Logică computațională
Curs 11

Lector dr. Pop Andreea-Diana

Algebre booleene

- introduse de George Boole (1815-1864)
- stau la baza definirii funcţiilor booleene
  - care sunt utilizate în realizarea circuitelor logice

Definirea axiomatică (1)

O ***algebră booleană*** este o structură $(A, \wedge, \vee, \bar{\ }, 0, 1)$, unde:

- $|A| \geq 2$, A conţinând cel puţin 2 elemente diferite, 0 şi 1, $0 \neq 1$
- $\wedge, \vee$ sunt operaţii binare
- $\bar{\ }$ este operator unar
- există elementul unic 0 – elementul zero, cu proprietăţile:

  $x \wedge 0 = 0 \wedge x = 0$ şi $x \vee 0 = 0 \vee x = x$, $\forall x \in A$
- există elementul unic 1 – elementul unitate, cu proprietăţile:

  $x \wedge 1 = 1 \wedge x = x$ şi $x \vee 1 = 1 \vee x = 1$, $\forall x \in A$
- elementul zero, 0 şi cel unitate, 1 sunt primul respectiv ultimul element, iar $\bar{x}$ este complementul lui x:

  $x \wedge \bar{x} = 0$ şi $x \vee \bar{x} = 1$, $\forall x \in A$
- dubla negaţie:

  $\bar{\bar{x}} = x$, $\forall x \in A$

Definirea axiomatică (2)

- operaţiile $\wedge$, $\vee$ sunt comutative:

  $x \wedge y = y \wedge x$ şi $x \vee y = y \vee x$, $\forall x, y \in A$
- operaţiile $\wedge$, $\vee$ sunt asociative:

  $x \wedge (y \wedge z) = (x \wedge y) \wedge z$ $(= x \wedge y \wedge z)$ şi

  $x \vee (y \vee z) = (x \vee y) \vee z$ $(= x \vee y \vee z)$, $\forall x, y, z \in A$
- au loc proprietăţile de distributivitate ale operatorilor $\wedge$ şi $\vee$:

  $x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$ şi

  $x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z)$, $\forall x, y, z \in A$
- are loc proprietatea de idempotenţă pentru ambele operaţii:

  $x \wedge x = x$ şi $x \vee x = x$, $\forall x \in A$
- au loc legile lui De Morgan:

  $\overline{x \wedge y} = \overline{x} \vee \overline{y}$ şi $\overline{x \vee y} = \overline{x} \wedge \overline{y}$, $\forall x, y \in A$
- au loc proprietăţile de absorbţie:

  $x \wedge (x \vee y) = x$ şi $x \vee (x \wedge y) = x$, $\forall x, y \in A$

Observații

- Într-o algebră booleană are loc principiul dualităţii:

  „Pentru orice egalitate între două expresii booleene $U = V$, există o nouă egalitate $U' = V'$, obţinută prin interschimbarea operaţiilor $\wedge$, $\vee$ şi a elementelor: 0, 1”.
- majoritatea axiomelor algebrei booleene, sunt perechi de axiome duale

Algebra booleană binară

$\boldsymbol{B} = (B_2=\{0,1\}, \wedge, \vee, \bar{\ }, 0, 1)$

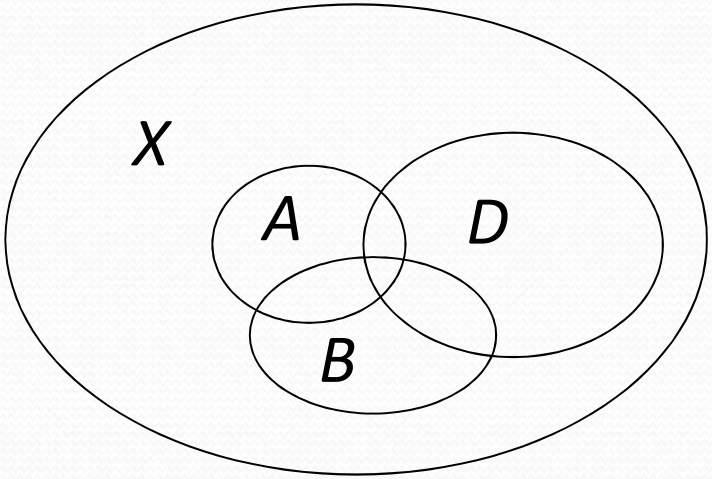
$\vee$	0	1
0	0	1
1	1	1

$\wedge$	0	1
0	0	0
1	0	1

x	$\bar{x}$
0	1
1	0

Alte exemple de algebre booleene

- $(F_P, \wedge, \vee, \neg, \mathrm{F}, \mathrm{T})$
- $(\mathscr{P}(X), \cap, \cup, C, \varnothing, X)$



Problemă practică

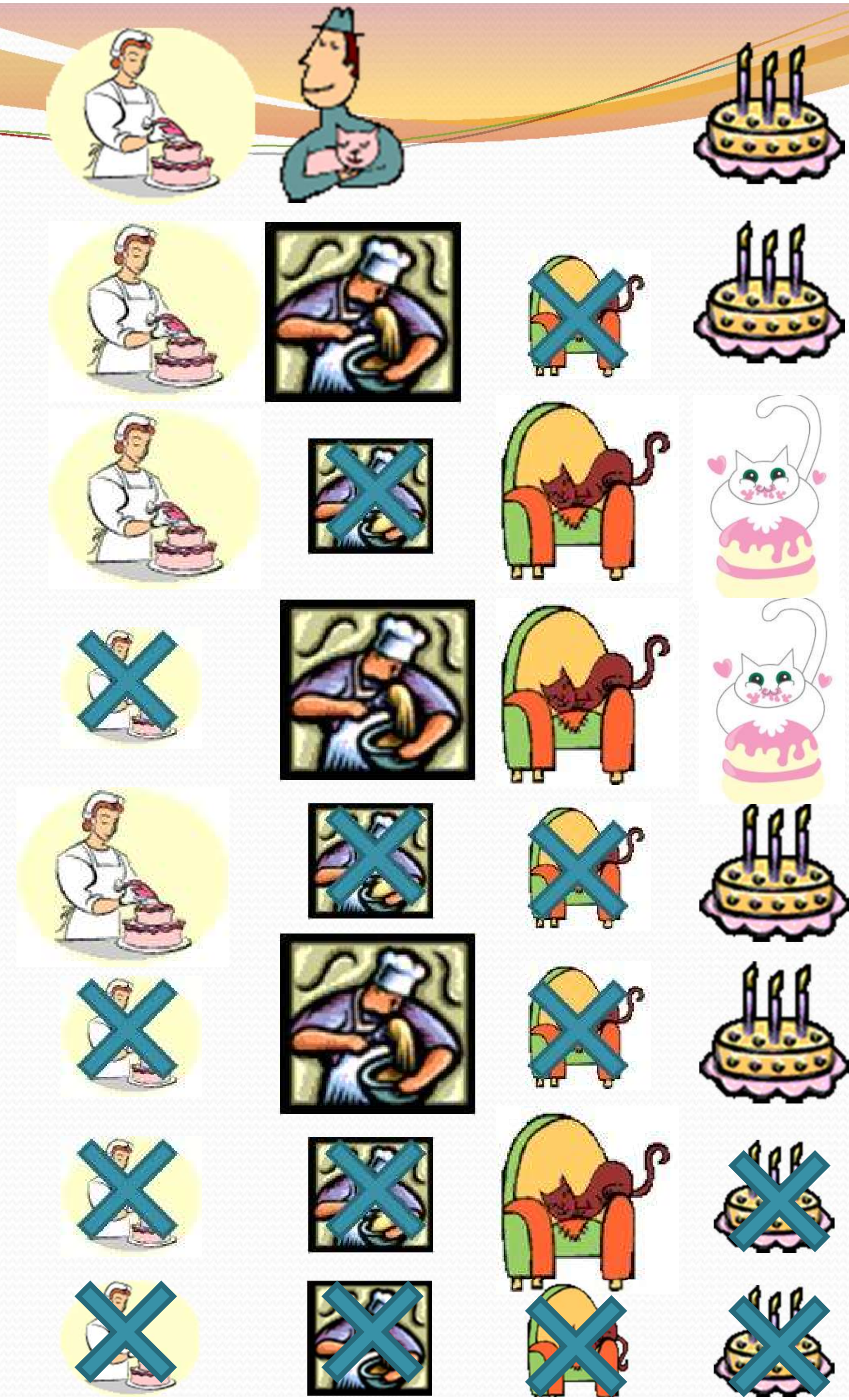
- Cine face tortul?

Soluțiile posibile

$x, \quad y, \quad z$

$f(x, y, z) =$ 



Funcții booleene

Fie $\boldsymbol{B} = (B_2, \wedge, \vee, \bar{\ }, 0, 1)$ algebra booleană binară, $B_2 = \{0, 1\}$ şi $n \in \boldsymbol{N}^*$. O *funcţie booleană de n variabile* este o funcţie definită recursiv astfel:

1. Funcţia *proiecţie*: $P_i : B_2^n \to B_2$, $P_i(x_1, \ldots, x_i, \ldots, x_n) = x_i$, care păstrează doar variabila x_i, este o funcţie booleană.
2. Dacă avem două funcţii booleene $f, g : B_2^n \to B_2$, atunci $f \wedge g, f \vee g, \overline{f}$ sunt funcţii booleene, unde:
   $(f \wedge g)(x_1, \ldots, x_n) = f(x_1, \ldots, x_n) \wedge g(x_1, \ldots, x_n)$
   $(f \vee g)(x_1, \ldots, x_n) = f(x_1, \ldots, x_n) \vee g(x_1, \ldots, x_n)$
   $\overline{f}(x_1, \ldots, x_n) = \overline{f(x_1, \ldots, x_n)}$

- Orice funcţie booleană este obţinută prin aplicarea de un număr finit de ori a regulilor 1 şi 2 de mai sus.

Toate funcțiile booleene definite pe $\boldsymbol{B_2}^{\boldsymbol{n}} \rightarrow \boldsymbol{B_2}$ pentru $n=1$

x	$f_0(x)$	$f_1(x)$	$f_2(x)$	$f_3(x)$
0	0	0	1	1
1	0	1	0	1

Teoreme

- $\forall\ n \in \boldsymbol{N}^*$, există 2^{2^n} funcţii booleene de n variabile.

- Mulţimea tuturor funcţiilor booleene de $n \in \boldsymbol{N}^*$ variabile formează o algebră booleană: $(\mathbf{FB}(\boldsymbol{n}), \wedge, \vee, \overline{\ }, f_0, f_{2^{2^n}-1})$, unde $f_0(x_1, x_2, \ldots, x_n) = 0$ şi $f_{2^{2^n}-1}(x_1, \ldots, x_n) = 1$ funcţiile constante 0, respectiv 1.

Notații

- $x^{\alpha} = \begin{cases} x, \text{dacă } \alpha = 1 \\ \overline{x}, \text{dacă } \alpha = 0 \end{cases}, x \in \{0, 1\}$

- Pentru $x,\ \alpha \in \{0, 1\}$, au loc: $x^0 = \overline{x}$, $x^1 = x$ şi

$$0^0 = \overline{0} = 1;\ 0^1 = 0;$$
$$1^0 = \overline{1} = 0;\ 1^1 = 1$$

- Astfel, se obţine: $x^{\alpha} = \begin{cases} 1, \text{dacă } x = \alpha \\ 0, \text{dacă } x \neq \alpha \end{cases}, x,\ \alpha \in \{0, 1\}$

Formele canonice ale funcțiilor booleene (1)

O funcţie booleană $f:(B_2)^n \to B_2$, $n \in \boldsymbol{N}^*$ poate fi transformată în cele două forme echivalente:

- *forma canonică disjunctivă* (FCD):

$$(1)\ f(x_1,\ldots,x_n) = \bigvee_{(\alpha_1,\ldots,\alpha_n)\in(B_2)^n} \left(f(\alpha_1,\ldots,\alpha_n) \wedge x_1^{\alpha_1} \wedge \ldots \wedge x_n^{\alpha_n}\right)$$

- *forma canonică conjunctivă* (FCC):

$$(2)\ f(x_1,\ldots,x_n) = \bigwedge_{(\alpha_1,\ldots,\alpha_n)\in(B_2)^n} \left(f(\alpha_1,\ldots,\alpha_n) \vee x_1^{\overline{\alpha_1}} \vee \ldots \vee x_n^{\overline{\alpha_n}}\right)$$

Formele canonice ale funcțiilor booleene (2)

O funcţie booleană $f:(B_2)^n \to B_2$, $n \in \boldsymbol{N}^*$ este unic determinată de valorile sale $f(\alpha_1, \alpha_2, \ldots, \alpha_n)$, unde $(\alpha_1, \alpha_2, \ldots, \alpha_n) \in (B_2)^n$:

- forma canonică disjunctivă:

$$(1) \Leftrightarrow (1') \; f(x_1,\ldots,x_n) = \bigvee_{(\alpha_1,\ldots,\alpha_n)\in(B_2)^n \text{ şi } f(\alpha_1,\ldots,\alpha_n)=1} (x_1^{\alpha_1} \wedge \ldots \wedge x_n^{\alpha_n})$$

- forma canonică conjunctivă:

$$(2) \Leftrightarrow (2') \; f(x_1,\ldots,x_n) = \bigwedge_{(\alpha_1,\ldots,\alpha_n)\in(B_2)^n \text{ şi } f(\alpha_1,\ldots,\alpha_n)=0} (x_1^{\overline{\alpha_1}} \vee \ldots \vee x_n^{\overline{\alpha_n}})$$

Observație

- Forma canonică *disjunctivă* este recomandată în caz contrar, când există un *număr mare de zerouri* ale funcţiei şi un număr mic de valori 1.
- Forma canonică *conjunctivă* este utilă când există un număr mic de zerouri şi un *număr mare de valori* 1 (realizări).

- Funcţia booleană f_0 nu poate fi scrisă în forma canonică disjunctivă, deoarece nu ia valoarea 1 pentru niciun argument.
- Funcţia booleană $f_{2^{2^n}-1}$ nu poate fi scrisă în forma canonică conjunctivă, deoarece nu ia valoarea 0 pentru niciun argument.

Definiție

Fie $f:(B_2)^n \to B_2$, $n \in \boldsymbol{N}^*$ o funcţie booleană cu n variabile.

- O conjuncţie de variabile se numeşte ***monom***.
- Un monom care conţine toate cele n variabile se numeşte ***monom canonic*** sau ***minterm*** de n variabile. Are forma:

  $x_1^{\alpha_1} \wedge \ldots \wedge x_n^{\alpha_n}, \alpha_i \in B_2$

- Disjuncţia care conţine toate cele n variabile, având forma:

  $x_1^{\alpha_1} \vee \ldots \vee x_n^{\alpha_n}, \alpha_i \in B_2$ se numeşte ***maxterm*** de n variabile.

Proprietăți

- $\forall n \in \mathbf{N}^*$, există exact 2^n maxtermi, notaţi cu $M_0, M_1, ..., M_{2^n - 1}$ şi exact 2^n mintermi, notaţi cu $m_0, m_1, ..., m_{2^n - 1}$.
- Un *maxterm* este o funcţie booleană care ia valoarea 0 doar pentru un argument.
- Un *minterm* este o funcţie booleană care ia valoarea 1 doar pentru un argument.

Observație

- Indicele unui minterm de *n* variabile este obţinut prin conversia în zecimal a numărului binar format cu cifrele ce reprezintă puterile celor *n* variabile ale expresiei acestuia.
- Indicele unui maxterm de *n* variabile este obţinut prin conversia în zecimal a numărului binar, format cu dualele cifrelor ce reprezintă puterile celor *n* variabile ale expresiei acestuia.

Propoziție

- Conjuncţia a doi mintermi distincţi este 0:

$$m_i \wedge m_j = 0, \forall i \neq j, \quad i, j = 0, \ldots, 2^{n-1}$$

- Disjuncţia a doi maxtermi distincţi este 1:

$$M_i \vee M_j = 1, \forall i \neq j, \quad i, j = 0, \ldots, 2^{n-1}$$

- Un minterm şi un maxterm cu acelaşi indice sunt funcţii duale.

$$M_i = \overline{m_i}, \overline{M}_i = m_i, \forall i = 0, \ldots, 2^{n-1}$$

Observație

- Forma canonică conjunctivă, *FCC*, este conjuncţia maxtermilor corespunzători argumentelor pentru care funcţia ia valoarea 0.

- Forma canonică disjunctivă, *FCD*, este disjuncţia mintermilor corespunzători argumentelor pentru care funcţia ia valoarea 1.

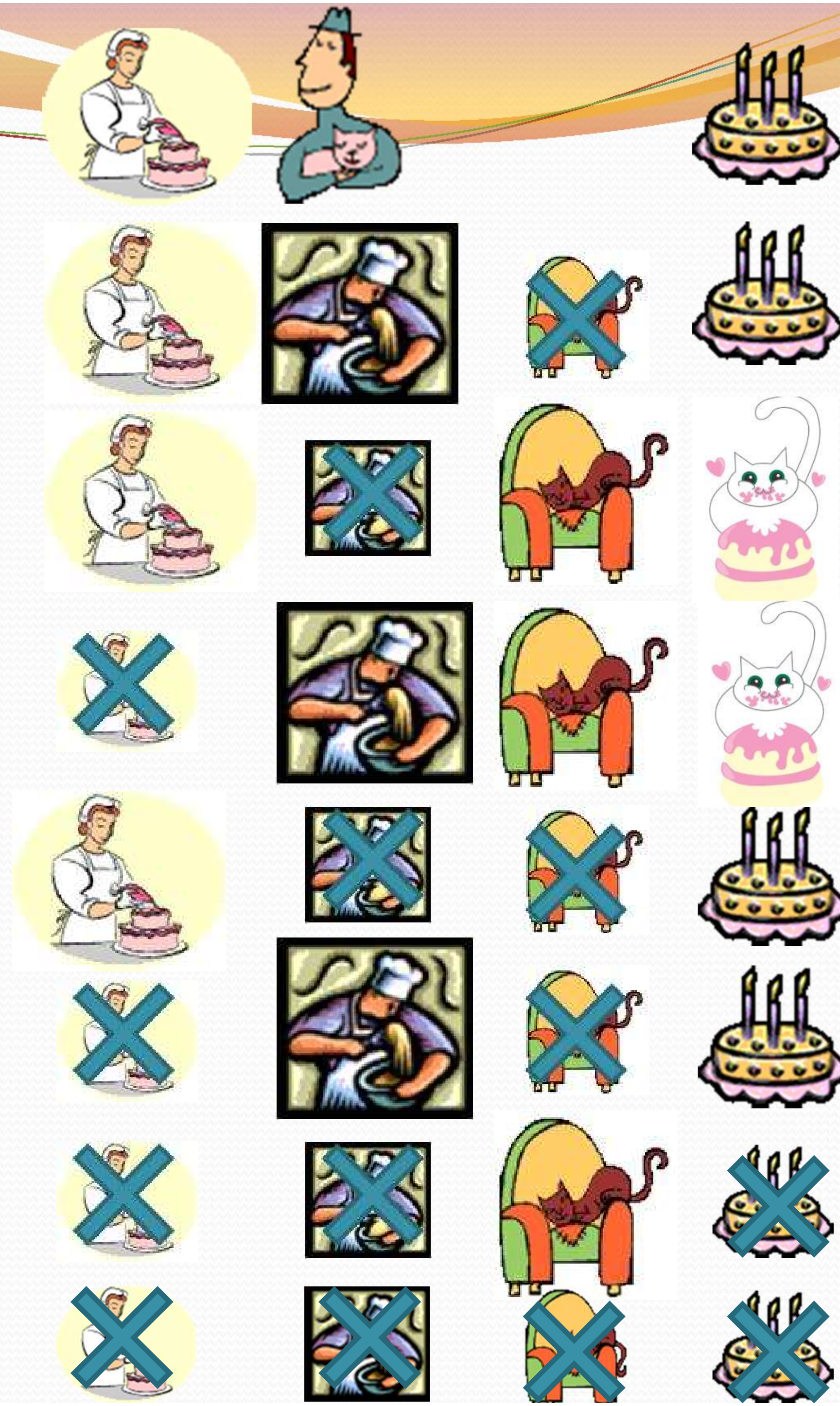
Soluțiile posibile



$x, \qquad y, \qquad z$

$f(x, y, z) =$ 

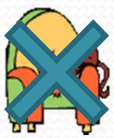
- Putem să le „condensăm”?



Suportul funcției

- *suportul* funcţiei booleene f 

$$S_f = \{ (x_1, x_2, \ldots, x_n) \in B_2^{\,n} \mid f(x_1, x_2, \ldots, x_n) = 1 \}$$

    $(1, 1, 0) \in S_f$

    $(1, 0, 1) \notin S_f$

Relația mai mic sau egal

- Relaţia *mai mic sau egal* $f \leq g \Leftrightarrow S_f \subseteq S_g$

$m = x \wedge y \wedge \overline{z}$ și $m' = x \wedge \overline{z}$

$$S_m = \{(1, 1, 0)\} \quad \subset \quad S_{m'} = \{(1, 1, 0), (1, 0, 0)\}$$

$$m \leq m'$$

Factorizarea

- *monoame adiacente* (*vecine*) dacă ele diferă doar prin puterea (semnul: negaţie sau nu) variabilei cu indicele „k_i”:

$$m = x_{k_1}^{\alpha_{k_1}} \wedge .... \wedge x_{k_j}^{\alpha_{k_j}} \wedge x_{k_i} \wedge x_{k_l}^{\alpha_{k_l}} \wedge ... \wedge x_{k_s}^{\alpha_{k_s}}$$

$$m' = x_{k_1}^{\alpha_{k_1}} \wedge .... \wedge x_{k_j}^{\alpha_{k_j}} \wedge \overline{x_{k_i}} \wedge x_{k_l}^{\alpha_{k_l}} \wedge ... \wedge x_{k_s}^{\alpha_{k_s}}$$

- *factorizarea monoamelor m* şi *m’* este operaţia prin care se obţine, eliminând variabila cu indicele „k_i”, monomul

$$m \vee m' = x_{k_1}^{\alpha_{k_1}} \wedge .... \wedge x_{k_j}^{\alpha_{k_j}} \wedge x_{k_l}^{\alpha_{k_l}} \wedge ... \wedge x_{k_s}^{\alpha_{k_s}}$$

mai mare decât *m* şi *m’*.

Simplificarea

- A *simplifica* o funcţie booleană, înseamnă a obţine o formă echivalentă a sa cu un număr cât mai mic de apariţii de variabile.

- Dacă se pornește de la FCD(f), prin factorizări repetate, într-un număr finit de paşi se poate obţine o formă simplificată a funcţiei.

Mulțimea monoamelor maximale

- Mulţimea $M(f)$ se numeşte *mulţimea monoamelor maximale* ale funcţiei booleene $f : B_2{}^n \to B_2$ dacă:

$$\forall m \in M(f), m \in FB(n), m \leq f;$$

$$\forall m \in M(f), \nexists m' \in FB(n), \text{astfel încât } m < m' \leq f$$

Mulțimea monoamelor centrale

- Mulţimea $C(f)$ se numeşte *mulţimea monoamelor centrale* ale funcţiei booleene $f : {B_2}^n \to B_2$ dacă:

$$\forall m \in C(f), m \in M(f)$$

$$\forall m \in C(f), \text{nu are loc } m \le \vee m', \text{ unde } m' \in M(f) - \{m\}$$

Algoritmul de simplificare a funcţiilor booleene

Date de intrare: *f – o funcţie booleană în formă canonică disjunctivă*

Date de ieşire: $f'_1, f'_2, \ldots, f'_k$ - variantele simplificate ale funcţiei f

Se determină $M(f)$ şi $C(f)$.

dacă $M(f) = C(f)$

atunci $f'_1 = \bigvee_{m \in M(f)} m$ STOP1 // caz 1--- soluţie unică

altfel

dacă $C(f) \neq \emptyset$

atunci $g \overset{not.}{=} \bigvee_{m \in C(f)} m$

$f'_i = g \vee h_i$, $i = \overline{1,k}$, unde h_i este disjuncţia unui număr cât mai mic de monoame maximale astfel încât $S_{h_i} = S_f - S_g$

STOP2 // caz 2 --- k soluţii

altfel // nu există monoame centrale

$f'_i = h_i$, $i = \overline{1,k}$, unde h_i este disjuncţia unui număr cât mai mic de monoame maximale astfel încât $S_{h_i} = S_f$

STOP3 // caz 3 --- k soluţii

sf_dacă

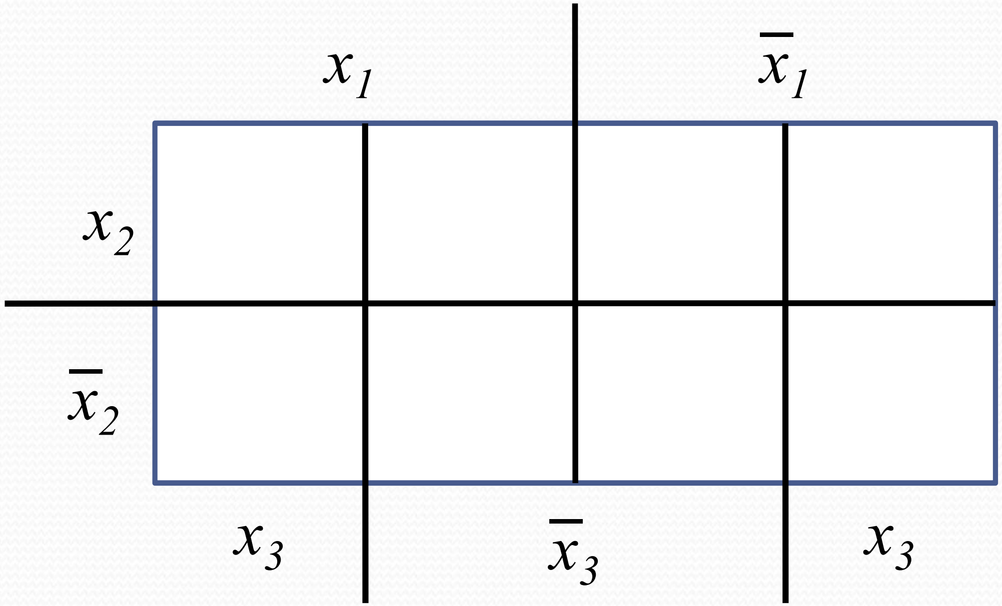
sf_dacă

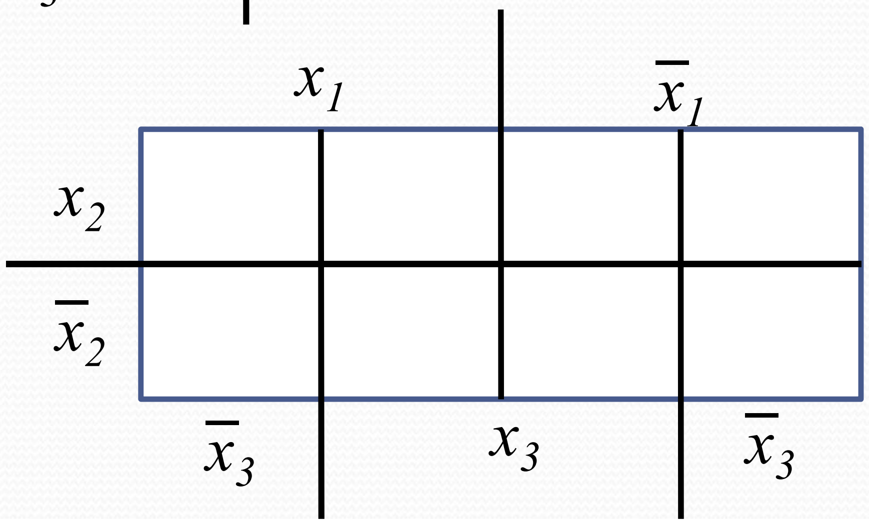
Sf_algoritm

Metoda diagramelor Veitch

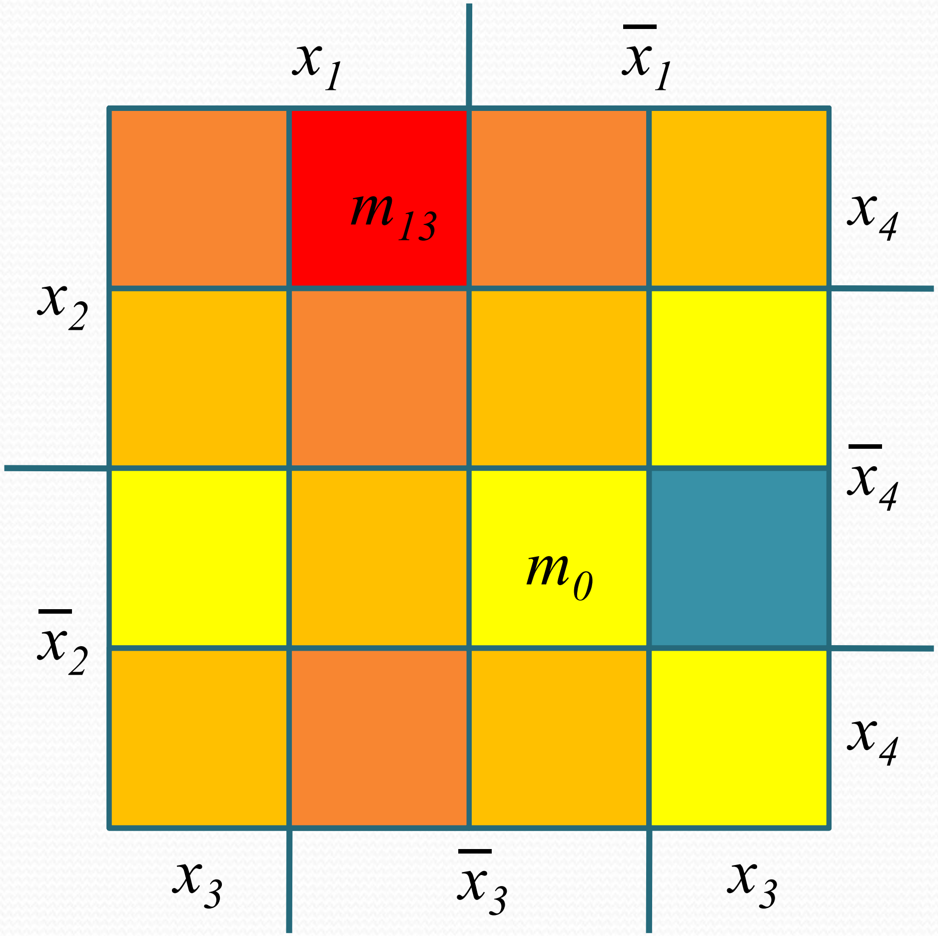
- Această metodă se bazează pe reprezentarea grafică, sub forma unei diagrame, a suportului funcţiei.
- Este foarte utilă pentru funcţii cu un număr mic de variabile: 2, 3 sau 4.

Construirea diagramei Veitch





Completarea diagramei



$$m_{13} = x_1 x_2 \bar{x}_3 x_4$$

Exercițiu

$$f(x_1, x_2, x_3, x_4) = x_1x_2\overline{x}_3x_4 \vee \overline{x}_1\overline{x}_2\overline{x}_3\overline{x}_4 \vee x_1x_2\overline{x}_3\overline{x}_4 \vee x_1\overline{x}_2\overline{x}_3\overline{x}_4$$

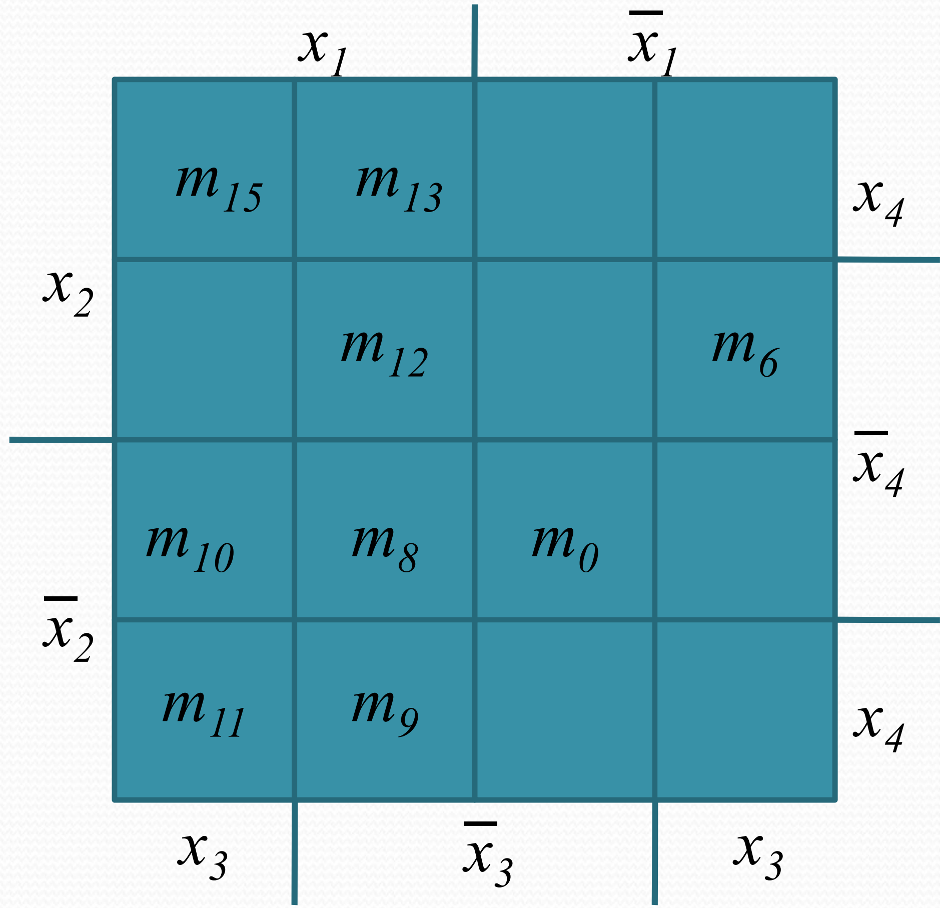
$$\vee\, x_1\overline{x}_2\overline{x}_3x_4$$

$$\vee\, x_1x_2x_3x_4$$

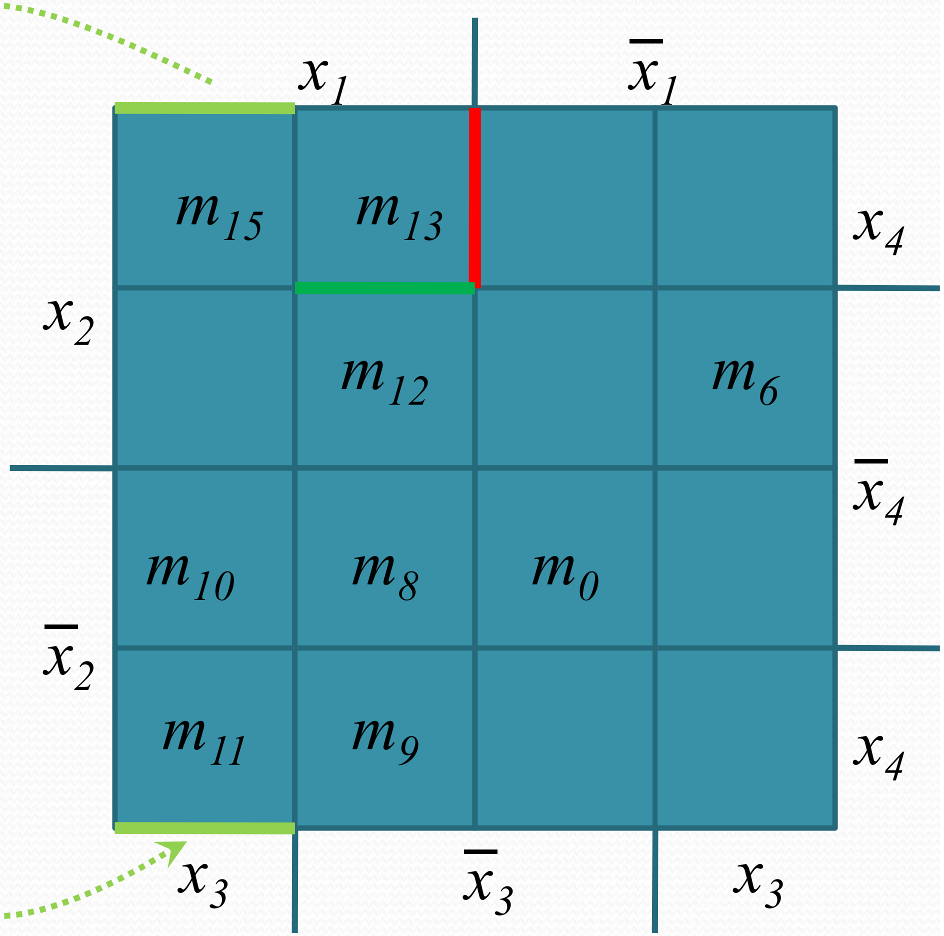
$$\vee\, \overline{x}_1x_2x_3\overline{x}_4$$

$$\vee\, x_1\overline{x}_2x_3\overline{x}_4$$

$$\vee\, x_1\overline{x}_2x_3x_4$$



Mintermi adiacenți



Factorizarea (1)

Se grupează 2^k, $k \in \mathbf{N}$ mintermi adiacenți, k – cât mai mare

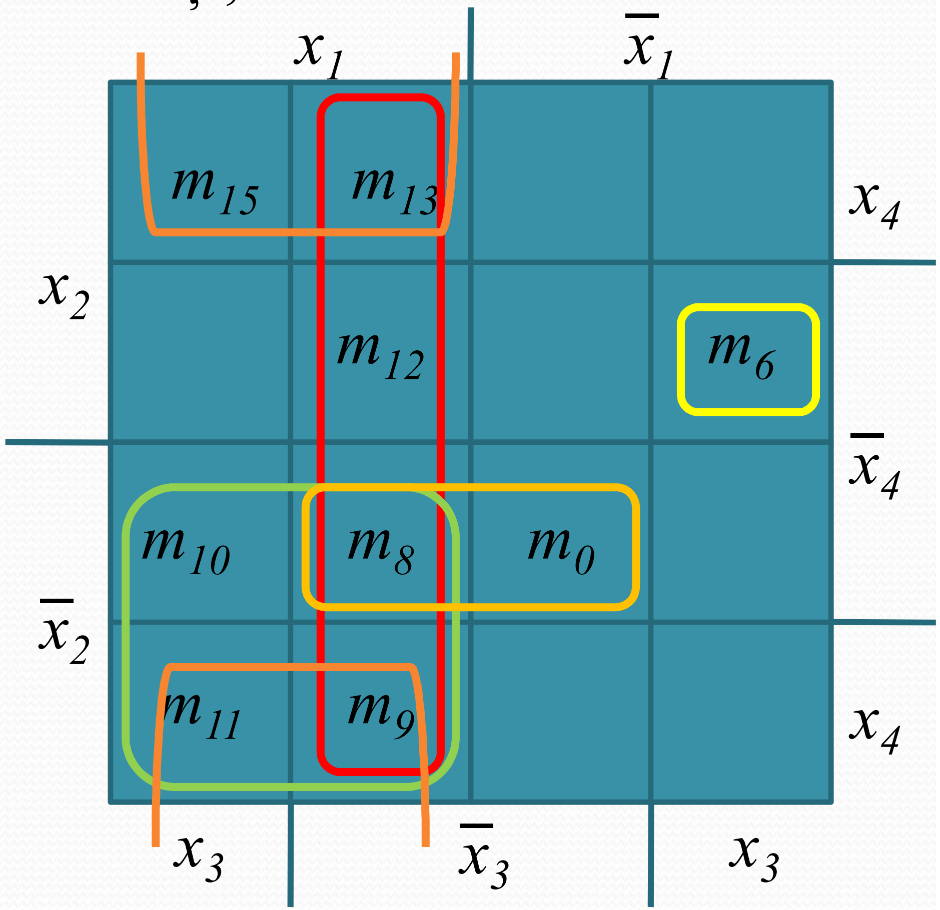
$max_1 = m_{13} \vee m_{12} \vee m_8 \vee m_9$

$max_2 = m_{10} \vee m_{11} \vee m_8 \vee m_9$

$max_3 = m_{11} \vee m_9 \vee m_{15} \vee m_{13}$

$max_4 = m_8 \vee m_0$

$max_5 = m_6$



Factorizarea (2)

- se grupează 2^k, $k \in \mathbf{N}$ mintermi adiacenți
- ! prima și ultima line, prima și ultima coloană sunt vecine!
- același minterm poate fi utilizat de mai multe ori
- ! se neglijează grupurile incluse!
- nu se lasă mintermi neîncercuiți

Formulele monoamelor maximale

$max_1 = m_{13} \vee m_{12} \vee m_8 \vee m_9 = x_1\overline{x}_3$

$max_2 = m_{10} \vee m_{11} \vee m_8 \vee m_9$

$= x_1\overline{x}_2$

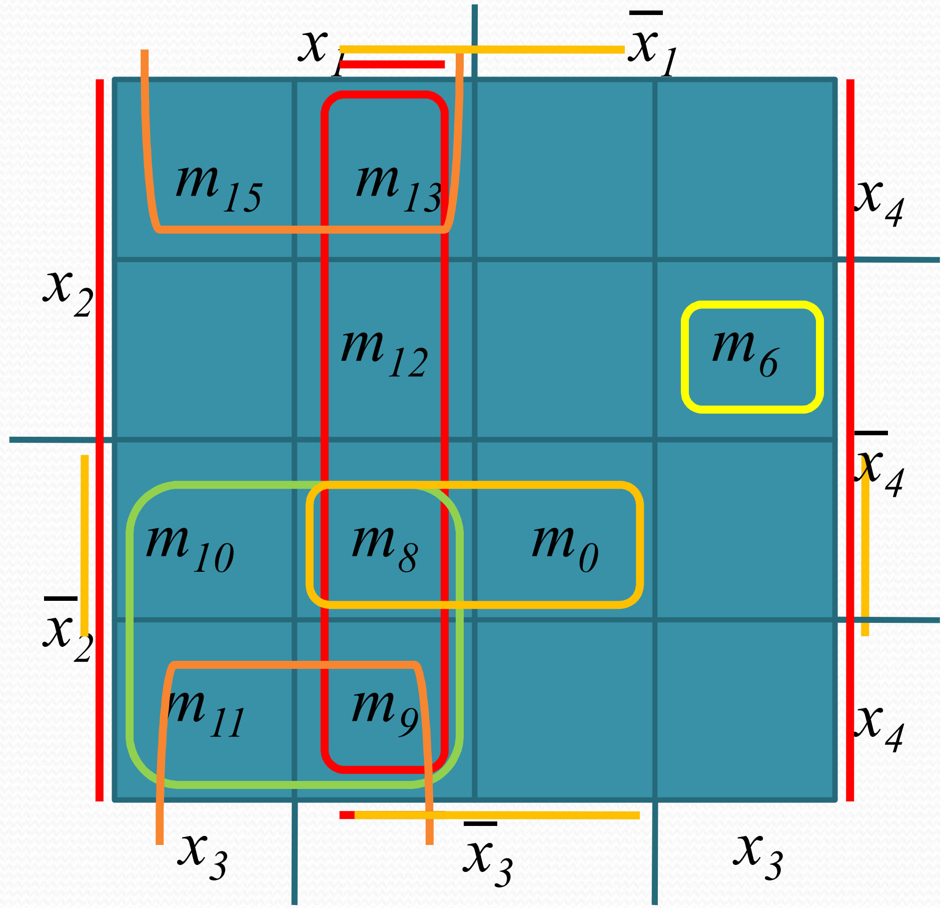
$max_3 = m_{11} \vee m_9 \vee m_{15} \vee m_{13}$

$= x_1 x_4$

$max_4 = m_8 \vee m_0$

$= \overline{x}_2\overline{x}_3\overline{x}_4$

$max_5 = m_6 = \overline{x}_1 x_2 x_3 \overline{x}_4$

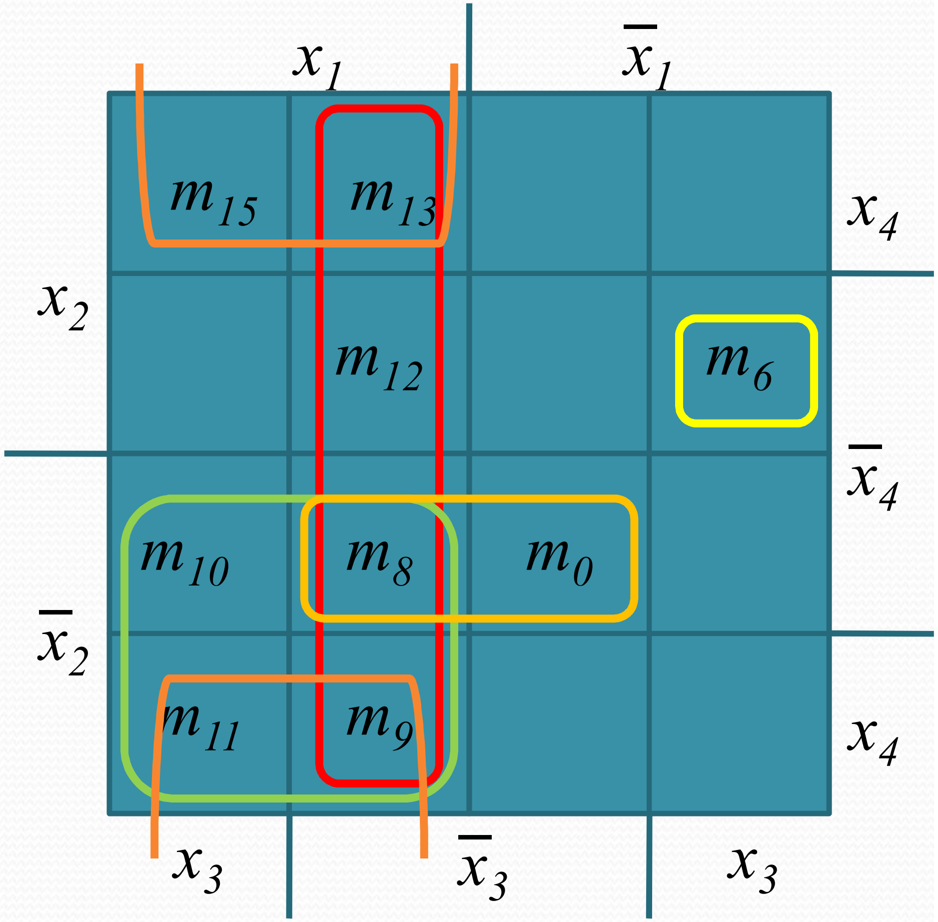


Mulţimea monoamelor maximale şi mulţimea monoamelor centrale

$M(f) =$

$\{max_1, max_2, max_3, max_4, max_5\}$

$M(f) = C(f)$



Forma simplificată a funcţiei

- Cazul I al algoritmului de simplificare
- O singură formă simplificată a funcţiei:

$$f^{s}(x_1, x_2, x_3, x_4) = x_1\overline{x}_3 \vee x_1\overline{x}_2 \vee x_1x_4 \vee \overline{x}_2\overline{x}_3\overline{x}_4 \vee \overline{x}_1x_2x_3\overline{x}_4$$

Construirea diagramei Karnaugh

$x_1 \backslash x_2 x_3$	00	01	11	10
0				
1				

- ! O singură cifră **distinctă**!

Logică computațională
Curs 12

Lector dr. Pop Andreea-Diana

Metoda analitică a lui Quine-Mc'Clusky

- se bazează pe completarea a două tabele ajutătoare
  - unul pentru factorizare, utilizat la calcularea mulţimii monoamelor maximale
  - unul pentru identificarea mulţimii monoamelor centrale
- se aplică formei canonice disjunctive a funcției
- poate fi utilizată pentru oricâte variabile

Aplicarea metodei

1. ordonarea mulţimii suport a funcţiei cu *n* variabile, crescător sau descrescător după numărul de valori 1 conţinut de fiecare *n*-uplu

$$f(x_1, x_2, x_3, x_4) = m_4 \vee m_6 \vee m_7 \vee m_8 \vee m_9 \vee m_{10} \vee m_{11} \vee m_{12}$$

S_f ={(0,1,0,0), (0,1,1,0), (0,1,1,1), (1,0,0,0), (1,0,0,1), (1,0,1,0), (1,0,1,1), (1,1,0,0)}

S_f ={(0,1,1,1), (1,0,1,1),
(0,1,1,0), (1,0,0,1), (1,0,1,0), (1,1,0,0),
(0,1,0,0), (1,0,0,0)}

Construirea primei tabele

- Capul de tabel conține numele variabilelor funcției.
- Pe fiecare linie se va completa suportul câte unui minterm din expresia funcției, grup după grup, în ordine crescătoare/descrescătoare a numărului valorilor de 1 din n-upluri.
- Grupurile se delimitează prin linii orizontale. După completarea întregii mulțimi suport a funcției, se va trasa o linie dublă.
- Doar două grupuri vecine pot conține suporturi ale monoame adiacente, astfel că factorizarea se poate efectua doar între n-upluri din grupuri vecine.

Prima tabelă

$S_f = \{(0,1,1,1), (1,0,1,1), (0,1,1,0), (1,0,0,1), (1,0,1,0), (1,1,0,0), (0,1,0,0), (1,0,0,0)\}$

Grupul	x_1	x_2	x_3	x_4	
I	0	1	1	1	m_7
	1	0	1	1	m_{11}
II	0	1	1	0	m_6
	1	0	0	1	m_9
	1	0	1	0	m_{10}
	1	1	0	0	m_{12}
III	0	1	0	0	m_4
	1	0	0	0	m_8

Operația de factorizare

- constă în introducerea unei linii noi în tabel, care conţine aceleaşi valori (0, 1 sau –) pe coloanele variabilelor comune celor două monoame care participă la factorizare şi simbolul „–” pe coloana corespunzătoare variabilei eliminate (o singură variabilă se va simplifica la un moment dat.
- Monoamele care participă la factorizare vor fi bifate deoarece acestea nu sunt maximale.
- Rezultatele factorizării a două grupuri de *n*-upluri vecine vor forma un nou grup (delimitat printr-o linie orizontală) de *n*-uplur în tabel, care poate fi utilizat mai departe în alte factorizări de ordin superior.
- Sfârşitul tuturor factorizărilor dintre două grupuri vecine de monoame cu acelaşi număr de simboluri „–” se marchează cu o dublă linie orizontală, pentru a sugera grafic faptul că s-a încheiat factorizarea de un anumit ordin şi se trece la factorizare de ordin mai mare.
- Acest proces continuă până când nu se mai pot face factorizări între două grupuri vecine delimitate printr-o singură bară orizontală. Tabelul se încheie cu o triplă bară orizontală.

Factorizarea

Grupul		x_1	x_2	x_3	x_4	
I	√	0	1	1	1	m_7
		1	0	1	1	m_{11}
II	√	0	1	1	0	m_6
		1	0	0	1	m_9
		1	0	1	0	m_{10}
		1	1	0	0	m_{12}
III		0	1	0	0	m_4
		1	0	0	0	m_8
IV=I+II		0	1	1	–	$m_7 \vee m_6$

Factorizare simplă

Factorizarea

	Grupul		x_1	x_2	x_3	x_4	
	I	√	0	1	1	1	m_7
		√	1	0	1	1	m_{11}
	II	√	0	1	1	0	m_6
		√	1	0	0	1	m_9
		√	1	0	1	0	m_{10}
		√	1	1	0	0	m_{12}
	III	√	0	1	0	0	m_4
		√	1	0	0	0	m_8
Factorizare simplă	$IV=I+II$		0	1	1	–	$m_7 \vee m_6$
		√	1	0	–	1	$m_{11} \vee m_9$
		√	1	0	1	–	$m_{11} \vee m_{10}$
	$V=II+III$		0	1	–	0	$m_6 \vee m_4$
		√	1	0	0	–	$m_9 \vee m_8$
		√	1	0	–	0	$m_{10} \vee m_8$
			–	1	0	0	$m_{12} \vee m_4$
			1	–	0	0	$m_{12} \vee m_8$
Factorizare dublă	$VI=IV+V$		1	0	–	–	$m_{11} \vee m_9 \vee m_{10} \vee m_8$

Identificarea monoamelor maximale

- Mulţimea **monoamelor maximale** este formată din toate monoamele corespunzătoare liniilor nebifate din tabel.

Monoamele maximale

	Grupul		x_1	x_2	x_3	x_4	
	I	√	0	1	1	1	m_7
		√	1	0	1	1	m_{11}
	II	√	0	1	1	0	m_6
		√	1	0	0	1	m_9
		√	1	0	1	0	m_{10}
		√	1	1	0	0	m_{12}
	III	√	0	1	0	0	m_4
		√	1	0	0	0	m_8
Factorizare simplă	*IV=I+II*		0	1	1	–	$m_7 \vee m_6 = \overline{x}_1 x_2 x_3 = max_1$
		√	1	0	–	1	$m_{11} \vee m_9$
		√	1	0	1	–	$m_{11} \vee m_{10}$
	V=II+III		0	1	–	0	$m_6 \vee m_4 = \overline{x}_1 x_2 \overline{x}_4 = max_2$
		√	1	0	0	–	$m_9 \vee m_8$
		√	1	0	–	0	$m_{10} \vee m_8$
			–	1	0	0	$m_{12} \vee m_4 = x_2 \overline{x}_3 \overline{x}_4 = max_3$
			1	–	0	0	$m_{12} \vee m_8 = x_1 \overline{x}_3 \overline{x}_4 = max_4$
Factorizare dublă	*VI=IV+V*		1	0	–	–	$m_{11} \vee m_9 \vee m_{10} \vee m_8 = x_1 \overline{x}_2 = max_5$

Mulțimea monoamelor maximale

- $M(f)=\{\overline{x}_1x_2x_3,\ \overline{x}_1x_2\overline{x}_4,\ x_2\overline{x}_3\overline{x}_4,\ x_1\overline{x}_3\overline{x}_4,\ x_1\overline{x}_2\}$
  $=\{max_1,\ max_2,\ max_3,\ max_4,\ max_5\}$

Identificarea monoamelor centrale

- se face utilizând un tabel care arată corespondenţa dintre monoamele maximale (pe coloane) şi mintermii (pe linii) care au contribuit prin factorizare la obţinerea acestora. O căsuţă din tabel se marchează cu o steluţă dacă mintermul corespunzător liniei a fost utilizat pentru obţinerea monomului maximal de pe coloană.

se încercuiesc * singure pe linie

Monoamele centrale – conțin pe coloană ⊛

monoame maximale / mintermi	max_1	max_2	max_3	max_4	max_5
m_7	⊛				
m_{11}					⊛
m_6	*	*			
m_9					⊛
m_{10}					⊛
m_{12}			*	*	
m_4		*	*		
m_8				*	*

Mulțimea monoamelor centrale

- $C(f) = \{max_1, max_5\}$

- Cazul II

Identificarea formelor simplificate

- $g(x_1, x_2, x_3, x_4) = max_1 \vee max_5 = \overline{x}_1 x_2 x_3 \vee x_1 \overline{x}_2$

Se hașurează coloanele ce conțin ⊛

monoame maximale / mintermi	max_1	max_2	max_3	max_4	max_5
m_7	⊛				
m_{11}					⊛
m_6	*	*			
m_9					⊛
m_{10}					⊛
m_{12}			*	*	
m_4		*	*		
m_8				*	*

Se hașurează liniile ce conțin * hașurate

monoame maximale / mintermi	max_1	max_2	max_3	max_4	max_5
m_7	⊛				
m_{11}					⊛
m_6	*	*			
m_9					⊛
m_{10}					⊛
m_{12}			*	*	
m_4		*	*		
m_8				*	*

Forma simplificată

- Se observă că cel mai simplu mod de a acoperi mintermii neacoperiți de funcția *g* (liniile nehașurate) este:
    - $h(x_1, x_2, x_3, x_4) = max_3$

- $f'(x_1, x_2, x_3, x_4) = g(x_1, x_2, x_3, x_4) \vee h(x_1, x_2, x_3, x_4) =$

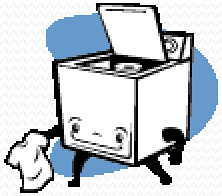
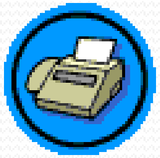
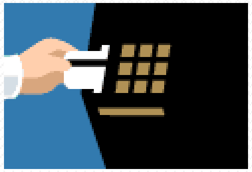
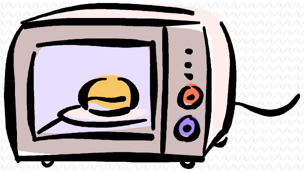
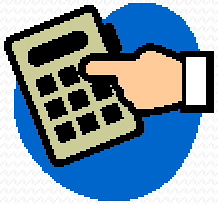
$$= \overline{x}_1 x_2 x_3 \vee x_1 \overline{x}_2 \vee x_2 \overline{x}_3 \overline{x}_4$$

Logică computațională
Curs 13

Lector dr. Pop Andreea-Diana

Circuite logice

- circuite electronice simple

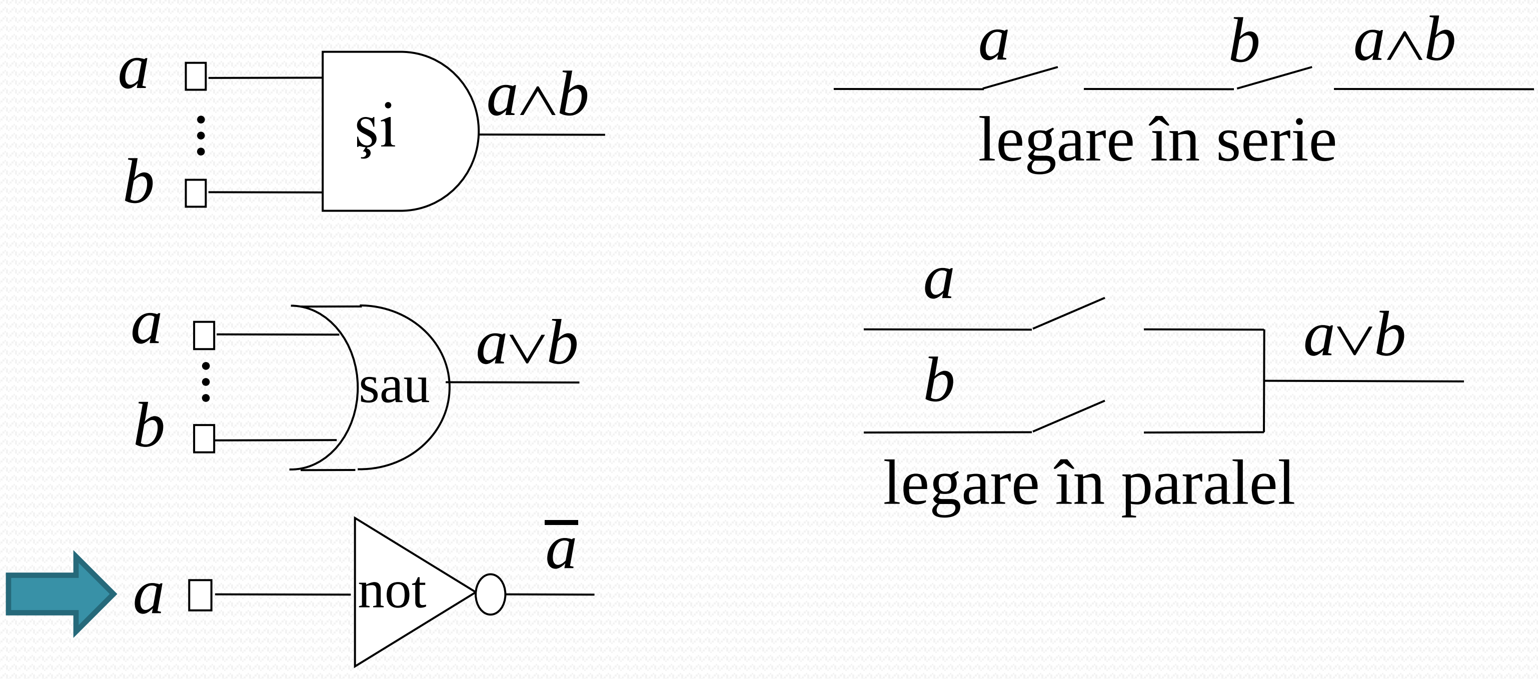
      

- modelarea – se face cu ajutorul *funcţiilor booleene* şi a *circuitelor logice* care descriu algebric şi grafic funcţionarea acestora.

Porțile logice

- sunt elementele de bază ale unui circuit logic
- sunt utilizate pentru modelarea circuitelor
- **Definiție:** O ***poartă*** este un minicircuit logic care realizează una dintre operaţiile logice de bază: $\wedge$, $\vee$, $\overline{\ }$.

Porțile logice – conform standardelor IEEE

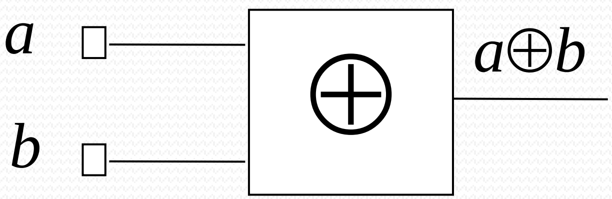


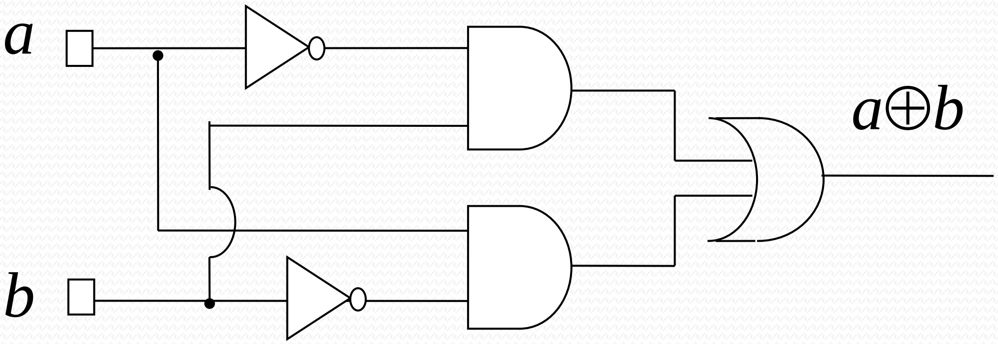
Circuite integrate

- 14-16 ”pini”
  - o parte porți de intrare
  - o parte sunt utilizate pentru conexiunea la curent

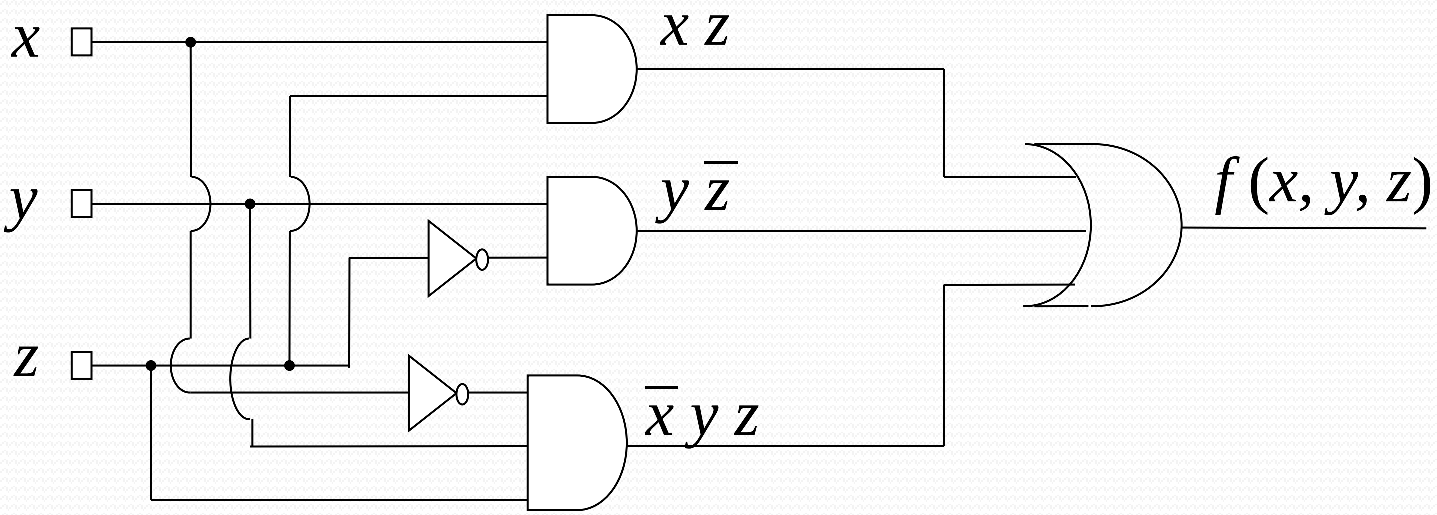
- Observație: forma disjunctivă este cel mai simplu de realizat

Exercițiu – desenați circuitul



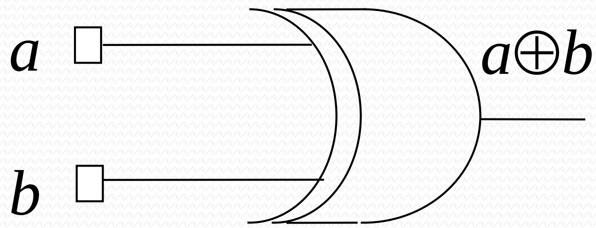


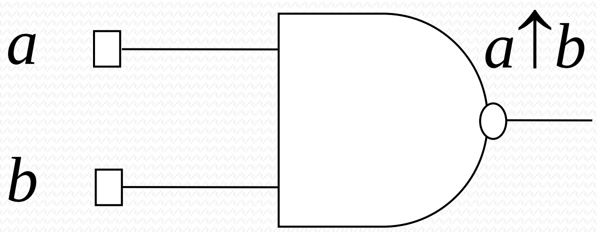
Exerciţiu – $f(x, y, z)$=?

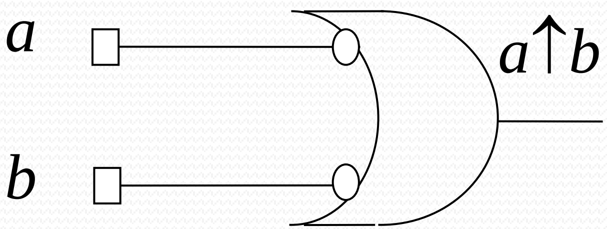


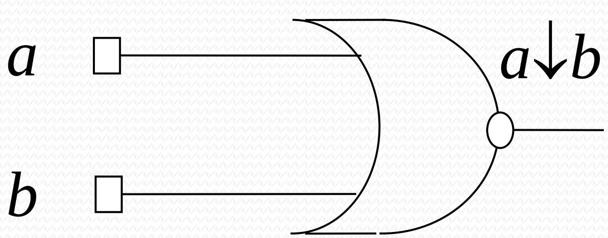
$$f(x, y, z) = x\,z \vee y\,\bar{z} \vee \bar{x}\,y\,z$$

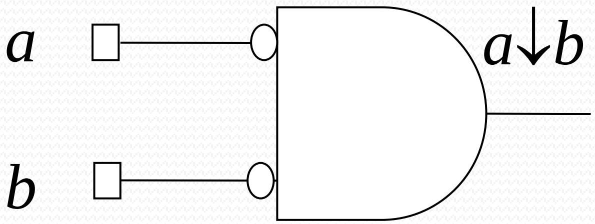
Porți derivate





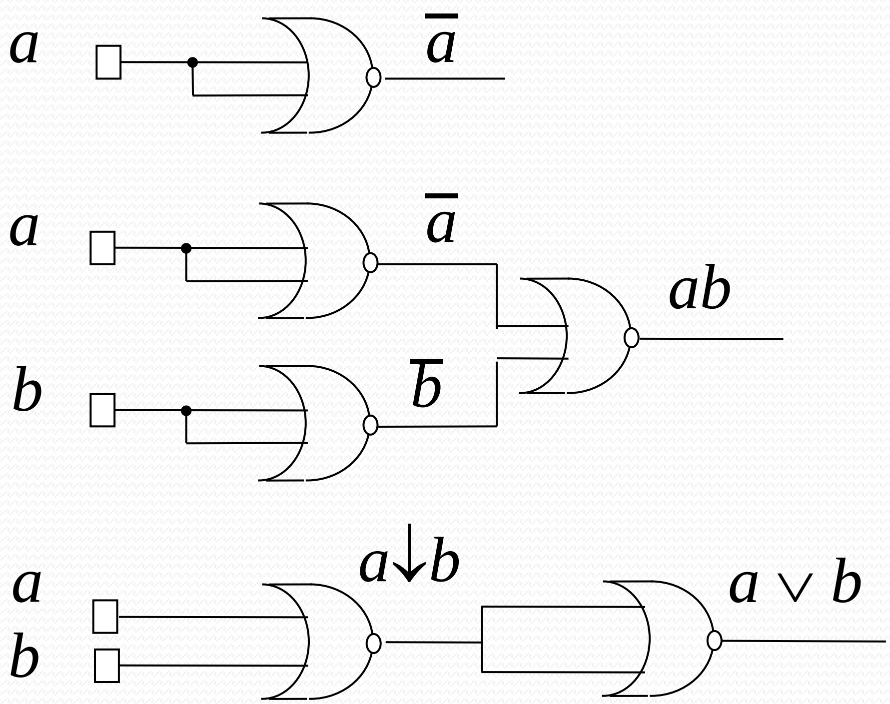






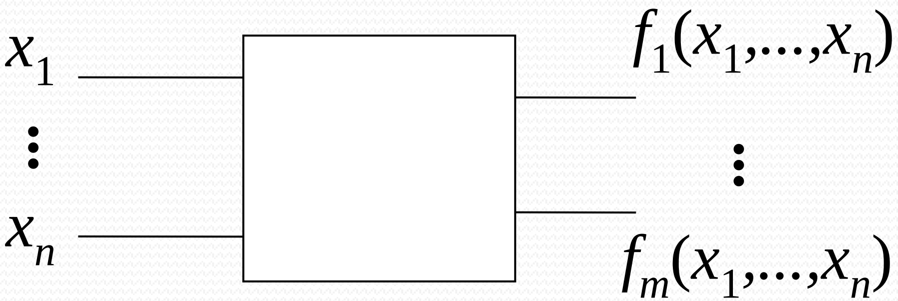
Exerciţiu

- Desenaţi circuitele operaţiilor logice „şi”, „sau”, „not” folosind doar poartă „nor” / „nand”



Circuit combinațional

- Un circuit logic cu *m* ieşiri se numeşte ***circuit combinaţional***.



Circuite logice combinaționale $\in$ Hard-ul calculatorului

- decodorul
- circuitul comparator
- circuitul sumator

- detectorul de paritate
- ”shift”
- …

Pașii principali pentru desenarea circuitelor

1. identificarea intrărilor (variabilelor) / ieșirilor (funcțiilor)
2. construirea tabelei de valori asociate
3. obținerea expresiilor funcțiilor
4. simplificarea funcțiilor
5. desenarea circuitului

Decodorul

1.

- intrare: 4 cifre binare - x_1, x_2, x_3, x_4
- ieșire: $f_i(x_1, x_2, x_3, x_4) = 1$ pentru $x_1x_2x_3x_4\,_{(2)} = i\,_{(10)}$, $i = \overline{0, 9}$

Decodorul (2)

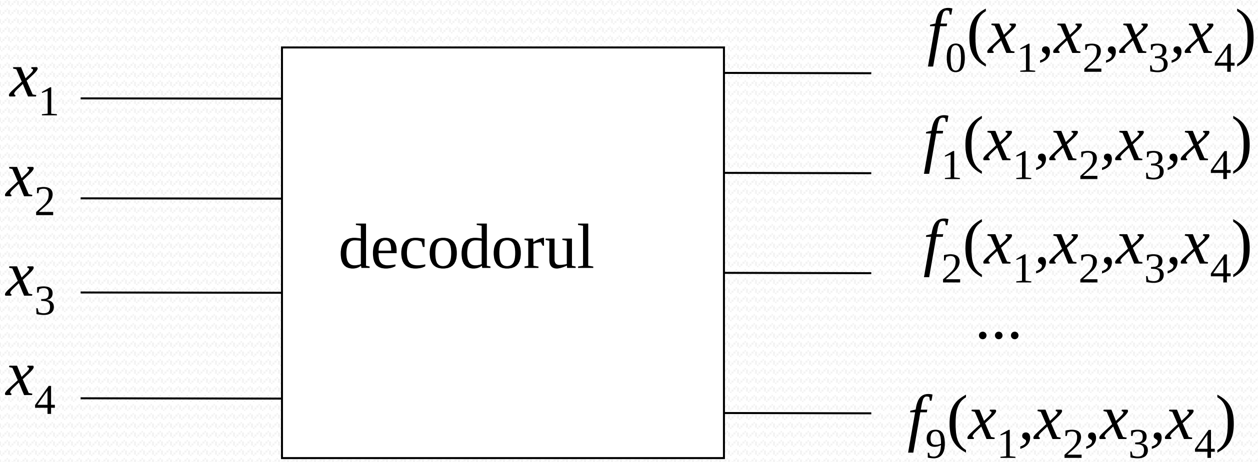
2. 3.

x_1	x_2	x_3	x_4	f_0	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	FCD (cu un singur element)
0	0	0	0	1	0	0	0	0	0	0	0	0	0	$f_0(x_1,x_2,x_3,x_4) = \overline{x}_1\overline{x}_2\overline{x}_3\overline{x}_4$
0	0	0	1	0	1	0	0	0	0	0	0	0	0	$f_1(x_1,x_2,x_3,x_4) = \overline{x}_1\overline{x}_2\overline{x}_3x_4$
0	0	1	0	0	0	1	0	0	0	0	0	0	0	$f_2(x_1,x_2,x_3,x_4) = \overline{x}_1\overline{x}_2x_3\overline{x}_4$
0	0	1	1	0	0	0	1	0	0	0	0	0	0	$f_3(x_1,x_2,x_3,x_4) = \overline{x}_1\overline{x}_2x_3x_4$
0	1	0	0	0	0	0	0	1	0	0	0	0	0	$f_4(x_1,x_2,x_3,x_4) = \overline{x}_1x_2\overline{x}_3\overline{x}_4$
0	1	0	1	0	0	0	0	0	1	0	0	0	0	$f_5(x_1,x_2,x_3,x_4) = \overline{x}_1x_2\overline{x}_3x_4$
0	1	1	0	0	0	0	0	0	0	1	0	0	0	$f_6(x_1,x_2,x_3,x_4) = \overline{x}_1x_2x_3\overline{x}_4$
0	1	1	1	0	0	0	0	0	0	0	1	0	0	$f_7(x_1,x_2,x_3,x_4) = \overline{x}_1x_2x_3x_4$
1	0	0	0	0	0	0	0	0	0	0	0	1	0	$f_8(x_1,x_2,x_3,x_4) = x_1\overline{x}_2\overline{x}_3\overline{x}_4$
1	0	0	1	0	0	0	0	0	0	0	0	0	1	$f_9(x_1,x_2,x_3,x_4) = x_1\overline{x}_2\overline{x}_3x_4$

… **0**

Circuitul decodor – forma generală

4. …
5. …



Circuitul comparator

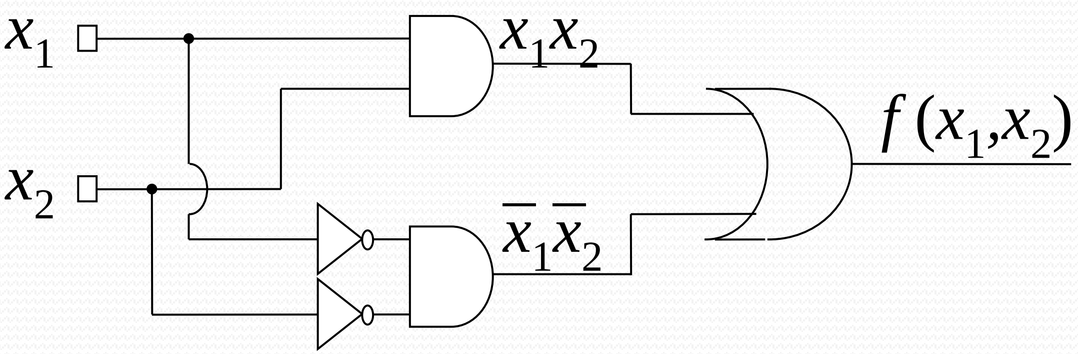
- verifică dacă două cifre binare sunt sau nu identice 1.

2.

x_1	x_2	$f(x_1,x_2)$
0	0	1
0	1	0
1	0	0
1	1	1

3. $f(x_1,x_2) = \overline{x}_1\overline{x}_2 \vee x_1x_2$ 4.

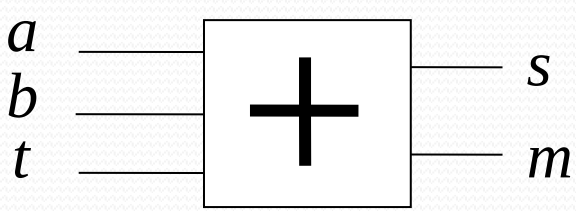
5.



Sumatorul binar

- calculează suma a două cifre binare: *a* şi *b* de pe aceeaşi poziţie dintr-un număr binar

1.
- intrare: *a*, *b*, transportul *t*
- ieșire: *s* (= *a* + *b*), transportul *m*



2.

t	a	b	s	m
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

3.

$$s(a,b,t)= \bar{t}\,\bar{a}b \vee \bar{t}\,a\bar{b} \vee t\bar{a}\,\bar{b} \vee tab$$

$$m(a,b,t)= \bar{t}\,ab \vee t\bar{a}\,b \vee ta\bar{b} \vee tab$$

4.Simplificarea

$$m(a,b,t)= \bar{t}\,ab \vee t\bar{a}\,b \vee ta\bar{b} \vee tab$$

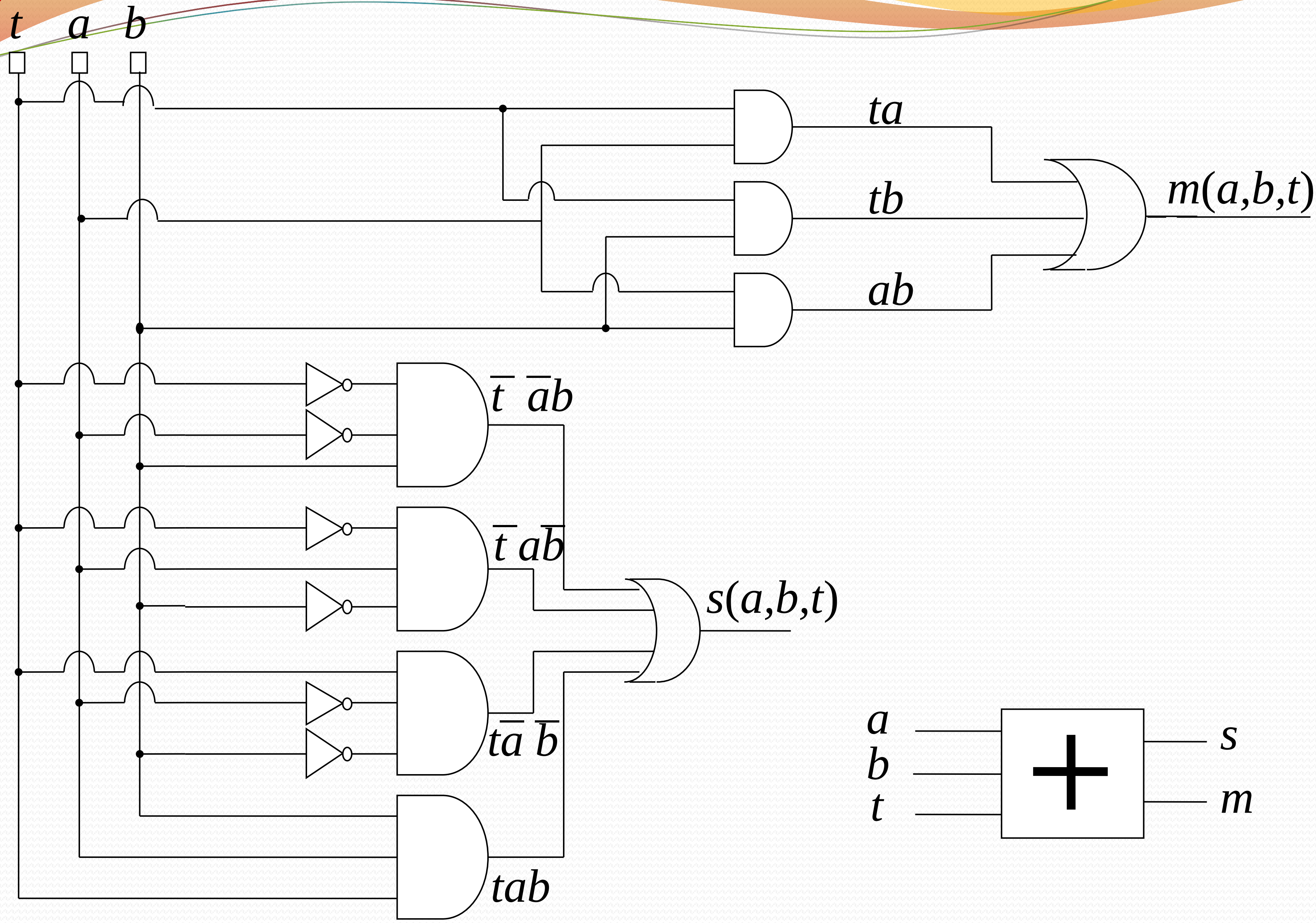
$$s(a,b,t)= \bar{t}\;\bar{a}b \vee \bar{t}\,a\bar{b} \vee t\bar{a}\,\bar{b} \vee tab$$

	t	t	$\bar{t}$	$\bar{t}$
a	m_7		m_2	
$\bar{a}$		m_4		m_1
	b	$\bar{b}$	$\bar{b}$	b

	t	t	$\bar{t}$	$\bar{t}$
a	m_7	m_6		m_3
$\bar{a}$	m_5			
	b	$\bar{b}$	$\bar{b}$	b

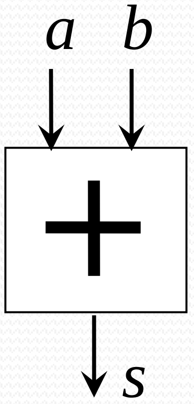
$$m(a,b,t)= ta \vee tb \vee ab$$

5. Circuitele

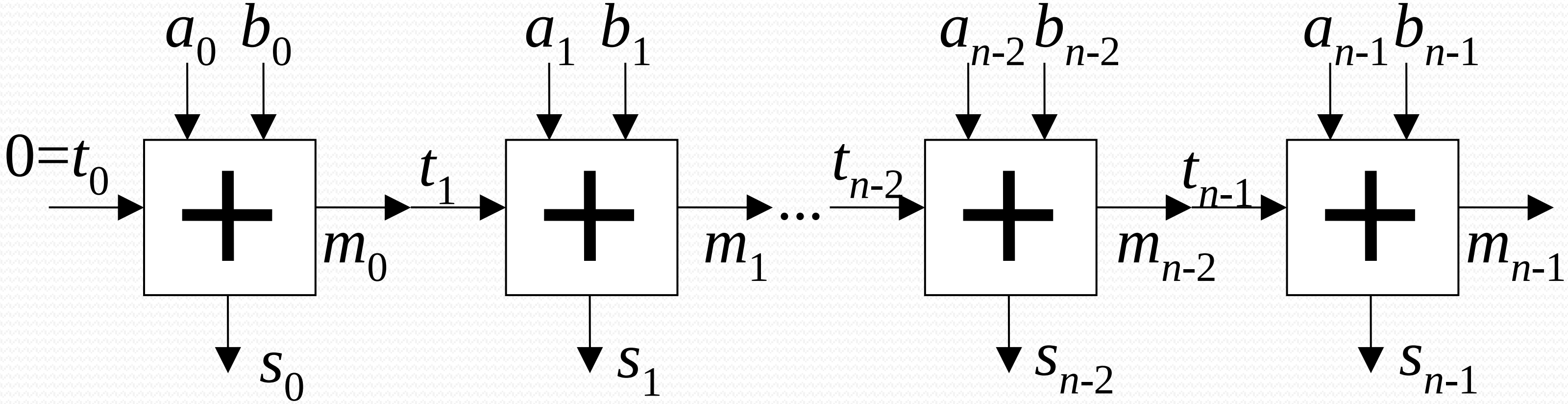


Sumatorul binar cu *n* poziții

- $a = a_{n\text{-}1} \ldots a_{0\ (2)}$ și $b = b_{n\text{-}1} \ldots b_{0\ (2)}$
- $s = s_{n\text{-}1} \ldots s_{0\ (2)}$

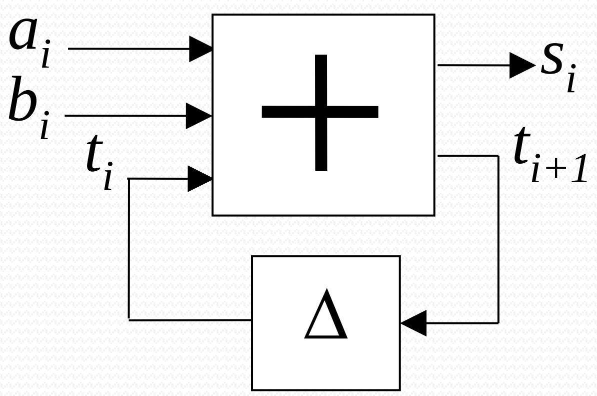


Compunere de sumatoare simple



Circuit cu întârziere

- cifra de transport obţinută la un pas se foloseşte în pasul următor



Indicații ”anti - încâlcire”