

L5. IMPLEMENTAREA CU PORȚI A STRUCTURILOR LOGICE COMBINAȚIONALE. MINIMIZAREA FOLOSIND DIAGrame VEITCH KARNAUGH. DECODOARE.

1. Obiective

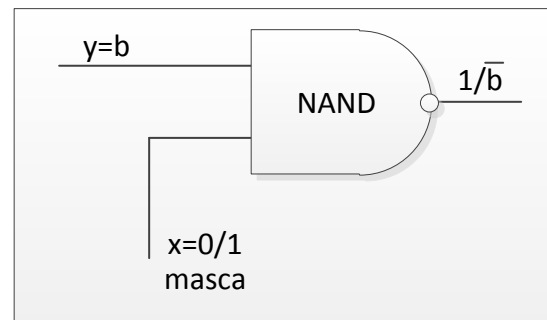
Prin parcurgerea acestei ședințe de laborator studenții vor fi capabili:

- Să implementeze o funcție logică descrisă în limbaj natural;
- Să implementeze funcții logice folosind numai porți de tip NAND și numai de tip NOR;
- Să minimizeze funcții logice folosind metoda Veitch Karnaugh;
- Să implementeze un codificator.

2. Porți logice NAND și NOR

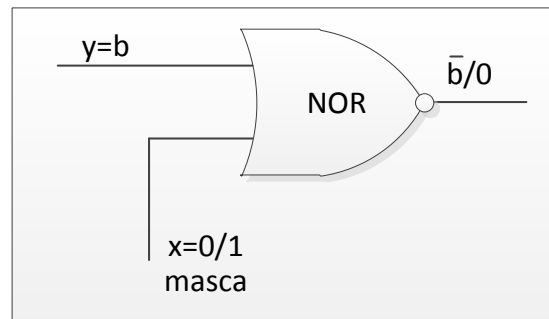
2.1. Poarta NAND (NOT AND)

x	y	$\overline{xy}$
0	0	1
0	1	1
1	0	1
1	1	0



2.2. Poarta NOR (NOT OR)

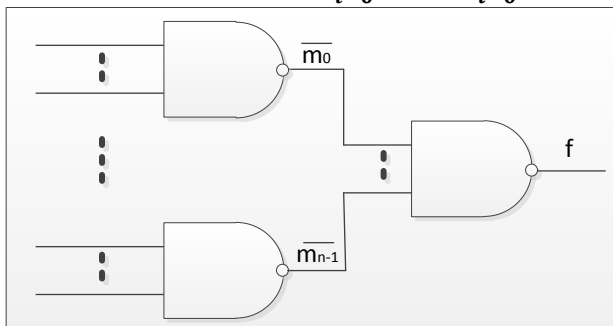
x	y	$\overline{x+y}$
0	0	1
0	1	0
1	0	0
1	1	0



3. Implementări ale funcțiilor logice cu porți NAND sau NOR

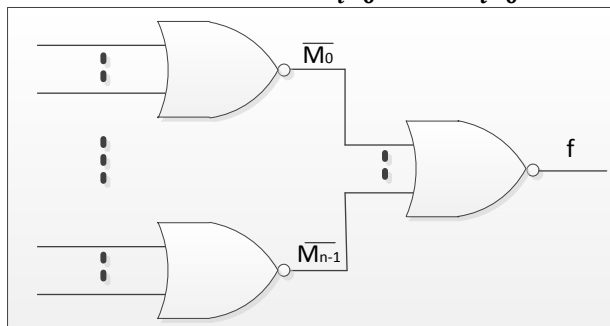
3.1. Implementarea cu porți NAND

$$f(x_0, x_1, \dots, x_{n-1}) = \sum_{i=0}^{2^n-1} m_i = \prod_{i=0}^{2^n-1} \overline{m}_i$$

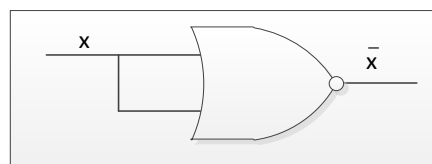
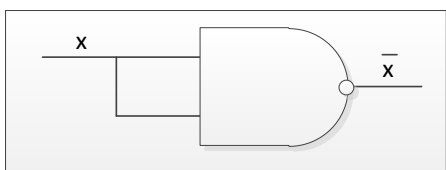


3.2. Implementarea cu porți NOR

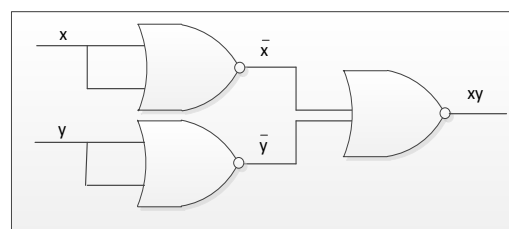
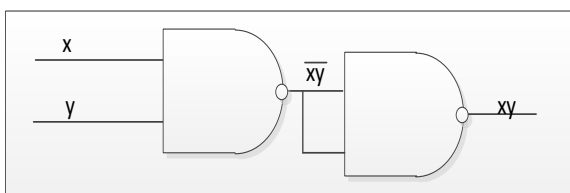
$$f(x_0, x_1, \dots, x_{n-1}) = \prod_{i=0}^{2^n-1} M_i = \sum_{i=0}^{2^n-1} \overline{M}_i$$



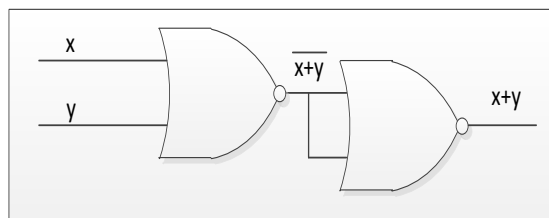
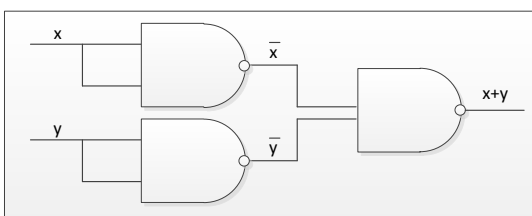
Implementarea porții NOT



Implementarea porții AND



Implementarea porții OR



Costul este numărul de intrări în circuitele logice elementare folosite în implementarea funcției.

Numărul de nivele este numărul maxim de porți străbătute de semnal de la intrare la ieșire.

4. Minimizarea cu metoda Veitch Karnaugh

Minimizarea unei funcții logice presupune obținerea unei expresii cu cost minim pentru un număr de nivele dat.

O funcție logică poate fi minimizată prin:

- 1) Aplicarea proprietăților și teoremele algebrei logice;
- 2) Metode sistematice (metoda Veitch-Karnaugh sau metoda Quine McCluskey).

Metoda de minimizare Veitch-Karnaugh

Un sir de n variabile poate fi partiționat în două grupe astfel:

$$x_{n-1}x_{n-2} \dots x_1x_0$$

$$\begin{cases} x_{n-1} \dots x_p \\ x_p \dots x_0 \end{cases}$$

cu $p = \begin{cases} \frac{n}{2} & n \text{ este par} \\ \frac{n+1}{2} & n \text{ este impar} \end{cases}$. Se vor parcurge următoarele etape:

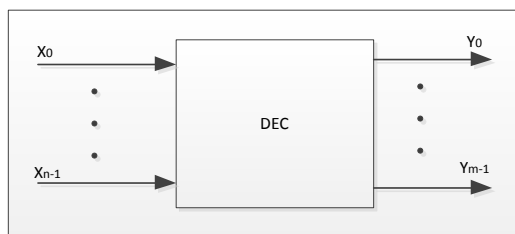
- 1) Se construiește o matrice ce conține un număr de linii și coloane determinat de partiția de variabilă fixată: $[2^k, 2^p]$, unde k reprezintă numărul de elemente din primul termen al partiției, iar p numărul de elemente din al doilea termen al partiției;
- 2) Pentru ordonarea liniilor și a coloanelor se folosește codul Gray, care are proprietatea de adiacență. Adunând două conjuncții sau înmulțind două disjuncții adiacente dispare termenul care diferă;
- 3) Se completează matricea cu valori de 1 (conjuncții) sau de 0 (disjuncții) sau $\emptyset$ (indiferent) ale funcției în celulele matricei, corespunzător combinațiilor variabilelor;
- 4) Se grupează celulele vecine în grupe cât mai mari, formate dintr-un număr de elemente egal cu o putere a lui 2 (1,2,4 sau 8). Celulele laterale care sunt simetrice față de o axă mediană imaginară sunt vecine;
- 5) Fiecare grupă formează un termen minimizat, termen ce nu conține variabilele ce-și schimbă semnul și astfel dispar.

Observații!

- Un element poate face parte din mai multe grupe;
- Fiecare grupă trebuie să conțină cel puțin un element nou față de restul grupelor;
- Orice element aparține măcar unei grupe, fie aceasta formată chiar dintr-un singur element;
- Variabilele independente se scriu conform regulilor de la formele canonice.

5. Circuite logice

Decodificatorul este un circuit digital combinațional cu n intrări ($x_0, x_1, \dots, x_{n-1}$) și m ieșiri ($y_0, y_1, \dots, y_{m-1}$), cu $m = 2^n$. La un anumit moment este activă o singură ieșire y_k , având numărul de ordine k reprezentat codificat de intrări, în timp ce toate celelalte ieșiri sunt inactive.



Codificatorul generează un anumit cod binar la ieșire pentru o anumită intrare activă. Funcționează invers ca un decodificator.

Convertoarele de cod sunt circuite combinaționale care transformă un cod într-un alt cod. Un astfel de circuit are n_i intrări și n_o ieșiri, n_i fiind numărul de biți prin care se codifică codul de intrare și n_o numărul de biți prin care se codifică codul rezultat.

6. Aplicații rezolvate

6.1. Se dă funcția $f(x_2, x_1, x_0) = R_1(1,3,4)$.

x_2	x_1	x_0	$f(x_2, x_1, x_0)$	
0	0	0	0	M_0
0	0	1	1	m_1
0	1	0	0	M_2
0	1	1	1	m_3
1	0	0	1	m_4
1	0	1	0	M_5
1	1	0	0	M_6
1	1	1	0	M_7

$$FCD = m_1 + m_3 + m_4 = \overline{m_1} \cdot \overline{m_3} \cdot \overline{m_4};$$

Costul = 12;

Nr_nivele=2.

$$FCC = M_0 \cdot M_2 \cdot M_5 \cdot M_6 \cdot M_7 = \overline{M_0} + \overline{M_2} + \overline{M_5} + \overline{M_6} + \overline{M_7};$$

Costul = 20;

Nr_nivele=2.

6.2. Să se minimizeze funcția $f(x_3, x_2, x_1, x_0) = R_1(1,2,6,8,10,11,15)$ utilizând metoda Veitch Karnaugh.

$x_3x_2 \backslash x_1x_0$	00	01	11	10
00	0	1	0	1
01	0	0	0	1
11	0	0	1	0
10	1	0	1	1

Se obțin grupările

$$G_1 = \overline{x_3} \cdot \overline{x_2} \cdot \overline{x_1} x_0;$$

$$G_2 = \overline{x_3} x_1 \overline{x_0};$$

$$G_3 = x_3 x_1 x_0;$$

$$G_4 = x_3 \overline{x_2} \cdot \overline{x_0}.$$

rezultând că $f = G_1 + G_2 + G_3 + G_4$.

6.3. Să se implementeze un convertor din cod binar în cod Gray:

x_3	x_2	x_1	x_0	y_3	y_2	y_1	y_0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	0
0	1	0	0	0	1	1	0
0	1	0	1	0	1	1	1
0	1	1	0	0	1	0	1
0	1	1	1	0	1	0	0
1	0	0	0	1	1	0	0
1	0	0	1	1	1	0	1
1	0	1	0	1	1	1	1
1	0	1	1	1	1	1	0
1	1	0	0	1	0	1	0
1	1	0	1	1	0	1	1
1	1	1	0	1	0	0	1
1	1	1	1	1	0	0	0

y_3

$x_1x_0 \backslash x_3x_2$	00	01	11	10
00				
01				
11	1	1	1	1
10	1	1	1	1

$$y_3 = x_3$$

 y_2

$x_1x_0 \backslash x_3x_2$	00	01	11	10
00				
01	1	1	1	1
11				
10	1	1	1	1

$$y_2 = \bar{x}_2x_3 + x_2\bar{x}_3 = x_2 \oplus x_3$$

 y_1

$x_1x_0 \backslash x_3x_2$	00	01	11	10
00			1	1
01	1	1		
11	1	1		
10			1	1

$$y_1 = \bar{x}_1x_2 + x_1\bar{x}_2 = x_1 \oplus x_2$$

 y_0

$x_1x_0 \backslash x_3x_2$	00	01	11	10
00		1		1
01		1		1
11		1		1
10		1		1

$$y_0 = \bar{x}_0x_1 + x_0\bar{x}_1 = x_0 \oplus x_1$$

7. Aplicații propuse

7.1. Se dau funcțiile logice

i) $f(x_2, x_1, x_0) = x_1x_0 + x_1x_2 + x_2x_0$

ii) $f(x_2, x_1, x_0) = R_1(2,3,4,5)$

Să se implementeze funcțiile folosind

a) numai porți de tip NAND;

b) numai porți de tip NOR;

și să se specifice costul și numărul de nivele al implementării.

7.2. Să se minimizeze cu metoda Veitch Karnaugh următoarele funcții:

a) $f(x_3, x_2, x_1, x_0) = R_1(4,5,8,9,10,12,13);$

b) $f(x_3, x_2, x_1, x_0) = R_1(0,1,2,4) + R_0(3,5,10);$

c) $f(x_3, x_2, x_1, x_0) = R_0(1,3,4,6,9,11,12,14);$

d) $f(x_3, x_2, x_1, x_0) = R_1(0,1,6,7,10,11,12,13);$

- e) $f(x_3, x_2, x_1, x_0) = R_0(2, 3, 4, 5, 8, 9, 14, 15);$
 f) $f(x_3, x_2, x_1, x_0) = R_1(2, 3, 6, 7, 8, 9, 12, 13);$
 g) $f(x_3, x_2, x_1, x_0) = R_0(0, 1, 4, 5, 10, 11, 14, 15);$
 h) $f(x_3, x_2, x_1, x_0) = R_1(0, 4, 8, 12);$
 i) $f(x_3, x_2, x_1, x_0) = R_1(3, 7, 11, 15);$
 j) $f(x_3, x_2, x_1, x_0) = R_0(4, 5, 6, 7, 8, 9, 10, 11);$
 k) $f(x_3, x_2, x_1, x_0) = R_1(0, 2, 7, 9) + R_\emptyset(3, 8, 10);$
 l) $f(x_3, x_2, x_1, x_0) = R_1(1, 4, 11, 14) + R_\emptyset(3, 6, 9, 12);$
 m) $f(x_3, x_2, x_1, x_0) = R_1(2, 4, 7, 8, 13, 14) + R_\emptyset(0, 5, 6, 10, 12, 15).$

7.3. Să se implementeze următoarele tipuri de convertoare:

- a) Gray TO Exces 3;
 b) TO Complementar Exces 3;
 c) Exces 3 TO BCD.

Indicatii

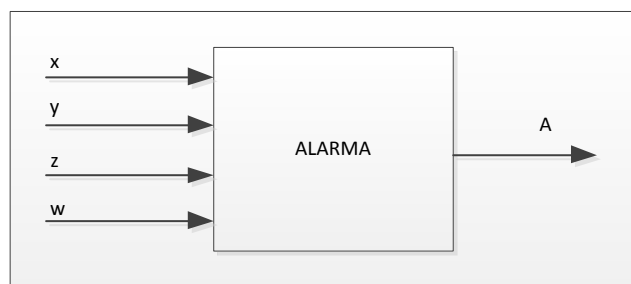
a)								b)							
x_3	x_2	x_1	x_0	y_3	y_2	y_1	y_0	x_3	x_2	x_1	x_0	y_3	y_2	y_1	y_0
0	0	0	0	0	0	1	1	0	0	0	0	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
0	0	0	1	0	1	0	0	0	0	0	1	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
0	0	1	1	0	1	0	1	0	0	1	0	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
0	0	1	0	0	1	1	0	0	0	1	1	1	1	0	0
0	1	1	0	0	1	1	1	0	1	0	0	1	0	1	1
0	1	1	1	1	0	0	0	0	1	0	1	1	0	1	0
0	1	0	1	1	0	0	1	0	1	1	0	1	0	0	1
0	1	0	0	1	0	1	0	0	1	1	1	1	0	0	0
1	1	0	0	1	0	1	1	1	0	0	0	0	1	1	1
1	1	0	1	1	1	0	0	1	0	0	1	0	1	1	0
1	1	1	1	1	1	0	1	1	0	1	0	0	1	0	1
1	1	1	0	1	1	1	0	1	0	1	1	0	1	0	0
1	0	1	0	1	1	1	1	1	1	0	0	0	0	1	1
1	0	1	1	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	1	1	0	1	0	0	1	0
1	0	0	1	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	1	1	1	0	0	0	0	1
1	0	0	0	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	1	1	1	1	0	0	0	0

c)

x_3	x_2	x_1	x_0	y_3	y_2	y_1	y_0
0	0	0	0	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
0	0	0	1	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
0	0	1	0	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
0	0	1	1	0	0	0	0
0	1	0	0	0	0	0	1
0	1	0	1	0	0	1	0
0	1	1	0	0	0	1	1
0	1	1	1	0	1	0	0
1	0	0	0	0	1	0	1
1	0	0	1	0	1	1	0
1	0	1	0	0	1	1	1
1	0	1	1	1	0	0	0
1	1	0	0	1	0	0	1
1	1	0	1	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
1	1	1	0	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
1	1	1	1	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$

7.4. Să se proiecteze folosind porți logice un sistem de alarmă. Se consideră că alarma va fi activată în unul din cazurile:

- 1) Este acționat un buton de validare și ușa este deschisă; sau
- 2) Este acționat butonul de validare, ușa este închisă, fereastra este deschisă și este trecut de ora 22:00.



Indicații

Sunt necesare patru variabile de intrare. Se notează variabilele de intrare în felul următor: x – butonul de validare; y – ușa; z – fereastra și w – dacă este trecut de ora 22:00.

8. Referințe bibliografice

- [1] Gh.Toacșe, *Introducere în microprocesoare*, Editura Științifică și Enciclopedică, București, 1996.
- [2] Manta V., Ungureanu F., *Introducere în știința sistemelor și a calculatoarelor*, Volumul I, Editura "Gh.Asachi", Iași, 2002.