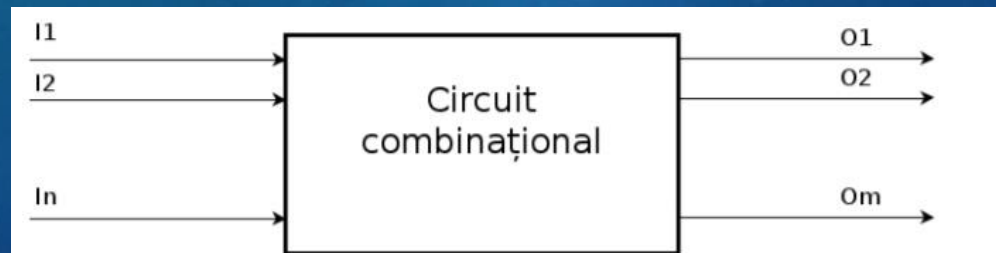


# TEMA NR.5: CIRCUITELE LOGICE COMBINAȚIONALE

1. Noțiuni generale elementele funcționale
2. Sinteza Convertoarelor de cod

**Circuit logic combinațional (CLC)** - circuit logic format din elemente logice cu  $n$  intrări și cu  $m$  ieșiri, la care combinația valorilor de la ieșire depinde doar de combinația valorilor variabilelor de la intrare în momentul dat de timp și nu depinde de combinația variabilelor de la intrare din momentele de timp anterioare.



Sinteza CLC se realizează conform următoarelor etape :

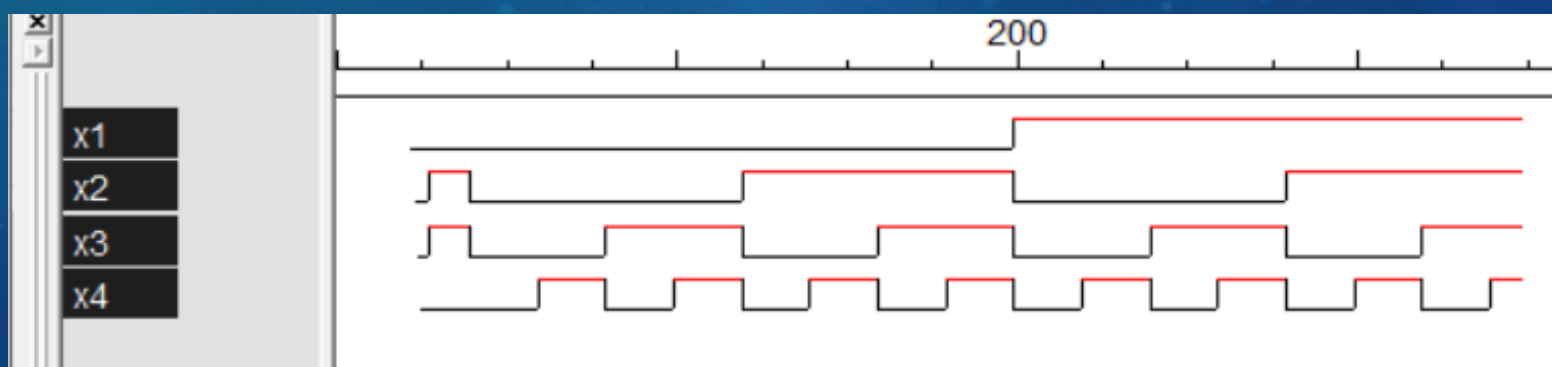
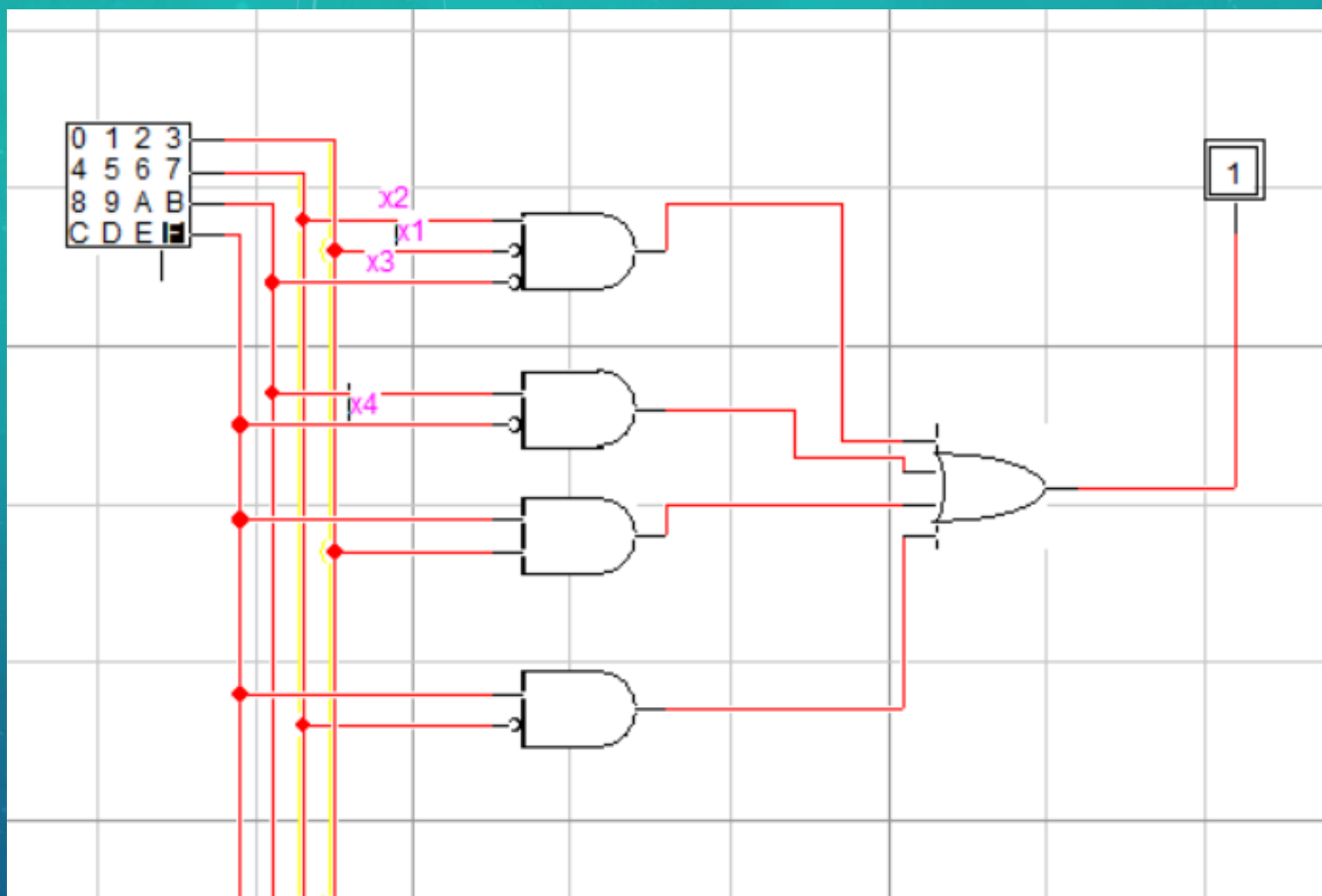
- 1) Descrierea verbală sau grafică a funcțiilor care trebuie să le îndeplinească dispozitivul respectiv;
- 2) Elaborarea tabelului de adevăr a dispozitivului sinteza cărui trebuie să fie efectuată în baza descrierii lui la etapa precedentă;
- 3) Minimizarea funcțiilor din tabelul de adevăr;
- 4) Depistarea părților comune a funcțiilor minimize pentru a fi realizate în circuit o singură dată;
- 5) Realizarea schemei dispozitivului în baza funcțiilor minimize luând în considerație părțile comune și utilizând baza de elemente necesară.

Pentru a aduce funcțiile minimizate cât mai aproape de baza elementară necesară ele se transformă în una din acele 8 forme care este cea mai apropiată de baza de elemente respective.

Exemplu:  $Y = V(1, 2, 3, 4, 5, 6, 9, 10, 11, 13, 14, 15)$

$x_1x_2$ $x_3x_4$	00	01	11	10
00		1		
01	1	1	1	1
11	1			1
10	1	1	1	1

$$y = \overline{x_1}x_2\overline{x_3} \vee x_3\overline{x_4} \vee x_1x_4 \vee \overline{x_2}x_4$$





2. **Convertoar de cod** – CLC cu  $n$  intrări și  $m$  ieșiri destinat transformării unui cod binar în altul.

Aria tipurilor de convertoare de cod este foarte largă, iar valorile  $n$  și  $m$  ale lor pot coincide, dar pot fi și diferite, în dependență de tipul de coduri de la intrarea și, respectiv, de la ieșirea convertorului de cod.

În calitate de convertoare de cod la care numărul de intrări coincide cu numărul de ieșiri pot servi cele care transformă codul direct al unui număr binar în codul lui invers, sau în cel complementar. Tot în această categorie intră convertoarele unui cod binar- zecimal în altul etc.

Convertoare de cod cu număr diferit de intrări și ieșiri sînt acelea care efectuează conversia numerelor dintr-un sistem de numerație în altul, convertoarele care transformă codul binar-zecimal de patru biți în codul pentru indicatoarele numerice de șapte segmente (șapte biți) ș.a.

Convertoare de cod cu număr diferit de intrări și ieșiri sînt acelea care efectuează conversia numerelor dintr-un sistem de numerație în altul, convertoarele care transformă codul binar-zecimal de patru biți în codul pentru indicatoarele numerice de șapte segmente (șapte biți) ș.a.

Sinteza convertoarelor de cod, indiferent de tipul lor, are loc în felul următor:

1. Elaborarea tabelului de adevăr cu următorii parametri: numărul de variabile este egal cu numărul biților codului, care se aplică la intrările convertorului, iar numărul funcțiilor cu numărul de biți ai codului, care trebuie obținut la ieșirile convertorului de cod.

*Notă: Funcțiile logice pot fi parțial determinate, dacă numărul combinațiilor codului de intrare este mai mic decât  $2^n$ .*

2. Minimizarea tuturor funcțiilor din tabelul de adevăr.
3. Depistarea conjuncțiilor comune ale formelor minimale ale tuturor funcțiilor, pentru a evita dublarea elementelor logice, care realizează părți comune ale mai multor funcții.
4. Implementarea circuitului.

# Sinteza unui convertor de cod binar-zecimal 87(-2)(-4) - 4221

Nr. crt.	8 7 (-2) (-4)				4 2 2 1			
	$x_4$	$x_3$	$x_2$	$x_1$	$f_4$	$f_3$	$f_2$	$f_1$
0	0	0	0	0	0	0	0	0
1	0	1	1	1	0	0	0	1
2	1	0	1	1	0	0	1	0
3	0	1	0	1	0	0	1	1
4	1	0	0	1	0	1	1	0
5	0	1	1	0	1	0	0	1
6	1	0	1	0	1	1	0	0
7	0	1	0	0	1	1	0	1
8	1	0	0	0	1	1	1	0
9	1	1	1	1	1	1	1	1
10	0	0	0	1	*	*	*	*
11	0	0	1	0	*	*	*	*
12	0	0	1	1	*	*	*	*
13	1	1	0	0	*	*	*	*
14	1	1	0	1	*	*	*	*
15	1	1	1	0	*	*	*	*

Diagramele Karnaugh pentru minimizarea funcțiilor  $f_4, f_3, f_2, f_1$ .



$x_4x_3$ $x_2x_1$	00	01	11	10
00		1	*	1
01	*		*	
11	*		1	
10	*	1	*	1

$f_4$

$x_4x_3$ $x_2x_1$	00	01	11	10
00		1	*	1
01	*		*	1
11	*		1	
10	*		*	1

$f_3$

$x_4x_3$ $x_2x_1$	00	01	11	10
00			*	1
01	*	1	*	1
11	*		1	1
10	*		*	

$f_2$

$x_4x_3$ $x_2x_1$	00	01	11	10
00		1	*	
01	*	1	*	
11	*	1	1	
10	*	1	*	

$f_1$



În rezultatul minimizării au fost obținute următoarele funcții logice:

$$f_4 = x_3 \bar{x}_1 \vee x_4 \bar{x}_1 \vee x_4 x_3;$$

$$f_3 = x_3 \bar{x}_2 \bar{x}_1 \vee x_4 \bar{x}_2 \vee x_4 \bar{x}_1 \vee x_4 x_3;$$

$$f_2 = x_4 \bar{x}_2 \vee \bar{x}_2 x_1 \vee x_4 x_1;$$

$$f_1 = x_3.$$

$$z_1 = x_4 \bar{x}_1 \vee x_4 x_3;$$

$$z_2 = x_4 \bar{x}_2.$$

$$f_4 = x_3 \bar{x}_1 \vee z_1;$$

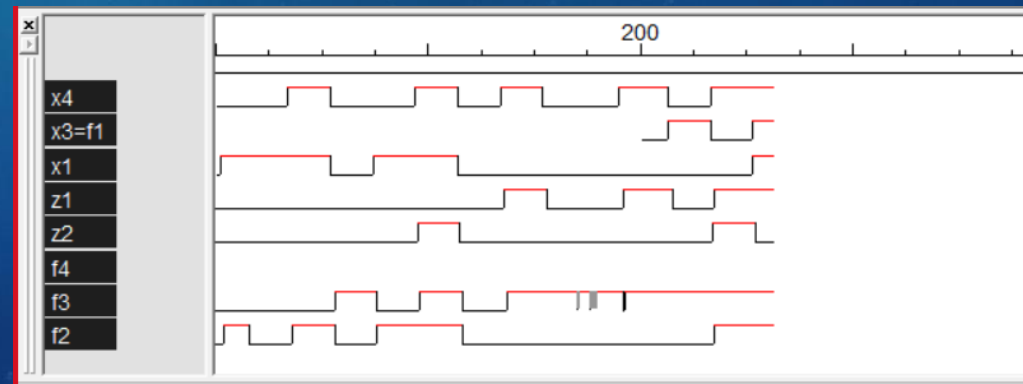
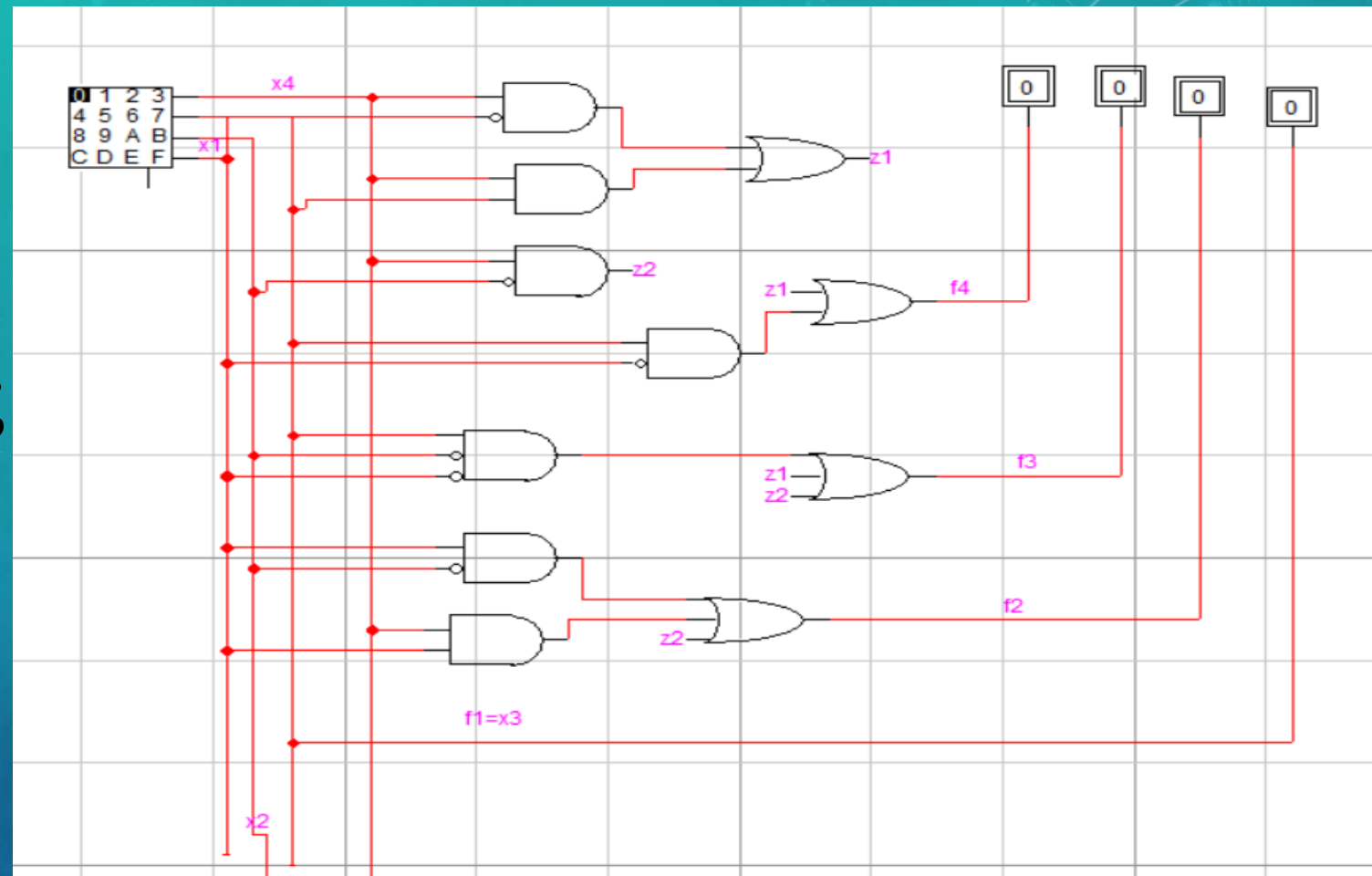
$$f_3 = x_3 \bar{x}_2 \bar{x}_1 \vee z_1 \vee z_2;$$

$$f_2 = \bar{x}_2 x_1 \vee x_4 x_1 \vee z_2;$$

$$f_1 = x_3.$$

$$z_1 = x_4 \bar{x}_1 \vee x_4 x_3;$$

$$z_2 = x_4 \bar{x}_2.$$



# TEMA NR.6:

## SINTEZA DECODIFICATOARELOR SI CODIFICATOARELOR

1. **Decodificatorul** – este elementul funcțional combinațional destinat decodificării cuvintelor binare aplicate la intrarea lui. Un decodificator complet cu  $n$  intrări va avea  $2^n$  ieșiri unde  $n$  este numărul biților cuvântului de intrare a decodicatorului.

Decodificarea are loc în felul următor: Dacă la intrare se aplică una din oarecare  $2^n$  combinații atunci la ieșire vom avea semnalul 1 doar la ieșirea care corespunde acestei combinații, la ieșirea celorlalte  $2^n - 1$  ieșiri vom avea zero, cu alte cuvinte pentru orice combinație de la intrarea decodicatorului la ieșirea vom avea o singură unitate iar la restul ieșirilor va fi zero.

Un asemenea cod se mai numește cod unitar.

Nr. ord	x <sub>3</sub>	x <sub>2</sub>	x <sub>1</sub>	y <sub>0</sub>	y <sub>1</sub>	y <sub>2</sub>	y <sub>3</sub>	y <sub>4</sub>	y <sub>5</sub>	y <sub>6</sub>	y <sub>7</sub>
0	0	0	0	1	0	0	0	0	0	0	0
1	0	0	1	0	1	0	0	0	0	0	0
2	0	1	0	0	0	1	0	0	0	0	0
3	0	1	1	0	0	0	1	0	0	0	0
4	1	0	0	0	0	0	0	1	0	0	0
5	1	0	1	0	0	0	0	0	1	0	0
6	1	1	0	0	0	0	0	0	0	1	0
7	1	1	1	0	0	0	0	0	0	0	1

$$y_0 = \overline{x_3} \overline{x_2} \overline{x_1}$$

$$y_1 = \overline{x_3} \overline{x_2} x_1$$

$$y_2 = \overline{x_3} x_2 \overline{x_1}$$

$$y_3 = \overline{x_3} x_2 x_1$$

$$y_4 = x_3 \overline{x_2} \overline{x_1}$$

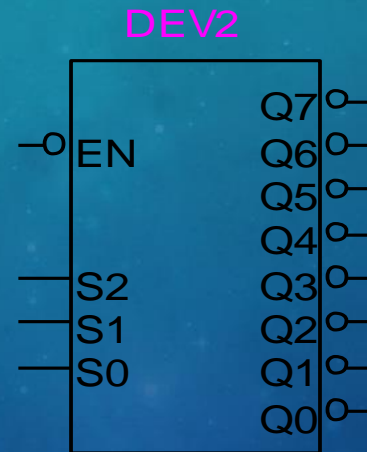
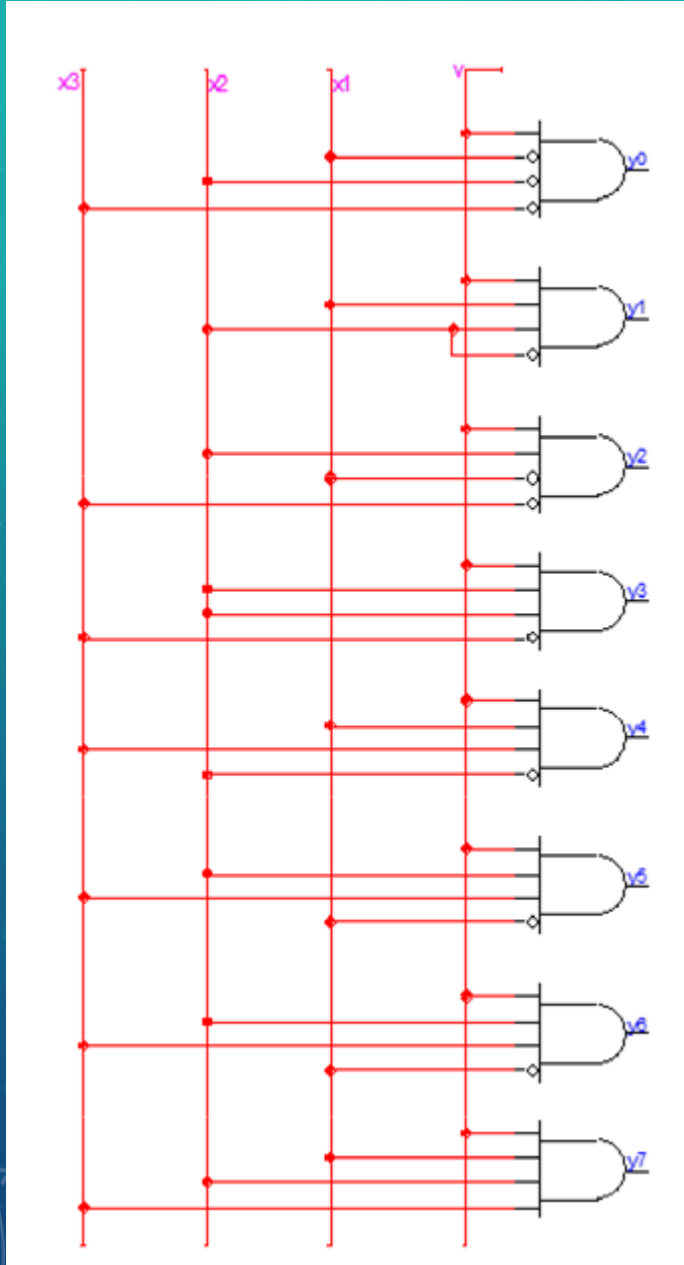
$$y_5 = x_3 \overline{x_2} x_1$$

$$y_6 = x_3 x_2 \overline{x_1}$$

$$y_7 = x_3 x_2 x_1$$

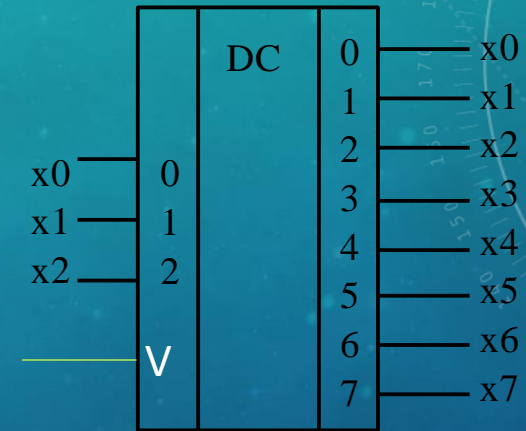


# Circuitul logic si reprezentarea grafică a decodicatorului:



V - Intrarea de validare (E)

Reprezentarea grafică



Întrarea de validare a decodicatorului se folosește sau pentru validarea funcțiilor decodicatorului sau pentru extinderea funcțiilor pe care el le îndeplinește.

O funcție de extindere: Un grup de decodificatoare în cazul de față 4 dec se utilizează pentru decodificarea celui mai puțin semnificativi biți al cod care trebuie decodificat. Numărul acestor biți depinde de numărul de intrări ai decodicatorului utilizat. Celelalte ranguri ale codului care trebuie decodificate se utilizează cu ajutorul unui alt cod la selectarea decodificatoarelor din primul grup.

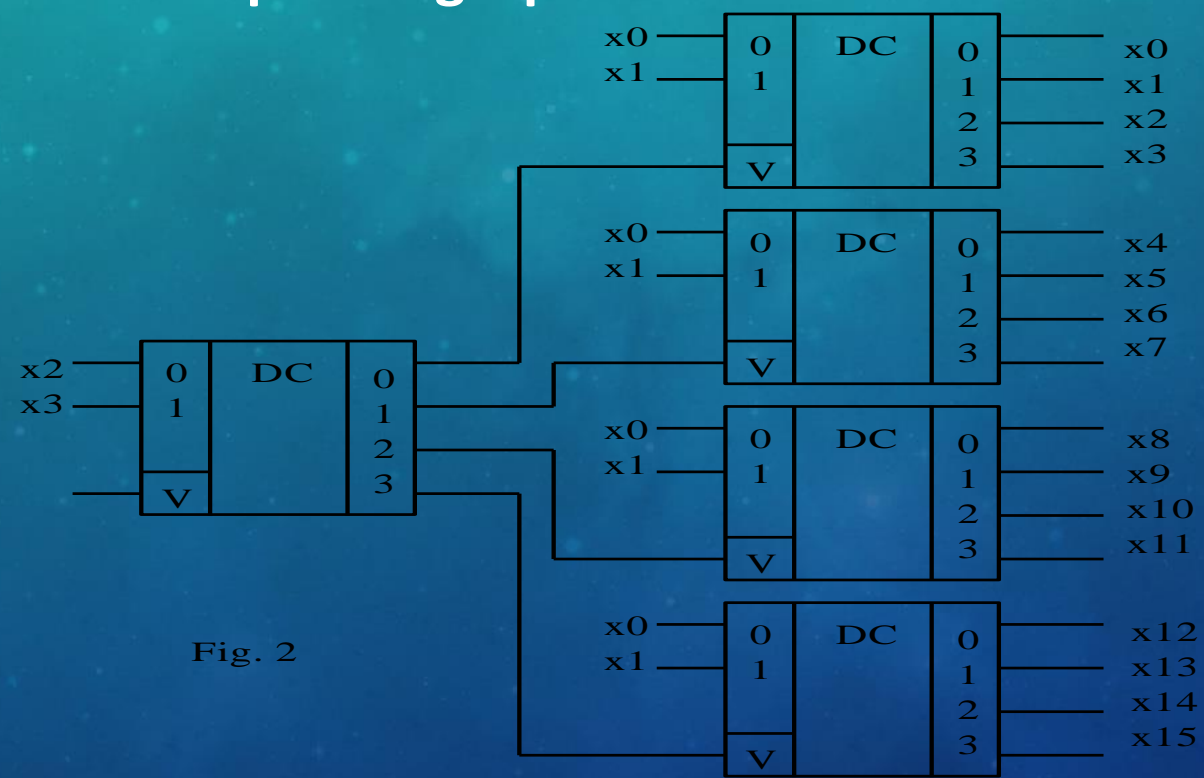


Fig. 2

**Decodificator incomplet** – De multe ori în calculatoare nu se folosesc toate  $2^n$  combinații a cuvintelor de  $n$  biți Și-n acest caz nu este neapărată necesar de a avea un decodificator cu  $n$  intrări și  $2^n$  eșiri. Decodificatoarele la care numărul de ieșiri  $n$  este  $<2^n$  și mai  $>2^n-1$  se numesc decodificator incomplete.

Sinteza unui decodificator incomplet poate fi redusă la minimizarea a  $m$  funcții particular determinate de  $n$  variabile fiecare unde  $n$  este numărul intrărilor decodicatorului iar  $m$  este numărul ieșirilor decodicatorului ( $m < 2^n$ ).

Aceste funcții au următorul specific: ele sunt egale cu 1 doar pentru o singură combinație și anume, pentru acea combinație pe care ieșirea respectivă trebuie să o decodifice și nu sunt determinate pentru  $2^n-m$  combinații și anume acele combinații care nu sunt utilizate în decodicatorul respectiv. În așa fel structura decodicatorului incomplet va avea mai mici cheltuieli de aparataj decât în cazul când sinteza lui se va efectua doar prin eliminarea elementelor logice care ar decodifica combinațiile neutilizate.



Cifre zecimal	Codul binar zecimal													
	5	3	2	-1										
	X <sub>4</sub>	X <sub>3</sub>	X <sub>2</sub>	X <sub>1</sub>	y <sub>0</sub>	y <sub>1</sub>	y <sub>2</sub>	y <sub>3</sub>	y <sub>4</sub>	y <sub>5</sub>	y <sub>6</sub>	y <sub>7</sub>	y <sub>8</sub>	y <sub>9</sub>
0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
1	0	0	1	1	0	1	0	0	0	0	0	0	0	0
2	0	0	1	0	0	0	1	0	0	0	0	0	0	0
3	0	1	0	0	0	0	0	1	0	0	0	0	0	0
4	0	1	1	1	0	0	0	0	1	0	0	0	0	0
5	1	0	0	0	0	0	0	0	0	1	0	0	0	0
6	1	0	1	1	0	0	0	0	0	0	1	0	0	0
7	1	1	0	1	0	0	0	0	0	0	0	1	0	0
8	1	1	0	0	0	0	0	0	0	0	0	0	1	0
9	1	1	1	1	0	0	0	0	0	0	0	0	0	1
10	0	0	0	1	*	*	*	*	*	*	*	*	*	*
11	0	1	0	1	*	*	*	*	*	*	*	*	*	*
12	0	1	1	0	*	*	*	*	*	*	*	*	*	*
13	1	0	0	1	*	*	*	*	*	*	*	*	*	*
14	1	0	1	0	*	*	*	*	*	*	*	*	*	*
15	1	1	1	0	*	*	*	*	*	*	*	*	*	*

<div>X<sub>4</sub>X<sub>3</sub></div> <div>X<sub>2</sub>X<sub>1</sub></div>	00	01	11	10
00	y <sub>0</sub>	y <sub>3</sub>	y <sub>8</sub>	y <sub>5</sub>
01	*	*	y <sub>7</sub>	*
11	y <sub>1</sub>	y <sub>4</sub>	y <sub>9</sub>	y <sub>6</sub>
10	y <sub>2</sub>	*	*	*

$$y_0 = \overline{x_4 x_3 x_2}$$

$$y_1 = \overline{x_4 x_3} x_1$$

$$y_2 = x_2 \overline{x_1}$$

$$y_3 = \overline{x_4 x_3} x_2$$

$$y_4 = \overline{x_4 x_3} x_2$$

$$y_5 = \overline{x_4 x_3} x_2$$

$$y_6 = \overline{x_4 x_3} x_2$$

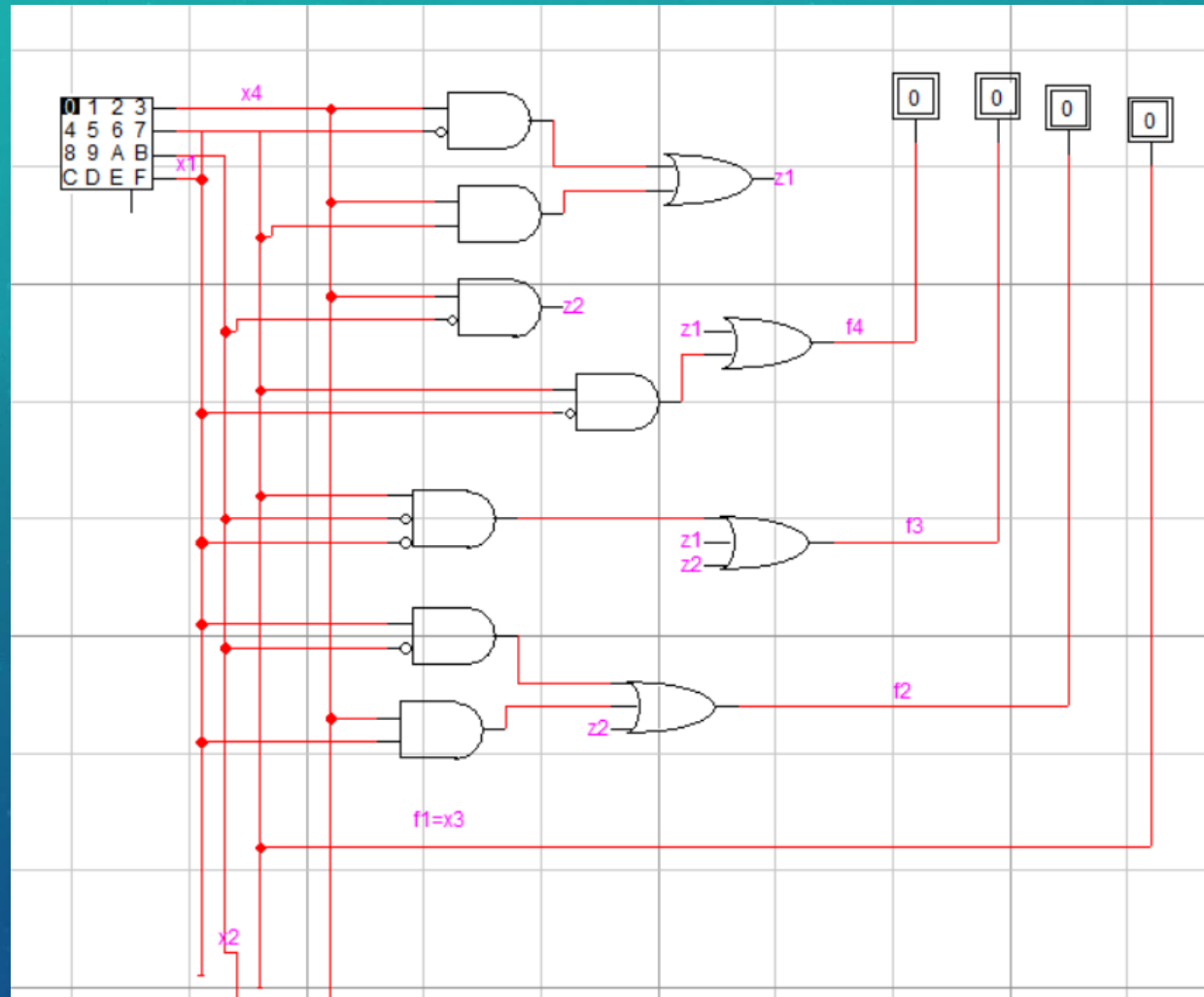
$$y_7 = \overline{x_2} x_1$$

$$y_8 = \overline{x_4 x_3} x_1$$

$$y_9 = \overline{x_4 x_3} x_2$$



# Circuitul logic:



**Codificator** - un element funcțional combinațional care este destinat transformării unui semnal unitar de intrare într-un cod binar la ieșire. Codificatorul codifică semnalele de intrare. Codificarea se utilizează în calculatoarele numerice atât în unitățile centrale cât și în cele periferice.

Sinteza oricărui codificator are specificul său care constă în faptul că variabilele de intrare reprezintă un cod unitar (o singură unitate) iar variabilele de ieșire sunt variabilele logice obișnuite, adică ele pot avea în orice poziție valoarea 0 sau 1. Acest specific tabel de adevăr al Codificatorului practic exclude procedura de minimizare ca etapă a sintezei iar valorile funcțiilor se stabilesc ca disjuncția valorilor variabilelor de intrare unde ele sunt egale cu 1.

Cifra zecimala	Intrarile										Iesirile			
											4	4	1	-2
	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	f4	f3	f2	f1
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0	0	0	0	0	1	0
2	0	0	1	0	0	0	0	0	0	0	0	1	0	1
3	0	0	0	1	0	0	0	0	0	0	0	1	1	1
4	0	0	0	0	1	0	0	0	0	0	0	1	0	0
5	0	0	0	0	0	1	0	0	0	0	1	0	1	0
6	0	0	0	0	0	0	1	0	0	0	1	1	0	1
7	0	0	0	0	0	0	0	1	0	0	1	1	1	1
8	0	0	0	0	0	0	0	0	1	0	1	1	0	0
9	0	0	0	0	0	0	0	0	0	1	1	1	1	0

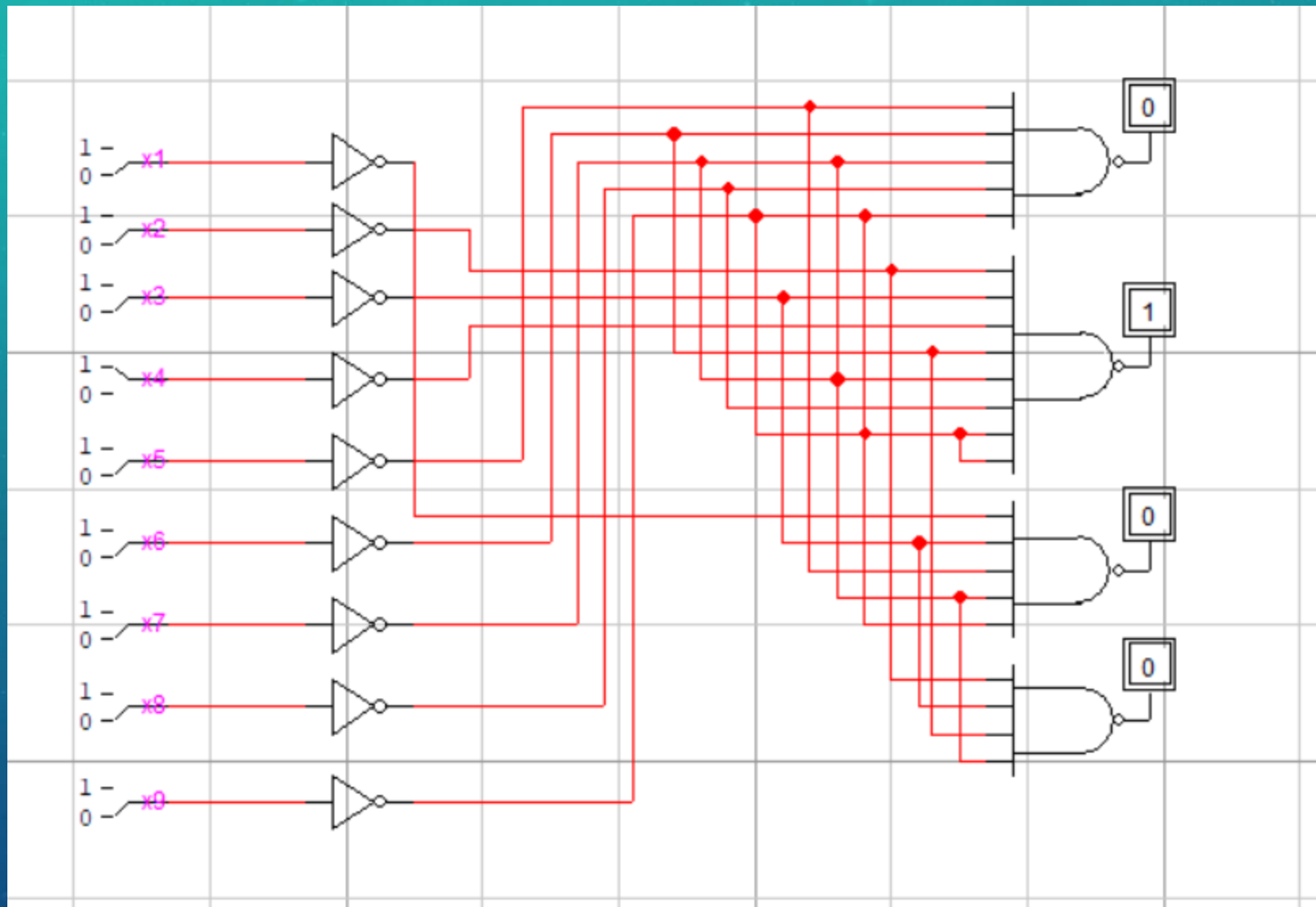
$$f4 = x5 \vee x6 \vee x7 \vee x8 \vee x9$$

$$f3 = x2 \vee x3 \vee x4 \vee x6 \vee x7 \vee x8 \vee x9$$

$$f2 = x1 \vee x3 \vee x5 \vee x7 \vee x9$$

$$f1 = x2 \vee x3 \vee x6 \vee x7$$

# Circuitul logic în setul de elemente SI-NU:





**Vă mulțumesc  
pentru atenție!**



The background of the image is a light gray surface covered with numerous 3D question marks. These question marks are rendered in a light gray color with soft shadows, giving them a three-dimensional appearance. They are scattered across the frame in various orientations and positions, some appearing larger and more prominent than others. The overall effect is one of curiosity and inquiry.

ÎNTREBĂRI