )

10 Generator SPA de lungime nemaximă ( $Q_A \oplus Q_C$ )

11 Generator SPA de lungime maximă ( $Q_A \oplus Q_D$ ).

Deci, schema reprezintă un generator de succesiuni pseudoaleatoare (GSPA), programabil cu ajutorul configurațiilor  $AB$ .

**Problema 3.84.** Să se explice funcționarea schemei din figura 3.78.

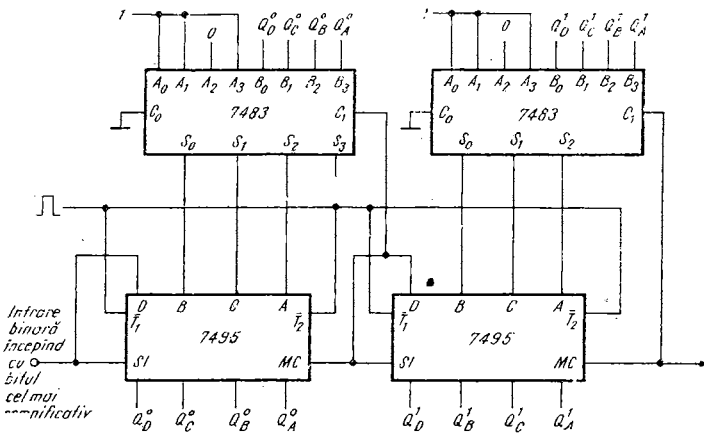


Fig. 3.78.

Cum se face inițializarea schemei ?

*Indicație*

Este un convertor serie binar — BCD.

**Problema 3.85.** Să se realizeze cu registre 7495 :

- un convertor paralel-serie de 8 biți ;
- un convertor serie-paralel de 8 biți.

**Problema 3.86.** Pentru schema din figura 3.79 să se deseneze formele de undă la ieșire, dacă în registru este inițial 0000 și se înscrie apoi 1101.

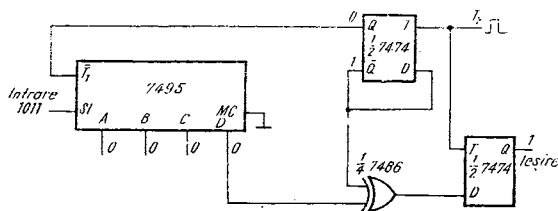


Fig. 3.79.

*Rezolvare*

Formele de undă sînt prezentate în figura 3.80.

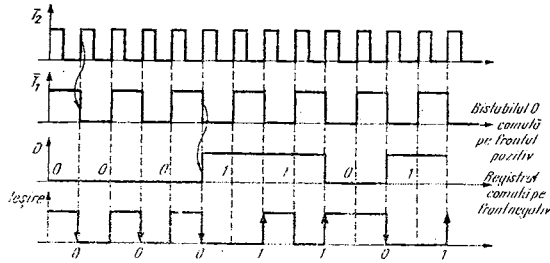


Fig. 3.80.

Este o schemă de codificare în fază : informația din  $D$  este transmisă la ieșire astfel încât, 0 este codificat prin front negativ și 1, prin front pozitiv, sincronizat cu ceasul  $T_1$ . Este o schemă de codificare în fază cu două ceasuri. Se utilizează la înregistrarea informației pe suport magnetic, deoarece în acest caz, codificarea prin nivel, nu este utilă întrucît nu produce variații de flux.

**Problema 3.87.** Să se proiecteze o schemă de decodificare a informației codificate în fază.

*Rezolvare*

O schemă posibilă este dată în figura 3.81.

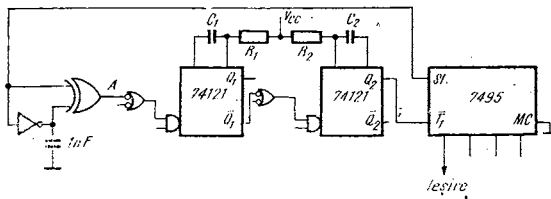


Fig. 3.81.

În figura 3.82 sînt prezentate formele de undă în principalele puncte din schemă.

Circuitele NU și SAU EXCLUSIV formează un detector de fronturi (vezi problemele 4.35, 4.38). Monostabilul  $Q_1$ , comutînd pe frontul negativ al detectorului are durata mai mare decît  $T/2$  pentru a se elimina tranzițiile ce nu conțin informație utilă. Monostabilul  $Q_2$  furnizează ceasul pentru registrul de deplasare. Trebuie să aibă durata mai mică de  $T/2$  pentru a se înscrie în registru informația utilă.

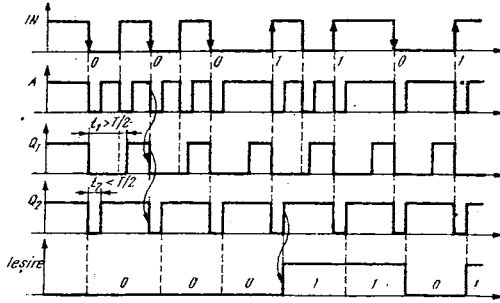


Fig. 3.82. \*

Pentru o rată de transfer de 5 000 biți/s rezultă  $T = 200 \mu\text{s}$ .

Alegînd  $t_1 = \frac{3}{4} T = 150 \mu\text{s}$  și  $C = 100 \text{ nF}$ , rezultă  $R = 1,8 \text{ K}\Omega$ .

Alegînd  $T_2 = \frac{1}{4} T = 50 \mu\text{s}$  și  $C = 10 \text{ nF}$ , rezultă  $R = 7,5 \text{ K}\Omega$ .

**Problema 3.88.** a) Să se demonstreze că schema cu 4 registre de deplasare serie din figura 3.83 realizează înmulțirea unui polinom  $P$  adus serie la intrare, începînd cu rangul cel mai semnificativ, cu polinomul  $1 + X^4 + X^8 + X^{16}$ .

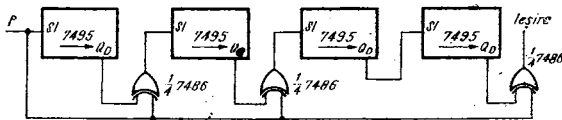


Fig. 3.83.

b) Să se modifice schema astfel încît să realizeze aceeași funcție dar să aibă sumatoarele conectate exterior lanțului de deplasare.

*Soluție*

- b) Vezi figura 3.84.

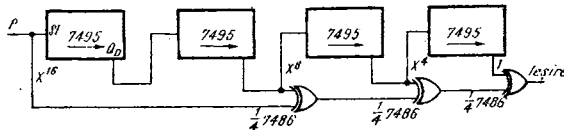


Fig. 3.84

**Problema 3.89.** Să se realizeze o schemă de împărțire a unui polinom prin  $1 + X^4 + X^8 + X^{16}$  avînd sumatoare incluse.

*Soluție*

Vezi figura 3.85.

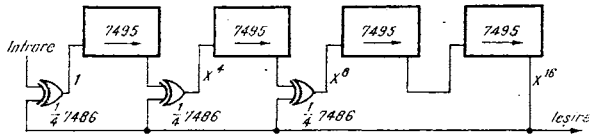


Fig. 3.85.

*Observație.* În standardele ECMA și ISO pentru prelucrarea datelor ce se înscriu pe caseta magnetică este prevăzut polinomul  $1 + X^2 + X^{15} + X^{16}$ .

**Problema 3.90.** Să se explice funcționarea comparatorului secvențial a două numere  $A$  și  $B$ , prezentat în figura 3.86. Cele două numere sînt stocate în două registre și sînt deplasate serial începînd cu bitul cel mai semnificativ.

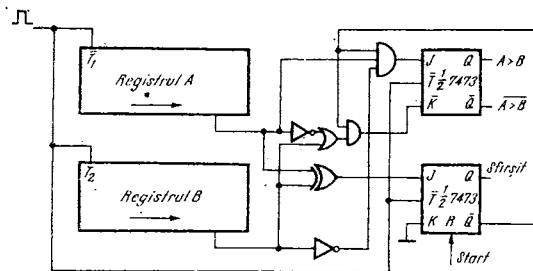


Fig. 3.86.

Propuneți o schemă care să realizeze aceeași funcție cu mai puține circuite integrate.

**Problema 3.91.** Să se proiecteze un sumator secvențial a două numere  $A$  și  $B$  de 8 biți.

*Soluție*

Vezi figura 3.87. Numerele inițial stocate în registrele  $A$  și  $B$  sînt deplasate serial începînd cu bitul cel mai puțin semnificativ. Transpor-

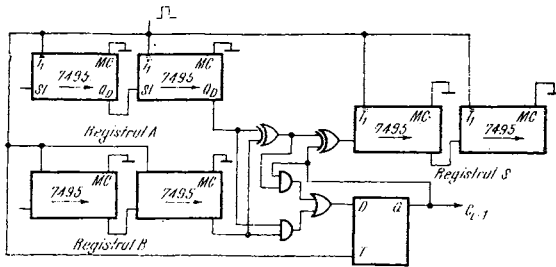


Fig. 3.87.

ul de la un bit la altul e memorat în bistabilul  $D$ . Rezultatul se înscrie în registrul  $S$ . Sînt necesare 9 circuite integrate.

**Problema 3.92.** Care sînt asemănările și deosebirile între memoria RWM (RAM) și memoria ROM (vezi problema 2.105) ?

*Rezolvare*

Memoria RWM (Read/Write Memory — memorie citește scrie) numită impropriu și RAM (Random Access Memory — memorie cu acces aleator) este într-un anume sens asemănătoare cu ROM-ul :

- există un DCD pentru adrese și o matrice de memorie ;
- organizarea externă este reprezentată prin numărul de cuvinte  $\times$  numărul de biți pe cuvînt ;
- capacitatea RAM-ului este determinată tot de numărul de biți memorați în matricea de memorie.

Deosebirile între RAM și ROM sînt următoarele :

- În timp ce ROM-ul conține în fiecare intersecție a matricei de memorie cîte un tranzistor aparținînd porților de ieșire, RAM-ul conține în fiecare intersecție cîte un bistabil. Se memorează astfel, nu o informație fixă, ci una variabilă, 0 sau 1 în fiecare poziție-bit. Bistabilii sînt de tip „latch” și memorează datele de la intrarea RAM-ului, dacă sînt selectați prin adresare și se dă comanda de scriere (Write).

- RAM-ul conține în plus față de ROM comanda de citire/scriere (R/W — Read/Write) și intrările în memorie.

- În timp ce la ROM-uri, lungimea cuvintelor este de 4 sau 8 biți, la RAM-uri este mai redusă, de 1 sau 4 biți.

- Matricea de memorie la RAM-uri este în majoritatea cazurilor pătrată (cu dimensiune  $n \times n$  biți), fapt care adaugă în structura internă un MUX al datelor la ieșire.



● Pentru capacități mici de memorare se utilizează RAM-uri TTL ( $16 \times 4,256 \times 1,1024 \times 1$  biți). Tehnologia MOS statică se utilizează între  $256 \times 1 - 4096 \times 1$  biți, iar capacitățile mari ( $1024 \times 1,4096 \times 1.8K \times 1,16K \times 1,64K \times 1$ ) sînt acoperite de memoriile MOS dinamice.

● La memoriile RAM dinamice există un regim de lucru în plus, acela de reîmprospătare (refresh) a datelor memorate la anumite intervale de timp, pentru a evita pierderea informației.

● Dacă unui RAM conținând o anumită informație i se întrerupe alimentarea la restabilirea acesteia, informația este pierdută. La ROM, în aceeași situație, datele rămîn în memorie.

**Problema 3.93.** Care este structura internă a unei memorii RAM 74S89 sau 74S189 organizată în  $16$  cuvinte  $\times 4$  biți pe cuvînt ?

*Rezolvare*

Structura internă este dată în figura 3.88. Se observă că bistabilii din matrice au cîte 3 comenzi : decodificarea adresei, intrarea și ieșirea.

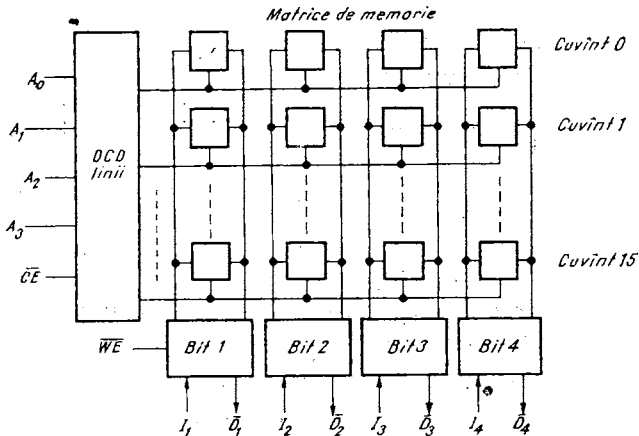


Fig. 3.88.

Funcționarea memoriei este permisă pentru  $\overline{CE}$  (Chip Enable) = 0 astfel :

- pentru  $\overline{WE}$  (Write Enable) = 1 are loc citirea pe  $\overline{D}_1 - \overline{D}_4$  ;
  - pentru  $\overline{WE} = 0$  se scriu în memorie datele de la intrările  $I_1 - I_4$ .
- Memoria inversează datele la citire.

**Problema 3.94.** Care este schema logică simplificată a unei memorii RAM organizată  $1024 \times 1$  biți (Intel 2102) ?

*Rezolvare*

Memoria conține un MUX pe ieșire (fig. 3.89) cu ajutorul căruia se transmit datele de pe cele 32 de căi de la ieșirea matricei de memorie, pe o singură cale : ieșirea memoriei,  $D_{OUT}$ .

**Problema 3.95.** Care este schema logică internă a unei memorii RAM organizată  $1024 \times 4$  cu intrările comune cu ieșirile (Intel 2114) ?

*Rezolvare*

Matricea de memorie este organizată  $64 \times 64$  biți (fig. 3.90).

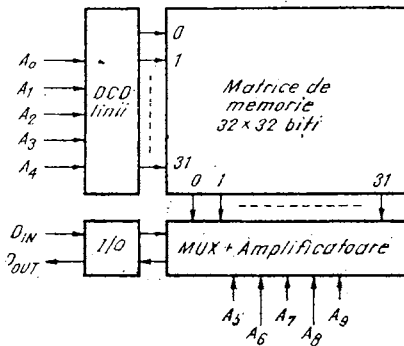


Fig. 3.89.

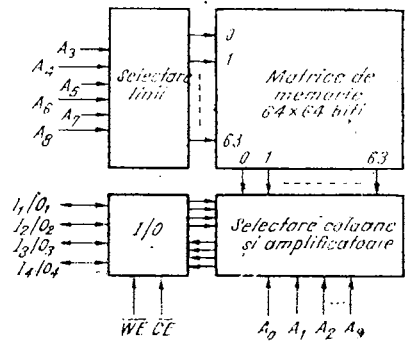


Fig. 3.90.

Memoriile de acest tip, cu intrările comune cu ieșirile, sînt utile în cazul cuplării pe BUS-uri bidirecționale.

**Problema 3.96.** Care este schema logică internă a unei memorii RAM organizată  $4096 \times 1$  cu intrările comune cu ieșirile, Intel 2147.

*Indicație*

Matricea de memorie are  $64 \times 64$  biți și ieșiri multiplexate.

**Problema 3.97.** Să se proiecteze o memorie RAM de 4K cuvinte a 8 biți (un octet) cu circuite integrate Intel 2114 ( $1024 \times 4$  biți).

*Rezolvare*

Se utilizează 8 circuite integrate (fig. 3.91). Este necesară extensia, atît a numărului de cuvinte, cît și a numărului de biți pe cuvînt.

Procedeele sînt similare cu cele de la extinderea capacității memoriei ROM (vezi problemele 2.109—2.112) :

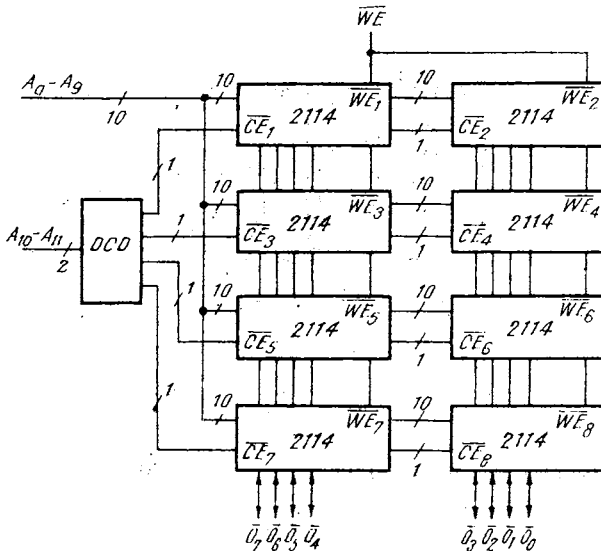


Fig. 3.91.

- Se conectează în paralel adresele  $A_0-A_9$ ,  $\overline{WE}$ , intrările și ieșirile  $\overline{O}_0-\overline{O}_3$ , intrările și ieșirile  $\overline{O}_4-\overline{O}_7$  la toate circuitele integrate;
- Se decodifică biții suplimentari de adresă  $A_{10}-A_{11}$  comandîndu-se dezinhibarea pe  $\overline{CE}$ .

**Problema 3.98.** Să se realizeze o memorie de  $4096 \times 4$  biți ajutorul unor RAM-uri statice de  $4096 \times 1$  biți (2147).

*Indicație*

Se utilizează 4 circuite de memorie RAM comandate în paralel pe adresă,  $\overline{CE}$  și  $\overline{WE}$ .

**Problema 3.99.** Să se proiecteze o memorie cu o capacitate de  $8\text{ K} \times 1$  biți cu RAM-uri statice de  $1\text{ K} \times 1$  biți.

*Rezolvare*

Se utilizează  $8 \times 2102$  conectate ca în figura 3.92.

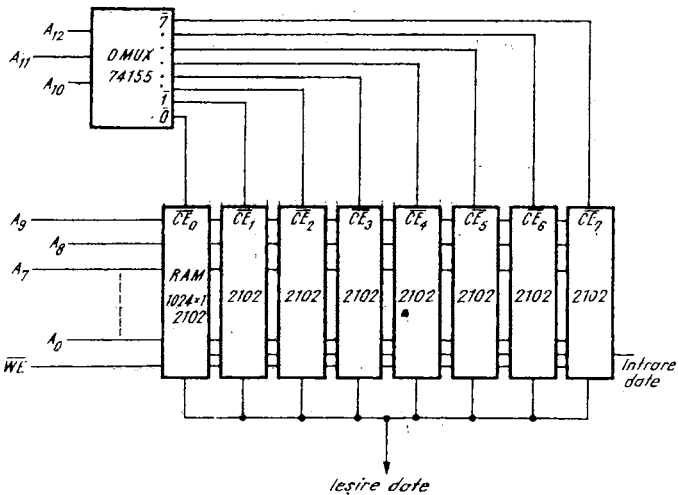


Fig. 3.92.

**Problema 3.100.** Să se realizeze cu CI RAM statice de  $4096 \times 1$  (2147) o memorie adresabilă pe 13 biți, cu cuvinte de 4 biți.

*Indicație*

Se utilizează  $8 \times 2147$ . Este necesară extensia numărului de cuvinte și a numărului de biți pe cuvânt.

**Problema 3.101.** Să se realizeze o memorie RAM de 16 K octeți ( $16k \times 8$  biți) cu CI de tip :

- Intel 2102 ;
- Intel 2114 ;
- Intel 2147.

*Indicație*

Este necesar următorul număr de memorii :

- $128 \times 2102$
- $32 \times 2114$
- $32 \times 2147$ .

Pentru reducerea schemei este indicat să se utilizeze circuite integrate RAM de capacitate cât mai mare. Acestea sînt mai economice : costul pe bit scade la creșterea capacității. În plus crește siguranța în funcționare și realizarea practică devine mai simplă.

**Problema 3.102.** Să se definească următorii parametri ai funcționării memoriilor RAM statice :

- timpul de acces,  $t_A$  ;

- b) ciclul de citire,  $t_{RC}$  (Read Cycle) :  
 c) ciclul de scriere,  $t_{WC}$  (Write Cycle).

*Rezolvare*

a) Timpul de acces,  $t_A$ , este intervalul de la schimbarea adresei la intrare, la apariția datelor la ieșire. Este de zeci de nanosecunde la memoriile TTL, de sute de nanosecunde la cele MOS.

b)  $t_{RC}$  este intervalul minim între două schimbări consecutive de adrese în vederea citirii.

c)  $t_{WC}$  este timpul minim între două schimbări succesive de adrese în vederea scrierii.

În tabelul următor se dau acești timpi pentru câteva tipuri de memorii :

RAM	$t_A = t_{RC} = t_{WC}$ (ns)
2102A	350
2102A-2	250
2114-2	200
2147	90

**Problema 3.103.** a) Care sînt diferențele între structura internă a memoriei RAM dinamice de  $4096 \times 1$  (tip intel 2107, Signetics 2680) și cea a unei memorii statice cu aceeași organizare (Intel 2147).

b) Cum se poate face ciclul de reîmprospătare (refresh), necesar pentru a menține informația în memoria dinamică ?

*Rezolvare*

Structura internă a memoriei 2107 este prezentată în figura 3.93. Față de memoria statică, aceasta conține în plus blocul GCT (Generator și Control al Tactului) care comandă selectorul de linii, de coloane și blocul de intrare-ieșire (I/O), furnizînd impulsurile necesare celulei dinamice. Fiecare celulă a matricei de memorie stochează datele cu ajutorul unei capacități MOS (vezi problemele 4.62, 4.65). Aceasta trebuie reîncărcată la intervale de timp mai mici de 2 ms (la 70°C) pentru a nu pierde informația memorată.

b) Reîmprospătarea informației pentru RAM 2107 se face prin citirea conținutului celulelor. Deoarece în matricea de memorie se adresează

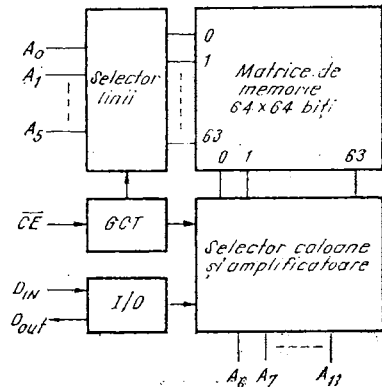


Fig. 3.93.

simultan toate cele 64 de celule de pe o linie, reîmprospătarea constă în 64 de citiri, la adresele selectate prin  $A_0-A_5$ . Zona de adresare  $A_6-A_{11}$ , nu contează deoarece nu ne interesează transmiterea la ieșire a biților citiți.

Cele 64 de citiri pentru reîmprospătare trebuie făcute deci, o dată la cel mult 2 ms prin una din metodele următoare :

1. În timp de 2 ms se citesc toate cele 64 linii la intervale de timp mai mici sau egale cu  $31,25 \mu\text{s}$ . Deci, la cel mai târziu  $31,25 \mu\text{s}$  se întrerupe orice altă operație cu memoria dând prioritate citirii de reîmprospătare.

2. Se efectuează cel puțin o dată la  $2 \text{ ms}$ , într-un interval compact toate citirile de reîmprospătare. Astfel, dacă timpul minim admis între 2 citiri succesive este  $t_A$  avem durata reîmprospătării de  $64 t_A$ . În restul timpului,  $2 \text{ ms} - 64 t_A$ , memoria poate fi comandată normal.

Dacă memoria este cuplată cu un microprocesor, primul mod de „refresh” este indicat pentru ca dialogul cu RAM-ul să nu fie întrerupt pe intervale mari de timp ( $64 t_A$ ).

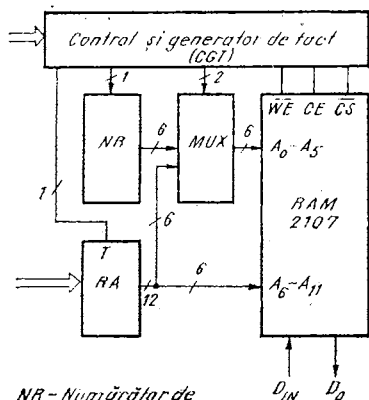
**Problema 3.104.** Să se proiecteze o schemă de adresare și comandă a reîmprospătării pentru memoria RAM 2107.

#### Indicație

Schema bloc este prezentată în figura 3.94. NR numără adresele  $A_0-A_5$  de reîmprospătare într-unul din cele 2 regimuri discutate în problema anterioară : pentru primul caz, CGT trebuie să furnizeze cite un impuls de numărare la fiecare  $31,25 \mu\text{s}$  ; pentru cel de-al doilea, trebuie să furnizeze trenuri de 64 de impulsuri la fiecare 2 ms.

MUX transmite memoriei fie conținutul numărătorului în timpul ciclului de reîmprospătare, fie pe cel al registrului de adrese în timpul scrierii sau citirii.

CGT controlează funcționarea memoriei și a circuitelor adiționale, distribuind impulsurile în sistem.



NR - Numărător de reîmprospătare  
RA - Registru de adrese

Fig. 3.94.

**Problema 3.105.** Care sînt deosebirile principale între memoria RAM dinamică 2104 (Intel) și 2107 anterior prezentată :

- ca adresare
- ca schemă internă.

*Rezolvare*

a) Ambele aceste memorii sînt organizate în format de  $4096 \times 1$  biți. Deosebirea, ca adresare, constă în aceea că, la memoria 2104 sînt afectate doar 6 terminale pentru recepționarea adresei. Deci, cei 12 biți de adresare se vor transmite multiplexat memoriei : întîii 6 biți pentru adresarea liniilor, apoi următorii 6, selectînd coloana. Aceasta face posibilă încapsularea memoriei 2104 într-o capsulă cu numai 16 terminale (2107 are 22). Adresarea multiplexată permite încapsularea comodă a memoriilor de capacitate mai mare : 16 K, 64 K biți.

b) Ca urmare a adresării multiplexate, la memoria 2104 e necesară introducerea a două registre tampon (latch — fig. 3.95) și a două generatoare de tact. Înscirerea adresei se face cu semnalele  $\overline{RAS}$  (Row Address Strobe — strobul adresei de linie) și  $\overline{CAS}$  (Column Address Strobe — strobul adresei de coloană), pe front negativ.

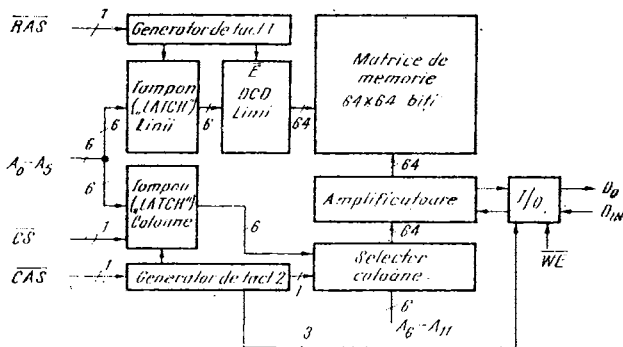


Fig. 3.95.

**Problema 3.106.** Să se explice particularitățile de adresare a memoriei dinamice de  $16K \times 1$ . Se vor lua ca referință modelele echivalente F16K (Fairchild), Z6116 (Zilog), 2117 (Intel).

*Indicație*

Memoria de 16K se adresează multiplexat pe 7 biți, printr-un procedeu similar cu RAM-ul 2104 (vezi problema anterioară).

**Problema 3.107.** a) Explicați, ținînd cont de schema internă a RAM-ului dinamic 2117 ( $16K \times 1$ ), cum este posibil ca adresarea multiplexată să nu mărească timpul de acces.

b) Când poate fi aplicat semnalul de  $\overline{\text{CAS}}$ ? Ce sens au limitele  $t_{RCDmin}$ ,  $t_{RCDmax}$  ( $t_{RCD}$ :  $\overline{\text{RAS}}$  to  $\overline{\text{CAS}}$  Delay Time – timp de întârziere a lui  $\overline{\text{CAS}}$  față de  $\overline{\text{RAS}}$ ) ?

c) Explicați cum crește timpul de acces al memoriei în cazul depășirii limitei  $t_{RCDmax}$ . Dat fiind natura dinamică a circuitului, vedeți yreo limitare a acestui timp ?

### Rezolvare

a) Schema bloc internă este similară cu cea din figura 3.95, cu deosebirea că adresarea se face pe 7 biți, matricea de memorie fiind de  $128 \times 128$  biți. Este evident că timpul de acces față de primul grup de biți de adresă, deci față de  $\overline{\text{RAS}}$  ( $t_{RAC}$ ) va fi mai mare decât cel al grupului de biți de adresă comandați de  $\overline{\text{CAS}}$  ( $t_{CA}$ ) deoarece conține în plus propagări prin matricea de memorie și amplificatoare. Schema internă permite blocarea acțiunii adresei de coloană până ce rîndul comandat de  $\overline{\text{RAS}}$  ajunge la amplificatoare. În felul acesta, timpul de acces va fi dat numai de propagările interne ale primilor 7 biți de adresă înscrși cu  $\overline{\text{RAS}}$ .

b) S-a arătat mai sus că schema internă dispune de circuite care pot să inhibe acțiunea de selecție a adreselor comandate prin  $\overline{\text{CAS}}$ . Datorită lor semnalul  $\overline{\text{CAS}}$  poate fi aplicat după  $\overline{\text{RAS}}$  la un interval minim,  $t_{RCD}$ , format din (vezi ciclul de citire al memoriei 2117 din figura 3.96) :

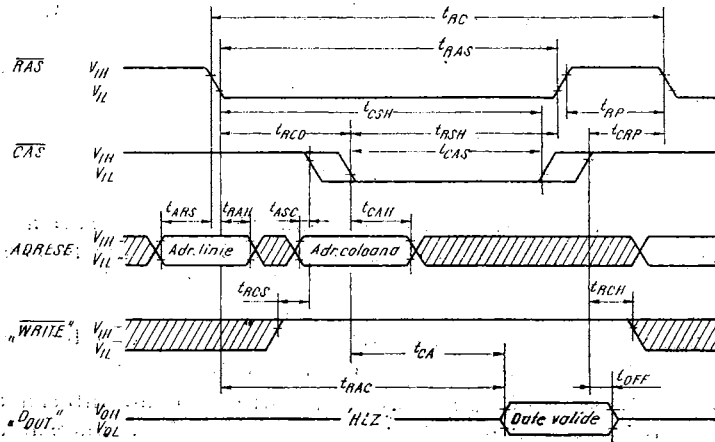


Fig. 3.96.



—  $t_{RAH}$ , timpul cît trebuie menținută pe intrări adresa rîndului, după frontul negativ al RAS (Row Address Hold time). După trecerea acestui timp poate fi schimbată adresa de rînd cu cea de coloană.

—  $t_{ASC}$ , timpul cît trebuie așteptat de la apariția tranzițiilor adresei de coloană pînă la tranziția negativă a  $\overline{CAS}$  (Column Address Set-up time). Acest timp are valoare negativă (vezi anexa), ceea ce înseamnă că adresa de coloană se poate schimba după apariția  $\overline{CAS}$ .

Deci rezultă că  $t_{RCDmin} = t_{RAH}$ .

Limita maximă a lui  $t_{RCD}$  o constituie momentul cînd rîndul din matricea de memorie selectat de primii 7 biți de adresă, ajunge la amplificatoarele de ieșire. Din foaia de catalog,  $t_{RCDMax}$  este determinat de  $t_{RAC} - t_{CA}$ .

c) Depășirea valorii  $t_{RCDMax}$  are ca urmare prelungirea timpului de acces al memoriei cu  $\Delta t = t_{RCD} - t_{RCDMax}$ . Natura dinamică a memorării informației impune pentru  $\Delta t$  o limită maximă de ordinul microsecundelor.

**Problema 3.108.** Calculați timpul minim pentru ciclul de citire al RAM-ului dinamic de 16K. Explicați diferența față de data de catalog  $t_{RCmin}$  (Read Cycle time — timpul ciclului de citire).

*Indicație*

În figura 3.96 este prezentat un ciclu de citire. Valoarea  $t_{RCmin}$  din catalog indică funcționarea în întreaga gamă de temperatură, 0–70°C și nu poate fi obținută prin însumarea valorilor minime ale timpilor din figură.

**Problema 3.109.** a) Explicați modalitatea de înscriere a datelor în RAM-ul de 16K × 1.

b) Există vreo prioritate între semnalele de  $\overline{WRITE}$  și  $\overline{CAS}$ ? Față de cînc se măsoară timpii de „set-up“ și „hold“ ai datelor indicați în foile de catalog?

c) Ce implicații are prezența unuia, înaintea celuilalt, a semnalelor  $\overline{CAS}$  și  $\overline{WRITE}$ ?

d) Ce restricții determină  $t_{WCSmin}$  (timpul de set-up al comenzii de scriere față de  $\overline{CAS}$ ) și  $t_{CWDmin}$  (timpul de întârziere al  $\overline{CAS}$  față de  $\overline{WRITE}$ )?

*Rezolvare*

a) Informația existentă la  $D_{IN}$  este înscrisă în memorie cu condiția ca, semnalul  $\overline{RAS}$  să fie activ (nivel 0) și atît  $\overline{CAS}$  cît și  $\overline{WRITE}$  să fi efectuat o tranziție negativă.

b) Nu există nici o prioritate a unuia din cele 2 semnale, ultimul front negativ, generînd semnalul de înscriere pentru tamponul (latch) de intrare date.



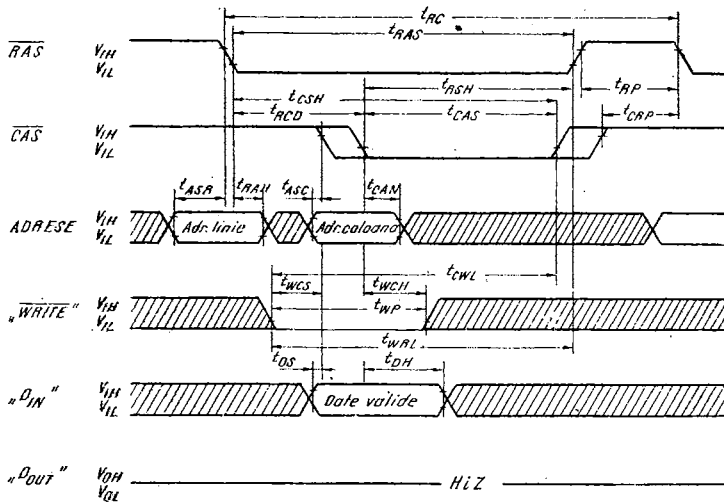


Fig. 3.98.

Dacă  $t_{CWD} > t_{CWDmin}$  se execută un ciclu de citire-scriere și ieșirea de date va conține informația citită din celula selectată.

În cazul cînd nici una din condițiile de mai sus nu se îndeplinește, prezența datelor la ieșire, după timpul de ieșire, va fi nedeterminată, fără ca aceasta să afecteze funcționarea memoriei.

**Problema 3.110.** a) Ce înseamnă mod de lucru în pagină la memoria dinamică  $16K \times 1$  biți?

b) Explicați cum se realizează ciclurile de citire și de scriere în modul de lucru în pagină.

c) Cum se extinde acest mod de operare de la un circuit, la o matrice de circuite de memorie?

*Rezolvare*

a) Modul de lucru în pagină (Page mode) presupune cicluri repetate la adrese diferite de coloană, păstrîndu-se constantă adresa de linie. Se realizează, prin înscrierea adresei de rînd în circuit și menținerea semnalului RAS activ (nivel 0) pe parcursul tuturor ciclurilor în care adresa de rînd va rămîne neschimbată. În mod evident, timpul de acces rămîne, practic pentru toate ciclurile  $t_{CA}$ , ceea ce înseamnă o reducere substanțială.

b) În figura 3.99 se prezintă ciclul de citire pentru modul de lucru în pagină; se poate observa cum, după a doua aplicare a semnalului

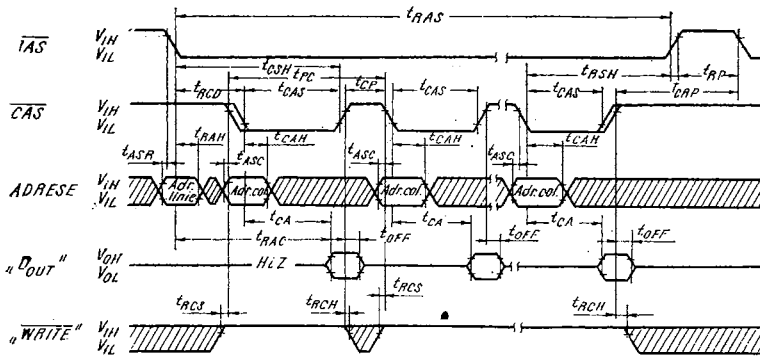


Fig. 3.99.

CAS, timpul de acces este dat doar de  $t_{CA}$ , fiind mult mai scurt. În figura 3.100 se exemplifică un ciclu de scriere de tipul „Early Write“

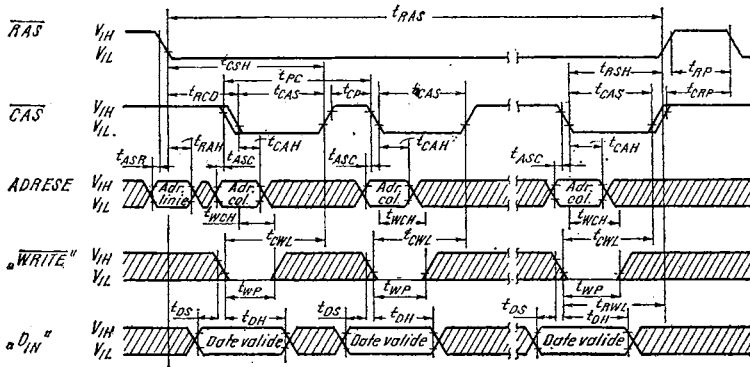


Fig. 3.100.

(vezi problema anterioară) pentru modul de lucru în pagină. Trebuie remarcat că :

- parametrul  $t_{RASMax}$  limitează timpul în care este posibilă operația cu aceeași adresă de rând, fără aplicarea unui nou semnal RAS. Suma dintre mărimea maxim admisibilă a pulsului de RAS și durata minimă a pulsului de preîncărcare a RAS-ului determină ciclul maxim în mod pagină.

● parametrul  $t_{PCmin}$  (Ciclu de Pagină) se referă la suma dintre durată minimă a unui puls de  $\overline{CAS}$  și cea de preîncărcare a  $\overline{CAS}$ -ului. Deci :

$$t_{PCmin} = t_{CASmin} + t_{CPmin}$$

c) La un singur circuit integrat putem baleia numai 128 adrese de coloană la modul de lucru în pagină. Pentru a extinde la mai multe CI se va aplica la toate circuitele semnalul  $\overline{RAS}$  pentru a stroba adresele de rând, iar codurile exprimând poziția CI în matrice vor fi decodificate servind drept semnal de selecție pe coloane,  $\overline{CAS}$ ; numai circuitele care primesc atât  $\overline{CAS}$  cât și  $\overline{RAS}$  vor executa un ciclu de scriere sau citire.

**Problema 3.111.** Explicați cum se poate face menținerea (refresh) informației în memoria dinamică de tip Intel 2117.

*Rezolvare*

Trebuie adresate toate cele 128 linii în timp de 2 ms. Aceasta se poate face prin :

- a) cicluri normale de memorie realizate cu frecvența respectivă ;
- b) efectuarea numai a unor cicluri de  $\overline{RAS}$ . Astfel se poate ajunge la o reducere substanțială a consumului de putere ; alimentarea memoriei la  $V_{CC}$  poate fi complet întreruptă în timpul efectuării acestor cicluri fără a afecta reîmprospătarea.

**Problema 3.112.** Indicați o metodă de proiectare a sistemelor logice secvențiale sincrone, SLSS.

*Rezolvare*

Soluția acestei probleme constituie o sinteză a tot ceea ce s-a prezentat anterior în această lucrare.

În proiectarea SLSS se pornește de la modelarea acestora ca sisteme cu stări finite (vezi figura 3.101 și problemele 1.1 și 1.2).

Etapele urmărite în proiectare sînt :

I. Stabilirea modalității de definire a sistemului  $S$  ce trebuie proiectat : graf, tabel de tranziții, organigramă, forme de undă (vezi problemele 1.3 — 1.9) sau, dacă este cazul, exprimarea funcțiilor caracteristice printr-unul din aceste moduri.

II. Minimizarea SLSS prin reducerea spațiului de stare,  $Q$  (vezi problemele 1.11 și 1.12). Se obține astfel un sistem  $S_e$ , echivalent cu  $S$  dar cu mai puține stări.

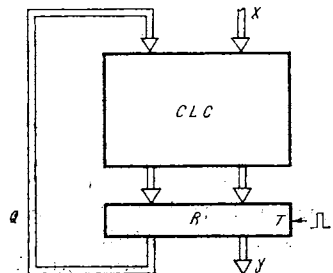


Fig. 3.101.

III. Explicarea funcțiilor caracteristice,  $f$  și  $g$ , ale SLSS redus anterior : graf, tabel de tranziții ș.a.

IV. Stabilirea configurației particulare a sistemului  $S_e$  (număr de intrări, ieșiri, stări) și minimizarea prin codificarea în spațiile  $X, Y, Q$ .

V. Stabilirea dimensiunii registrului  $R$  (egală cu numărul de stări + numărul de ieșiri) și a tipului de bistabili cu care se implementează.

VI. Stabilirea modelului Mealy particular al sistemului  $S_e$ .

VII. Construirea tabelului de tranziții exprimat binar pentru sistemul  $S_e$ .

Acest tabel este de forma următoare :

Momentul $t$						Momentul $t+1$					
Coduri intrări			Coduri stări			Coduri stări			Coduri ieșiri		
$x'$	$x''$	...	$q_1$	$q_2$	...	$q_1$	$q_2$	...	$y'$	$y''$	...

În partea stîngă (momentul  $t$ ), tabelul se completează cu codurile intrărilor și stărilor, notate în ordine crescătoare, în binar.

În dreapta (momentul  $t + 1$ ), tabelul se completează cu codurile stărilor și ieșirilor corespunzătoare vectorilor din stînga conform cu definirea sistemului  $S_e$ , stabilită în etapa III.

VIII. Stabilirea tabelului de adevăr al CLC asociat sistemului  $S_e$ .

Pentru aceasta se completează tabelul de tranziții exprimat binar cu tabelul de tranziții al bistabililor registrului  $R$ .

● Dacă bistabilii sînt de tip  $D$  lasă să treacă la ieșire, datele de la intrare, pe frontul ceasului. Deci, tabelul de adevăr al CLC va fi identic cu cel de tranziții.

● Bistabilii de tip  $JK$  prelucrează informația de la intrări conform următorului tabel de tranziție :

$Q^t$	$J$	$K$	$Q^{t+1}$
0	0	×	0
0	1	×	1
1	×	1	0
1	×	0	1

Deci, în partea dreaptă a tabelului de tranziții exprimat binar, pentru fiecare  $q$ , trebuie completate corespunzător coloanele  $J$  și  $K$ .

● Pentru utilizarea unor registre cu funcționare mai complexă (7495), a unor numărătoare sincrone etc., trebuie să se țină cont de tabelul de tranziții al acestora.

IX. Implementarea CLC asociat lui  $S_6$  printr-una din metodele prezentate în capitolul 2.

**Problema 3.113.** Să se proiecteze un numărător binar sincron de la 0 la 7, utilizând bistabili de tip  $D$ .

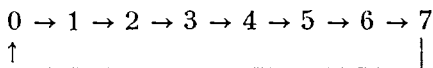
*Rezolvare*

Se aplică algoritmul de proiectare descris în rezolvarea problemei anterioare. Etapele sînt :

I. Se remarcă faptul că sistemul,  $S$ , care trebuie proiectat, nu este definit prin nici una din formele prezentate în capitolul 1. Comportarea lui  $S$  este descrisă în enunțul problemei.

II. Nu se poate reduce spațiul de stare al sistemului  $S$ , deci, rezultă că  $S \equiv S_6$ .

III. Sistemul avînd compararea descrisă în enunțul problemei poate fi reprezentat prin graful de tranziții următor :



IV. Se observă că sistemul nu are intrări, are 8 stări (codificabile cu 3 biți) și spațiul de ieșire este identic cu cel de stare :  $\{X\} \equiv \emptyset$ ,  $\{Y\} \equiv \{Q\}$ .

Codificarea stărilor se va face prin exprimarea în binar a numerelor zecimale respective.

V. Registrul  $R$  va avea 3 bistabili de tip  $D$ .

VI. Modelul Mealy particular al sistemului, ținînd cont de etapele IV și V, va fi cel din figura 3.102.

VII. Tabelul de tranziție exprimat binar este obținut prin completarea în partea dreaptă cu numărul din stînga adunat cu 1, conform grafului sistemului :

Starea prezentă (t)			Starea următoare (t+1)		
A ( $2^2$ )	B ( $2^1$ )	C ( $2^0$ )	$D_A$	$D_B$	$D_C$
0	0	0	0	0	1
1	0	0	0	1	0
2	0	1	0	1	1
3	0	1	1	0	0
4	1	0	0	1	0
5	1	0	1	1	0
6	1	1	1	1	1
7	1	1	0	0	0

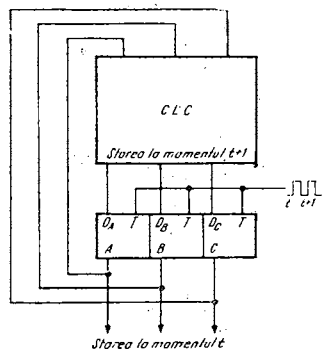


Fig. 3.102.

VIII. Tabelul de adevăr al CLC asociat este identic cu tabelul de tranziție datorită proprietății bistabilului  $D$  de a transmite la ieșire, ceea ce are la intrare.

Deci CLC va fi definit de trei funcții  $D_A$ ,  $D_B$ ,  $D_C$  de trei variabile  $A$ ,  $B$ ,  $C$ .

		A			
		0	1	0	1
B	{	0	1	0	1
		1	0	1	0
		C			

IX. Fiind vorba de implementarea mai multor funcții, soluțiile tipice sînt:

a) implementarea cu porți.

În acest caz se minimizează funcțiile cu diagrama Karnaugh-Veitch din figura 3.103.

Fig. 3.103.

Minimizările pentru f.c.n.d. sînt prezentate în figura 3.104.

Implementarea CLC se face cu  $1 \times 7400$  și  $1 \times 7410$ . Nu sînt necesare inversoare deoarece bistabilii  $D$  au accesibilă și ieșirea  $\bar{Q}$ . Deci întreg numărătorul se va realiza cu 4 CI ( $1/27474$  rămînînd neutilizat).

b) implementarea cu DCD și porți (figura 3.105).

În acest caz implementarea numărătorului va avea 5 CI, unul fiind MSI, deci e mai puțin convenabilă.

Sînt posibile și implementări ce nu se supun metodelor standard, ci presupun fantezia proiectantului combinată cu o oarecare experiență și capacitatea sa de a se orienta în fiecare caz particular. De exemplu pentru numărătorul nostru este posibil a se folosi proprietatea că, starea următoare se obține prin adăugarea lui 1, la starea precedentă. Deci,

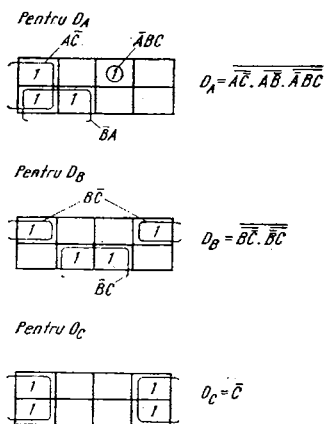


Fig. 3.104.

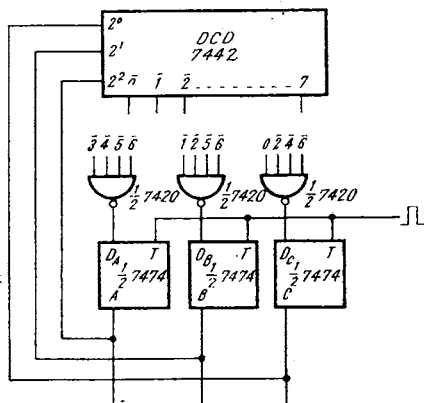


Fig. 3.105.



SLC poate fi un sumator cu 1. De asemenea registrul de bistabili poate fi un 7495 folosit pe mod paralel. Schema va conține astfel doar 2 circuite integrate (fig. 3.106).

**Problema 3.114.** Să se realizeze numărătorul din problema anterioară utilizând bistabili JKMS.

*Rezolvare*

Se urmează algoritmul prezentat în problema 3.112. Etapele I—IV se rezolvă similar ca în problema 3.113.

V. Registrul va avea 3 bistabili de tip JK.

VI. Particularizarea modelului Mealy este prezentată în figura 3.107.

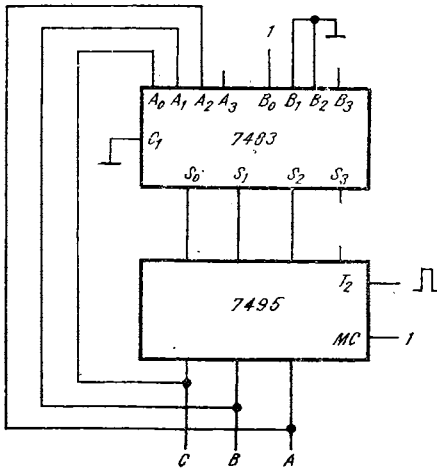


Fig. 3.106.

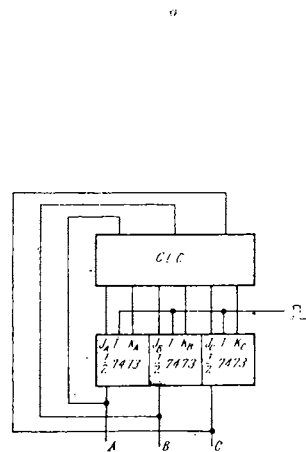


Fig. 3.107.

VII. Tabelul de tranziții binar este identic cu cel din problema anterioară.

VIII. Pentru stabilirea tabelului de adevăr al CLC trebuie ținut cont și de tabelul de tranziții al bistabilului JK. Cu ajutorul lui com-

$Q_t$	$J$	$K$	$Q_{t+1}$
0	0	×	0
0	1	×	1
1	×	1	0
1	×	0	1

pletăm coloanele  $J_A, K_A, J_B, K_B, J_C, K_C$  astfel încît să se obțină la ieșiri coloanele  $A, B, C$ .

Starea prezentă			Starea următoare								
$A$	$B$	$C$	$J_A$	$K_A$	$A$	$J_B$	$K_B$	$B$	$J_C$	$K_C$	$C$
0	0	0	0	×	0	0	×	0	1	×	1
0	0	1	0	×	0	1	×	1	×	1	0
0	1	0	0	×	0	×	0	1	1	×	1
0	1	1	1	×	1	×	1	0	×	1	0
1	0	0	×	0	1	0	×	0	1	×	1
1	0	1	×	0	1	1	×	1	×	1	0
1	1	0	×	0	1	×	0	1	1	×	1
1	1	1	×	1	0	×	1	0	×	1	0

IX. Implementarea se va face cu porți. Minimizînd cele 6 funcții rezultă :

$$J_A = K_A = BC$$

$$J_B = K_B = C$$

$$J_C = K_C = 1$$

Numărătorul va fi cel din figura 3.108.

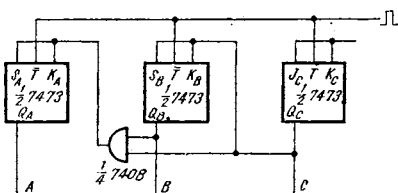


Fig. 3.108.

Comparînd cu varianta realizată cu bistabili D se observă că, deși CLC conține un număr dublu de funcții, implementarea rezultă mai simplă în acest caz.

Implementarea se poate face și pe bază de observație a tabelului de tranziție :  $A$  comută cînd  $B$  și  $C$  sînt 1,  $B$  comută cînd  $C$  este 1, iar  $C$  comută pe fiecare front negativ al impulsului de tact. La sisteme mai complicate o astfel de implementare nu este posibilă, rămîind valabilă metoda prezentată.

**Problema 3.115.** Să se proiecteze un numărător Johnson definit de graful de tranziții din figura 3.109.

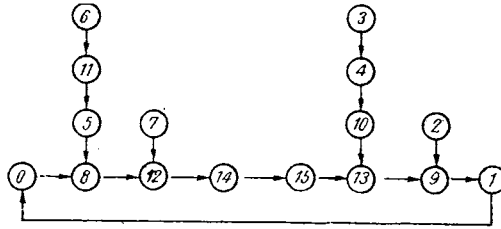


Fig. 3.109.

- a) cu bistabili D, 7474;
- b) cu bistabili JK, 7473;
- c) cu registre 7495.

Să se compare soluțiile ca număr de CI.

*Indicații*

Se urmărește algoritmul prezentat în problema 3.112.

I. Sistemul  $S$  este descris prin graf de tranziții.

II. Spațiul stărilor nu poate fi redus, deci  $S_e \equiv S$ .

III. Nu este cazul

IV.  $\{X\} \equiv \emptyset$ ;  $\{Y\} \equiv \{Q\}$ .

Mulțimea de stări,  $Q$ , identică cu cea de ieșire,  $Y$ , are 16 elemente, codificabile cu 4 biți.

V. Registrul  $R$  va avea 4 bistabili și se poate realiza cu :

- a)  $2 \times 7474$ ; b)  $2 \times 7473$ ; c)  $1 \times 7495$ .

VI. Modelul sistemului în cele trei cazuri este prezentat în figura 3.110.

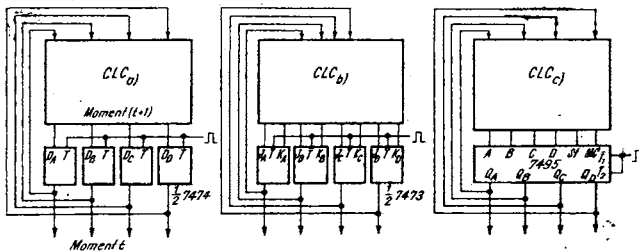


Fig. 3.110.

VII. Tabelul de tranziții exprimat binar se construiește notînd în stînga, ordonat, stările la momentul  $t$ , iar în dreapta pe cele următoare, conform grafului.

	$Q_A^t$	$Q_B^t$	$Q_C^t$	$Q_D^t$	momentul $t+1$			
					$Q_A$	$Q_B$	$Q_C$	$Q_D$
0	0	0	0	0	1	0	0	0
1	0	0	0	1	0	0	0	0
2	0	0	1	0	1	0	0	1
3	0	0	1	1	0	1	0	0
4	0	1	0	0	1	0	1	0
5	0	1	0	1	0	0	0	0
6	0	1	1	0	1	0	1	1
7	0	1	1	1	1	1	0	0
8	1	0	0	0	1	1	0	0
9	1	0	0	1	0	0	0	1
10	1	0	1	0	1	1	0	1
11	1	0	1	1	0	1	0	1
12	1	1	0	0	1	1	1	0
13	1	1	0	1	1	0	0	1
14	1	1	1	0	1	1	1	1
15	1	1	1	1	1	1	0	1

VIII. a) Tabelul de adevăr al  $CLC_a$  este identic cu cel de tranziții binare. Trebuie realizate 4 funcții de 4 variabile :

$$D_A = Q_A^{t+1}(Q_A^t, Q_B^t, Q_C^t, Q_D^t)$$

$$D_B = Q_B^{t+1}(Q_A^t, Q_B^t, Q_C^t, Q_D^t)$$

$$D_C = Q_C^{t+1}(Q_A^t, Q_B^t, Q_C^t, Q_D^t)$$

$$D_D = Q_D^{t+1}(Q_A^t, Q_B^t, Q_C^t, Q_D^t)$$

b) Se ține cont și de tabelul de tranziții exprimat binar al bistabililor JK (vezi problema anterioară). Rezultă următorul tabel de adevăr al  $CLC_b$  :

	$Q_A^t$	$Q_B^t$	$Q_C^t$	$Q_D^t$	$J_A$		$J_B$		$J_C$		$J_D$	
					$K_A$	$K_B$	$K_C$	$K_D$				
0	0	0	0	0	1	×	0	×	0	×	0	×
1	0	0	0	1	0	×	0	×	0	×	×	1
2	0	0	1	0	1	×	0	×	×	1	1	×
3	0	0	1	1	0	×	1	×	×	1	×	1
4	0	1	0	0	1	×	×	1	1	×	0	×

$Q_A^t$	$Q_B^t$	$Q_C^t$	$Q_D^t$	$J_A$	$K_A$	$J_B$	$K_B$	$J_C$	$K_C$	$J_D$	$K_D$
5	0	1	0	1	×	×	1	0	×	×	1
6	0	1	1	0	1	×	×	1	×	0	1
7	0	1	1	1	1	×	×	0	×	1	×
8	1	0	0	0	×	0	1	×	0	×	×
9	1	0	0	1	×	1	0	×	0	×	×
10	1	0	1	0	×	0	1	×	×	1	1
11	1	0	1	1	×	1	1	×	×	1	×
12	1	1	0	0	×	0	×	0	1	×	0
13	1	1	0	1	×	0	×	1	0	×	×
14	1	1	1	0	×	0	×	0	×	0	1
15	1	1	1	1	×	0	×	0	×	1	×

c) Se completează tabelul de tranziții al sistemului cu ajutorul următoarelor tabele ce descriu comportarea registrului 7495.

MC	SI	A	$Q_A^{t+1}$	$Q_A^t$	MC	$B_B$	$Q_B^{t+1}$	$Q_B^t$	MC	C	$Q_C^{t+1}$	$Q_C^t$	MC	D	$Q_D^{t+1}$
0	0	×	0	0	0	×	0	0	0	×	0	0	0	×	0
0	1	×	1	1	0	×	1	1	0	×	1	1	0	×	1
1	×	0	0	×	1	0	0	×	1	0	0	×	1	0	0
1	×	1	1	×	1	1	1	×	1	1	1	×	1	1	1

Cu aceasta, tabelul de adevăr al  $CLC_0$  este următorul :

Nr. stare	$Q_A^t$	$Q_B^t$	$Q_C^t$	$Q_D^t$	MC	SI	A	$Q_A^{t+1}$	B	$Q_B^{t+1}$	C	$Q_C^{t+1}$	D	$Q_D^{t+1}$
0	0	0	0	0	0	1	×	1	×	0	×	0	×	0
1	0	0	0	1	1	×	0	0	0	0	0	0	0	0
2	0	0	1	0	0	1	×	1	×	0	×	0	×	1
3	0	0	1	1	1	×	0	0	1	1	0	0	0	0
4	0	1	0	0	0	1	×	1	×	0	×	1	×	0
5	0	1	0	1	1	×	1	1	0	0	0	0	0	0
6	0	1	1	0	0	1	×	1	×	0	×	1	×	1
7	0	1	1	1	1	×	1	1	1	1	0	0	0	0
8	1	0	0	0	0	1	×	1	×	1	×	0	×	0
9	1	0	0	1	1	×	0	0	0	0	0	0	1	1
10	1	0	1	0	0	1	×	1	×	1	×	0	×	1
11	1	0	1	1	1	×	0	0	1	1	0	0	1	1
12	1	1	0	0	0	1	×	1	×	1	×	1	×	0
13	1	1	0	1	1	×	1	1	0	0	0	0	1	1
14	1	1	1	0	0	1	×	1	×	1	×	1	×	1
15	1	1	1	1	1	×	1	1	1	1	0	0	1	1

Pentru determinarea lui, s-a considerat tabelul de tranziții binare  $Q^{t+1}(Q^t)$  și s-au completat coloanele MC, SI, A, B, C, D conform cu tabelele de tranziții ale registrului 7495.

Se începe cu completarea coloanei MC comparînd  $Q_A^t, Q_B^t, Q_C^t$  cu  $Q_B^{t+1}, Q_C^{t+1}, Q_D^{t+1}$ :

— dacă cele 2 grupuri de biți sînt identice, înseamnă că MC poate fi la alegere, 0 sau 1, adică  $Q_B^{t+1}, Q_C^{t+1}, Q_D^{t+1}$  se pot obține fie prin deplasarea serie a  $Q_A^t, Q_B^t, Q_C^t$ , fie prin înscriere în paralel.

— dacă cele 2 grupuri nu sînt identice, cel de la momentul  $t + 1$  nu poate fi obținut decît prin înscriere în paralel a datelor B, C, D (pentru stările 3, 5, 7, 9, 13, 15).

Nu este indicat ca MC să fie ales tot timpul egal cu 1 deoarece registrul ar funcționa doar ca 4 bistabili D, celelalte facilități ale deplasării serie rămînd neutilizate. În majoritatea cazurilor în care este posibilă alegerea lui MC se va opta pentru  $MC = 0$ , cu excepția acelor situații în care  $MC = 1$  conduce la o minimizare mai importantă a funcțiilor logice. Astfel pentru stările 1 și 11 s-a stabilit  $MC = 1$  pentru a obține simplu funcția  $MC = D$ .

IX. Implementarea se face cu metodele cunoscute. La punctul c) se obține numărul minim de CI. Minimizarea funcțiilor logice combinaționale este dată în figura 3.111. Se utilizează termenii redundanți pentru a obține simplificarea funcțiilor.

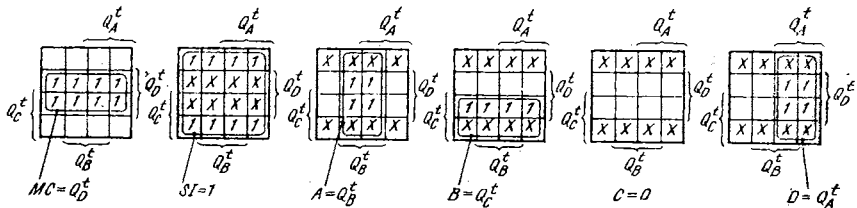


Fig. 3.111.

Implementarea numărătorului Johnson cu autoamorsare utilizînd registrul 7495 este prezentată în figura 3.112. Se observă că CLC<sub>0</sub> s-a redus doar la conexiuni între intrările și ieșirile registrului; nu se adaugă nici o poartă logică.

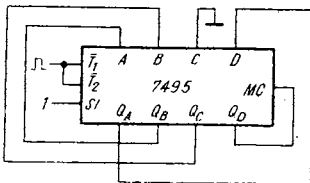


Fig. 3.112.

#### Observație

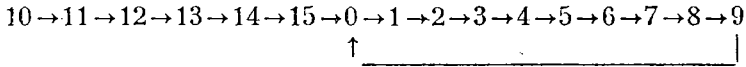
Implementarea SLSS cu 7495, cînd este posibilă utilizarea acestuia în regim de deplasare serie ( $MC = 0$ ), este foarte economică datorită complexității interne a registrului.

**Problema 3.116.** Să se proiecteze un numărător sincron pînă la 15 :

- a) cu bistabili D ;
- b) cu registru de deplasare ;
- c) cu bistabili JK.

Să se compare soluțiile ca număr de circuite integrate.

**Problema 3.117.** Să se proiecteze un numărător zecimal cu graful următor :



avînd proprietatea că pentru orice stare neapartînînd ciclului, stare ce poate apare în momentul inițial sau ca urmare a unor perturbații în funcționare, se reintră în ciclul de numărare.

**Problema 3.118.** Să se reia problema anterioară în cazul cînd auto-corecția și inițializarea se fac într-un singur tact, conform grafului din figura 3.113.

Să se compare ca număr de CI cu soluția din problema anterioară.

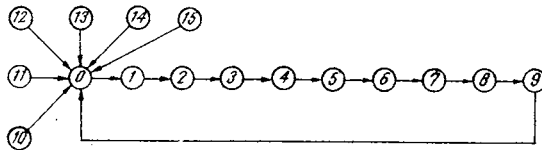


Fig. 3.113.

**Problema 3.119.** Să se proiecteze un sistem logic secvențial cu proprietățile :

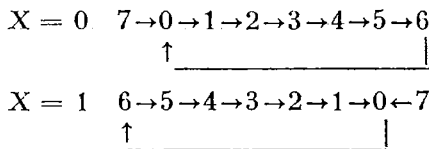
- pentru  $X = 0$  generează la ieșire numerele de la 0 la 6 în ordine crescătoare ;
- pentru  $X = 1$  generează numerele de la 6 la 0 (numără invers) ;
- ieșirea  $Y$  este 1 pentru fiecare configurație care este multiplu de 3.

Se vor utiliza 1 DCD, bistabili de tip D și porți.

*Indicație*

Se urmează algoritmul din problema 3.112.

I—III. Graful de tranziție este



Starea a 7-a care nu aparține ciclurilor, într-o proiectare cu autoinițializare și autocorecție trebuie să permită reintrarea în ciclu.

IV. Sistemul are :

— două elemente în mulțimea de intrare, codificate cu 1 bit ( $X = 0, X = 1$ );

— 7 elemente (8 în cazul autocorecției, în spațiul stărilor, codificabile cu 3 biți);

— spațiul de ieșire îl include pe cel de stare conținând în plus ieșirea  $Y$  de semnalizare a multiplilor lui 3.

V. Registrul va conține 4 bistabili : 3 pentru memorarea numerelor între 0 și 6 și unul pentru ieșirea  $Y$ . Acesta din urmă este necesar doar dacă ne interesează ca să eliminăm semnalele parazite datorate hazardului combinațional.

VI. Modelul Mealy al sistemului este dat în figura 3.114.

VIII. Tabelul de adevăr al CLC este

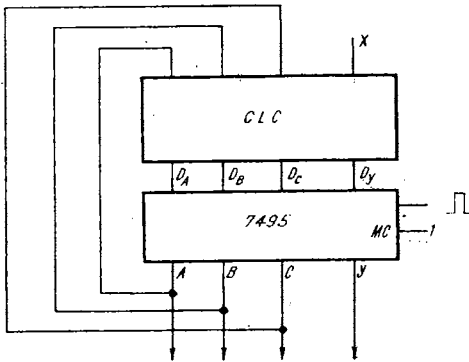
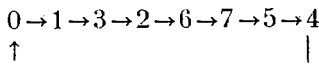


Fig. 3.114.

	Starea prezentă				Starea ulterioară			
	X	A	B	C	$D_A$	$D_B$	$D_C$	$D_Y$
0	0	0	0	0	0	0	1	0
1	0	0	0	1	0	1	0	0
2	0	0	1	0	0	1	1	1
3	0	0	1	1	1	0	0	0
4	0	1	0	0	1	0	1	0
5	0	1	0	1	1	1	0	1
6	0	1	1	0	0	0	0	0
7	0	1	1	1	0	0	0	0
8	1	0	0	0	1	1	0	1
9	1	0	0	1	0	0	0	0
10	1	0	1	0	0	0	1	0
11	1	0	1	1	0	1	0	0
12	1	1	0	0	0	1	1	1
13	1	1	0	1	1	0	0	0
14	1	1	1	0	1	0	1	0
15	1	1	1	1	0	0	0	0

IX. Trebuie implementat un CLC cu 4 funcții cu 4 intrări. Circuitele vor fi  $1 \times 74154, 3 \times 7430, 1 \times 7420, 1 \times 7495$ . Rămâne neutilizat un ȘI NU cu 4 intrări.

**Problema 3.120.** Să se proiecteze un numărător sincron cu bistabili D și JK, având tranzițiile definite de :





**Problema 3.121.** a) Să se deducă graful de tranziții pentru numărătorul din figura 3.67.

b) Să se reprojeteze numărătorul cu bistabili JK.

*Soluție*

a) Vezi figura 3.115.

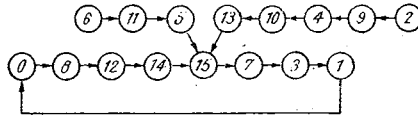
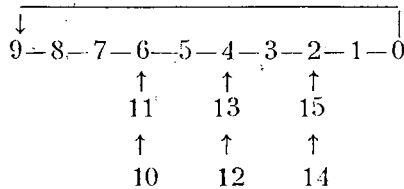


Fig. 3.115.

Numărătorul Johnson se caracterizează prin aceea că, de la o tranziție la alta, în ciclu nu variază decât un singur bit. Acest fapt permite o decodificare, fără semnale parazite.

**Problema 3.122.** Să se proiecteze sistemul definit de graful următor



a) cu bistabili D ;

b) cu registru 7495 ;

c) cu bistabili JK ;

d) cu numărător 74193.

**Problema 3.123.** Proiectați cu minimum de componente sistemul secvențial sincron care să realizeze următoarele :

a) să furnizeze semnal la ieșire la apariția stărilor pare cuprinse între 0 și 8.

b) să furnizeze semnal la ieșire la apariția stărilor impare cuprinse între 8 și 0.

c) să semnalizeze apariția stărilor prime din intervalul 0 — 15.

d) să semnalizeze multiplii de 4 din intervalul 15—0.

*Indicație*

Pentru reducerea spațiilor X și Y se codifică intrările :  $A = 00$  ;  $B = 01$  ;  $C = 10$  ;  $D = 11$ . Dacă ieșirile nu sînt codificate, pentru a

evita semnalele parazite sistemul va avea un registru cu 8 bi-stabili.

Graful sistemului este dat în figura 3.116.

Configurația particulară realizată cu registre 7495 utilizate ca bista-bili D va fi cea din figura 3.117.

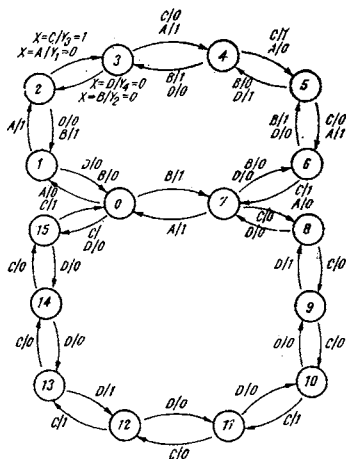


Fig. 3.116.

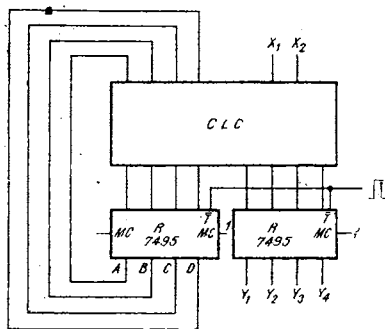


Fig. 3.117.

Trebuie deci implementat un circuit combinațional cu 8 funcții de 6 variabile.

Încercați și soluția cu numărarea prin comanda unui sumator scăzător.

Dacă se codifică și ieșirile vom obține un CLC cu 6 funcții de 6 variabile. Se reduce implementarea în acest caz?

**Problemă 3.124.** Să se implementeze sistemul definit de tabelul de tranziții următor :

Q \ X	0	1
	A	A, 0
B	B, 0	C, 0
C	C, 0	D, 1
D	D, 1	A, 0

*Indicație*

Tabelul de tranziții exprimat binar va fi:

Intrări			Ieșiri		
X	Y <sub>1</sub>	Y <sub>2</sub>	Y <sub>1</sub>	Y <sub>2</sub>	Z
0	0	0 (A)	0	0 (A)	0
	0	1 (B)	0	1 (B)	0
	1	1 (C)	1	1 (C)	0
	1	0 (D)	1	0 (D)	1
1	0	0 (A)	0	1 (B)	0
	0	1 (B)	1	1 (C)	0
	1	1 (C)	1	0 (D)	1
	1	0 (D)	0	0 (A)	0

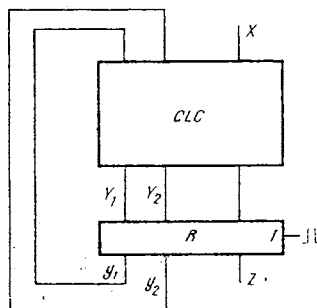


Fig. 3.118.

Sistemul realizat cu bistabili D va fi modelat ca în figura 3.118.

**Problema 3.125.** Să se proiecteze sistemul logice definit de tabelul de tranziții

Q \ X	0	1	2
1	2,1	2,0	5,0
2	1,0	4,1	4,1
3	2,1	2,0	5,0
4	3,0	2,1	2,1
5	6,1	4,0	3,0
6	8,0	9,1	6,1
7	6,1	2,0	8,0
8	4,1	4,0	7,0
9	7,0	9,1	7,1

*Indicație*

Se urmează algoritmul prezentat în problema 3.112.

I. Sistemul este descris prin tabel de tranziții.

II. Spațiul stărilor sistemului poate fi redus (vezi problema 1.11).

III. Tabelul de tranziții al sistemului redus  $S_e$ , este dat în aceeași problemă.

IV. Cele 5 stări rezultate ale sistemului redus se codifică cu 3 biți. Intrările în număr de 3 se codifică cu 2 biți. Sistemul are ieșirea  $Y = 0$  sau 1.

V. Registrul  $R$  va avea 4 biți, 3 pentru stări și unul pentru ieșire.

VI. Schema echivalentă particulară a sistemului redus este dată în figura 3.119.

**Problema 3.126.** Să se implementeze sistemul secvențial definit de organigrama din figura 3.120, unde  $S_K$  ( $K = 1 \div 7$ ) sînt stări, iar  $X$  este o condiționare a funcționării.

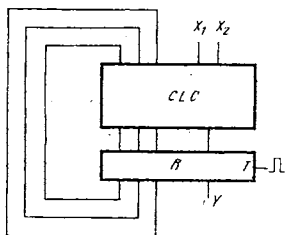


Fig. 3.119.

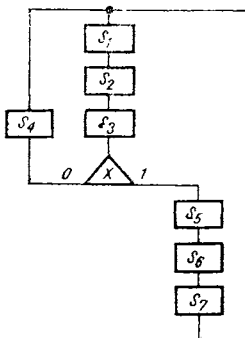


Fig. 3.120.

*Indicație*

Sistemul avînd 7 stări, acestea se pot codifica cu 3 biți  $A$ ,  $B$ ,  $C$ . Asocierea este arbitrară (de ex.:  $S_1-000$ ,  $S_2-001$ ,  $S_3-010$ ,  $S_4-011$ ,  $S_5-100$ ,  $S_6-101$ ,  $S_7-110$ ) și e indicat să se aleagă astfel încît să se obțină o implementare minimă a sistemului.

În cazul organigramei schema de realizare a sistemului secvențial va fi cea din figura 3.121.

Tabelul de tranziție al sistemului va fi

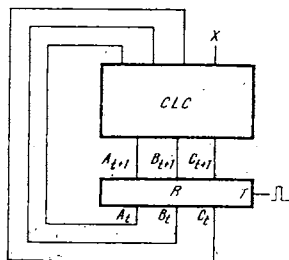


Fig. 3.121.

$A_t$	$B_t$	$C_t$	$X$	$A_{t+1}$	$B_{t+1}$	$C_{t+1}$
0	0	0	0	0	0	1
0	0	0	1	0	0	1
0	0	1	0	0	1	0
0	0	1	1	0	1	0
0	1	0	0	0	1	1
0	1	0	1	1	0	0
0	1	1	0	1	0	0
0	1	1	1	1	0	0
1	0	0	0	1	0	1
1	0	1	0	1	1	0
1	0	1	1	1	1	0
1	1	0	0			
1	1	0	1			
1	1	1	0			
1	1	1	1			

} stări  
neutilizate

**Problema 3.127.** Să se implementeze cu bistabili T un numărător binar sincron pînă la 1024 utilizînd pentru CLC :

- a) PLA
- b) ROM
- c) porți ȘI.

Să se compare soluțiile ca viteză, număr de CI utilizate și complexitate a acestora.

*Indicații*

a) Schema automatului Mealy care realizează numărarea pînă la 1024 este cea din figura 3.122.

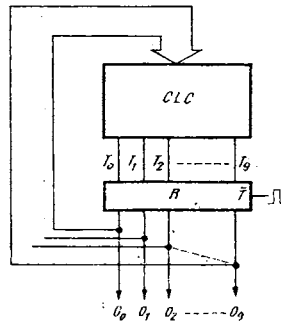


Fig. 3.122.

Proiectăm  $R$  cu bistabili JK, folosiți ca bistabili T. Funcțiile logice ale CLC vor fi :

$$T_0 = P_1$$

$$T_1 = P_1 + P_3 + P_5 + P_7 + P_9 + \dots$$

$T_1$  va conține 512 termeni canonici. Minimizarea acestei funcții cu metode clasice este dificilă și se poate efectua cu ajutorul calculatorului. Se poate însă aplica observația că bistabilul corespunzător rangului  $2^i$  comută cînd bistabilul anterior este pe 1. Deci :

$$T_1 = Q_0$$

De asemenea,  $T_2$  comută cînd bistabilii anteriori sînt pe 1. Deci :

$$T_2 = Q_0 \cdot Q_1$$

În mod similar :

$$T_3 = Q_0 \cdot Q_1 \cdot Q_2$$

$$T_n = Q_0 \cdot Q_1 \cdot \dots \cdot Q_{n-1}$$

PLA-ul va fi programat ca în figura 3.123

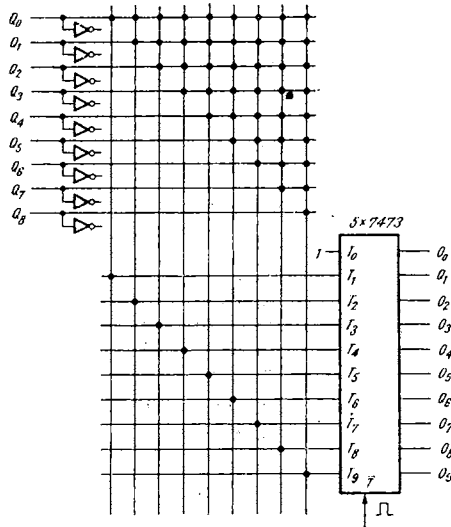


Fig. 3.123.

Deci PLA-ul cu care se realizează sistemul conține o matrice de ȘI-uri și una de SAU-uri de  $9 \times 9$  puncte.

ROM-ul echivalent acestui PLA are dimensiunea  $512 (2^9) \times 9$  biți.

b) Pentru a implementa același sistem cu ROM, e necesară decodificarea tuturor termenilor canonici deci adresare pe 10 biți, ieșirea fiind pe 9 biți. Deci implementarea cu ROM presupune o capacitate de  $1024 \times 9$  biți, de două ori mai mare ca la implementarea cu PLA.

c) Ținând cont de minimizare sistemul se poate implementa simplu cu porți (vezi figura 3.124) ȘI cu 2 intrări.

Dezavantajul schemei constă în reducerea frecvenței de numărare datorită întârzierilor pe circuite (vezi problema 4.42).

O soluție de implementare cu porți neînseriate se realizează cu ajutorul unui CLC cu 6 CI.

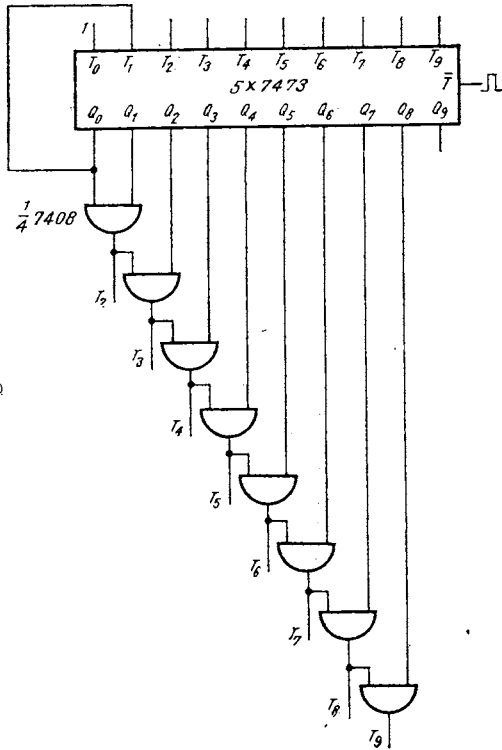


Fig. 3.124.

Costul ridicat al unui circuit LSI nu justifică înlocuirea a 6 CI de porți cu PLA sau ROM.

**Problema 3.128.** Proiectați cu PLA un numărător sincron modulo 10 numărând în cod exces 3.

**Problema 3.129.** Să se implementeze sistemul din problema 3.123.

a) cu ROM și registre de deplasare ;

b) cu PLA ;

c) să se calculeze dimensiunea ROM-ului echivalent cu PLA-ul.

*Indicație*

a) Dacă atât intrările cât și ieșirile sînt necodificate este necesar un ROM de  $2^8 = 256$  cuvinte de 8 biți.

Codificând intrările, dimensiunea ROM-ului se reduce la  $2^6 = 64$  cuvinte de 8 biți.

O codificare a ieșirilor, deși reduce numărul de biți ce trebuie programați în ROM nu este deosebit de utilă deoarece nu se reduce efectiv numărul de CI (ROM-urile sînt organizate doar în cuvinte de  $2^n$  biți și 6 nu este putere a lui 2).

**Problema 3.130.** Să se implementeze sistemul definit de organigrama din figura 3.125, care semnalizează cu 1 la ieșire apariția stărilor  $A, D, F, H$ :

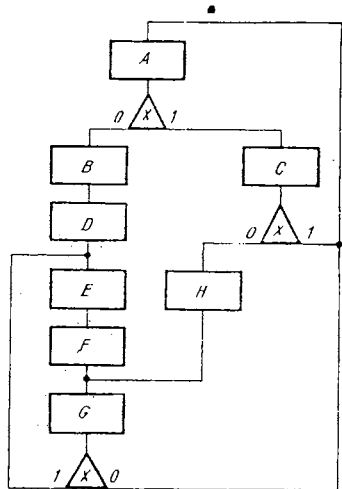


Fig. 3.125.

- a) cu registru de bistabili D ;
- b) cu bistabili JK ;
- c) cu registru 7495.

Pentru ce codificare a stărilor rezultă implementările cu număr minim de CI?

**Problema 3.131.** Să se proiecteze sistemul definit de tabelul de tranziții din problema 1.3.

*Indicație*

Sistemul are 1 intrare, X, o ieșire, Y, și stările se codifică cu 2 biți.



**Problema 3.132.** Să se implementeze sistemul descris de organigrama din figura 1.15:

- a) cu bistabili de tip D.
- b) cu bistabili de tip JK,
- c) cu registre 7495.

**Problema 3.133.** Să se implementeze sistemul logic secvențial definit de graful din figura 1.20.

- a) cu bistabili de tip D, decodificator și porți;
- b) cu bistabili de tip JK și multiplexoare;
- c) cu registru 7495 și porți.

**Problema 3.134.** Să se proiecteze sistemul logic secvențial definit în problema 1.7.

*Indicație*

Sistemul are 2 biți de intrare ( $X'$ ,  $X''$ ) și 3 biți de ieșire-stare ( $q_1$ ,  $q_2$ ,  $q_3$ ), deci CLC, în cazul proiectării cu bistabili D trebuie să realizeze 3 funcții cu 5 intrări.

**Problema 3.135.** Să se proiecteze sistemul logic definit de formele de undă din figura 1.28 cu una din metodele cunoscute.

*Indicație*

Pentru graful din figura 1.30, cu codificarea A-00, B-01, C-10, D-11, rezultă următorul tabel de tranziții exprimat binar:

X	stare ( $t$ )		stare ( $t+1$ )		Y
0	0	0(A)	0	1	1
0	0	1(B)	1	0	0
0	1	0(C)	0	0	0
0	1	1(D)			
1	0	0(A)	0	1	0
1	0	1(B)	1	0	0
1	1	0(C)	1	1	1
1	1	1(D)	0	0	0

Din starea D, pentru  $X = 0$  nu are loc nici o tranziție. Se poate considera că sistemul rămâne în aceeași stare.

Deoarece sistemul trebuie să comute pe front negativ, implementarea se poate face cu bistabili 7473 sau cu registre 7495.

**Problema 3.136.** Să se proiecteze SLSS definit prin formele de undă din figura 1.32.

### Indicație

Deoarece sistemul comută cu front pozitiv, registrul va conține bistabili de tip D, 7474.

**Problema 3.137.** Să se implementeze un numărător binar sincron înainte și înapoi cu o capacitate de 512 numere, utilizând memorii ROM și bistabili de tip D. Ieșirea semnalizează apariția numerelor prime în intervalul de numărare.

### Indicație

Sistemul are o intrare,  $X$ , de selecție a sensului de numărare, 512 stări (codificabile cu 9 biți) și o ieșire  $Y$ . Memoria ROM are o capacitate de  $1024 \times 10$  biți.

**Problema 3.138.** Să se proiecteze sistemul definit de organigrama din figura 3.126.

### Indicație

Se observă că organigrama presupune testarea înlănțuită a două condiții de intrare  $X'$  și  $X''$ . Această testare se poate realiza în două moduri :

a) Testarea succesivă a condițiilor de intrare : se intercalează o stare neapartinând sistemului (de exemplu  $G$ ) în ramura  $X' - X''$ . Operând această modificare și codificând stările, organigrama se transformă ca în figura 3.127.

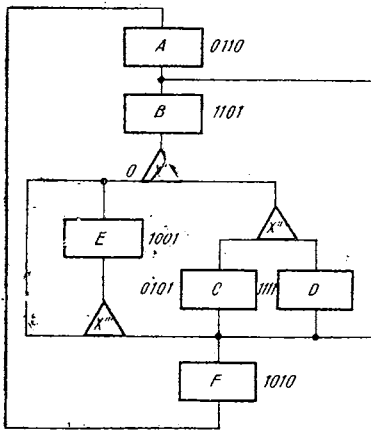


Fig. 3.126.

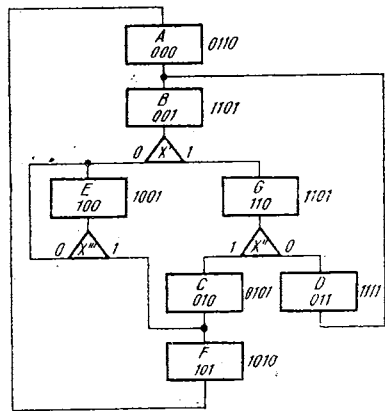


Fig. 3.127.

Implementarea se poate realiza ca în figura 3.128: testarea condițiilor de intrare se obține cu ajutorul multiplexorului. În funcție de biții cei mai puțin semnificativi ai cuvântului de stare, MUX transmite la ieșire valoarea  $X'$ ,  $X''$  sau  $X'''$ , conform tabelului următor :

$Q_1$	$Q_0$	$W$
0	0	$X'''$
0	1	$X'$
1	0	$X''$

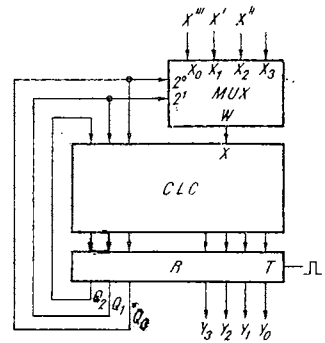


Fig. 3.128.

Evident, codurile stărilor  $B$ ,  $E$  și  $G$  care condiționează testarea trebuie alese astfel încât să conțină perechi de biți distincte pentru comanda MUX.

Prin utilizarea MUX, CLC va avea o intrare  $X = X'$ ,  $X''$  și  $X'''$  în funcție de starea respectivă. Tabelul de adevăr al CLC este următorul :

$t$				$t + 1$						
$X$	$Q_2$	$Q_1$	$Q_0$	$Q_2$	$Q_1$	$Q_0$	$Y_3$	$Y_2$	$Y_1$	$Y_0$
0	0	0	0(A)	0	0	1(B)	1	1	0	1
0	0	0	1(B)	1	0	0(E)	1	0	0	1
0	0	1	0(C)	1	0	1(F)	1	0	1	0
0	0	1	1(D)	0	0	1(B)	1	1	0	1
0	1	0	0(E)	1	0	0(E)	1	0	0	1
0	1	0	1(F)	0	0	0(A)	0	1	1	0
0	1	1	0(G)	0	1	1(D)	1	1	1	1
0	1	1	1	×	×	×	×	×	×	×
1	0	0	0(A)	0	0	1(B)	1	1	0	1
1	0	0	1(B)	1	1	0(G)	1	1	0	1
1	0	1	0(C)	1	0	1(F)	1	0	1	0
1	0	1	1(D)	0	0	1(B)	1	1	0	1
1	1	0	0(E)	1	0	1(F)	1	0	1	0
1	1	0	1(F)	0	0	0(A)	0	1	1	0
1	1	1	0(G)	0	1	0(C)	0	1	0	1
1	1	1	1	×	×	×	×	×	×	×

Starea a 7-a nu aparține sistemului și poate fi utilizată ca stare redundanță pentru obținerea minimizării. S-a considerat că în starea  $G$  se menține același vector de ieșire ca în starea anterioară,  $B$ .

b) Testarea simultană a condițiilor de intrare : se realizează un circuit logic combinațional CLC' astfel încît să se implementeze organigrama din figura 3.129.

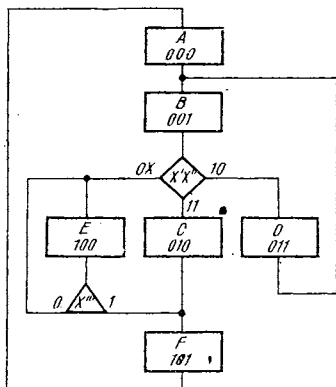


Fig. 3.129.

Tabelul de adevăr al acestui circuit este următorul (în cazul utilizării bistabililor  $D$ ) :

$X'$	$X''$	$X'''$	$Q_2$	$Q_1$	$Q_0$	$Q_2$	$Q_1$	$Q_0$
×	×	×	0	0	0	0	0	1
0	×	×	0	0	1	1	0	1
1	0	×	0	0	1	0	1	1
1	1	×	0	0	1	0	1	0
×	×	×	0	1	1	0	0	1
×	×	0	1	0	0	1	0	0
×	×	1	1	0	0	1	0	1
×	×	×	1	0	1	0	0	0
×	×	×	1	1	0	×	×	×
×	×	×	1	1	1	×	×	×

**Problema 3.139.** Să se proiecteze sistemul definit de organigrama din figura 3.130, eliminându-se fenomenele de hazard la ieșire.

*Indicație*

Pentru testarea condițiilor se va utiliza un MUX de 3 căi.

Sistemul va avea 6 intrări, 13 stări și 4 ieșiri.

Pentru eliminarea hazardului, ieșirile trebuie trecute printr-un registru de bistabili care realizează sincronizarea (vezi problema 1.2).

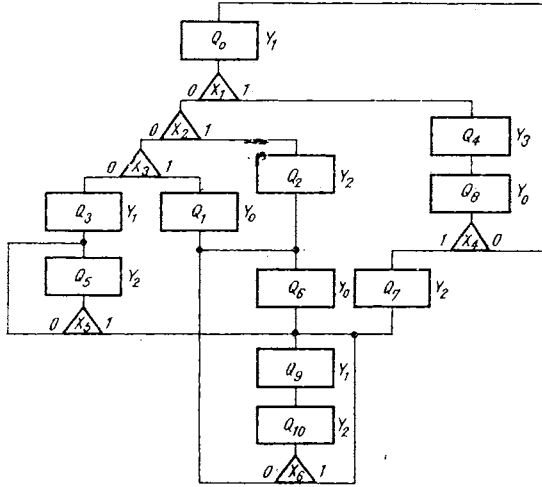


Fig. 3.130.

**Problema 3.140.** Se consideră un SLSS definit de graful din figura 3.131 și având intrările  $A, B, C$ .

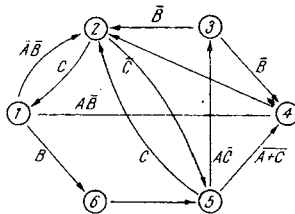


Fig. 3.131.

- Să se construiască organigrama sistemului.
- Care sînt condițiile ce trebuie îndeplinite de variabilele de intrare definind tranzițiile pentru ca sistemul să fie fizic realizabil?
- Să se implementeze sistemul eliminîndu-se hazardul combinatorial de la ieșiri.

*Indicație*

a) Pentru a construi organigrama se pornește dintr-una din stări (de ex. 1) și se testează pe rînd condițiile de intrare  $X_1 = A\bar{B}$ , apoi  $X_2 = B$ . Dacă nu sînt îndepliniți nici  $X_1$ , nici  $X_2$  înseamnă că se îndeplinește  $X_3 = \bar{A}\bar{B}$  și sistemul trece în starea 2 ș.a.m.d.

b) Pentru ca sistemul să fie fizic realizabil trebuie ca :

— suma condițiilor de intrare ce determină tranziția din fiecare nod să fie 1. Într-adevăr pentru nodul 1 avem :  $\bar{A}\bar{B} + A\bar{B} + B = 1$ , pentru nodul 2,  $C + \bar{C} = 1$ , pentru 3,  $B + \bar{B} = 1$ . Tranziția din nodurile 4 și 6 se face necondiționat, iar pentru nodul 5 avem  $\bar{A} + \bar{C} + A\bar{C} + C = \bar{A}\bar{C} + A\bar{C} + C = 1$  ;

— suma parțială a condițiilor de intrare ce definesc tranzițiile din fiecare nod să fie diferită de 1. De exemplu, pentru nodul 1 avem :

$$\bar{A}\bar{B} + A\bar{B} \neq 1$$

$$B + \bar{A}\bar{B} \neq 1$$

$$B + A\bar{B} \neq 1$$

Verificând toate aceste condiții rezultă că sistemul este fizic realizabil.

c) Sistemul are 3 intrări dar trebuie testată și apariția combinațiilor acestora (de exemplu  $A\bar{B}$ ). Trebuie să fie alese pentru testare condițiile ce conduc la utilizarea numărului minim de circuite.

Organigrama este de tipul celei din problema 3.138, cu mai multe testări înlănțuite.

**Prezentarea conținutului**

În capitolele anterioare s-au proiectat sisteme considerând doar relațiile logice între variabilele de intrare, ieșire, stare și funcțiile logice, tabelele de tranziții ale circuitelor utilizate în implementare. Astfel de sisteme, realizate practic, au o probabilitate mică de funcționare corectă, demonstrând insuficiențele unei proiectări exclusiv logice. Aceasta trebuie completată luând în considerare și parametrii tehnologici ai circuitelor (timp de propagare, „fan-out“ ș.a.). Capitolul 4 are drept scop prezentarea particularităților circuitelor în funcție de tehnologia de realizare (bipolară sau unipolară) și a implicațiilor acestora asupra implementării.

Problemele 4.1—4.27 se referă la structurile interne și parametrii porților TTL standard, rapide (HTTL), de putere mică (TTLLP) și Schottky (STTL).

Se compară aceste familii de circuite ca timp de propagare, putere absorbită, „fan-out“.

Se prezintă avantajele și dezavantajele diverselor etaje de ieșire ale TTL-urilor, ca și aspecte legate de ieșirile „open collector“ (cu colector în gol) și „three state“ (cu trei stări).

Problemele 4.28—4.39 prezintă scheme realizate cu porți, a căror funcționare se bazează în principal pe întârzierile în răspunsul circuitelor datorate timpilor de propagare: oscilatoare, monostabili, detectoare de fronturi.

O atenție deosebită este acordată influenței timpilor de propagare asupra funcționării sistemelor și anume :

1. limitarea frecvenței de funcționare corectă a sistemelor ;
2. apariția fenomenelor de hazard (semnale parazite la ieșiri).

De aceste aspecte ca și de cele legate de utilizarea circuitelor sub limitele maxim admisibile ale „fan-out“-ului, ale pragurilor de curenți, tensiuni, ale puterii disipate, trebuie să se țină cont într-o proiectare completînd-o iterativ pe cea strict logică.

Problemele 4.49—4.56 familiarizează cititorul cu cîteva aspecte ale realizării și utilizării circuitelor ECL (Emitter Coupled Logic) : sînt prezentate structurile interne ale unor porți uzuale precum și criteriile de dimensionare a rezistențelor din emitoare.

În a doua parte a capitolului sînt descrise circuite MOS. Se subliniază particularitățile inversoarelor statice MOS cu canal p cu îmbogățire (PELT, PELS) sau sărăcire de purtători (PDLT). Se indică modul cum se pot obține porțile logice în această tehnologie.

Sînt descrise inversoarele MOS dinamice cu 3 și 4 tranzistoare (problemele 4.62 și 4.65) și unele tipuri de porți dinamice.

Registrele de deplasare MOS, statice și dinamice (problemele 4.68—4.74) sînt analizate atît ca scheme interne cît și ca eficiență a funcționării dinamice, ca putere disipată și ca limitare a frecvenței de lucru. Se propun scheme pentru mărirea frecvenței de operare cu registrele de deplasare MOS și de cuplare a circuitelor MOSp cu cele TTL.

Un loc important în cadrul acestui capitol îl ocupă circuitele CMOS (problemele 4.75—4.90).

Este descris inversorul CMOS : schemă internă, timpi de propagare, consum de putere, static și dinamic, particularități ale etajului de intrare, fan-out, margine de zgomot.

Sînt prezentate scheme interne de circuite CMOS de bază.

Se analizează problemele la interfața de cuplare a circuitelor CMOS cu cele pMOS, nMOS, TTL.

Pentru aprofundarea implicațiilor pe care le au în proiectare parametri tehnologici se recomandă reconsiderarea unor probleme din capitolele anterioare ținînd cont de aspectele prezentate în capitolul 4, așa cum s-a procedat de exemplu în 4.42 și 4.48.

**Problema 4.1.** Să se definească termenii SSI, MSI, LSI și VLSI.

#### *Rezolvare*

SSI înseamnă integrare pe scară redusă (Small Scale Integration) și definește clasa CI conținînd mai puțin de 50 tranzistoare pe pastilă (Ex. : porți, bistabili).

MSI, integrare pe scară medie, este clasa CI cu 50—500 tranzistoare pe pastilă (Ex. : MUX, DMUX, numărătoare, registre).

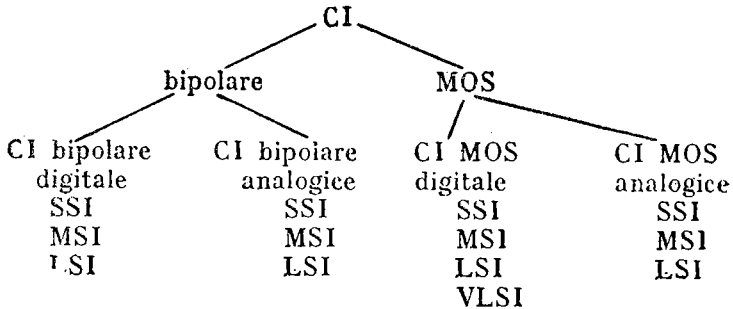


LSI, integrare pe scară largă este clasa CI cu mai mult de 500 tranzistoare pe pastilă (Ex.: RAM, ROM, PLA).

VLSI, integrare pe scară foarte largă este clasa CI de cea mai mare complexitate, cu număr mare de tranzistoare pe pastilă (Ex.: microprocesoare, calculatoare într-o singură pastilă).

**Problema 4.2.** Să se definească tipurile de tehnologii ale CI monolitice.

*Soluție*



**Problema 4.3.** Cunoscând  $U_{IL} = 0,8 \text{ V}$ ,  $U_{IH} = 2 \text{ V}$ ,  $U_{OL} = 0,4 \text{ V}$ ,  $U_{OH} = 2,4 \text{ V}$  să se calculeze curenții și nivelele de tensiune din schema unei porți NAND TTL.

Se consideră  $0,7 \text{ V}$  tensiunea pe joncțiunile deschise.

*Soluție*

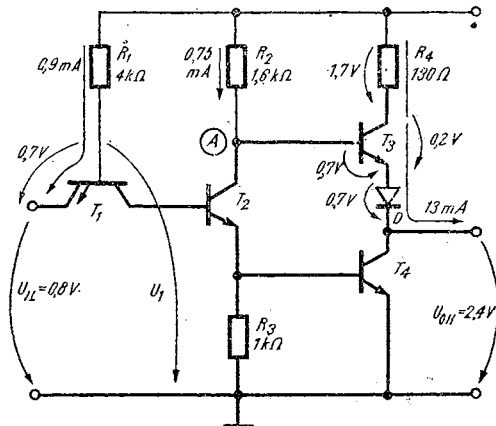


Fig. 4.1.

Aplicind la intrare  $U_{IL} = 0,8 \text{ V}$  (fig. 4.1), jonctiunea  $BE$  a tranzistorului  $T_1$  se deschide, pe  $R_1$  vor cdea  $3,5 \text{ V}$ , producind un curent de cca  $0,9 \text{ mA}$ . Deci tensiunea  $U_1$  este de  $1,4 \text{ V}$ . Inseamnă că jonctiunile  $BC$  și  $BE$  ale  $T_1$  și  $T_2$  nu se deschid (dacă s-ar deschide,  $U_1$  ar trebui să asigure  $2 \times 0,7 \text{ V}$  plus căderea de tensiune pe  $R_3$ ).

Tranzistorul  $T_2$  fiind blocat, întregul curent din sursa de  $5 \text{ V}$  trece prin  $R_2$  în baza lui  $T_3$ , deschizându-l. De asemenea  $D$  este deschisă.  $T_4$  este blocat deoarece prin  $R_3$  nu circulă curent.

Tensiunea în punctul  $A$  va fi de  $3,8 \text{ V}$ . Rezultă curentul prin  $R_4$  de  $0,75 \text{ mA}$ .

Presupunind tensiunea  $V_{CE} = 0,2 \text{ V}$  pentru  $T_3$  saturat, curentul prin  $R_4$ , care este egal cu cel furnizat în sarcină va fi de  $13 \text{ mA}$ .

Cînd se ridică potențialul emitorului la  $U_{IH}$  (fig. 4.2), jonctiunea  $BE$  a lui  $T_1$  se închide, curentul comutînd pe jonctiunea  $BC$  care se deschide, odată cu baza lui  $T_2$ . Circulînd curent prin  $R_3$  se deschide și  $T_4$  limitînd tensiunea pe  $R_4$  la  $0,7 \text{ V}$ .

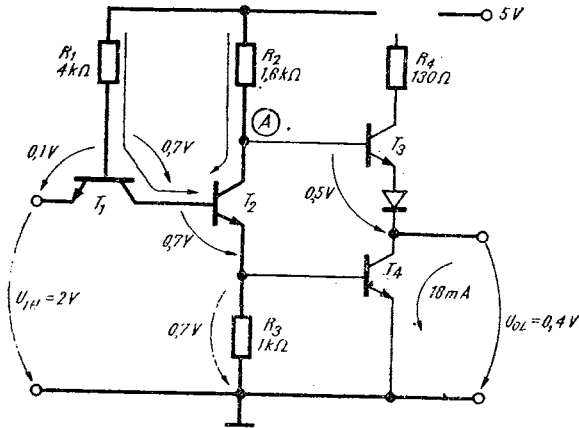


Fig. 4.2.

Deci curentul prin  $R_1$  va fi :

$$I = \frac{5 - 3 \cdot 0,7}{4} = 0,725 \text{ mA}$$

Rezultă că pentru  $T_1$ ,  $U_{BE} = 0,14 \text{ V}$ , deci jonctiunea  $BE$  e blocată  
În punctul  $A$  tensiunea va fi :

$$U_A = U_{R3} + U_{cET_2,sat} = 0,9 \text{ V}$$

În acest caz  $U_{BE T_3} + U_D = 0,5 \text{ V}$ , deci  $T_3$  și  $D$  sînt blocate. Curentul prin  $R_2$  va fi

$$I = \frac{5 - 0,9}{1,6} = 2,56 \text{ mA}$$

și trece prin  $T_2$ . Din acest curent  $0,7 \text{ mA}$  trec prin  $R_3$ , restul de  $1,8 \text{ mA}$  intrînd în baza lui  $T_4$ . Pentru  $\beta_{T_4} = 10$  se asigură în sarcină  $18 \text{ mA}$ .

**Problema 4.4.** Să se explice diferențele care apar între  $t_{PLH}$  și  $t_{PHL}$  pornind de la schema internă a unei porți ȘI NU seria standard.

*Rezolvare*

Conform definiției  $t_{PLH}$  este timpul de propagare prin circuit cînd semnalul de ieșire trece din  $L$  în  $H$ , iar  $t_{PHL}$  se definește pentru trecerea din  $H$  în  $L$  la ieșire. Întotdeauna pentru seria standard  $t_{PLH} > t_{PHL}$  ( $18 \text{ ns}$  față de  $8 \text{ ns}$ , deci cam de 2 ori mai mare), deoarece la trecerea  $L \rightarrow H$  la ieșire intervine ieșirea din saturare de 2 tranzistoare  $T_2$  și  $T_4$  înseriate (fig. 4.2), pe cînd la trecerea  $H \rightarrow L$  iese din saturare doar  $T_2$  (fig. 4.1).

**Problema 4.5.** Să se explice de ce circuitul ȘI 7411 are timpi de propagare mai mari decît ȘI NU-ul 7410.

*Rezolvare*

Schema internă a circuitului ȘI este cea din figura 4.3.

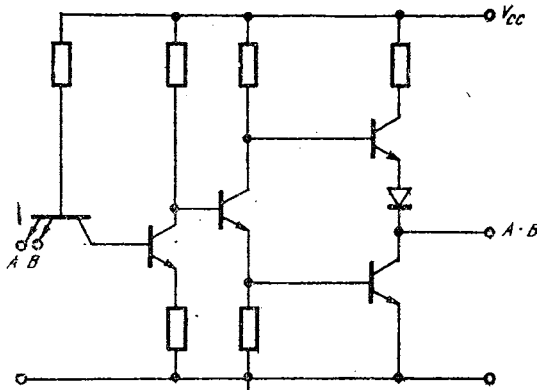


Fig. 4.3.

Acest circuit mai conține un tranzistor inversor în plus față de ȘI NU, deci va avea timpi de propagare mai mari.

**Problema 4.6.** Datorită diferenței dintre  $t_{PLH}$  și  $t_{PHL}$  la seria standard, poate apare o modificare a duratei impulsului aplicat unei scheme. Să se arate cum modifică durata impulsului un lanț cu număr par de ȘI-uri și de unul de ȘI NU-uri.

*Rezolvare*

Pentru ȘI, modul de conectare și formele de undă sînt date în figura 4.4.

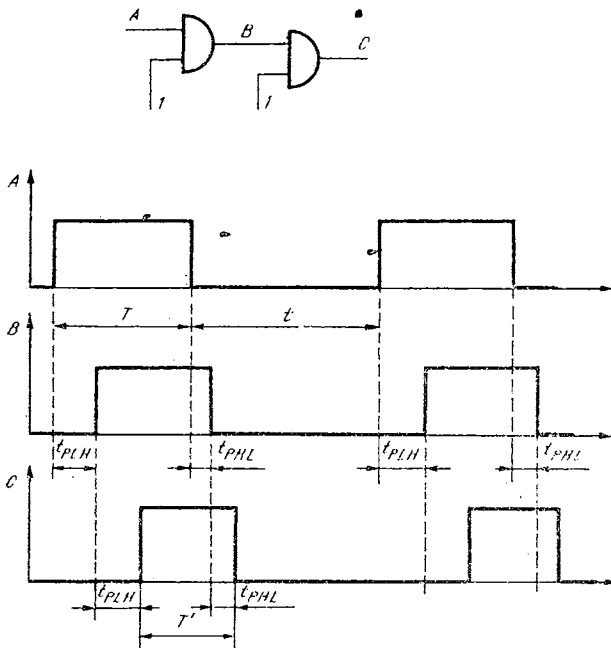


Fig. 4.4.

Din diagrame rezultă că durata  $T$  a impulsului scade la :

$$T' = T + 2t_{PHL} - 2t_{PLH}$$

Generalizînd pentru  $n$  etaje de ȘI-uri :

$$T' = T - n(t_{PLH} - t_{PHL})$$

Pentru ȘI NU, modul de conectare și formele de undă sînt date în figura 4.5.

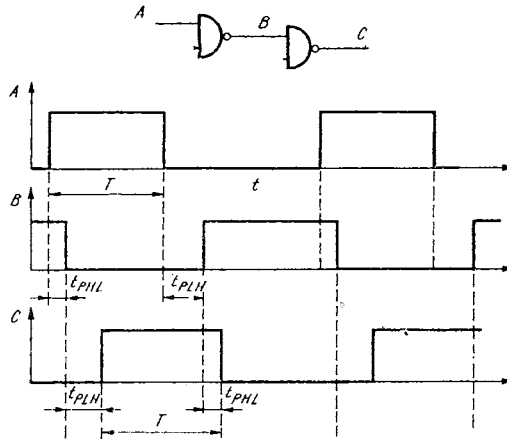


Fig. 4.5.

În acest caz, datorită inversării, după un număr par de etaje se reface durata  $T$  a impulsului.

**Problema 4.7.** Se proiectează un circuit logic combinațional cu porți, semnalul de intrare variînd cu frecvența de 6,25 MHz și factor de umplere 1/2. Cum se modifică factorul de umplere la propagarea prin 6 porți ȘI? Dar la propagarea prin 5 porți ȘI NU? Dar prin 6 porți ȘI NU?

Se dau  $t_{PHL} = 10$  ns,  $t_{PLH} = 2t_{PHL}$

*Rezolvare*

Forma de undă de la intrare este dată în figura 4.6.



Fig. 4.6.

Pentru ȘI,  $T$  scade la  $T'$ :

$$T' = T - n(t_{PLH} - t_{PHL}) = 20 \text{ ns}$$

Factorul de umplere va fi :

$$f = \frac{T'}{t'} = \frac{20}{140} = \frac{1}{7}$$

La propagarea prin 5 porți ȘI NU (prin număr impar de porți inversoare) rezultă :

$$T' = t + t_{PHL} - t_{PLH} = 70 \text{ ns}$$

$$f = \frac{T'}{t'} = \frac{70}{90} = \frac{7}{9}$$

La propagarea prin număr par de porți inversoare  $f$  nu se schimbă.

**Problema 4.8.** Semnalul la intrarea unei scheme combinaționale are frecvența de 10 MHz și factor de umplere 1/2. Câte porți ȘI se pot inseria pentru a asigura funcționarea corectă a schemei (nu distrug total impulsul de intrare). Se dau  $t_{PLH} = 2t_{PHL} = 20 \text{ ns}$ .

*Rezolvare*

Forma de undă de la intrare este dată în figura 4.7.

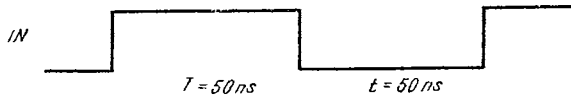


Fig. 4.7.

Durata  $T$  scade la  $T'$  astfel :

$$T' = T - n(t_{PLH} - t_{PHL}) = 50 - 10 \text{ ns}$$

Pentru 5 porți inseriate se distruge total impulsul de intrare. Răspunsul este  $n = 4$ . Dar în general în scheme trebuie ținut cont și de circuitul care este comandat în continuare : trebuie să i se asigure un impuls de o durată suficientă pentru comutare. În cazul nostru pentru  $n = 4$  rezultă  $T' = 10 \text{ ns}$ , prea mic pentru un palier ce comanda circuite din seria TTL standard (mai ales dacă mai ținem cont și de fronturi). De exemplu, pentru ca un bistabil din seria standard să funcționeze corect, durate impulsului de ceas trebuie să fie de minimum 20 ns. Deci nu sînt posibile mai mult de 3 porți ȘI inseriate.

### Observație

La frecvențe înalte contează numărul și tipul porților înseriate în calea semnalului deoarece îi modifică forma acestuia.

**Problema 4.9.** Să se explice de ce la seria TTL Schottky nu apar prelucrări ale factorului de umplere al semnalului la frecvențe înalte.

### Rezolvare

Schema unei porți ȘI NU TTL Schottky este dată în figura 4.8.

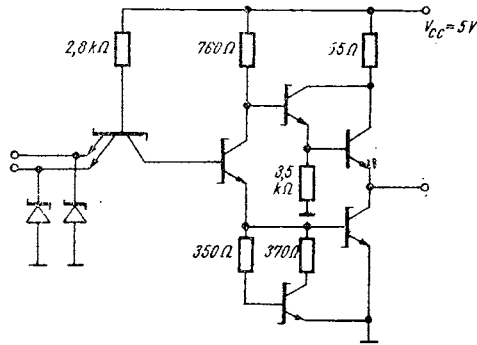


Fig. 4.8.

Simbolul de tranzistor din figură reprezintă un tranzistor cu o diodă Schottky în paralel pe joncțiunea  $BE$ . Dioda Schottky este caracterizată prin aceea că nu stochează sarcini în joncțiune, deci  $t_{stocare} = 0$  și are o tensiune, la deschidere, mai mică decât joncțiunea tranzistorului, împiedicând saturarea acestuia. Deoarece tranzistorii din schemă lucrează la limita de saturare, timpii de propagare vor fi mai mici ca la seria standard și

$$t_{PLH} = t_{PHL} \approx 3 \text{ ns}$$

ceea ce explică neinfluențarea factorului de umplere al semnalului de intrare.

**Problema 4.10.** Să se calculeze puterea absorbită de o poartă NAND din seriile TTL standard, LTTL (low power — putere mică) și HTTL (high speed — viteză mare).

Se dau pentru :

— TTL standard :  $R_1 = 4 \text{ k}$ ,  $R_2 = 1,6 \text{ k}$ ,  $R_3 = 1 \text{ k}$ ,  $R_4 = 130 \Omega$ , (notațiile sînt cele din problema 4.3) ;

- LTTL :  $R_1 = 40 \text{ k}$ ,  $R_2 = 20 \text{ k}$ ,  $R_3 = 12 \text{ k}$ ,  $R_4 = 500 \text{ } \Omega$  ;
  - HTTL :  $R_1 = 2,8 \text{ k}$ ,  $R_2 = 760 \text{ } \Omega$ ,  $R_3 = 470 \text{ } \Omega$ ,  $R_4 = 58 \text{ } \Omega$ ,  $R_5 = 4 \text{ k}$ .
- Schema HTTL este dată în figura 4.9.

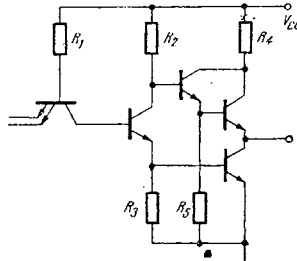


Fig. 4.9.

Să se explice de ce se obține viteză mai mare la HTTL.

**Problema 4.11.** O poartă TTL standard are ieșirea în starea  $H$  pusă în scurtcircuit la masă. Să se calculeze disipația de putere pe poartă în acest caz.

*Soluție.* Rezultă 180 mW, deci o creștere însemnată a puterii disipate pe poartă.

**Problema 4.12.** Să se prezinte avantajele și dezavantajele configurațiilor posibile de ieșire „totem pole“ pentru circuitele TTL.

*Rezolvare*

1. Totem pole tranzistor-diodă (fig. 4.10).

*Avantaje*

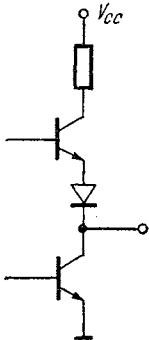


Fig. 4.10.

- consum redus de putere
- dimensiune mică (tehnologic)
- ieșirea poate fi trasă la un potențial mai mare decât  $V_{cc}$ .

*Dezavantaje*

- $U_H$  este redus datorită căderii de tensiune pe dioda deschisă.
2. Totem pole cu darlington (fig. 4.11).

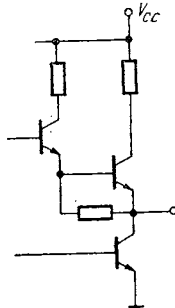


Fig. 4.11.

*Avantaje*

- capacitate mare de comandă în starea  $H$



- $U_H = V_{CC} - U_{BE}$  la  $I_{out} = 0$
- dimensiune mică.

*Dezavantaje*

- ieșirea nu poate fi la un potențial mai mare decât  $V_{CC} +$  căderea de tensiune pe o diodă deschisă.
3. Totem pole cu repetor pe emitor (darlington split) — figura 4.12.

*Avantaje*

- capacitate mare de comandă în starea  $H$
- $U_H = U_{CC} - U_{BE}$  la  $I_{out} = 0$

*Dezavantaje*

- mai mari decât circuitul anterior (o rezistență în plus)
- tensiunea la ieșire nu poate fi mai mare ca  $V_{CC} + V_D$

4. Totem pole cu darlington și rezistență la masă (fig. 4.13).

*Avantaje*

- capacitate mare de comandă în starea  $H$
- $U_H = V_{CC} - 2U_{BE}$ , e mai mic, deci crește viteza
- ieșirea poate fi ridicată la mai mult de  $V_{CC}$ .

*Dezavantaje*

- disipare mare
- imunitate mică la zgomot în starea  $H$ .

**Problema 4.13.** Să se compare ieșirea totem pole a unui circuit integrat cu cea „open collector“ (colector în gol).

*Rezolvare*

Ieșirea open collector (fig. 4.14) prezintă avantajul de a se putea cupla în paralel cu alte ieșiri de același fel, realizând funcția „ȘI CABLAT“.

*Dezavantaje*

- impedanță de ieșire mare ( $R$  față de cea a unui repetor pe emitor) în starea  $H$ ;
- fronturi și timpi de propagare mari din  $L$  în  $H$  în sarcină capacitivă;
- necesită rezistență adițională.

**Problema 4.14.** Să se descrie funcționarea unei structuri de ieșire cu trei stări (three state). Care sînt avantajele acesteia ?

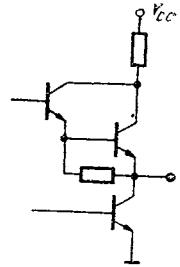


Fig. 4.12.

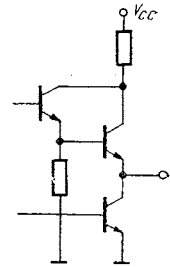


Fig. 4.13.

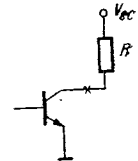


Fig. 4.14.

### Rezolvare

Pentru exemplificare se consideră schema unui inversor „three state” din figura 4.15.

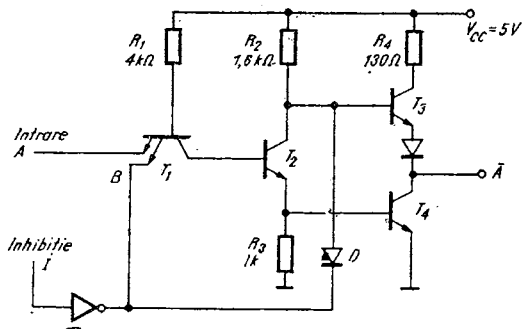


Fig. 4.15.

Pentru 0 pe intrarea de INHIBIȚIE avem semnal 1 pe emitorul  $B$  și dioda  $D$  este blocată. Circuitul funcționează ca un inversor obișnuit.

Pentru 1 pe intrarea INHIBIȚIE se deschide joncțiunea  $BE$  a lui  $T_1$ ,  $T_2$  și  $T_4$  sînt blocați, dar curentul ce trece prin  $R_2$  spre baza lui  $T_3$  este tras prin  $D$ ,  $T_3$  blocîndu-se și el. Deci indiferent de intrare, circuitul se comportă la ieșire ca o impedanță mare (tranzistor blocat).

Avantajele acestor circuite constau în aceea că se pot pune în paralel fără a se adăuga rezistențe ca la „open colector” și fără a încărcă celelalte circuite cuplate cu el. De asemenea față de circuitele „open colector”, la „three state” crește viteza de comutare.

**Problema 4.15.** Stabiliți tabelul de adevăr ce definește funcționarea unei scheme formate din cuplarea în paralel a 2 ieșiri de inversoare „three state”.

### Soluție

$A_1$	$A_2$	$I_1$	$I_2$	0
×	×	1	1	ieșirea în starea de impedanță mare ( $HiZ$ )
$A_1$	×	0	1	$\overline{A_1}$
×	$A_2$	1	0	$A_2$
0	0	0	0	0
0	1	0	0	} nu sînt permise deoarece sînt ieșiri TTL în paralel, se disipă putere mare
1	0	0	0	

S-au notat cu  $A$  intrările logice și cu  $I$  cele de inhibiție.

**Problema 4.16.** Desenați detaliat circuitul din problema anterioară și calculați curenții prin circuit în stările nepermise. Presupuneți OV tensiunea de saturare a tranzistorului.

**Problema 4.17.** Când ieșirile mai multor circuite „three state“ din seria TTL standard sînt puse în paralel, curentul de pierdere al fiecărui circuit în starea de impedanță mare) este de  $40 \mu\text{A}$ . Cîte circuite pot fi conectate în paralel pe ieșire, dacă curentul disponibil pentru un inversor neinhibat în starea  $H$  este  $5,2 \text{ mA}$  ?

**Problema 4.18.** Să se prezinte comparativ pentru circuitele TTL curenții de intrare și ieșire în stările 1 și 0.

*Rezolvare*

S.ria	$I_I$	star a 1	$I_o$	$I_I$	star.a 0	$I_o$
TTL standard	$40 \mu\text{A}$		$-400 \mu\text{A}$		$-1,6 \text{ mA}$	$16 \text{ mA}$
HTTL	$50 \mu\text{A}$		$-500 \mu\text{A}$		$-2 \text{ mA}$	$20 \text{ mA}$
LTTL (TTLLP)	$10 \mu\text{A}$		$-100 \mu\text{A}$		$-0,18 \text{ mA}$	$2 \text{ mA}$
TTLs	$50 \mu\text{A}$		$-500 \mu\text{A}$		$-2 \text{ mA}$	$20 \text{ mA}$

Semnul minus arată că, curenții ies din multiemitorul de la intrare, respectiv din circuitul de ieșire.

Tensiunile tipice sînt  $U_{OL} = 0,4 \text{ V}$ ,  $U_{OH} = 2,4 \text{ V}$ .

Se observă că „fan out“-ul, raportul  $\left[ \frac{I_o}{I_I} \right]_{L, H}$  este 10 în cadrul fiecărei clase de circuite TTL.

**Problema 4.19.** Să se calculeze valoarea maximă a rezistenței ce poate fi cuplată între două porți din seria standard, fără a afecta funcționarea acestora.

*Rezolvare*

Modul de conectare al rezistenței este prezentat în figura 4.16.

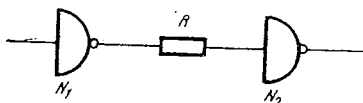


Fig. 4.16.

$$\frac{N_1}{L} \Big| \frac{N_2}{H} \quad R_{Max} = \frac{U_{IL} - U_{OL}}{I_{IL}} = \frac{0,8 - 0,4}{1,6} = 250 \ \Omega$$

Dacă  $R$  este mai mare nu se mai asigură curentul de intrare suficient pentru  $N_2$ .

$$\frac{N_1}{H} \Big| \frac{N_2}{L} \quad R_{Max} = \frac{U_{OH} - U_{IH}}{I_{IH}} = \frac{2,4 - 2}{40 \ \mu A} = \frac{400 \text{ mV}}{40 \ \mu A} = 10 \text{ k}$$

Deci starea  $L$  pentru  $N_1$  este cea care limitează valoarea rezistenței  $R$ .

Prin cuplarea lui  $R$  se distruge marginea de zgomot pentru  $N_2$ .

**Problema 4.20.** Să se calculeze  $R_{Max}$  ce poate fi cuplat între două porți LTTL fără a afecta funcționarea.

**Problema 4.21.** Care este numărul de intrări LTTL ce poate fi comandat cu o poartă standard?

**Problema 4.22.** Calculați „fan out“-ul unei porți LTTL încărcată cu porți TTL standard.

**Problema 4.23.** Care este „fan out“-ul unui circuit TTLS comandînd circuite TTL standard?

**Problema 4.24. a)** Să se proiecteze cu circuite „open collector“ un circuit ȘI cu 8 intrări;

**b)** Să se dimensioneze rezistența  $R$  pentru sarcină de 5 intrări TTL standard.

*Rezolvare*

**a)** Se utilizează circuitul 7409 (4AND cu colectorul în gol) conectat ca în figura 4.17.

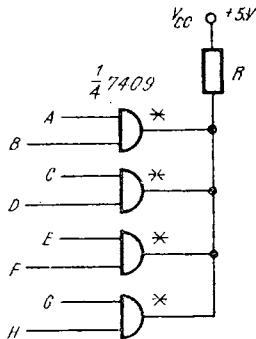


Fig. 4.17.

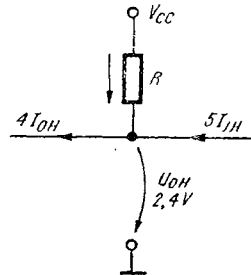


Fig. 4.18.

b) Pentru starea 1 la ieșire trebuie să se asigure pentru intrările de sarcină 2,4 V și curenții de 40  $\mu$ A, Schema echivalentă este cea din figura 4.18, tranzistorii blocați absorbind fiecare  $I_{OH} = 250 \mu$ A (dat în catalog).

Pentru starea 0 la ieșire,  $R$  trebuie să asigure 0,4 V și curenții de ieșire (16 mA) și de intrare (1,6 mA) pentru CI de comandă, respectiv din sarcină.

$$R_{Max} = \frac{V_{CC} - V_{OL}}{I_{OL} + 5 I_{IL}} = \frac{4,6 \text{ V}}{(16 - 8) \text{ mA}} = 570 \Omega$$

S-a presupus că numai o poartă ȘI de comandă este deschisă. În cazul în care sînt deschise mai multe, curențul de 16 mA disponibil se distribuie prin acestea. Este evident că valoarea de la 0 logic este cea limitativă. Alegem  $R = 570 \Omega$  și vor fi respectate și condițiile de asigurare a funcționării pentru 1 logic.

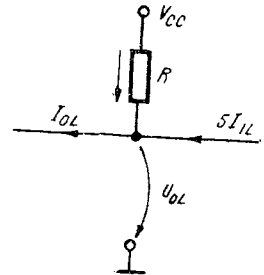


Fig. 4.19.

**Problema 4.25.** a) Să se proiecteze cu circuite „open collector“ din seria standard

$$f = \overline{AB} \cdot \overline{CD} \cdot \overline{DE} \cdot \overline{BC}$$

b) Să se calculeze cîte circuite din seria LTTL se pot comanda dacă  $R = 1 \text{ k}\Omega$ .

*Indicație a.* Se utilizează circuitul 7401 (4 NAND cu 2 intrări, o.c.).

**Problema 4.26.** a) Să se realizeze cu un singur circuit integrat „open collector“ din seria standard

$$f = \bar{A} \cdot \bar{B} \cdot \bar{C} \cdot \bar{D} \cdot \bar{E} \cdot \bar{F}$$

b) Să se dimensioneze  $R$  astfel încît să poată fi comandate 12 intrări din seria HTTL.

*Indicație a.* Se utilizează 7405 (6 NON o.c.).

**Problema 4.27.** Pentru realizarea unei funcții logice se cuplează în paralel 10 ieșiri din seria standard TTL.

a) Să se dimensioneze  $R$  pentru comanda a 8 intrări LTTL.

b) Cîte intrări HTTL se pot comanda cu aceeași rezistență ?

c) Cîte ieșiri TTL standard se pot comanda cu  $R$  ?

d) Să se redimensioneze  $R$  pentru comanda a 8 intrări TTL standard.

e) Pentru care din rezistențele calculate la punctele a și d se obține  $t_{PLH}$  mai mare și de ce ?

**Problema 4.28.** Să se descrie cu ajutorul formelor de undă funcționarea schemei din figura 4.20.

*Rezolvare*

Formele de undă sint date în figura 4.21.

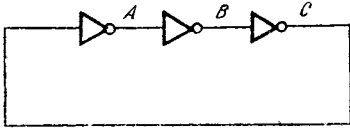


Fig. 4.20.

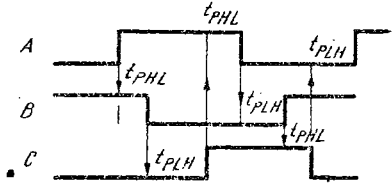


Fig. 4.21.

Circuitul este un oscilator cu frecvența

$$f = \frac{1}{(t_{PLH} + 2 t_{PHL}) + (t_{PHL} + 2 t_{PLH})} = \frac{1}{3(t_{PLH} + t_{PHL})}$$

Generalizînd pentru cazul a  $n$  circuite ( $n$  impar) :

$$f = \frac{1}{n(t_{PLH} + t_{PHL})}$$

deci frecvența variază discret cu  $n$ .

**Problema 4.29.** a) Cum se modifică funcționarea schemei anterioare dacă se montează  $R$  și  $C$  ca în figura 4.22.

b) Să se deseneze formele de undă în  $MNPQ$ .

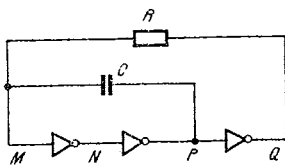


Fig. 4.22.

c) Să se dimensioneze  $R$  și  $C$  pentru a obține o frecvență de oscilație de 1 MHz.

d) Explicați de ce primul inversor din schemă trebuie să aibă caracteristică de transfer cu histerzis (Trigger Schmitt).

e) Care este valoarea maxim admisibilă pentru  $R$  ?

*Rezolvare*

d) În situația cînd are loc tranziția  $NPQ = 010$ ,  $C$  care era încărcat astfel încît să asigure 1 în  $M$ , se descarcă prin  $R$ , potențialul în  $M$  trecînd spre 0 logic. Invers, pentru  $MPQ = 101$ ,  $C$  se încarcă la 1 logic. Regimul tranzitoriu al lui  $C$  care determină de altfel frecvența de oscilație produce o variație lentă a semnalului din punctul  $M$ . Se știe că

la astfel de situații un CI obișnuit oscilează, deoarece pentru tensiuni de intrare între cele 2 praguri (0,8–2 V), tranzistorii din etajul final sînt în regiunea activă normală și apare o reacție pozitivă (figura 4.23): dacă  $I_{CT_3}$  crește, crește  $U_A$ , scade  $U_0$ , crește  $I_{CT_4}$ , crește  $I_{BT_4}$ , deci va crește și mai mult  $I_{CT_3}$ .

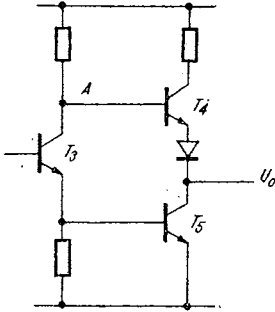


Fig. 4.23.

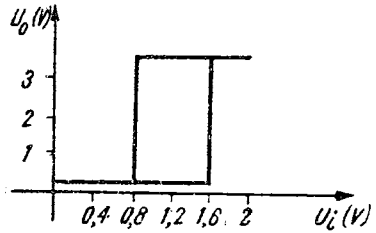


Fig. 4.24.

Utilizînd un circuit „trigger Schmitt” avînd caracteristica de transfer sub forma unui ciclu de histerzis, ca în figura 4.24, comutarea se produce cînd tensiunea pe condensator atinge pragurile și nu mai apar oscilații.

e) vezi problema 4.19.

**Problema 4.30.** Să se explice funcționarea schemei din figura 4.25.

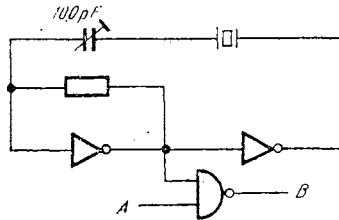


Fig. 4.25.

### Indicație

Frecvența cuarțului impune frecvența de oscilație a ansamblului. Pentru a nu atenua prea mult semnalul de reacție, rezistența serie a cuarțului nu trebuie să depășească 250 Ω.

Poarta NAND e utilizată pentru închiderea sau deschiderea, cu semnalul de pe A, a oscilațiilor ce se transmit la B.

**Problema 4.31.** a) Desenați diagramele de timp ce descriu funcționarea schemei (în punctele  $M$ ,  $N$ ,  $P$ ,  $Q$ ) din figura 4.26.

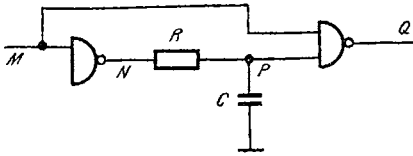


Fig. 4.26.

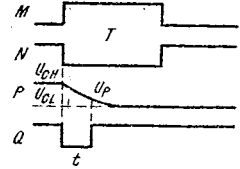


Fig. 4.27.

b) Calculați durata de încărcare a  $C$  semnificativă pentru funcționarea circuitului, dacă :

$$U_{OH} = 4,3 \text{ V}, U_{OL} = 0, U_{P(\text{prag})} = 1,5 \text{ V}$$

c) Care este valcarea maximă posibilă pentru  $R$  ?

d) Care NAND trebuie să fie cu intrare de „trigger Schmitt” și de ce ?

e) Cum influențează durata semnalului de intrare funcționarea circuitului ?

#### Rezolvare

a) Circuitul se comportă ca un monostabil cu durata  $t$  comandat pe frontul pozitiv, avînd starea stabilă  $1$  (fig. 4.27).

$$b) t = \tau \ln \frac{U_{OH} - U_{OL}}{U_P - U_L} \cong \tau \ln \frac{U_{OH}}{U_P} = RC \ln \frac{4,3}{1,5} = RC$$

e) Dacă  $T < t$ , circuitul nu se mai comportă ca monostabil, ci va inversa doar semnalul de intrare. Nu mai contează, în acest caz regimul tranzitoriu al condensatorului.

**Problema 4.32.** Reluați problema anterioară pentru circuitul din figura 4.28.

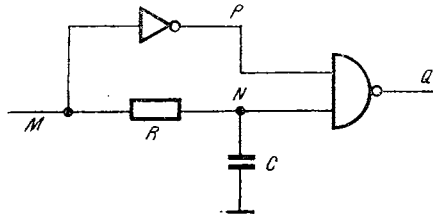


Fig. 4.28



*Indicație*

Circuitul este un monostabil comutînd pe frontul negativ, avînd starea stabilă 1.

**Problema 4.33.** Să se descrie prin forme de undă funcționarea circuitelor din figura 4.29.

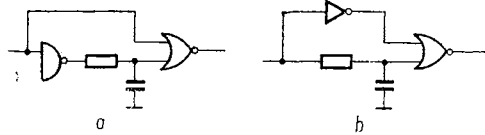


Fig. 4.29.

*Indicație*

- a) monostabil comutînd cu frontul negativ și stare stabilă 0
- b) monostabil comutînd cu frontul pozitiv și stare stabilă 0.

**Problema 4.34.** Se dă circuitul din figură 4.30.

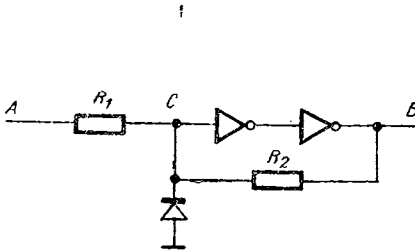


Fig. 4.30.



Fig. 4.31.

- a) aplicînd la intrare semnalul din figura 4.31, să se deducă răspunsul circuitului la ieșire și să se explice funcționarea schemei.
- b) Să se dimensioneze  $R_1$  și  $R_2$ .

*Indicație.*

Circuitul este un „trigger Schmitt“ comutînd cu tensiune de prag negativă.

**Problema 4.35.** Să se descrie prin diagrame de timp funcționarea circuitului din figura 4.32.

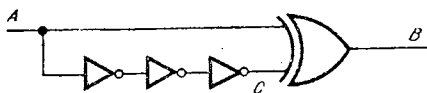


Fig. 4.32.

cunoscînd  $t_{PLH} = 2 t_{PHL} = 20$  ns.

*Rezolvare*

Circuitul este un detector de fronturi: semnalizează cu un impuls de 50 ns către masă apariția unei tranziții (front pozitiv sau negativ) la intrare (fig. 4.33).

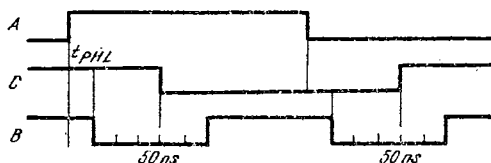


Fig. 4.33.

**Problema 4.36.** Să se descrie prin diagrame de timp funcționarea circuitului din figura 4.34, dacă  $t_{PLH} = t_{PHL} = 10$  ns.

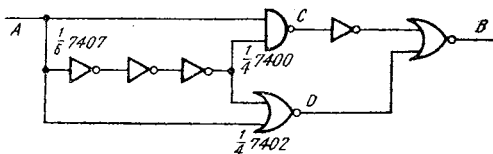


Fig. 4.34.

*Soluție*

Formele de undă sînt prezentate în figura 4.35.



Fig. 4.35.



### Rezolvare

Intervalul între două impulsuri de tact trebuie să fie mai mare decât suma timpilor de propagare prin primul bistabil (maxim 40 ns) și prin cele 2 porți (maxim 10 ns și respectiv 22 ns), pentru ca informația să poată ajunge la intrarea  $J$ . Corespunzător acestui interval de 72 ns, rezultă că frecvența la care poate funcționa schema trebuie să fie mai mică de 13,8 MHz.

**Problema 4.40.** Numărătorul cu 74193 din figura 3.26 cu reacție directă  $BR-LD$  este cuplat la intrările unui DCD 74154.

a) Să se determine duratele de rămânere în  $\theta$  la ieșirile  $\bar{O}_0 \div \bar{O}_{15}$  ale decodificatorului dacă pe  $CD$  se aplică impulsuri cu perioada  $T = T_i + T_p$  ( $T_i$  — durata impulsului,  $T_p$  — durata pauzei).

b) Care este durata de rămânere în  $\theta$  a impulsului pe  $BR$ ?

### Rezolvare

a) Ieșirile  $\bar{O}_{14} - \bar{O}_1$  ale DCD trec succesiv în  $\theta$ , avînd durata  $T$  de menținere a acestei stări. DCD semnalizează apariția numărului 0 la ieșirea 74193 ( $\bar{O}_0 = 0$ ) un interval egal cu  $T_i$ , iar  $\bar{O}_{15} = 0$  pe durata  $T_p$  (vezi fig. 3.27).

b) Formele de undă la apariția semnalului  $BR$  sînt cele din figura 4.39.  $BR$  trece în  $\theta$  cînd numărătorul este în starea 0000 și apare frontul negativ al ceasului ( $CD$ ), întîrziat față de acesta cu timpul de propagare de la  $CD$  la  $BR$ :

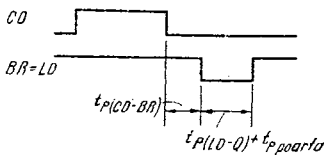


Fig. 4.39.

$t_p(CD-BK) = 25$  ns. Rămîne în  $\theta$  un timp egal cu cel necesar bistabililor să comute (timpul de propagare de la  $LD$  la  $Q - t_p(LD-Q) = 27 - 40$  ns) plus timpul necesar pentru comutarea porții ȘI NU care adună ieșirile

$Q$  ale numărătorului în semnalul  $BR$  (vezi schema internă a 74193 din anexă). Putem deci estima că durata rămîinerii în  $\theta$  a semnalului  $BR$  va fi de 35—47 ns.

**Problema 4.41.** Ieșirile unui ceas cu 4 faze obținut cu ajutorul unui registru în care se deplasează informația 0111 sînt conectate la porți ȘI NU ca în figura 4.40.

a) Să se deseneze formele de undă la cele 6 faze ale ceasului, neglijînd timpii de propagare.

b) Să se deseneze formele de undă ținînd cont de timpii de propagare. Care este situația cea mai defavorabilă pentru  $t_{PLH}$  și  $t_{PHL}$  ai bistabililor registrului pentru care  $\bar{O}_5$  și  $\bar{O}_6$  au semnale parazite.

### Rezolvare

a) Formele de undă se dau în figura 4.41.

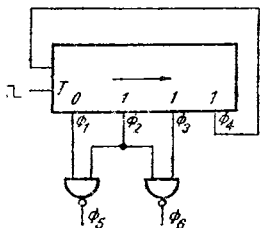


Fig. 4.40.

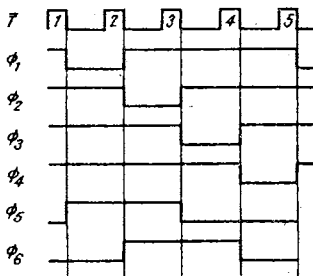


Fig. 4.41.

b) Se vor desena detaliat formele de undă pentru impulsurile 2, 3 și 4 (fig. 4.42). Situația cea mai defavorabilă, în care bistabilii registrului au  $t_{PHL} > t_{PLH} + t'_{PHL}$  conduce la apariția unui impuls parazit la  $\phi_5$  pe frontul negativ al impulsului 2 și la  $\phi_6$  pe frontul negativ al impulsului 3. S-a considerat că timpii de propagare ai porților ȘI NU ( $t_p$ ) sînt mai mici decît cei ai registrului ( $t_p$ ). Pentru  $t_{PHL} \leq t_{PLH} + t'_{PHL}$  nu apar impulsurile parazite.

**Problema 4.42. a)** Să se determine pentru numărătorul binar sincron pînă la 1024 din figura 3.124, frecvența maximă a impulsurilor de ceas pentru care este posibilă funcționarea.

b) Să se modifice schema astfel încît CLC să conțină două nivele de porți logice. Care va fi frecvența maximă de funcționare în acest caz ?

Se vor considera următorii timpii maximi de propagare (Circuite Signetics)

$$7408 : t_{PLH} = 27 \text{ ns} ; t_{PHL} = 19 \text{ ns}$$

$$7473 : t'_{PLH} = 25 \text{ ns} ; t'_{PHL} = 40 \text{ ns}$$

### Rezolvare

a) Impulsurile de tact trebuie date cu o frecvență astfel încît să se asigure timpul total de propagare prin bistabili și porți ( $t_p$ ). Acest timp

este maxim cînd toți bistabilii  $Q_1 - Q_8$  sînt pe 1 și  $Q_0$  trece din 0 în 1 : toate ȘI-urile fiind pe 0, comută pe 1. Rezultă :

$$t_{pMax} = t'_{PLH} + 8 t_{PLH} = 241 \text{ ns}$$

De abia după acest timp apare 1 la ieșirea  $T_9$  și poate fi dat următorul front negativ al tactului pentru comutarea bistabililor. Deci frecvența maximă la care poate funcționa schema va fi :

$$f = \frac{1}{241 \text{ ns}} = 4,14 \text{ MHz}$$

b) În schema din figura 3.124 frecvența maximă de lucru este redusă datorită faptului că CLC conține 8 nivele de porți. Acestea se pot reduce la două nivele cu schema din figura 4.43. Considerînd

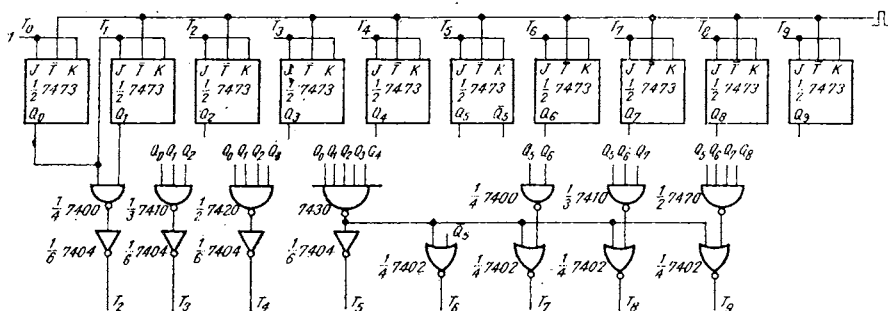


Fig. 4.43.

pentru porțile respective timpii maximi  $t_{PLH} = 22 \text{ ns}$  și  $t_{PHL} = 15 \text{ ns}$  rezultă că, în cazul cel mai defavorabil :

$$t_{pMax} = t'_{PHL} + t_{PLH} + t_{PHL} = 81 \text{ ns}$$

Deci această schemă poate funcționa la o frecvență de 3 ori mai mare decît cea cu 8 nivele de porți.

**Problema 4.43.** Să se determine toate stările parazite care apar la ieșirea unui decodificator comandat de un numărator asincron 1-2-4 realizat cu bistabilii JK. Se consideră timpii de propagare de la ceas la  $Q$ ,  $t_{PLH} = 16 \text{ ns}$  și  $t_{PHL} = 25 \text{ ns}$ .

Ce modificări apar dacă numărătorul este sincron ?

*Rezolvare*

Formele de undă se dau în figura 4.44. Se observă că datorită faptului că bistabilii în numărătorul asincron comută în cascadă apar 1 sau chiar 2 stări parazite între cele utile. Acestea se transmit la ieșirea DCD care va funcționa defectuos.

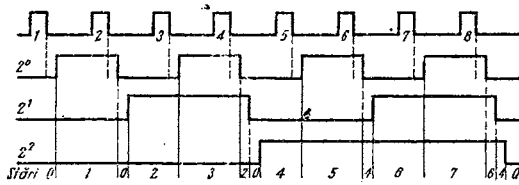


Fig. 4.44.

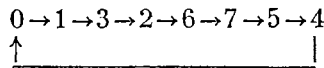
Dacă numărătorul este sincron, bistabilii comută simultan și DCD nu mai are semnale parazite la ieșire.

**Problema 4.44.** Să se determine frecvența maximă de lucru a unui numărător asincron pînă la 16 realizat cu bistabili JK ( $t_{PLH} = t_{PHL} = t_p = 20$  ns).

*Rezolvare*

Situațiile cele mai defavorabile apar pe fronturile negative ale impulsurilor 8 și 16 (vezi și problema anterioară) cînd comută, unul după altul, toți cei 4 bistabili ai numărătorului. Perioada impulsurilor de tact trebuie să fie mai mare decît  $4 t_p$ . Asigurînd prezența stărilor la ieșire un interval de minimum 40 ns, rezultă că numărătorul va funcționa corect pînă la 10 MHz.

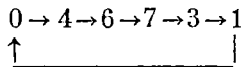
**Problema 4.45.** Să se explice de ce se înlătură semnalele parazite la decodificarea unui numărător în cod Gray.



*Soluție*

Deoarece codul Gray are proprietatea că se modifică doar un singur bit de la o tranziție la alta : deci comută un singur bistabil.

**Problema 4.46.** a) Să se calculeze frecvența maximă de lucru a numărătorului Moebius (Johnson) realizat cu registru de deplasare 7495 :



b) Să se arate dacă apar semnale parazite la decodificare.

**Problema 4.47.** Să se calculeze frecvența maximă de lucru a schemei din figura 4.45 știind că timpii tipici de propagare sînt :

Circuitul	$t_{PLH}$ (ns)	$t_{PHL}$ (ns)
7495	18	21
7402	12	8
7404	12	8
7408	18	12
7468	15	11

**Problema 4.48.** Se conectează divizorul de frecvență programabil din figura 3.40 astfel încît să se obțină divizarea maximă. Să se deseneze, ținînd cont de timpii de propagare, formele de undă la  $CR$ ,  $J$ ,  $K$ ,  $QE$  și  $LD$  pentru impulsurile 15, 16 și 30.

Se consideră pentru circuite timpii de propagare din tabelul următor :

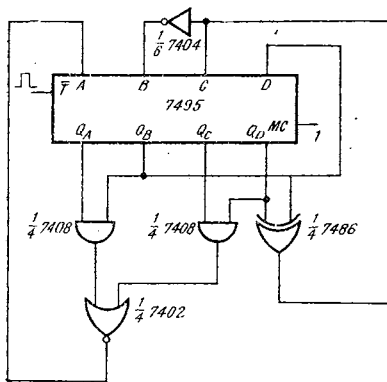


Fig. 4.45.

Circuitul	$t_{PLH}$ (ns)		$t_{PHL}$ (ns)	
	tipic	max	tipic	max
74193				
$t_p CD - CR$	19	26	16	24
7476	16	25	25	40
7404	12	22	8	15
7400	11	22	7	15

### Soluție

Formele de undă sînt prezentate în figura 4.46. Se observă că la al 16-lea impuls în situația cea mai defavorabilă pentru întirzieri se obține un semnal parazit de durată maximă :

$$t = t_{PLH(193)} + t_{PHL(04)} + t_{PLH(00)Max} - t_{PHL(04)} + t_{PLH(76)} + t_{PHL(00)min} = 32 \text{ ns.}$$

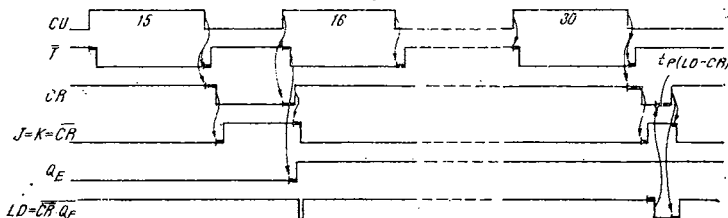


Fig. 4.46.



Acesta poate produce o încărcare nedorită a divizorului. Calculînd cu timpii medii, impulsul parazit este de numai 7 ns și nu poate încărca divizorul. Deci, schema poate funcționa dar este nefiabilă la o eventuală reproductibilitate fără selectarea circuitelor.

**Problema 4.49.** Să se descrie funcționarea porții NOR (OR) ECL (fig. 4.47), explicînd rolul elementelor din schemă.

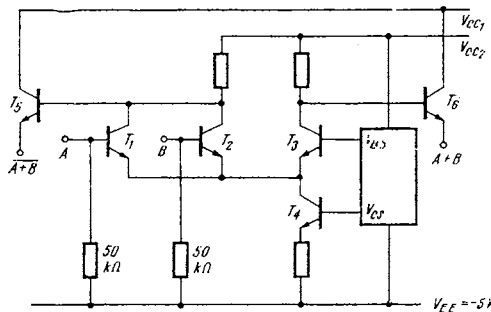


Fig. 4.47.

#### Rezolvare

Circuitul funcționează cu nivele negative. În general

$$V_{OH} = -1 \text{ V}$$

$$V_{OL} = -1,6 \text{ V.}$$

Cînd potențialul în  $A$  sau  $B$  este mai negativ decît  $V_{BE}$ , tranzistorii  $T_1$ , respectiv  $T_2$  conduc,  $T_3$  blocîndu-se (datorită generatorului de curent constant  $T_4$ ). Tensiunea în emitorul lui  $T_5$  devine mai negativă, iar cea din emitorul lui  $T_6$  tinde către masă.

Evident la etajele de ieșire (repetoare pe emitor) trebuie adăugată o rezistență de sarcină exterioară. Există două terminale  $V_{CC1}$ ,  $V_{CC2}$ , separate pentru etajele de ieșire și restul schemei pentru a elimina zgomotele datorate cuplajelor. (Variațiile de curenți de sarcină în timpul comutării nu se transmit în  $V_{CC2}$ ).

Ieșirea pe repetor asigură o impedanță mică de comandă ( $7 \Omega$ ) și variații de tensiune compatibile cu intrările.

Generatorul de curent constant mărește viteza de comutare și impedanța de intrare în dispozitiv.

Timpi de propagare : 2 ns.

Putere disipată : 25 mW/porțată.

**Problema 4.50.** Desenați structura internă a unui circuit NAND (AND) ECL.

*Soluție*

Vezi figura 4.48.

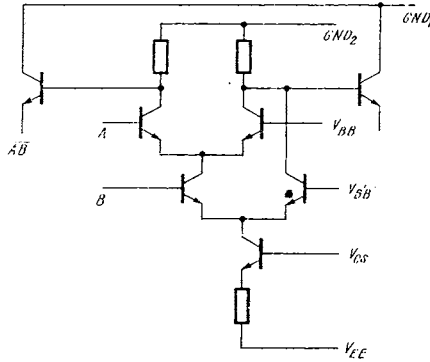


Fig. 4.48.

**Problema 4.51.** Desenați structura internă a unui circuit NOR ECL.

*Soluție*

Vezi figura 4.49.

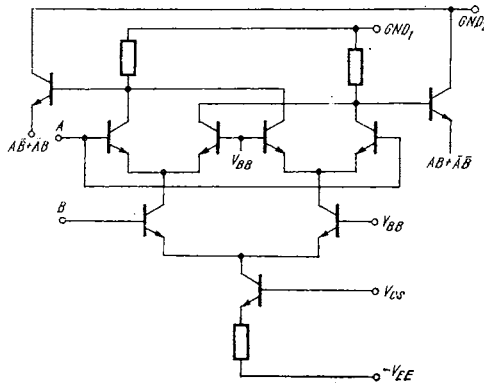


Fig. 4.49.

**Problema 4.52.** Ce funcții va realiza poarta ECL din problema 4.49 dacă se complementează intrările ?

**Problema 4.53.** Să se realizeze cu porți OR/NOR ECL cu 2 intrări, funcții NAND de 8 variabile.

*Indicație*

Se complementează variabilele de intrare și se conectează emitorii în paralel.

**Problema 4.54.** Să se explice de ce cu CI ECL se obțin atât funcția cît și negația acesteia.

**Problema 4.55.** Să se dimensioneze rezistența din emitorul unei ieșiri de circuit ECL, conectată ca în figura 4.50., dacă se dau :

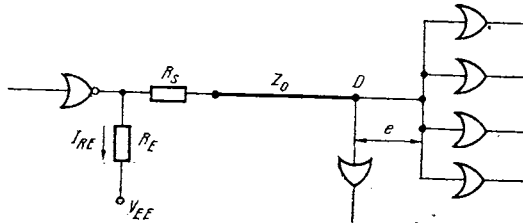


Fig. 4.50.

$R_0$  (rezistența medie de ieșire) =  $7 \Omega$

$R_S = Z_0 - R_0$  este rezistența serie de adaptare

$V_{CC1} = V_{CC2} = +2 \text{ V}$

$V_{EE} = -3,2 \text{ V}$   $V_{OH} = -0,82 \text{ V}$

Tranzițiile introduc în linie o variație de  $0,4 \text{ V}$ , iar curentul prin  $R_E$  trebuie să preia o variație cu  $30\%$  mai mare pentru a evita reflexiile.

*Indicație*

Presupunînd că la ieșire apare saltul spre  $V_{OH}$ ,  $Z_0$  acționează ca o rezistență conectată la  $V_{OH}$ . Circuitul echivalent va fi cel din figura 4.51.

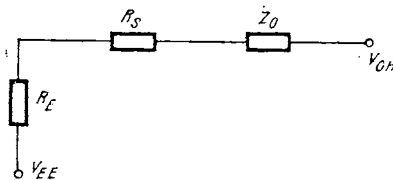


Fig. 4.51.

Deci

$$I_{R_E} = \frac{V_{OH} - V_{EE}}{R_E + R_S + Z_O} \geq \frac{0,4 + 30\% \cdot 0,4}{Z_O}$$

de unde rezultă  $R_E$

$$\begin{aligned} R_E &\leq R_O + Z_O \left( \frac{V_{OH} - V_{EE}}{0,52} + 2 \right) = \\ &= 7 \Omega + Z_O \left( \frac{-0,82 + 3,2}{0,52} + 2 \right) = 6,4 \cdot Z_O + 7 \Omega \end{aligned}$$

**Problema 4.56.** Să se calculeze  $R_E$  care se cuplează în condițiile problemei anterioare la o linie de  $Z_O = 50 \Omega$ .

**Problema 4.57.** Cum trebuie cuplată grila tranzistorului — rezistență de sarcină a unui inversor static MOS cu canal  $p$  cu îmbogățire astfel încât acesta să lucreze.

- în regim triodă (PELT —  $p$  enhancement load triode)
- în regim saturat (PELS)

Cum arată caracteristicile de transfer în aceste cazuri?

*Rezolvare*

a) Pentru ca  $T_1$  să lucreze în regim de triodă trebuie ca grila să fie la un potențial mai negativ decât drena (fig. 4.52).

$$U_{GG} \cong 3 U_{DD}$$

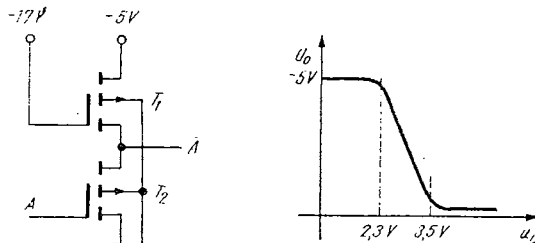


Fig. 4.52.

b) Pentru o sarcină saturată  $U_{GG} = U_{DD}$ , este necesară o singură sursă de alimentare (fig. 4.53). Dar excursia la ieșire nu atinge  $U_{DD}$  ci,  $U_{DL} = U_{pna g T_1}$ . În acest caz există și dezavantajul că rezistența de sarcină este mai mare, deci inversorul va putea comanda o sarcină capacitivă mai mică.

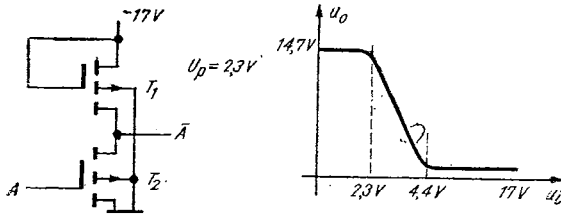


Fig. 4.53.

**Problema 4.58.** Să se realizeze cu PELS un circuit static ȘI NU cu 2 intrări  $A$  și  $B$  la care există corespondențele

$$a) \quad \frac{-14,7 \text{ V} \mid 1}{0 \text{ V} \mid 0}$$

$$b) \quad \frac{-14,7 \text{ V} \mid 0}{0 \text{ V} \mid 1}$$

*Soluție*

Vezi figura 4.54.

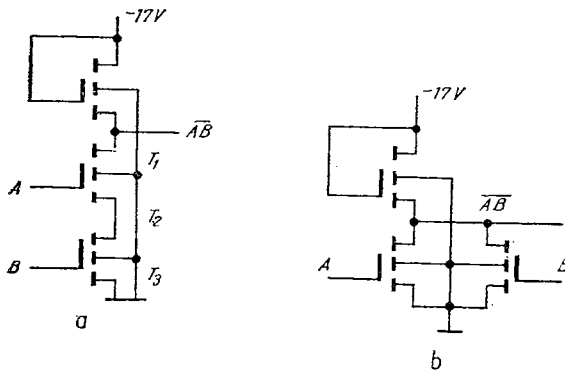


Fig. 4.54.

**Problema 4.59.** Să se realizeze cu PELT un circuit static SAU NU la care 1 este asociat nivelului negativ.

**Problema 4.60.** Să se indice schemele realizate cu tranzistoare MOS cu canal  $p$  cu sărăcire cu sarcină în regim de triodă (PDLT-D-deplstion) pentru funcțiile :

$$a) f = \bar{A}$$

$$b) f = \overline{AB + CD}$$

$$c) f = \overline{(A + B)(C + D)}$$

*Soluție*

Pentru cazul a) soluția este prezentată în figura 4.55.

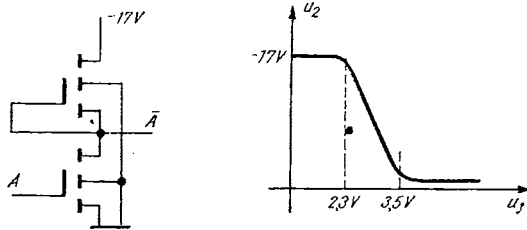


Fig. 4.55.

**Problema 4.61.** Să se implementeze cu porți MOS funcția SAU EXCLUSIV (XOR), dacă la intrare există atât variabilele cât și complementele acestora.

*Indicație*

O soluție generală este prezentată în figura 4.56.

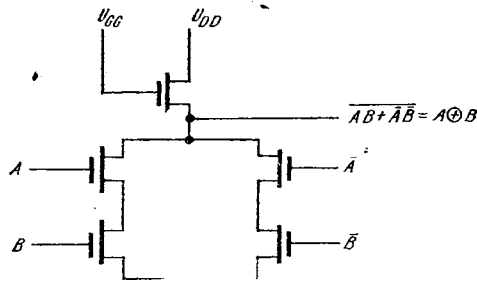


Fig. 4.56.

Simbolurile folosite nu țin cont de tipul de MOS, de polarități. Schema se poate particulariza în fiecare caz în parte.

**Problema 4.62.** Să se descrie prin diagrame de timp funcționarea inverzorului MOS dinamic cu trei tranzistoare din figura 4.57.

*Rezolvare*

Capacitățile  $C_1$  și  $C_2$  sînt parazite și  $C_1 \gg C_2$ . Presupunînd că la momentul  $t_0$ ,  $A = 0(V)$  și  $\phi_1$  trece în 1, (fig. 4.58)  $T_2$  se blochează,  $T_1$  conduce și încarcă pe  $C_1$ . Cînd  $\phi_2$  trece în 1, (la momentul  $t_1$ ),  $T_3$  conduce, sarcina din  $C_1$  transferîndu-se în  $C_2$ . Deoarece  $C_1$  este mare, variația sa de tensiune în timpul transferului de sarcină nu este prea mare. Dacă apoi ( $t_2 - t_3$ ),  $A$  trece în 1,  $\phi_1 = 1$ ,  $\phi_2 = 0$ , se descarcă parțial  $C_1$ , dar  $C_2$  rămîne încărcat deoarece  $T_3$  este blocat. La  $t_4$ , cînd  $\phi_2 = 1$ ,  $T_3$  va conduce iar  $C_2$  se va descărca prin  $T_3$  și  $T_2$ .

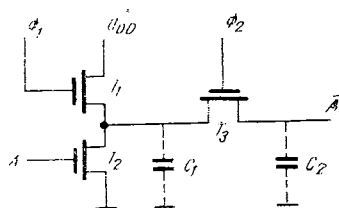


Fig. 4.57.

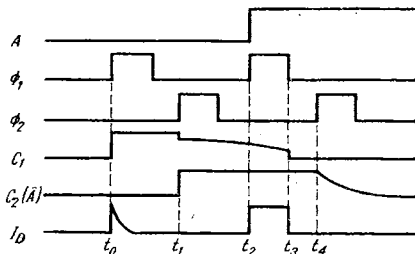


Fig. 4.58.

Se observă că ieșirea devine complementul intrării cu o întîrziere de  $(t_1 - t_0) = (t_4 - t_2)$ , dată de diferența în timp a celor două ceasuri.

Curentul  $I_D$  extras din sursă trece prin circuit la  $t_0$  pentru a încărca  $C_1$  și apoi în intervalul  $t_2 - t_3$  cînd conduc atît  $T_1$  cît și  $T_2$ . Se observă că din această cauză dispariția de putere e mai redusă ca la inverzorul static.

**Problema 4.63.** Implementați o schemă de ȘI NU (NAND) cu 2 intrări, cu MOS dinamic.

*Soluție*

Vezi figura 4.59.

**Problema 4.64.** Realizați un SAU NU (NOR) MOS dinamic cu 3 intrări.

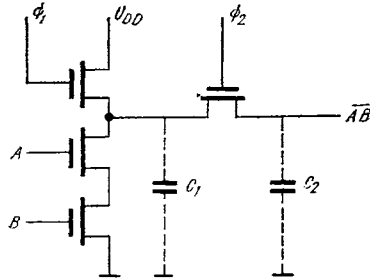


Fig. 4.59.

**Problema 4.65.** Să se descrie prin diagrame de timp funcționarea inversorului MOS dinamic cu 4 tranzistoare din figura 4.60.

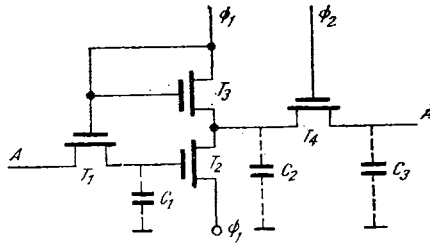


Fig. 4.60.

**Rezolvare**

Presupunem că în starea inițială  $t_0$  (fig. 4.61), la intrare,  $A = 1$ ,  $\phi_1 = \phi_2 = 0$ ,  $C_1$  este încărcată. La  $t_1$ ,  $A = 0$ ,  $\phi_1 = 1$ , descărcând

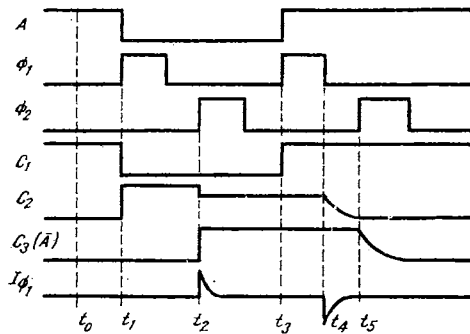


Fig. 4.61.



pe  $C_1$  prin  $T_1$  și încărcând pe  $C_2$  prin  $T_3$ . La  $t_2$  se transferă sarcina din  $C_2$  în  $C_3$  prin  $T_4$  saturat. La  $t_3$ ,  $A = 1$ ,  $\varnothing_1 = 1$  ceea ce deschide pe  $T_1$  permițând încărcarea lui  $C_1$ . În intervalul  $t_3-t_4$ ,  $C_2$  nu se poate descărca deoarece terminalele lui  $T_2$  și  $T_3$  sînt la același potențial. Astfel, cînd ceasul  $\varnothing_1$  trece în 0, la  $t_4$ , starea circuitului este următoarea:  $T_1$  și  $T_3$  blocate,  $C_1$  este încărcat și  $C_2$  se descarcă prin  $T_2$ . La  $t_5$ ,  $T_2$  este deschis,  $C_3$  se descarcă prin  $T_4$  și  $T_2$ . Ieșirea este complementul intrării cu o întârziere  $t_2-t_1$ .

Variația de curent pe calea  $\varnothing_1$ ,  $I_{\varnothing}$ , are loc doar la încărcarea lui  $C_2$  prin  $T_3$ . Puterea consumată de acest circuit este mai mică decît la inverterul dinamic cu 3 tranzistoare (problema 4.62). Informația este memorată în sarcina de pe condensatoare.

**Problema 4.66.** Proiectați un ȘI NU (NAND) MOS dinamic cu celulă cu 4 tranzistoare, cu 2 intrări și descrieți funcționarea prin forme de undă.

**Problema 4.67.** Proiectați bistabili cu ceas, de tip :

- a) RS
- b) JK
- c) T

cu inversoare și porți NAND sau NOR cu MOS :

- 1) static
- 2) dinamic.

**Problema 4.68.** Să se explice funcționarea unui registru de deplasare static de un bit.

Celula de deplasare de 1 bit statică este prezentată în figura 4.62.

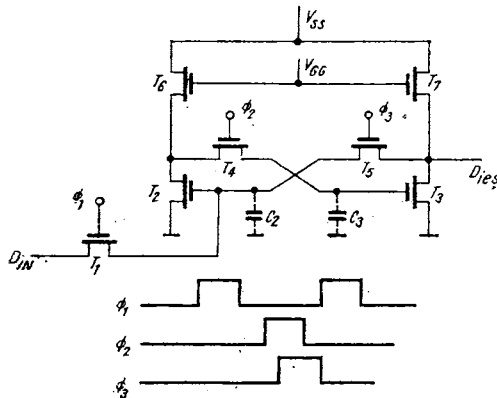


Fig. 4.62.

Când  $\phi_1 = 1$ , informația de la intrare ( $D_{IN}$ ) e transferată în  $C_2$  (capacitatea parazită de intrare a  $T_2$ ) determinându-se starea lui  $T_2$  ( $\overline{D}$ ). Când  $\phi_2 = 1$ ,  $T_4$  se deschide permițând încărcarea lui  $C_3$  și determinând starea lui  $T_3$  ( $\overline{\overline{D}} = D$ ). Când  $\phi_3 = 1$ , se deschide  $T_5$  închizând reacția de la ieșire la intrarea lui  $T_2$ . Astfel este memorată informația care a fost prezentă la  $D_{IN}$  și memorarea durează pînă la schimbarea informației la intrare și repetarea secvențelor de ceas.

**Problema 4.69.** Proiectați un registru de deplasare MOS dinamic de 1 bit și descrieți-i funcționarea prin diagrame de timp.

*Indicație*

Se inseriază mai multe inversoare MOS dinamice cu 3 sau 4 tranzistoare. Diagrama de timp pentru o celulă de deplasare vor fi cele din figura 4.63, dacă se ține seama că informația se transmite la ieșire cu frontul lui  $\phi_2$ .

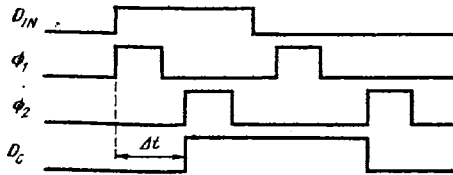


Fig. 4.63.

**Problema 4.70.** Se consideră registrul de deplasare dinamic de un bit din figura 4.64.

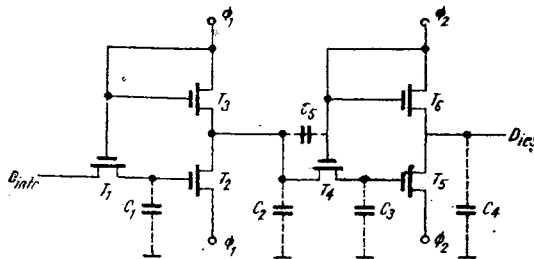


Fig. 4.64.

Se presupune că  $C_2$  și  $C_4$  se încarcă prin rezistențele dinamice  $r_d$  ale tranzistoarelor  $T_3$  și  $T_6$ . Intervalul între ceasuri  $t_2 - t_1 \gg Cr_d$ .

Calculați în funcție de frecvență :

a) eficiența registrului

b) puterea disipată.

*Rezolvare*

a) Tensiunea pe fiecare condensator va fi :

$$U_c(t) = U_{\varnothing}(1 - e^{-t/r_d C})$$

și căderea de tensiune pe  $r_d$  va fi

$$U_{r_d}(t) = U_{\varnothing} e^{-t/r_d C}$$

Puterea disipată pe fiecare tranzistor în timpul unui ciclu de ceas este

$$P_T = \frac{1}{r_d} \int_{t_1}^{t_2} U_{r_d}^2(t) dt = \frac{1}{r_d} \int_{t_1}^{t_2} U_{\varnothing}^2 e^{-2t/r_d C}$$

Cînd  $t_2 - t_1 \gg r_d C$

$$P_T \cong \frac{1}{2} U_{\varnothing}^2 C$$

Energia stocată în condensator în timpul unui ciclu de ceas este aceeași. Deci energia pierdută pe tranzistoare este egală cu cea stocată în condensatoare. Eficiența registrului va fi deci 50%.

b) Puterea disipată este  $\frac{1}{2} U_{\varnothing}^2 C \cdot f$ .

**Problema 4.71.** Pentru registrul dinamic din problema anterioară, stabiliți limitele de frecvență în care acesta operează corect.

*Rezolvare*

Sarcina lui  $C_2$  din prima celulă inversoare trebuie reținută în intervalul  $t_m$  dintre frontul negativ al lui  $\varnothing_1$  (fig. 4.65) și cel pozitiv al lui  $\varnothing_2$ . De asemenea trebuie reținută sarcina în  $C_4$  în intervalul  $t_s$ , de la frontul pozitiv al lui  $\varnothing_2$  pînă la cel pozitiv al lui  $\varnothing_1$ . Frecvența minimă de operare va fi

$$f_{min} = \frac{1}{t_m + t_s}$$

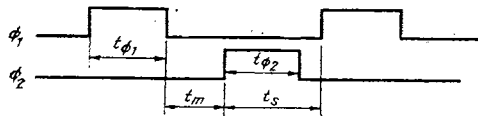


Fig. 4.65.

Funcția are un minim cînd  $t_m = t_s$ , deci cele 2 ceasuri sînt egal distribuite în timp.

Frecvența maximă de operație e determinată de constantele de timp de încărcare și descărcare a condensatoarelor care impun o valoare minimă pentru duratele lui  $\varnothing_1$  și  $\varnothing_2$ . Semnalele de ceas nu trebuie să se suprapună, adică  $t_m > 0$  (interval numit clock skew). În caz contrar datele se pot propaga prin mai multe celule în timpul suprapunerii, fenomenul nefiind controlabil. Notînd cu  $t_{\varnothing} = t_{\varnothing_1} = t_{\varnothing_2}$ , durata impulsurilor de ceas și cu  $t_f =$  durata fronturilor, rezultă:

$$f_{Max} = \frac{1}{2t_{\varnothing} + 2t_f + 2t_m}$$

De reținut că timpii de propagare și de comutare cresc cu temperatura

**Problema 4.72.** Realizați cu registre de deplasare MOS statice de 1024 de biți, lucrînd la maximum 3 MHz o memorie de 1024 × 8 comandată la 20 MHz.

**Rezolvare**

Pentru mărirea frecvenței de lucru a sistemului se comandă în paralel registrele MOS (fig. 4.66). Cu circuite rapide TTL se realizează o demultiplexare la intrare și o multiplexare la ieșire a datelor.

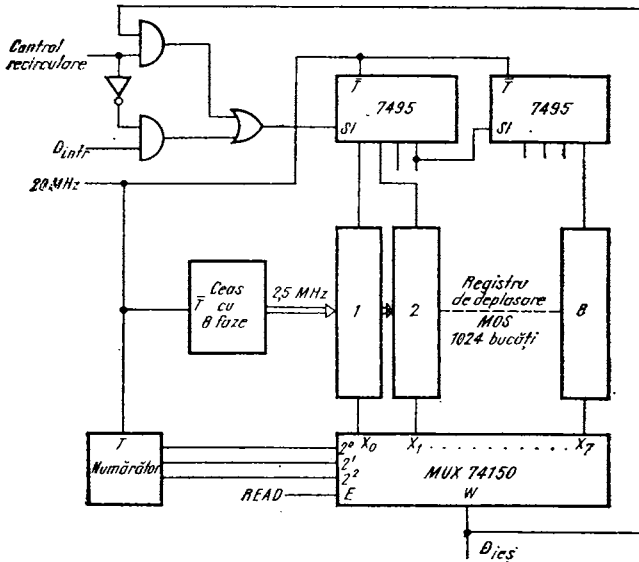


Fig. 4.66

**Problema 4.73.** Realizați o schemă de mărire de 4 ori a frecvenței de operare a 4 registre de deplasare MOS. Proiectați schema ceasului cu 4 faze necesar pentru comanda registrelor.

**Problema 4.74.** Să se proiecteze interfețele MOS-TTL și TTL-MOS pentru registrul de deplasare dinamic FDN 156 (fig. 4.67). Datele de catalog pentru acest circuit cu canal p sînt următoarele :

	<i>min</i>	<i>tip</i>	<i>max</i>	
$V_{\phi H} =$	-2	0	+0,3 V	} nivelele logice ale intrării de ceas $\phi$
$V_{\phi L} =$	-28	-12	-9 V	
$V_{IH} =$	-2	0	+0,3 V	} nivelele logice ale intrării de date I
$V_{IL} =$	-28	-12	-9 V	
$V_{P1} =$	-28	-26	-24 V	} tensiuni de alimentare
$V_{P2} =$	-28		+0,3 V	

Toate tensiunile sînt date în raport cu pinul  $P_0$

$$\pm I_{Qmax} = -I_{P2max} = 20 \text{ mA}$$

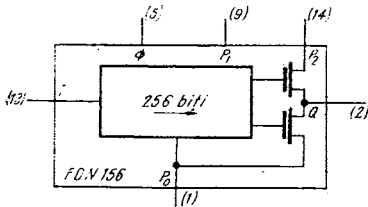


Fig. 4.67.

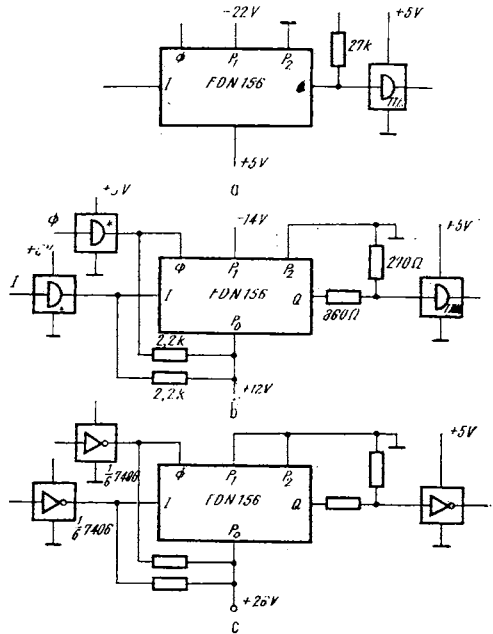


Fig. 4.68.

	<i>min</i>	<i>tip</i>	<i>max</i>	
$P_{OH}$	—	250	500	} impedanța văzută la ieșire în cele două stări logice
$R_{OL}$	—	250	500	

### Soluție

Există mai multe scheme posibile : vezi figura 4.68.

**Problema 4.75.** Comparați caracteristicile de transfer în tensiune ale unui inversor TTL și ale unui CMOS (MOS complementar).

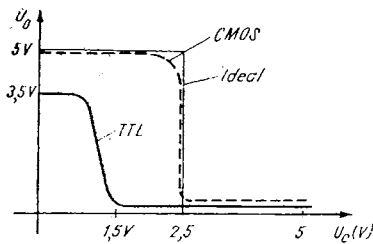


Fig. 4.69.

### Rezolvare

Din caracteristicile de transfer date în figura 4.69 se observă că inversorul CMOS este foarte aproape de cel ideal, caracterizat printr-un câștig infinit în zona de tranziție, și având valoarea tensiunii de tranziție, jumătate din tensiunea de alimentare (imunitate maximă la zgomote).

**Problema 4.76.** Comparați parametrii seriei 54/74 TTL Low power cu ai seriei 54/74 CMOS.

### Rezolvare

Familia	$V_{CC}$	$V_{IH\ max.}$ (V)	$I_{IL\ max.}$ (mA)	$V_{T1\ min.}$ (V)	$I_{T1\ min.}$ ( $\mu$ A)	$V_{OH\ max.}$ (V)	$I_{OL}$	$V_{OH\ min.}$ (V)	$I_{OH}$ ( $\mu$ A)	$t_{rLH}$ (ns)	$t_{rHL}$ (ns)	Poziția statică	$P_{poartă}$ , 1 MHz 50 pF (mW)
54/74	5	0,7	0,18	2,0	10	0,3	2 mA	2,4	100	35	31	1 mW	2,25
54/74 LP	5	0,8	—	3,5	—	0,4	360 $\mu$ A	2,4	100	45	60	10 nW	1,25
54/74 C	10	2,0	—	8,0	—	1,0	10 $\mu$ A	9,0	10	30	25	30 nW	5

**Problema 4.77.** Se consideră inversorul CMOS din figura 4.70. Se știe că :  $T_1$  suportă un curent de drenă  $I_{D1} = 1,5$  mA iar  $T_2$ ,  $I_{D2} =$  mA, tensiunea de prag a circuitului este  $U_p = 3$  V, capacitatea de ieșire a circuitului este de 20 pF.

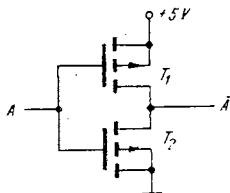


Fig. 4.70.

Să se calculeze timpii de propagare pentru ambele tranziții dacă inversorul este încărcat în sarcină cu capacitățile de intrare a 10 circuite, fiecare de 5 pF, iar capacitatea firelor este de 30 pF.

*Rezolvare*

Capacitatea totală din sarcină este :

$$C = 20 \text{ pF} + 10 \cdot 5 \text{ pF} + 30 \text{ pF} = 100 \text{ pF}$$

Pentru tranziția  $L \rightarrow H$ , se deschide  $T_1$ ,  $T_2$  blocându-se. Nivelul  $H$  (de 3 V) se atinge în timpul

$$t_{PLH} \cong \frac{CU_p}{I_{D1}} = 200 \text{ ns}$$

Pentru tranziția  $H \rightarrow L$ ,  $T_1$  se blochează, deschizându-se  $T_2$ .  
Rezultă :

$$t_{PHL} \cong \frac{C(5 \text{ V} - U_p)}{I_{D2}} = 200 \text{ ns}$$

**Problema 4.78.** a) Cum se calculează consumul de putere static și dinamic pentru o poartă CMOS ?

b) Prezentați metodele de reducere a consumului de putere din sursă.

*Rezolvare*

a) Puterea consumată dinamic este cea necesară pentru comanda capacității din sarcină și este :

$$P_d = C \cdot V_{CC}^2 \cdot f \text{ (vezi problema 4.70)}$$

Puterea consumată „static“ este cea disipată în timpul comutării (fig. 4.71).

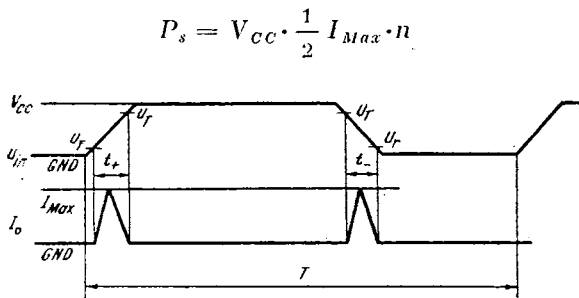


Fig. 4.71.

unde :

$$n = \frac{V_{CC} - 2V_T}{V_{CC}} \cdot \frac{t_r + t_f}{T}$$

Deci :

$$P_s = \frac{I_{Max}}{2} (V_{CC} - 2V_T) \cdot (t_+ + t_-) \cdot f$$

$f = \frac{1}{T}$  fiind frecvența de repetiție a semnalului.

b) Ambele puteri scad cu reducerea sursei de alimentare,  $V_{CC}$  și a frecvenței de lucru  $f$ . Capacitatea de sarcină trebuie să fie mică, la fel și duratele fronturilor  $t_+$  și  $t_-$ .

**Problema 4.79.** Care este schema internă a unei porți NAND cu 2 intrări realizată cu CMOS ?

*Soluție*

Vezi figura 4.72.

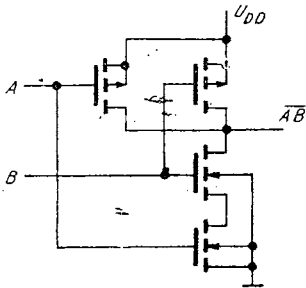


Fig. 4.72.

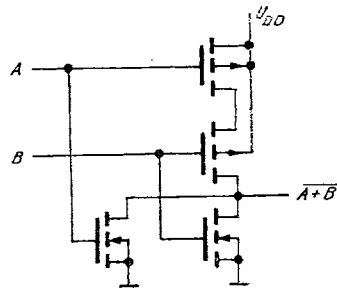


Fig. 4.73.

**Problema 4.80.** Care este schema internă a unei porți NOR cu 2 intrări CMOS ?

*Soluție*

Vezi figura 4.73.

**Problema 4.81.** Explicați rolul rezistenței și diodelor din circuitul de intrare al unui inversor CMOS (vezi fig. 4.74).

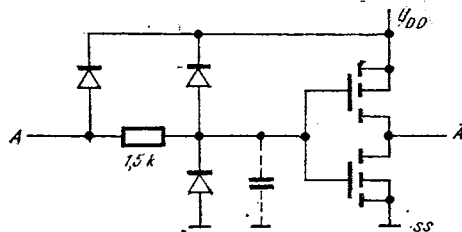


Fig. 4.74.



### Rezolvare

Tensiunea de străpungere a stratului de oxid, de cca 90 V, poate fi atinsă foarte ușor prin acumularea de sarcini statice (impedanța de intrare este mai mare de  $10^9 \Omega$ ). Diodele și rezistența sînt adăugate pentru a împiedica distrugerea circuitului prin acumularea acestor sarcini, prin depășirea de către tensiunea de intrare a  $\pm 30$  V. Curentul de intrare este limitat la cca 10 mA maximum.

**Problema 4.82.** Dacă un circuit CMOS este conectat, la intrare pot apare tensiuni mai mari decît  $U_{DD}$  sau mai mici decît  $U_{SS}$  (negative). Este necesară o rezistență serie la intrare,  $R$ , pentru a limita curentul la 10 mA. Să se dimensioneze această rezistență.

### Rezolvare

Presupunem nivelele maxime de intrare:  $U_i^+ > U_{DD}$  și  $U_i^- < 0$ .

$$\text{În acest caz } R_1 \geq \frac{U_i^+ - U_{DD} - 0,7 \text{ V}}{I_{Max}} = 1,5 \text{ K}$$

$$R_1 \geq \frac{U_i^- - 0,7 \text{ V}}{I_{Max}} = 1,5 \text{ K}$$

Se alege valoarea care este acoperitoare în fiecare caz concret. Se indică adăugarea unei rezistențe către masă pentru a elimina eventualele sarcini statice induse.

**Problema 4.83.** Să se interfațeze circuite CMOS cu circuite pMOS și nMOS.

### Rezolvare

#### Interfațare cu pMOS.

Deoarece tensiunea maximă de alimentare pentru CMOS este de 15V, interfațarea cu pMOS lucrînd pînă la aceste tensiuni se face direct. Pentru pMOS alimentat la mai mult de 15 V soluția este dată în figura 4.75.

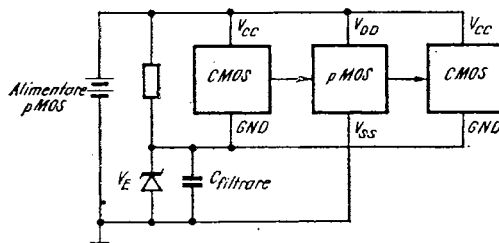


Fig. 4.75.

Din sursa de alimentare a pMOS se formează cu ajutorul diodei Zener o tensiune de valoare mai mică pentru alimentarea circuitelor CMOS.

În cazul când există și circuite TTL, cele CMOS alimentându-se la aceeași sursă de 5 V, soluția este cea din figura 4.76.

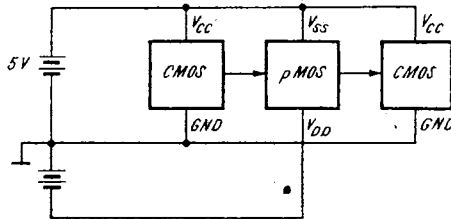


Fig. 4.76.

Interfașarea cu nMOS se poate face direct deoarece aceste circuite necesită tensiuni de alimentare mai mici (5–12 V), fiind direct compatibile cu CMOS (fig. 4.77).

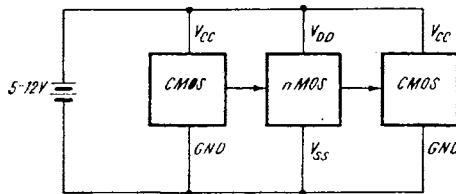


Fig. 4.77.

**Problema 4.84.** Rezolvați problema cuplării CMOS–TTL.

*Rezolvare*

*Cuplarea TTL — CMOS.*

Problema care se pune în acest caz este dacă nivelul H din TTL este suficient pentru comanda CMOS. Când este cazul, pentru ridicarea nivelului H se montează o rezistență (pull up resistor) către sursă (fig. 4.78).

*Cuplarea CMOS — TTL* presupune ca parametru critic, asigurarea de către ieșirea CMOS a curentului de intrare maxim (în starea 0) a TTL.

Curentul de intrare pentru seria de mică putere (LTTL) este suficient de mic (180  $\mu$ A) încât din CMOS se pot comanda două astfel de cir-

cuite. În ceea ce privește seria TTL standard, aceasta avînd curentul de intrare de 1,6 mA nu poate fi comandată direct din CMOS (există și unele excepții — vezi problema 4.85).

**Problema 4.85.** Arătați cu care din circuitele NOR sau NAND CMOS se poate comanda o poartă TTL standard. Explicați cum e posibil acest fapt.

*Rezolvare*

Tipic, circuitele CMOS dau la ieșire pentru 0,4 V, 360  $\mu$ A (chiar 420  $\mu$ A la 25°C). Dar, în timp ce la poarta NAND 74COO, curentul în starea L e furnizat de 2 tranzistoare în serie (vezi problema 4.79), la poarta NOR 74C02 (problema 4.80) curentul trece prin 2 tranzistoare în paralel. Aceste 2 tranzistoare pot suporta același curent de virf ca și cele serie de la NAND; deci poarta NOR poate da la ieșire în starea L curent dublu față de cea NAND. Dacă se admite și o creștere la 0,8 V a potențialului de ieșire în starea L, se poate conta pe o creștere de încă 2 ori a curentului de ieșire. Deci la 25°C, și  $U_{OL} = 0,8$  V, poarta NOR 74C02 poate furniza

$$I_{OL} = 4 \times 420 \mu\text{A} = 1,68 \text{ mA}$$

curent suficient pentru o intrare TTL standard.

Se pierde însă din marginea de zgomot; pierderea nu este chiar de 100% dacă ținem cont că majoritatea intrărilor TTL necesită un curent de 1 mA, 1,6 mA fiind curentul maxim absorbit, în cazul cel mai defavorabil.

Din cele prezentate rezultă că circuitul NOR 74C02 poate fi utilizat ca tampon de cuplare CMOS—TTL standard.

**Problema 4.86.** Pentru oscilatorul cu 3 inversoare CMOS din figura 4.79 să se dimensioneze  $R_1$ ,  $R_2$  și C pentru a obține o frecvență a oscilațiilor de 7 MHz.

*Indicație*

$$f \cong \frac{1}{2 C(0,405 R_e + 0,693 R_1)} \quad \text{unde } R_e = \frac{R_1 R_2}{R_1 + R_2}$$

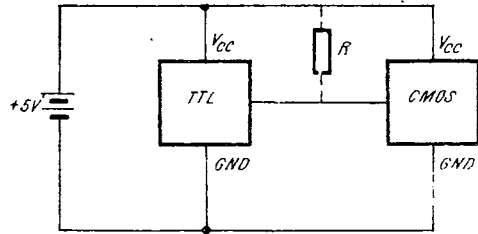


Fig. 4.78.

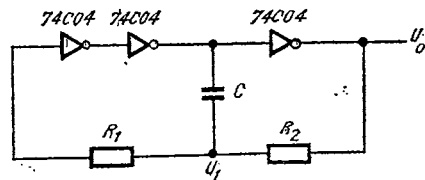


Fig. 4.79.

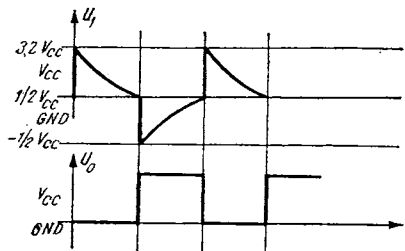


Fig. 4.80.

$$\text{Dacă } R_1 = R_2 = R; f \cong \frac{0,559}{RC}$$

$$R_2 \gg R_1, f = \frac{0,455}{RC}$$

$$R_2 \ll R_1, f = \frac{0,722}{RC}$$

Formule de undă în  $U_1$  și la ieșire  $U_0$  sînt prezentate în figura 4.80.

**Problema 4.87.** Desenați schema internă a unui SAU EXCLUSIV (XOR) cu CMOS.

**Problema 4.88.** Care este tabelul de adevăr al multiplexorului CMOS 74C151? Se deosebește acesta de cel al MUX 74151 realizat în tehnologie TTL?

**Problema 4.89.** Cîte circuite din seria TTLP pot comanda 3 inversoare CMOS? Dar din seria standard?

**Problema 4.90.** Cum variază marginea de zgomot cu  $V_{cc}$  la CI CMOS.

*Soluție*

Vezi figura 4.81.

Marginile de zgomot pentru  $L$  și  $H$  sînt de cea 1 V, indiferent de  $V_{cc}$ . (Deci sînt mai mari ca cele TTL, de numai 0,4 V).

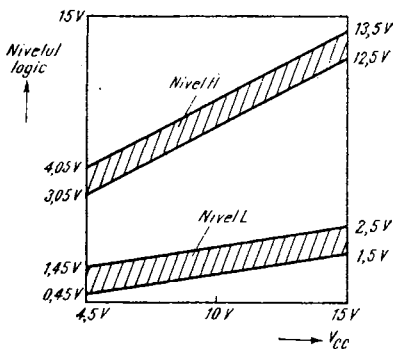


Fig. 4.81.

### Prezentarea conținutului

În realizarea practică a sistemelor cu CI trebuie să se țină cont și de modul de conectare între acestea. În cazul unor conexiuni prea lungi sau incorect plasate, pot apărea semnale parazite datorate reflexiilor, diafoniilor care perturbă funcționarea sistemului.

Unul din scopurile prezentului capitol este de a evidenția câteva aspecte privind calculul parametrilor liniilor de transmisiune constituite de cablurile, traseele pe circuite imprimate, care conectează între ele CI ale sistemelor: impedanța caracteristică, viteza de propagare, capacitatea, întârzierea (problemele 5.1—5.7).

Ca urmare a reflexiilor pe aceste linii apar modificări ale formelor de undă. În problemele 5.8—5.11 sînt deduse formule pentru calculul acestor modificări. Într-o proiectare corectă este necesar ca acestea să devină neesențiale într-un interval mai mic decît timpul de propagare al circuitelor, pentru a nu le perturba funcționarea (problemele 5.17—5.19). S-a arătat anterior (problema 4.29) că circuitele integrate TTL oscilează dacă punem pe intrări semnale de amplitudine între 0,8—2 V. De aceea, ca receptoare de linii se utilizează în general circuite cu cîștu de histerezis la intrare.

Prin măsurări asupra semnalelor la generator sau receptorul de linie, se pot calcula parametrii diverselor configurații de trasee pe circuitele imprimate. Cunoașterea impedanței caracteristice are o importanță deosebită deoarece se urmărește proiectarea unor linii cît mai aproape de adaptare (cu impedanța de sarcină egală cu cea caracteristică a liniei) în scopul reducerii maxime a reflexiilor.

În încheierea acestui capitol sînt tratate cîteva aspecte privind modalitățile în care se fac măsurările în sistemele digitale : determinarea benzii osciloscopului pentru a putea măsura fronturile semnalelor, modul cum se fac corecții ale determinărilor și adaptarea sondei pentru a nu deforma impulsurile.

**Problema 5.1.** Pe un cablu coaxial de lungime  $l = 60$  cm se propagă un semnal cu viteza  $\frac{2}{3}$  din cea a luminii (este vorba de un cablu obișnuit cu dielectric solid). Determinați întârzierea  $T$  a cablului în nanosecunde.

Viteza luminii este  $C = 30$  cm/ns. •

*Rezolvare*

$$T = \frac{l}{v} = \frac{60 \text{ cm}}{\frac{2}{3} \cdot 30 \text{ cm/ns}} = 3 \text{ ns}$$

**Problema 5.2.** Determinați capacitatea  $C$  a cablului din problema precedentă dacă impedanța sa caracteristică este  $Z_0 = 50 \Omega$ .

*Rezolvare*

$$C = \frac{T}{Z_0} = \frac{3 \text{ ns}}{50 \Omega} = 60 \text{ pF}$$

**Problema 5.3.** Care sînt formulele de calcul pentru impedanța caracteristică a unor linii de transmisiune în următoarele situații :

a) un singur fir cu diametrul  $D$  la înălțimea  $H$  deasupra unui plan de masă infinit ?

b) două fire paralele de diametre  $D$ , aflate la distanța  $S$  unul de celălalt ?

c) două trasee paralele de lățime  $L$ , separate printr-un dielectric de grosime  $S$  (cazul traseelor paralele situate pe două fete ale unui circuit imprimat) ?

d) două trasee paralele de lățime  $L$ , situate la distanța  $S$ , pe aceeași față a unui circuit imprimat ?

e) cablu coaxial cu diametrul cablului interior  $D_1$  și diametrul cablului exterior  $D_2$  ?

*Soluție*

Formulele din figura 5.1 dau pe  $Z_0$  dacă liniile sînt în aer liber sau vid. Într-un mediu cu constanta dielectrică relativă  $\epsilon_r$ , impedanțele caracteristice se reduc cu:

$$Z'_0 = \frac{Z_0}{\sqrt{\epsilon_r}}$$

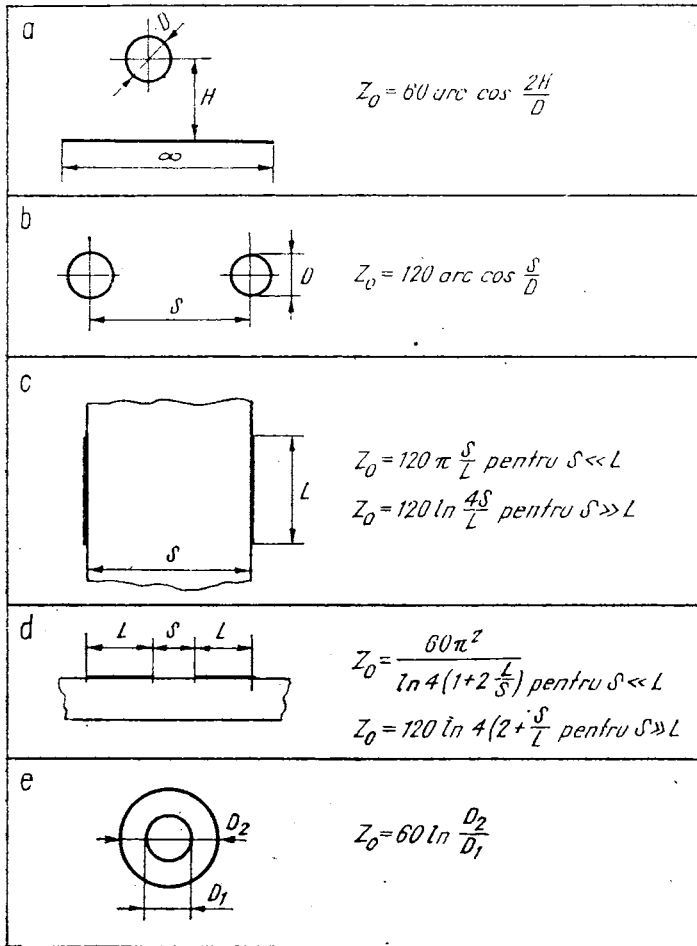


Fig. 5.1.

**Problema 5.4.** a) Să se calculeze impedanța caracteristică a unui cablu coaxial cu diametrul interior de 0,33 mm și cel exterior de 0,76 mm.

b) Ce capacitate are acest cablu dacă întârzierea pe care o produce este de 7 ns ?

c) Știind că viteza de propagare prin cablu este de 20 cm/ns să se determine lungimea acestuia.

*Soluție*

a) 50 Ω ; b) 140 pF ; c) 140 cm.

**Problema 5.5.** Ce impedanță caracteristică are linia de transmisie formată din 2 conductori paraleli de lățime 0,5 cm situați la 3 cm unul de altul pe aceeași față a unei plăci de sticlătextolit eposi ( $\epsilon_r = 4,7$ ) ?

*Rezolvare*

$$Z_0 = \frac{120 \pi}{\sqrt{\epsilon_r}} \ln 4 \left( 2 + \frac{S}{L} \right) = \frac{120 \pi}{\sqrt{4,7}} \ln 4 \left( 2 + \frac{3}{0,5} \right) = 192 \ \Omega$$

**Problema 5.6.** Să se determine impedanța caracteristică a unei linii formate din 2 trasee paralele de 1,5 cm lățime, distanțate la 0,3 cm, pe aceeași față a unei plăci de circuit imprimat de  $\epsilon_r = 2$ .

*Soluție*

$$Z_0 = 1,58 \ \text{K}$$

**Problema 5.7.** Determinați impedanța caracteristică a unei linii de transmisie formată din 2 conductori cablați paraleli de 1 cm lățime separați printr-un strat de 0,2 cm, cu constanta dielectrică  $\epsilon_r = 2$ .

*Rezolvare*

$$Z_0 = \frac{120 \pi S}{\sqrt{\epsilon_r} L} = 53 \ \Omega$$

**Problema 5.8.** Exprimați tensiunea pe rezistența cuplată în capătul unei linii de transmisie de impedanță  $Z_0$ , dacă se aplică la intrare un salt de tensiune  $U_i$ . Rezistența internă a generatorului care debitează pe linie se consideră nulă.

*Rezolvare*

Sensurile curenților și tensiunilor sînt figurate în 5.2.

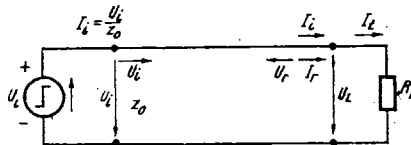


Fig. 5.2.

Inițial pe linie se trimite din generator tensiunea  $U_i$  și curentul  $I_i = U_i / Z_0$ . Când unda generată ajunge la  $R_L$ , o parte din semnal se reflectă, mergînd înapoi spre generator. Aplicînd teorema Kirchoff :

$$I_L = I_i + I_r \quad (1)$$

Deoarece în momentul sosirii undei incidente la sfîrșitul liniei trebuie să existe doar o singură tensiune, avem relația :

$$U_L = U_r + U_i \quad (2)$$



Dar

$$I_L = \frac{U_L}{R_L}, \quad I_i = \frac{U_r}{Z_0}; \quad I_r = -\frac{U_r}{Z_0} \quad (3)$$

unde semnul minus arată că  $U_r$  merge spre generator.

Înlocuind (3) în (1) rezultă :

$$\frac{U_L}{R_L} = \frac{U_i - U_r}{Z_0} \quad (4)$$

Înlocuind (2) în (4) avem :

$$\frac{U_r + U_i}{R_L} = \frac{U_i - U_r}{Z_0}$$

de unde :

$$U_r = \left( \frac{R_L - Z_0}{R_L + Z_0} \right) U_i = \rho_L U_i$$

$\rho_L$  se numește coeficient de reflexie. Pentru  $R_L$  variind între 0 (linie în scurtcircuit) și  $\infty$  (linie deschisă),  $\rho_L$  variază între  $-1$  și  $1$ .

Indicele  $L$  arată că acest coeficient se referă la sarcina (Load) de la sfârșitul liniei.

Odată calculată tensiunea de reflexie, rezultă

$$U_L = (1 + \rho_L) U_i = 2U_i \frac{R_L}{R_L + Z_0}$$

Tensiunea pe rezistența  $R_L$  are aceeași formă ca cea obținută pe divizorul format de  $Z_0$  și  $R_L$  cu deosebirea că tensiunea de intrare apare dublată.

**Problema 5.9.** Să se calculeze coeficientul de reflexie, tensiunea reflectată, tensiunea în sarcină pentru o linie de  $50 \Omega$ , comandată cu generator de tensiune de  $5 \text{ V}$  ideal și avînd ca terminație :

- a) scurtcircuit
- b)  $R_L = 10 \Omega$
- c)  $R_L = 50 \Omega$
- d)  $R_L = 100 \Omega$
- e)  $R_L = \infty$  (linie deschisă).

Să se reprezinte modul cum variază tensiunea reflectată în cele 5 situații.

Timpul de propagare pe linie este  $T$ .

Rezolvare

a)  $\rho_L = -1$ ,  $U_r = -U_i = -5 \text{ V}$ ,  $U_L = 0$

Schema și variația tensiunii pe linie se dau în figura 5.3. La momentul 0 se reflectă cei 5 V de la generator, la  $T$  se reflectă  $-5 \text{ V}$ .

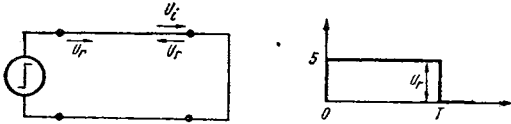


Fig. 5.3.

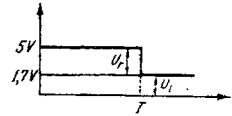


Fig. 5.4.

b)  $\rho_L = \frac{10 - 50}{10 + 50} = -0,66$

$U_r = \rho_L U_i = -0,66 \cdot 5 = -3,3 \text{ V}$

$U_L = (1 + \rho_L) U_i = (1 - 0,66) \cdot 5 = 1,7 \text{ V}$

Forma de undă pentru  $U_r$  este dată în figura 5.4.

c)  $\rho_L = 0$

$U_r = 0$

$U_L = U_i = 5 \text{ V}$ .

În cazul adaptării,  $R_L = Z_0$ , nu există reflexii, întreaga tensiune a generatorului se transmite în sarcină.

d)  $\rho_L = \frac{100 - 50}{100 + 50} = 0,33$

$U_r = \rho_L \cdot U_i = 0,33 \cdot 5 \text{ V} = 1,65 \text{ V}$

$U_L = (1 + \rho_L) U_i = 1,33 \cdot 5 \text{ V} = 6,65 \text{ V}$ .

În figura 5.5 este reprezentată tensiunea reflectată.

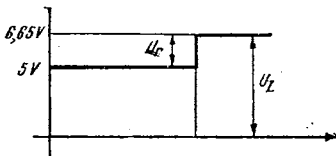


Fig. 5.5.

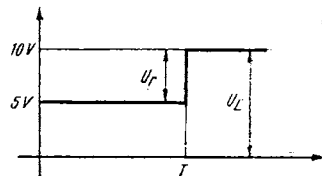


Fig. 5.6.

e)  $\rho_L = 1$

$$U_r = \rho_L U_i = 5 \text{ V (vezi figura 5.6)}$$

$$U_L = (1 + \rho_L) U_i = 2 U_i = 10 \text{ V.}$$

**Problema 5.10.** Să se reia problema 5.8 considerînd rezistența generatorului  $R_g$  diferită de zero.

*Rezolvare*

Se pot utiliza formulele deduse în rezolvarea problemei 5.8 remarcînd că tensiunea care apare inițial pe linie nu este  $U_i$ , ci  $U_g$  (figura 5.7). Se înlocuiește  $U_i$  din formulele determinate

cu  $U_i = \frac{Z_0}{R_g + Z_0} \cdot$

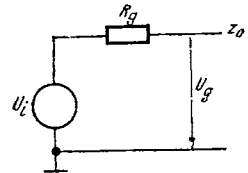


Fig. 5.7.

Deci :

$$U_{rL} = \rho_L \frac{Z_0}{R_g + Z_0} U_i$$

$$U_L = (1 + \rho_L) \frac{Z_0}{R_g + Z_0} U_i$$

La momentul de timp  $2T$ , unda reflectată se întoarce la generator, reflectîndu-se din nou. Coeficientul de reflexie va fi :

$$\rho_g = \frac{R_g - Z_0}{R_g + Z_0}$$

Se va reflecta tensiunea

$$U_{rg} = \rho_g U_{rL}$$

și va rămîne pe  $R_g$  tensiunea

$$U_g = (1 + \rho_g) U_{rL}$$

**Problema 5.11.** Să se calculeze și să se deseneze formele de undă în intervalul a  $9T$ , la generator și în sarcină pentru o linie cu  $Z_0 = 93 \Omega$ ,  $R_g = 31 \Omega$ ,  $R_L = \infty$  la care se aplică un salt treaptă de 1 V.

*Rezolvare*

Se calculează :

$$\rho_L = \frac{R_L - Z_0}{R_L + Z_0} = 1, \quad \rho_g = \frac{R_g - Z_0}{R_g + Z_0} = -0,5$$

Pentru calculul tensiunilor se vor utiliza formulele :

*Pentru sarcină*

— tensiunea pe rezistența de sarcină la momentul  $t$  are o variație  $U_L^t$  dependentă de tensiunea transmisă pe linie la momentul de timp anterior,  $U_{rg}^{t-1}$ :

$$U_L^t = (1 + \rho_L) U_{rg}^{t-T} = 2 U_{rg}^{t-T}$$

$T$  fiind întârzierea liniei.

— tensiunea reflectată de sarcină este :

$$U_{rL}^t = \rho_L U_{rg}^{t-T} = U_{rg}^{t-T}$$

*Pentru generator*

— variația de tensiune la generator,  $U_g^{t+T}$ , la momentul  $(t+1)$  depinde de coeficientul de reflexie la generator,  $\rho_g$  și de tensiunea reflectată de sarcina  $U_{rL}^t$  la momentul de timp anterior :

$$U_g^{t+T} = (1 + \rho_g) U_{rL}^t = 0,5 U_{rL}^t$$

— tensiunea reflectată de rezistența generatorului este :

$$U_{rg}^{t+T} = \rho_g U_{rL}^t = -0,5 U_{rL}^t$$

Cu aceste formule se calculează :

*La momentul  $t = 0$*

Tensiunea transmisă pe linie de generator va fi :

$$U_{rg}^0 = \frac{Z_0}{R_g + Z_0} U_g = 0,75 \text{ V}$$

*La momentul  $t = T$*

Saltul  $U_{rg}^0$  s-a propagat pînă la rezistența de sarcină pe care apare saltul de tensiune :

$$U_L^T = 2 U_{rg}^0 = 1,5 \text{ V.}$$

Se reflectă înapoi pe linie, tensiunea :

$$U_{rL}^t = U_{rg}^0 = 0,75 \text{ V}$$

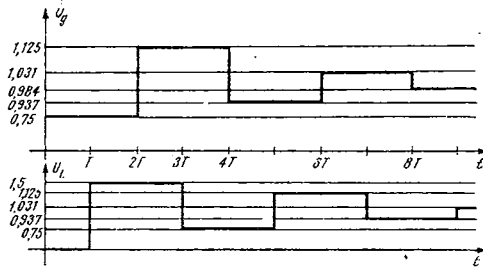


Fig. 5.8.

La momentul  $t = 2T$

Semnalul reflectat anterior,  $U_{rL}^T$  se propagă la generator unde apare saltul:

$$U_g^{2T} = 0,5 \quad U_{rL}^T = 0,375 \text{ V}$$

Se reflectă la generator tensiunea:

$$U_{rg}^{2T} = -0,5 \quad U_{rL}^T = -0,375 \text{ V}$$

La momentul  $3T$

În sarcină apare variația de tensiune:

$$U_L^{3T} = 2 \quad U_{rg}^{2T} = -0,75 \text{ V}$$

Se va reflecta către generator:

$$U_{rL}^{3T} = U_{rg}^{2T} = -0,375 \text{ V}$$

La momentul  $4T$

$$U_g^{4T} = 0,5 \quad U_{rL}^{3T} = -0,1875 \text{ V}$$

$$U_{rg}^{4T} = -0,5 \quad U_{rL}^{3T} = 0,1875 \text{ V}$$

La momentul  $5T$

$$U_L^{5T} = 2 \quad U_{rg}^{4T} = 0,375 \text{ V}$$

$$U_{rL}^{5T} = U_{rg}^{4T} = 0,1875 \text{ V}$$

Calculul se desfășoară în continuare în mod similar. În figura 5.8 sînt prezentate formele de undă la bornele generatorului și pe rezistența de sarcină.

**Problema 5.12.** Cu un generator de impulsuri cu rezistență internă de  $50 \Omega$  se comandă un cablu coaxial de 2 m. La celălalt capăt al ca-

blului apare după 1 ns de la frontul pozitiv al generatorului o treaptă de 15 V. Știind că generatorul dă în gol impulsuri de 5 V să se calculeze :

- viteza de propagare a semnalelor pe cablu ;
- impedanța caracteristică a cablului ;
- capacitatea cablului ;
- diametrul cablului interior știind că diametrul exterior este de 5 mm.

*Rezolvare*

$$a) v = \frac{L}{T} = \frac{2 \text{ m}}{10 \text{ ns}} = 20 \text{ cm/ns}$$

b) Se calculează  $\rho_L = 1$ .

Pentru primul salt care apare în sarcină (la momentul  $T$ ) avem :

$$U_L^T = \rho_L U_i = \frac{Z_0}{Z_0 + R_g} U_g$$

Din această expresie se deduce :

$$Z_0 = \frac{U_L^T}{U_g - U_L^T} R_g$$

Înlocuind cu datele problemei se obține  $Z_0 = 75 \Omega$ .

$$c) C = \frac{T}{Z_0} = \frac{10 \text{ ns}}{75} = 133 \text{ pF}$$

d) Se utilizează formula din problema 5.3 e) :

$$75 = 60 \ln \frac{5}{D_1}$$

din care rezultă  $D_1 = 1,4 \text{ mm}$ .

**Problema 5.13.** a) Să se determine impedanța caracteristică a unui cablu coaxial de 16,8 m care, fiind conectat cu un capăt la un generator de impulsuri cu impedanța de 50 produce o scădere a amplitudinii impulsului de la 5 V la 3 V. (Celălalt capăt al cablului este în gol).

b) Care este viteza de propagare a semnalului pe cablu dacă treapta de 3 V are o durată de 168 ns ?

c) Să se calculeze amplitudinile tensiunii la generator la momentele  $2T$ ,  $3T$  și  $4T$ .

*Rezolvare*

a) Prin conectarea cablului la generator se produce divizarea impulsului pe rezistența internă a generatorului și pe  $Z_0$ . Deci :

$$Z_0 = \frac{U_i}{U_{g01} - U_i} \cdot R_S = \frac{3}{5 - 3} \cdot 50 = 75 \Omega$$

b) Treapta de 3 V durează  $2T$ ; timpul necesar undei reflectate să se întoarcă la generator. Deci  $T = 84 \text{ ns}$ . De unde :  $v = \frac{L}{T} = \frac{16,8 \text{ m}}{84 \text{ ns}} = 20 \text{ cm/ns}$ .

c) Se calculează :

$$\rho_L = 1; \quad \rho_g = \frac{50 - 75}{50 + 75} = -0,2$$

La momentul  $T$ , în sarcină apare tensiunea

$$U_L = 2U_i = 6 \text{ V}$$

și se reflectă pe linie

$$U_{rL} = U_i = 3 \text{ V}$$

La momentul  $2T$ , la generator are loc variația de tensiune :

$$U_g^{2T} = (1 + \rho_g)U_{rL} = 2,4 \text{ V}$$

care se adaugă la  $U_g$ . Deci amplitudinea tensiunii la generator la  $2T$  va fi 5,4 V.

Tensiunea reflectată va fi :

$$U_{rg} = \rho_g U_{rL} = -0,6 \text{ V}$$

La momentul  $3T$  din sarcină se reflectă tensiunea

$$U_{rL}^{3T} = 2U_{rg} = -1,2 \text{ V}$$

dar la generator se mențin cei 5,4 V.

La momentul  $4T$ , tensiunea reflectată ajunge la generator producînd o variație :

$$U_g^{4T} = -0,96 \text{ V}$$

Deci amplitudinea scade la 4,44 V.

**Problema 5.14.** Să se deducă expresii generale de calcul pentru variațiile de tensiune în sarcină, la generator și tensiunile reflectate pe liniile de transmisiune în funcție de  $\rho_L$ ,  $\rho_g$  și tensiunea aplicată inițial pe linie,  $U_i$ .

*Rezolvare*

Se exprimă variațiile de tensiune pe linie în funcție de  $U_i$  la diversele momente de timp,  $T$  fiind durată de propagare pe linie.

La momentul  $T$

— în sarcină :  $U_L^T = (1 + \rho_L)U_i$

— se reflectă :  $U_{rL} = \rho_L U_i$

La momentul  $2T$

— la generator :  $U_g^{2T} = (1 + \rho_g)U_{rL}^T = \rho_L(1 + \rho_g)U_i$

— se reflectă :  $U_{rL}^{2T} = \rho_g U_{rL}^T = \rho_g \rho_L U_i$

La momentul  $3T$

— în sarcină :  $U_L^{3T} = (1 + \rho_L)U_{rL}^{2T} = \rho_L \rho_g (1 + \rho_L)U_i$

— se reflectă :  $U_{rL}^{3T} = \rho_L U_{rL}^{2T} = \rho_g \rho_L^2 U_i$

La momentul  $4T$

— la generator :  $U_g^{4T} = (1 + \rho_g)U_{rL}^{3T} = \rho_g \rho_L^2 (1 + \rho_g)U_i$

— se reflectă :  $U_{rL}^{4T} = \rho_g U_{rL}^{3T} = \rho_g^2 \rho_L^2 U_i$

Generalizînd se obține :

La momentul  $(2n + 1)T$  — pentru  $n \geq 0$

— în sarcină :  $U_L(2n + 1)T = \rho_g^n \rho_L^n (1 + \rho_L)U_i$

— se reflectă :  $U_{rL}^{(2n+1)T} = \rho_g^n \rho_L^{n+1} U_i$

La momentul  $2nT$  — pentru  $n - 1$

— la generator  $U_g^{2nT} = \rho_g^{n-1} \rho_L^n (1 + \rho_g)U_i$

Se reflectă :  $U_{rL}^{2nT} = \rho_g^n \rho_L^n U_i$

Cu aceste expresii generale se pot calcula formele de undă în sarcină și la ieșirea generatorului, tensiunile reflectate pe linie.

Tensiunea  $U_i$  care intervine în calcul este tensiunea dată de generatorul de semnal în gol,  $U_g$ , divizată pe rezistența sa internă și pe  $Z_0$  :

$$U_i = \frac{Z_0}{Z_0 + R_g} U_g$$

**Problema 5.15.** Să se calculeze și să se deseneze formele de undă la începutul și sfîrșitul unei linii de transmisiune, pe intervalul a  $6T$ ,  $T$  fiind durata de propagare pe linie, dacă se dau :

—  $Z_0 = 150 \Omega$

—  $R_g = 130 \Omega$

—  $R_L = 500 \Omega$

Să se determine marginea de zgomot a circuitului avînd rezistența de intrare  $R_L$ .

**Problema 5.16.** Să se demonstreze prin metoda grafică faptul că dioda de „clampare“ montată la intrarea CI TTL reduce reflexiile pe linie.

*Rezolvare*

Considerăm o linie de  $150 \Omega$  care leagă ieșirea de intrarea unei porți TTL fără diodă de „clampare“. Scările se aleg astfel încît  $Z_0$  să se traseze la  $45^\circ$  pentru o construcție mai ușoară (figura 5.9).



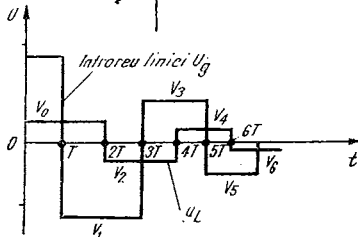
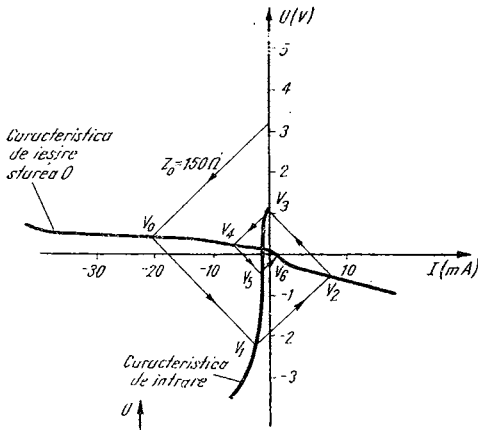


Fig. 5.9.

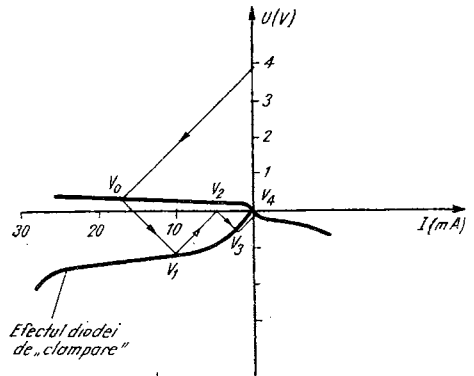


Fig. 5.10.

Pentru aceeași linie, dacă se cuplează dioda de „clampare“ apar modificările din figura 5.10.

Se observă că oscilațiile sînt mai mici și durează mai puțin.

O explicație posibilă este că, pentru tensiuni negative, impedanța de intrare mare în CI e redusă de dioda deschisă și linia e mai aproape de adaptare.

**Problema 5.17.** Să se calculeze lungimea maximă a liniilor de conexiune dintre 2 porți TTL standard, HTTL, LTTL, TTLS, ECL, CMOS presupunînd următoarele :

- viteza de propagare a semnalului este de 17,6 cm/ns (tipică pentru cablaj pe sticlotextolit epoxi cu  $\epsilon_r = 4,7$ ).
- pe intervalul de propagare  $t_p$  trebuie să aibă loc cel puțin 5 reflexii.
- timpii medii de propagare se vor considera :

	$t_{PLH}$	$t_{PHL}$
TTL	20 ns	10 ns
HTTL	8 ns	6 ns

LTTL	60 ns	30 ns
STTL	5 ns	5 ns
ECL	2 ns	2 ns
CMOS	50 ns	50 ns

*Indicație*

Timpul de propagare pe linie pentru TTL standard trebuie să fie de 5 ori mai mic decât timpul de propagare prin circuit :

$$T = \frac{t_{pNL}}{5} = 2 \text{ ns}$$

Rezultă lungimea liniei

$$l = v \cdot T = 17,6 \text{ cm/ns} \cdot 2 \text{ ns} \cong 35 \text{ cm}.$$

Dimensionarea se face evident pentru intervalul minim de propagare.

În enunțul problemei s-a presupus că după 5 reflexii semnalul ajunge sub marginea de zgomot a circuitelor.

**Problema 5.18.** Să se calculeze lungimea firului de conexiune pentru comanda de ceas dintre cei 2 bistabili din figura 5.11.

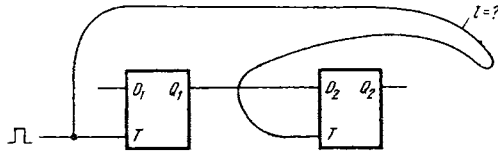


Fig. 5.11.

cunoscînd că :

— viteza de propagare pe linie este de 17,6 cm/ns (sau linia întîrzie cu 0,057 ns/cm).

— timpul de propagare minim al bistabililor D este 2 ns.

*Rezolvare*

Este necesar ca propagarea ceasului pe linie să aibă loc în mai puțin de 2 ns pentru că altfel comută primul bistabil și informația se va înscrie eronat în cel de-al doilea.

$$\text{Deci } l < \frac{2 \text{ ns}}{0,057 \text{ ns/cm}} = 35 \text{ cm}.$$

Mai ales dacă cei doi bistabili se găsesc pe plăci diferite e posibil să se atingă această lungime de conexiune.

**Problema 5.19.** Cît de mare poate fi linia de ceas din problema anterioară dacã între ieșirea  $Q_1$  și intrarea  $D_2$  sînt intercalate 2 porți TTLS ? Sã se stabileascã factorii de care depinde frecvența maximã a ceasului.

*Indicație*

În acest caz lungimea liniei poate crește deoarece intervin și timpii de propagare prin porți. Cînd timpul de propagare nu este esențial se pot intercala porți doar în scopul rezolvãrii problemelor de cablaj.

Perioada minimã a ceasului = întîrzierea în bistabil + timpii de propagare prin porți + „set up time“ (timpul cu care trebuie sã aparã informația la intrarea  $D$  înaintea comutãrii pentru a putea fi preluatã corect).

**Problema 5.20.** De ce trebuie conectate intrãrile neutilizate ale porților integrate la masã sau sursã, sau la alte intrãri utile ale circuitului ?

*Soluție*

Dacã se lasã în gol ar putea culege zgomote prin cuplaje parazite cu circuitele vecine și astfel ar perturba funcționarea circuitului.

**Problema 5.21.** La ce potențial trebuie conectate intrãrile neutilizate ale circuitelor urmãtoare pentru a permite funcționarea corectã a circuitului

a) NAND ?

b) NOR ?

c)  $AB + CD$  ?

*Soluție*

a) prin rezistență la  $V_{cc}$ ; b) GND; c) prin rezistență la  $V_{cc}$ .

**Problema 5.22.** Se fac mãsurațori de fronturi într-o schemã cu CI TTL (7 ns). Care va fi indicația unui osciloscop cu bandã de frecvență între 0 și frecvența  $f$  de :

a) 10 MHz ?

b) 50 MHz ?

c) 300 MHz ?

*Rezolvare*

Frontul asociat osciloscopului cu frecvența de sus din bandã  $f$  este :

$$t_0 = \frac{0,35}{f},$$

Deci în cazul problemei :

a)  $t_0 = 35$  ns; b) 7 ns; c) 3,5 ns.

Se cunoaște cã frontul de mãsuraț  $t_m$  și cel al osciloscopului se adunã pãtratic. Astfel cã frontul ce se vede pe osciloscop este :

$$t_v = \sqrt{t_m^2 + t_0^2}$$

Deci :

a)  $t_v = t_r$

În acest caz osciloscopul nu este bine ales pentru măsurătoare. Frontul de măsurat fiind mult mai mic decât cel al osciloscopului, nu contează. Aceasta este de fapt o metodă de a măsura frontul osciloscopului.

b)  $t_r = 7 \sqrt{2} \cong 10 \text{ ns}$

În această situație se pot măsura fronturi tipice circuitelor integrate, dar trebuie corectată eroarea datorată osciloscopului.

c)  $t_v = t_m$

În această situație, osciloscopul avînd front mult mai bun decât semnalul vizualizat, nu influențează măsurătoarea.

**Problema 5.23.** Frontul unui semnal măsurat cu un osciloscop de 75 MHz este de 10 ns. Care este frontul real al semnalului ?

*Soluție*

$$\sqrt{10^2 - 4,66^2} = 8,8 \text{ ns.}$$

**Problema 5.24.** Care va fi frontul vizualizat pe un osciloscop de 50 MHz dacă frontul real al semnalului este de 10 ns ?

**Problema 5.25.** Dorim să determinăm banda unui osciloscop. Aplicăm un semnal cu front cunoscut de 5 ns. Măsurăm frontul semnalului cu ajutorul acestui osciloscop : rezultă 17,5 ns. Care este frecvența de sus a osciloscopului ?

**Problema 5.26.** Un osciloscop are impedanța de intrare  $1 \text{ M}\Omega$ ,  $30 \text{ pF}$ . Calculați capacitatea ce trebuie reglată la o sondă atenuatoare 10 : 1 astfel ca aceasta să nu distorsioneze semnalul.

*Rezolvare*

Trebuie ca divizorul de impulsuri format să fie compensat. Rezultă  $C = 3 \text{ pF}$  (figura 5.12).

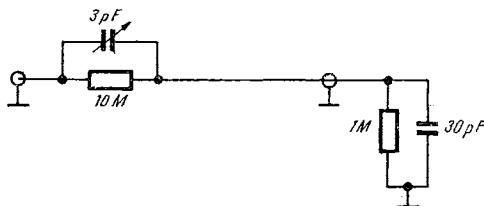


Fig. 5.12.

**Problema 5.27.** Reluați problema 5.26 în cazul impedanței de intrare de  $1 \text{ M}\Omega$ ,  $100 \text{ pF}$  pentru un atenuator 20 : 1.

**Prezentarea conținutului**

Acest capitol se referă inițial la unele aspecte specifice SLS utilizate în structuri microprogramate : modul cum se pot reduce circuitele logice combinaționale asociate (problemele 6.1 ÷ 6.6).

De asemenea se prezintă implementări ale unor structuri specifice microprogramării :

- CROM (Controlor + ROM) -- probleme 6.7—6.10 ;
- RALU (Registre + ALU) — problemele 6.11—6.26.

Se indică modul de cuplare al acestora în procesoare, structura cuvântului de microprogram.

În partea a doua a capitolului sînt prezentate programe de aplicații cu microprocesorul de tip INTEL 8080 (6.30—6.46).

**Problema 6.1.** Să se implementeze un sistem la care semnificația a 8 biți de intrare poate fi interpretată independent. Numărul total de intrări este 10 și sistemul are 30 de stări. Cuvîntul de comandă în exterior are 4 biți.

*Rezolvare*

Implementarea este dată în figura 6.1.

Cu MUX dimensiunea ROM-ului devine

$$N = 12 \times 2^8$$

față de cazul în care nu se folosește MUX și în care :

$$N' = 9 \times 2^{15}$$

Deci, cu MUX, ROM-ul este de 96 de ori mai mic.

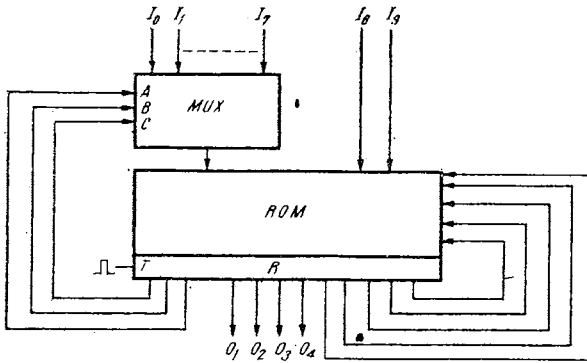


Fig. 6.1.

**Problema 6.2.** Să se reia problema anterioară modificându-se ieșirea în sensul că se realizează prin 16 cuvinte distincte de 16 biți fiecare.  
*Indicatie*

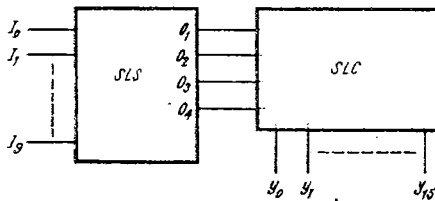


Fig. 6.2.

Se adaugă la sistemul logic secvențial SLS din problema anterioară, sistemul logic combinațional, SLC care acționează asupra ieșirilor (figura 6.2).

**Problema 6.3.** Evoluția sistemului din figura 6.3 este determinată de starea în care se află, de răspunsul sistemului  $SL_2$  pe căile  $X_0$  și de intrarea  $X$ .  $SL_2$  primește 8 comenzi unice și independente pe  $Y_0$  și răspunde cu comenzi date pe numai cite una din cele 8 căi  $X_0$ .

Să se deducă structura de principiu care rezultă datorită acestui mod particular de funcționare, în așa fel încât ROM-ul asociat să fie minim.

*Rezolvare*

Structura sistemului este prezentată în figura 6.4.

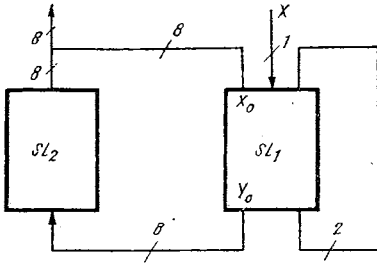


Fig. 6.3.

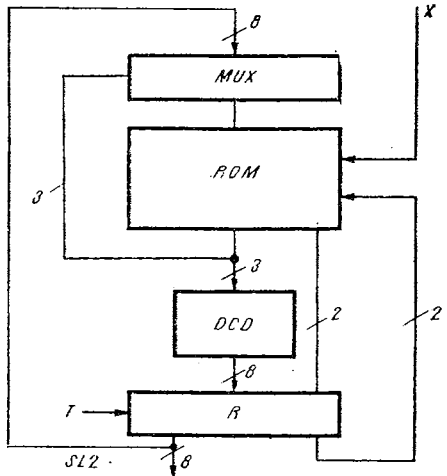


Fig. 6.4.

**Problema 6.4.** Să se deducă structura sistemului logic SL, din figura 6.5 în cazul în care tranziția dintr-o stare în alta nu este condiționată de mai mult de un singur bit din cei 8 de intrare. Se consideră ieșirile ca fiind asincrone.

*Rezolvare*

Numărul intrărilor în CLC-ul asociat sistemului este foarte mare (16). Datorită proprietății sistemului de a avea tranziția condiționată de un singur bit din cei 8 de intrare se poate descompune structura combinațională într-un CLC' și un MUX de 8 căi, MUX8 (fig. 6.6).

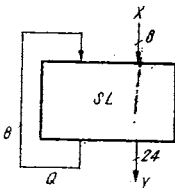


Fig. 6.5.

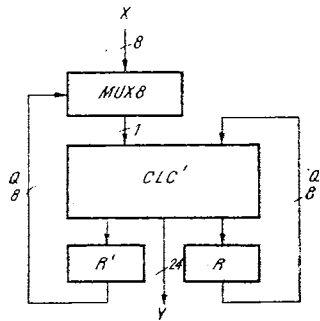


Fig. 6.6.

Se adaugă astfel 3 căi suplimentare de reacție ( $Q'$ ) prin care se indică bitul de intrare ce trebuie transmis către CLC' pentru a determina starea următoare.

Se mărește astfel cu 3 numărul de ieșiri al circuitului logic combinațional, dar creșterea numărului de CI necesare pentru implementare datorită lui  $Q'$  nu este importantă.

Cu ordine de mărime este mai însemnată scăderea dimensiunii circuitului combinațional ca urmare a reducerii intrărilor de la 16 la 9. Deci, introducerea lui MUX8 are ca efect obținerea unei scheme cu CLC', posibil de implementat cu mult mai puține CI în comparație cu CLC.

**Problema 6.5.** Să se optimizeze structura SL din problema anterioară știind că fiecare bit din spațiul de intrare se testează de cel mult 32 de ori.

*Rezolvare*

Printr-o codificare adecvată a stărilor, acestea pot fi grupate în clase de echivalență conținând câte 32 de stări. Fiecare clasă va conține doar stări care generează tranziția condiționat de același bit din  $X$ , sau stări din care tranziția are loc necondiționat. În acest caz vor putea fi utilizați 3 biți din  $Q$  pentru indicarea acestor clase de echivalență (figura 6.7) și pentru selectarea bitului testat asociat. Cu aceste restricții nu mai este necesară introducerea reacției  $Q'$  (vezi problema anterioară). Deci apar avantaje datorită absenței registrului  $R'$  și unui număr micșorat de ieșiri ale CLC asociat sistemului.

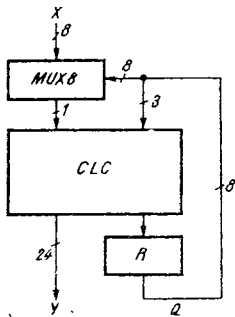


Fig. 6.7.

Printr-o codificare adecvată a stărilor, acestea pot fi grupate în clase de echivalență conținând câte 32 de stări. Fiecare clasă va conține doar stări care generează tranziția condiționat de același bit din  $X$ , sau stări din care tranziția are loc necondiționat. În acest caz vor putea fi utilizați 3 biți din  $Q$  pentru indicarea acestor clase de echivalență (figura 6.7) și pentru selectarea bitului testat asociat. Cu aceste restricții nu mai este necesară introducerea reacției  $Q'$  (vezi problema anterioară). Deci apar avantaje datorită absenței registrului  $R'$  și unui număr micșorat de ieșiri ale CLC asociat sistemului.

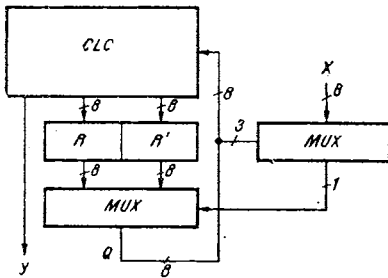


Fig. 6.8.

**Problema 6.6.** a) Să se indice un procedeu pentru reducerea la 8 a numărului de intrări în CLC din problema anterioară.

b) Să se prezinte avantajele metodei.

*Soluție*

a) Vezi figura 6.8.

b) Dacă CLC-urile din problemele 6.5 și 6.6 sînt implementate cu



ROM, atunci capacitatea de memorie necesară în cele două cazuri este de 16 Kbiți și, respectiv 10 Kbiți. Deci prin introducerea celui de-al doilea MUX în schemă se înjumătățește capacitatea memoriei ROM.

O astfel de soluție este cu atât mai eficientă cu cât numărul de ieșiri din CLC pentru reacție este mai mare.

**Problema 6.7.** Să se realizeze o unitate de control utilizând structuri ROM (CROM) care să aibă următoarele caracteristici :

— să poată rula un număr de microprograme distincte definite prin 2 cuvinte de 4 biți, care să utilizeze cel mult 256 de microinstrucțiuni distincte ;

— să poată realiza saltul peste o microinstrucțiune, necondiționat și condiționat (în funcție de 16 condiții distincte) ;

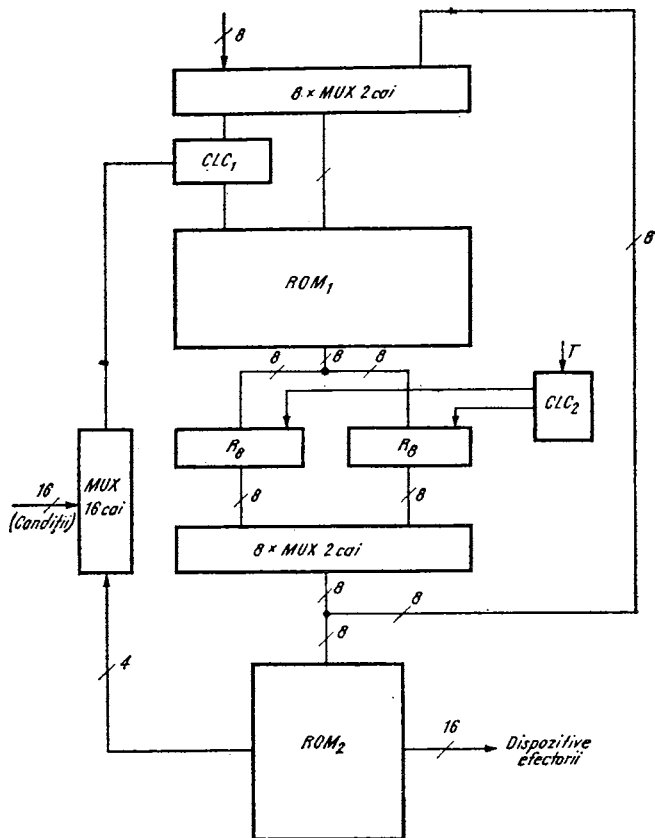


Fig. 6.9.

- să realizeze saltul la o subrutină cu posibilitatea de revenire în microprogramul principal ;
- fiecare microinstrucțiune are o organizare pe 16 biți, structura fiind variabilă.

#### *Indicație*

Schema de principiu e dată în figura 6.9, unde :

- CLC<sub>1</sub> — asigură modificarea bitului al 8-lea în funcție de îndeplinirea condiției selectate din ROM<sub>2</sub>.
- CLC<sub>2</sub> — asigură funcționarea corectă a celor două registre R<sub>3</sub>, dintre care unul selectează adresa curentă pentru ROM<sub>2</sub>, iar celălalt adresa de la care s-a intrat în subrutină, pentru a putea permite revenirea în programul principal.

**Problema 6.8.** Să se reia problema anterioară pentru cazul în care se utilizează o singură memorie ROM.

**Problema 6.9.** Să se construiască un CROM pentru o structură microprogramată ce este controlată printr-un cuvânt de microprogram de 32 de biți. Se utilizează 256 de instrucțiuni și sînt posibile următoarele moduri de tranziții:

- 0) prin incrementarea adresei curente ;
- 1) prin salt condiționat peste instrucțiunea următoare ;
- 2) prin salt la o adresă generată de un cîmp al cuvîntului de microprogram ;
- 3) prin salt condiționat la o adresă generată de un cîmp al cuvîntului de microprogram ;
- 4) prin salt la o adresă generată extern ;
- 5) prin salt condiționat la o adresă generată de microprogram, cu menținerea adresei curente incrementate (salt la subrutină) ;
- 6) salt necondiționat la adresa zero ;
- 7) întoarcerea din subrutină condiționată prin generarea adresei reținute la ordinul de salt din subrutină.

Salturile sînt condiționate de testarea unui indicator din 16 posibili.

#### *Soluție*

Structura rezultată va fi aceea a unui automat complex, conform reprezentării din figura 6.10, unde notațiile au următoarea semnificație :

MUX — multiplexor de selectare a sursei ce generează adresa următoare de microprogram ;

R — registru de ieșire în care este menținută adresa de microprogram curentă ;

R' — registru pentru memorarea adresei de reîntoarcere din subrutină ;

S<sub>8</sub> — sumator de 8 biți comandat pentru adunarea cu 1 sau cu 2 ;

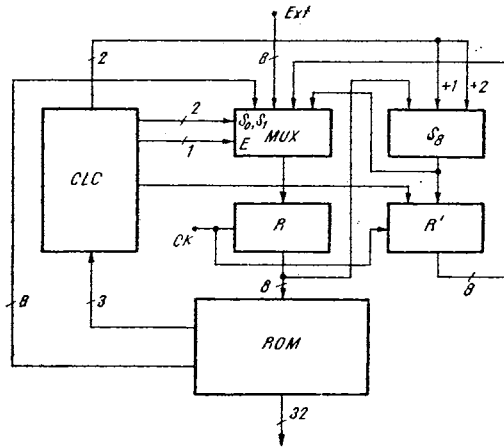


Fig. 6.10.

CLC — circuit logic combinațional ce asigură transcodarea comenzii primite din ROM ;

ROM — memorie pentru microprograme ;

EXT — cale pe care este primită adresa de microprogram generată extern.

**Problema 6.10.** Cum trebuie modificată soluția problemei anterioare, pentru a obține un CROM care să permită rularea unor microprograme, cu posibilitatea realizării a 8 salturi la subrutină, fără nici o comandă de revenire intermediară ?

*Indicație*

Se înlocuiește R' cu o memorie de tip stivă.

**Problema 6.11.** Să se proiecteze un sistem RALU (Registre și ALU) care să realizeze următoarele funcții :

(M) → R adică : înscrie conținutul adresei M în registrul R.

R → (M) : înscrie R la adresa M.

(M) \* R → R : operare între (M) și R cu rezultat în R.

unde :

M este una din cele 16 adrese de memorie a unui RAM de 16 cuvinte a 4 biți (7489).

R este un registru operativ de 4 biți (acumulator).

\* este o operație ce se efectuează între conținutul adresei M și al registrului R, operație ce poate fi : adunare binară, scădere binară,  $\oplus$ , +, .., operație nedefinită N<sub>1</sub>, operație nedefinită N<sub>2</sub>, ștergerea (ZERO).

Toate funcțiile vor fi realizate într-o singură secvență, sincron cu un impuls de ceas.

**Rezolvare**

Se codifică în felul următor funcțiile și operațiile executate :

FUNCȚIA (F)	F <sub>0</sub>	F <sub>1</sub>
(M) → R	0	0
R → (M)	0	1
(M) * R	1	0
NO OP	1	1

NO OP – Nici O Operație  
(Sistemul este oprit)

Operația *	O <sub>0</sub>	O <sub>1</sub>	O <sub>2</sub>
Sumă	0	0	0
Diferență	0	0	1
⊕	0	1	0
+	0	1	1
•	1	0	0
N <sub>1</sub>	1	0	1
N <sub>2</sub>	1	1	0
ZERO	1	1	1

Se alege schema bloc din figura 6.11, unde :

D este bistabil de tip D pentru memorarea depășirii la operațiile aritmetice.

C<sub>1</sub> este un circuit logic combinațional care realizează operațiile \* între M și R (ALU).

C<sub>2</sub> este un circuit logic combinațional ce realizează funcțiile F între M și R.

T este calea pentru impulsurile de tact.

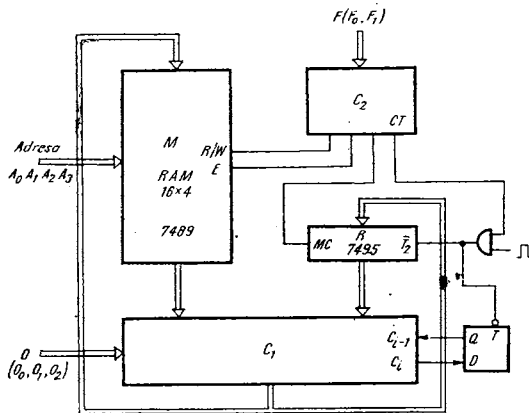


Fig. 6.11.

Pentru proiectare trebuie definite operațiile  $N_1$  și  $N_2$  astfel încât să se poată realiza toate funcțiile  $F$  ale sistemului. Definirea va fi:  
 $N_1$  transferă  $(M) \rightarrow R$ , indiferent de conținutul lui  $R$ ;  
 $N_2$  transferă  $R \rightarrow (M)$ , indiferent de conținutul lui  $M$ ;  
 $C_2$  este descris de următorul tabel de adevăr:

$F$	$F_0$	$F_1$	R/W	E	MC	CT
$(M) \rightarrow R$	0	0	1	0	1	1
$R \rightarrow (M)$	0	1	0	0	$\times$	1
$(M) * R \rightarrow R$	1	0	1	0	1	1
NO OP	1	1	$\times$	1	$\times$	0

Astfel, pentru  $(M) \rightarrow R$  se citește registrul de memorie ( $R/W = 1$ ) care este adresat din exterior, este selectată memoria ( $E = 0$ ), registrul este cuplat pe înscrisere paralelă ( $MC = 1$ ) și se dă tact la  $R$  ( $CT = 1$ ). Deci, conținutul adresei  $(M)$  este citit din memorie, este trecut prin  $C_1$  care, în acest caz nu trebuie să-l prelucreze (va efectua operația  $N_1$ ) și se va înscris în registrul  $R$ . În cazul  $(M) * R \rightarrow R$ , comenzile date de  $C_2$  vor fi aceleași, diferența va consta în faptul că  $C_1$  va opera conform codului 0 asupra lui  $(M)$  și  $R$ .

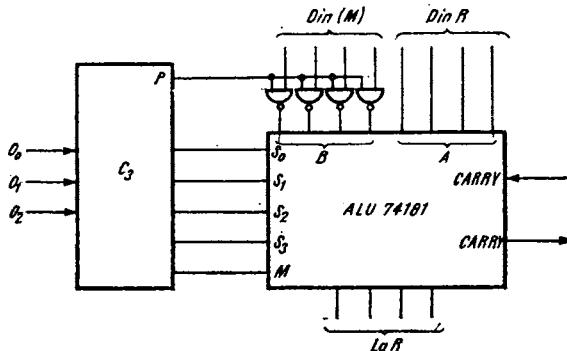


Fig. 6.12.

$C_1$  va avea forma din figura 6.12, unde  $C_3$  este un convertor de cod de la 3 biți la 6 biți, definit de următorul tabel de adevăr:

Operația	O <sub>0</sub>	O <sub>1</sub>	O <sub>2</sub>	M	S <sub>2</sub>	S <sub>1</sub>	S <sub>0</sub>	P
suma	0	0	0	0	1	0	0	1
diferența	0	0	1	0	0	1	1	0
⊕	0	1	0	1	0	1	1	0
+	0	1	1	1	1	1	1	0
·	1	0	0	1	1	0	1	1
N <sub>1</sub>	1	0	1	1	1	1	1	0
N <sub>2</sub>	1	1	0	1	1	0	1	1
ZERO	1	1	1	1	0	0	1	1

Implementarea pentru C<sub>2</sub> și C<sub>3</sub> se face cu metodele cunoscute. Pentru (M) → R se comandă sistemul cu codul

$$F_0F_1 = 00, O_0O_1O_2 = 110 (N_1)$$

iar pentru R → (M)

$$F_0F_1 = 01 \text{ și } O_0O_1O_2 = 101$$

### Problema 6.12.

a) Cum se realizează cu sistemul din problema precedentă următoarele operații :

1. ștergerea registrului R (CLEAR R).
2. ștergerea bistabilului de transport (CLEAR CARRY).
3. înscrierea transportului (SET CARRY).
4. complementarea registrului R.

b) Care este frecvența maximă a impulsurilor de ceas pentru care se asigură funcționarea corectă a sistemului ?

*Soluție*

a) 2. Pentru ștergerea bistabilului de transport (CLEAR CARRY) este necesară următoarea succesiune de operații :

R → (M)

R-(M) → R

(M) → R

Deci, se memorează conținutul registrului R, apoi se scade (M) din R și, (M) avînd același conținut, rezultatul scăderii va fi 0, deci bistabilul de transport, indiferent de starea sa anterioară va trece în 0. În continuare se reinscrie în R numărul existent la începutul operării.

**Problema 6.13.** În problema 6.11 prin modificarea codurilor asociate funcțiilor și operațiilor RALU se poate simplifica structura schemelor combinaționale C<sub>2</sub> și C<sub>3</sub> ?

**Problema 6.14.** a) Să se reia problema 6.11 completându-se modul de funcționare cu posibilitatea de a introduce și extrage date (cuvinte de 4 biți) din sistem.

b) Să se realizeze RALU cu posibilitatea introducerii și extragerii datelor pe aceleași căi (cuplare bidirecțională).

c) Pentru sistemele anterioare să se calculeze frecvența maximă de lucru.

*Indicație*

a) Se definește un mod de funcționare prin :

$$R * (M) \rightarrow R$$

$$R * (M) \rightarrow M$$

$$\rightarrow R \text{ (înscrie în } R)$$

$$R \rightarrow \text{(identic cu NO OP).}$$

**Problema 6.15.** Să se reproiecteze RALU anterior definit astfel încât să aibă o structură modulară : să se poată extinde, prin cuplarea cu unul identic pentru a putea prelucra cuvinte de 8 biți.

**Problema 6.16.** Să se reconsidere RALU din problema 6.11 realizând în locul operațiilor  $\oplus$ ,  $+$ ,  $\cdot$ , următoarele :

1. ROT — rotație în registrul format de R și bistabilul D.  
 $R_0R_1R_2R_3D \rightarrow R_3R_0R_1R_2D$

2. DPLS — deplasare stînga în același registru  
 $R_0R_1R_2R_3D \rightarrow R_1R_2R_3DD$

3. DPLD — deplasare dreapta în același registru  
 $R_0R_1R_2R_3D \rightarrow OR_0R_1R_2R_3$

*Indicație*

Se va utiliza proprietatea registrului 7495 de a deplasa serie conținutul său, realizîndu-se, cînd este cazul, un registru de 5 biți prin extensie cu bistabilul de transport, D.

**Problema 6.17.** a) Cum trebuie corelată modificarea configurației  $F_0F_1 O_0O_1O_2$  cu impulsurile de tact pentru RALU din problema 6.11.

b) Să se conceapă un sistem de cuplare a biților F și O astfel încît corelarea cu impulsurile de tact să se facă automat.

*Soluție*

b) Pentru schema propusă în figura 6.13, registrul care preia biții F și O este activ pe frontul pozitiv al impulsului de tact, iar RALU este activ pe frontul negativ. În registrul FO se înscrie funcția și operația pentru RALU pe durata T a impulsului (care este cel puțin egală cu timpul maxim de prelucrare a sistemului) efectuarea lor începînd odată cu frontul negativ.

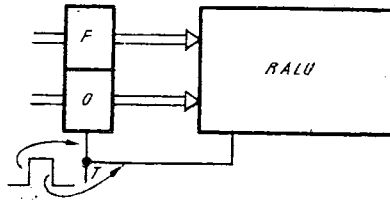


Fig. 6.13.

**Problema 6.18.** Utilizând circuite 7495 să se implementeze un registru de 16 biți, organizat în 4 cuvinte de câte 4 biți, care să poată fi încărcat de la două surse de date, în 4 impulsuri de tact, fără a se utiliza circuite adiționale.

*Rezolvare*

Accesul informației de la sursa de date 1 (figura 6.14) se face pe intrările serie SI ale fiecărui registru cînd  $MC = 0$ . Pe fiecare tact se rea-

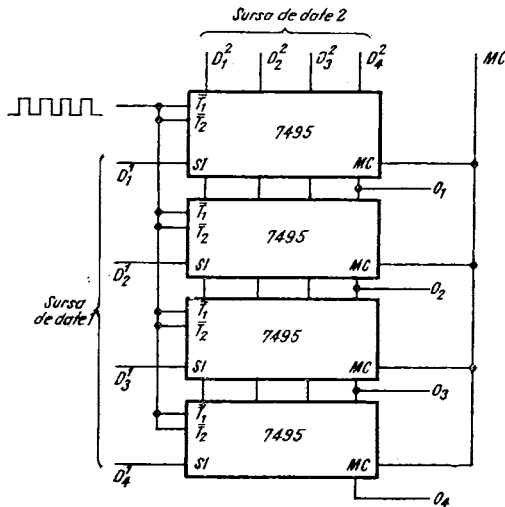


Fig. 6.14.

izează deplasarea serie a informației înregistrare, după 4 tacturi informația fiind încărcată. Pentru a permite accesul datelor de la sursa 2, e necesar ca  $MC = 1$ ; în acest caz datele se înscriu și se deplasează paralel în registre.



*Observație*

Schema realizează o multiplexare secvențială deoarece adună datele de pe câte 2 căi transmițându-le pe o singură cale.

**Problema 6.19.** Să se reconsidere RALU din problema 6.11 în cazul în care se prelucrează serie-paralel cuvinte de 16 biți în 6 secvențe, iar numărul adreselor de memorie este 4.

*Indicație*

Schema de principiu devine cea din figura 6.15.

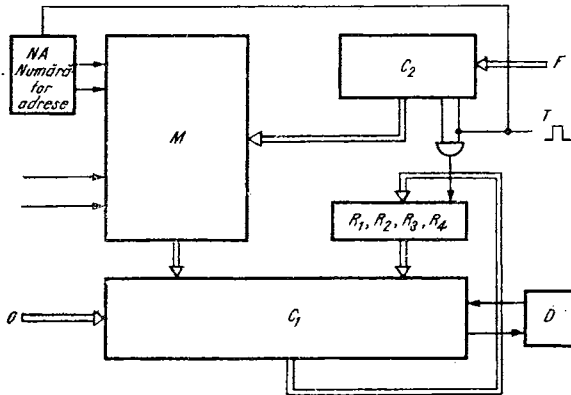


Fig. 6.15.

unde NA este un numărător prin care se baleiază cele 4 celule în care este stocat un cuvânt de 16 biți și R devine un sistem de 4 registre ca cel definit în problema 6.18.

**Problema 6.20.** Cum se modifică sistemul din problema anterioară dacă numărul adreselor de memorie este 16?

**Problema 6.21.** Să se implementeze un RALU care să efectueze următorul set de instrucțiuni într-o singură secvență:

$$\left. \begin{array}{l} R + (M) \rightarrow R - \text{adunare} \\ R - (M) \rightarrow R - \text{scădere} \\ R \oplus (M) \rightarrow R - \text{SAU EXCLUSIV} \\ R + (M) \rightarrow R - \text{SAU} \\ R \cdot (M) \rightarrow R - \text{SI} \end{array} \right\} \text{cu rezultatul în R}$$

R → (M)  
 (M) → R  
 NO OP — nu operează

CLR CARRY — șterge transportul  
 SET CARRY — pune transportul pe 1  
 ROT — rotație  
 ROTD — împreună cu bistabilul de transport D  
 DPLS deplasare stînga cu D  
 DPLD deplasare dreapta cu D  
 DATA IN — introducere de date în sistem  
 DATA OUT — extragere de date.  
 Schema de conexiuni a sistemului este dată în figura 6.16, unde :

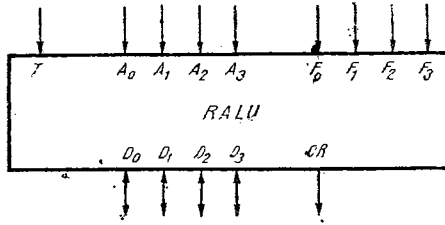


Fig. 6.16.

$D_0D_1D_2D_3$  — sînt căi bidirecționale de date  
 $F_0F_1F_2F_3$  — codul instrucțiunilor  
 $A_0A_1A_2A_3$  — adresarea memoriei interne  
 T — intrare de ceas  
 CR — ieșire care dă starea bistabilului CARRY.

Cîte terminale trebuie să aibă circuitul integrat conținînd acest RALU ?

*Soluție*

Vezi figura 6.17.

Conținutul ROM-ului 7488 este dat în tabelul următor :

	Pentru T = 1				Pentru T = 0											
$F_3F_2F_1F_0$	$S'_0$	$S'_1$	P	E	W	R	STC	CLG	$S_0$	$S_1$	$S_2$	$S_3$	M	S	N	Q
Add	1	1	0	1	0	1	1	1	0	0	1	0	0	1	1	
Sc	1	1	0	1	0	1	1	0	1	1	0	0	0	0	1	1
$\oplus$	1	1	0	1	0	1	1	0	1	1	0	1	0	0	0	
+	1	1	0	1	0	1	1	0	1	1	1	1	1	0	0	
·	1	1	0	1	0	1	1	1	1	0	1	1	0	0	0	
R-M	0	0	0	0	0	1	1	0	0	0	0	1	0	0	0	
M-R	1	1	0	1	0	1	1	0	1	0	1	1	0	0	0	
NO OP	0	0	0	1	1	1	1								0	0

	Pentru T = 1					Pentru T = 0									
CLR CARRY	0	0	0	1	0	0	0								
SET CARRY	0	0	0	0	1	0	0								
ROT	1	0	0	1	1	0	0								
ROT CR	1	0	0	1	1	0	1								
DPL D	1	0	0	1	1	1	0								
DPL S	0	1	0	1	1	0	0								
DATA IN	1	1	0	1	1	1	1	0	0	0	0	1	1	0	0
DATA OUT	0	0	1	1	1	1	1							0	0

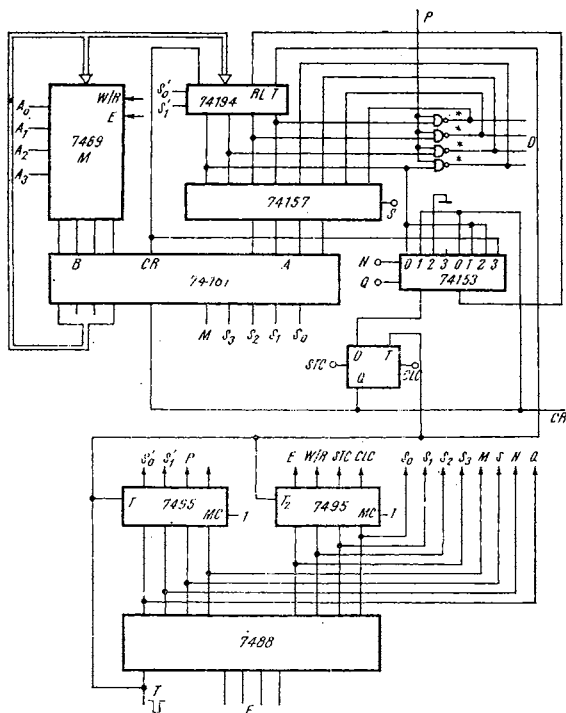


Fig. 6.17.

Frontul negativ al T validează starea de 16 biți generată de ROM prin înscrierea primilor 8 biți în 2 registre 7495 și prin selectarea celorlalți 8 biți cu starea T = 0.

Frontul pozitiv al T înscrie rezultatul prelucrării în registrul 74194 și bistabilul CARRY.

**Problema 6.22.** În sistemul definit de problema anterioară să se înscrie numărul 2 în adresa de memorie  $N_3(A_3A_2A_1A_0 = 0011)$  printr-o succesiune de instrucțiuni convenabilă. Conținutul celorlalte adrese nu va fi afectat.

*Soluție*

$R \rightarrow (M_3)$   
 $R \oplus M_3 \rightarrow R$  } se șterge registrul R  
 $R \rightarrow (M_3)$  — se șterge adresa de memorie  
 SET CR  
 $R \text{ add } (M_3) \rightarrow R$  } formarea lui 1  
 DPL D — înmulțire cu 2  
 $R \rightarrow (M_3)$  2 în  $M_3$

**Problema 6.23.** Să se implementeze un RALU care să opereze cu cuvinte de 4 biți, având 16 registre și un acumulator.

*Soluție*

O structură posibilă este cea din figura 6.18, unde :

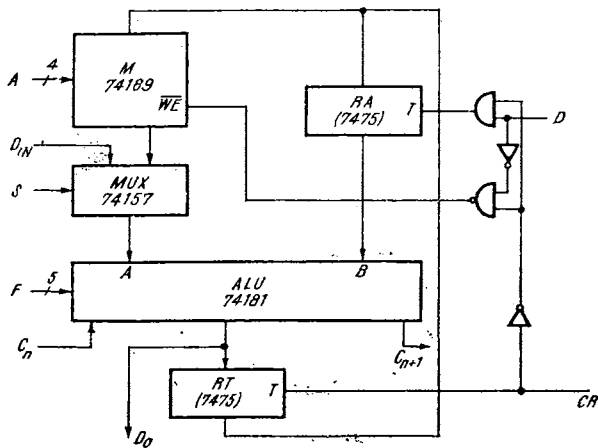


Fig. 6.18.

M este o memorie de 16 cuvinte a câte 4 biți adresabilă pe calea A din exterior :

MUX selectează prin S ca operand de tip A la ALU, conținutul M sau semioctetul de la  $D_{IN}$  ;

ALU este o unitate logico-aritmetică de 4 biți cu funcția comandată pe F ;

RT este registru tampon ce permite realizarea unei structuri de tip MS (master-slave) pentru închiderea corectă a buclei de reacție ;  
 RA este registru acumulator ;

$C_n$  este depășirea de la ordinele binare anterioare ;

$C_{n+1}$  este depășirea către ordinele binare superioare ;

D selectează destinația ;

CK este ceasul sistemului ;

A selectează adresa unuia din cele 16 registre ale M.

**Problema 6.24.** Să se conceapă pentru un RALU un sistem de 16 registre în așa fel încît oricare registru să poată fi selectat sincron ca operand stîng sau ca operand drept.

*Soluție*

Din sistemul de registre din figura 6.19 se poate citi informația independentă din  $M_L$  și  $M_R$  (realizate cu circuite de tip 7489) de la adre-

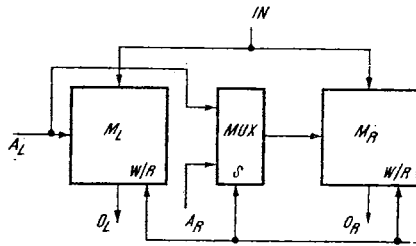


Fig. 6.19.

sele  $A_L$  și respectiv  $A_R$ . Înscriserea se realizează în ambele memorii la adresa  $A_L$ . Este simulată astfel o memorie de 16 cuvinte de 4 biți dublu adresabilă la citire și simplu adresabilă la scriere.

**Problema 6.25.** Care este avantajul utilizării într-un RALU a sistemului de registre din problema anterioară ?

**Problema 6.26.** Cum se poate realiza un RALU de 16 biți utilizînd sistemul de registre definit în problema 6.24 ? De ce și în ce sens se modifică timpul de operare al unui astfel de RALU ?

**Problema 6.27.** Se dau următoarele structuri :

a) CONTROL, cu schema de conexiuni externe din figura 6.20 unde :

- $A_{n+1}$  este adresa de microprogram generată pentru ciclul următor ;

- $B_n$  este adresa de salt generată de microprogram ;

- M este codul ce definește tranziția în microprogram ;

- T este un bit ce indică faptul că este îndeplinită condiția testată în microinstrucțiunea curentă ;
- CK este ceasul sistemului.
- b) RALU, definit pentru cuvinte de 4 biți, cu schema de conexiuni externe ca în figura 6.21, unde :

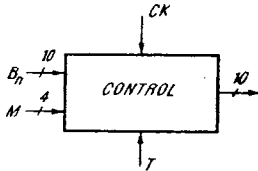


Fig. 6.20.

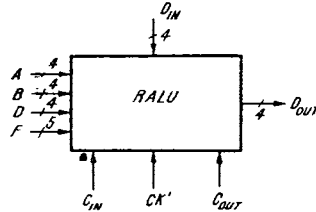


Fig. 6.21.

- A este adresa operandului stîng ;
- B este adresa operandului drept ;
- D este adresa destinației ;
- F este funcția ALU ;
- $D_{IN}$  este cale de date de intrare în RALU ;
- $D_0$  este cale de date de ieșire din RALU ;
- $C_{IN}$  este intrare de transport ;
- $C_0$  este ieșire de transport.

c) I/O, sistem de intrare/ieșire definit pentru 4 biți și controlat de 2 biți, conform reprezentării din figura 6.22, în care BUS este calea externă pe care se cuplează procesorul.

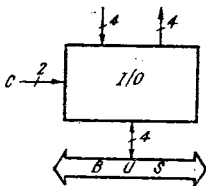


Fig. 6.22.

Se cere să se construiască structura cuvîntului de microprogram necesară pentru a comanda un sistem microprogramat de procesare ce lucrează cu cuvinte de 16 biți.

Se va considera că numărul de indicatori testați pentru a determina evoluția microprogramului este de 15.

### Soluție

Structura cuvîntului de microprogram va fi următoarea :  $A_3 \dots A_0, B_3 \dots B_0, D_3 \dots D_0, F_4 \dots F_0, M_3 \dots M_0, T_3 \dots T_0, B_9 \dots B_0, C_1 C_0$ .  
Semnificația celor 8 cîmpuri ale microinstrucțiunii este următoarea :

- $A_3 \dots A_0$  — sursa operandului A ;
- $B_3 \dots B_0$  — sursa operandului B ;
- $D_3 \dots D_0$  — destinația rezultatului ;

- $F_3 \dots F_0$  — funcția realizată de ALU ;
- $M_3 \dots M_0$  — funcția de tranziție a controlului ;
- $T_3 \dots T_0$  — cod de selectare a bitului testat ;
- $B_9 \dots B_0$  — adresa de salt generată de microprogram ;
- $C_1, C_0$  — biți de control al transportului către exteriorul sistemului.

**Problema 6.28.** Pornind de la datele problemei anterioare să se implementeze structura unui procesor de 16 biți la care căile de date și adrese sînt comune.

**Problema 6.29.** Să se proiecteze un sistem microprogramat care să poată genera prin microprogram operanzi pentru RALU.

**Problema 6.30.** Se poate asocia unui cîmp al unui cuvînt de microprogram o semnificație multiplă ? Care sînt consecințele ce apar asupra întregii structuri a cuvîntului de microprogram ?

**Problema 6.31.** Viteza de lucru a unui sistem microprogramat este afectată de structura cuvîntului de microinstrucțiuni ? Cum ?

**Problema 6.32.** a) Să se scrie un program de realizarea operației la sumare zecimală între numere de două cifre cu microprocesorul INTEL 8080. Se pornește de la starea în care :

- adresa numărului A este în D și E ;
- adresa numărului B este în H și L ;
- rezultatul va fi stocat la adresa din D și E.

b) Să se calculeze timpul necesar operației.

**Problema 6.33.** Să se scrie în condiții similare un program pentru scădere zecimală.

**Problema 6.34.** Să se scrie pentru INTEL 8080 un program de realizare a operației de sumare zecimală pentru numere de 16 cifre. Starea inițială este :

— în registrele D și E este stocată adresa la care se găsesc primii 2 digiți (cei mai puțin semnificativi) ai numărului A, restul fiind stocați în următoarelor 7 locații de memorie,

— în registrele H și L este stocată adresa la care se găsesc digiții cei mai puțin semnificativi ai numărului B, ceilalți fiind memorați la următoarele 7 adrese.

Suma este stocată în locul numărului A.

*Rezolvare*

Mnemonic	Operand	Comentariu	
LXI	D	Set address to	DE
LXI	H	„	HL
MVI	C, 8		
XRA		Clear Carry	

LOOP	LDAX	D	Load Avgend to Ace
	ADC	M	Ad Addend to Avgend
	DAA		
	STAX	D	Replace results
	INX	H	Renew address HL
	INX	D	” DE
	DCR	C	Check end of calculation
	INZ	LOOP	If not zero go to loop

**Problema 6.35.** Să se scrie un program similar pentru operația de scădere.

**Problema 6.36.** Să se scrie un program de înmulțire a două numere formate din două cifre. Să se propună și un mod de organizare a datelor.

*Indicație*

Dacă cele două numere sînt

$D_1D_0$  (deînmulțitul)

$I_1I_0$  (înmulțitorul),

se adună la 0 de  $I_1$  ori  $D_0D_1$ , se înmulțește cu 10 (deplasare cu un ordin zecimal) și la rezultat se adună de  $I_0$  ori  $D_1D_0$ .

Programul din problema 6.34 poate fi utilizat ca subrutină prin definirea convenabilă a conținutului lui DE și HL care nu vor mai fi modificate (se înlătură INX H și INX D din program).

**Program 6.37.** Să se conceapă organigrama înmulțirii a două numere formate din  $2n$  cifre fiecare.

**Problema 6.38.** Să se scrie programul asociat organigramei din problema anterioară.

**Problema 6.39.** Utilizînd microprocesorul INTEL 8080 cite subrutine care se includ se pot realiza într-un program ?

**Problema 6.40.** Conținutul registrului acumulator este următorul :

$A_7 A_6 A_5 A_4 A_3 A_2 A_1 A_0$

0 1 1 0 1 1 0 1

a) Care va fi conținutul acestuia după succesiunea de operații : RLC, RAL, IWR, III, XRA ?

b) Dacă după această succesiune se realizează instrucțiunea CC, se modifică sau nu starea registrului SP ?

*Rezolvare*

b) Nu se modifică, deoarece „carry“ a fost anulat prin instrucțiunea XRA.

**Problema. 6.41.** Să se prezinte modul de utilizare al stivei pentru realizarea programului descris de organigrama din figura 6.23, în care  $P_k$  reprezintă programe, iar  $C_k$  sînt condiții oarecare.



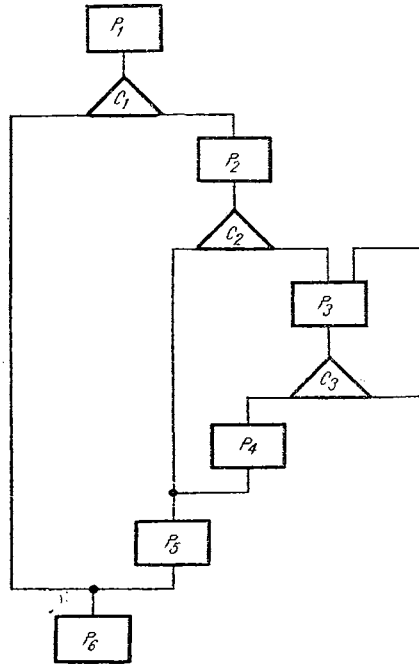


Fig. 6.23.

Care este capacitatea de memorie ce trebuie alocată pentru organizarea stivei în memoria operativă ?

**Problema 6.42.** Se dă un șir de octeți S. Să se înlocuiască fiecare apariție a octetului aflat în registrul B cu octetul aflat în registrul C. Lungimea șirului S se află în D, iar adresa sa de început în registrele HL.

*Soluție*

TRANS	MOV, M, A	* A ← S(i)	*
	CMP B		
	JNZ T1		
	MOV M, C	* dacă A = B, atunci S(i) ← C*	
T1	INX H	* i ← i + 1 *	
	DCR D	* D ← D - 1 *	
	JNZ TRANS	* reia ciclul dacă nu s-a ajuns la sfârșitul șirului *	
	STOP		

**Problema 6.43.** Să se scrie o rutină pentru găsirea valorii minime dintr-un tabel TAB de numere binare de un octet. Numerele se află în memorie la adrese consecutive începînd cu adresa dată de valoarea din registrele HL; adresa sfîrşitului tabelului (adresa ultimului octet) se află în registrele DE. La ieşirea din rutină valoarea minimă se va afla în registrul A, iar adresa valorii minime în ADR.

*Soluţie*

MIN	MOV	B, M	* B ← TAB (1) *
ML	INX	H	* i ← i + 1 *
	MOV	A, M	* B > TAB(i + 1) ⇒
	JNC	X <sup>1</sup>	B ← TAB (i + 1)
	MOV	B, A	ADR ← ADRESA
			(TAB(i + 1)) *
	SHLD	ADR	
X <sub>1</sub>	MOV	A, H	* HL = DE ⇒ return, altfel se
			continuă căutarea *
	CMP	D	
	JNZ	ML	
	MOV	A, L	
	CMP	E	
	JNZ	ML	
	MOV	A, B	
	RET		
ADR	DS	2	

**Problema 6.44.** Să se scrie o rutină care ordonează crescător un tabel de numere binare. Tabelul este specificat la fel ca în problema precedentă.

*Soluţie*

SORT	SHLD	S1	* S1 ← adresa (TAB(i)) *
	CALL	MIN	* A ← min (TAB(i), TAB(n)) &
			ADR ← adresa (min(...)) *
	XCNG		* S2 ← DE
	SHLD	S2	
	LHLD	S1	* HL ← adresa (TAB(i)) *
	LHLD	ADR	* DE ← adresa (min(TAB(0),
	XCHG		TAB (n))) *
	MOV	B, A	*
	MOV	A, M	TAB (i) ⇔ min (TAB(i), TAB(n))
	MOV	M, B	
	STAX	D	*
	LHLD	S2	* DE ← adresa (TAB)(n)
	XCHG		*

	LHLD	S1	* HL ← adresa (TAB(i + 1))
	INX	H	*
	MOV	A, H	* dacă adresa (TAB(i + 1)) =
	CMP	D	
	JNZ	SORT	= adresa (TAB(n)) return, astfel
	MOV	A, L	se continuă procesul
	CMP	E	
	JNZ	SORT	
	RET		*
S1	DS	2	
S2	DS	2	

**Problema 6.45.** Să se scrie o rutină care convertește într-un octet binar un număr hexazecimal reprezentat prin două caractere ASCII memorate la adresele AX și BX. La ieșirea din rutină acumulatorul trebuie să conțină 1 dacă cele două caractere ASCII formau un număr hexavalid și 0 în caz contrar. Rezultatul conversiei (în cazul în care  $A = 1$ ) se va memora la adresa AX.

*Soluție*

CONV	LDA	AX	* dacă caracterul din AX nu e
	CALL	DOM	cuprins în intervale $0 \div 9$ ; A $\div$ F
	CPI	00	rutina de conversie se termină cu
	RZ		0 în acumulator (1)
	RLC		*
	RLC		* altfel rutina DOM întoarce echi-
	RLC		valentul binar al caracterului;
	RLC		acesta este trecut în cvartetul cel
	RLC		mai semnificativ *
	MOV,	B, A	* $B \leftarrow A$ *
	LDA	BX	* vezi (1)
	CALL	DOM	
	CPI	00	
	RZ		*
	ORA	B	* $A \leftarrow \text{binar}(AX, BX)$ *
	STA	AX	* $AX \leftarrow A$
	MVI	A, 01	* $A \leftarrow 01$ (răspuns pentru conți-
	RET		nut corect al celulelor AX, BX)
DOM	CPI	30	* dacă
	JC	OUT	$30 \leq A < 39$ sau
	CPI	3A	$41 \leq A < 47$

	JC	C1	⇒ caracter ASCII corect, altfel
	CPI	41	eroare ( $A \leftarrow 0$ )
	JC	OUT	
	CPI	47	
	JC	C2	
OUT	XRA	A	*
	RET		
C1	SUI	30	*
	RET		
C2	SUI	37	*
	RET		
AX	DS	1	•
BX	DS	1	

**Problema 6.46.** Se dau două șiruri I și S specificate prin adresele de început (DE, respectiv HL) și lungimile lor (B, respectiv C). Să se scrie o rutină care detectează apariția șirului S ca subșir în I.

*Soluție*

MATCH	MOV,	A, C	* LX ← lungimea
	STA	L	șirului S *
	SHLD	S1	* S1 ← adresa de început a șiru-
			rului S *
MATCH1	MOV	A, B	* dacă lungimea care a mai rămas
	CMP	C	de parcurs din I e mai mică decât
	JNC	M1	lungimea lui S, $S \not\subset I$ și $A \leftarrow 0$
	XRA	A	
	RET		*
M1	CDAX	D	* Dacă $S(i) = I(j)$ salt
	CMP	M	la LOOP
	JZ	LOOP	
	INX	D	* $j \leftarrow j + 1$ *
	DCR	B	
	JMP	MATCH1	
LOOP	INX	H	* $i \leftarrow i + 1$ *
	DCR	B	
	DCR	C	* dacă s-a ajuns cu succes la ulti-
	INZ	M3	mul caracter al șirului S, $A \leftarrow 01$ ;
	MVI	A, 01	return
	RET		*

M3	LDAX	D	* Dacă S(i) = I(j) go to LOOP
	CMP	M	*
	INX	D	* j ← j + 1 *
	JZ	LOOP	
	LHLD	S1	* Dacă S(i) ≠ I(j), HL ← adresa
	LDA	LX	de început a șirului S & C ← lun-
	MOV	C, A	gimea (S); go to MATCH1
	JMP	MATCH1	*
LX	DS	1	
S1	DS	2	

## Notății

- $\overline{A}$ ,  $\overline{A + BC}$  -- negarea variabilei, respectiv a expresiei de sub bară.  
 $AB$ ,  $A \cdot B$  -- funcția ȘI(AND) între variabilele A și B.  
 $A + B$  -- funcția SAU(OR) între variabilele A și B.  
 $A \oplus B$  -- funcția SAU EXCLUSIV între A și B.  
 $\text{AND}$  -- funcția ȘI.  
 $\text{AOI}$  -- AND OR INVERT: funcția ȘI SAU NU.
- $\text{BCD}$  -- Binary Coded Decimal: cod binar zecimal, care asociază fiecărui rang zecimal 4 biți reprezentând codul binar al numerelor  $0 \div 9$ .  
 $\text{BCD}$  -- cu exces 3 -- cod BCD la care există corespondența număr zecimal -- număr binar +3. Această codificare are avantajul că prin negarea variabilelor se obține complementul față de 9 al numărului, simplificând operația de scădere.
- $\text{BR}$  -- BoRrow: ieșire a numărătorului 74193 care semnalizează cu nivel activ 0, trecerea numărătorului prin starea 0, la numărarea înapoi.
- $\text{BUS}$  -- magistrală de date, adrese sau comenzi într-un sistem.
- $\text{CD}$  -- Count Down: intrare de comandă a numărării înapoi la 74193.  
 $\text{CI}$  -- Circuit Integrat.  
 $\text{CK}$  -- Clock: semnal de ceas în sistemele secvențiale.  
 $\text{CL}$  -- CLear: intrare de ștergere (aducere în 0) a circuitelor.  
 $\text{CLC}$  -- Circuit Logic Combinațional.  
 $\text{CMOS}$  -- Complementary MOS: tehnologie MOS cu tranzistoare complementare.  
 Convertor A/D -- convertor Analog/Digital.  
 Convertor D/A -- convertor Digital/Analog.  
 $\text{CR}$  -- CaRry: transport la sumare sau numărare înainte.  
 $\text{CROM}$  -- Controler + ROM: unitate de bază a structurilor microprogramate.  
 $\text{CU}$  -- Count Up: intrare de comandă a numărării înainte la 74193.
- $\text{DCD}$  -- DeCoDificator.  
 $\text{DMUX}$  -- DeMULTipleXor.
- $\text{ECL}$  -- Emitter Coupled Logic: tehnologie bipolară cu logică cuplată prin emitor, deci foarte rapidă.
- $\text{ENABLE}$  -- (engl. -- a permite) intrare care comandă inhibarea sau dezinhibarea funcționării circuitelor.
- $f$  -- funcție logică booleană.  
 $\text{fan-out}$  -- (engl. -- evantai de ieșire) numărul de circuite de același fel care pot fi comandate de etajul de ieșire al unui circuit.  
 $\text{fnc}$  (f.c.n.c.) -- formă canonică normală conjunctivă (pr. 1.28).  
 $\text{fnd}$  (f.c.n.d.) -- formă canonică normală disjunctivă (pr. 1.28).

GND — GROUND : masa circuitelor integrate.

Gray (cod) — procedeu de codificare cu avantajul că schimbă numai un bit între poziții adiacente.

H — High : nivel logic 1.

HiZ — (High Z) : stare de impedanță mare la CI „three state“.

implementare — realizarea cu CI a unor circuite, sisteme definite prin funcții logice, tabele de tranziții ș.a.

JK — bistabil de tip JK, intrările acestui bistabil.

JKMS — Bistabil de tip JK Master Slave.

K, (Kilobit) — unitate de măsură a capacității de memorie egală cu 1024 biți.

L — Low : nivel logic 0.

LATCH — bistabil în care memorarea se face pe palierul impulsului de tact.

LD — Load : intrare de încărcare.

LED — Light Emitting Diode : dioda electroluminiscentă sau dispozitiv de afișaj conținând astfel de diode.

LTTL — Low power TTL : serie de circuite TTL cu consum redus de putere.

LSI — Large Scale Integration : integrare pe scară largă, cu peste 500 de tranzistoare pe circuit.

MOS — Metal Oxide Semiconductor : tehnologie unipolară.

MUX — MultipleXor.

MSI — Medium Scale Integration : integrare pe scară medie, cu cca 50--500 de tranzistoare pe circuit.

NAND — circuit sau funcție logică ȘI NU.

NC — NeConectat : terminal de CI neconectat.

NOR — circuit sau funcție logică SAU NU.

nMOS — tehnologie MOS cu canal n.

open collector — colector în gol : circuite la care etajul de ieșire conține un tranzistor cu colectorul conectat în exterior pentru a permite utilizatorului să-și fixeze valoarea rezistenței din colector.

OR — circuit sau funcție logică SAU.

$P_0, P_1, \dots, P_k$  — termeni canonici de tip produs. Se mai notează pentru simplificare și cu  $0, 1, \dots, k$ .

PDLT — P Depletion Load Triode : tehnologie MOS cu canal p cu sărăcire și sarcina în regim de triodă.

PELT — P Enhancement Load Triode : tehnologie MOS cu canal p cu îmbogățire de purtători și sarcina în regim de triodă.

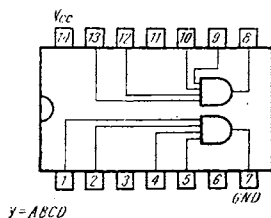






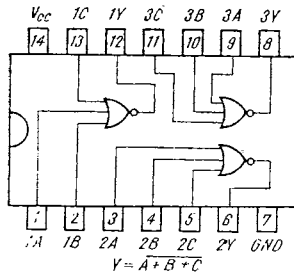
## Catalog sumar cu circuitele utilizate

4 Porți ȘI NU cu 2 intrări	7400 (CDB 400E)
4 Porți SAU NU cu 2 intrări	7402
6 Inversoare	7404 (CDB 404E)
6 Inversoare cu colectorul în gol	7405 (CDB 405E)
6 Inversoare de putere cu colectorul în gol	7406 (CDB 406E)
4 Porți ȘI cu 2 intrări	7408 (CDB 408E)
4 Porți ȘI cu 2 intrări cu colector în gol	7409 (CDB 409E)
3 Porți ȘI NU cu 3 intrări	7410 (CDB 410E)
3 Porți ȘI cu 3 intrări	7411
2 Porți ȘI NU cu 4 intrări de „trigger Schmitt“	7413 (CDB 413E)
2 Porți ȘI NU cu 4 intrări	7420
2 Porți ȘI cu 4 intrări	7421



3 Porți SAU NU cu 3 intrări

7427

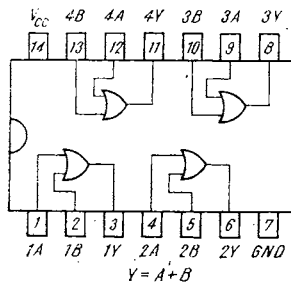


1 Poartă ȘI NU cu 8 intrări

7430  
(CDB 430E)

4 Porți SAU cu 2 intrări

7432



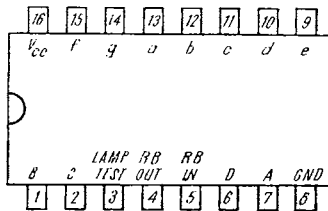
2 Porți ȘI NU de putere cu 4 intrări

7410  
(CDB 440E)

Vezi simbolul logic in fig. 2.33.

TABEL DE ADEVAR

INTRARI BCD				IESIRI ZECIMAL									
D	C	B	A	$\bar{0}$	$\bar{1}$	$\bar{2}$	$\bar{3}$	$\bar{4}$	$\bar{5}$	$\bar{6}$	$\bar{7}$	$\bar{8}$	$\bar{9}$
L	L	L	L	L	H	H	H	H	H	H	H	H	H
L	L	L	H	H	L	H	H	H	H	H	H	H	H
L	L	H	L	H	H	L	H	H	H	H	H	H	H
L	L	H	H	H	H	H	L	H	H	H	H	H	H
L	H	L	L	H	H	H	H	L	H	H	H	H	H
L	H	L	H	H	H	H	H	H	L	H	H	H	H
L	H	H	L	H	H	H	H	H	H	L	H	H	H
L	H	H	H	H	H	H	H	H	H	H	L	H	H
H	L	L	L	H	H	H	H	H	H	H	H	L	H
H	L	L	H	H	H	H	H	H	H	H	H	H	L
H	L	H	L	H	H	H	H	H	H	H	H	H	H
H	L	H	H	H	H	H	H	H	H	H	H	H	H
H	H	L	L	H	H	H	H	H	H	H	H	H	H
H	H	L	H	H	H	H	H	H	H	H	H	H	H
H	H	H	L	H	H	H	H	H	H	H	H	H	H
H	H	H	H	H	H	H	H	H	H	H	H	H	H



TABEL DE ADEVĂR

Nr. zecimal sau funcție	LT	RBI	D	C	B	A	BI/RBO a b c d e f g							
							a	b	c	d	e	f	g	
0	1	1	0	0	0	0	1	0	0	0	0	0	0	1
1	1	x	0	0	0	1	1	1	0	0	1	1	1	1
2	1	x	0	0	1	0	1	0	0	1	0	0	1	0
3	1	x	0	0	1	1	1	0	0	0	0	1	1	0
4	1	x	0	1	0	0	1	1	0	0	1	1	0	0
5	1	x	0	1	0	1	1	0	1	0	0	1	0	0
6	1	x	0	1	1	0	1	1	1	0	0	0	0	0
7	1	x	0	1	1	1	1	0	0	0	1	1	1	1
8	1	x	1	0	0	0	1	0	0	0	0	0	0	0
9	1	x	1	0	0	1	1	0	0	0	1	1	0	0
10	1	x	1	0	1	0	1	1	1	1	0	0	1	0
11	1	x	1	0	1	1	1	1	1	0	0	1	1	0
12	1	x	1	1	0	0	1	1	0	1	1	1	0	0
13	1	x	1	1	0	1	1	0	1	1	0	1	0	0
14	1	x	1	1	1	0	1	1	1	1	0	0	0	0
15	1	x	1	1	1	1	1	1	1	1	1	1	1	1
BI	x	x	x	x	x	x	0	1	1	1	1	1	1	1
RBI	1	0	0	0	0	0	0	1	1	1	1	1	1	1
LT	0	x	x	x	x	x	1	0	0	0	0	0	0	0

DECODOR BCD/7 segmente

7448

Conexiunile externe sînt identice cu cele ale 7446

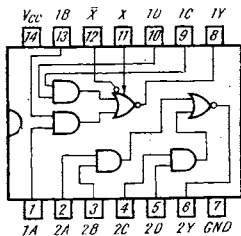
TABEL DE ADEVĂR

Nr. zecimal sau funcție	LT	RBI	D	C	B	A	BI/RBO a b c d e f g							
							a	b	c	d	e	f	g	
0	1	1	0	0	0	0	1	1	1	1	1	1	1	0
1	1	x	0	0	0	1	1	0	1	1	0	0	0	0
2	1	x	0	0	1	0	1	1	1	0	1	1	0	1
3	1	x	0	0	1	1	1	1	1	1	0	0	0	1
4	1	x	0	1	0	0	1	0	1	1	0	0	1	1
5	1	x	0	1	0	1	1	1	0	1	1	0	1	1

Nr. zecimal sau funcție	LT	RBI	D	C	B	A								
							BI/RBO	a	b	c	d	e	f	g
6	1	×	0	1	1	0	1	0	0	1	1	1	1	1
7	1	×	0	1	1	1	1	1	1	1	0	0	0	0
8	1	×	1	0	0	0	1	1	1	1	1	1	1	1
9	1	×	1	0	0	1	1	1	1	1	0	0	1	1
10	1	×	1	0	1	0	1	0	0	0	1	1	0	1
11	1	×	1	0	1	1	1	0	0	1	1	0	0	1
12	1	×	1	1	0	0	1	0	1	0	0	0	1	1
13	1	×	1	1	0	1	1	1	0	0	1	0	1	1
14	1	×	1	1	1	0	1	0	0	0	1	1	1	1
15	1	×	1	1	1	1	1	0	0	0	0	0	0	0
BI	×	×	×	×	×	×	0	0	0	0	0	0	0	0
RBI	1	0	0	0	0	0	0	0	0	0	0	0	0	0
LT	0	×	×	×	×	×	1	1	1	1	1	1	1	1

2 Porți ȘI -SAU -NU cu 2×2 intrări, expandabile

7450



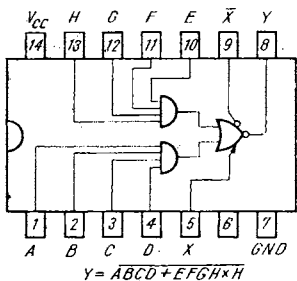
$$Y = (\overline{AB}) \cdot (C\overline{D}) + (\overline{X})$$

2 Porți ȘI SAU NU cu 2×2 intrări

7451  
(CDB 451E)

Este identic cu 7450 dar fără expandare

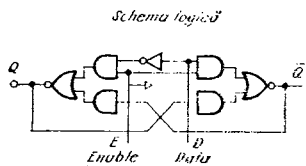
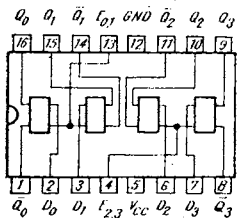
Poartă ȘI SAU NU cu 2×4 intrări, expandabil	7453 (CDB 153E)
Poartă ȘI SAU NU cu 2×4 intrări	7454 (CDB 454E)
Poartă ȘI SAU NU cu 4×2 intrări	7455



2 Expandoare cu 4 intrări	7460 (CDB 160E)
Bistabil JKMS cu ștergere și „setare“ (comută pe front negativ)	7472 (CDB 472E)
2 Bistabili JKMS cu ștergere (comută pe front negativ)	7473 (CDB 473E)
2 Bistabili D cu ștergere și „setare“ (comută pe front pozitiv)	7474 (CDB 474E)
4 Bistabili „LATCHI“	7475

TABEL DE ADEVĂR

Intrări		Ieșiri	
D	E	Q	$\bar{Q}$
L	H	L	H
H	H	H	L
X	L	Q <sub>0</sub>	$\bar{Q}_0$



Sumator binar de 4 biți	7483 (CDB 4831E)
-------------------------	---------------------

Comparator de 4 biți	7485
----------------------	------

Vezi simbolul logic în figura 2.57.

INTRĂRI							IEȘIRI		
A <sub>3</sub> , B <sub>3</sub>	A <sub>2</sub> , B <sub>2</sub>	A <sub>1</sub> , B <sub>1</sub>	A <sub>0</sub> , B <sub>0</sub>	A > B	A < B	A = B	A & B	A < B	A = B
A <sub>3</sub> > B <sub>3</sub>	x	x	x	x	x	x	H	L	L
A <sub>3</sub> < B <sub>3</sub>	x	x	x	x	x	x	L	H	L
A <sub>3</sub> = B <sub>3</sub> A <sub>2</sub> > B <sub>2</sub>	x	x	x	x	x	x	H	L	L
A <sub>3</sub> = B <sub>3</sub> A <sub>2</sub> < B <sub>2</sub>	x	x	x	x	x	x	L	H	L
A <sub>3</sub> = B <sub>3</sub> A <sub>2</sub> = B <sub>2</sub> A <sub>1</sub> > B <sub>1</sub>	x	x	x	x	x	x	H	L	L
A <sub>3</sub> = B <sub>3</sub> A <sub>2</sub> = B <sub>2</sub> A <sub>1</sub> < B <sub>1</sub>	x	x	x	x	x	x	L	H	L
A <sub>3</sub> = B <sub>3</sub> A <sub>2</sub> = B <sub>2</sub> A <sub>1</sub> = B <sub>1</sub> A <sub>0</sub> > B <sub>0</sub>	x	x	x	x	x	x	H	L	L
A <sub>3</sub> = B <sub>3</sub> A <sub>2</sub> = B <sub>2</sub> A <sub>1</sub> = B <sub>1</sub> A <sub>0</sub> < B <sub>0</sub>	x	x	x	x	x	x	L	H	L
A <sub>3</sub> = B <sub>3</sub> A <sub>2</sub> = B <sub>2</sub> A <sub>1</sub> = B <sub>1</sub> A <sub>0</sub> = B <sub>0</sub>	x	x	x	H	L	L	H	L	L
A <sub>3</sub> = B <sub>3</sub> A <sub>2</sub> = B <sub>2</sub> A <sub>1</sub> = B <sub>1</sub> A <sub>0</sub> = B <sub>0</sub>	L	H	L	L	H	L	L	H	L
A <sub>3</sub> = B <sub>3</sub> A <sub>2</sub> = B <sub>2</sub> A <sub>1</sub> = B <sub>1</sub> A <sub>0</sub> = B <sub>0</sub>	L	L	L	L	L	H	L	L	H
A <sub>3</sub> = B <sub>3</sub> A <sub>2</sub> = B <sub>2</sub> A <sub>1</sub> = B <sub>1</sub> A <sub>0</sub> = B <sub>0</sub>	L	L	L	L	L	L	L	L	N
A <sub>3</sub> = B <sub>3</sub> A <sub>2</sub> = B <sub>2</sub> A <sub>1</sub> = B <sub>1</sub> A <sub>0</sub> = B <sub>0</sub>	L	L	L	L	L	L	H	H	L

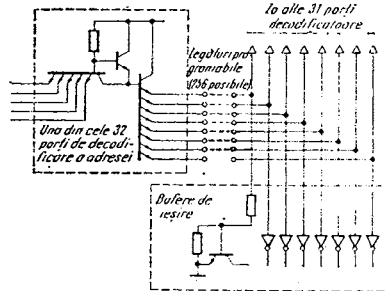
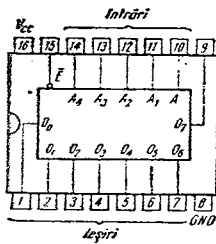
4 Porți SAU EXCLUSIV cu 2 intrări	7486 (CDB 4861E)
-----------------------------------	---------------------

ROM de 256 BIȚI (32 × 8)	7488
--------------------------	------

TABEL DE SELECTARE CUVÎNT

CUVÎNT	INTRARE				
	A <sub>4</sub>	A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>
0	L	L	L	L	L
1	L	L	L	L	H
2	L	L	L	H	L
3	L	L	L	H	H
⋮	⋮	⋮	⋮	⋮	⋮
28	H	H	H	L	L
29	H	H	H	L	H
30	H	H	H	H	L
31	H	H	H	H	H

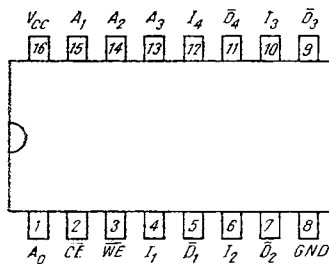




RAM bipolar de 64 de biți (16 × 4)

7489

Mod	CE	WE	$I_n$	$D_n$
Citește	0	1	×	datele memorate
Serie 0	0	0	0	1
Serie 1	0	0	1	0
Neactivat	1	×	×	1



Numărător binar de 4 biți

7493  
(CDB 493E)

Registru de deplasare de 4 biți cu acces paralel

7495  
(CDB 495E)

Monostabil

74121  
(CDB 4121E)

Vezi simbolul logic în fig. 3.11

Fără componente externe, durata impulsului la ieșire este de 30 ns (tipic).

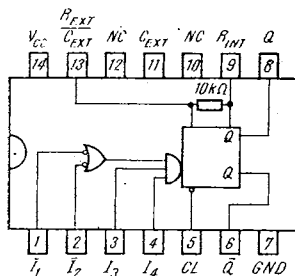
Se poate utiliza rezistența internă (2 kΩ, tipic) pentru a forma constanta de timp cu capacitatea externă, conectând 9 la V<sub>CC</sub>.

Pentru  $R_t = 2 \text{ k}\Omega \div 40 \text{ k}\Omega$  și  $C_t = 10 \text{ pF} \div 10 \text{ }\mu\text{F}$ , durata impulsului de ieșire se calculează cu formula :

$$T = C_t R_t \ln 2.$$

Monostabil recomutabil

74122



Pentru a utiliza rezistența internă se conectează R<sub>INT</sub> la V<sub>CC</sub>.  
Condensatorul extern se cuplează între C<sub>EXT</sub> și R<sub>EXT</sub>/C<sub>EXT</sub> (plusul).

Codificator cu priorități

74148

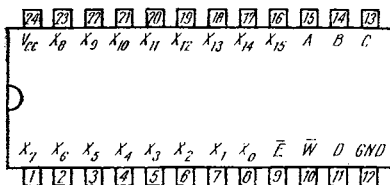
Vezi simbolul logic în figura 2.43

TABEL DE ADEVĂR

INTRĂRI									IEȘIRI				
$\bar{E}$	$\bar{I}_0$	$\bar{I}_1$	$\bar{I}_2$	$\bar{I}_3$	$\bar{I}_4$	$\bar{I}_5$	$\bar{I}_6$	$\bar{I}_7$	$\bar{A}_2$	$\bar{A}_1$	$\bar{A}_0$	GS	EO
H	X	X	X	X	X	X	X	X	H	H	H	H	H
L	H	H	H	H	H	H	H	H	H	H	H	H	L
L	X	X	X	X	X	X	X	L	L	L	L	L	H
L	X	X	X	X	X	X	L	H	L	L	H	L	H
L	X	X	X	X	L	H	H	H	L	H	L	L	H
L	X	X	L	H	H	H	H	H	H	L	L	L	H
L	X	X	L	H	H	H	H	H	H	L	H	L	H
L	X	L	H	H	H	H	H	H	H	H	L	L	H
L	L	H	H	H	H	H	H	H	H	H	L	L	H

Multiplexor de 16 căi

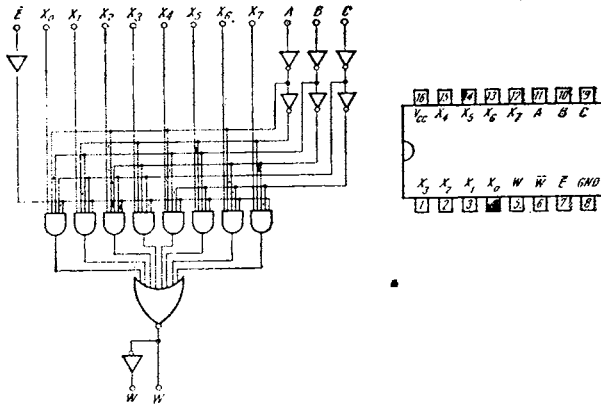
74150



$$W = E(\overline{A}BCDX_0 + \overline{A}BCDN_1 + \overline{A}\overline{B}CDN_2 + \overline{A}\overline{B}CDN_3 + \overline{A}\overline{B}CDN_4 + \overline{A}\overline{B}CDX_5 + \overline{A}BCDX_6 + \overline{A}BCDX_7 + \overline{A}BCDN_8 + \overline{A}BCDN_9 + \overline{A}BCDN_{10} + \overline{A}BCDX_{11} + \overline{A}BCDX_{12} + \overline{A}BCDN_{13} + \overline{A}BCDN_{14} + \overline{A}BCDX_{15})$$

Multiplexor de 8 căi

74151

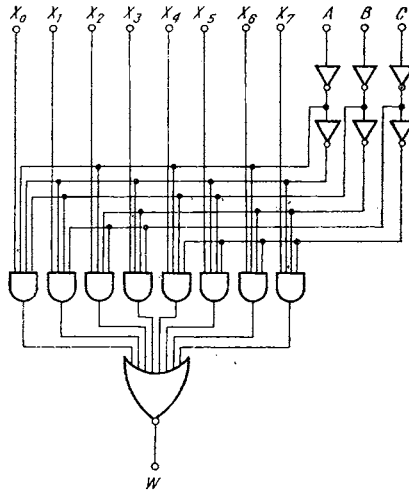


$$W = E(\bar{A}\bar{B}\bar{C}X_0 + \bar{A}\bar{B}CX_1 + \bar{A}B\bar{C}X_2 + \bar{A}BCX_3 + A\bar{B}\bar{C}X_4 + A\bar{B}CX_5 + AB\bar{C}X_6 + ABCX_7)$$

Multiplexor de 8 căi

74152

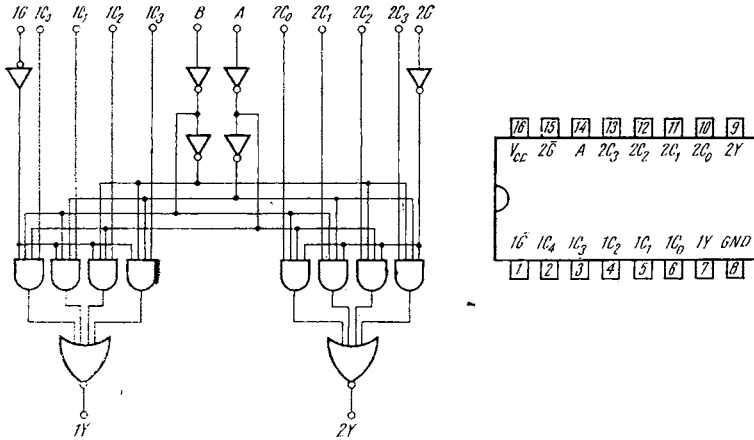
Vezi simbolul logic în fig. 2.36



$$\bar{W} = \bar{A}\bar{B}\bar{C}X_0 + \bar{A}\bar{B}CX_1 + \bar{A}B\bar{C}X_2 + \bar{A}BCX_3 + A\bar{B}\bar{C}X_4 + A\bar{B}CX_5 + AB\bar{C}X_6 + ABCX_7$$

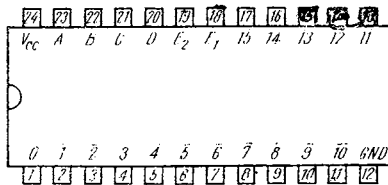
2 Multiplexoare de 4 căi

7-1153



DEMULTIPLER de 4 biți

7-1154



TABEL DE ADEVĂR

INTRĂRI						IEȘIRI																
E <sub>1</sub>	E <sub>2</sub>	D	C	B	A	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
L	L	L	L	L	L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	L	L	H	H	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	L	H	L	H	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	L	H	H	H	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	H	L	L	H	H	H	H	L	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	H	H	L	H	H	H	H	H	L	H	H	H	H	H	H	H	H	H	H	H
L	L	L	H	H	H	H	H	H	H	H	H	L	H	H	H	H	H	H	H	H	H	H
L	L	H	L	L	L	H	H	H	H	H	H	H	L	H	H	H	H	H	H	H	H	H
L	L	H	L	L	H	H	H	H	H	H	H	H	H	L	H	H	H	H	H	H	H	H
L	L	H	L	H	L	H	H	H	H	H	H	H	H	H	L	H	H	H	H	H	H	H
L	L	H	L	H	H	H	H	H	H	H	H	H	H	H	L	H	H	H	H	H	H	H
L	L	H	H	L	L	H	H	H	H	H	H	H	H	H	H	L	H	H	H	H	H	H
L	L	H	H	L	H	H	H	H	H	H	H	H	H	H	H	L	H	H	H	H	H	H
L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	L	H	H	H	H	H	H
L	H	×	×	×	×	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
H	L	×	×	×	×	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
H	H	×	×	×	×	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H

2 Demultiplexoare de 2 biți

74155

(1 Demultiplexor de 3 biți)

Vezi fig. 2.26

TABEL DE ADEVĂR (1)

2 DEMULTIPLINOARE DE 2 BIȚI

INTRĂRI				IEȘIRI			
SELECTARE STROBARE DATE				1Y0	1Y1	1Y2	1Y3
B	A	1G	1C				
×	×	H	×	H	H	H	H
L	L	L	H	L	H	H	H
L	H	L	H	H	L	H	H
H	L	L	H	H	H	L	H
H	H	L	H	H	H	H	L
×	×	×	L	H	H	H	H

(2)

INTRĂRI				IEȘIRI			
SELECTARE		STROBARE DATE					
B	A	2G	2C	2Y0	2Y1	2Y2	2Y3
X	X	H	X	H	H	H	H
L	L	L	L	L	H	H	H
L	H	L	L	H	L	H	H
H	L	L	L	H	H	L	H
H	H	L	L	H	H	H	L
X	X	H	H	H	H	H	H

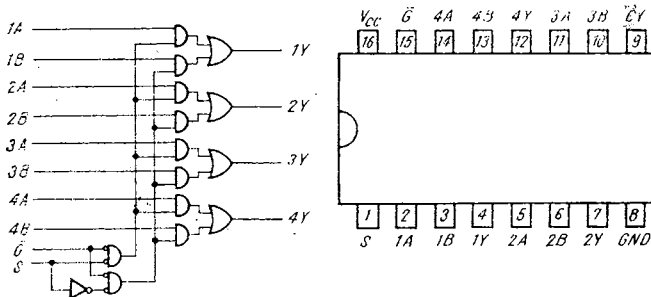
DEMULTIPLEXOR DE 3 BIȚI

(3)

INTRĂRI				IEȘIRI							
SELECTARE			STROBARE SAU DATE	(0) 2Y0	(1) 2Y1	(2) 2Y2	(3) 2Y3	(4) 1Y0	(5) 1Y1	(6) Y12	(7) 1Y3
C	B	A	E = 1G = 2G								
X	X	X	H	H	H	H	H	H	H	H	H
L	L	L	L	L	H	H	H	H	H	H	H
L	L	H	L	H	L	H	H	H	H	H	H
L	H	L	L	H	H	L	H	H	H	H	H
L	H	H	L	H	H	H	L	H	H	H	H
H	L	L	L	H	H	H	H	L	H	H	H
H	L	H	L	H	H	H	H	H	L	H	H
H	H	L	L	H	H	H	H	H	H	L	H
H	H	H	L	H	H	H	H	H	H	H	L

4 Multiplexoare neinversoare de câte 2 căi

74157



TABEL DE ADEVĂR

$\bar{G}$	S	A	B	Y
H	×	×	×	L
L	L	L	×	L
L	L	H	×	H
L	H	×	L	L
L	H	×	H	H

4 Multiplexoare inversoare de cite 2 căi

74158

TABEL DE ADEVĂR

$\bar{G}$	S	A	B	Y
H	×	×	×	H
L	L	L	×	H
L	L	H	×	L
L	H	×	L	H
L	H	×	H	L

Generator/Detector de paritate pe 8 biți

74180

Vezi problema 2.74

Unitate logico-aritmetică

74181

Vezi simbolul logic din figura 2.66<sup>\*</sup>



TABELUL 1

SELECȚIA	DATE ACTIVE PE UNU		
	M = H FUNCȚIE LOGICĂ	M = L: OPERAȚII LOGICE	
		C <sub>n</sub> = H (FĂRĂ TRANSPORT)	C <sub>n</sub> = L (CU TRANSPORT)
S <sub>3</sub> S <sub>2</sub> S <sub>1</sub> S <sub>0</sub>			
L L L L	$F = \bar{A}$	$F = A$	$F = A \text{ PLUS } 1$
L L L H	$F = \bar{A} + B$	$F = A + B$	$F = (A + B) \text{ PLUS } 1$
L L H L	$F = \overline{AB}$	$F = A + \bar{B}$	$F = (A + \bar{B}) \text{ PLUS } 1$
L L H H	$F = 0$	$F = \text{MINUS } 1 \text{ (COM-PLEMENTFAȚĂ DE } 2)$	$F = \text{ZERO}$
L H L L	$F = \overline{AB}$	$F = A \text{ PLUS } \overline{AB}$	$F = A \text{ PLUS } \overline{AB}$ PLUS 1
L H L H	$F = \bar{B}$	$F = (A + B) \text{ PLUS } \overline{AB}$	$F = (A - B) \text{ PLUS } \overline{AB}$ PLUS 1
L H H L	$F = A \oplus B$	$F = A \text{ MINUS } B$ MINUS 1	$F = A \text{ MINUS } B$
L H H H	$F = \overline{AB}$	$F = \overline{AB} \text{ MINUS } 1$	$F = \overline{AB}$
H L L L	$F = \bar{A} + B$	$F = A \text{ PLUS } AB$	$F = A \text{ PLUS } AB$ PLUS 1
H L L H	$F = \overline{A \oplus B}$	$F = A \text{ PLUS } B$	$F = A \text{ PLUS } B$ PLUS 1
H L H L	$F = B$	$F = (A + \bar{B}) \text{ PLUS } AB$	$F = (A + \bar{B}) \text{ PLUS } AB$ PLUS 1
H L H H	$F = AB$	$F = AB \text{ MINUS } 1$	$F = AB$
H H L L	$F = 1$	$F = A \text{ PLUS } A (+)$	$F = A \text{ PLUS } A$ PLUS 1
H H L H	$F = A + \bar{B}$	$F = (A + B) \text{ PLUS } A$	$F = (A + B) \text{ PLUS } A$ PLUS 1
H H H L	$F = A + B$	$F = (A + \bar{B}) \text{ PLUS } A$	$F = (A + \bar{B}) \text{ PLUS } A$ PLUS 1
H H H H	$F = A$	$F = A \text{ MINUS } 1$	$F = A$

\* Fiecare bit este deplasat în următoarea poziție semnificativă

TABELUL 2

SELECȚIA	DATE ACTIVE PE ZERO		
	M = H FUNCȚIE LOGICĂ	M = L: OPERAȚII ARITMETICE	
		C <sub>n</sub> = L (FĂRĂ TRANSPORT)	C <sub>n</sub> = H (CU TRANSPORT)
S <sub>3</sub> S <sub>2</sub> S <sub>1</sub> S <sub>0</sub>			
L L L L	$F = \bar{A}$	$F = A \text{ MINUS } 1$	$F = A$
L L L H	$F = \overline{AB}$	$F = AB \text{ MINUS } 1$	$F = AB$
L L H L	$F = \bar{A} + B$	$F = \overline{AB} \text{ MINUS } 1$	$F = \overline{AB}$

SELECȚIA	DATE ACTIVE PE ZERO		
	M=H FUNȚIE LOGICĂ	M = L : OPERAȚII ARITMETICE	
		C = L : (FĂRĂ TRANSPORT)	C = H : (CU TRANSPORT)
S <sub>3</sub> S <sub>2</sub> S <sub>1</sub> S <sub>0</sub>			
L L H H	F = 1	F = MINUS 1 (2 COMPI.)	F = ZERO
L H L L	F = $\overline{A + B}$	F = A PLUS ( $\overline{A + B}$ )	F = A PLUS (A + $\overline{B}$ ) PLUS 1
L H L H	F = $\overline{B}$	F = AB PLUS(A + $\overline{B}$ )	F = AB PLUS (A + $\overline{B}$ ) PLUS 1
L H H L	F = $\overline{A \oplus B}$	F = A MINUS B MINUS 1	F = A MINUS B
L H H H	F = $\overline{A + \overline{B}}$	F = A + $\overline{B}$	F = (A + $\overline{B}$ ) PLUS 1
H L L L	F = $\overline{AB}$	F = A PLUS (A + B)	F = A PLUS (A + B) PLUS 1
H L L H	F = A $\oplus$ B	F = A PLUS B	F = A PLUS B PLUS 1
H L H L	F = B	F = $\overline{AB}$ PLUS(A + B)	F = $\overline{AB}$ PLUS(A + B) PLUS 1
H L H H	F = A + B	F = A + B	F = (A + B) PLUS 1
H H L L	F = 0	F = A PLUS A + )	F = A PLUS A PLUS 1
H H L H	F = $\overline{AB}$	F = AB PLUS A	F = AB PLUS A PLUS 1
H H H L	F = AB	F = $\overline{AB}$ PLUS A	F = $\overline{AB}$ PLUS A PLUS 1
H H H H	F = A	F = A	F = A PLUS 1

RAM Schottky «three state» de 64 de biți (16 × 4)

74S189

Conexiunile externe sînt identice cu ale 7489

Tablul de adevăr es'e :

MOD	$\overline{CE}$	$\overline{WE}$	$I_n$	$\overline{D}_n$
Citește	0	1	×	Datele memorate
Serie 0	0	0	0	Hi-Z
Serie 1	0	0	1	Hi-Z
Neactivat	1	×	×	Hi-Z

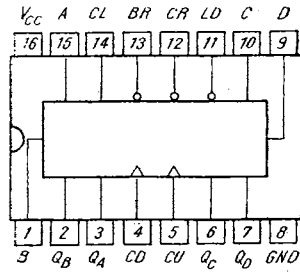
Numărător zecimal sincron reversibil

74192  
(CDB 4192E)

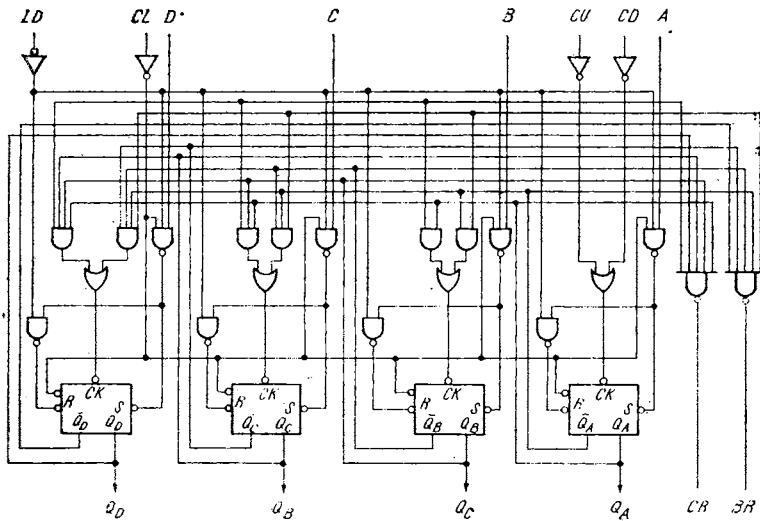
Numărător binar sincron de 4 biți reversibil

74193  
(CDB 4193E)

Conexiuni externe :



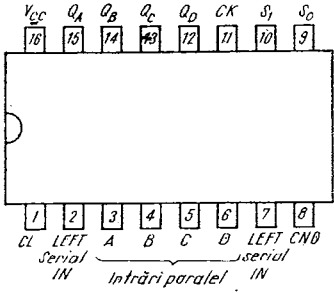
Schema logică internă :



Parametru	De la intrarea :	la ieșirea :	Min	Tipic	Max.
$f_{Max}$			25 MHz	32 MHz	
$t_{set\ up}$				11 ns	20 ns
$t_{PLH}$	CU	CR		17 ns	26 ns
$t_{PHL}$				16 ns	24 ns
$t_{PLH}$	CD	BR		16 ns	24 ns
$t_{PHL}$				25 ns	38 ns
$t_{PLN}$	CD sau CU	Q		16 ns	24 ns
$t_{PHL}$				31 ns	47 ns

Note: Aducerea la zero are prioritate asupra intrărilor de încărcare și numărare. La numărarea directă este necesar ca CD = 1, iar la numărarea inversă, CU = 1.

Registru de deplasare bidirecțional de 4 biți 74194



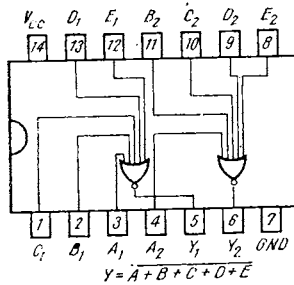
TABEL DE ADEVĂR

CLEAR (set- genc)	MOD		CLOCK (CEAS)	INTRĂRI				IEȘURI			
	S1	S0		SERIE		PARALEL		QA	QB	QC	QD
				LEFT (stînga)	RIGHT (dreapta)	A	B				
L	x	x	x	x	x	x	x	L	L	L	L
H	x	x	l	x	x	x	x	Q <sub>A0</sub>	Q <sub>B0</sub>	Q <sub>C0</sub>	Q <sub>D0</sub>
H	H	H	↑	x	x	a	b	a	b	c	d
H	L	H	↑	x	H	x	x	l	Q <sub>An</sub>	Q <sub>Bn</sub>	Q <sub>Cn</sub>
H	L	H	↑	x	L	x	x	l	Q <sub>An</sub>	Q <sub>Bn</sub>	Q <sub>Cn</sub>
H	H	L	↑	l	x	x	x	Q <sub>An</sub>	Q <sub>Cn</sub>	Q <sub>Dn</sub>	l
H	H	L	↑	l	x	x	x	Q <sub>Bn</sub>	Q <sub>Cn</sub>	Q <sub>Dn</sub>	l
H	L	L	x	x	x	x	x	Q <sub>A0</sub>	Q <sub>B0</sub>	Q <sub>C0</sub>	Q <sub>D0</sub>

Notațiile sînt aceleași ca la 7495.

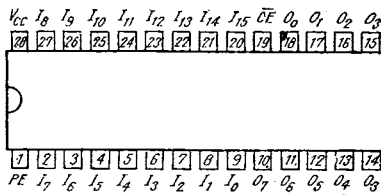
2 Porți SAU NU cu 5 intrări

74260

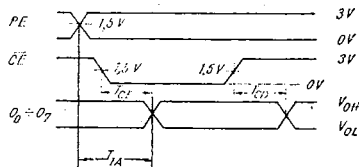


FPLA

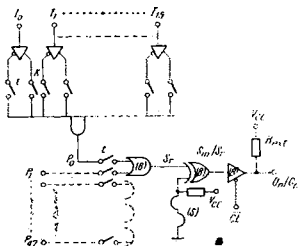
82S100/101



Ciclul de citire



Schema logică echivalentă pentru o ieșire



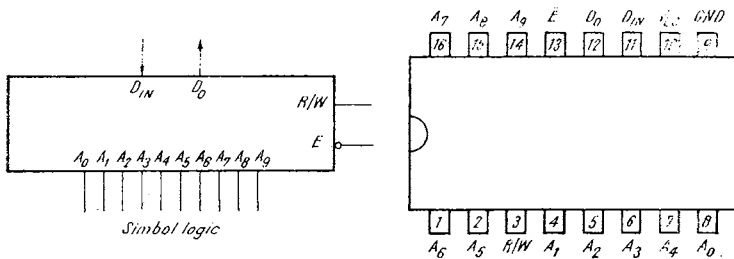
Pentru fiecare din cele 8 ieșiri se poate obține  $O_p$  (activ H) sau  $O_p$  (activ L), dar nu amândouă, programarea efectuându-se cu ajutorul siguranței fuzibile S.

RAM MOS static de 1 K × 1 biți

2102

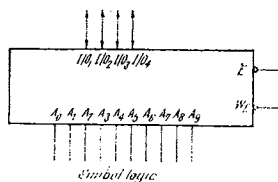
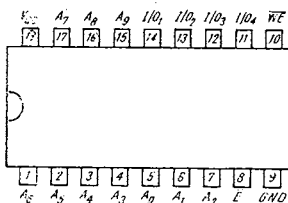
TABEL DE ADEVĂR

MOD	$\bar{E}$	R/W	DIN	$D_o$
Neactivat	H	×	×	H <sub>i</sub> - Z
Serie 0	L	L	L	L
Serie 1	L	L	H	H
Citeste	L	H	×	$D_a$



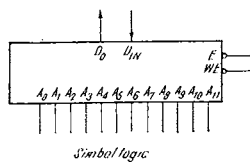
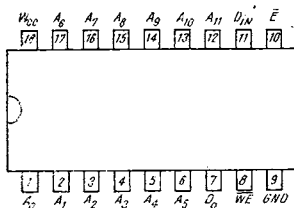
RAM MOS static de 1 K × 4 biți

2114



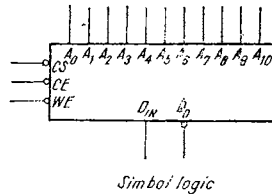
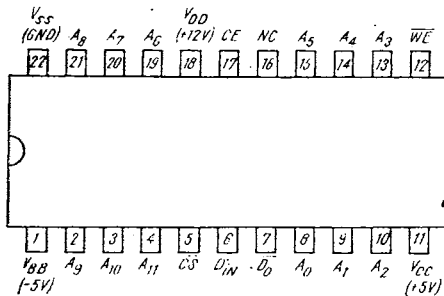
RAM MOS static de 4096 × 1 biți

2147

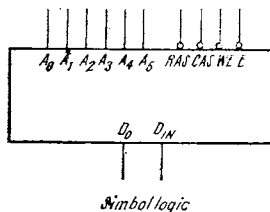
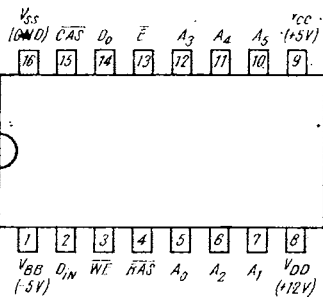


TABEL DE ADEVĂR

MOD	$\overline{E}$	$\overline{WE}$	$D_n$
Neactivat	H	×	Hi - Z
Serie	L	L	Hi - Z
Citeste	L	H	$D_0$

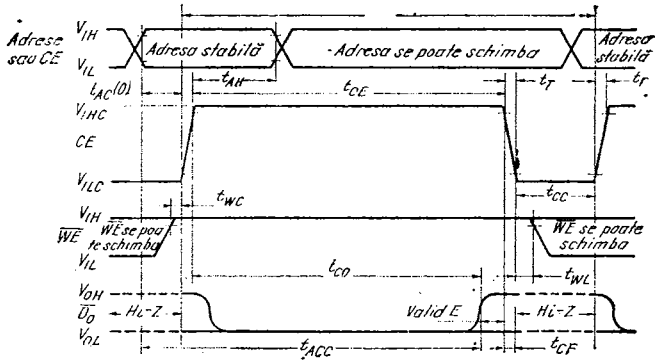


$\overline{A_0} \dots \overline{A_5}$	Intrări de adresă	$\overline{WE}$	Permite scrierea
$\overline{CAS}$	Strobul adresei de coloană	$V_{BB}$	Sursa -5 V
$\overline{E}$	Enable	$V_{CC}$	Sursa +5 V
$D_{IN}$	Intrare de date	$V_{DD}$	Sursă +12V
$D_0$	Ieșire de date	$V_{SS}$	Masa (GND)
$\overline{RAS}$	Strobul adresei de linie		

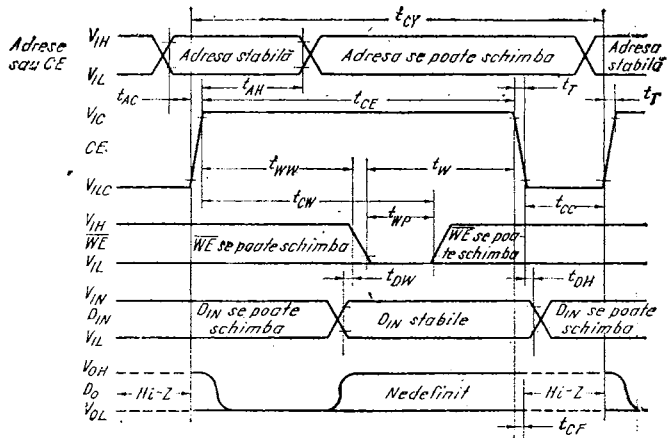




Ciclu de citire și reîmprospătare



Ciclu de scriere



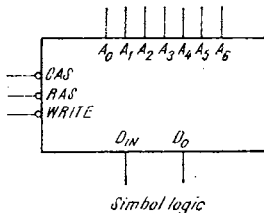
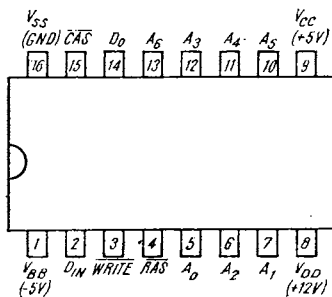
Caracteristici electrice

Parametru	Simbol	2107B		2107B-1	
		Min.	Max.	Min.	Max.
Ciclu de citire					
Timp de ciclu	$t_{CY}$	400		470	
Durata pulsului CE	$t_{CE}$	230	4000	300	4000
Întârziere la apariția datelor față de CE	$t_{CO}$		180		250
Timp de acces	$t_{ACC}$		200		270
Timp între CE și dispariția $\overline{WE}$	$t_{WL}$	0		0	
Timp de «set-up» al $\overline{WE}$ față de CE	$t_{WC}$	0		0	
Ciclu de scriere					
Timp de ciclu	$t_{CY}$	400		470	
Durata pulsului CE	$t_{CE}$	230	4000	300	4000
Timp între $\overline{WE}$ și dispariția CE	$t_W$	125		150	
Timp între CE și dispariția lui $\overline{WE}$	$t_{CW}$	150		150	
Timp de «set-up» al datelor față de $\overline{WE}$	$t_{DW}$	0		0	
Timp de «hold» al datelor la intrare	$t_{DH}$	0		0	
Durata pulsului $\overline{WE}$	$t_{WP}$	50		50	
Timp de «set-up» al CE față de $\overline{WE}$	$t_{WC}$	75		75	

Timpii sînt dați în ns.

RAM MOS dinamic de 16 K × 1 biți

2117



Ciclu de citire (vezi fig. 3.96)

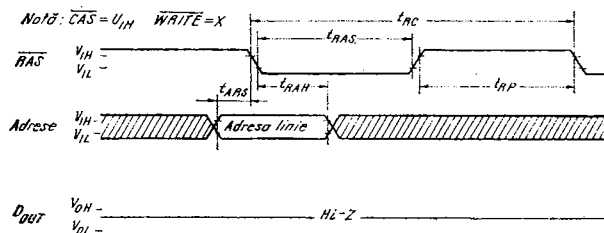
Ciclu de scriere (Early Write) (vezi fig. 3.98)

Ciclu «citește-modifică-serie» (vezi fig. 3.97)

Ciclu de citire în mod «pagină» (vezi fig. 3.99)

Ciclu de scriere în «mod pagină» (Early Write) (vezi fig. 3.100)

Ciclu de «refresh» (Ras-only)



### Caracteristici electrice

Parametru	Simbol	Z-6116-2		Z-6116-3	
		Min.	Max.	Min.	Max.
Ciclu de scriere sau citire	$t_{RC}$	375		375	
Ciclu de citește-serie	$t_{RWC}$	375		375	
Ciclu mod pagină	$t_{PC}$	170		225	
Timp de acces de la $\overline{RAS}$	$t_{RAC}$		150		200
Timp de acces de la $\overline{CAS}$	$t_{CA}$		100		135
Întârzierea la închiderea ieșirii	$t_{OPP}$	0	40	0	50
Timp de front	$t_T$	3	35	3	50
Timp de preîncălzire al $\overline{RAS}$	$t_{RP}$	100		120	
Durata pulsului de $\overline{RAS}$	$t_{RAS}$	150	10 000	200	10 000
Timp de «hold» al $\overline{RAS}$	$t_{RSU}$	100		135	
Timp de «hold» al $\overline{CAS}$	$t_{CSU}$	150		200	

Parametru	Simbol	Z-6116-2		Z-6116-3	
		Min.	Max.	Min.	Max.
Durata pulsului de $\overline{\text{CAS}}$	$t_{\text{CAS}}$	100	10 000	135	10 000
Timp de întârziere între $\overline{\text{RAS}}$ și $\overline{\text{CAS}}$	$t_{\text{RCD}}$	20	50	25	65
Timpul de preîncărcare al $\overline{\text{RAS}}$ față de $\overline{\text{CAS}}$	$t_{\text{CRP}}$	-20		-20	
Timp de «set-up» pentru adresele de rând	$t_{\text{ASR}}$	0		0	
Timp de «hold» pentru adresele de rând	$t_{\text{RAH}}$	20		25	
Timp de «set-up» pentru adresele de coloană	$t_{\text{ASC}}$	-10		-10	
Timp de «hold» pentru adresele de coloană	$t_{\text{CAH}}$	45		55	
Timp de «set-up» pentru citire	$t_{\text{RCS}}$	0		0	
Timp de «hold» pentru citire	$t_{\text{RCS}}$	0		0	
Timp de «hold» pentru scriere ( $\overline{\text{WRITE}}$ )	$t_{\text{WCH}}$	45		55	
Durata pulsului de scriere ( $\overline{\text{WRITE}}$ )	$t_{\text{WP}}$	45		55	
Timp între $\overline{\text{WRITE}}$ și dispariția $\overline{\text{RAS}}$	$t_{\text{RWL}}$	60		80	
Timp între $\overline{\text{WRITE}}$ și dispariția $\overline{\text{CAS}}$	$t_{\text{CWL}}$	60		80	
Timp de «set-up» pentru date	$t_{\text{DS}}$	0		0	
Timp de «hold» pentru date	$t_{\text{DH}}$	45		55	
Timp de preîncărcare al $\overline{\text{CAS}}$ (la mod pag.)	$t_{\text{CP}}$	60		80	
Perioada «refresh»-ului	$t_{\text{REF}}$		2 ms		2 ms
Timp de «set-up» pentru $\overline{\text{WRITE}}$	$t_{\text{WCS}}$	-20		-20	
Întârziere între $\overline{\text{CAS}}$ și $\overline{\text{WRITE}}$	$t_{\text{CWD}}$	70		90	
Întârziere între $\overline{\text{RAS}}$ și $\overline{\text{WRITE}}$	$t_{\text{RWD}}$	120		160	

Timpul în afară de  $t_{\text{REF}}$  sunt dați în ns.

## Microprocesorul Intel 8080

Pentru programator, microprocesorul se prezintă astfel:

- 7 registre de lucru
- memorie
- numărător de adrese al instrucțiunilor
- mareajul de stivă
- input/output

### Registrele de lucru

Intel 8080 are un acumulator de 8 biți plus 6 alte registre. Acumulatorul este referit prin litera A, celelalte fiind B, C, D, E, H, L. Unele instrucțiuni lucrează cu perechi de registre, referite prin literele B, D, H, PSW: (BC), (DE), (HL), (biți de condiție, A).

**Memoria**, din punct de vedere al programatorului este un șir de octeți. Capacitatea maximă de adresare: 64 Kocteți (registru de adrese are 16 biți).

**Numărătorul de adrese al instrucțiunilor** este un registru de 16 biți accesibil prin program; conținutul lui indică adresa instrucțiunii ce urmează a fi executate.

**Mareajul pentru stivă** este un registru de 16 biți accesibil prin program.

**Input/output** — maximul 256 periferice; fiecareia i se atașează un număr între 0 și 255.

### Setul de instrucțiuni

În continuare se vor folosi următoarele notații:

- $\langle B_2 \rangle$  — al doilea octet al instrucțiunii  
 $\langle B_3 \rangle$  — al treilea octet al instrucțiunii
- r — unul din registrele A, B, ..., L
- C — unul din biții de condiție
- M — locație de memorie indicată de conținutul registrelor H și L  
(adr) — adresa unei locații de memorie indicată prin octeții 2 și 3 ai unei instrucțiuni
- PC — numărător de adrese (PCH primul octet  
PCL al doilea octet)
- rp — pereche de registre C {rh — registrul superior dintr-o pereche  
rl — registrul inferior dintr-o pereche
- SSS — registru sursă pentru date  
DDD — registru destinație pentru date

Numele	DDD sau SSS
A	111
B	000
C	001
D	010
E	011
H	100
L	101

XXX -- nu se ia în considerație

#### Semnalizările de condiție

Z -- zero, S -- semn, P -- paritate, CY -- carry, AC -- carry auxiliar  
 ( ) -- conținutul locației de memorie sau al registrului din paranteză  
 ← -- Transfer. De ex. (A) ← (B)  
 Conținutul registrului B trece în A ; B rămâne de obicei neafectat

A -- ȘI logic  
 V -- SAU Exclusiv  
 A -- SAU  
 + -- Adunare  
 - -- Scădere prin complementul lui 2  
 \* -- Înmulțire  
 ↔ -- Schimbare a conținutului  
 Complementul față de 1

NNN -- Reprezentarea numărului de restart

#### Instrucțiuni de transfer de date

Instrucțiunile din acest grup transferă date la și de la registre și memorie. Nu sînt afectate semnalizările de condiție (condition flags).

Mnemonie	Număr de octeți	Semnificație
MOV r <sub>1</sub> , r <sub>2</sub>	1	(r <sub>1</sub> ) ← (r <sub>2</sub> )
MOV r, M	1	(r) ← (M)
MOV M, r	1	(M) ← (r)
MVl r	2	(r) ← B <sub>2</sub>
<B <sub>2</sub> >		
MVI M	2	(M) ← <B <sub>2</sub> >
<B <sub>2</sub> >		
LXI rp	3	(rh) ← <B <sub>3</sub> >
<B <sub>2</sub> >		(rl) ← <B <sub>2</sub> >
<B <sub>3</sub> >		Octetul al treilea al instrucțiunii este trecut în registrul de ordin superior al perechii de registre rp ; octetul al doilea, în cel de ordin inferior
LDA adr	3	(A) ← ((adr)) Conținutul locației de memorie indicată prin adresă (doi octeți) este transferat în registrul A
STA adr	3	((adr)) ← (A)
LHLP adr	3	(L) ← ((adr)) (H) ← ((adr II)) Conținutul locației de memorie indicată prin adresă (2 octeți) este transferat în registrul L. În II este transferat conținutul locației consecutive

Mnemonică	Număr de octeți	Semnificație
SHLD	3	$((adr)) \leftarrow (L)$ $((adr H)) \leftarrow (H)$
LDAX rp	1	$(A) \leftarrow ((rp))$ Adresa este în perechea de registre rp. Nu pot fi indicate decât rp = B(B, C) și rp = D(D, E)
STAX rp	1	$((rp)) \leftarrow (A)$ La fel ca mai sus
XCHG	1	$(H) \leftrightarrow (D)$ $(L) \leftrightarrow (E)$ Are loc o schimbare a conținutului registrelor cu un simplu transfer

### Instrucțiuni aritmetice

Se execută asupra datelor din registre și memorie; sint afectate semnalizările de condiție.

Scăderile sint făcute prin complementul lui 2, punind

carry = 1 pentru a indica borrow

carry = 0 pentru lipsa lui

ADD r	1	$(A) \leftarrow (A) + (r)$
ADD M	2	$(A) \leftarrow (A) + (M)$
ADI	2	$(A) \leftarrow (A) + \langle B_2 \rangle$
$\langle B_2 \rangle$		
ADC r	1	$(A) \leftarrow (A) + (r) + (\text{carry})$
ADC M	2	$(A) \leftarrow (A) + (M) + (\text{carry})$
ACI	2	$(A) \leftarrow (A) + (B_2) + (\text{carry})$
$\langle B_2 \rangle$		
SUB r	1	$(A) \leftarrow (A) - (r)$
SUB M	2	$(A) \leftarrow (A) - (M)$
SUI	2	$(A) \leftarrow (A) - (B_2)$
$\langle B_2 \rangle$		
SBB r	1	$(A) \leftarrow (A) - (r) - (\text{carry})$
SBB M	2	$(A) \leftarrow (A) - (M) - (\text{carry})$
SBI	2	$(A) \leftarrow (A) - (B_2) - (\text{carry})$
$\langle B_2 \rangle$		
INR r	1	$(r) \leftarrow (r) + 1$ Nu este afectată semnalizarea de carry
INR M	3	$(M) \leftarrow (M) + 1$ Nu este afectat carry
DCR r	1	$(r) \leftarrow (r) - 1$ Nu este afectat carry
DCR M	3	$(M) \leftarrow (M) - 1$ La fel ca mai sus
INX rp	1	$(rh) (rl) \leftarrow (rh)(rl) + 1$ Conținutul perechii de registre este incrementat cu o unitate. Nu este afectată nici o semnalizare de condiție.
DCX rp	1	$(rh) (rl) \leftarrow (rh) (rl) - 1$ La fel ca mai sus
DAD rp	3	$(H) (L) \leftarrow (H) (L) + (rh) (rl)$ Este afectat numai carry
DAA	1	Corecția zecimală a acumulatorului, pentru a forma numere BCZ de cîte 4 biți. Toate semnalizările sint afectate.

### Instrucțiuni logice

Execută operații în logică booleană asupra datelor din registre și memorie și asupra semnalizărilor de condiție.

În general sint afectate toate semnalizările, în urma acestor instrucțiuni.

Mnemonică	Număr de octeți	Semnificație
ANA r	1	$(A) \leftarrow (A) \wedge (r)$ Se șterge semnalizarea de carry
ANA M	2	$(A) \leftarrow (A) \wedge (M)$ La fel ca mai sus se șterge carry
ANi <B <sub>2</sub> >	2	$(A) \leftarrow (A) \wedge (B_2)$ Se șterg semnalizările de carry și carry auxiliar
NRA r	1	$(A) \leftarrow (A) \vee (r)$ Se șterg carry și carry auxiliar
NRA M	2	$(A) \leftarrow (A) \vee (M)$ La fel ca mai sus
NRi <B <sub>2</sub> >	2	$(A) \leftarrow (A) \vee (B_2)$ La fel ca mai sus
ORA r	1	$(A) \leftarrow (A) \vee (r)$
ORA M	2	$(A) \leftarrow (A) \vee (M)$
ORI <B <sub>2</sub> >	2	$(A) \leftarrow (A) \vee (B_2)$
CMP r	1	$(A) - (r)$ Conținutul lui A este comparat cu cel al registrului r, semnalizările de condiție fiind puse ca rezultat al unei scăderi. A rămâne neschimbat
CMP M	2	$(A) - (M)$ La fel ca mai sus
CPI <B <sub>2</sub> >	2	$(A) - (B_2)$ La fel ca mai sus
RLC	1	$(A_{n+1}) \leftarrow (A_n); (A_0) \leftarrow (A_7)$ $(carry) \leftarrow (A_7)$ Acumulatorului i se rotește conținutul la stînga cu o poziție Numai carry este afectat
RRC	1	$(A_n) \leftarrow (A_{n-1}); (A_7) \leftarrow (A_0)$ $(carry) \leftarrow (A_0)$ Rotație la analogă la dreapta cu o poziție
RAL	1	$(A_{n+1}) \leftarrow (A_n) (carry) \leftarrow (A_7)$ $(A_0) \leftarrow (carry)$ Rotație la stînga prin bitul de semnalizare carry Numai carry este afectat
RAR	1	$(A_n) \leftarrow (A_{n+1}) (carry) \leftarrow (A_0)$ $(A_7) \leftarrow (carry)$ Rotație analogă la dreapta
CMA	1	$(\overline{A}) \leftarrow (\overline{A})$ Nici o semnalizare de condiție nu este afectată
CMC	1	$(carry) \leftarrow (\overline{carry})$
STC	1	$(carry) \leftarrow 1$

### Instrucțiuni de branșare (salt)

Acționează asupra mersului secvențial al programului.

Semnalizările de condiție nu sint afectate.



Există două feluri de salturi :

- necondiționate care operează numai asupra numărătorului de program (de adrese a instrucțiunilor) PC
- condiționate, la care se examinează înaintea saltului semnalizările de condiție

CCC	Condiție
000	NZ ( $Z = 0$ )
001	Z ( $Z = 1$ )
010	NC ( $C = 0$ )
011	C ( $C = 1$ )
100	PO ( $P = 0$ )
101	PE ( $P = 1$ )
110	P ( $S = 0$ )
111	M ( $S = 1$ )

JMP adr	3	(PC) $\leftarrow$ (B <sub>2</sub> ) (adr)
J condition adr	3	If (CCC)CCC (PC) $\leftarrow$ (adr)



## Bibliografie

1. Kalman R. E., Falb P. L., Arbib M. A., **Teoria sistemelor dinamice**, Ed. Tehnică 1975, București.
2. Zadeh L. A., Polak E., ș.a., **Teoria sistemelor**, Ed. Tehnică, București, 1973
3. Kaufmann A., Precigout M., **Elemente de teoria mulțimilor și algebră modernă**. Dunod, Paris, 1966
4. Blakeslee T. R., **Digital Design with Standard MSI and LSI**, John Wiley, 1975
5. Barna A., Porat D. I., **Integrated Circuits in Digital Electronics**, John Wiley, 1973
6. Morris R. L. ș.a., **Proiectare cu circuite integrate TTL**, Ed. Tehnică 1974
7. Clare C., **Designing Logic Systems Using State Mashine**, Mc Graw Hill, Inc., 1972
8. A. Osborne. **An Introduction to Microcomputers**, Adam Osborne and Associates Incorporated, 1975
9. **Proceedings of the IEEE**, 1976—1978
10. **IEEE Transaction on Computers** 1971—1978
11. **Hewlett Packard Journal** 1975—1978
12. **Fairchild Semiconductor — The TTL Applications Handbook — 1973**
13. **Fairchild Smiconductor — Advanced Logic Book — 1975**
14. **Computer Design** 1976—1978
15. **Digital Design** 1975—1978
16. **Electronic Design** 1976—1978
17. Sanda Dumitrescu, Gh. Ștefan, **Modelarea structurilor logice digitale ca sisteme cu stări finite**, Buletinul Institutului politehnic tomul XXXIX, Nr. 3, 1977
18. Sanda Dumitrescu, Gh. Ștefan, **Proiectarea și minimizarea automată a sistemelor secvențiale sincrone**, Buletinul Institutului politehnic, 1978
19. Sanda Dumitrescu, **Testarea schemelor numerice**, Referat de doctorat, 1977
20. Sanda Dumitrescu Maican, **Sisteme cu autocorectarea erorilor**, Referat de doctorat, 1978