UNIUNEA EUROPEANĂ

GUVERNUL ROMÂNIEI

Instrumente Structurale
2007-2013

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

Proiectarea Logică

17. Analiza circuitelor combinationale II

Analiza circuitelor combinaţionale II

În continuare se vor aborda principiile optimizării circuitelor logice combinaţionale modelate prin expresii formate din sume de produse (SDP) în două niveluri sau, echivalent, prin forme tabelare (FT) cum ar fi, spre exemplu, tabelele de implicanţi. Translatarea între cele două forme este imediată, în ambele sensuri. Dat fiind faptul că translatarea între cele doua forme este fără echivoc, pentru fixarea ideilor, se poate considera în continuare că orice circuit este prezentat sub prima formă, suma de produse.

Minimizarea exactă unei funcţii scalare Booleene are drept principal obiectiv, pentru respectiva funcţie, stabilirea unei expresii algebrice echivalente, prin sume de produse, minime ca număr de termeni şi având, eventual, un număr minim de literali.
Astfel de exprimare se mai numeşte, tradiţional, acoperire prin sumă de produse. Sunt utilizate, în acest scop, toate punctele din domeniul de definiţie în care funcţia este definită prin valoarea 1 şi punctele în care valoarea funcţiei nu este precizată, dacă astfel de puncte există în specificaţia funcţiei.
Găsirea unei forme în produs de sume este similară şi simplu de dedus din forma în sumă de produse.

Înainte să se considere aspecte atât teoretice cât şi de natură tehnologică, pragmatice, este util să se aibă în vedere un exemplu, de complexitate redusă dar care să contureze aspectele definitorii ale acestei probleme, esenţiale în proiectarea logică.

Exemplul 2.1. Pentru funcţia specificată prin suma de mintermi:
$f = m_5 + m_6 + m_9 + m_{10} + m_{13} + m_{14}$,
se doreşte stabilirea unor posibile implementări, cât mai simple
Numărul (minim) de variabile pentru această funcţie este patru, şi aceste variabile vor fi notate prin: x_8, x_4, x_2 şi x_1.

Exprimarea mintermilor în format binar este prezentată în tabelul 4:

Tabelul 2.1
Funcţia din exemplul 2.1

Mintermii funcţiei f	Variabilele funcţiei x_8	x_4	x_2	x_1
m_5	0	1	0	1
m_6	0	1	1	0
m_9	1	0	0	1
m_{10}	1	0	1	0
m_{13}	1	1	0	1
m_{14}	1	1	1	0

Translatarea literală, în ordine, a mintermilor funcţiei conduce la această expresie, în sume de produse canonice, a funcţiei considerate:

$$f = x_8'x_4x_2'x_1 + x_8'x_4x_2x_1' + x_8x_4'x_2'x_1 + x_8x_4'x_2x_1' + x_8x_4x_2'x_1 + x_8x_4x_2x_1', \quad (1)$$

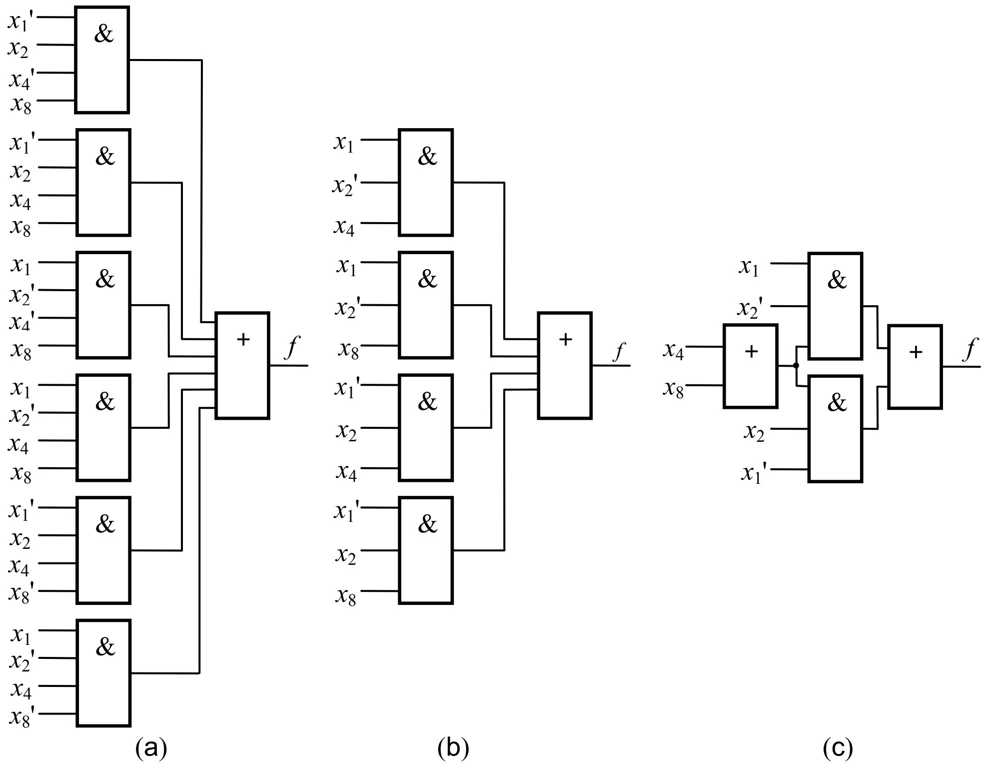

Figura 2.1. Realizări posibile ale funcţiei din exemplul 2.1.

Ţinând cont de comutativitatea operatorului + şi de faptul că în algebrele Boole-ene are loc identitatea $a + a = a$, expresia funcţiei f se poate scrie, după o prealabilă grupare convenabilă a termenilor produs, astfel:

$$f = (x_8'x_4x_2'x_1 + x_8x_4x_2'x_1) + (x_8x_4'x_2'x_1 + x_8x_4x_2'x_1) + (x_8'x_4x_2x_1' + x_8x_4x_2x_1') + (x_8x_4'x_2x_1' + x_8x_4x_2x_1'), \quad (2)$$

În expresia anterioară a funcţiei f, din fiecare paranteză se pot factoriza produse de câte trei variabile, astfel:

$$f = x_4x_2'x_1(x_8' + x_8) + x_8x_2'x_1(x_4 + x_4') + x_4x_2x_1'(x_8' + x_8) + x_8x_2x_1'(x_4' + x_4), \quad (3)$$

Considerând identităţile Boole-ene $a + a' = 1$ şi $a \cdot 1 = a$, expresia funcţiei f se rescrie astfel:

$$f = x_4x_2'x_1 + x_8x_2'x_1 + x_4x_2x_1' + x_8x_2x_1'. \quad (4)$$

Se poate remarca, în expresia anterioară a funcţiei f, următoarea factorizare:

$$f = (x_4 + x_8)x_2'x_1 + (x_4 + x_8)x_2x_1'. \quad (5)$$

Expresia iniţială a funcţiei, expresia (1), forma minimizată (4) şi forma factorizată (5) sunt reprezentate schematic, simbolic, în figura 2 (a), (b) şi respectiv (c).

Cercetând circuitele din figura 2.1 se poate observa cu uşurinţă că dintre cele trei realizări ale funcţiei f, cea mai complexă este cea din figura 2.1.(a).
Comparând, în continuare, circuitele 2.1.(b) şi 2.1.(c) se poate spune că în cazul circuitului 2.1(c) sunt utilizate mai puţine componente dar, în acelaşi timp se observă că, spre deosebire de celelalte două circuite are mai multe niveluri. Un număr crescut de niveluri poate implica, depinzând de tehnologie, o viteză de lucru mai scăzută pentru un circuit combinaţional.

Alegerea între un circuit mai economicos, dar mai lent, şi unul mai costisitor, dar mai rapid, este tranşată prin considerente care ţin seama de contextul în care este plasat circuitul aflat în discuţie. Este de reţinut că toate considerentele ce vor fi făcute în continuare se vor referi exclusiv la circuitele cu două niveluri.

De remarcat faptul că în cazul circuitelor combinaţionale cu două niveluri, aşa cum se poate vedea în figura 2.1(b), fiecare produs din suma de produse reprezintă o poartă (ŞI) iar fiecare literal dintr-un produs reprezintă o intrare într-o poartă. Întreaga sumă de produse corespunde unei porţi (SAU). Similar se întâmplă în cazul implementării circuitelor combinaţionale prin produse de sume: fiecare sumă reprezintă o poartă (SAU) iar întregul circuit este realizat printr-o poartă (ŞI). Ca şi în cazul sumelor de produse, un literal dintr-o sumă este o linie de intrare într-o poartă SAU. Trebuie remarcat, pe scurt, că numărul de linii de intrare într-o poartă este, în general, limitat iar numărul respectiv depinde de tehnologia în care este construită poarta în cauză.

Porţile au costuri după cum şi liniile de intrare în porţi au costurile lor. Raportul dintre costul unei linii de intrare într-o poartă şi costul acelei porţi este dependent de tipul porţii, de tehnologia în care se realizează poarta etc. Se poate aprecia, în general, că atunci când se pune problema unei porţi suplimentare, costul acesteia este de câteva ori mai mare decât costul unei linii de intrare într-o poartă deja existentă în circuitul respectiv. Din acest punct de vedere, micşorarea numărului de porţi este un criteriu important în găsirea, în general, a unei forme mai simple pentru o funcție Boole-eană dată.

2.1 Principiile optimizării logice.

Obiectivul minimizării logice în două niveluri este reducerea mărimii reprezentării funcţiilor Booleene în oricare din formele de reprezentare *sume de produse* ori *produse de sume*.
Se poate remarca faptul că oricare din cele două forme ale funcţiilor Booleene poate fi dedusă din cealaltă cu ajutorul legilor De Morgan iar această transformare păstrează numărul de termeni şi de literali. Prin urmare, se poate concentra abordarea doar asupra minimizării unei singure forme şi anume suma de produse, fără să se piardă din generalitate.

Obiectivul detaliat al minimizării logice în două niveluri poate varia puţin în funcţie de stilurile de implementare. Circuitele PLA sunt un astfel de stil de implementare. Prima ţintă a minimizării logice, pentru acest stil de implementare este reducerea numărului de produse şi ţinta secundară este reducerea literalilor din produse.

Alte obiective de optimizare sunt relevante atunci când funcţiile modelate prin forme în două niveluri sunt implementate altfel decât prin PLA-uri. Reprezentările logice în doua niveluri pentru funcţii scalare, spre exemplu, pot fi implementate prin porţi complexe a căror mărime este corelată cu numărul de literali din forma factorizată a acelei reprezentări. În astfel de situaţii obiectivul major este minimizarea numărului de literali.

Minimizarea logică pentru o funcţie scalară sau o funcţie vectorială se face după aceleaşi principii, dar cazul vectorial este mult mai complex. Minimizarea disjunctă a componentelor scalare ale unei funcţii vectoriale poate conduce la rezultate suboptimale deoarece optimizarea nu poate exploata comunitatea unor termeni produs. Un rezultat important în optimizarea logică în două niveluri este echivalenţa funcţiilor vectoriale binare de variabile binare cu funcţiile binare scalare de variabile multi-valorice. Din astfel de raţiuni, pentru început, abordarea se va concentra asupra tehnicilor de optimizare ale funcţiilor scalare de variabile bi-valorice.

2.2 Definiţii

Se vor considera funcţii binare (Boole-ene) de variabile binare incomplet specificate, de forma:

$$\boldsymbol{f}\colon \mathbf{B}^n \rightarrow \{0, 1, *\}^m,$$

deoarece funcţiile complet specificate sunt un caz particular al funcţiilor cu puncte nedefinite, nespecificate ("*don't care*").

Pentru fiecare linie de ieşire f_i $(1 \leq i \leq m)$ se definesc trei mulţimi de puncte disjuncte, partiţii ale domeniului de definiţie:

- mulţimea punctelor în care funcţia f_i are valoarea 1,
- mulţimea punctelor în care funcţia f_i are valoarea 0 şi
- mulţimea punctelor în care funcţia f_i are valoarea * sau mulţimea punctelor nedefinite.

Aceste partiţii sunt notate respectiv prin *mp*1, *mp*0, respectiv *mp** (submulţimi ale domeniului de definiţie $\mathbf{B}^n$).
Prin această partiţionare a domeniului de definiţie, fiecare funcţie scalară incomplet specificată poate fi privită ca un triplet de funcţii complet specificate.

Funcţiile sunt reprezentate (prin sume de produse ori tabelar) ca liste de implicanţi. Conceptul implicantului *multi-ieşire* (*multi-funcţie*) este general ca natură, deoarece combină vectorul valorilor liniilor de intrare cu vectorul valorilor corespunzătoare liniilor de ieşire.
În consideraţiile care urmează se va restricţiona noţiunea de implicant doar la valoarea 1 sau * (neprecizată) a unei funcţii.

În consecinţă, partea de ieşire a unui implicant multi-ieşire este bivalorică pentru o componentă scalară a respectivei funcţii vectoriale, având următoarea semnificaţie:
- o valoare 1 implică fie o valoare 1, fie o valoare * a componentei scalare respective, în timp ce,
- o valoare 0 nu implică atribuirea unei valori a respectivei componente scalare; pur şi simplu acea componentă scalară este ignorată în punctul corespunzător părţii de intrare.

În astfel de situaţii partea de intrare a implicantului, cu alte cuvinte, se referă la unul sau mai multe puncte din domeniul de definiţie, unde anumite componente scalare ale funcţiei vectoriale iau valoarea 1 ori * şi anume acele componente pentru care partea de ieşire a implicantului multi-valoric este nenulă.

> <u>Definiţia 2.1</u> Un *implicant multi-ieşire* (*multi-funcţie*) al unei funcţii vectoriale $\boldsymbol{f}: \mathbf{B}^n \rightarrow \{0, 1, *\}^m$
> este o pereche de vectori linie de dimensiune n şi m numiţi *partea de intrare* şi respectiv *partea de ieşire*.

Partea de intrare are componente cu valori din mulţimea {0, 1, *} şi reprezintă un produs de literali. Partea de ieşire are componente cu valori din mulţimea {0, 1}. Pentru fiecare componenta a ieşirii o valoare nenulă implică o valoare 1 sau * a respectivei funcţii scalare în punctul (punctele) corespunzător (corespunzătoare) părţii de intrare.

Mintermii multi-ieşire sunt implicanţi supuşi unor restricţii deosebite.

> <u>Definiţia 2.2</u> Un *minterm multi-ieşire* al unei funcţii precum cea definită anterior este un implicant multi-ieşire a cărui parte de intrare are componente cu valori din mulţimea {0, 1} (exact n literali) şi care implică valoare nenulă pentru una şi numai pentru una dintre componentele (funcţiile scalare) de ieşire.

Exemplul care urmează ilustrează printr-o funcţie vectorială simplă, cu două componente scalare, aspectele definitorii ale implicanţilor canonici, mintermii, în cazul funcţiilor vectoriale.

<u>Exemplul 2.2.</u> $\boldsymbol{f} = (f_1, f_2)$;
$f_1 = a'b'c' + a'b'c + ab'c + abc + abc'$,

$f_2 = a'b'c + ab'c$

Un implicant multi-ieşire al acestei funcţii vectoriale ***f*** este (*01|11) .

Acest implicant multi-ieşire reprezintă patru mintermi multi-ieşire:

(001|10), (101|10), (001|01) şi (101|01).

Datorită faptului că un implicant, necanonic, poate acoperi mai mulţi implicanţi canonici (mintermi) multi-ieşire, se poate concepe reprezentarea unei funcţii printr-o mulţime de implicanţi necanonici. Iar numărul implicanţilor necanonici ai respectivei mulţimi să fie, posibil, mai mic decât numărul implicanţilor canonici ai funcţiei. O astfel de mulţime cu implicanţi necanonici, asociată unei funcţii date, este definită în continuare.

Definiţia 2.3 O *acoperire* a unei funcţii vectoriale binare este o mulţime (o listă) de implicanţi care conţin (acoperă) mintermii acelei funcţii.

Se notează prin ***F*** acoperirea funcţiei ***f***. Mulţimile *mp*1, *mp*0, *mp** pot fi modelate prin acoperiri, unde implicanţii şi mintermii sunt corespunzători funcţiilor complet specificate respective. Acoperirile *mp*1, *mp*0 şi *mp** sunt notate, tradiţional, prin F^{ON}, F^{OFF} şi respectiv F^{DC}. O acoperire ***F*** a unei funcţii ***f*** satisface inegalitatea: $\boldsymbol{F}^{ON} \subseteq \boldsymbol{F} \subseteq \boldsymbol{F}^{ON} \cup \boldsymbol{F}^{DC}$. Atunci când o funcţie este complet definită, $\boldsymbol{F}^{ON}$ şi ***F*** sunt identice (deoarece $\boldsymbol{F}^{DC}$ este mulţimea vidă).

Mărimea sau cardinalitatea unei acoperiri este numărul de implicanţi ai acelei acoperiri.

Definiţia 2.4. O *acoperire minimă* este o acoperire de cardinalitate minimă.

În cele ce urmează se va considera că obiectivul minimizării logice combinaţionale exacte în două niveluri este *determinarea unei acoperiri minime*.
Este, adesea, util să se determine acoperiri minimale, locale, numite *acoperiri minimale*, deoarece calculul acestora poate fi atins cu resurse (memorie şi timp) mai reduse.
Obiectivul minimizării euristice logice combinaţionale în două niveluri este determinarea unei acoperiri minimale. Astfel de acoperiri sunt adesea foarte apropiate în cardinalitate de acoperirile minime şi pot astfel oferi soluţii eficiente pentru problemele practice. În mod normal minimalitatea locală se defineşte în termeni de conţinere, includere.

Definiţia 2.5 O acoperire *iredundantă* a unei funcţii este o acoperire care nu este un superset propriu al niciunei alte acoperiri pentru aceeaşi funcţie.

Altfel spus, îndepărtarea oricărui implicant dintr-o acoperire minimă nu mai acoperă, nu mai conţine, funcţia. Echivalent, nici un implicant nu este conţinut în orice subset de implicanţi ai acoperirii. O proprietate de minimalitate mai slabă este minimalitatea în raport cu conţinerea unui singur implicant, numită adeseori, conţinerea singulară.

Definiţia 2.6 O acoperire este *minimală* în *raport cu conţinerea singulară* dacă nici un implicant nu este conţinut în orice alt implicant al acoperirii.

O acoperire iredundantă este deasemenea minimală în raport cu conţinerea singulară, dar nu şi reciproc. Raţiunea acestei afirmaţii este aceea că un implicant poate fi redundant deoarece este conţinut într-un subset de implicanţi ai acoperirii şi nu de un singur implicant. Din acest motiv, redundanţa este o proprietate mai puternică.

Exemplul 2.3. Se reia funcţia ***f*** = (f_1, f_2) din exemplul 2.2.
O acoperire minimală de cardinalitate trei este specificată prin:

(00*|10)

(*01|11)
(11*|10)

O acoperire iredundantă de cardinalitate patru este definită prin:

(00*|10)
(*01|01)
(1*1|10)
(11*|10)

O acoperire redundantă de cardinalitate cinci, dar care este minimală în raport cu conţinerea singulară este exemplificată prin:

(00*|10)
(*01|01)
→ (*01|10)
(1*1|10)
(11*|10)

De reţinut că cel de-al treilea implicant este conţinut în reuniunea dintre primul şi cel de-al patrulea implicant (cel indicat).

O altă proprietate a implicanţilor şi a acoperirilor este aceea de a fi primi, respectiv prime.

Definiţia 2.7. Un implicant este *prim* dacă nu este conţinut de nici un alt implicant al funcţiei. O acoperire este *primă* dacă toţi implicanţii săi sunt primi.

În raport cu definiţia 2.7 se cuvin făcute câteva remarci complementare:
- definiţia calităţii de implicant prim a unui implicant este legată de toţi implicanţii posibili ai funcţiei şi nu doar de cei ai acoperirii considerate;
- pentru funcţiile scalare, un implicant prim corespunde unui produs de literali, unde nici un literal nu poate fi eliminat fără să se piardă proprietatea de implicant.

În termeni generici, un implicant prim corespunde unui implicant cu dimensiune maximală. Dimensiunea maximală înseamnă cea mai mare dimensiune ce poate fi atinsă fără să se intersecteze $\boldsymbol{F}^{OFF}$ sau, echivalent, *mp*0.

Pentru funcţiile vectoriale, un implicant prim conţine şi un număr maxim de componente scalare (funcţii scalare) ale funcţiei vectoriale reprezentate în partea de ieşire a implicantului. Implicanţii primi sunt adesea numiţi primi, pentru o exprimare mai scurtă, mai simplă, fără să existe riscul de echivoc.

Exemplul 2.4. Se reia funcţia vectorială de variabile binare din exemplele precedente. Lista tuturor implicanţilor primi arată astfel:

(00*|10)
(*01|11)
(1*1|10)
(11*|10)

Prin definiţie nici un implicant prim nu este conţinut în alt implicant prim al funcţiei. Implicanţii anteriori reprezintă o acoperire primă, dar care nu este minimă. In adevăr, al treilea implicant prim este conţinut în reuniunea dintre al doilea şi al patrulea implicant prim. Prin definiţia unui implicant prim, dacă acoperirea este primă, atunci aceasta este minimală în raport cu conţinerea singulară.
Înlăturând cel de-al treilea implicant prim se obţine o acoperire minimă care este necesar iredundantă.

Se presupune, acum, că funcţia este incomplet specificată şi mulţimea *mp** (sau F^{DC}) pentru ambele ieşiri este specificată, simplu, prin implicantul cu partea de intrare 100.

Atunci, implicanţii primi ar fi:

(*0*|10),
(*01|11),
(1**|10).

Aceşti implicanţi primi formează o acoperire minimă.

Anumiţi implicanţi primi au proprietatea specială că trebuie să fie incluşi în orice acoperire a funcţiei. Aceşti implicanţi sunt numiţi implicanţi primi esenţiali.

<u>Definiția 2.8.</u> Un implicant prim este *esenţial* dacă există un minterm al funcţiei acoperit (conţinut) în exclusivitate (faţă de mulţimea tuturor implicanţilor primi ai funcţiei) de acest implicant prim.

Exemplul următor abordează câteva proprietăţi importante ale implicanţilor primi.

<u>Exemplul 2.5.</u> Funcţia considerată în exemplele anterioare are asociaţi aceşti implicanţi primi:

(00*|10)
(*01|11)
(1*1|10)
(11*|10)

Pentru prima componentă scalară se observă că primul implicant prim este esenţial deoarece acoperă în exclusivitate mintermul (000|10), iar al patrulea implicant prim este deasemenea esenţial pentru că acoperă în exclusivitate mintermul (110 |10).

A doua funcţie scalară este implicată de numai un singur implicant, mai exact de al doilea implicant. Acest implicant este, în consecinţă, esenţial.
Dintre toți implicanţii primi, doar al treilea nu este esenţial.