



UNIUNEA EUROPEANĂ



GUVERNUL ROMÂNIEI



Instrumente Structurale
2007-2013



Platformă de e-learning și curriculum e-content pentru învățământul superior tehnic

Proiectarea Logică

14. Reprezentarea funcțiilor logice – tabelul de adevăr

REPREZENTAREA FUNCȚIILOR LOGICE – TABELUL DE ADEVĂR

APLICAȚII CU CIRCUITE PLA ȘI AFIȘAJE CU 7 SEGMENTE

Un circuit PLA este o macro-celulă rectangulară constând dintr-o arie de tranzistoare aliniate astfel încât să formeze rândurile în corespondență cu termenii produs și coloanele în corespondență cu relația intrări-ieșiri. Coloanele intrărilor și ieșirilor partiționează *PLA* în două sub-arii: planul de intrare și respectiv planul de ieșire (așa cum se poate urmări în figura 4).

Fiecare rând al *PLA*-ului este în corespondență biunivocă cu un termen produs aparținând respectivei sume de produse.

Fiecare tranzistor din planul de intrare este în corespondență biunivocă cu un literal al formei sumă de produse.

Oricare tranzistor din planul de ieșire este legat de o ieșire scalară a termenului produs.

Din aceste motive prima țintă a minimizării logice este reducerea numărului de produse și ținta secundară este reducerea literalilor din produse.

Alte obiective de optimizare sunt relevante atunci când funcțiile modelate prin forme în două nivele sunt implementate altfel decât prin *PLA*-uri.

Reprezentări logice în două nivele pentru funcții scalare, spre exemplu, pot fi implementate prin porți complexe a căror mărime este corelată cu numărul de literalii din forma factorizată a acelei reprezentări. În astfel de situații obiectivul major este minimizarea numărului de literalii.

Minimizarea logică pentru o funcție scalară sau o funcție vectorială se face după aceleași principii, dar cazul vectorial este mult mai complex.

Minimizarea disjunctă a componentelor scalare ale unei funcții vectoriale poate conduce la rezultate suboptimale deoarece optimizarea nu poate exploata comunitatea unor termeni produs.

Un rezultat important în optimizarea logică în două nivele este echivalența funcțiilor vectoriale booleene de variabile booleene cu funcțiile booleene scalare de variabile multi-valorice. Din astfel de rațiuni, pentru început abordarea se va concentra asupra tehnicilor de optimizare ale funcțiilor scalare de variabile binare și multi-valorice.

În vederea reliefării diferențelor dintre minimizarea vectorială și minimizarea scalară a componentelor unei funcții vectoriale se consideră decodificatorul dispozitivului de afișare cu șapte segmente din exemplul următor.

Exemplul 8.

Se consideră proiectarea unui decodificator (a se vedea figura 5) care să preia informația de pe o magistrală cu patru linii (x_8, x_4, x_2 și x_1) pe care sunt vehiculate valori zecimale codificate binar și să activeze corespunzător un dispozitiv de afișare cu șapte segmente.

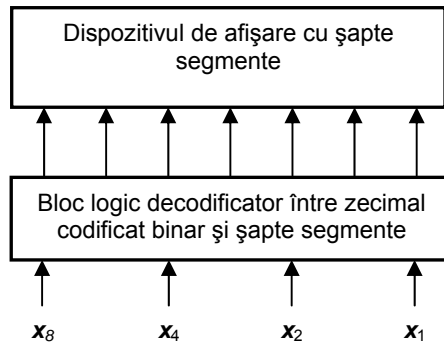


Figura 5. Schema bloc a decodificatorului.

Modul în care sunt formate cifrele zecimale cu dispozitivul de afișare este arătat în figura 6.

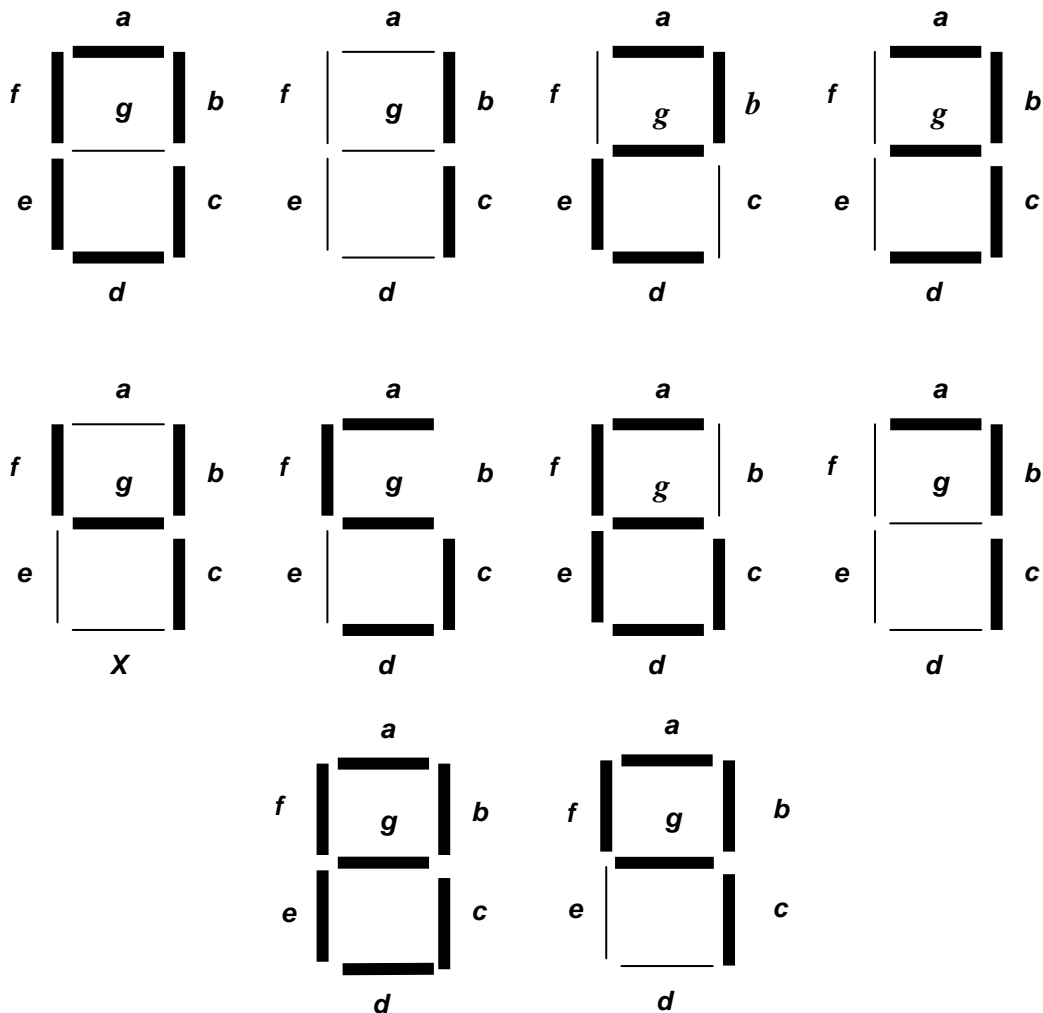


Figura 6. Configurația cifrelor zecimale pentru dispozitivul de afișare cu 7 segmente.

Tabelul 2.
 Valorile liniilor de ieșire ale decodificatorului pentru dispozitivul de
 afișare cu șapte segmente în funcție de valorile liniilor de intrare.

Index	x_8	x_4	x_2	x_1	a	b	c	d	e	f	g
0	0	0	0	0	1	1	1	1	1	1	0
1	0	0	0	1	0	1	1	0	0	0	0
2	0	0	1	0	1	1	0	1	1	0	1
3	0	0	1	1	1	1	1	1	0	0	1
4	0	1	0	0	0	1	1	0	0	1	1
5	0	1	0	1	1	0	1	1	0	1	1
6	0	1	1	0	1	0	1	1	1	1	1
7	0	1	1	1	1	1	1	0	0	0	0
8	1	0	0	0	1	1	1	1	1	1	1
9	1	0	0	1	1	1	1	0	0	1	1
10	1	0	1	0	X	X	X	X	X	X	X
11	1	0	1	1	X	X	X	X	X	X	X
12	1	1	0	0	X	X	X	X	X	X	X
13	1	1	0	1	X	X	X	X	X	X	X
14	1	1	1	0	X	X	X	X	X	X	X
15	1	1	1	1	X	X	X	X	X	X	X

Correspondența dintre valorile binare ale vectorului intrărilor ($x_8 x_4 x_2 x_1$) în decodificator și modul de activare al segmentelor de afișare, conform figurii 6, este prezentat în tabelul 1 (înfățișat anterior).

Deoarece pentru cifrele zecimale sunt utilizate doar codurile binare de la 0000 la 1001, pentru codurile binare rămase neutilizate (de la 1010 la 1111) valorile ieșirilor decodificatorului sunt nespecificate (valori **X**).

Pentru realizarea decodificatorului trebuie aleasă o modalitate de implementare. Se poate implementa decodificatorul cu ajutorul unei memorii cu conținut fix, spre exemplu. În această situație proiectarea se încheie, practic, aici.

Tot ceea ce mai trebuie făcut este încărcarea conținutului coloanelor **a, b, ..., g** din primele 10 linii ale tabelului într-o memorie cu cel puțin 10 locații de câte, minimum, 7 biți fiecare.

Se pot găsi, ușor, astfel de memorii, chiar dacă sunt mai mari decât ar fi necesar. Se poate întâmpla ca o astfel de soluție să nu fie de utilitate din vari motive (costul memoriei, disponibilitatea altor circuite etc.).

De remarcat faptul că există disponibile componente, circuite integrate MSI, deja manufacturate care implementează această funcție combinațională. Este cazul, spre exemplu, componentei **TTL 74LS48** și se mai pot găsi și altele similare.

S-ar putea ca din motive de afișare a cifrei 6 să nu fie acceptabile nici una dintre aceste componente și în această situație trebuie găsite alte soluții.

Deoarece decodificatorul poate fi implementat printr-un circuit combinațional reprezentabil printr-o formulă booleană, sumă de produse, se poate recurge la utilizarea diagramelor Karnaugh sau la utilizarea programului *ESPRESSO* pentru minimizarea euristica (sau exactă).

Ambele căi de abordare vor fi utilizate și comparate pentru înțelegerea particularităților respectivelor metode.

Metoda diagramelor Karnaugh este relativ simplă de aplicat deoarece este convenabil numărul de variabile.

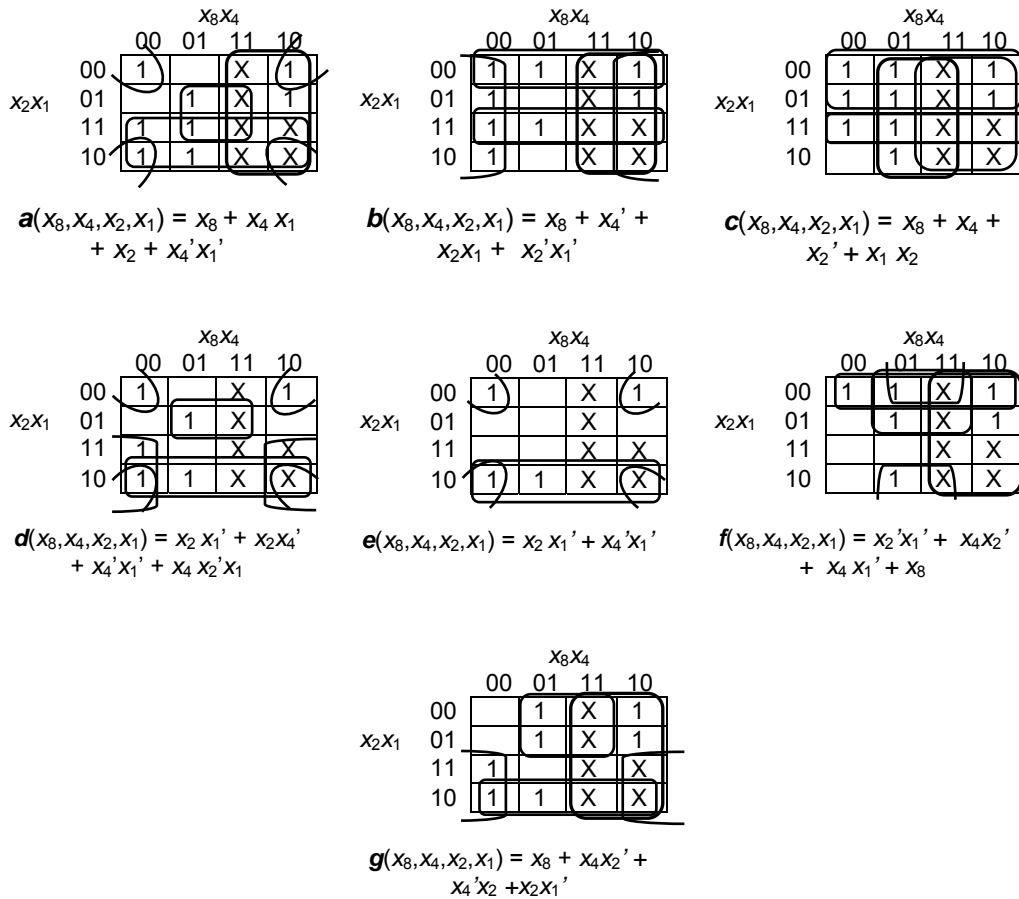


Figura 7. Minimizarea scalară, prin metoda diagramelor Karnaugh, a funcțiilor decodificatorului pentru 7 segmente.

Utilizând rezultatele obținute prin metoda Karnaugh, fără să se considere posibilitatea termenilor partajați, este necesar un circuit PAL cu patru variabile de intrare, șapte linii de ieșire cu cel puțin patru termeni produs (pentru fiecare linie de ieșire) pentru implementarea decodificatorului acesta.

S-ar putea folosi, spre exemplu, componenta P16H8 PAL. Se poate întâmpla ca printre condițiile proiectării circuitului să se specifice utilizarea unui circuit PLA pentru implementare, atunci trebuie ținut cont de faptul că una dintre limitările, relativ severe, ale acestor componente este numărul de termeni produs unici care sunt utilizați.

Aceștia corespund numărului de „fire” orizontale din structura circuitului.

O componentă PLA tipică poate oferi 16 linii de intrare, opt linii de ieșire și 48 de termeni produs (cum este, spre exemplu, componenta F100 PLA).

Metodele de optimizare a circuitelor reprezentate prin sume de produse (sau circuite în două nivele) pun întotdeauna în evidență termenii partajabili.

Fișierul de date de intrare pentru minimizatorul *ESPRESSO* reflectă tabelul de corespondență al decodificatorului:

```

.i 4
.o 7
.ilb x8 x4 x2 x1
.ob a b c d e f g
.p 16
0000 1111110
0001 0110000
0010 1101101
0011 1111001
0100 0110011
0101 1011011
0110 1011111
0111 1110000
1000 1111111
1001 1110011
1010 -----
1011 -----
1100 -----
1101 -----
1110 -----
1111 -----
.e

```

Rezultatele obținute prin metoda de minimizare euristică conduc la doar 9 termeni produs distincți, după cum arată fișierul cu rezultate al minimizatorului *ESPRESSO*:

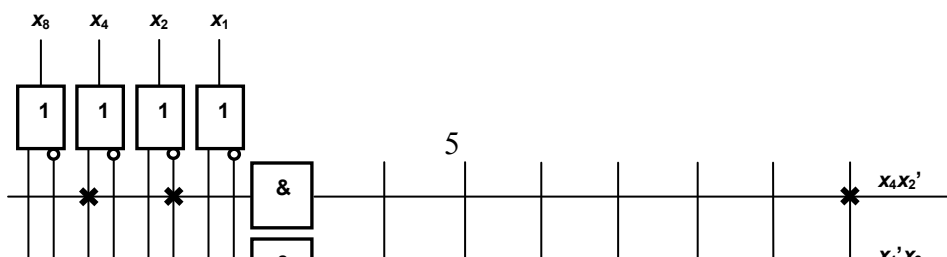
```

.i 4
.o 7
.ilb x8 x4 x2 x1
.ob a b c d e f g
.p 9
-10- 0000001
-01- 0001001
-0-1 0110000
-101 1011010
--00 0110010
--11 1110000
-0-0 1101100
1--- 1000011
-110 1011111
.e

```

Corespunzător acestor termeni produs se pot scrie expresiile fiecărei linii de ieșire:

$$\begin{aligned}
 \mathbf{a} &= x_4 x_2' x_1 + x_2 x_1 + x_4' x_1' + x_8 + x_4 x_2 x_1' \\
 \mathbf{b} &= x_4' x_1 + x_2' x_1' + x_2 x_1 + x_4' x_1' \\
 \mathbf{c} &= x_4' x_1 + x_4 x_2' x_1 + x_2' x_1' + x_2 x_1 + x_4 x_2 x_1' \\
 \mathbf{d} &= x_4' x_2 + x_4 x_2' x_1 + x_4' x_1' + x_4 x_2 x_1' \\
 \mathbf{e} &= x_4' x_1' + x_4 x_2 x_1' \\
 \mathbf{f} &= x_4 x_2' x_1 + x_2' x_1' + x_8 + x_4 x_2 x_1' \\
 \mathbf{g} &= x_4 x_2' + x_4' x_2 + x_8 + x_4 x_2 x_1'
 \end{aligned}$$



Se remarcă, într-o primă aproximație, apariția unei complexități mai mari a funcțiilor segmentelor de afișare (63 de literali) comparativ cu rezultatele obținute prin metoda de minimizare a diagramei Karnaugh.

La o privire mai atentă, totuși, se poate observa faptul că numărul de termeni produs distincți a scăzut de la 15 la 9.

Chiar dacă expresiile individuale ale fiecărei funcții au, fiecare în parte, un număr mai mare de termeni produs, datorită gradului ridicat de partajare a acestora complexitatea, globală a acestei variante, este mai mică.

Mărimea dispozitivelor PLA este determinată, în primul rând, de numărul de termeni produs. Minimizatorul *ESPRESSO*, a fost proiectat, între altele, pentru astfel de circuite.

Partajarea termenilor produs nu este de nici un folos în implementarea circuitelor combinaționale prin componente PAL care nu pot pune în valoare termenii produs partajați dintre liniile de ieșire.

Dacă se urmărește o implementare prin dispozitive PAL este indicat să se specifice minimizatorului *ESPRESSO* o abordare individuală, scalară, a fiecărei funcții în parte (nu în sistem).

□