

## Cursul nr. 4

## C4.1. Compatibilitatea între familiile de CI logice

Interconectarea CI aparținând diferitelor familii logice pentru obținerea unui sistem presupune compatibilitatea nivelurilor de tensiune la intrare și la ieșire. Compatibilitatea nivelurilor de tensiune se realizează cu ajutorul circuitelor de interfață, care fac o deplasare de nivel.

## Interfața TTL-CMOS

Când circuitele TTL trebuie să comande circuite CMOS alimentate dintr-o singură tensiune, nivelul minim în starea 1 logic pentru TTL (2.4V) este mai mic decât nivelul minim în starea 1 logic pentru CMOS (3.5V), după cum se observă pe fig. C4.1.

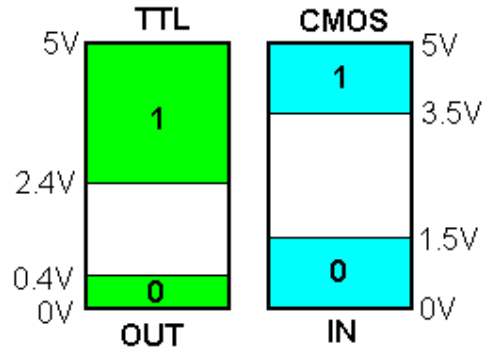


Fig. C4.1 Nivelele logice TTL și CMOS

Pentru a se crește nivelul semnalului de la ieșirea TTL în starea 1 logic, se montează o rezistență  $R_x$  (fig. C4.2a). În fig. C4.2b se prezintă modul de conectare a rezistenței  $R_x$  în cazul alimentării CMOS cu tensiune mai mare de 5V.

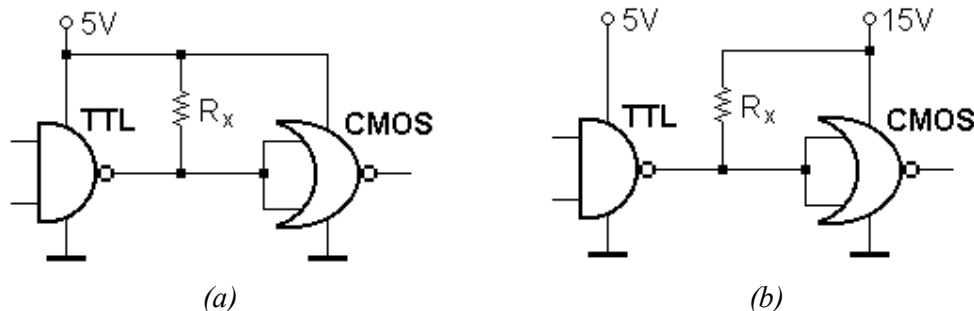


Fig. C4.2 Interfața TTL-CMOS: (a) alimentare comună de 5V;  
(b) alimentarea CMOS cu tensiune mai mare de 5V

Valoarea rezistenței  $R_x$  depinde de tipul subfamiliei TTL, conform tabelului C4.1:

Tabelul C4.1

$R_x$	Subfamilia TTL			
	Standard	Rapidă	Low-power	Shottky
$R_{x\min}$ [ $\Omega$ ]	390	270	1500	820
$R_{x\max}$ [ $k\Omega$ ]	4.7	4.7	27	12

## Interfața CMOS-TTL

CMOS în starea 0 la ieșire trebuie să asigure curentul de intrare al porții TTL în starea 0. Cuplarea directă CMOS-TTL se poate face doar cu subfamilia TTL Low-Power Shottky. Pentru celelalte tipuri de TTL se folosesc circuite separatoare CMOS (buffer) care, în starea 0 au o capacitate de curent corespunzătoare la mai multe intrări TTL. Circuitele separatoare pot fi inversoare sau neinversoare și pot absorbi la ieșirea lor în starea 0 minim 3.2mA, ceea ce corespunde la 2 intrări TTL (o intrare TTL în starea 0 injectează un curent de maxim 1.6mA).

### Interfața CMOS-ECL

Interfața CMOS-ECL se realizează cu CI speciale, cu rol de conversie ECL-CMOS și CMOS-ECL.

Când se cere ca sistemul CMOS să funcționeze la o viteză mai mare, se poate utiliza circuitul din fig. C4.3. Dioda  $D$  fixează ieșirea circuitului în starea  $0$  la valoarea sursei  $V_{EE}$  ( $-5.2V$ ).

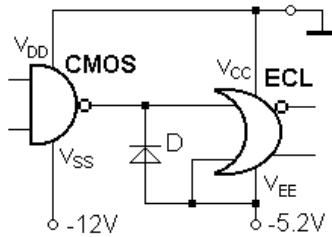


Fig. C4.3 Interfața CMOS-ECL

### Interfața TTL-I<sup>2</sup>L

Interfața TTL-I<sup>2</sup>L reprezentată în fig. C4.4 este un simplu inversor cu colectorul în gol.

Interfața lucrează astfel:

- Pentru  $0$  logic la ieșirea TTL, tranzistorul  $T_i$  din interfață se blochează. Prin intermediul rezistenței exterioare  $R_{i,ext}$  se asigură  $0.7V$  în baza lui  $T_1$  din circuitul I<sup>2</sup>L. La ieșirea I<sup>2</sup>L se obține  $0$  logic;
- Pentru  $1$  logic la ieșirea TTL, tranzistorul  $T_i$  din interfață trece în conducție, având  $U_{CE,sat} < 0.7V$ .  $T_1$  din circuitul I<sup>2</sup>L se blochează iar curentul  $I$  circulează prin tranzistorul  $T_i$  din interfață. La ieșirea I<sup>2</sup>L se obține  $1$  logic.

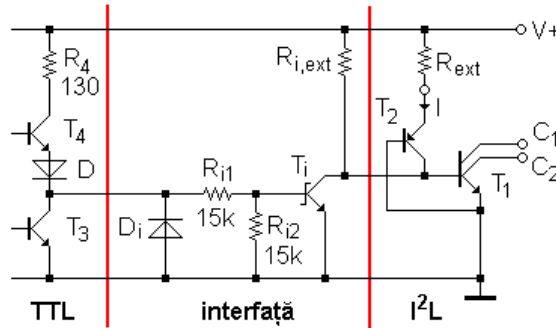


Fig. C4.4 Interfața TTL-I<sup>2</sup>L

## C4.2. Circuite logice combinaționale

### C4.2.1 Analiza și sinteza circuitelor logice combinaționale

#### A. Rețele ȘI-SAU și rețele ȘI-NU

O rețea ȘI-SAU utilizează porți ȘI pentru transpunerea produselor (operațiilor ȘI) și porți SAU pentru transpunerea sumelor (operațiilor SAU) la implementarea unei funcții booleene dată în formă normală disjunctivă (sume de produse).

**EXEMPLUL 4.1.** Funcția  $f(A, B, C, D) = A\bar{C} + BCD$  se implementează direct într-o rețea ȘI-SAU ca în figura 4.1a.

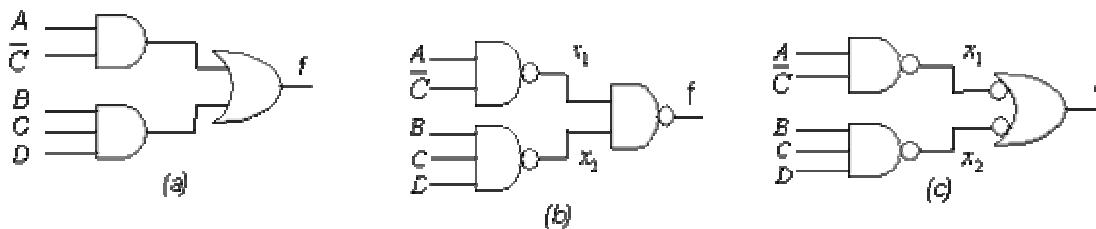


Fig. 4.1 Implementarea funcției  $f(A, B, C, D) = A\bar{C} + BCD$ : (a) rețea ȘI-SAU; (b) rețea ȘI-NU; (c) rețea ȘI-NU preferată.

Aplicând principiul dublei negații și a legilor lui DeMorgan din algebra booleană putem transforma expresia funcției dată în exemplul 4.1. astfel încât ea să poată fi implementată numai cu porți ȘI-NU (rețea ȘI-NU) ca în figura 4.1b:

$$f(A, B, C, D) = \overline{\overline{AC} + \overline{BCD}} = \overline{\overline{AC} \overline{BCD}} = \overline{x_1 x_2} \text{ unde } x_1 = \overline{AC} \text{ și } x_2 = \overline{BCD}.$$

Transformând în continuare expresia funcției cu ajutorul legilor lui DeMorgan se obține  $\overline{x_1 x_2} = \overline{x_1} + \overline{x_2}$  formă ce este implementată în figura 4.1c într-o rețea ȘI-NU ce utilizează la ieșire o poartă ȘI-NU echivalentă (poartă SAU cu intrări negate). Schema din figura 4.1c este preferată celei din figura 4.1b deoarece ea seamănă cu cea din figura 4.1a, negațiile de la ieșirile porților de la intrare compensându-se reciproc cu negațiile intrărilor porții de ieșire.

## B. Rețele SAU-ȘI și rețelele SAU-NU

O rețea SAU-ȘI utilizează porți SAU pentru transpunerea sumelor (operațiilor SAU) și porți ȘI pentru transpunerea produselor (operațiilor ȘI) la implementarea unei funcții booleene dată în formă normală conjunctivă (produse de sume).

**EXEMPLUL 4.2.** Funcția  $f(A, B, C, D) = (B + C + D)(\overline{A} + D)$  se implementează direct într-o rețea SAU-ȘI ca în figura 4.2.a

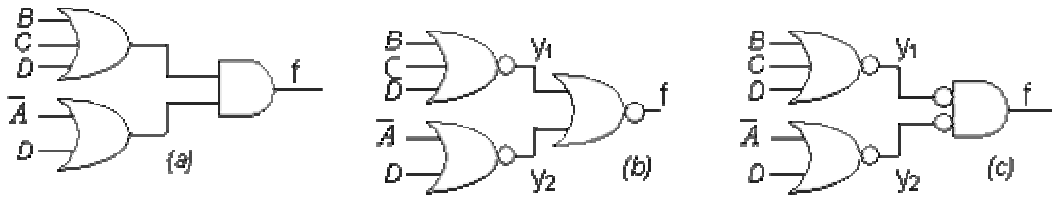


Fig. 4.2 Implementarea funcției  $f(A, B, C, D) = (B + C + D)(\overline{A} + D)$ : (a) rețea SAU-ȘI; (b) rețea SAU-NU; (c) rețea SAU-NU preferată.

Utilizând și pentru această funcție legile algebrei booleene putem să-i schimbăm forma astfel încât să poată fi implementată cu o rețea SAU-NU (fig. 4.2b):

$$f(A, B, C, D) = (B + C + D)(\overline{A} + D) = \overline{\overline{B + C + D} \overline{\overline{A} + D}} = \overline{y_1 + y_2},$$

unde  $y_1 = \overline{B + C + D}$  și  $y_2 = \overline{\overline{A} + D}$ .

Cu ajutorul legilor lui DeMorgan se poate schimba în continuare forma expresiei funcției astfel încât circuitul obținut, rețeaua SAU-NU din figura 4.2c în care se utilizează la ieșire o poartă SAU-NU echivalentă, să semene cu rețeaua SAU-ȘI din figura 4.2a. Astfel  $\overline{y_1 + y_2} = \overline{y_1} \overline{y_2}$

## C. Circuite pe două sau mai multe nivele

Circuitele asemănătoare celor din figurile 4.1 și 4.2 sunt circuite pe două nivele. În figura 4.3a este reprezentat circuitul din figura 4.1c. la care s-au pus în evidență cele două nivele. La primul nivel se află poarta ce produce semnalul de ieșire al circuitului iar la nivelul doi porțile ce recepționează semnalele de intrare în circuit. Dacă circuitul folosește intrările negate iar acestea nu sunt disponibile, se folosește un circuit pe trei nivele, nivelul trei conținând în acest caz numai porți NU ca în figura 4.3b. În general, un circuit are  $n$  nivele dacă cel puțin un semnal străbate de la intrare până la ieșire  $n$  porți.

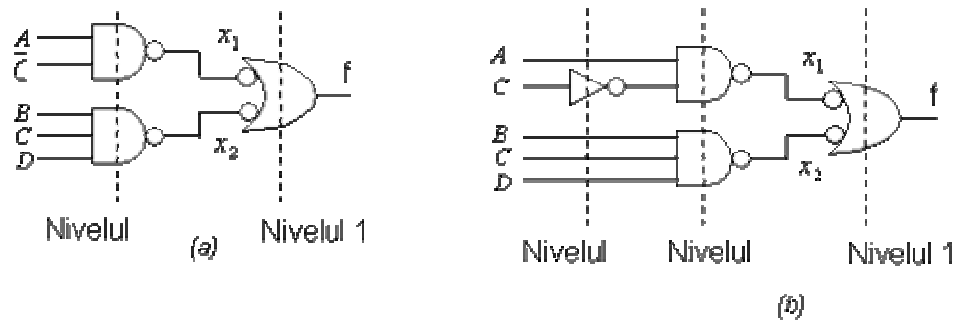


Fig. 4.3 (a) Circuit pe două nivele; (b) Circuit pe trei nivele

Adesea, implementarea unui circuit logic este restricționată de numărul de intrări ale porților logice utilizate. Se recurge, în acest caz, la rețele pe mai multe nivele. De exemplu funcția  $f(A,B,C,D,E)=A \cdot B \cdot C \cdot D \cdot E$  poate fi implementată cu o singură poartă ȘI cu 5 intrări (fig. 4.4a), sau în cazul când nu sunt disponibile decât porți ȘI cu două intrări se poate alege soluția implementării pe 4 nivele (fig. 4.4b) sau pe 5 nivele (fig. 4.4c).

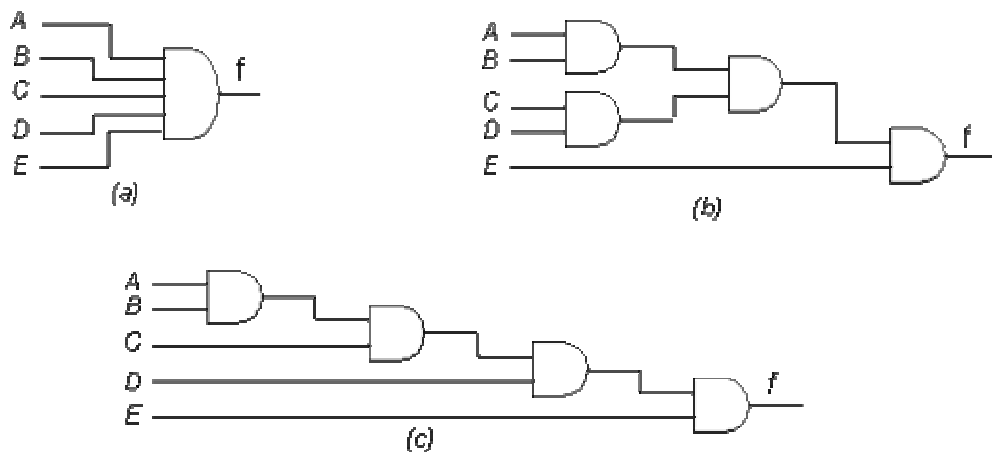


Fig. 4.4 Implementarea funcției  $f(A,B,C,D,E) = A \cdot B \cdot C \cdot D \cdot E$  în rețele multinivel: (a) Un singur nivel cu o poarta ȘI cu 5 intrări; (b) 3 nivele cu 4 porți ȘI cu 2 intrări; (c) 4 nivele cu 4 porți ȘI cu 2 intrări

**D. Circuitele ȘI-SAU-NU** constau dintr-un set de porți ȘI ale căror ieșiri se leagă la intrările unei porți SAU-NU, putând fi utilizate astfel la implementarea funcțiilor logice în formă normală disjunctivă (sumă de produse), în circuite pe două nivele. În general, configurația circuitelor ȘI-SAU-NU este dată de numărul de intrări ale porților ȘI. Astfel circuitul care utilizează 3 porți ȘI cu una dintre porți având 2 intrări, alta 3 și alta 4 intrări, poate fi identificat ca circuit ȘI-SAU-NU 2-3-4. În figura 4.5a este reprezentat circuitul ȘI-SAU-NU 7454 (4x2 intrări), din seria de circuite integrate logice standard 7400. Funcția realizată de circuit este  $f = \overline{AB + CD + EF + GH}$ .

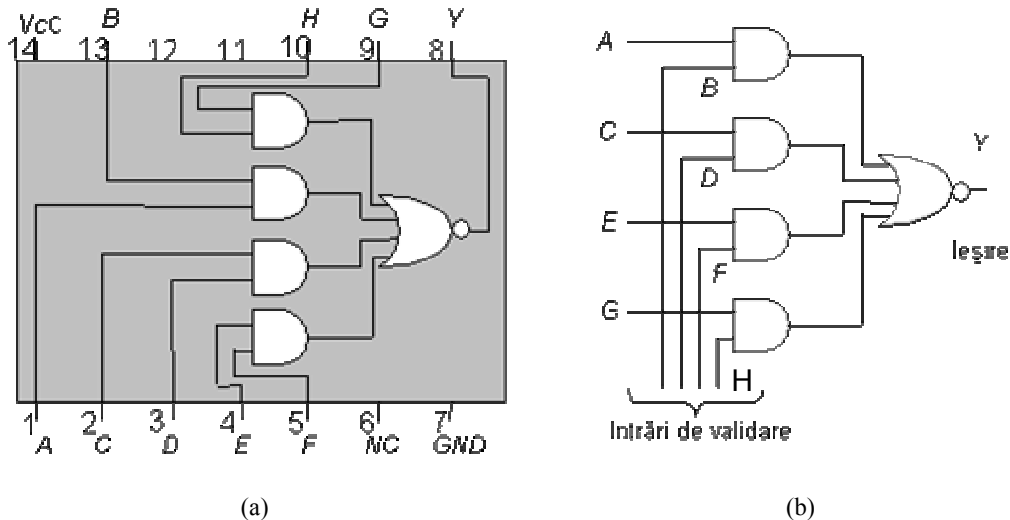


Fig. 4.5 Circuitul ȘI-SAU-NU 7454 (4x2 intrări): (a) Configurație terminale(vedere de sus); (b) 7454 utilizat ca multiplexor 4 în 1

Circuitul 7454 poate fi utilizat într-o mulțime de aplicații. Una dintre utilizări poate fi aceea de multiplexor 4 în 1, ca în figura 4.5b, caz în care 4 intrări (de exemplu A, C, E G) sunt folosite ca intrări de date iar celelalte 4 intrări (B, D, F, și H) ca intrări de validare. Prin aplicarea unui 1 logic numai pe una din intrările de validare, poarta corespunzătoare se deschide permițând transmiterea la ieșire a semnalului de la intrarea de date a acelei porți. De exemplu, pentru a trimite la ieșire intrarea C, vom aplica la intrările de validare  $D=1$  și  $B=F=H=0$ .

**E. Factorizarea** este o tehnică folosită pentru obținerea unor forme ale funcțiilor logice cu mai multe nivele, ce se implementează în circuite fără restricții în ceea ce privește propagarea semnalelor. Aceste circuite utilizează de regulă mai puține porți și deci sunt mai economice. Tehnica se utilizează de asemenea și în situația folosirii porților cu număr mic de intrări. În acest caz, se reduce numărul de variabile (simple sau negate, numite generic **literali**) din produse sau sume, astfel încât acesta să fie mai mic sau cel mult egal cu numărul de intrări ale porților. Factorizarea folosește distributivitatea algebrei booleene și cere experiență în proiectarea circuitelor logice.

**EXEMPLUL 4.3.** Să se implementeze funcția  $f(A,B,C,D) = A\bar{B} + A\bar{D} + A\bar{C}$

- printr-o rețea ȘI-NU pe două nivele;
- printr-o rețea mai simplă utilizând factorizarea.

În figura 4.6a. este reprezentat circuitul care implementează direct funcția pe două nivele, iar în figura 4.6b circuitul obținut pe baza formei simplificate a funcției prin utilizarea distributivității și a legilor lui DeMorgan din algebra booleană  $f(A,B,C,D) = A(\bar{B} + \bar{D} + \bar{C}) = A(\overline{BCD})$ .

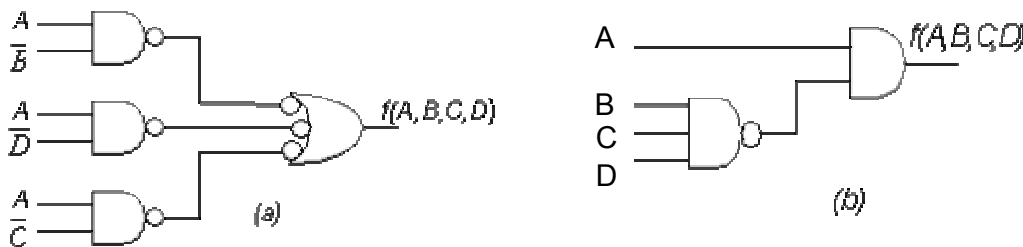


Fig. 4.6. (a) Implementarea directă a funcției; (b) Implementarea funcției după factorizare

Se observă că în primul caz s-au utilizat 4 porți și 9 intrări iar în al doilea caz numai 2 porți și 5 intrări.

### C4.2.2. Simplificarea funcțiilor logice

Scopul principal al simplificării funcțiilor logice (booleene) este acela al minimizării costului de realizare a acestora cu elemente fizice de circuit. La simplificarea funcțiilor logice în forma normală disjunctivă (sumă de produse) pe două nivele minimizarea costului implică reducerea numărului de termeni, cu efect asupra reducerii numărului de porți și reducerea numărului de variabile din cadrul fiecărui produs, cu efect asupra reducerii numărului de intrări ale porților utilizate.

**Simplificarea** circuitului poate fi **limitată** de următorii factori:

- Numărul de intrări al porții (fan-in);
- Numărul de ieșiri ce pot fi comandate (fan-out);
- Viteza de transfer (care poate dicta utilizarea implementărilor pe 2 nivele, mai rapide decât a celor lente pe 3, 4 sau mai multe nivele);
- Hazardul, adică modificarea nedorită a nivelului de la ieșirea porții.

Minimizarea funcțiilor logice poate fi făcută prin metode euristice, care depind de abilitatea și experiența proiectantului, utilizând legile și principiile algebrei booleene.

#### EXEMPLUL 4.4

$f(X,Y,Z) = \overline{X}Y(Z + \overline{Y}Z) + \overline{Y}Z = \overline{X}YZ + \overline{X}Y\overline{Y}Z + \overline{Y}Z = \overline{X}YZ + \overline{Y}Z = \overline{X}Z + \overline{Y}Z = (\overline{X} + \overline{Y})Z$ . La minimizare s-au utilizat distributivitatea, contradicția și terțul exclus.

Forma finală a funcției nu este în mod cert forma optimă. Pentru a realiza minimizarea funcțiilor booleene în mod eficient se utilizează tehnici standard, cum ar fi cea care utilizează diagramele Veitch-Karnaugh.

**Diagrama Veitch-Karnaugh** este o extensie a conceptelor de tabel de adevăr, diagrame Venn și mintermeni. Transformarea unei diagrame Venn într-o diagramă Karnaugh se face urmând pașii arătați în figura 4.7.

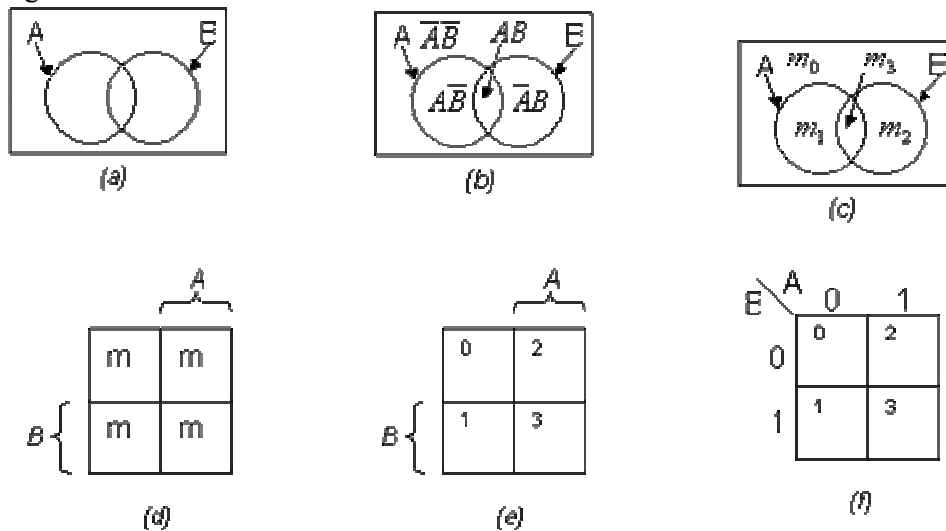


Fig. 4.7 Transformarea diagramelor Venn în diagrame Veitch-Karnaugh

Astfel diagrama Venn din figura 4.7a, în care celor două variabile le corespund mulțimile A și B, este reprezentată în altă formă în figura 4.7b, în care se pun în evidență mulțimile disjuncte  $AB, \overline{A}B, A\overline{B}, \overline{A}\overline{B}$ . Cum acestea au chiar expresia mintermenilor, diagrama poate fi redesenată ca în figura 4.7c. Alocând fiecărui mintermen aceeași suprafață pe care o reprezentăm la fel, diagrama ia forma din figura 4.7d. Păstrând numai indicii mintermenilor, diagrama va arăta ca în figura 4.7e., diagrama Veitch, sau ca în figura 4.7f, diagrama Karnaugh. Se poate spune că diagrama Karnaugh

este o formă particulară a tabelului de adevăr. Dacă în tabelul de adevăr fiecărui mintermen îi corespunde o linie, în diagrama Karnaugh îi corespunde o celulă (un pătrat). În figura 4.8a. este reprezentată o diagramă Veitch de 3 variabile. Diagrama Veitch de 4 variabile se obține prin extensia diagramei Veitch de 3 variabile așa cum se poate observa în figura 4.8b.

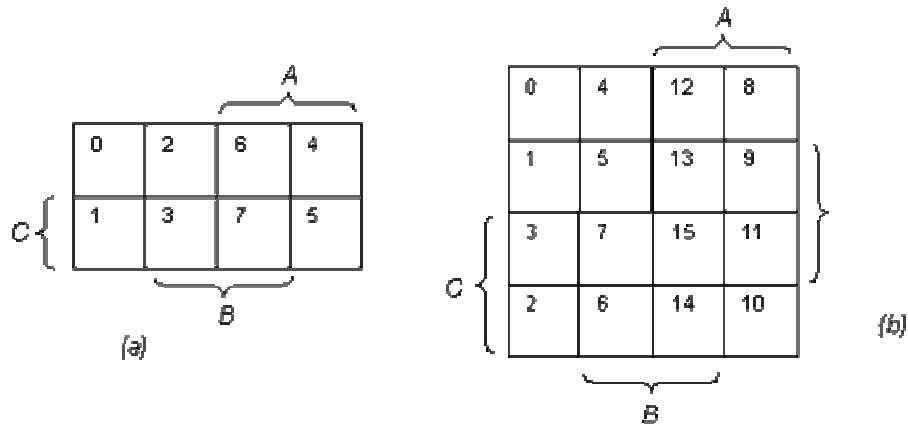


Fig. 4.8 Diagrame Veitch de: (a) 3 variabile și (b) de 4 variabile

Diagrama pentru 5 variabile (A,B,C,D,E) este formată din două diagrame pentru 4 variabile (B,C,D,E), una conținând termenii cu  $\bar{A}$  iar cealaltă termenii cu A. În figura 8.9a diagrama Veitch de 5 variabile este reprezentată spațial, cele două jumătăți ale diagramei fiind așezate una peste cealaltă. În figura 8.9b este reprezentată spațial diagrama Veitch pentru 6 variabile (A,B,C,D,E,F) care se obține din 4 diagrame Veitch de 4 variabile (C,D,E,F).

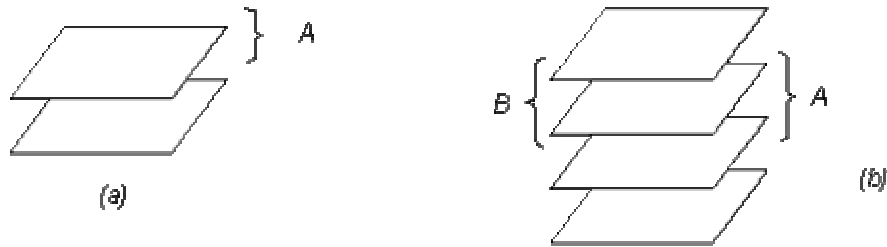


Fig. 8.9 Diagrame Veitch de (a) 5 variabile și (b) 6 variabile.

**Observație:** Diagrame pentru mai mult de 6 variabile nu se utilizează ele nefiind practice datorită complexității.

Formele canonice ale funcțiilor pot fi ușor transpuse în diagrame Veitch-Karnaugh și invers. Pentru aceasta se asociază funcției 1 logic pentru fiecare mintermen care apare în expresia în forma disjunctivă (sumă de produse) și 0 logic pentru fiecare maxtermen care apare în expresia în forma conjunctivă (produs de sume) a funcției.

**EXEMPLUL 4.5.** Să se completeze diagrama Veitch pentru funcția dată prin expresia  $f(A,B,C) = \sum m(0,3,5) = m_0 + m_3 + m_5 = \prod M(1,2,4,6,7) = M_1 M_2 M_4 M_6 M_7$

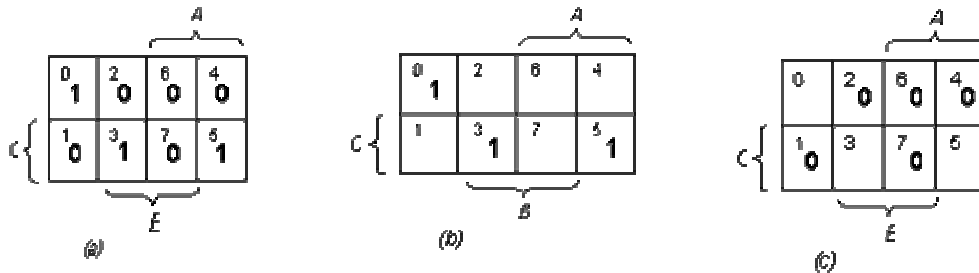


Fig. 4.10 Transpunerea formelor canonice ale funcțiilor în diagrame Veitch.

De regulă, când funcția este reprezentată în forma canonică disjunctivă, se omit maxtermenii, deci implicit nu se marchează 0 în diagramă (fig. 4.10b), iar atunci când funcția este reprezentată în forma canonică conjunctivă se omit mintermenii, deci nu se încarcă în diagramă decât 0 (fig. 4.10c). Diagrama Veitch poate fi utilizată pentru realizarea listei mintermenilor sau maxtermenilor din formele normale disjunctive respectiv conjunctive ale funcției.

#### C4.2.3. Simplificarea funcțiilor booleene utilizând diagrame Veitch-Karnaugh

Simplificarea în cadrul diagramelor Veitch-Karnaugh (V-K) se bazează pe faptul că adiacența logică corespunde adiacenței fizice. Doi mintermeni sunt adiacenți logic dacă diferă doar printr-o variabilă. De exemplu  $\overline{A}BC\overline{D}$  și  $\overline{A}BCD$  sunt adiacenți ei diferind doar prin variabila  $D$ . Conform teoremelor algebrei booleene  $\overline{A}BC\overline{D} + \overline{A}BCD = \overline{A}BC$ . În general, orice combinație a doi termeni adiacenți duce la eliminarea unei variabile. Diagrama V-K este construită astfel încât celulele învecinate (adiacente fizic) să corespundă mintermenilor adiacenți logic. Gruparea a doi sau mai mulți termeni adiacenți se face intuitiv prin încercuire și duce în final la eliminarea uneia sau mai multor variabile.

#### Reguli de simplificare:

1. Fiecare pătrat/celulă (ce corespunde unui mintermen) a unei diagrame V-K de 2 variabile are 2 celule adiacente; fiecare celulă a unei diagrame V-K de 3 variabile are 3 celule adiacente și așa mai departe. În general fiecare celulă a unei diagrame V-K de  $n$  variabile are  $n$  celule adiacente.
2. Grupările de termeni (celule) pot cuprinde un număr de celule, putere a lui 2 (2, 4, 8, etc.). Gruparea de 2 celule elimină o variabilă, gruparea de 4 celule elimină 2 variabile ș.a.m.d., în general gruparea a  $2^n$  celule elimină  $n$  variabile.
3. Se grupează împreună un număr cât mai mare de celule eliminând în acest fel cât mai multe variabile din termenii expresiei funcției minimizezate.
4. Se încearcă acoperirea (gruparea) tuturor celulelor ce conțin 1 în cât mai puține grupări. Grupări puține înseamnă puțini termeni în forma normală a funcției minimizezate. O celulă este acoperită dacă este inclusă în cel puțin o grupare. O celulă poate fi utilizată pentru a realiza mai multe grupări, dacă este nevoie.
5. Grupările din diagramă încep cu acele celule "singuratică" care au cele mai puține celule adiacente ce conțin 1. Mintermenii cu multe celule adiacente ce conțin 1, au mai multe posibilități de grupare și vor fi lăsați pentru a fi grupați mai târziu.

**EXEMPLUL 4.6.** Se presupune că o bandă de montaj dintr-un atelier industrial poate fi oprită dacă cel puțin 2 din cele 3 posturi de lucru cer acest lucru. Să se deducă expresia funcției logice corespunzătoare, să se minimizeze utilizând diagramele Veitch-Karnaugh și să se implementeze cu porți ȘI-NU.

**Rezolvare:** se notează cu A, B și C cele trei variabile corespunzătoare. Valoarea lor este 1 dacă se cere oprirea și zero dacă nu se cere acest lucru. Tabelul de adevăr are forma:

A	B	C	Termenul canonic	Valoarea
0	0	0	$P_0 = \overline{A} \cdot \overline{B} \cdot \overline{C}$	0
0	0	1	$P_1 = \overline{A} \cdot \overline{B} \cdot C$	0
0	1	0	$P_2 = \overline{A} \cdot B \cdot \overline{C}$	0
0	1	1	$P_3 = \overline{A} \cdot B \cdot C$	1
1	0	0	$P_4 = A \cdot \overline{B} \cdot \overline{C}$	0
1	0	1	$P_5 = A \cdot \overline{B} \cdot C$	1
1	1	0	$P_6 = A \cdot B \cdot \overline{C}$	1
1	1	1	$P_7 = A \cdot B \cdot C$	1



Diagrama Veitch-Karnaugh și grupările de celule se prezintă în fig. 4.11.

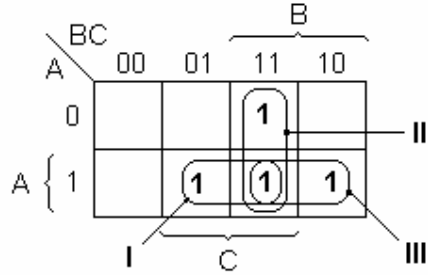


Fig. 4.11. Diagrama Veitch-Karnaugh și grupările de celule

După minimizarea suprafețelor elementare I, II și III rămân numai acei termeni care nu-și schimbă valoarea logică la trecerea dintr-o căsuță în alta a suprafeței elementare. Se obține astfel:

$$\text{I: } x_1 = A \cdot C, \text{ II: } x_2 = B \cdot C \text{ și III: } x_3 = A \cdot B$$

Forma minimizată a funcției este

$$y = x_1 + x_2 + x_3 = AC + BC + AB$$

Pentru a implementa cu același tip de porți (ȘI-NU) se aplică dubla negație și teoremele lui DeMorgan. Se obține:

$$y = \overline{\overline{AC + BC + AB}} = \overline{\overline{AC} \cdot \overline{BC} \cdot \overline{AB}}$$

Pentru implementare trebuie să se folosească 3 porți ȘI-NU cu câte 2 intrări fiecare și o poartă ȘI-NU cu 3 intrări (fig. 4.12).

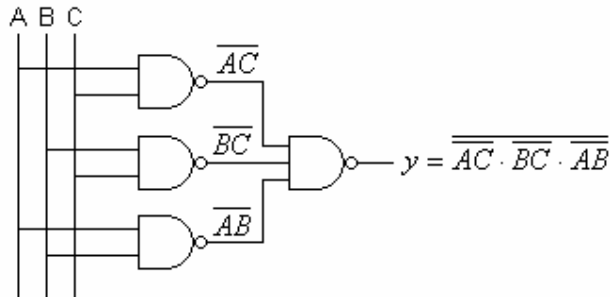


Fig. 4.12. Implementarea formei minimizate