

4

FUNCTII BINARE

4.1 Algebra booleană

În secolul al XIX-lea, matematicianul englez George Boole (1815-1864) formalizează logica aristotelică, bazată pe dihotomia adevărat-fals, sub forma unei algebre cunoscută sub numele de **algebră Boole** sau **algebră booleană**.

Există mai multe moduri în care poate fi prezentată sistematic logica booleană. Pe de o parte s-a încercat folosirea unui număr minim de axiome: s-a reușit chiar dezvoltarea algebrei Boole pornind de la o singură axiomă (foarte complexă și neintuitivă). Pe de altă parte s-a încercat folosirea unui număr minim de funcții logice de bază. Toate aceste soluții sunt însă limitative din punct de vedere ingineresc, deoarece "realizarea concretă a structurilor numerice este bine sprijinită numai de un cadru teoretic flexibil, în care diversitatea formelor de reprezentare este mult mai importantă decât rigiditatea și austeritatea performanțelor pur formale" ([Ștefan, 2000]).

În cele ce urmează am definit o algebră booleană folosind **postulatele lui Huntington**, un grup de afirmații propuse pentru prima dată în 1904 ([Mowle, 1976]).

O **algebră booleană** este un ansamblu $\langle M, +, \cdot, = \rangle$ format din mulțimea suport M cu un număr finit de elemente, operația binară SAU notată cu simbolul $+$, operația binară ȘI notată cu simbolul $\cdot$ și o relație de echivalență între elementele mulțimii M , notată cu simbolul $=$, dacă sunt îndeplinite următoarele 6 postulate:

- P1** operațiile sunt închise:
 $\forall x, y \in M, \quad x + y \in M; \quad x \cdot y \in M$
- P2** pentru fiecare operație există un element neutru:
 $\forall x \in M, \text{ exista } 0 \in M, \text{ a.i. } x + 0 = x$
 $\forall x \in M, \text{ exista } 1 \in M, \text{ a.i. } x \cdot 1 = x$
- P3** operațiile sunt comutative:
 $\forall x, y \in M, \quad x + y = y + x; \quad x \cdot y = y \cdot x$

P4 operațiile sunt distributive:

$$\forall x, y, z \in M, \quad x + (y \cdot z) = (x + y) \cdot (x + z)$$

$$x \cdot (y + z) = (x \cdot y) + (x \cdot z)$$

P5 pentru fiecare element $x \in M$ există un element $\bar{x} \in M$:

$$\forall x \in M, \text{ exista } \bar{x} \in M, \text{ a.i. } x + \bar{x} = 1; \quad x \cdot \bar{x} = 0$$

P6 există cel puțin 2 elemente distincte în mulțimea M :

$$\exists x, y \in M, \text{ a.i. } x \neq y$$

Se poate ușor observa că fiecare propoziție este exprimată prin două echivalențe boolene. Una se obține din cealaltă prin interschimbarea operațiilor binare SAU cu ȘI și a constantelor 0 cu 1. Această proprietate specială se numește **dualitate**.

Principiul dualității este foarte important pentru că odată ce am demonstrat că o echivalență booleană este adevărată, atunci și echivalența duală este adevărată. Acest lucru se poate demonstra prin aplicarea postulatelor de mai sus în ordinea folosită pentru demonstrația echivalenței inițiale. În general, dualitatea este valabilă pentru orice proprietate booleană.

Folosind postulatele lui Huntington vom demonstra o serie de **teoreme fundamentale**:

T1 Legile lui 1 și 0

$$\forall x \in M, \quad x \cdot 0 = 0; \quad x + 1 = 1$$

Demonstrație:

$$x \cdot 0 \stackrel{P2}{=} (x \cdot 0) + 0 \stackrel{P5}{=} (x \cdot 0) + (x \cdot \bar{x}) \stackrel{P4}{=} x \cdot (0 + \bar{x}) \stackrel{P2}{=} x \cdot \bar{x} \stackrel{P5}{=} 0$$

S-a pornit din membrul stâng și s-a introdus operația SAU (postulatul 2); elementul neutru astfel introdus este înlocuit de $x \cdot \bar{x}$ (postulatul 5) în vederea aplicării distributivității (postulatul 4); se elimină elementul neutru pentru operația SAU (postulatul 2) și se aplică din nou postulatul 5.

Pentru demonstrarea expresiei duale se parcurge aceeași secvență de postulate, dar se folosesc expresiile duale:

$$x + 1 \stackrel{P2}{=} (x + 1) \cdot 1 \stackrel{P5}{=} (x + 1) \cdot (x + \bar{x}) \stackrel{P4}{=} x + (1 \cdot \bar{x}) \stackrel{P2}{=} x + \bar{x} \stackrel{P5}{=} 1$$

T2 Legile complementului

Complementul lui 0 este $\bar{0}$, iar $\bar{\bar{0}} = 1$; Complementul lui 1 este $\bar{1}$, iar $\bar{\bar{1}} = 0$.

Demonstrație:

$$x + 0 \stackrel{P2}{=} x, \text{ deci } \bar{0} + 0 = \bar{0}. \text{ Dar } \bar{0} + 0 \stackrel{P5}{=} 1, \text{ deci } \bar{0} = 1.$$

A doua relație se poate demonstra prin dualitate.

T3 Legea de unicitate a complementului

Orice element $x \in M$ are un complement unic, notat prin $\bar{x}$

Demonstrație:

Presupunem că elementul x are două complemente $\bar{x}_1$ și $\bar{x}_2$

$$\begin{aligned} \bar{x}_1 + 0 &\stackrel{P2}{=} \bar{x}_1 \stackrel{P5}{=} \bar{x}_1 + (x \cdot \bar{x}_2) \stackrel{P4}{=} (\bar{x}_1 + x) \cdot (\bar{x}_1 + \bar{x}_2) \stackrel{P5}{=} 1 \cdot (\bar{x}_1 + \bar{x}_2) \stackrel{P5}{=} \\ &= (\bar{x}_2 + x) \cdot (\bar{x}_2 + \bar{x}_1) \stackrel{P4}{=} \bar{x}_2 + (x \cdot \bar{x}_1) \stackrel{P5}{=} \bar{x}_2 + 0 \stackrel{P2}{=} \bar{x}_2 \end{aligned}$$

T4 Legile de idempotență

$$\forall x \in M, \quad x + x = x; \quad x \cdot x = x$$

Demonstrație:

$$x + x \stackrel{P2}{=} (x + x) \cdot 1 \stackrel{P5}{=} (x + x) \cdot (x + \bar{x}) \stackrel{P4}{=} x + (x \cdot \bar{x}) \stackrel{P5}{=} x + 0 \stackrel{P2}{=} x$$

A doua relație se poate demonstra prin dualitate.

T5 Legile de absorbție

$$\forall x, y \in M, \quad x + (x \cdot y) = x; \quad x \cdot (x + y) = x$$

Demonstrație:

$$x + (x \cdot y) \stackrel{P2}{=} (x \cdot 1) + (x \cdot y) \stackrel{P4}{=} x \cdot (1 + y) \stackrel{T1}{=} x \cdot 1 \stackrel{P2}{=} x$$

Demonstrația este simplificată aici prin folosirea legilor lui 1 și 0 (T₁). A doua relație se poate demonstra prin dualitate.

T6 Legea de involuție

$$\forall x \in M, \quad \bar{\bar{x}} = x$$

Demonstrație:

$$\bar{\bar{x}} \stackrel{P2}{=} \bar{x} + 0 \stackrel{P5}{=} \bar{x} + (x \cdot \bar{x}) \stackrel{P4}{=} (\bar{x} + x) \cdot (\bar{x} + \bar{x}) \stackrel{P5}{=} (\bar{x} + x) \cdot 1 \stackrel{P2}{=} (\bar{x} + x)$$

$$\bar{\bar{x}} \stackrel{P2}{=} \bar{x} \cdot 1 \stackrel{P5}{=} \bar{x} \cdot (x + \bar{x}) \stackrel{P4}{=} (\bar{x} \cdot x) + (\bar{x} \cdot \bar{x}) \stackrel{P5}{=} (\bar{x} \cdot x) + 0 \stackrel{P2}{=} (\bar{x} \cdot x)$$

Prin substituție obținem următorul rezultat: $x + \bar{\bar{x}} \cdot x = \bar{\bar{x}}$. Pe de altă parte, conform legilor de absorbție (T5): $x + \bar{\bar{x}} \cdot x = x$. Putem renunța la paranteze, dacă admitem precedența operației ȘI asupra lui SAU. Concluzia este că $\bar{\bar{x}} = x$.

T7 Legile de asociativitate

$$\forall x, y, z \in M, \quad (x + y) + z = x + (y + z); \quad (x \cdot y) \cdot z = x \cdot (y \cdot z)$$

Demonstrație:

Notăm $(x + y) + z = U$ și $x + (y + z) = V$. Se demonstrează următoarele echivalențe: $U \cdot V = U$ și $U \cdot V = V$. Prin proprietatea de tranzitivitate a relației de echivalență rezultă $U = V$. A doua relație se poate demonstra prin dualitate.

$$\begin{aligned} ((x + y) + z) \cdot (x + (y + z)) &\stackrel{P4}{=} (((x + y) + z) \cdot x) + (((x + y) + z) \cdot (y + z)) \stackrel{P4, T4, T5}{=} \\ &= x + (((x + y) + z) \cdot (y + z)) \stackrel{P4}{=} x + (((x + y) + z) \cdot y) + (((x + y) + z) \cdot z) \stackrel{P4, T4, T5}{=} \\ &= x + (y + ((x + y) + z) \cdot z) \stackrel{P4, T4, T5}{=} x + (y + z). \text{ Deci s-a arătat că } U \cdot V = V. \end{aligned}$$

$$\begin{aligned} ((x + y) + z) \cdot (x + (y + z)) &\stackrel{P4}{=} (((x + y) \cdot (x + (y + z))) + (z \cdot (x + (y + z)))) \stackrel{P4, T4, T5}{=} \\ &= ((x + y) \cdot (x + (y + z))) + z \stackrel{P4}{=} ((x \cdot (x + (y + z))) + (y \cdot (x + (y + z)))) + z \stackrel{P4, T4, T5}{=} \\ &= (((x \cdot (x + (y + z))) + y)) + z \stackrel{P4, T4, T5}{=} (x + y) + z. \text{ Deci s-a arătat că } U \cdot V = U. \end{aligned}$$

Calcululele efectuate aici folosesc multe paranteze. Există unele convenții care derivă din reprezentările algebrice uzuale și care au rolul de a simplifica scrierea expresiilor boolene. Se obișnuiește ca operația ȘI să aibă precedență asupra operației SAU, iar uneori să fie chiar omisă în reprezentare. De exemplu, expresia $xy + z$ trebuie interpretată ca fiind $((x \cdot y) + z)$.

T8 Legile lui De Morgan

$$\forall x, y \in M, \quad \overline{x + y} = \overline{x} \cdot \overline{y}; \quad \overline{x \cdot y} = \overline{x} + \overline{y}$$

Demonstrație:

$$(x + y) \cdot (\overline{x} \cdot \overline{y}) \stackrel{P4}{=} (x \cdot \overline{x} \cdot \overline{y}) + (y \cdot \overline{x} \cdot \overline{y}) \stackrel{T7, P5, P2}{=} 0$$

$$(x + y) + (\overline{x} \cdot \overline{y}) \stackrel{T7}{=} (x + \overline{x} \cdot \overline{y}) + y \stackrel{P4, P5, P2}{=} x + \overline{y} + y \stackrel{P5, T1}{=} 1$$

Expresiile $x + y$ și $\overline{x} \cdot \overline{y}$ sunt expresii boolene, deci elemente din mulțimea M , și se supun postulatului 5. Deci prima lege, care poartă numele lui Augustus De Morgan (1806-1871), a fost demonstrată. Echivalența duală este și ea adevărată.

4.2 Funcții binare

O **funcție binară** cu n intrări este o aplicație a mulțimii $\{0,1\}^n$ în mulțimea formată din 2 elemente $\{0,1\}$. Domeniul de definiție al funcției este mulțimea configurațiilor binare de n biți: $\{0,1\}^n = \{(x_1 x_2 \dots x_n) | x_1 \in \{0,1\}, x_2 \in \{0,1\}, \dots, x_n \in \{0,1\}\}$. Cardinalul mulțimii $\{0,1\}^n$ este 2^n .

Numărul maxim de funcții binare distincte care pot fi generate cu n variabile se poate stabili ușor: pentru fiecare dintre cele 2^n configurații binare posibile o funcție are valorile 0 sau 1; dacă pentru i configurații funcția are valoarea 1, atunci pentru celelalte are valoarea 0, iar numărul de funcții distincte este $C_{2^n}^i$. Cum numărul i variază de la 0 până la 2^n , numărul maxim de funcții binare N este:

$$N = \sum_{i=0}^{2^n} C_{2^n}^i = 2^{2^n}$$

Se poate observa că pentru $n = 2$ există 16 funcții binare distincte, care sunt reprezentate prin tabelul de adevăr 4.1.

Tabelul 4.1

x	y	f_0	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}
0	0	0	1	0	0	0	1	1	1	0	0	0	1	1	1	0	1
0	1	0	0	1	0	0	1	0	0	1	1	0	1	0	1	1	1
1	0	0	0	0	1	0	0	1	0	1	0	1	1	1	0	1	1
1	1	0	0	0	0	1	0	0	1	0	1	1	0	1	1	1	1

Funcțiile f_0 și f_{15} sunt constante și sunt de fapt cele 2 funcții care pot fi generate cu $n = 0$ variabile. Ele se mai numesc **funcție nulară**, respectiv **funcție unară**.

Funcția f_{10} este **funcția identitate** pentru variabila x , iar funcția f_5 este **negația** variabilei x . Setul de funcții f_0 , f_{15} , f_{10} și f_5 reprezintă numărul maxim de funcții binare generate cu variabila x . Setul de funcții f_0 , f_{15} , f_9 și f_6 reprezintă același lucru pentru y .

Toate celelalte funcții din tabel sunt funcții de două variabile, variabile notate aici cu x și y . Ele se implementează cu ajutorul unor **operatori logici** sau **porți logice**. Există denumiri consacrate pentru cele mai importante funcții binare, precum și **reprezentări grafice standard**. Tabelul din figura 4.1 ilustrează o parte dintre ele ([Mange, 1987]).

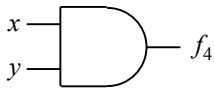
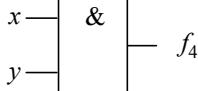
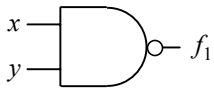
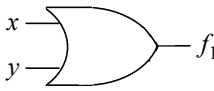
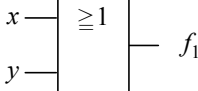
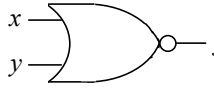
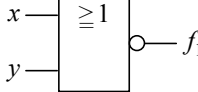
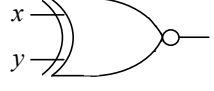
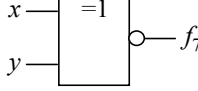
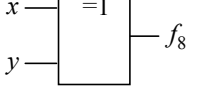
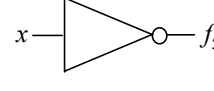
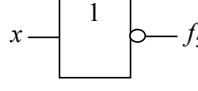
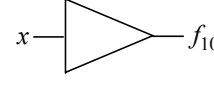
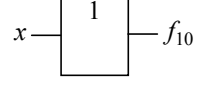
Denumirea funcției și relația booleană	Denumirea circuitului	Reprezentări grafice standard	
		MIL-STD-806B	CEI
conjunția $f_4 = x \cdot y$	ȘI (AND)		
negarea conjunției $f_{11} = \overline{x \cdot y}$	ȘI-NU (NAND)		
disjuncția $f_{14} = x + y$	SAU (OR)		
negarea disjuncției $f_1 = \overline{x + y}$	SAU-NU (NOR)		
echivalența $f_7 = \overline{x \oplus y}$	COINCIDENȚA (COMPARATOR)		
negarea echivalenței $f_8 = x \oplus y$	SAU EXCLUSIV (XOR)		
negația $f_5 = \overline{x} ; f_6 = \overline{y}$	INVERSOR (NOT)		
identitate $f_{10} = x ; f_9 = y$	NEINVERSOR (BUFFER)		

Fig. 4.1 Porți logice care implementează funcții binare

Nu toate funcțiile din tabelul 4.1 sunt reprezentate în figura de mai sus. Funcția f_{13} , de exemplu, are valoarea logică 0 dacă $x = 1$ și $y = 0$. Funcția se numește **implicație directă**, iar relația booleană se scrie $f_{13} = x \rightarrow y = \overline{x} + y$. Circuitul logic nu are o denumire consacrată pentru că el nu poate fi implementat cu o singură poartă logică. Expresia algebrică arată că sunt necesare cel puțin două porți: un inversor și o poartă SAU cu 2 intrări. Funcția f_{12} este **implicația inversă**, care se scrie $f_{12} = y \rightarrow x = x + \overline{y}$. Funcția f_3 are aceleași valori cu coloana variabilei x , cu excepția cazului $x = y = 1$, motiv pentru care se numește **interdicție**, în sensul y interzice x . Ea este de fapt **negația implicației directe**. Funcția f_2 este interdicția în sensul x interzice y și **negația implicației inverse**. Și pentru implementarea interdicțiilor sunt necesare cel puțin două porți logice.

Există mai multe reprezentări grafice standard pentru porțile logice. Tabelul din figura 4.1 le prezintă pe cele mai uzuale. Primul standard folosit este standardul militar american **MIL-STD-806B**, care este încă preferat de majoritatea utilizatorilor, pentru că este simplu și sugestiv. Celălalt standard a fost stabilit de Comisia Electrotehnică Internațională (**Commission Électrotechnique Internationale-CEI**) și are recunoaștere internațională la ora actuală. Un standard foarte apropiat de acesta din urmă este standardul american **ANSI/IEEE STD 91-1984**.

4.3 Moduri de reprezentare

Funcțiile binare se reprezintă de obicei prin **tabele de adevăr**, **expresii boolene**, **forme geometrice** sau **diagrame Veitch-Karnaugh**.

Reprezentarea prin **tabel de adevăr**, cum este de exemplu tabelul 4.1, este cea mai simplă formă de reprezentare. Fiecare variabilă binară a funcției are câte o coloană, iar numărul maxim de combinații binare posibile stabilește numărul de linii din tabel. Valorile funcției pentru fiecare din aceste combinații binare sunt trecute într-o coloană separată. Dacă funcția are însă multe variabile, atunci dimensiunile tabelului pot deveni mult prea mari.

Expresia booleană a unei funcții binare poate fi o **formă canonică**, o **formă elementară** sau o **formă neelementară**. O formă canonică este o expresie care conține **termeni canonici**, adică termeni care conțin toate variabilele independente ale funcției, sub formă directă sau negată. Un termen canonic poate fi de tip **produs logic**, și se notează cu P_i , sau de tip **sumă logică**, și se notează cu S_i , unde i variază de la 0 la $2^n - 1$, dacă funcția are n variabile.

Forma canonică disjunctivă a unei funcții binare este formată din disjuncția termenilor P_i și se scrie sub forma $f_{cd} = \bigcup_{i=0}^{2^n-1} (a_i \cdot P_i)$, unde a_i este 0 sau 1, în funcție de absența sau prezența termenului P_i din sumă.

Înainte de a exemplifica reprezentarea prin această formă canonică, vom enunța o teoremă, cunoscută sub numele de **teorema de expansiune booleană**:

Dacă $f: \{0,1\}^n \rightarrow \{0,1\}$ este o funcție binară, atunci pentru orice $(x_1, x_2, \dots, x_n) \in \{0,1\}^n$ se poate scrie: $f(x_1, x_2, \dots, x_n) = \bar{x}_1 \cdot f(0, x_2, \dots, x_n) + x_1 \cdot f(1, x_2, \dots, x_n)$.

Demonstrația este evidentă: dacă $x_1 = 0$, atunci $f(x_1, x_2, \dots, x_n) = f(0, x_2, \dots, x_n)$, iar dacă $x_1 = 1$, atunci $f(x_1, x_2, \dots, x_n) = f(1, x_2, \dots, x_n)$.

Exemplul 4.1

Funcția de 2 variabile descrisă prin tabelul de adevăr de mai jos se poate scrie:

$$f(x_1, x_2) = \bar{x}_1 \cdot \bar{x}_2 \cdot f(0,0) + \bar{x}_1 \cdot x_2 \cdot f(0,1) + x_1 \cdot \bar{x}_2 \cdot f(1,0) + x_1 \cdot x_2 \cdot f(1,1)$$

x_1	x_2	$f(x_1, x_2)$
0	0	1
0	1	0
1	0	1
1	1	1

$$f(x_1, x_2) = \bar{x}_1 \cdot \bar{x}_2 + x_1 \cdot \bar{x}_2 + x_1 \cdot x_2 =$$

$$= P_0 + P_2 + P_3 = \sum(0, 2, 3)$$

Funcția are numai două variabile, deci este scrisă ca o sumă logică de termeni canonici de tip produs. Indicii termenilor P_i sunt pozițiile din tabel pentru care valorile funcției sunt 1 logic, poziții numerotate de la 0 la $2^n - 1$, adică 3. Indicii se pot deduce și din expresia algebrică a funcției, dacă luăm fiecare termen canonic și facem conversia în zecimal a numărului binar obținut prin înlocuirea variabilei negată cu 0 și a variabilei directe cu 1, ținând seama de rangul fiecărei variabile. Aici x_1 este pe poziția bitului cel mai semnificativ, iar x_2 pe cea a bitului cel mai puțin semnificativ.

Forma canonică conjunctivă a unei funcții binare este formată din conjuncția termenilor S_i și se scrie sub forma $f_{cc} = \bigcap_{i=0}^{2^n-1} (a_i + S_i)$, unde a_i este 0 sau 1, în funcție de prezența sau absența termenului S_i din produs.

Exemplul 4.2

Funcția $f(x_1, x_2)$ din exemplul 4.1 devine $f(x_1, x_2) = x_1 + \bar{x}_2 = S_1$. Se vede imediat că $\bar{f}(x_1, x_2) = P_1 = \bar{x}_1 \cdot x_2$, deci $f(x_1, x_2) = \overline{\bar{x}_1 \cdot x_2} = x_1 + \bar{x}_2$, conform legilor lui De Morgan. Rezultatul obținut este de fapt un termen canonic de tip sumă, numai că de această dată variabila negată se înlocuiește cu 1 și cea directă cu 0.

Iată un alt exemplu: $f(x_1, x_2, x_3) = (x_1 + x_2 + \bar{x}_3) \cdot (x_1 + \bar{x}_2 + x_3) \cdot (x_1 + \bar{x}_2 + \bar{x}_3) = S_1 \cdot S_2 \cdot S_3 = \prod(1, 2, 3)$.

Forma elementară a unei funcții binare conține termeni care nu sunt canonici. Formele elementare pot fi disjunctive sau conjunctive. O funcție poate fi reprezentată în **formă neelementară** dacă există doi sau mai mulți termeni care conțin aceeași variabilă.

Exemplul 4.3

Funcția de 3 variabile $f(x_1, x_2, x_3)$ din exemplul anterior se poate reprezenta în două forme elementare disjunctive, $f(x_1, x_2, x_3) = \bar{x}_2 \cdot \bar{x}_3 + x_1 \cdot x_2 + x_1 \cdot x_3 = \bar{x}_2 \cdot \bar{x}_3 + x_1$, într-o formă elementară conjunctivă, $f(x_1, x_2, x_3) = (x_1 + \bar{x}_2) \cdot (x_1 + \bar{x}_3)$, sau într-o formă neelementară, $f(x_1, x_2, x_3) = \bar{x}_2 \cdot \bar{x}_3 + x_1 \cdot (x_2 + x_3)$.

Toate aceste forme algebrice se mai numesc și **forme normale** pentru că sunt alcătuite din **termeni normali**, adică termeni în care nici una dintre variabile nu apare decât o singură dată. Un termen care nu este normal poate fi redus întotdeauna la un termen normal, sau la o constantă, prin folosirea postulatelor și teoremelor algebrei boolene.

Un termen canonic de tip produs se mai numește și **mintermen** pentru că poate fi definit ca termen produs care are valoarea logică 1, pentru o singură combinație binară a variabilelor funcției, sau altfel spus, pentru o singură linie a tabelului de adevăr.

Un termen canonic de tip sumă se mai numește și **maxtermen** pentru că poate fi definit ca termen sumă care are valoarea logică 0, pentru o singură combinație binară a variabilelor funcției, sau pentru o singură linie a tabelului de adevăr. Pentru toate celelalte $2^n - 1$ combinații binare maxtermenul are valoarea logică 1, deci ocupă cu această valoare numărul maxim de combinații posibile pentru o funcție diferită de constanta 1.

O funcție binară de n variabile poate fi reprezentată **geometric** pe un **cub n -dimensional**. Fiecare vârf al cubului are anumite coordonate și corespunde unui termen canonic de tip produs. Vom porni de la forma canonică disjunctivă a funcției și vom marca cu mici sfere vârfurile corespunzătoare termenilor canonici de tip produs. Funcția poate fi reprezentată printr-un set de **subcuburi de diferite dimensiuni** care acoperă în totalitate vârfurile marcate ale cubului. Pentru a vizualiza ușor cubul în spațiul tridimensional vom exemplifica reprezentarea geometrică folosind o funcție de 3 variabile. Figura 4.2 ilustrează reprezentarea funcției $f(x_1, x_2, x_3)$ din exemplele anterioare.

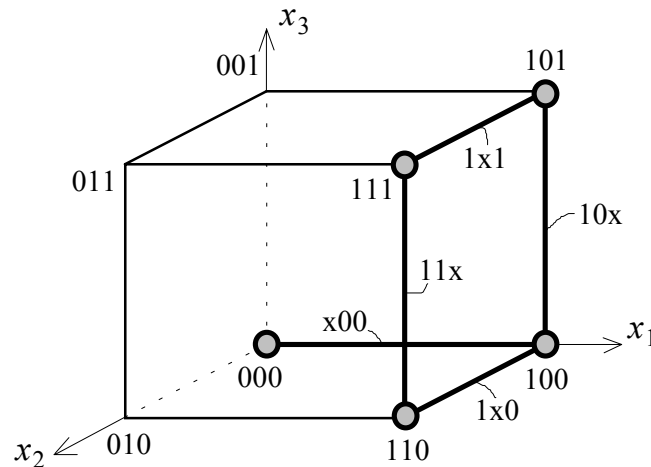


Fig. 4.2 Reprezentarea geometrică a funcției binare $f(x_1, x_2, x_3)$

Exemplul 4.4

Funcția de 3 variabile $f(x_1, x_2, x_3)$ din exemplele anterioare se scrie sub forma canonică disjunctivă $f(x_1, x_2, x_3) = P_0 + P_4 + P_5 + P_6 + P_7$. Vârfurile cubului corespunzătoare mintermenilor funcției se marchează. Termenului $P_0(\bar{x}_1\bar{x}_2\bar{x}_3)$ îi corespunde originea de coordonate $x_1x_2x_3 = 000$, termenului $P_4(x_1\bar{x}_2\bar{x}_3)$ îi corespunde vârful de coordonate $x_1x_2x_3 = 100$, și așa mai departe. Vârfurile cubului sunt subcuburi 0-dimensionale și funcția este complet acoperită de toate cele 5 subcuburi de dimensiune 0: 000, 100, 101, 110 și 111.

Oricare două vârfuri alăturate ale cubului sunt legate printr-o muchie. Aceste laturi formează subcuburi 1-dimensionale și funcția este complet acoperită de toate cele 5 subcuburi de dimensiune 1: x00, 10x, 1x1, 1x0 și 11x. Simbolul x arată că valoarea variabilei respective nu mai contează. Conectarea nodurilor 000 și 100 duce la formarea subcubului 1-dimensional x00. Valorile nule ale variabilelor x_2 și x_3 sunt cele care localizează muchia respectivă.

Vârfurile de coordonate 100, 101, 110 și 111 aparțin aceluiași plan, care formează un subcub 2-dimensional: 1xx. Cele două subcuburi 1xx și x00 formează o **acoperire** completă a funcției $f(x_1, x_2, x_3)$.

O funcție binară f **implică** o altă funcție binară g , dacă pentru orice combinație de variabile pentru care $f = 1$, atunci și $g = 1$. Deci $g = 1$ pentru toate combinațiile binare de intrare pentru care și $f = 1$ și poate pentru încă alte câteva. Se mai spune că g îl **include** pe f , sau că g îl **acoperă** pe f .

Un **implicant prim** al unei funcții binare este un termen produs care implică funcția, dar eliminarea oricărei variabile din expresia lui face ca ceea ce rămâne să nu mai implice funcția. În reprezentarea geometrică implicantul prim este un subcub care nu este inclus într-un subcub de dimensiune mai mare. Implicantii primi din exemplul de mai sus sunt subcuburile 1xx și x00.

Un **implicant prim esențial** este un implicant prim care conține cel puțin un subcub 0-dimensional, care îi aparține numai lui. În exemplul considerat subcuburile 1xx și x00 sunt implicantii primi esențiali pentru că subcubul 000 îi aparține numai subcubului x00, iar subcuburile 110, 101 și 111 aparțin numai subcubului 1xx.

Diagramele Veitch-Karnaugh oferă o reprezentare grafică foarte sugestivă a funcției binare. O diagramă Veitch-Karnaugh pentru o funcție binară de n variabile este un tablou care conține 2^n compartimente, câte unul pentru fiecare din cele 2^n combinații binare posibile ale variabilelor funcției. În interiorul acestor compartimente se trec valorile funcției, 1 sau 0, pentru fiecare combinație de variabile. Fiecare compartiment corespunde de fapt unei linii din tabelul de adevăr, sau unui mintermen, sau unui vârf al cubului n -dimensional.

În figura 4.3 sunt reprezentate două diagrame pentru o funcție de 4 variabile. Ele sunt aproape identice, diferă numai modul de reprezentare a variabilelor care definesc liniile și coloanele tablourilor. Diagrama din stânga este o diagramă Veitch, iar cea din dreapta o diagramă Karnaugh. În cele ce urmează preferăm reprezentarea Veitch, care, odată ce suntem obișnuiți cu ea, pare mai eficientă, dar vom păstra denumirea tradițională de diagramă Veitch-Karnaugh.

Variabilele pot fi amplasate și altfel, nu neapărat așa cum am făcut noi în figura 4.3. Singura condiție este ca două compartimente alăturate să fie definite de combinații binare de variabile care să fie adiacente, adică să difere printr-un singur bit. Pentru că tabloul are 4 linii și 4 coloane, două dintre cele 4 variabile vor defini liniile, iar celelalte două coloanele. Secvențele care caracterizează liniile și coloanele sunt 00, 01, 11 și 10, adică numărarea în cod Gray. Observăm că la trecerea de la o configurație la alta se modifică un singur bit. În acest fel, două compartimente vecine sunt adiacente, inclusiv coloana din stânga cu cea din dreapta și rândul de sus cu cel de jos.

Pentru un număr mai mare de variabile nu se mai pot vizualiza în plan compartimentele adiacente. O soluție este reprezentarea în 3 dimensiuni a diagramelor. Astfel, pentru funcții de 5 variabile vom avea două diagrame ca cele de mai sus, dispuse în două plane adiacente, în total 32 de compartimente. Pentru funcții de 6 variabile se pot imagina patru plane similare, definite de secvențele în cod Gray ale celor două variabile noi, dar pentru un număr și mai mare de variabile nu se mai poate face reprezentarea funcției.

Figura 4.4 exemplifică reprezentarea funcției de 3 variabile din exemplele anterioare.

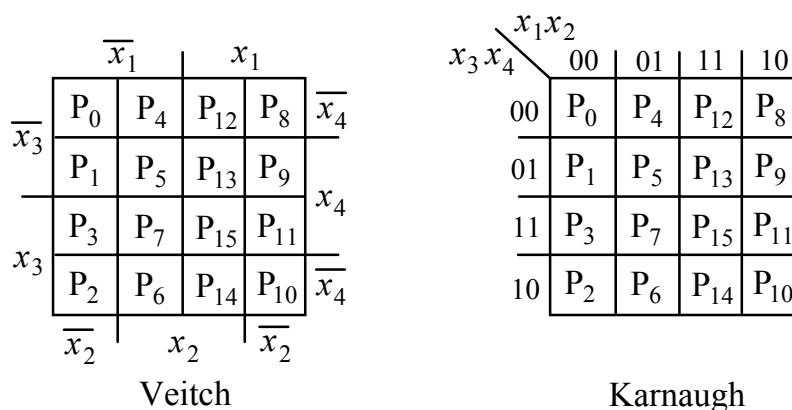


Fig. 4.3 Diagrama Veitch și diagrama Karnaugh pentru o funcție de 4 variabile

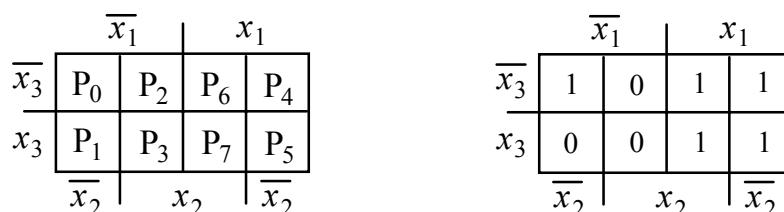


Fig. 4.4 Diagrama Veitch pentru funcția $f(x_1, x_2, x_3) = x_1 + \bar{x}_2 \cdot \bar{x}_3$

4.4 Metode de minimizare

Este normal ca implementarea cu porți logice a unei funcții binare să se facă pornind de la o expresie booleană cât mai simplă, care se numește **forma minimă** a funcției. Există mai multe metode de minimizare care pot fi utilizate de proiectant: **minimizarea algebrică**, **minimizarea cu diagrame Veitch-Karnaugh**, **metoda Quine-McCluskey**, **metoda consensurilor** sau **metoda ESPRESSO**.

Minimizarea algebrică presupune aplicarea directă a postulatelor și teoremelor algebrei boolene. Eficiența metodei depinde de abilitatea și experiența celui care o utilizează. Nu este recomandată pentru că, în lipsa unei formalizări suficiente, nimic nu garantează găsirea formei minime a funcției.

Minimizarea cu diagrame Veitch-Karnaugh oferă o convergență mai sigură către o soluție minimală. Este optimă pentru minimizarea manuală a funcțiilor cu număr redus de variabile pentru că permite vizualizarea grafică a implicantilor primi și găsirea imediată a unei acoperiri convenabile.

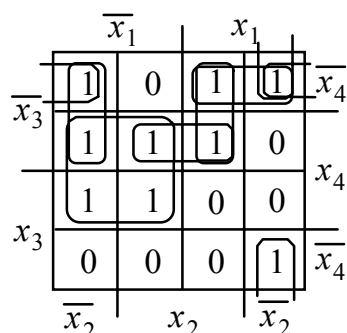
Algoritmul de extragere a formei minime presupune parcurgerea următoarelor etape:

1. stabilirea numărului minim de suprafețe dreptunghiulare de arie maximă formate dintr-un număr de compartimente egal cu o putere a lui 2, care conțin 1 logic
2. scrierea produselor logice asociate fiecărei suprafețe evidențiate la punctul 1
3. sumarea logică a produselor.

Înainte de a exemplifica această metodă vom redefini o serie de noțiuni, în concordanță cu reprezentarea lor pe diagrama Veitch-Karnaugh. O grupare de compartimente care nu poate fi inclusă într-o grupare mai mare formează un implicant prim. Un implicant prim care conține cel puțin un compartiment care nu intră în componența altui implicant prim este un implicant prim esențial. Mulțimea implicantilor primi esențiali formează **nucleul** funcției. Forma minimă a funcției conține nucleul și eventual o parte din implicantii primi neesențiali, adică implicantii primi care nu sunt esențiali, astfel încât toate compartimentele care conțin valoarea logică 1 să fie acoperite.

Exemplul 4.5

Funcția reprezentată pe diagrama Veitch-Karnaugh din figura 4.5 are două forme minime perfect echivalente: $f(x_1, x_2, x_3, x_4) = \bar{x}_1 \cdot x_4 + x_1 \cdot \bar{x}_2 \cdot \bar{x}_4 + x_1 \cdot x_2 \cdot \bar{x}_3 + \bar{x}_1 \cdot \bar{x}_2 \cdot \bar{x}_3 = \bar{x}_1 \cdot x_4 + x_1 \cdot \bar{x}_2 \cdot \bar{x}_4 + x_1 \cdot x_2 \cdot \bar{x}_3 + \bar{x}_2 \cdot \bar{x}_3 \cdot \bar{x}_4$.



nucleul: $\bar{x}_1 x_4$, $x_1 \bar{x}_2 \bar{x}_4$

implicantii primi neesențiali:

$x_2 \bar{x}_3 x_4$, $x_1 x_2 \bar{x}_3$, $x_1 \bar{x}_3 \bar{x}_4$

$\bar{x}_1 \bar{x}_2 \bar{x}_3$, $\bar{x}_2 \bar{x}_3 \bar{x}_4$

Fig. 4.5 Lista implicantilor primi pentru funcția dată

Pentru a evalua comparativ diverse forme minime s-a introdus un criteriu global de minimizare, care are în vedere atât numărul de porți logice cât și numărul total de intrări folosite, fără a mai lua în calcul și inversoarele. Cele două reprezentări de mai sus folosesc 5 porți (4 porți ȘI și o poartă SAU) cu un număr total de 15 intrări.

Prin complementarea diagramei de mai sus, sau altfel spus, prin gruparea compartimentelor care conțin valoarea logică 0, obținem complementul funcției f , adică: $\bar{f}(x_1, x_2, x_3, x_4) = \bar{x}_1 \cdot x_2 \cdot \bar{x}_4 + \bar{x}_1 \cdot x_3 \cdot \bar{x}_4 + x_1 \cdot \bar{x}_2 \cdot x_4 + x_1 \cdot x_2 \cdot x_3$. Expresia funcției f devine: $f(x_1, x_2, x_3, x_4) = (x_1 + \bar{x}_2 + x_4) \cdot (x_1 + \bar{x}_3 + x_4) \cdot (\bar{x}_1 + x_2 + \bar{x}_4) \cdot (\bar{x}_1 + \bar{x}_2 + \bar{x}_3)$ prin folosirea de două ori a legilor lui De Morgan. Costul este acum de 5 porți și 16 intrări, deci nici un avantaj față de reprezentarea anterioară.

Să considerăm acum un alt exemplu din care vom vedea că gruparea compartimentelor care conțin valoarea logică 0 ar putea fi uneori avantajoasă:

Exemplul 4.6

Să considerăm funcția reprezentată pe diagrama din figura 4.6. Forma minimă obținută prin gruparea compartimentelor care conțin 1 logic este dată în figură. Costul acestei reprezentări este de 4 porți și 11 intrări.

Dacă evaluăm complementul funcției f : $\bar{f}(x_1, x_2, x_3, x_4) = x_4 + \bar{x}_1 \cdot x_2 \cdot x_3 + x_1 \cdot x_2 \cdot \bar{x}_3$, deci $f(x_1, x_2, x_3, x_4) = \bar{x}_4 \cdot (x_1 + \bar{x}_2 + \bar{x}_3) \cdot (\bar{x}_1 + \bar{x}_2 + x_3)$, obținem un cost de 3 porți și 9 intrări, un avantaj substanțial față de soluția anterioară.

În figura 4.7 sunt prezentate două implementări diferite ale funcției $f(x_1, x_2, x_3, x_4)$. Prima soluție folosește ecuația formei minime de mai sus, iar cealaltă folosește expresia $f(x_1, x_2, x_3, x_4) = \overline{x_4 + x_1 + \bar{x}_2 + \bar{x}_3 + \bar{x}_1 + \bar{x}_2 + x_3}$, care se obține din precedenta, prin aplicarea legilor lui De Morgan. Folosirea porților de același fel este un avantaj.

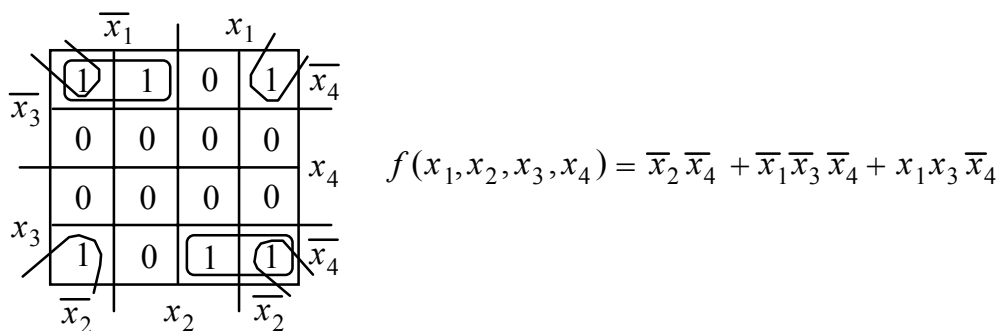


Fig. 4.6 Minimizarea funcției reprezentate pe diagramă

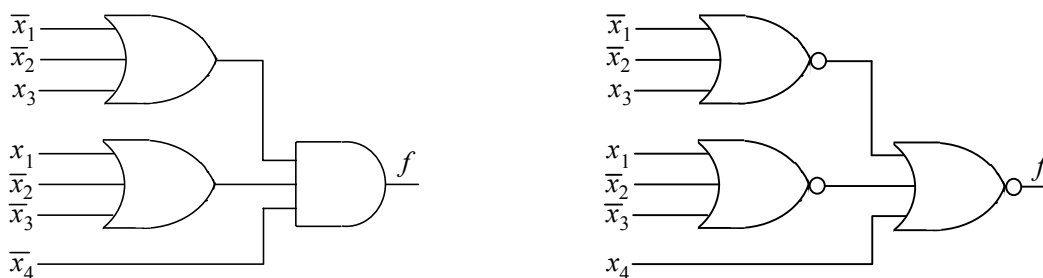


Fig. 4.7 Două soluții de implementare a funcției $f(x_1, x_2, x_3, x_4)$

	$\bar{x}_1$	x_1		
$\bar{x}_3$	0	X	1	0
x_3	0	0	1	X
	$\bar{x}_2$	x_2	$\bar{x}_2$	
	$\bar{x}_1$	x_1		
$\bar{x}_3$	0	X	1	0
x_3	0	0	1	X
	$\bar{x}_2$	x_2	$\bar{x}_2$	

Fig. 4.8 Minimizarea funcțiilor incomplet specificate

Funcțiile incomplet specificate sunt funcții binare ale căror valori logice sunt indiferente pentru o parte din combinațiile variabilelor de intrare. Astfel de situații apar atunci când suntem siguri că anumite configurații binare nu se aplică niciodată pe intrări, sau dacă se aplică, atunci valoarea respectivă a ieșirii nu este citită. Valoarea logică a funcției pentru aceste configurații binare nu contează, fiind notată cu simbolul X, numit "don't care".

"Libertatea dobândită prin existența unor ieșiri de tip *don't care* va trebui fructificată pentru a optimiza structura circuitului. La nivelul nostru de abordare, a optimiza înseamnă a minimiza" ([Ștefan, 2000]).

Exemplul 4.7

Să considerăm funcția reprezentată pe diagrama din figura 4.8. Orice simbol X din interiorul oricărui compartiment poate fi înlocuit fie cu valoarea logică 1, fie cu valoarea logică 0, după cum ne convine, astfel ca expresia algebrică a funcției să fie cât mai simplă. Dacă grupăm compartimentele care conțin 1 logic, atunci ne convine să facem grupările ca în figura din stânga, adică compartimentele coloanei din dreapta să conțină 1, iar toate celelalte să conțină 0 logic. Expresia funcției devine: $f(x_1, x_2, x_3, x_4) = x_1 \cdot x_4 + x_1 \cdot x_2 \cdot \bar{x}_3$. Dacă evaluăm complementul ne convine ca toate compartimentele să conțină valoarea logică 0 și se obține: $\bar{f}(x_1, x_2, x_3, x_4) = \bar{x}_1 + \bar{x}_2 + x_3 \cdot \bar{x}_4$, sau $f(x_1, x_2, x_3, x_4) = x_1 \cdot x_2 \cdot x_3 \cdot \bar{x}_4$.

Diagramele condensate, numite și **diagrame cu funcții incluse**, au fost concepute pentru minimizarea unor funcții cu număr mai mare de variabile. Compartimentele lor nu vor conține numai valorile logice 0, 1 sau X, ci și expresii boolene. O funcție de n variabile poate fi reprezentată pe o diagramă de $n-m$ variabile, având și mintermeni de m variabile.

Exemplul 4.8

Să considerăm funcția $f(x_1, x_2, x_3) = x_1 \cdot x_2 + x_1 \cdot x_3 + \bar{x}_1 \cdot \bar{x}_2 \cdot \bar{x}_3$, reprezentată pe diagrama din figura 4.9. Variabila x_3 se introduce în diagramă, observând că pentru $x_1 x_2 = 00$, funcția capătă valoarea logică complementară variabilei x_3 , deci $f = \bar{x}_3$ etc.

	$\bar{x}_1$	x_1		
$\bar{x}_3$	1	0	1	0
x_3	0	0	1	1
	$\bar{x}_2$	x_2	$\bar{x}_2$	

→

	$\bar{x}_1$	x_1	
$\bar{x}_3$	0	1	x_3
x_3	0	1	x_3
	$\bar{x}_2$	x_2	$\bar{x}_2$

Fig. 4.9 Condensarea diagramei Veitch - Karnaugh

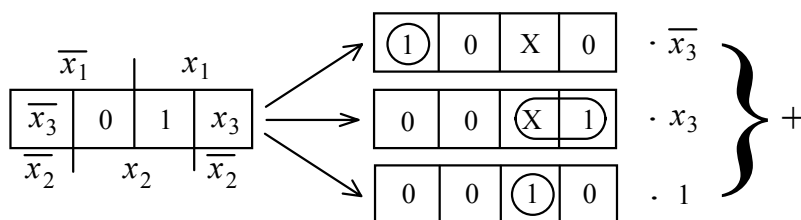


Fig. 4.10 Descompunerea diagramei condensate

Mai dificilă este descompunerea unei diagrame condensate. Desenul din figura 4.10 ilustrează modul în care se face descompunerea diagramei din exemplul precedent. Algoritmul folosit pentru această operație presupune parcurgerea următoarelor etape:

1. într-un compartiment care conține o variabilă sau o expresie booleană se trece 1, în celelalte compartimente similare se trece 0, compartimentele care conțin 0 își păstrează valoarea, iar cele care conțin 1 capătă valoarea X (don't care). Compartimentele care conțin simbolul X nu se modifică. Se minimizează funcția reprezentată în diagramă și variabila sau expresia booleană se aplică prin operatorul logic ȘI forme minime a funcției.
2. se repetă punctul 1 pentru toate compartimentele care conțin variabile sau expresii boolene.
3. în toate compartimentele care conțin variabile sau expresii boolene se trece 0, iar compartimentele care conțineau 0 sau 1 își păstrează valorile. Compartimentele care conțin simbolul X nu se modifică.
4. se aplică operatorul SAU expresiilor intermediare obținute la punctele 1, 2 și 3.

Exemplul 4.9

Desenul din figura 4.11 arată cum se face descompunerea diagramei condensate, pentru funcția de 7 variabile, reprezentată prin diagrama condensată de 4 variabile:

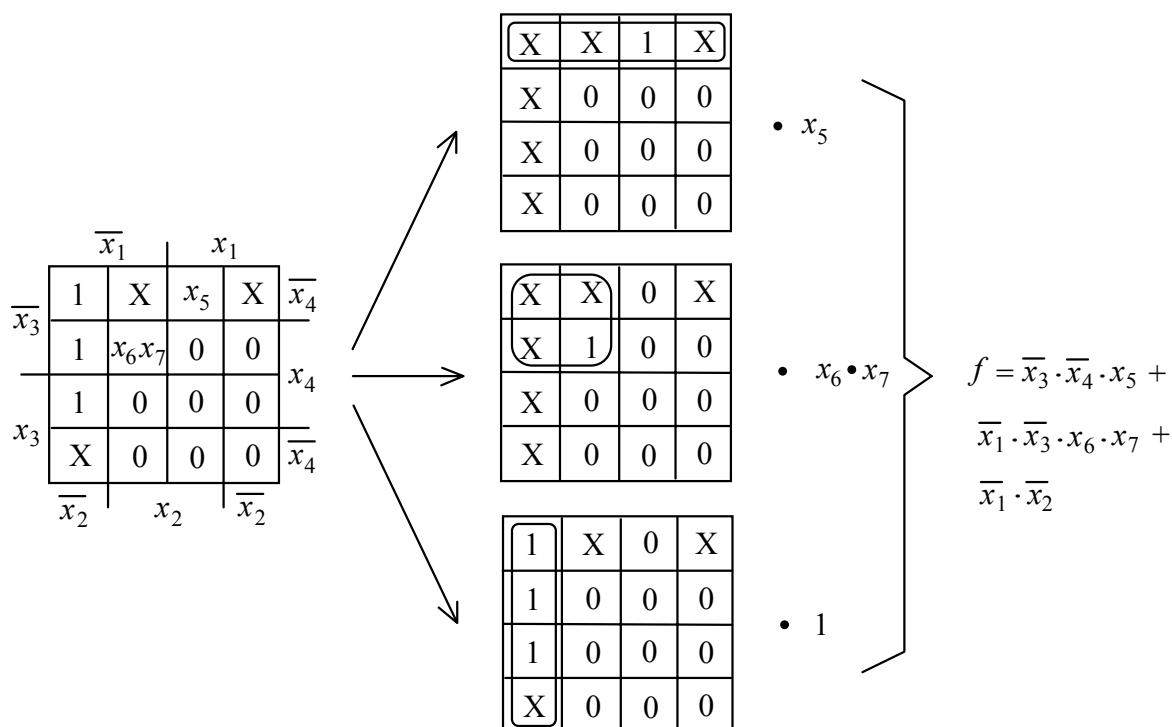


Fig. 4.11 Descompunerea unei diagrame condensate

Metoda Quine-McCluskey permite minimizarea unei funcții cu număr mai mare de variabile și are avantajul unei formalizări maxime, care permite o implementare ușoară a algoritmului de minimizare într-un program de calculator.

Vom introduce pentru început noțiunea de **pondere** a unui număr binar. Ponderea numărului binar x este un număr zecimal notat cu $w(x)$, care este egal cu numărul de biți de valoare 1 din reprezentarea binară a numărului x .

Exemplul 4.10

Ponderea numărului binar $x = 1011$ este $w(x) = 3$.

Se pornește de la forma canonică disjunctivă a funcției binare, iar algoritmul folosit pentru găsirea formei minime presupune parcurgerea următoarelor etape:

1. se face conversia din zecimal în binar a indicilor mintermenilor funcției binare
2. se împart numerele binare care reprezintă indicii mintermenilor în grupuri, după ponderea fiecărui indice, în sens crescător al ponderilor; grupurile se separă între ele prin linii orizontale și rezultă tabelul 1
3. se compară fiecare număr binar din grupul de pondere i cu fiecare număr binar din grupul de pondere $i + 1$, pentru toate grupurile. Două numere binare formează o nouă grupare dacă diferă printr-un singur bit de același rang. Gruparea nouă se trece în tabelul 2
4. de fiecare dată când ponderea i de la punctul 3 crește cu o unitate, se trage o linie orizontală în tabelul 2. După completarea tabelului 2, se repetă comparațiile de la punctul 3, pentru elementele din tabel și rezultă tabelul 3, și așa mai departe
5. când se ajunge la un tabel care conține un singur număr binar, atunci termenul corespunzător este un implicant prim al funcției. Ceilalți implicanți primi se găsesc căutând în toate tabelele construite termenii care nu au fost folosiți în comparații.

Exemplul 4.11

Să minimizăm funcția $f(x_1, x_2, x_3) = x_1 x_2 x_3 + x_1 \bar{x}_2 \bar{x}_3 + \bar{x}_1 x_2 x_3 + \bar{x}_1 x_2 \bar{x}_3 + \bar{x}_1 \bar{x}_2 x_3 + \bar{x}_1 \bar{x}_2 \bar{x}_3 = P_7 + P_4 + P_3 + P_2 + P_1 + P_0$ prin metoda Quine-McCluskey.

Prima coloană a tabelului 1 conține indicii zecimali ai mintermenilor funcției, grupați în ordinea crescătoare a ponderilor acestora, coloana din mijloc conține reprezentările binare ale indicilor, iar cea din dreapta expresiile algebrice ale mintermenilor.

I	$(I)_2$	P_I
0	✓✓✓ 000	$\bar{x}_1 \bar{x}_2 \bar{x}_3$
2	✓ 010 ✓	$\bar{x}_1 x_2 \bar{x}_3$
1	✓ 001 ✓	$\bar{x}_1 \bar{x}_2 x_3$
4	100 ✓	$x_1 \bar{x}_2 \bar{x}_3$
3	✓ 011 ✓✓	$\bar{x}_1 x_2 x_3$
7	111 ✓	$x_1 x_2 x_3$

Tabelul 1

Pentru construcția tabelului 2, observăm că termenul de indice 0 poate fi grupat cu termenul de indice 2, rezultând reprezentarea: $0-0$, o scriere echivalentă a relației: $\bar{x}_1 \bar{x}_2 \bar{x}_3 + \bar{x}_1 x_2 \bar{x}_3 = \bar{x}_1 \bar{x}_3$. Pentru a reține termenii care au fost comparați, termenul de indice 0 se marchează în partea stângă, iar cel de indice 2 în dreapta.

Grupările rezultate prin comparații se trec în tabelul 2. După completarea tabelului 2 se repetă comparațiile și rezultă tabelul 3, un tabel cu o singură linie, deci ne oprim. Forma minimă a funcției este disjuncția termenilor produs din toate tabelele construite, care nu au fost folosiți în comparații: $f(x_1, x_2, x_3) = \bar{x}_1 + \bar{x}_2 \bar{x}_3 + x_2 x_3$.

(0-2)	✓ 0-0	$\bar{x}_1 \bar{x}_3$			
(0-1)	✓ 00-	$\bar{x}_1 \bar{x}_2$	(0-2-1-3)	0--	$\bar{x}_1$
(0-4)	-00	$\bar{x}_2 \bar{x}_3$			
(2-3)	01- ✓	$\bar{x}_1 x_2$			
(1-3)	0-1 ✓	$\bar{x}_1 x_3$			
(3-7)	-11	$x_2 x_3$	<u>Tabelul 2</u>		<u>Tabelul 3</u>

Toți implicantii primi din expresia minimă a funcției f sunt esențiali și funcția are o formă minimă disjunctivă unică. Este posibil însă ca o funcție să aibă mai multe reprezentări minime.

Exemplul 4.12

Să minimizăm funcția $f(x_1, x_2, x_3, x_4) = P_{15} + P_{12} + P_{11} + P_9 + P_8 + P_7 + P_6 + P_4 + P_3 + P_0$. Putem simplifica tabelele construite dacă renunțăm la coloana care conține expresiile algebrice ale mintermenilor.

0	✓✓ 0000	(0-8)	✓ -000
8	✓✓ 1000 ✓	(0-4)	✓ 0-00
4	✓✓ 0100 ✓	(8-12)	1-00 ✓
12	1100 ✓✓	(8-9)	100-
9	✓ 1001 ✓	(4-12)	-100 ✓
6	✓ 0110 ✓	(4-6)	01-0
3	✓✓ 0110	(9-11)	10-1
11	✓ 1011 ✓✓	(6-7)	011-
7	✓ 0111 ✓✓	(3-11)	✓ -011
15	1111 ✓✓	(3-7)	✓ 0-11
		(11-15)	1-11 ✓
		(7-15)	-111 ✓

Tabelul 1

Tabelul 2

(0-8-12)	--00
(0-4-12)	
(3-11-15)	--11
(3-7-15)	

Tabelul 3

f se poate scrie ca disjuncția implicantilor primi:

$$f = \bar{x}_3 \bar{x}_4 + x_3 x_4 + x_1 \bar{x}_2 \bar{x}_3 + \bar{x}_1 x_2 \bar{x}_4 + x_1 \bar{x}_2 x_4 + \bar{x}_1 x_2 x_3$$

dar nu este în formă minimă

În tabelul 3 nu se mai pot face comparații, deci ne oprim cu construcția tabelelor. Căutăm în toate tabelele construite termenii care nu sunt marcați și care sunt implicantii primi ai funcției. Disjuncția lor nu este forma minimă a funcției, pentru că o parte din ei sunt neesențiali. Trebuie să găsim disjuncția numărului minim de implicantii primi care să ofere o acoperire completă a funcției, disjuncție numită și **sumă neredundantă**.

În acest scop se construiește un **tabel al implicantilor primi**, reprezentat în figura 4.12. Implicantii primi definesc liniile tabelului, iar mintermenii funcției definesc coloanele. Dacă un mintermen este inclus într-un implicant prim, atunci compartimentul de la intersecția coloanei și liniei respective se marchează. Pentru a determina nucleul funcției, se parcurg coloanele tabelului de la stânga la dreapta. Dacă există un singur marcaj pe coloană, atunci implicantul căruia îi corespunde marcajul este esențial.

Mintermeni	P ₁₅	P ₁₂	P ₁₁	P ₉	P ₈	P ₇	P ₆	P ₄	P ₃	P ₀
Implicanți primi	$x_1x_2x_3x_4$	$x_1x_2\bar{x}_3\bar{x}_4$	$x_1\bar{x}_2x_3x_4$	$x_1\bar{x}_2\bar{x}_3x_4$	$x_1\bar{x}_2\bar{x}_3\bar{x}_4$	$\bar{x}_1x_2x_3x_4$	$\bar{x}_1x_2x_3\bar{x}_4$	$\bar{x}_1x_2\bar{x}_3\bar{x}_4$	$\bar{x}_1\bar{x}_2x_3x_4$	$\bar{x}_1\bar{x}_2\bar{x}_3\bar{x}_4$
x_3x_4	- ✓ -	- - -	- ✓ -	- - -	- - -	- ✓ -	- - -	- - -	- ✓ -	- - -
$\bar{x}_3\bar{x}_4$	- - -	- ✓ -	- - -	- - -	- ✓ -	- - -	- - -	- ✓ -	- - -	- ✓ -
$\bar{x}_1x_2x_3$	- - -	- - -	- - -	- - -	- - -	✓	⊙	- - -	- - -	- - -
$x_1\bar{x}_2x_4$	- - -	- - -	✓	⊙	- - -	- - -	- - -	- - -	- - -	- - -
$\bar{x}_1x_2\bar{x}_4$	- - -	- - -	- - -	- - -	- - -	- - -	⊙	✓	- - -	- - -
$x_1\bar{x}_2\bar{x}_3$	- - -	- - -	- - -	⊙	✓	- - -	- - -	- - -	- - -	- - -

Fig. 4.12 Tabelul implicanților primi

Se poate observa aici că implicanții primi esențiali sunt x_3x_4 și $\bar{x}_3\bar{x}_4$ (primele două coloane, de exemplu). Fiecare linie din tabel care corespunde unui implicant prim esențial se "taie" cu o linie (liniile punctate orizontale), și fiecare coloană care conține un marcaj ce se află pe liniile "tăiate", se "taie" la rândul ei cu o linie verticală. Marcajele rămase "netăiate" (încercuite aici) corespund implicanților care formează mulțimea sumelor neredundante. O sumă neredundantă trebuie să conțină un singur implicant pentru fiecare coloană rămasă "netăiată". Rezultă că funcția din exemplul considerat are 4 forme minime echivalente:

$$\begin{aligned}
 f &= x_3x_4 + \bar{x}_3\bar{x}_4 + \bar{x}_1x_2x_3 + x_1\bar{x}_2x_4 = x_3x_4 + \bar{x}_3\bar{x}_4 + \bar{x}_1x_2x_3 + x_1\bar{x}_2\bar{x}_3 = \\
 &= x_3x_4 + \bar{x}_3\bar{x}_4 + x_1\bar{x}_2x_4 + \bar{x}_1x_2\bar{x}_4 = x_3x_4 + \bar{x}_3\bar{x}_4 + \bar{x}_1x_2\bar{x}_4 + x_1\bar{x}_2\bar{x}_3.
 \end{aligned}$$

Metoda consensurilor are unele avantaje față de metoda Quine-McCluskey deoarece permite reprezentarea funcției într-o formă elementară disjunctivă, nu neapărat canonică, și asigură reducerea dimensiunilor listelor de termeni prelucrați ([Pop, 1986]).

Consensul a doi termeni produs, care conțin aceeași variabilă, variabilă care este complementată într-unul dintre ei și necomplementată în celălalt, se obține prin înlăturarea variabilei respective și efectuarea produsului logic al celorlalte variabile rămase în cei doi termeni produs.

Exemplul 4.13

Consensul termenilor $x_1x_2x_3$ și $\bar{x}_3x_4x_5$ este $x_1x_2x_4x_5$, iar consensul termenilor $x_1x_2x_3$ și $\bar{x}_2x_3x_4$ este $x_1x_3x_4$.

Algoritmul folosit pentru găsirea formei minime prin metoda consensurilor presupune parcurgerea următoarelor etape:

1. se stabilesc perechile de termeni pentru care există consens și se adaugă consensurile termenilor la forma elementară disjunctivă a funcției
2. se elimină termenii care sunt acoperiți de alți termeni care se găsesc în expresia disjunctivă a funcției
3. se repetă punctele 1 și 2 până când nu se mai pot forma consensuri, sau toate consensurile care se pot forma sunt acoperite de termeni deja existenți.

Exemplul 4.14

Funcția de 4 variabile dată prin următoarea formă disjunctivă se poate transforma în felul următor: $f(x_1, x_2, x_3, x_4) = \bar{x}_1 x_2 x_3 x_4 + \bar{x}_1 \bar{x}_2 x_3 x_4 + \bar{x}_1 x_2 x_3 \bar{x}_4 + \bar{x}_1 \bar{x}_2 x_3 \bar{x}_4 + x_1 x_2 x_3 x_4 =$
 $= \bar{x}_1 x_3 x_4 + \bar{x}_1 x_2 x_3 x_4 + \bar{x}_1 \bar{x}_2 x_3 x_4 + \bar{x}_1 x_3 \bar{x}_4 + \bar{x}_1 x_2 x_3 \bar{x}_4 + \bar{x}_1 \bar{x}_2 x_3 \bar{x}_4 + x_1 x_2 x_3 x_4$. Aplicând de două ori legile de absorbție obținem: $f(x_1, x_2, x_3, x_4) = \bar{x}_1 x_3 x_4 + \bar{x}_1 x_3 \bar{x}_4 + x_1 x_2 x_3 x_4$. Se mai poate genera un consens și expresia funcției devine $f = \bar{x}_1 x_3 + \bar{x}_1 x_3 x_4 + \bar{x}_1 x_3 \bar{x}_4 + x_1 x_2 x_3 x_4$, sau după absorbție: $f = \bar{x}_1 x_3 + x_1 x_2 x_3 x_4 = x_2 x_3 x_4 + \bar{x}_1 x_3 + x_1 x_2 x_3 x_4 = x_2 x_3 x_4 + \bar{x}_1 x_3$. Aceasta este forma minimă a funcției date.

Metoda ESPRESSO este folosită mai ales în cazul funcțiilor cu număr foarte mare de variabile. Ea folosește o strategie euristică bazată pe aproximație, care evită construcția listelor de termeni și minimizează timpul de calcul ([Hill, 1993]).

"Dacă timpul în care este găsită o soluție optimă nu contează, atunci există întotdeauna o cale de a o obține. Dar, aproape întotdeauna, timpul contează. Degeaba știm să facem un lucru dacă nu ajungem la soluție în timp util. Acesta este motivul pentru care uneori ne mulțumim și cu soluții numai parțial optimizate" ([Ștefan, 2000]).

Metoda ESPRESSO se aplică repetitiv și generează de fiecare dată expresii boolene disjunctive mai simple decât precedentele lor. Fără a garanta forma minimă absolută, metoda oferă rezultate comparabile cu cele obținute prin metoda Quine-McCluskey. Algoritmul folosit pentru minimizarea funcției F presupune parcurgerea următoarelor etape:

1. se aplică procedura EXPAND funcțiilor F și $\bar{F}$, rezultatul fiind expresia redusă a funcției F , notată cu F^* .
2. se aplică procedura IRREDUNDANT expresiilor F^* și $\emptyset$, rezultând o nouă expresie redusă notată tot cu F^* .
3. Se aplică procedura ESSENTIAL_PRIMES expresiei F^* , iar rezultatul este expresia E .
4. Rezultatul procedurii anterioare se notează cu D și se elimină din expresia F^* , iar rezultatul este notat tot cu F^* .
5. Se aplică procedura REDUCE expresiei notate cu F^* și expresiei D , iar rezultatul este notat cu F^* .
6. Se aplică procedura EXPAND funcțiilor F^* și $\bar{F}$, iar rezultatul se notează cu F^* .
7. Se aplică procedura IRREDUNDANT expresiilor F^* și D , rezultând o nouă expresie redusă notată cu F^* .
8. Dacă expresia redusă F^* este mai simplă decât cea folosită la pasul 5, atunci se reia de la pasul 5; în caz contrar, se introduce o "perturbație" pentru a scoate procesul dintr-un eventual minim local. Dacă se reușește acest lucru, atunci se reia de la pasul 5; în caz contrar, forma minimă a funcției este $F = F^* + E$.

Exemplul 4.15

Considerăm funcția de 4 variabile $F = \bar{x}_1\bar{x}_2\bar{x}_3\bar{x}_4 + \bar{x}_1x_2\bar{x}_3 + x_1x_2x_4 + x_1x_3x_4$. Procedura EXPAND înlocuiește fiecare termen din expresia funcției F cu un set alcătuit din toți termenii distincți care formează o acoperire de rang imediat superior și care nu intersectează pe $\bar{F}$. Desenul din figura 4.13 exemplifică acțiunea procedurii asupra primului termen din expresia funcției. Sunt date și reprezentările echivalente în diagrame Veitch-Karnaugh, pentru a avea o imagine grafică a termenilor expandați.

Complementul $\bar{F} = (x_1 + x_2 + x_3 + x_4) \cdot (x_1 + \bar{x}_2 + x_3) \cdot (\bar{x}_1 + \bar{x}_2 + \bar{x}_4) \cdot (\bar{x}_1 + \bar{x}_3 + \bar{x}_4) = (x_1 + x_3 + \bar{x}_2x_4) \cdot (\bar{x}_1 + \bar{x}_4 + \bar{x}_2\bar{x}_3) = x_1\bar{x}_4 + x_1\bar{x}_2\bar{x}_3 + \bar{x}_1x_3 + x_3\bar{x}_4 + \bar{x}_1\bar{x}_2x_4 + \bar{x}_2\bar{x}_3x_4$.

Primul termen care formează o acoperire de rang imediat superior lui $\bar{x}_1\bar{x}_2\bar{x}_3\bar{x}_4$ este $\bar{x}_2\bar{x}_3\bar{x}_4$. Pentru a vedea dacă acest termen îl intersectează sau nu pe $\bar{F}$ facem un calcul algebric ușor: $\bar{x}_2\bar{x}_3\bar{x}_4 \cap \bar{F} = (\bar{x}_2\bar{x}_3\bar{x}_4) \cdot (x_1\bar{x}_4 + x_1\bar{x}_2\bar{x}_3 + \bar{x}_1x_3 + x_3\bar{x}_4 + \bar{x}_1\bar{x}_2x_4 + \bar{x}_2\bar{x}_3x_4) = x_1\bar{x}_2\bar{x}_3\bar{x}_4 + x_1\bar{x}_2\bar{x}_3\bar{x}_4 + 0 + 0 + 0 + 0 \neq 0$. Deci acest termen îl intersectează pe $\bar{F}$ și prin urmare este marcat pe desenul din figura 4.13 cu un asterisc. Procedăm la fel și cu ceilalți termeni și obținem expresia redusă a funcției F , notată cu F^* :

$$F^* = \bar{x}_1\bar{x}_3\bar{x}_4 + \bar{x}_1x_2\bar{x}_3 + x_1x_2x_4 + x_1x_3x_4.$$

Procedura IRREDUNDANT încearcă să găsească o sumă logică neredundantă, adică să elimine termenii redundanți din expresia disjunctivă a funcției F^* . Este formată din trei subproceduri care se aplică consecutiv: REDUNDANT, PARTIALLY_REDUNDANT și MINIMAL_COVER.

Subprocedura REDUNDANT separă toți implicantii primi care ar putea fi redundanți de cei care sunt **relativ esențiali**. Un implicant prim este relativ esențial față de o anumită expresie, dacă și numai dacă pentru acele valori ale variabilelor pentru care el are valoarea logică 1, expresia care nu conține implicantul respectiv este diferită de 1.

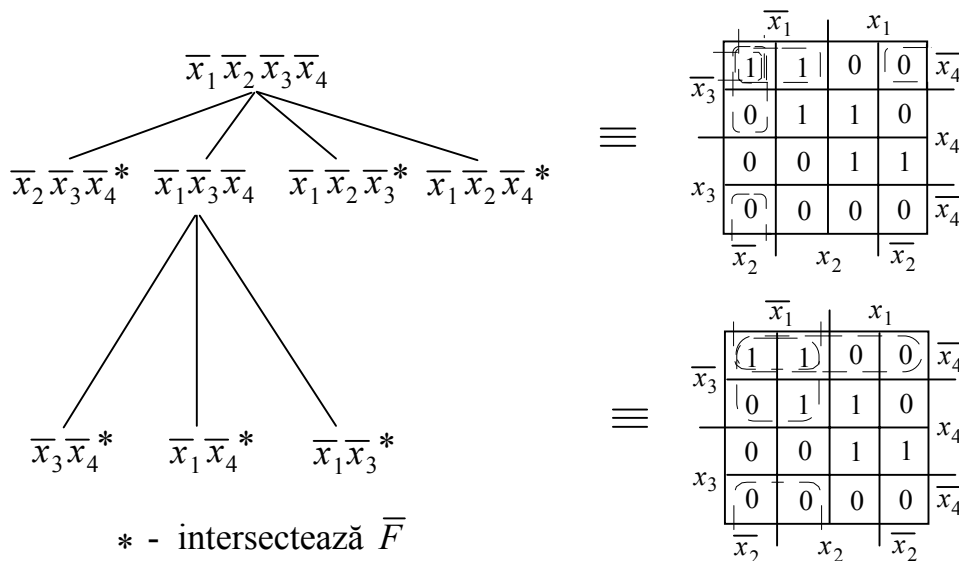


Fig. 4.13 Exemplificarea procedurii EXPAND

De exemplu, termenul $\bar{x}_1\bar{x}_3\bar{x}_4$ este relativ esențial pentru funcția F^* stabilită mai sus, deoarece pentru $x_1 = 0$, $x_3 = 0$ și $x_4 = 0$ termenul $\bar{x}_1\bar{x}_3\bar{x}_4 = 1$, iar expresia funcției din care se elimină termenul $\bar{x}_1\bar{x}_3\bar{x}_4$ devine: $1 \cdot x_2 \cdot 1 = x_2$, evident diferită de 1.

Se observă că toți implicantii din expresia lui F^* sunt relativ esențiali, deci expresia lui F^* nu se modifică: $F^* = \bar{x}_1\bar{x}_3\bar{x}_4 + \bar{x}_1x_2\bar{x}_3 + x_1x_2x_4 + x_1x_3x_4$.

Celelalte două subproceduri vor fi discutate la sfârșitul acestui exemplu, dar, în acest caz, nici ele nu modifică expresia lui F^* .

Procedura ESSENTIAL_PRIMES stabilește care sunt implicantii primi esențiali. În lipsa listei de mintermeni, un implicant prim este esențial dacă nu este acoperit în totalitate de orice alt set de implicantii primi ai funcției F , indiferent dacă aceștia sunt sau nu incluși în F^* . Acoperirea unui implicant de către alți implicantii primi poate fi găsită din consensul lui cu fiecare din termenii din F^* .

Pentru că nu întotdeauna se pot face consensuri, vom avea în vedere un **consens generalizat**. Consensul generalizat a doi termeni produs reprezintă o arie acoperită de fiecare dintre cei doi termeni. De exemplu, consensul lui $x_1x_2x_4$ cu $x_1x_3x_4$ este $x_1x_2x_3x_4$, adică mintermenul acoperit de fiecare dintre cei doi termeni.

Tabelul din figura 4.14 arată cum se face identificarea implicantilor primi esențiali folosind consensurile generalizate. Dacă suma consensurilor pe coloane acoperă termenul de pe linia respectivă, atunci acesta nu este esențial. Compartimentele marcate indică faptul că respectivele consensuri generalizate sunt identice cu termenii respectivi ai expresiei F^* și ele nu se iau în calculul sumelor consensurilor pe coloane. Pentru o mai bună înțelegere, se recomandă o vizualizare a implicantilor primi folosind echivalentul grafic al diagramei Veitch-Karnaugh.

Suma implicantilor primi esențiali este: $E = \bar{x}_1\bar{x}_3\bar{x}_4 + x_1x_3x_4$, iar noua expresie a lui F^* devine: $F^* = F^* - E = \bar{x}_1x_2\bar{x}_3 + x_1x_2x_4$.

	$\bar{x}_1\bar{x}_3\bar{x}_4$	$\bar{x}_1x_2\bar{x}_3$	$x_1x_2x_4$	$x_1x_3x_4$
$\bar{x}_1\bar{x}_3\bar{x}_4$	✓	$\bar{x}_1x_2\bar{x}_3\bar{x}_4$	0	0
$\bar{x}_1x_2\bar{x}_3$	$\bar{x}_1x_2\bar{x}_3\bar{x}_4$	✓	$x_2\bar{x}_3x_4$	0
$x_1x_2x_4$	0	$x_2\bar{x}_3x_4$	✓	$x_1x_2x_3x_4$
$x_1x_3x_4$	0	0	$x_1x_2x_3x_4$	✓
$\sum_j C^j$	$\bar{x}_1x_2\bar{x}_3\bar{x}_4$	$x_2\bar{x}_3(\bar{x}_1+x_4)$	$x_2x_4(x_1+\bar{x}_3)$	$x_1x_2x_3x_4$
$C^i \cap (\Sigma) = C^i$	Nu	Da	Da	Nu
Esențial	Da	Nu	Nu	Da

Fig. 4.14 Identificarea implicantilor esențiali prin consensuri

Procedura REDUCE încearcă să micșoreze acoperirea fiecărui implicant din F^* , fără a modifica funcția F^* . Pentru fiecare consens notat cu c^i , expresia $F^*(i) = (F^* - c^i) + D$ acoperă F^* prin alți implicanți. Procedura înlocuiește consensul c^i cu $c^i \cap \bar{F}^*(i)$, o porțiune din c^i care este acoperită numai de c^i . Pentru consensul $\bar{x}_1 x_2 \bar{x}_3$, vom scrie: $F^*(\bar{x}_1 x_2 \bar{x}_3) = (\bar{x}_1 x_2 \bar{x}_3 + x_1 x_2 x_4 - \bar{x}_1 x_2 \bar{x}_3) + \bar{x}_1 \bar{x}_3 \bar{x}_4 + x_1 x_3 x_4 = x_1 x_2 x_4 + \bar{x}_1 \bar{x}_3 \bar{x}_4 + x_1 x_3 x_4$. $\bar{F}^*(\bar{x}_1 x_2 \bar{x}_3) = x_3 \bar{x}_4 + \bar{x}_1 x_3 + \bar{x}_1 x_4 + x_1 \bar{x}_4 + \bar{x}_2 \bar{x}_3 x_4 + x_1 \bar{x}_2 \bar{x}_3$. $(\bar{x}_1 x_2 \bar{x}_3) \cap \bar{F}^*(\bar{x}_1 x_2 \bar{x}_3) = 0 + 0 + \bar{x}_1 x_2 \bar{x}_3 x_4 + 0 + 0 + 0 = \bar{x}_1 x_2 \bar{x}_3 x_4$. Se repetă calculele și pentru celălalt consens $x_1 x_2 x_4$ și obținem noua formă a expresiei F^* :

$$F^* = \bar{x}_1 x_2 \bar{x}_3 x_4 + x_1 x_2 \bar{x}_3 x_4$$

Procedura EXPAND folosește tot complementul lui F stabilit la început și se obține: $F^* = x_2 \bar{x}_3 x_4 + \bar{x}_1 x_2 \bar{x}_3 + x_1 x_2 x_4$.

Primele două subproceduri ale procedurii IRREDUNDANT folosesc funcția "don't care" D ca parametru. De data aceasta, la expresia lui F^* din subprocedura REDUNDANT se adaugă și expresia lui D . Observăm că nici unul din implicanții primi nu este relativ esențial, deci: $F^* = x_2 \bar{x}_3 x_4 + \bar{x}_1 x_2 \bar{x}_3 + x_1 x_2 x_4$.

Subprocedura PARTIALLY_REDUNDANT stabilește care sunt implicanții acoperiți de expresia $D + RE$ pentru a fi declarați **total redundanți** și eliminați. RE este suma implicantilor relativ esențiali. Implicanții din F^* care nu sunt nici relativ esențiali și nici total redundanți se numesc **parțial redundanți** și formează expresia R_p :

$$R_p = x_2 \bar{x}_3 x_4 + \bar{x}_1 x_2 \bar{x}_3 + x_1 x_2 x_4$$

Subprocedura MINIMAL_COVER caută un set minim de implicanți din R_p care acoperă expresia neacoperită de RE și setul D . Pentru fiecare implicant c^i din R_p va exista o sumă de implicanți G_{nc} , astfel încât expresia $(R_p - G_{nc}) + D + RE$ va acoperi implicantul c^i dacă și numai dacă un implicant este eliminat din G_{nc} . Rezultă că $F^* = x_2 \bar{x}_3 x_4$, și forma minimă a funcției este:

$$F = x_2 \bar{x}_3 x_4 + \bar{x}_1 \bar{x}_3 \bar{x}_4 + x_1 x_3 x_4$$

Verificați cu ajutorul altor metode de minimizare mai simple că aceasta este într-adevăr forma minimă a funcției de la care am pornit.

Probleme

- 4.1 Să se demonstreze prin cel puțin două metode următoarea echivalență booleană:

$$x \cdot y + y \cdot z + \bar{x} \cdot z = x \cdot y + \bar{x} \cdot z$$

- 4.2 Folosind axiomele și teoremele algebrei boolene, să se verifice următoarele echivalențe:

$$(x + y) \cdot (\bar{x} + \bar{y}) = \bar{x} \cdot y + x \cdot \bar{y}$$

$$x + \bar{x} \cdot y = x + y$$

$$(x + y) \cdot (y + z) \cdot (x + z) = x \cdot y + y \cdot z + x \cdot z$$

$$(\bar{x} + \bar{y}) \cdot (\bar{x} + y) \cdot (x + \bar{y}) = \bar{x} \cdot \bar{y}$$

$$\overline{x \cdot y + x \cdot z + y \cdot z} = \bar{x} \cdot \bar{y} + \bar{x} \cdot \bar{z} + \bar{y} \cdot \bar{z}$$

$$\bar{x} \cdot \bar{y} + y \cdot \bar{z} + x \cdot \bar{y} = \overline{y \cdot z}$$

$$(x + y) \cdot (\bar{x} + z) = x \cdot z + \bar{x} \cdot y$$

- 4.3 Se dă funcția de 3 variabile descrisă prin tabelul de adevăr din figura de mai jos. Să se minimizeze algebric și apoi folosind diagrama Veitch-Karnaugh și să se implementeze într-o variantă cu timp de propagare minim.

A	B	C	f	A	B	C	f
0	0	0	0	1	0	0	1
0	0	1	0	1	0	1	1
0	1	0	1	1	1	0	1
0	1	1	0	1	1	1	0

- 4.4 Să se implementeze funcția SAU-EXCLUSIV folosind numai porți ȘI-NU cu 2 intrări. Să se repete implementarea, folosind numai porți ȘI-SAU-NU. Comparați cele două soluții din punct de vedere al numărului de circuite integrate și al timpului de propagare.

- 4.5 O companie dorește să angajeze personal în diferite compartimente. Pentru a fi admiși, candidații trebuie să îndeplinească cel puțin una din următoarele condiții:

- să fie bărbat, necăsătorit, cu studii superioare;
- să fie necăsătorit(ă), cu studii și să aibă vârsta sub 30 ani;
- să fie femeie, necăsătorită și fără studii superioare;
- să fie bărbat, cu vârsta sub 30 ani;
- să fie necăsătorit(ă), cu vârsta peste 30 ani.

Să se proiecteze un circuit pentru selecția automată a candidaților.

- 4.6 Să se minimizeze următoarea funcție binară folosind metoda Quine-McCluskey și să se verifice prin metoda diagramelor Veitch-Karnaugh:

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

- 4.7 Folosind numai teorema consensurilor, inclusiv formularea duală, să se minimizeze următoarele expresii:

$$A \cdot \bar{B} \cdot \bar{C} + A \cdot \bar{B} \cdot D + \bar{B} \cdot C \cdot D + \bar{A} \cdot C \cdot D + A \cdot \bar{C} \cdot \bar{D}$$

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

([Wilkinson, 2002])

- 4.8 Se dă funcția: $f = P_1 + P_7 + P_8 + P_9 + P_{14}$, cu termenii redundanți P_3 , P_4 , P_{10} , P_{11} și P_{15} . Se cere:

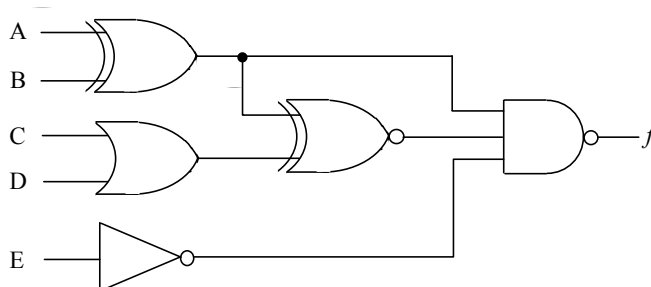
- minimizare cu ajutorul diagramei Veitch-Karnaugh;
- să se implementeze cu număr minim de porți ȘI-NU cu două intrări;
- să se implementeze cu număr minim de porți SAU-NU cu două intrări.

([Mureșan, 1996])

- 4.9 Arătați că o poartă AND cu n intrări poate fi înlocuită cu $n-1$ porți AND cu câte 2 intrări. Se poate afirma același lucru și despre porțile NAND?

([Wakerly, 1990])

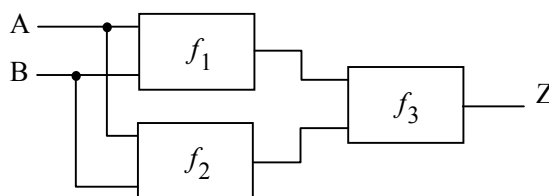
- 4.10 Să se înlocuiască circuitul din figură cu un circuit echivalent mai simplu:



([Wilkinson, 2002])

- 4.11 a) Dacă $f_1 = A + B$, f_3 este o poartă ȘI, iar $Z = A$ pentru circuitul din figura de mai jos, atunci este f_2 unic determinată?

- b) Dacă $Z = A$ atunci puteți găsi o funcție f_3 care depinde de ambele intrări și permite ca f_2 să fie unic determinată?



([Friedman, 1986])

- 4.12 Presupunem că transferul datelor într-un sistem numeric se face pe 4 biți, primii 3 fiind biți de date, iar ultimul, bit de paritate (indică dacă s-a transmis un număr par de biți de 1, de exemplu, pentru informația 101 rezultă cuvântul 1011). Proiectați un circuit NAND pe două nivele care semnalizează o eroare de 1 bit.

([Friedman, 1986])