

Cursul nr. 5

C5.1 Funcții incomplet definite

Funcțiile incomplet definite sunt acele funcții booleene care pentru anumiți termeni au valori incerte. În cadrul diagramelor V-K celulele corespunzătoare valorilor incerte conțin un simbol (de exemplu X) care poate avea valoarea 0 sau 1 funcție de necesități. Dacă ele ajută la formarea de grupări mai mari decât cele formate fără termenii incerti ele vor fi utilizate (deci X va lua valoarea 1 în cazul formelor disjunctive și 0 în cazul formelor conjunctive).

EXEMPLUL C5.1

Să se minimizeze funcția dată prin formele canonice disjunctive și conjunctive:

$$f(A,B,C,D) = \sum m(1,3,4,7,11) + d(5,12,13,14,15) = \prod M(0,2,6,8,9,10) \cdot D(5,12,13,14,15)$$

În fig. C5.1a este ilustrată diagrama Veitch a funcției în forma disjunctivă iar în fig. C5.1b pentru forma conjunctivă.

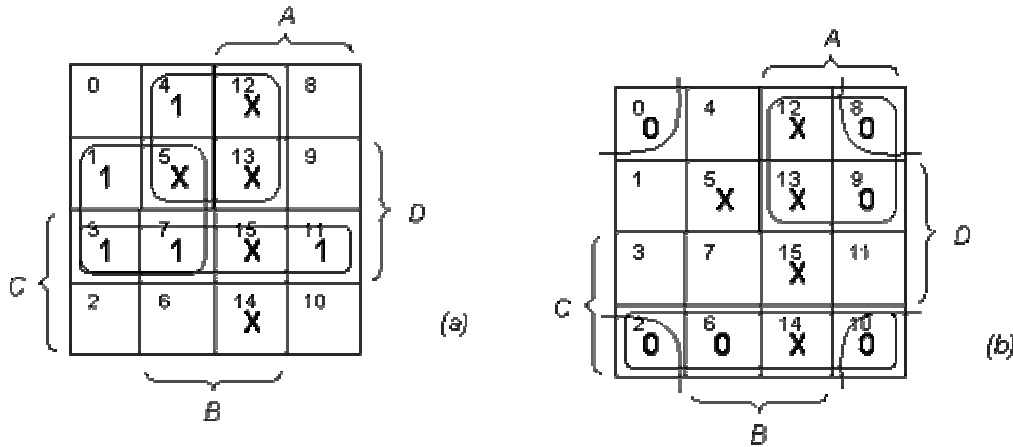


Fig. C5.1 Diagrame Veitch pentru funcția incomplet definită din exemplul 8.9

Din diagrama din fig. C5.1a se extrage expresia minimă a funcției în forma disjunctivă:

$$f(A,B,C,D) = B\bar{C} + \bar{A}D + CD$$

iar din diagrama din fig. C5.1b expresia minimă în forma conjunctivă:

$$f(A,B,C,D) = (B + D)(\bar{C} + D)(\bar{A} + C)$$

C5.2 Utilizarea diagramelor V-K pentru eliminarea hazardului static

Implementarea unei funcții logice în forma sa minimă poate fi compromisă de parametrii timpi de propagare prin porți, T_{PHL} (la tranziția de la 1 la 0) și T_{PLH} (la tranziția din 0 în 1). Acești parametri de timp au valori foarte mici (nanosecunde sau fracțiuni de nanosecunde, depinzând de tehnologie, fan-in, fan-out, etc.) dar în practică ei nu au aceeași valoare pentru două porți diferite chiar dacă sunt porți de același tip. Efectul acestor diferențe se resimte la ieșirea circuitului, tranziția anumitor intrări provocând evenimente nedorite, numite hazard.

EXEMPLUL C5.2

Se consideră funcția $z(x_1, x_2, x_3) = x_1x_2 + \bar{x}_1x_3$ implementată în circuitul din fig. C5.2.

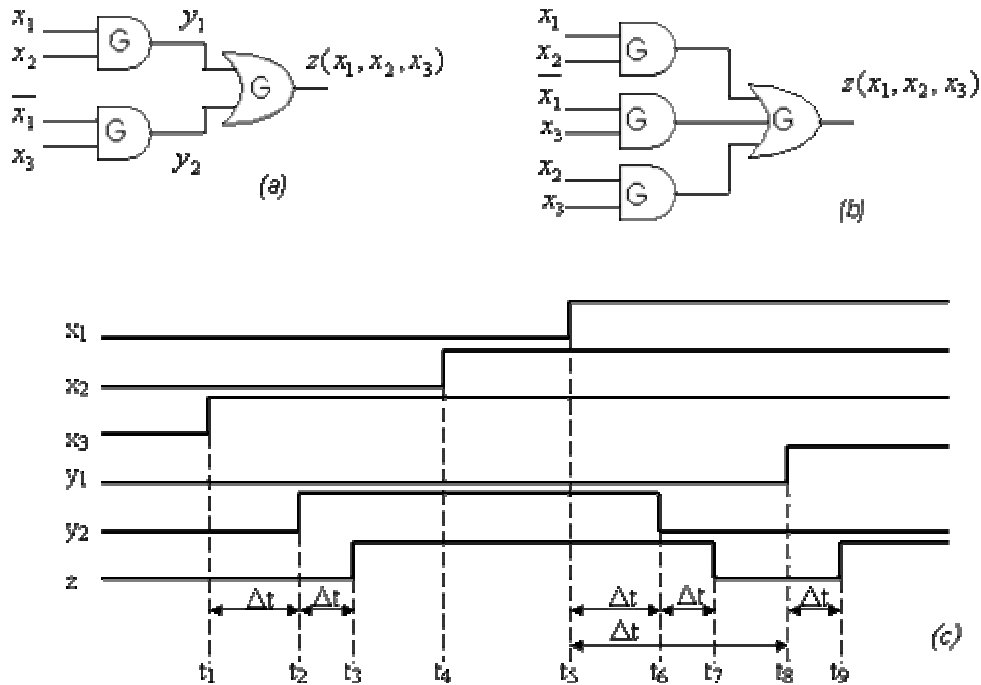


Fig. C5.2 Funcția din exemplul 8.10 implementată într-un: (a) Circuit cu hazard static; (b) Circuit fără hazard; (c) Diagrama temporală a circuitului cu hazard

Comportarea circuitului din fig. C5.2a este ilustrată în diagrama temporală din fig. C5.2c pentru o configurație particulară a semnalelor de intrare. Se consideră pentru toate porțile că $T_{PHL} = T_{PLH} = \Delta t_i$, dar $\Delta t_1 > \Delta t_2 > \Delta t_3$. Se observă că modificarea intrării x_3 în momentul t_1 provoacă, după o întârziere Δt_2 , trecerea ieșirii y_2 a porții $G2$ din 0 în 1, în momentul t_2 , care la rândul ei, determină în momentul $t_3 = t_2 + \Delta t_3$ trecerea ieșirii z a circuitului din 0 în 1. În momentul t_4 , x_2 se modifică din 0 în 1 dar nu se produce nici o modificare, poarta $G1$ rămânând închisă. În momentul t_5 , x_1 comută din 0 în 1. La momentul $t_6 = t_5 + \Delta t_2$, ieșirea porții $G2$, y_2 , se va modifica din 1 în 0, iar după un timp Δt_3 , la momentul t_7 și ieșirea z a circuitului va comuta din 1 în 0. La momentul $t_8 = t_5 + \Delta t_1$, după propagarea semnalului prin poarta $G1$, ieșirea y_1 va trece din 1 în 0 ceea ce provoacă după un timp Δt_3 comutarea ieșirii z din nou în starea 1 la momentul t_9 .

După cum se observă, hazardul apare la tranziția intrării x_1 din 0 în 1. Pentru o perioadă de timp ieșirea circuitului comută în 0, ceea ce relevă o comportare incorectă a circuitului. Cauzele hazardului din exemplul dat pot fi examinate în diagrama V-K din fig. C5.3a.

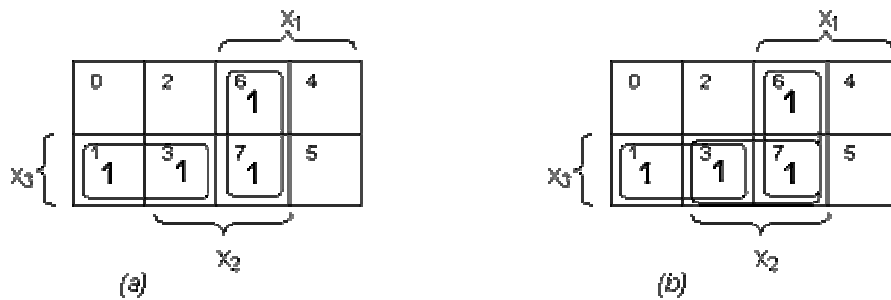


Fig. C5.3 Diagrama Veitch pentru circuitul: (a) cu hazard și (b) fără hazard

Acesta se produce la tranziția din starea definită de $x_1=0, x_2=1, x_3=1$ în $x_1=x_2=x_3=1$.

Pentru a preveni hazardul, se realizează încă o grupare, ca în fig. C5.2b, care va mări numărul de termeni ai sumei și va introduce încă o poartă ȘI, $G4$, în circuit, fig. C5.2b. În acest fel, pe perioada modificării ieșirilor porților $G1$ și $G2$, ieșirea lui $G4$ este în 1, astfel încât, ieșirea z rămâne și ea nemodificată.

În general, hazardul poate fi eliminat prin acoperirea fiecărei perechi de termeni adiacenți bineînțeles în detrimentul formei minime a funcției.

Hazardul luat în discuție este cunoscut ca hazard static 1, când ieșirea care trebuie să rămână pe 1 logic, trece temporar în 0 producând un impuls tranzitoriu (*glitch*). Este specific circuitelor ȘI-SAU care implementează formele normale disjunctive.

Există și hazard static 0, specific circuitelor SAU-ȘI, care implementează formele normale conjunctive, când ieșirea trebuie să rămână în 0, dar pentru o perioadă scurtă de timp ea trece în 1 logic. Pentru a elimina hazardul, în cazul acestor circuite, se introduc suplimentar implicați primi care acoperă toți termenii adiacenți din diagrama Veitch asociată. În circuit apar astfel porți SAU suplimentare.

C5.3 Circuite logice combinatoriale tipice

Sunt circuite logice combinatoriale care implementează funcții de complexitate medie, ce stau la baza structurii oricărui sistem numeric. În general circuitele logice combinatoriale sunt circuite integrate pe scară medie (codificatoare, decodificatoare, multiplexoare, demultiplexoare, sumatoare, comparatoare) dar și pe scară mare (memorii nevolatile - ROM, matrici logice programabile - PLA).

C5.3.1 Codificatorul

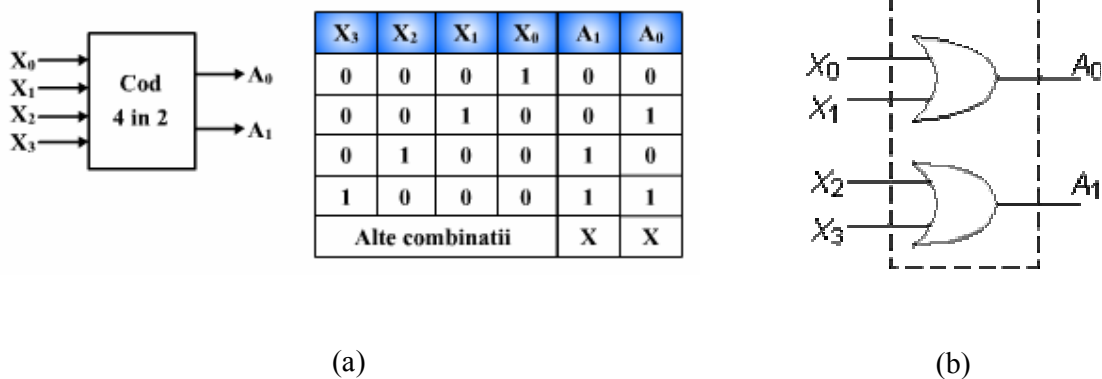
Codificatorul este circuitul logic combinatorial care generează la ieșire un cod unic pentru fiecare intrare activată.

Dacă codificatorul are n intrări atunci numărul de ieșiri m trebuie să satisfacă relația:

$$2^m \geq n \text{ sau } m \geq \log_2 n. \quad (1)$$

EXEMPLUL C5.3

Decodificatorul 4 în 2 are schema bloc și tabelul de adevăr în fig. C5.4a. Funcțiile A_1 și A_0 sunt incomplet definite, pentru ceilalți 12 mintermeni, care nu apar în tabel, valoarea funcțiilor este indiferentă. Pe baza tabelului de adevăr se poate proiecta circuitul ilustrat în fig. C5.4b, utilizând, pentru minimizare, diagramele Veitch din figurile C5.4c și d.



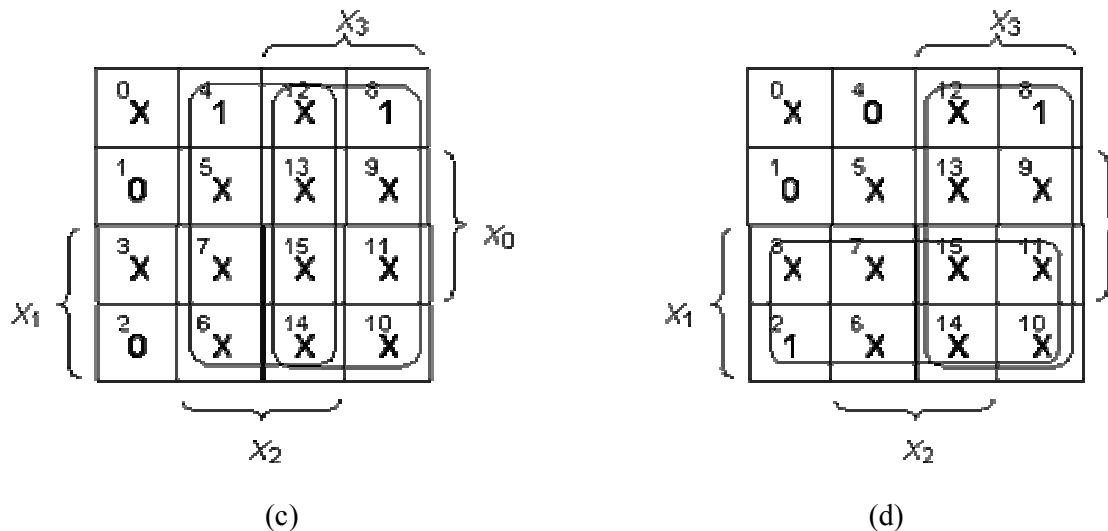


Fig. C5.4 Codificatorul 4 în 2: (a) Schemă bloc și tabelul de adevăr, (b) Circuit implementat; (c) Diagrama Veitch pentru funcția A_1 ; (d) Diagrama Veitch pentru funcția A_0

Din diagrama din fig. C5.4c rezultă $A_1=X_2+X_3$, iar din diagrama din fig. C5.4d, $A_0=X_1+X_3$.

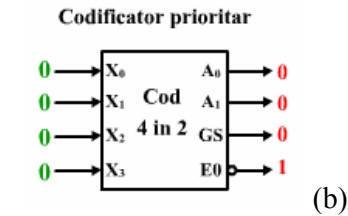
Codificatorul prioritar permite activarea mai multor intrări în același timp și trimite la ieșire semnalul de la intrarea cu prioritatea cea mai mare. Pentru simplificarea proiectării circuitului, prioritatea este dată de indicele (numărul) intrării.

EXEMPLUL C5.4

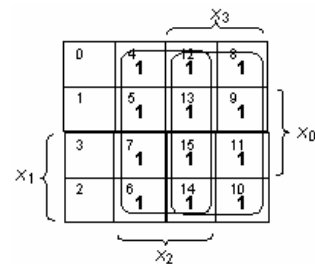
În fig. C5.5 este ilustrat modul de proiectare a unui codificator prioritar 4-în-2.

Intrări				Ieșiri			
X_3	X_2	X_1	X_0	A_1	A_0	GS	EO
0	0	0	0	0	0	0	1
0	0	0	1	0	0	1	0
0	0	1	0	0	1	1	0
0	0	1	1	0	1	1	0
0	1	0	0	1	0	1	0
0	1	0	1	1	0	1	0
0	1	1	0	1	0	1	0
0	1	1	1	1	0	1	0
1	0	0	0	1	1	1	0
1	0	0	1	1	1	1	0
1	0	1	0	1	1	1	0
1	0	1	1	1	1	1	0
1	1	0	0	1	1	1	0
1	1	0	1	1	1	1	0
1	1	1	0	1	1	1	0
1	1	1	1	1	1	1	0

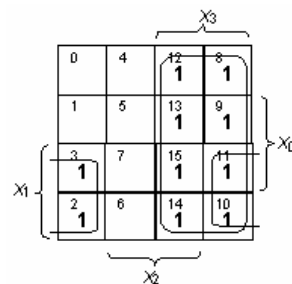
(a)



(b)



(c)



(d)

Fig. C5.5 Codificatorul prioritar 4 în 2: (a) Tabelul de adevăr; (b) Schema bloc; (c) Diagrama Veitch pentru ieșirea A_1 ; (d) Diagrama Veitch pentru ieșirea A_0 .

Pe lângă cele două ieșiri de date (A_1 și A_0) circuitul mai are și două ieșiri suplimentare EO care indică faptul că nici o intrare nu este activă ($EO=1$) și GS care indică faptul că una sau mai multe intrări sunt active ($GS=1$). Din diagrama din fig. C5.4c rezultă funcția $A_1 = X_2 + X_3$ iar din diagrama din fig. C5.5d funcția $A_0 = X_3 + X_1\overline{X_2}$. Din tabelul de adevăr din fig. C5.5a se pot scrie direct funcțiile EO și GS: $EO = \overline{GS} = \overline{X_3 + X_2 + X_1 + X_0}$.

Circuitul integrat codificator prioritar standard **74147** dispune de 10 intrări (0,1,..., 9) și 4 ieșiri (D, C, B, A). De notat că intrarea 0 nu este conectată la circuit deoarece ea nu apare în nici o expresie a funcțiilor de ieșire așa cum se sugerează și în exemplul C5.4, funcțiile A_1 și A_0 . Circuitul 74147 are și intrările și ieșirile active pe nivel scăzut.

Circuitul integrat logic **74148**, reprezentat în fig. C5.6, dispune de 8 intrări (0,1,...,7) și 3 ieșiri (A_2 , A_1 , A_0). În plus, circuitul mai are o intrare de validare EI, activă pe nivel scăzut, conectată la primul nivel al porților logice pentru a le controla funcționarea și 2 semnale de ieșire EO și GS cu semnificația descrisă în exemplul C5.4. 74148 are intrările și ieșirile active pe nivel scăzut.

Intrări									Ieșiri				
EI	0	1	2	3	4	5	6	7	A2	A1	A0	GS	EO
H	X	X	X	X	X	X	X	X	H	H	H	H	H
L	H	H	H	H	H	H	H	H	H	H	H	H	L
L	X	X	X	X	X	X	X	L	L	L	L	L	H
L	X	X	X	X	X	X	L	H	L	L	H	L	H
L	X	X	X	X	X	L	H	H	L	H	L	L	H
L	X	X	X	X	L	H	H	H	L	H	H	L	H
L	X	X	X	L	H	H	H	H	H	L	L	L	H
L	X	X	L	H	H	H	H	H	H	L	H	L	H
L	X	L	H	H	H	H	H	H	H	H	L	L	H
L	L	H	H	H	H	H	H	H	H	H	H	L	H

Fig. C5.6 (a) Codificatorul prioritar 74148- tabelul de adevăr

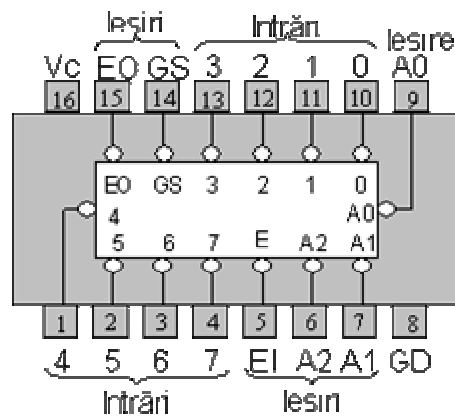


Fig. C5.6 (b) Codificatorul prioritar 74148 - configurația terminalelor

C5.3.2 Decodificatorul

Decodificatorul (fig. C5.7a) este un circuit logic combinațional care prezintă n intrări și în general 2^n ieșiri (uneori mai puține, de exemplu decodificatorul BCD-zecimal 7442).

Decodificatorul n -în- 2^n poate fi privit ca un generator de mintermeni, fiecare din ieșiri corespunzând exact unui mintermen.

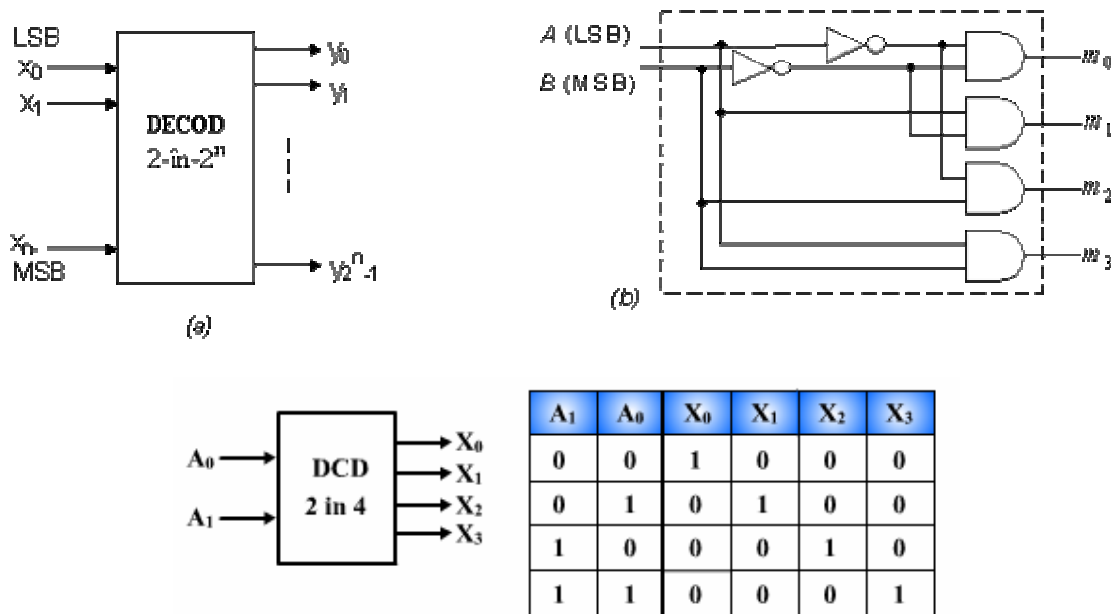


Fig. C5.7 Decodificatorul: (a) Decodificator n -în- 2^n ; (b) Decodificator 2-în-4 implementare; (c) Decodificator 2-în-4 funcționare

Cel mai simplu, dar și cel mai utilizat este decodificatorul 2-în-4 a cărui schemă este reprezentată în fig. C5.7b. Ieșirile reprezintă mintermenii funcției de 2 variabile: $m_0 = \bar{B}\bar{A}$, $m_1 = \bar{B}A$, $m_2 = B\bar{A}$, $m_3 = BA$. Acest circuit realizat cu porți ȘI se spune că are ieșirile active pe nivel ridicat (1 logic). Există și varianta cu ieșirile active pe nivel scăzut (0 logic), circuit care este realizat cu porți ȘI-NU (fig. C5.8a). Utilizând numai porți ȘI-NU decodificatorul 2-în-4 poate fi realizat și într-o altă variantă, prezentată în fig. C5.8b.

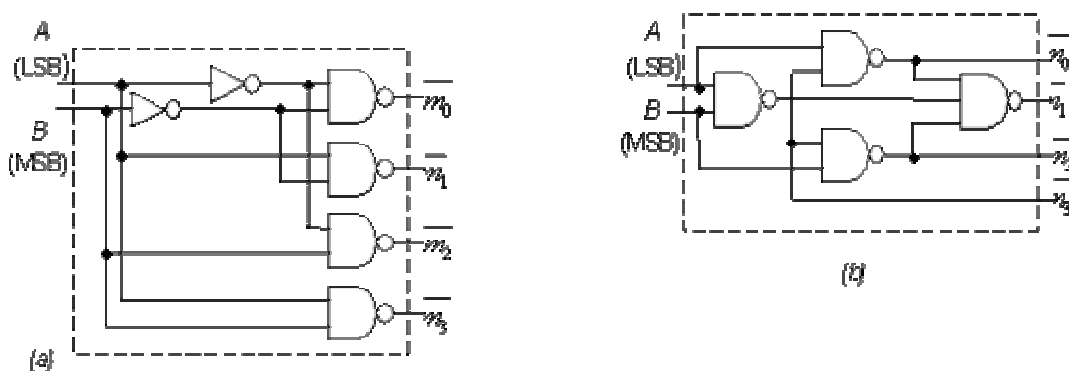


Fig. C5.8 Decodificatorul 2-în-4 realizat: (a) Cu porți ȘI-NU și ȘI; (b) Numai cu porți ȘI-NU

Circuitele din figurile C5.7b și C5.8a, realizate pe un nivel, necesită câte o poartă ȘI sau ȘI-NU pentru fiecare ieșire. Pentru decodificatoare cu 5 sau 6 intrări schema devine deja nepractică deoarece se ajunge la limita fan-in-ului acestor porți. Se recurge atunci la scheme de **decodificatoare ramificate** ca de exemplu cea din fig. C5.9a, care reprezintă un decodificator 3-în-8. Acest tip de decodificator se realizează pe mai multe nivele cu porți ȘI numai cu 2 intrări.

Un alt tip de structură de **decodificator** este cel așa numit **arbore dual**, ca cel reprezentat în fig. C5.9b, 4-în-16. La acest tip de structură, cele n linii de intrare se împart în două grupe a câte j respectiv k linii. Se folosesc 2 decodificatoare j -în- 2^j respectiv k -în- 2^k care generează 2^j respectiv 2^k semnale interne și 2^n porți ȘI care combină aceste semnale interne pentru a forma cele 2^n ieșiri ale decodificatorului.

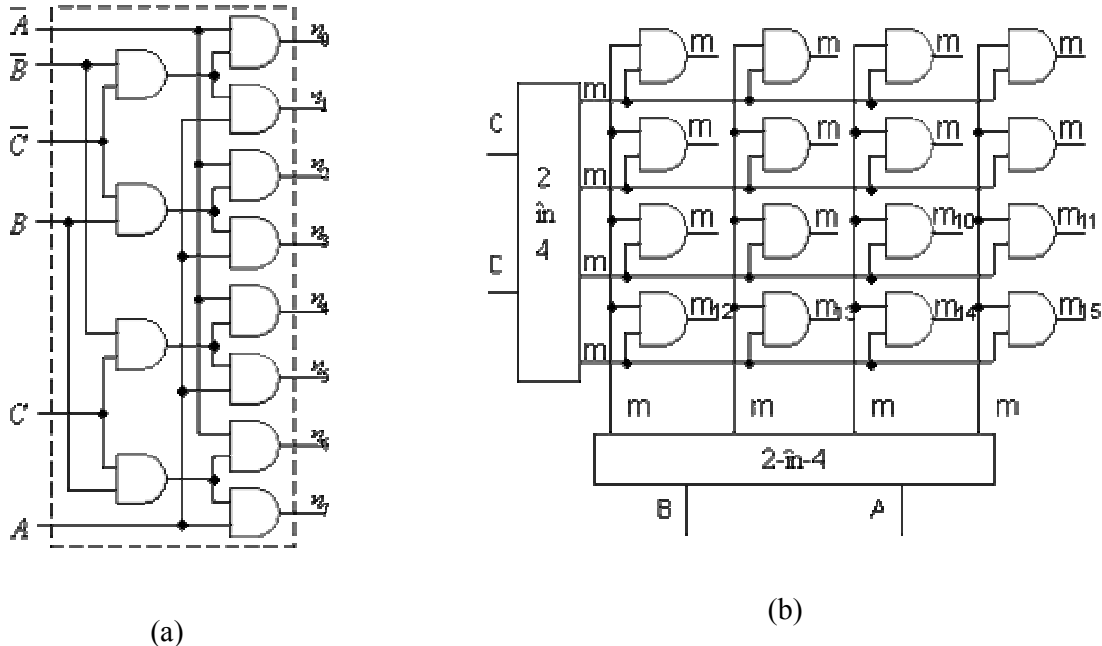


Fig. C5.9 Structuri de decodificare alternative: (a) ramificate; (b) arbore dual

Decodificatorul poate fi utilizat la implementarea funcțiilor logice date sub forma canonică, indiferent de nivelul activ al ieșirilor decodificatorului și de tipul porții utilizate la ieșire.

EXEMPLUL C5.5

Să se implementeze cu ajutorul decodificatorului funcția

$$f(A, B, C) = \sum m(0, 1, 4, 6, 7) = \prod M(2, 3, 5)$$

Se utilizează la implementare un decodificator 3-în-8 și una din următoarele variante:

1. Decodificator cu nivel activ ridicat la ieșire și poartă SAU:

$$f(A, B, C) = m_0 + m_1 + m_4 + m_6 + m_7$$

2. Decodificator cu nivel activ scăzut la ieșire și poartă ȘI-NU:

$$f(A, B, C) = \overline{m_2} \overline{m_3} \overline{m_5}$$

3. Decodificator cu nivel activ ridicat la ieșire și poartă SAU-NU:

$$f(A, B, C) = \overline{m_2} + \overline{m_3} + \overline{m_5}$$

4. Decodificator cu nivel activ scăzut la ieșire și poartă ȘI:

$$f(A, B, C) = \overline{m_2} \overline{m_3} \overline{m_5}$$

În fig. C5.10a s-a implementat varianta 1 iar în fig. C5.10b varianta 4.

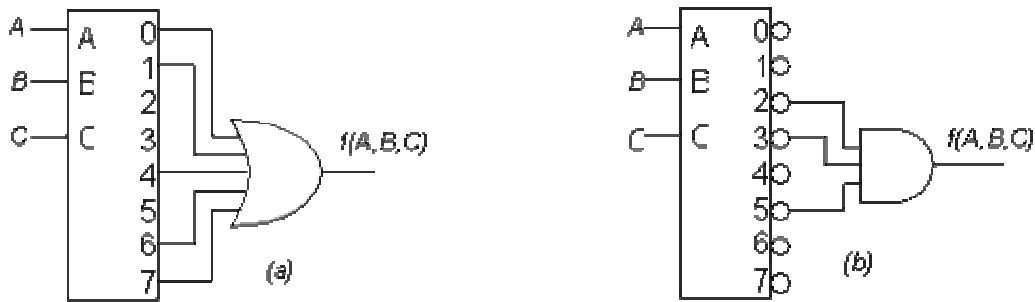


Fig. C5.10 Utilizarea decodificatorului la implementarea funcțiilor logice:
 (a) Varianta 1 din exemplul C5.5; (b) Varianta 4 din exemplul C5.5

Decodificatoarele ca și alte circuite includ de regulă, una sau mai multe **intrări de validare**, E (*enable*). Ele pot să permită relizarea funcției sau să o inhibe caz în care va forța toate ieșirile să treacă în starea inactivă. De exemplu în fig. C5.11 este reprezentat un decodificator 2-în-4 cu intrare de validare. Funcționarea sa poate fi sintetizată de expresia $y_k = m_k E$, toate ieșirile trecând în 0 atunci când $E=0$ și $y_k = m_k$ atunci când $E=1$.

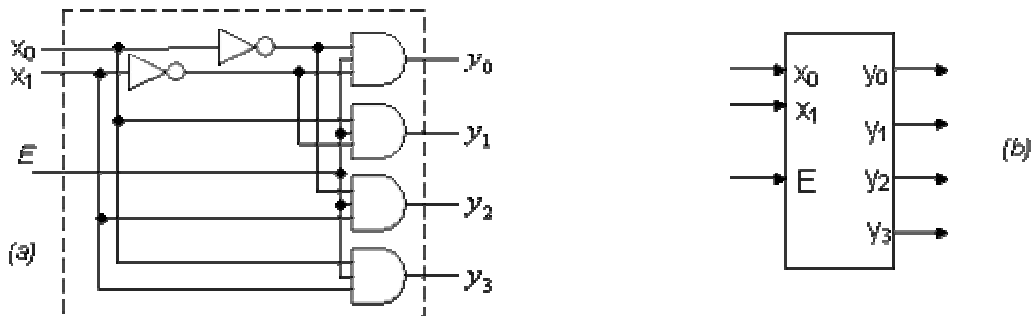


Fig. C5.11 Decodificatorul 2-în-4 cu intrare de validare: (a) Schema circuit; (b) Schema bloc

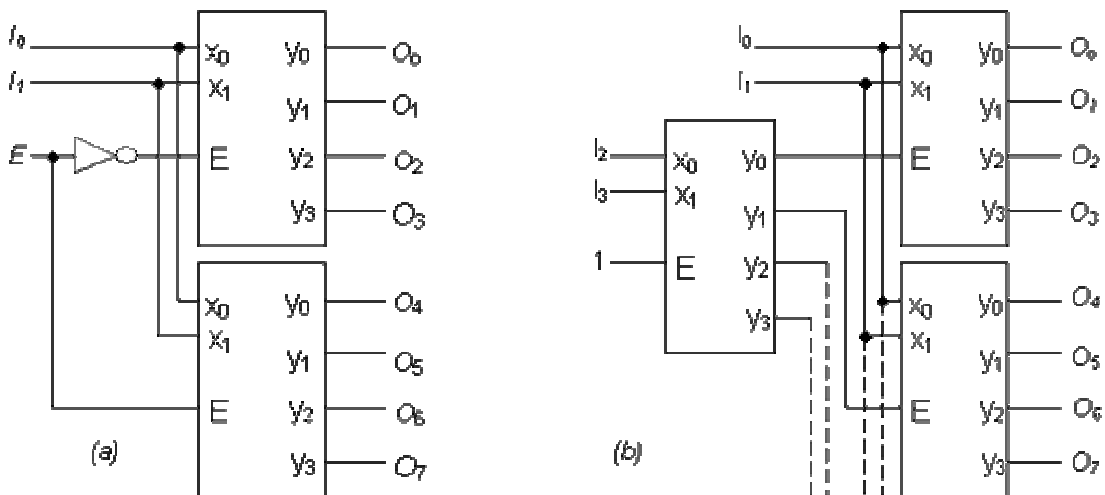


Fig. C5.12 Utilizarea decodificatorului 2-în-4 la implemetarea decodificatoarelor: (a) DCD 3-în-8;
 (b) DCD 4-în-16.

Decodificatorul din fig. C5.11 poate fi utilizat la realizarea decodificatoarelor 3-în-8 și 4-în-16 așa cum este ilustrat în fig. C5.12. În fig. C5.12a, când $I_2=0$ se validează circuitul de sus care permite decodificarea cuvintelor 000, 001, 010 și 011 (0-3 în zecimal) iar când $I_2=1$ se validează circuitul de jos pentru codurile 4 până la 7. În fig. C5.12b (reprezentare parțială) decodificatorul de la intrare, la

care se aplică biții I_2 și I_3 , validează unul din codificatoarele de ieșire ce au ca intrări de date liniile I_0 și I_1 .

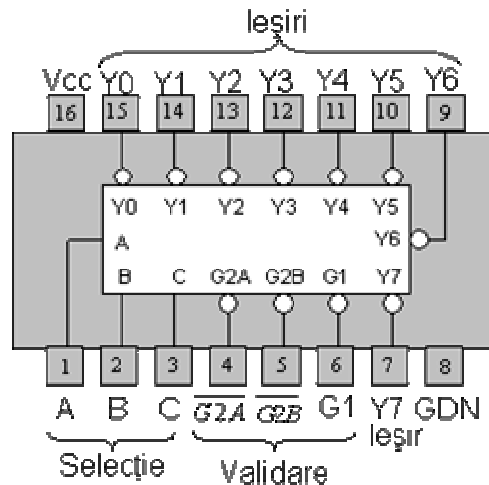
Există o mare varietate de circuite integrate decodificatoare cu număr de intrări n diferit și diferite configurații ale intrărilor de control.

Foarte utilizat este circuitul integrat **74138**, decodificator 3-în-8, care are ieșirile active pe nivel scăzut și este validat de 3 semnale $G1$, $G2A$ și $G2B$. Funcționarea sa poate fi sintetizată de relația

$Y_i = \overline{m_i}(G1G2AG2B)$. În fig. C5.13 este reprezentat tabelul de adevăr și configurația pinilor.

Intrări					Ieșiri							
Validare		Selecție										
$G1$	$\overline{G1^*}$	C	B	A	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
H	L	L	L	L	L	H	H	H	H	H	H	H
H	L	L	L	H	H	L	H	H	H	H	H	H
H	L	L	H	L	H	H	L	H	H	H	H	H
H	L	L	H	H	H	H	H	L	H	H	H	H
H	L	H	L	L	H	H	H	H	L	H	H	H
H	L	H	L	H	H	H	H	H	H	L	H	H
H	L	H	H	L	H	H	H	H	H	H	L	H
H	L	H	H	H	H	H	H	H	H	H	H	L
X	H	X	X	X	H	H	H	H	H	H	H	H
L	X	X	X	X	H	H	H	H	H	H	H	H

Fig. C5.13 (a) Decodificatorul 3-în-8 tip 74138 - Tabel de adevăr



$$\overline{G2^*} = \overline{G2A} + \overline{G2B}$$

Fig. C5.13 (b) Decodificatorul 3-în-8 tip 74138 - Configurația pinilor

Alt circuit logic standard utilizat este decodificatorul 4-în-16, **74154**. Acest circuit are două intrări de validare $G1$ și $G2$ active pe nivel scăzut ca și ieșirile. Funcționarea poate fi descrisă pe scurt prin

relația $Y_i = \overline{m_i}(G1G2)$.

Aplicații ale decodificatoarelor:

- **Decodificatoare de adrese** pentru selecția unei locații de memorie sau a unui periferic de intrare-ieșire. Memoriile și porturile perifericelor sunt legate la aceleași linii prin care sunt trimise *adresele* de selecție. La un moment dat numai un periferic (sau locație de memorie) poate fi selectat și anume acela care este legat la ieșirea activată a decodicatorului adresei.
- **Implementare de funcții logice** (exemplul C5.5), aplicație bazată pe faptul că decodicatorul este, cum s-a mai spus, un generator de mintermeni.
- **Decodificatoare BCD-zecimal**, care pot fi realizate prin proiectare specifică cu ajutorul diagramelor V-K și apoi implementate în circuite logice, 7442, sau din circuite 74154 prin folosirea numai a intrărilor 0-9.
- **Decodificatoare pentru afișoare pe 7 segmente**, care au ca intrări cei 4 biți ai codului BCD (zecimal codificat binar) iar ca ieșiri cele 7 segmente ale cifrelor zecimale.