

Dan NICULA

ELECTRONICĂ DIGITALĂ

Carte de învățatură 2.0



Editura Universității *TRANSILVANIA* din Brașov
ISBN 978-606-19-0563-8

2015

Lecția 16

Sinteza circuitelor logice secvențiale

16.1 Noțiuni teoretice

Metodologia de sinteză a automatelor presupune parcurgerea următoarelor etape:

1. Din descrierea existentă, se deduce funcționalitatea automatului și se descrie în una din formele de reprezentare: organigramă sau graf de tranziții.
2. Dacă este posibil, se reduc stările redundante. Stările redundante sunt stările din care automatul evoluează identic și în care se activează aceleași ieșiri.
3. Se codifică stările prin asocierea acestora cu coduri binare unice, proiectate cu anumite constrângeri (număr minim de biți, cu dependență redusă, cu variație minimă, one-hot).
4. Se alege tipul de bistabile folosite pentru implementare (D, T sau JK).
5. Se completează tabelul de tranziții cu toate stările și toate combinațiile intrărilor.
6. Se completează tabelul de tranziții cu stările următoare și ieșirile. Dacă implementarea se face cu bistabile T sau JK, se completează tabelul de tranziții cu coloanele asociate stimulilor bistabilelor.
7. Se minimizează funcțiile stării următoare (sau ale stimulilor pentru determinarea stărilor următoare) și ale ieșirilor.
8. Se deduc ecuațiile stării următoare (sau ale stimulilor pentru determinarea stărilor următoare) și ale ieșirilor.
9. Se desenează schema de implementare cu bistabilele alese și logica determinată de ecuațiile deduse anterior.

Tabelele următoare prezintă modul în care se determină ecuațiile stimulilor pentru bistabile, pe baza stării curente și a celei viitoare:

D	Q^+	Q	Q^+	$D = Q^+$
0	0	0	0	0
1	1	0	1	1
		1	0	0
		1	1	1

T	Q^+	Q	Q^+	$T = Q \oplus Q^+$
0	Q	0	0	0
1	\overline{Q}	0	1	1
		1	0	1
		1	1	0



J	K	Q^+	Q	Q^+	$J = Q^+$	$K = \overline{Q^+}$
0	0	Q	0	0	0	X
0	1	0	0	1	1	X
1	0	1	1	0	X	1
1	1	\overline{Q}	1	1	X	0

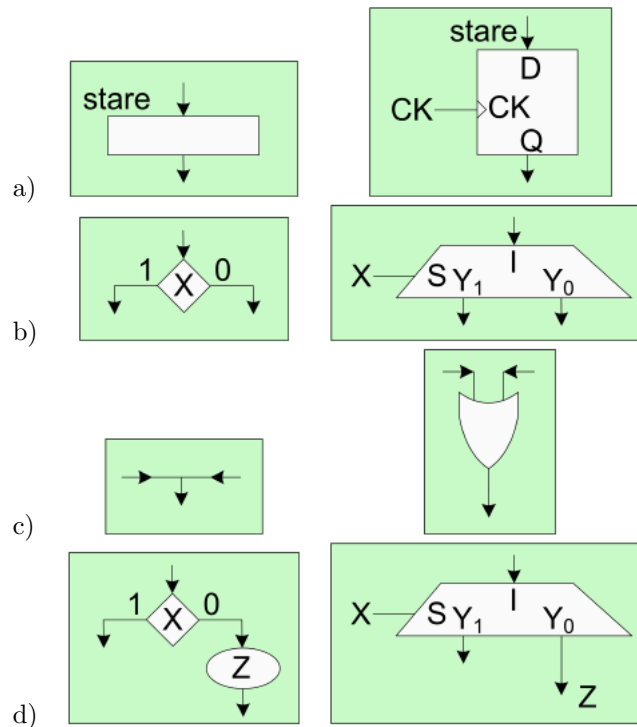


Figura 16.1 Transformarea simbolurilor din organigramă într-o implementare de automat cu stări codificate "one-hot": **a)** simbol de stare, **b)** simbol de decizie, **c)** joncțiune, **d)** simbol de ieșire imediată.

Implementarea unui automat "one-hot" presupune codificarea stărilor cu câte un bit pentru fiecare stare, astfel încât starea curentă să fie determinată de valoarea 1 a bistabilului ce corespunde stării. La un moment dat, un singur bistabil de stare este 1, celelalte bistabile fiind 0.

Pornind de la organigrama automatului, se poate realiza direct o implementare cu codificarea stărilor "one-hot" pe baza următoarelor asocieri:

- simbolurile de stare se înlocuiesc cu bistabile D;
- simbolurile de decizie se înlocuiesc cu un demultiplexor 1:2;
- joncțiunile (mai multe căi de intrare într-o stare) se înlocuiesc cu porți OR;
- ieșirile condiționate (în cazul automatelor Mealy) se obțin de la ieșirile demultiplexorului de decizie.

Fiecare simbol de stare este înlocuit cu un bistabil D, ca în figura 16.1-a astfel:

- punctul de intrare în stare este conectat la intrarea D;
- punctul de ieșire din stare este conectat la ieșirea Q;
- ceasul bistabilului este conectat la portul de ceas al automatului.

Fiecare simbol de decizie este înlocuit cu un demultiplexor 1:2, ca în figura 16.1-b astfel:

- punctul de intrare în simbolul de decizie se conectează pe intrarea de date a demultiplexorului;
- condiția deciziei se conectează pe intrarea de selecție a demultiplexorului;
- punctele de ieșire din simbolul de decizie sunt asociate cu ieșirile demultiplexorului.

Dacă într-o stare se ajunge din două sau mai multe cauze (stări sau condiții diferite) în organigramă apare o joncțiune. Intrările în joncțiune se transformă în intrări într-o poartă OR, așa ca în figura 16.1-c. Ieșirea porții OR reprezintă ieșirea din joncțiune care se conectează pe intrarea simbolului ce îi succede.

Intrarea în simbolurile de ieșire condiționată provine întotdeauna de la ieșirea unui simbol de decizie. Ieșirea condiționată se obține de la ieșirea corespunzătoare a demultiplexorului ce corespunde simbolului de decizie precedent, așa ca în figura 16.1-d. Dacă în organigramă apar mai multe simboluri de ieșiri imediate asociate aceleași ieșiri, se



adaugă o poartă OR în care intră semnalele provenite de la ieșirile demultiplexoarelor. Ieșirea porții OR va fi semnalul de ieșire imediată.

Ieșirile de stare (ale automatelor Moore) se obțin printr-o poartă OR în care intră ieșirile Q asociate bistabilelor stărilor în care ieșirea este activă.

16.2 Pentru cei ce vor doar să promoveze examenul

1. Implementați cu bistabile D și un număr minim de porți NAND automatele secvențiale sincrone cu funcționarea descrisă prin organigramele prezentate în figura 16.2.

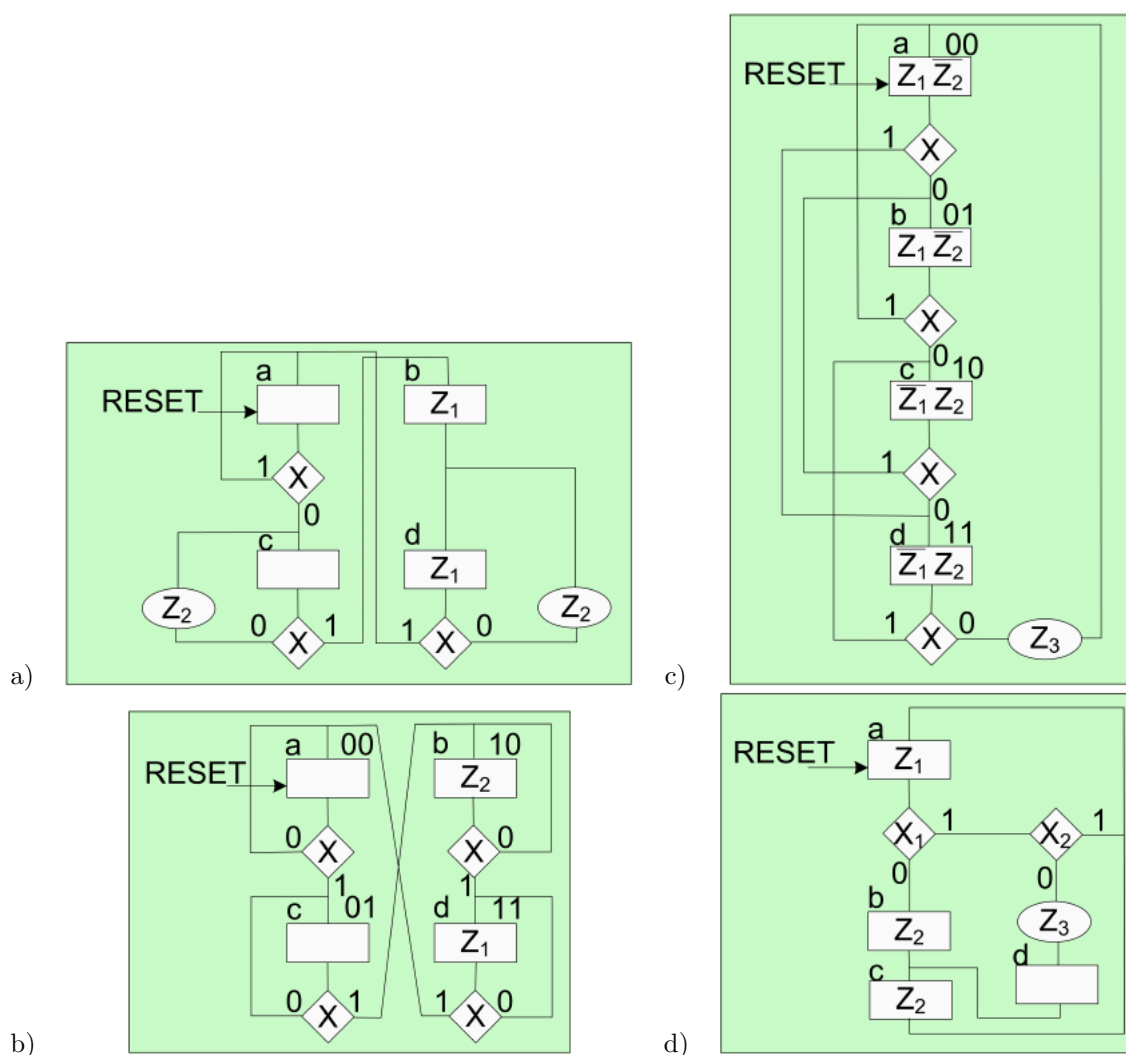


Figura 16.2 Organigrame pentru problema 1.

2. Proiectați un automat cu bistabile D care are o intrare de validare EN .
 Dacă $EN = 0$ automatul își păstrează starea.
 Dacă $EN = 1$, automatul ciclează între stările 00, 10, 01, 11.

16.3 Pentru cei ce vor să învețe

1. Implementați cu bistabile T și un număr minim de porți NAND automatele secvențiale sincrone având funcționarea descrisă prin organigramele prezentate în figura 16.3.

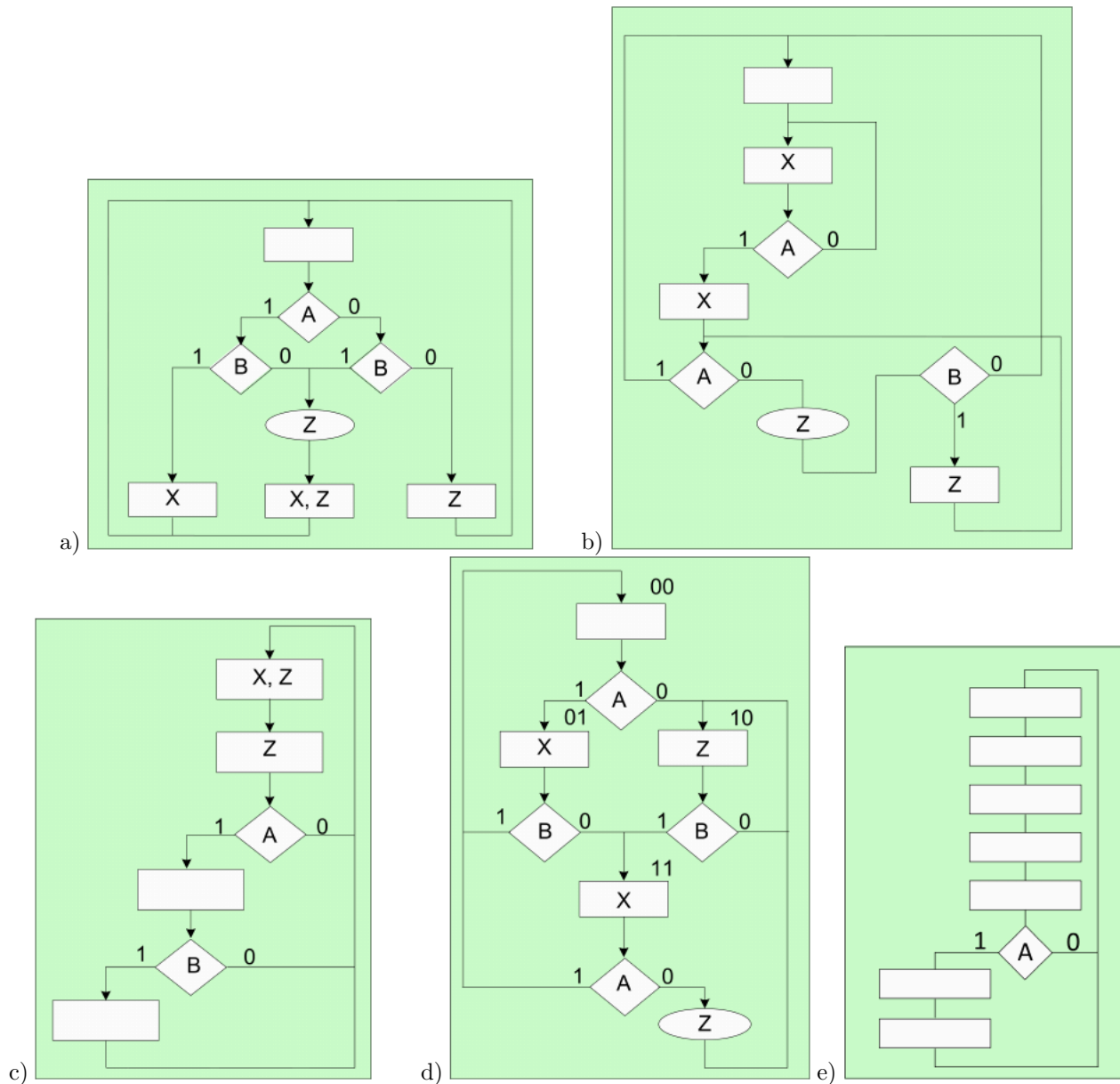


Figura 16.3 Organigrame pentru problema 1.

Soluție

d) Circuitul are două intrări (A și B) și două ieșiri (X și Z). Este un automat Mealy (deoarece conține un simbol de ieșire imediată Z). Cele 4 stări pot fi codificate cu un număr minim de 2 biți. Tabelul de tranziții conține coloane pentru două intrări, starea curentă (2 biți), starea viitoare (2 biți) și două ieșiri. În plus, se atașează două coloane pentru stimulii bistabilelor T.

Pe baza organigramei, în prima etapă se completează coloana stării viitoare. Ulterior, se deduc coloanele aferente intrărilor bistabilelor, prin referire la comportamentul bistabilelor T descris de tabelul:



Q	Q^+	T
0	0	0
0	1	1
1	0	1
1	1	0

Tabelul de tranziții este:

Linia	Intrări		Stare prezentă		Stare viitoare		Ieșiri		Stimuli T	
	A	B	Q_1	Q_0	Q_1^+	Q_0^+	X	Z	T_1	T_0
1	0	-	0	0	1	0	0	0	1	0
2	1	-	0	0	0	1	0	0	0	1
3	-	0	0	1	1	1	1	0	1	0
4	-	1	0	1	0	0	1	0	0	1
5	-	0	1	0	1	0	0	1	0	0
6	-	1	1	0	1	1	0	1	0	1
7	0	-	1	1	1	0	1	1	0	1
8	1	-	1	1	0	0	1	0	1	1

Analogia dintre liniile din tabelul de tranziții și căsuțele din diagrama V-K este prezentată în figura 16.4, alături de diagrame V-K asociate funcțiilor T_1, T_0, X, Z dependente de (A, B, Q_1, Q_0) .

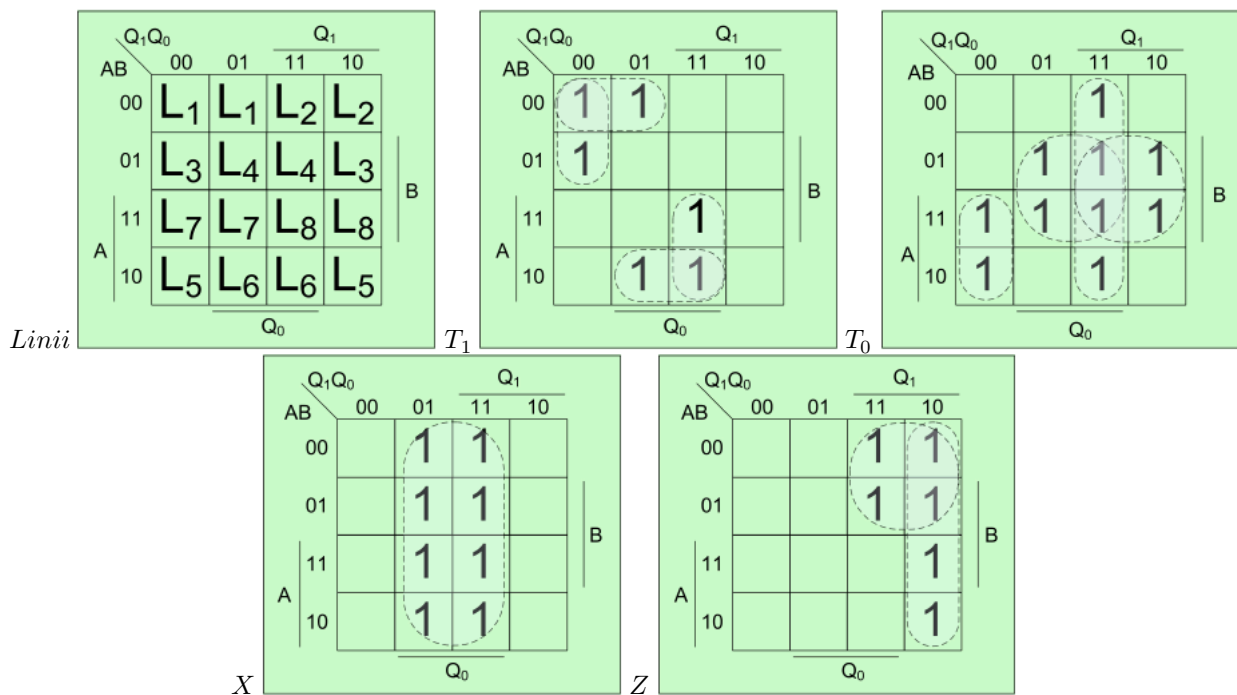


Figura 16.4 Diagrame V-K, pentru problema 1-d.

După minimizare, rezultă ecuațiile:

$$T_1 = Q_1 \cdot Q_0 \cdot A + Q_0 \cdot A \cdot \bar{B} + \bar{Q}_1 \cdot \bar{A} \cdot \bar{B} + \bar{Q}_1 \cdot \bar{Q}_0 \cdot \bar{A}$$

$$T_0 = Q_1 \cdot B + Q_1 \cdot Q_0 + Q_0 \cdot B + \bar{Q}_1 \cdot \bar{Q}_0 \cdot A$$

$$X = Q_0$$

$$Z = Q_1 \cdot \bar{A} + Q_1 \cdot \bar{Q}_0$$

Implementarea automatului este prezentată în figura 16.5.

- Un automat cu 3 stări și două intrări funcționează conform grafului de tranziții prezentat în figura 16.6, ("-" desemnează o intrare indiferentă).



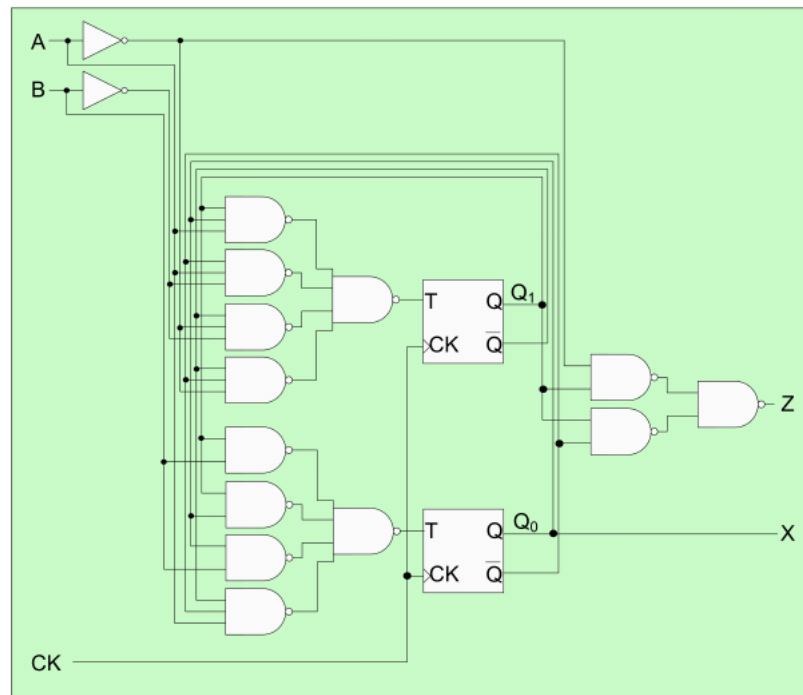


Figura 16.5 Implementarea automatului cu bistabile T, problema 1-d.

- a) Implementați automatul cu porți logice NAND și bistabile D, cu codificare cu număr minim de biți.
 b) Implementați automatul cu codificarea stărilor "one-hot".

Soluție

- a) Pentru codificarea cu număr minim de biți, pentru cele 3 stări se folosesc 2 biți (Q_1, Q_0). Se propune codificarea: $A = 00, B = 01, C = 10$.

Tabelul de tranziții este:

Intrări		Stare prezentă			Stare viitoare		
X	Y	Nume	Q_1	Q_0	Nume	D_1	D_0
0	-	A	0	0	A	0	0
1	-	A	0	0	B	0	1
-	1	B	0	1	C	1	0
0	0	B	0	1	A	0	0
1	0	B	0	1	B	0	1
-	0	C	1	0	A	0	0
-	1	C	1	0	C	1	0

Pe baza acestui tabel de tranziții se construiesc diagramele V-K prezentate în figura 16.6. După minimizare, se obțin funcțiile:

$$D_1 = Y \cdot \overline{Q_1} \cdot Q_0 + Y \cdot Q_1 \cdot \overline{Q_0}$$

$$D_0 = X \cdot \overline{Y} \cdot \overline{Q_1} + X \cdot \overline{Q_1} \cdot Q_0$$

- b) Codificarea "one-hot" constă în asocierea fiecărei stări cu un bistabil. La un moment dat, un singur bistabil are starea egală cu 1, semnificând faptul că automatul se află în acea stare. Se notează ieșirile bistabilelor asociate celor 3 stări cu Q_A, Q_B și Q_C .

Ecuatiile intrărilor bistabilelor se pot deduce din graful de tranziții. Condițiile de activare a unei stări sunt determinate de arcele care se termină pe nodul asociat acelei stări.

$$D_A = Q_A \cdot \overline{X} + Q_B \cdot \overline{X} \cdot \overline{Y} + Q_C \cdot \overline{Y}$$

$$D_B = Q_A \cdot X + Q_B \cdot X \cdot \overline{Y}$$

$$D_C = Q_B \cdot Y + Q_C \cdot Y$$



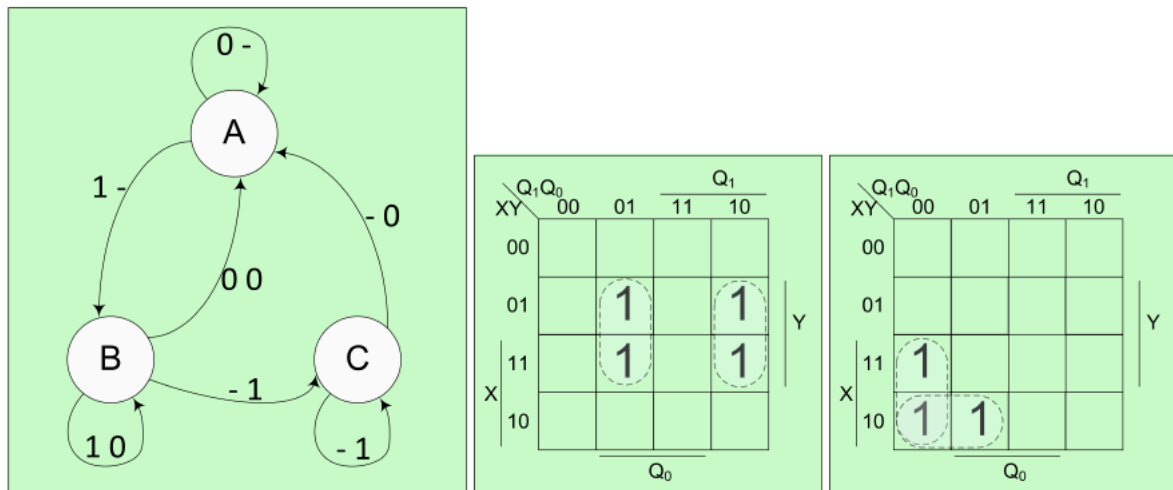


Figura 16.6 Graf de tranziție și diagrame V-K pentru problema 2.

3. Realizați tabelul de tranziții al automatului caracterizat de grafurile de tranziții prezentat în figura 16.7. Minimizați funcțiile și implementați automatul cu porți NAND și bistabile D. Notați biții stării curente Q_1 și Q_0 .
O alternativă de codare a stărilor "one-hot" este: $S_A = 0001$, $S_B = 1000$, $S_C = 0010$, $S_D = 0100$. Notați biții de stare curentă (Q_D , Q_C , Q_B , Q_A). Implementați automatul cu codificarea propusă.

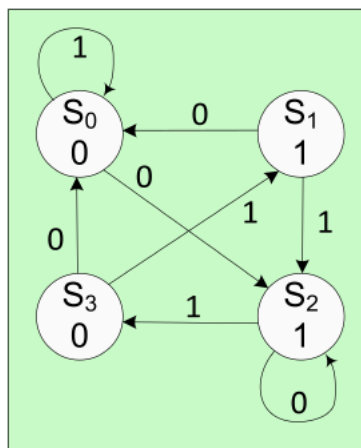


Figura 16.7 Graf de tranziții pentru problema 3.

4. Proiectați automatele descrise de grafurile de tranziții din figura 16.8 cu bistabile D și un număr minim de porți logice. Cu "X" a fost marcată o valoare indiferentă a intrării.

Proiectați automatele descrise de aceleași grafuri de tranziții cu bistabile JK sau bistabile T.

Pentru grafurile de tranziții a), determinați secvența stărilor și valorile ieșirilor în cazul aplicării pe intrare a secvenței 1100110011010101011110000111110000 (presupuneți starea inițială 00).

5. Proiectați cu bistabile T un automat care numără în secvența binară 0, 2, 3, 6, 7.

Soluție

Automatul are 5 stări. Sunt necesari 3 biți pentru a codifica stările. Automatul nu are intrări. Se codifică stările astfel încât ieșirile să provină de la biții de stare, fără logică adițională. Fiecărui bit de stare i se va asocia un bistabil T. Tabelul de tranziții al automatului este:



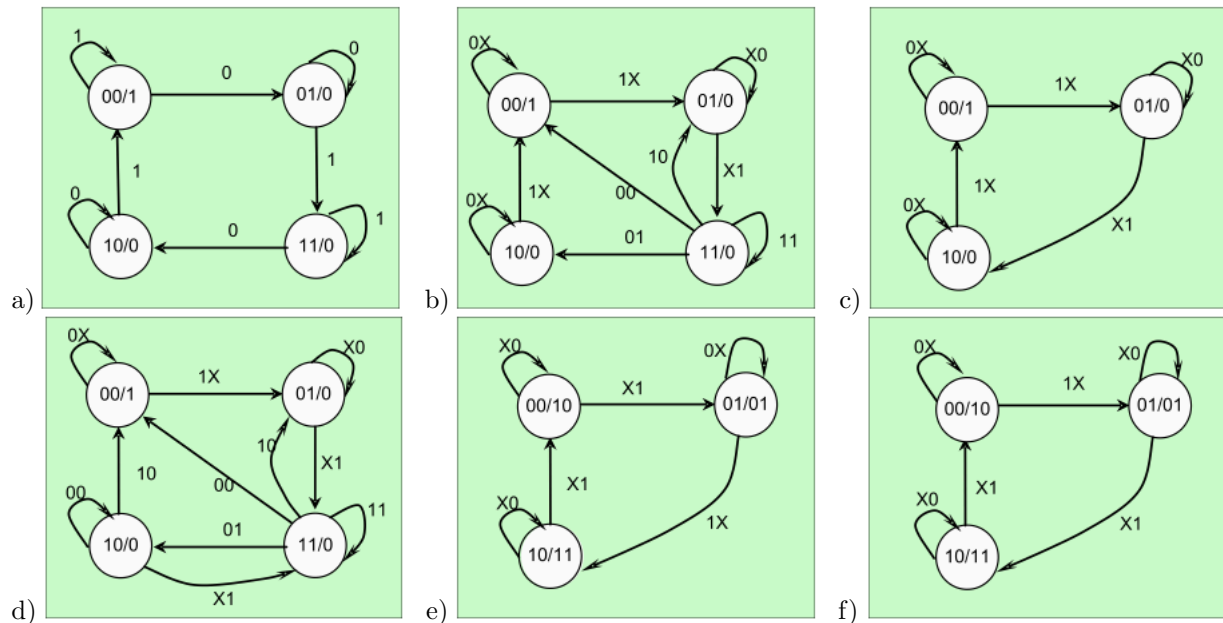


Figura 16.8 Grafuri de tranziții pentru problema 4.

Stare prezentă				Stare viitoare				Stimuli T		
Nume	Q_2	Q_1	Q_0	Nume	Q_2^+	Q_1^+	Q_0^+	T_2	T_1	T_0
0	0	0	0	2	0	1	0	0	1	0
2	0	1	0	3	0	1	1	0	0	1
3	0	1	1	6	1	1	0	1	0	1
6	1	1	0	7	1	1	1	0	0	1
7	1	1	1	0	0	0	0	1	1	1
1	0	0	1	0	0	0	0	0	0	1
4	1	0	0	0	0	0	0	1	0	0
5	1	0	1	0	0	0	0	1	0	1

Stimuli T_i se determină pe bază stării prezente Q_i și a stării viitoare Q_i^+ după algoritmul:

- dacă starea viitoare este identică cu starea prezentă ($Q_i = Q_i^+$) atunci $T_i = 0$;
- dacă starea viitoare este diferită de starea prezentă ($Q_i \neq Q_i^+$) atunci $T_i = 1$.

Stările care nu apar în secvență trebuie considerate ca indiferente sau, pentru siguranță, se poate proiecta automatul să revină în secvența de numărare.

Diagramele V-K asociate funcțiilor $T_i, i \in \{2, 1, 0\}$ și implementarea automatului cu bistabile T sunt prezentate în figura 16.9.

$$T_2 = Q_2 \cdot \overline{Q_1} + Q_1 \cdot Q_0$$

$$T_1 = \overline{Q_2} \cdot \overline{Q_1} \cdot \overline{Q_0} + Q_2 \cdot Q_1 \cdot Q_0$$

$$T_0 = Q_1 + Q_0$$

6. Pentru automatele descrise la problema 5 adăugați o intrare de validare activă în 1:

- dacă intrarea este egală cu 1 automatul funcționează ca și cel inițial,
- dacă intrarea este egală cu 0, automatul își păstrează starea.

Prezentați grafurile de tranziție ale automatelor cu și fără intrare de validare. Implementați modificarea în două variante:

- prin metodologia clasică de implementare a unui automat;
- adăugând fiecărui bistabil un multiplexor 2:1 care să implementeze facilitatea de păstrare a stării curente.

7. Proiectați cu bistabile D/T/JK automatul care numără în secvențele binare:

- 0, 1, 3, 4, 5, 6;
- 0, 6, 4, 3, 2;
- 0, 7, 6, 1, 3, 4;



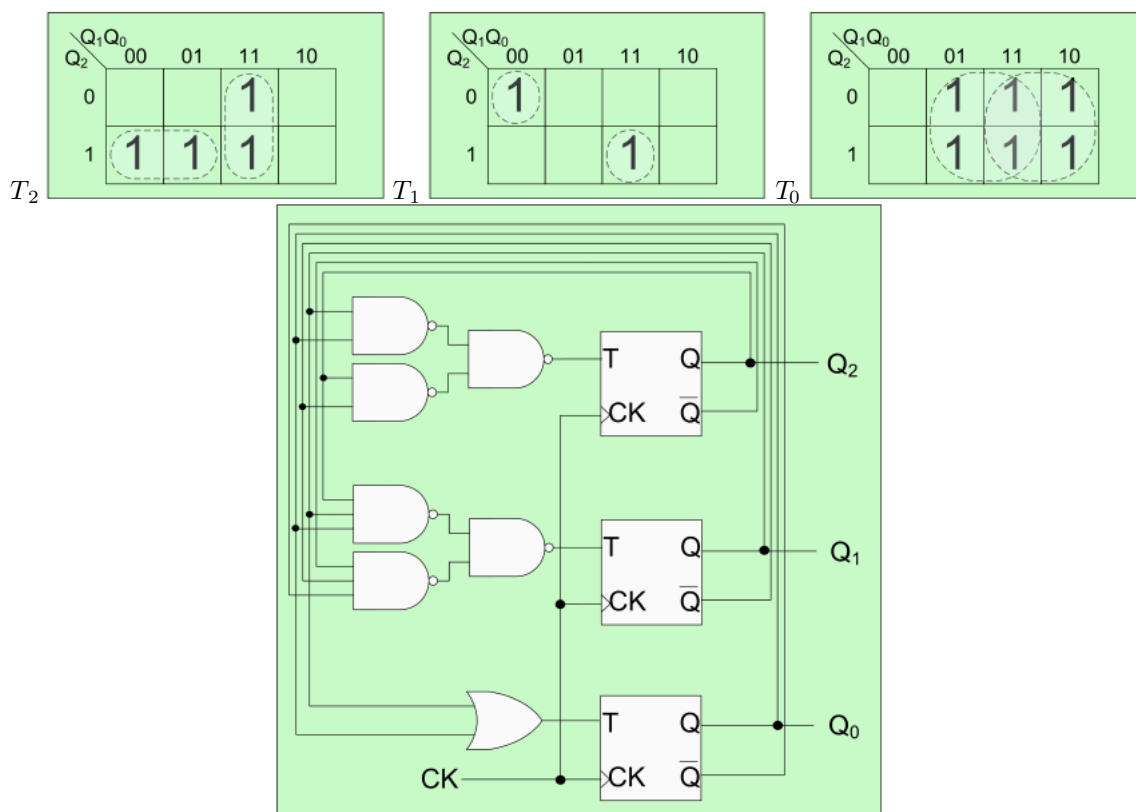


Figura 16.9 Diagrame V-K și implementare cu bistabile T a automatului descris la problema 5.

d) 0, 4, 5, 6, 1, 2, 3, 7;

e) 0, 1, 2, 3, 5, 6.

Propuneți circuite pentru inițializare în starea 0 și revenire în această stare din stările ce nu aparțin ciclurilor.

8. Proiectați cu bistabile D/JK/T un numărător în cod BCD.
9. Proiectați cu bistabile D/JK/T un numărător în cod Gray pe 2, 3 și 4 biți.
10. Proiectați un numărător de 4 biți în secvența 0, 1, 2, 3, 2, 4, 5, 6, 7. Se observă că starea 2 are două apariții în cadrul acestui ciclu.

Soluție

Deși automatul are 8 seturi distincte de ieșiri (posibil de codat cu 3 biți) automatul are 9 stări diferite. Stările în care ieșirea este egală cu 2 sunt diferite: în prima se ajunge în succesiunea 1, 2, 3, în cealaltă se ajunge în succesiunea 3, 2, 4. Se notează cele 3 ieșiri O_2, O_1, O_0 . Automatul are 9 stări codificate cu un număr minim de 4 biți. Biții stării curente se denumesc Q_3, Q_2, Q_1, Q_0 . Automatul nu are intrări, graful de tranziții al automatului fiind sub forma unui inel de 9 stări cu tranziții necondiționate dintr-o stare în alta. Pentru a minimiza circuitul combinațional al ieșirilor se pot codifica stările astfel încât 3 biți de stare să coincidă cu biții de ieșire. Ideea este ca stările în care ieșirea este 2 să difere printr-un singur bit, iar cei mai puțin semnificativi biți să fie 010. Deci, se for codifica stările cu ieșirea egală cu 2 astfel: $S_2 = 0010$ și $S_{10} = 1010$. Codificarea este prezentată în tabelul de tranziții:



Stare prezentă					Stare viitoare					Ieșiri			
Nume	Q_3	Q_2	Q_1	Q_0	Nume	Q_3^+	Q_2^+	Q_1^+	Q_0^+	Număr	O_2	O_1	O_0
S_0	0	0	0	0	S_1	0	0	0	1	0	0	0	0
S_1	0	0	0	1	S_2	0	0	1	0	1	0	0	1
S_2	0	0	1	0	S_3	0	0	1	1	2	0	1	0
S_3	0	0	1	1	S_{10}	1	0	1	0	3	0	1	1
S_{10}	1	0	1	0	S_4	0	1	0	0	2	0	1	0
S_4	0	1	0	0	S_5	0	1	0	1	4	1	0	0
S_5	0	1	0	1	S_6	0	1	1	0	5	1	0	1
S_6	0	1	1	0	S_7	0	1	1	1	6	1	1	0
S_7	0	1	1	1	S_0	0	0	0	1	7	1	1	1

11. Se consideră organigrama din figura 16.10.

a) Să se implementeze automatul descris de organigramă cu bistabile D și codificarea stărilor:

$S_1 = 00$, $S_2 = 01$, $S_3 = 10$, $S_4 = 11$.

b) Să se implementeze automatul descris de organigramă cu bistabile D și codificarea stărilor:

$S_1 = 00$, $S_2 = 11$, $S_3 = 10$, $S_4 = 01$. Să se compare implementarea cu cea realizată la punctul a).

c) Să se implementeze automatul descris de organigramă cu bistabile T și codificarea stărilor:

$S_1 = 00$, $S_2 = 11$, $S_3 = 10$, $S_4 = 01$.

d) Să se implementeze automatul descris de organigramă cu bistabile D și codificarea stărilor "one=hot":

$S_1 = 0001$, $S_2 = 0010$, $S_3 = 0100$, $S_4 = 1000$.

Studiați comportamentul circuitului în cazul apariției unor stări interzise 0011, 0000, 0111. Propuneți un circuit combinațional pentru semnalarea posibilelor erori de stare.

e) Să se implementeze automatul convertind direct organigrama în circuit, prin asocierea fiecărei stări cu un bistabil D.

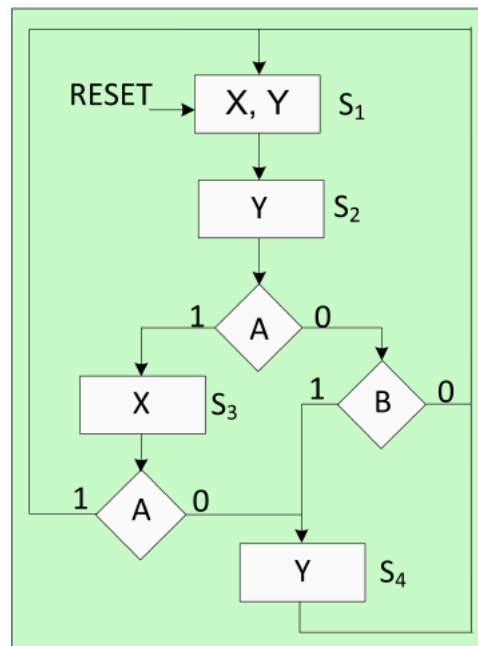


Figura 16.10 Organigramă pentru problema 11.

12. Să se implementeze un automat a cărui funcționare este descrisă de formulele de undă ale ieșirilor, prezentate în figura 16.11-a.

Soluție

Se observă că automatul prezintă 3 stări (a, b, c), nu are intrări și se resetează în starea a. În cazul codificării stărilor cu 2 biți (Q_1, Q_0): $a = 00$, $b = 01$, $c = 10$ rezultă tabelul de tranziții:



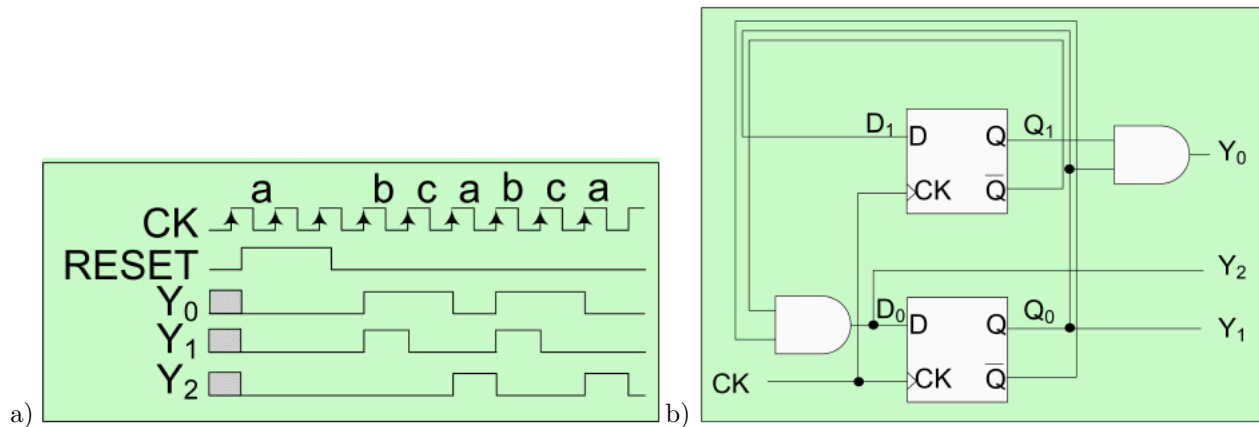


Figura 16.11 a) Forme de undă, b) Automatul pentru problema 12.

Stare prezentă			Stare viitoare			Ieșiri		
Nume	Q_1	Q_0	Nume	Q_1^+	Q_0^+	Y_2	Y_1	Y_0
a	0	0	b	0	1	1	0	0
b	0	1	c	1	0	0	1	1
c	1	0	a	0	0	0	0	1
-	1	1	-	-	-	-	-	-

În tabel s-a inclus și starea neprezentată în formele de undă (11) din care s-a proiectat o tranziție într-o stare indiferentă și ieșiri indiferente.

Rezultă ecuațiile stării viitoare și ale ieșirilor:

$$\begin{aligned}
 D_1 &= Q_0 \\
 D_0 &= \overline{Q_1} \cdot \overline{Q_0} \\
 Y_2 &= \overline{Q_1} \cdot \overline{Q_0} = D_0 \\
 Y_1 &= Q_0 = D_1 \\
 Y_0 &= Q_1 + Q_0
 \end{aligned}$$

Starea care urmează stării prezente 11 este $c = 10$, iar ieșirile în această stare sunt $Y_{2,1,0} = 011$.

Structura automatului implementat cu bistabile D este prezentată în figura 16.11-b.

13. Să se implementeze un automat cu o intrare X a cărei funcționare este descrisă de formele de undă ale ieșirilor, prezentate în figura 16.12-a.

Soluție

Pe baza formelor de undă se deduce organigrama prezentată în figura 16.12-b.

14. Proiectați un automat care prezintă la ieșire un puls de lățime 8 perioade de ceas la fiecare puls de o perioadă de ceas primit pe intrare. Prezentați condițiile de funcționare ale circuitului.

Soluție

Automatul are o intrare, denumită X și o ieșire denumită Y .

În cazul unei implementări ca automat Moore (ieșirea depinde doar de starea curentă), este nevoie de 9 stări: o stare de repaus, unde se așteaptă apariția pulsului de declanșare și 8 stări pe durata cărora ieșirea este activată.

În cazul implementării ca automat Mealy, există o stare de repaus în care se așteaptă pulsul de declanșare, dar ieșirea va fi activată 1 exact în momentul apariției pulsului la intrare (combinational), urmând ca automatul să parcurgă alte 7 stări pe durata cărora ieșirea să fie 1. Această variantă are doar 8 stări și poate fi implementată cu codificarea stărilor cu un număr minim de 3 biți. Schema bloc a automatului este prezentată în figura 16.13-a. Graful de tranziții al automatului Mealy este prezentat în figura 16.13-b.

Pe baza grafului de tranziții, se determină tabelul de tranziții (s-au notat cu Q_A, Q_B, Q_C biții stării curente):



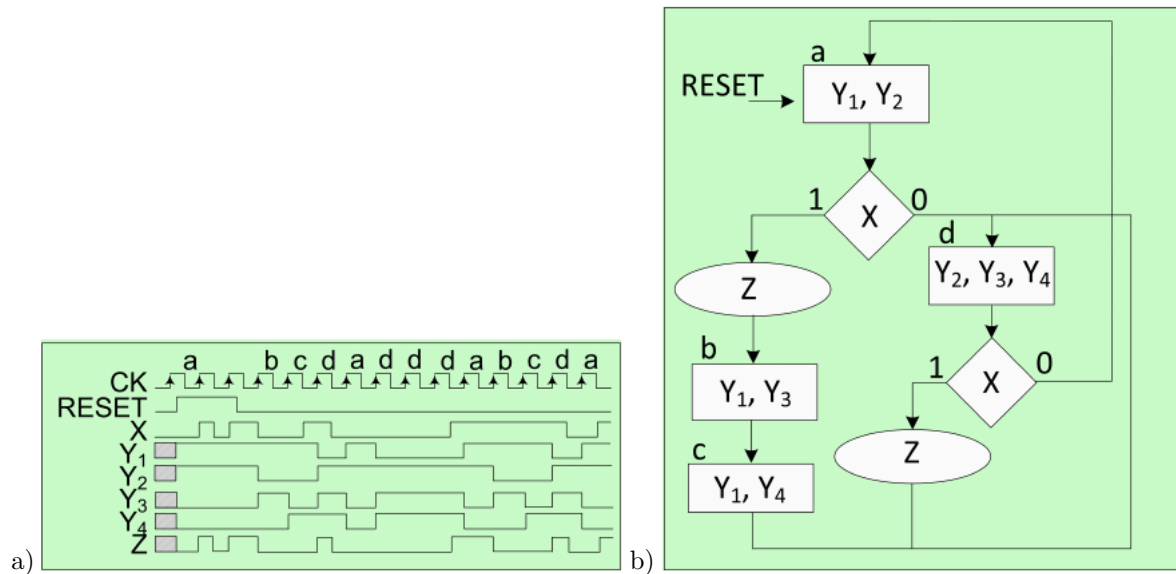


Figura 16.12 a) Forme de undă, b) Organigramă dedusă din formele de undă, problema 13.

Intrare X	Stare curentă			Stare viitoare			Ieșire Y
	Q_C	Q_B	Q_A	Q_C^+	Q_B^+	Q_A^+	
0	0	0	0	0	0	0	0
1	0	0	0	0	0	1	1
-	0	0	1	0	1	0	1
-	0	1	0	0	1	1	1
-	0	1	1	1	0	0	1
-	1	0	0	1	0	1	1
-	1	0	1	1	1	0	1
-	1	1	0	1	1	1	1
-	1	1	1	0	0	0	1

Cu ajutorul diagramelor Veitch-Karnaugh se determină expresiile minime ale intrărilor bistabilelor de stare și a ieșirii.

$$Q_A^+ = \overline{Q_A} \cdot X + \overline{Q_A} \cdot Q_B + \overline{Q_A} \cdot Q_C$$

$$Q_B^+ = \overline{Q_A} \cdot Q_B + Q_A \cdot \overline{Q_B}$$

$$Q_C^+ = \overline{Q_A} \cdot Q_C + \overline{Q_B} \cdot Q_C \cdot \overline{Q_B} \cdot \overline{Q_C}$$

$$Y = Q_A + X + Q_B + Q_C$$

Pe baza acestor funcții se obține implementarea cu porți și bistabile D prezentată în figura 16.14.

Circuitul funcționează corect dacă între două pulsuri consecutive există o pauză de minimum 8 perioade de tact ($X = 0$).

Se remarcă faptul că, în cazul acestei probleme, o implementare cu detector de front pozitiv (implementat cu un bistabil și o poartă cu două intrări) urmat de un registru de deplasare de 7 biți (care să întârzie pulsul) și apoi o poartă OR cu 8 intrări, ar fi fost o soluție mai eficientă și ca timp de proiectare și ca frecvență maximă de operare (figura 16.15).

15. Proiectați automatele care funcționează pe baza organigramelor prezentate în figurile 16.16-a,b. Implementați automatele cu codificarea stărilor "one-hot".

Soluție

a) Pe baza algoritmului prezentat în partea teoretică a lecției, automatul se implementează cu circuitul din figura 16.17, determinat prin transformarea directă a organigramei. De remarcat faptul că bistabilele trebuie inițializate în starea $a = 1, b = c = d = 0$.

16. Să se implementeze automatele care funcționează conform grafurilor de tranziție prezentate în figura 16.18-a,b,c,d cu bistabile D / bistabile T / bistabile JK.



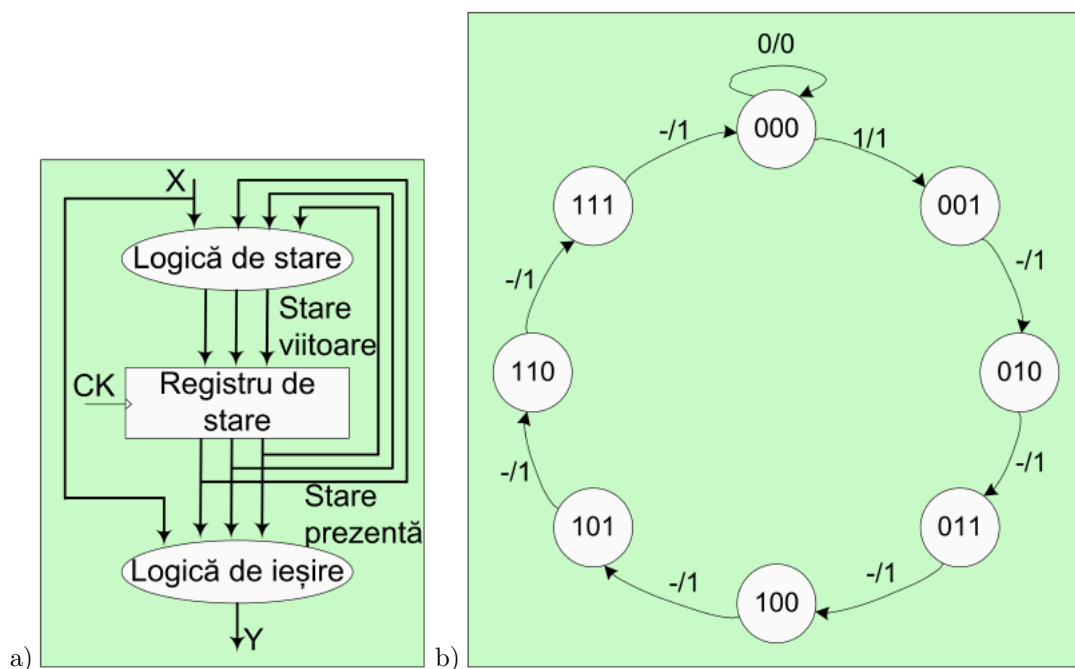


Figura 16.13 a) Schema bloc și b) graful de tranziții pentru problema 14.

Soluție

a) Automatul este de tip Mealy, are o intrare, o ieșire și 4 stări codificate cu 2 biți. Dacă intrarea este egală cu 1, automatul se comportă ca un numărător modulo 4 în sens crescător. Dacă intrarea este egală cu 0, automatul se comportă ca un numărător modulo 4 în sens descrescător. Ieșirea semnalizează depășirea la numărare (tranziție din 11 în 00 în sens crescător, sau din 00 în 11 în sens descrescător). Tabelul de tranziții pentru implementarea cu bistabile D este:

Intrare U_p	Stare curentă $Q_1 \quad Q_0$		Stare viitoare $D_1 \quad D_0$		Ieșire Y
0	0	0	1	1	1
1	0	0	0	1	0
0	0	1	0	0	0
1	0	1	1	0	0
0	1	0	0	1	0
1	1	0	1	1	0
0	1	1	1	0	0
1	1	1	0	0	1

Rezultă ecuațiile:

$$D_1 = \overline{U_p} \oplus Q_1 \oplus Q_0$$

$$D_0 = \overline{Q_0}$$

$$Y = U_p \cdot Q_1 \cdot Q_0 + \overline{U_p} + Q_1 + Q_0$$

b) Automatul este de tip Mealy, are două intrări, o ieșire și 4 stări codificate cu 2 biți. Prima intrare este prioritară și are o funcție de validare a comutării (dacă această intrare este egală cu 0, automatul își păstrează starea). A doua intrare, este relevantă doar când intrarea de validare este egală cu 1 și determină sensul de numărare pentru numărătorul modulo 4 (în sens crescător sau în sens descrescător). Ieșirea semnalizează depășirea la numărare (tranziție din 11 în 00 în sens crescător, sau din 00 în 11 în sens descrescător). Tabelul de tranziții pentru implementarea cu bistabile T este:



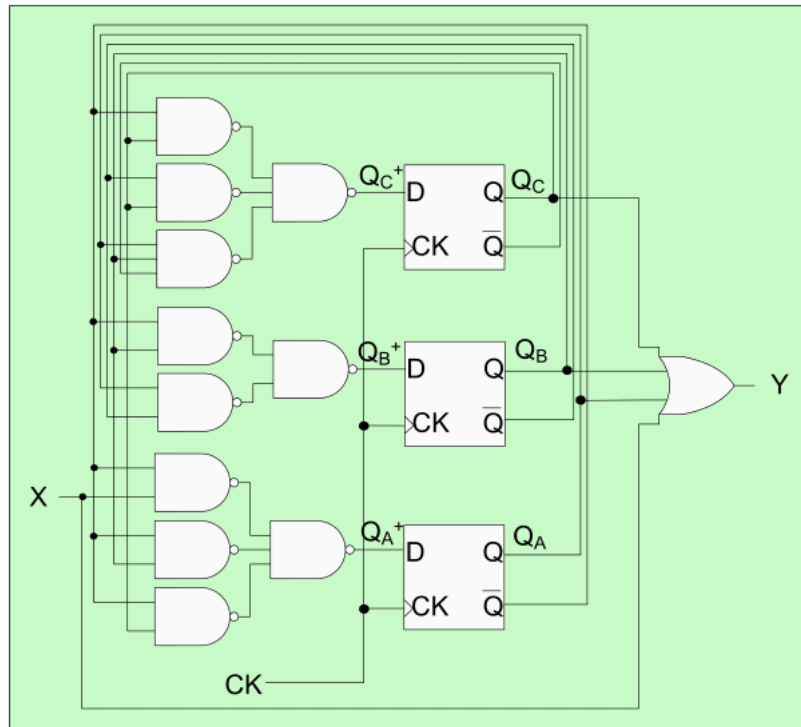


Figura 16.14 Implementarea automatului Mealy pentru problema 14.

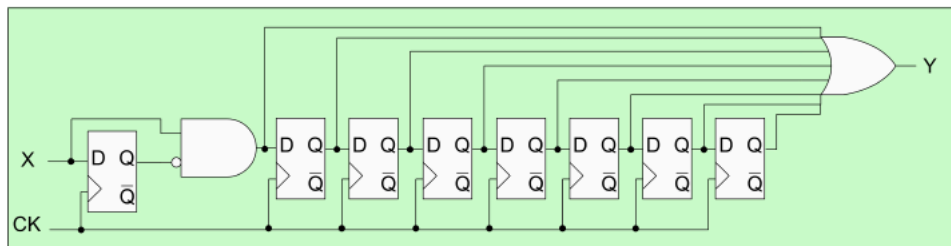


Figura 16.15 Implementarea alternativă a automatului de la problema 14.

Intrări		Stare curentă		Stare viitoare		Stimuli bistabile		Ieșire
En	Up	Q_1	Q_0	Q_1^+	Q_0^+	T_1	T_0	Y
0	-	0	0	0	0	0	0	0
1	0	0	0	1	1	1	1	1
1	1	0	0	0	1	0	1	0
0	-	0	1	0	1	0	0	0
1	0	0	1	0	0	0	1	0
1	1	0	1	1	0	1	1	0
0	-	1	0	1	0	0	0	0
1	0	1	0	0	1	1	1	0
1	1	1	0	1	1	0	1	0
0	-	1	1	1	1	0	0	0
1	0	1	1	1	0	0	1	0
1	1	1	1	0	0	1	1	1

Rezultă ecuațiile:

$$T_1 = En \cdot \overline{Up} \oplus Q_0$$

$$T_0 = En$$

$$Y = En \cdot (Up \cdot Q_1 \cdot Q_0 + \overline{Up} \cdot \overline{Q_1} \cdot \overline{Q_0})$$

c) Automatul este de tip Mealy, are două intrări, o ieșire și 4 stări codificate cu 2 biți. Prima intrare este



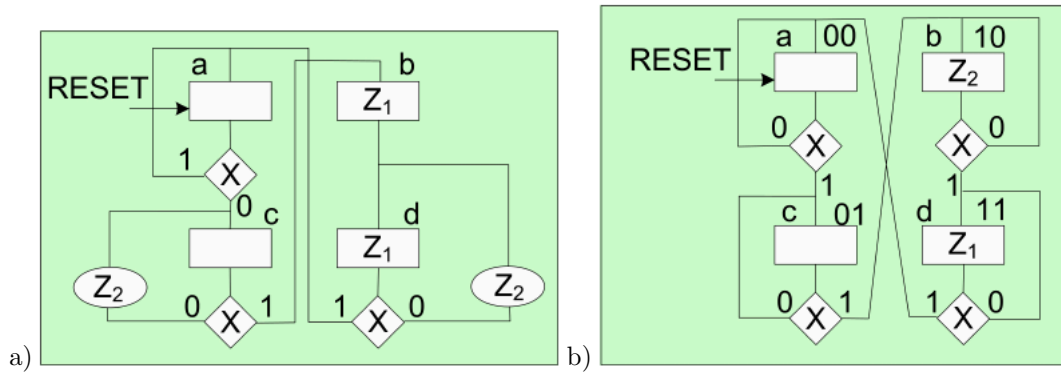


Figura 16.16 Organigrame pentru problema 15.

prioritară și are o funcție de validare a comutării (dacă această intrare este egală cu 0, automatul își păstrează starea). A doua intrare, este relevantă doar când intrarea de validare este egală cu 1 și determină sensul de numărare pentru numărătorul modulo 3 (în sens crescător sau în sens descrescător) în succesiunea 00, 01, 11. Ieșirea semnalizează depășirea la numărare (tranziție din 11 în 00 în sens crescător, sau din 00 în 11 în sens descrescător). Din cea de-a patra stare, codificată 10, se tranzitează în starea 11, indiferent de starea intrărilor. Tabelul de tranziții pentru implementarea cu bistabile JK este:

Intrări		Stare curentă		Stare viitoare		Stimuli bistabile				Ieșire
En	Up	Q_1	Q_0	Q_1^+	Q_0^+	J_1	K_1	J_0	K_0	Y
0	-	0	0	0	0	0	-	0	-	0
1	0	0	0	1	1	1	-	1	-	1
1	1	0	0	0	1	0	-	1	-	0
0	-	0	1	0	1	0	-	-	0	0
1	0	0	1	0	0	0	-	-	1	0
1	1	0	1	1	1	1	-	-	0	0
-	-	1	0	1	1	-	0	1	-	0
0	-	1	1	1	1	-	0	-	0	0
1	0	1	1	0	1	1	-	-	0	0
1	1	1	1	0	0	-	1	-	1	1

Rezultă ecuațiile:

$$J_1 = En \cdot Up \cdot Q_0 + En \cdot Q_1 + En \cdot \overline{Up} \cdot \overline{Q_0}$$

$$K_1 = En \cdot Q_0$$

$$J_0 = En + Q_1$$

$$K_0 = En \cdot Up \cdot Q_1 + En \cdot \overline{Up} \cdot \overline{Q_1}$$

$$Y = En \cdot Up \cdot Q_1 \cdot Q_0 + En \cdot \overline{Up} \cdot \overline{Q_1} \cdot \overline{Q_0}$$

d) Graful de tranziții prezintă un semi-automat (fără ieșiri). Se poate considera că biții de stare sunt și ieșiri. În acest caz se poate considera că este un automat de tip Moore cu două intrări și 4 stări codificate cu 2 biți. Cele două intrări determină funcționalitatea automatului conform tabelului:

I_1	I_0	Acțiune
0	0	Resetează automatul în starea 00
0	1	Păstrează starea, nu comută
1	0	Numără modulo 3 în secvența 00, 01, 11
1	1	Numără modulo 4 în secvența 00, 01, 10, 11

Tabelul de tranziții pentru implementarea cu bistabile JK este:



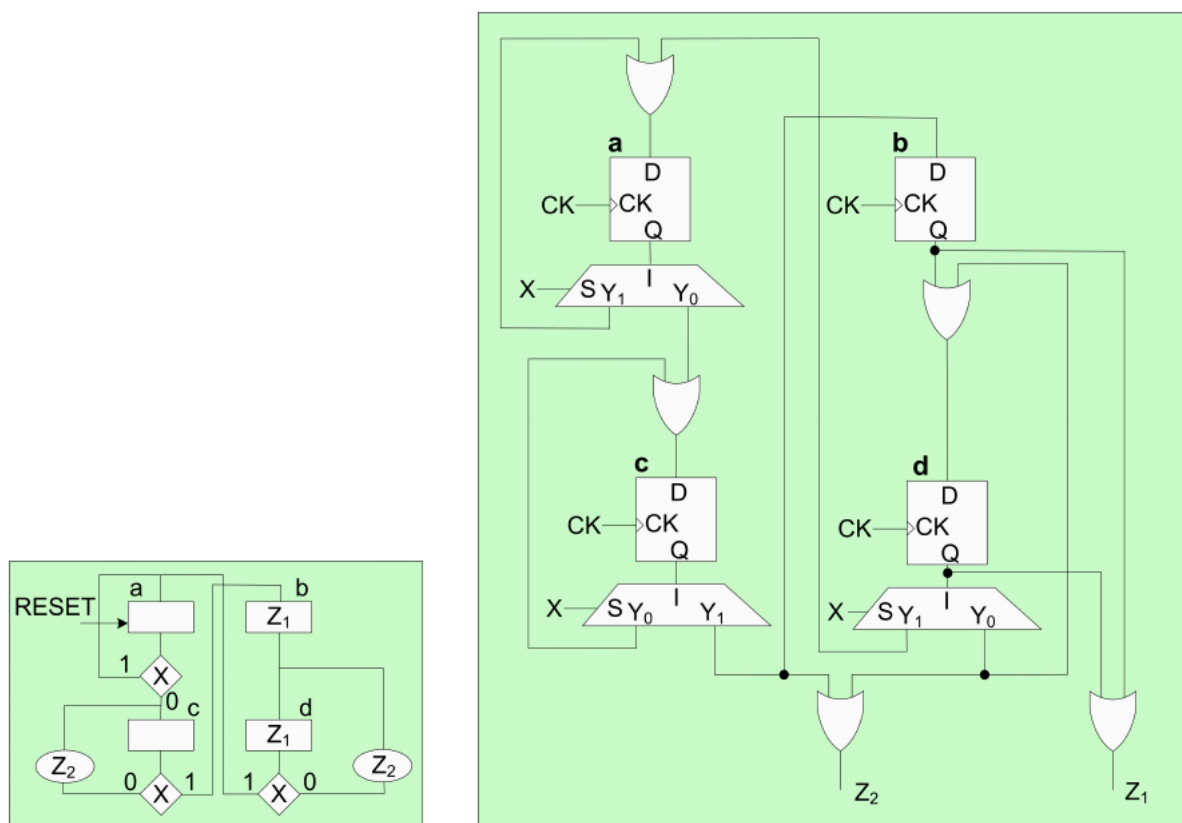


Figura 16.17 Implementarea automatului descris prin organigrama de la problema 15-a, prin transformarea directă a organigramei într-o structură, cu codificarea stărilor "one-hot".

Intrări		Stare curentă		Stare viitoare		Stimuli bistabile			
I_1	I_0	Q_1	Q_0	Q_1^+	Q_0^+	J_1	K_1	J_0	K_0
0	-	0	0	0	0	0	-	0	-
1	-	0	0	0	1	0	-	1	-
0	0	0	1	0	0	0	-	-	1
0	1	0	1	0	1	0	-	-	0
1	0	0	1	1	1	1	-	-	0
1	1	0	1	1	0	1	-	-	1
0	0	1	0	0	0	-	1	0	-
0	1	1	0	1	0	-	0	0	-
1	-	1	0	1	1	-	0	1	-
-	0	1	1	0	0	-	1	-	1
0	1	1	1	1	1	-	0	-	0
1	1	1	1	0	0	-	1	-	1

Rezultă ecuațiile:

$$J_1 = I_1 \cdot Q_0$$

$$K_1 = I_1 \cdot Q_0 + \bar{I}_1 \cdot \bar{I}_0$$

$$J_0 = I_1$$

$$K_0 = \bar{I}_1 \cdot \bar{I}_0 + I_1 \cdot I_0 + I_1 \cdot Q_1$$



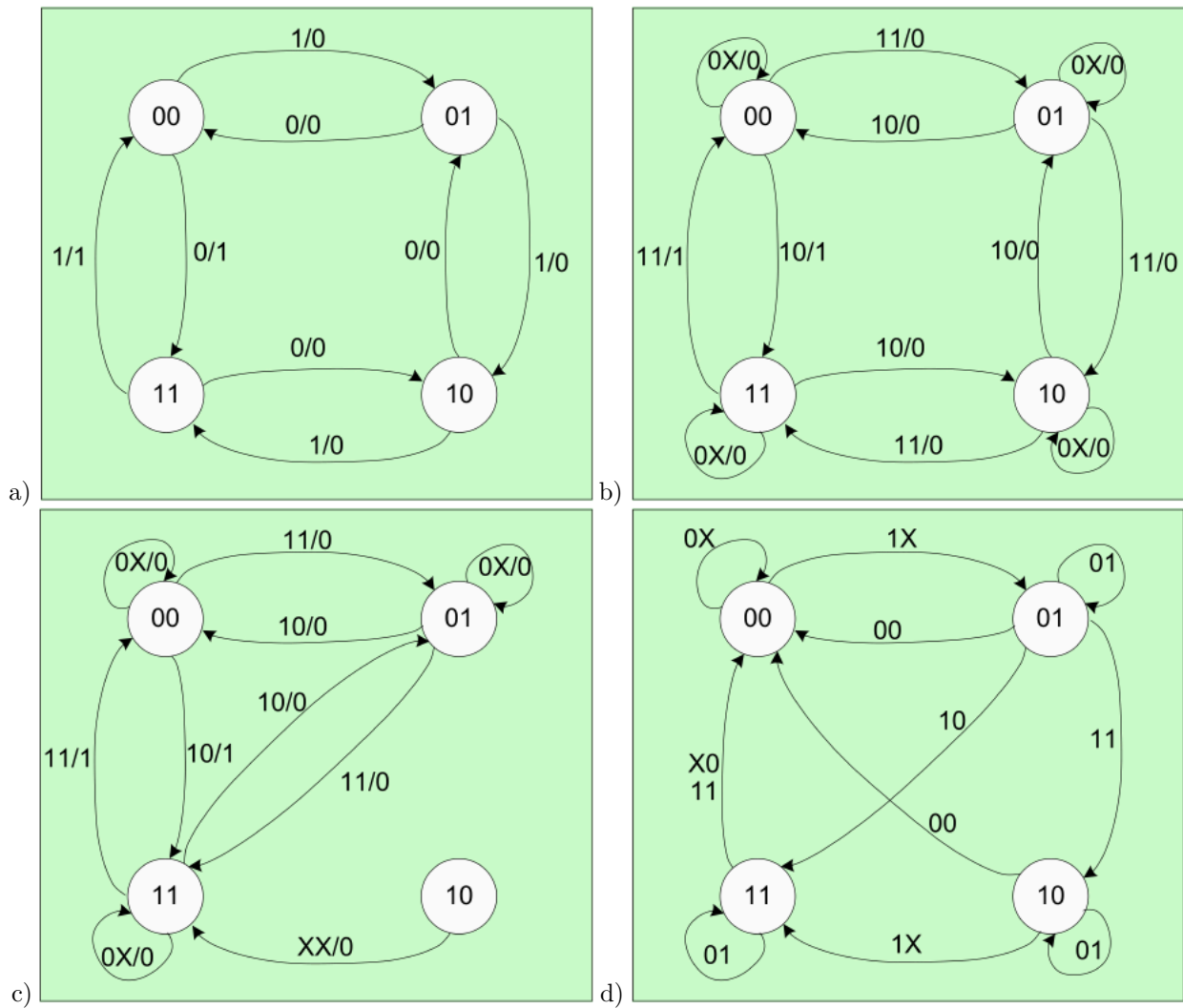


Figura 16.18 Grafuri de tranziții, problema 16.

16.4 Pentru cei ce vor să devină profesioniști

1. Să se implementeze automatul al cărui comportament este descris de organigrama din figura 16.19-a, utilizând doar două bistabile.

Soluție

Se observă că stările **b** și **e** sunt redundante: ieșirea Z este activă în ambele stări iar evoluția automatului este identică din cele două stări (dacă $X = 0$, starea următoare este **a**, iar dacă $X = 1$, starea următoare este **c**). Deci, una din stările **b** sau **e** pot fi eliminate. Organigrama modificată este prezentată în figura 16.19-b.

Având doar 4 stări, automatul se poate implementa cu 2 biți de cod pentru stare, deci, doar două bistabile.

2. Proiectați un automat având două intrări X și Y și o ieșire Z . Ecuațiile intrărilor în bistabile sunt:

$$J_A = A \cdot (B \oplus X), K_A = \overline{B} + Y,$$

$$J_B = B \oplus X \oplus Y, K_B = \overline{A} + X,$$

$$Z = A \cdot B \cdot X \cdot Y.$$

Realizați graful de tranziții al automatului.



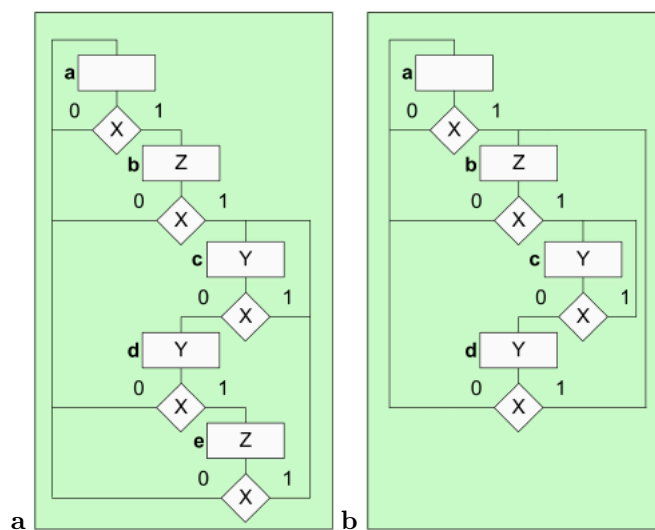


Figura 16.19 Organigrame pentru problema 1: **a** originală, **b** cu stări reduse.

