Arhitectura Calculatoarelor

Fizică - Informatică an II

gasner@uaic.ro

2. Circuite logice

Cuprins

- Funcții booleene
- Porți logice
- Circuite combinaționale
  - codoare şi decodoare
  - multiplexoare şi demultiplexoare
  - regiştri de deplasare
  - sumatoare
- Circuite secvențiale
  - circuite flip/flop
  - circuite flip-flop J/K master-slave
  - contor binar codat zecimal (BCD)

Analogic şi digital

- Semnal analogic
  - variație continuă în timp
  - spectru continuu de valori
- Semnal digital
  - variație bruscă („discontinuă”) în timp
  - spectru discret de valori
  - imensa majoritate a calculatoarelor utilizează semnal digital cu 2 niveluri – logică binară
- Conversia analogic-digital se realizează cu **modem** – modulator-demodulator (semnalele analogice sunt utilizate la transmisia la distanță a semnalelor digitale)

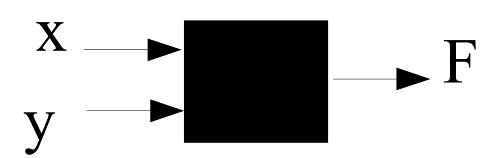
2.1 Funcții booleene

- mulțimea B={0,1}
- funcția f :$B^n \rightarrow B^m$
  - are *n* variabile şi *m* valori
  - corespunde unui circuit cu *n* intrări şi *m* ieşiri
- există (2^m)^2^n funcții
- *n*=1, *m*=1: 4 funcții unare:

x	$f_1(x)=0$	$f_2(x)=x$	$f_3(x)=\text{NOT } x$	$f_1(x)=1$
0	0	0	1	1
1	0	1	0	1

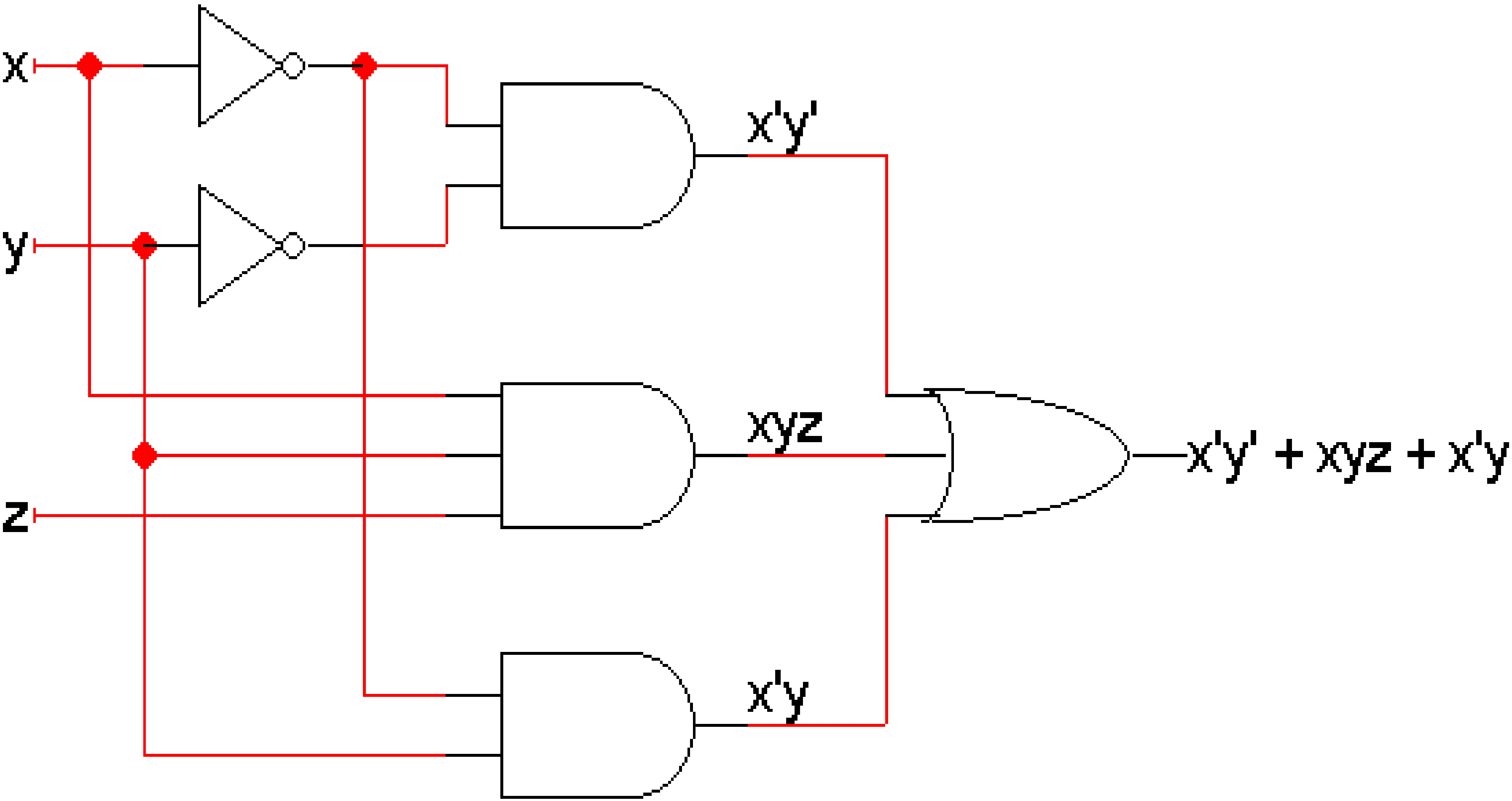
Funcții booleene cu două variabile

- 16 funcții cu 2 variabile de intrare şi 1 variabilă de ieşire



x	y	F_0	F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F_9	F_{10}	F_{11}	F_{12}	F_{13}	F_{14}	F_{15}
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

Axiome şi teorii

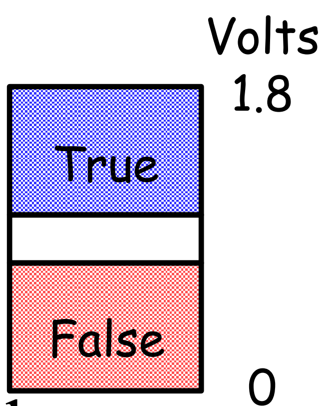


Operațiile calculatorului

- la nivel logic elementar, operațiile circuitelor de bază sunt operațiile logicii booleene
- se efectuează operațiile aritmetice în baza 2
- funcțiile booleene sunt implementate la nivel logic cu circuite combinaționale

Operațiile calculatorului

- este utilizată tensiunea electrică pentru reprezentarea celor două valori discrete 1 şi 0, cu care se formează numerele binare
- tensiunea electrică este utilizată şi la reprezentarea valorilor logice 'adevărat' (true) şi 'fals' (false)
- pentru simplicitate:
  - 1 este 'adevărat' (true)
  - 0 este 'fals' (false)



- această interpretarea este utilizată pentru implementarea funcțiilor speciale în hardware şi la efectuarea diverselor calcule

Exprimarea funcțiilor

- Şi funcțiile logice pot fi exprimate în două modalități:
  - O expresie booleană. finită, dar nu unică
  - tabelă de adevăr – unică şi finită

O expresie este
finită dar nu unică

$$f(x,y) = 2x + y$$
$$= x + x + y$$
$$= 2(x + y/2)$$
$$= \dots$$

O tabelă de adevăr este
unică dar finită

x	y	f(x,y)
0	0	0
…	…	…
2	2	6
…	…	…
23	41	87
…	…	…

Operații booleene elementare

Operație:	AND (produs) două intrări	OR (sumă) două intrări	NOT (complement) o intrare
Expresie:	xy, sau x•y	x + y	$\overline{x}$

Tabela de adevăr:

x	y	xy
0	0	0
0	1	0
1	0	0
1	1	1

x	y	x+y
0	0	0
0	1	1
1	0	1
1	1	1

x	NOTx
0	1
1	0

Porți logice elementare

- Fiecare operație elementară poate fi implementată hardware utilizând porțile logice elementare (primitive logic gate)

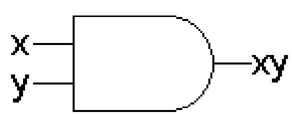
Operație:

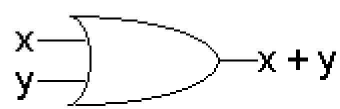
- AND (produs) două intrări
- OR (sumă) două intrări
- NOT (complement) o intrare

Expresie:

- xy, sau x•y
- x + y
- $\overline{x}$

Poarta:





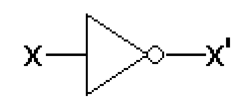


Tabela de adevăr:

x	y	xy
0	0	0
0	1	0
1	0	0
1	1	1

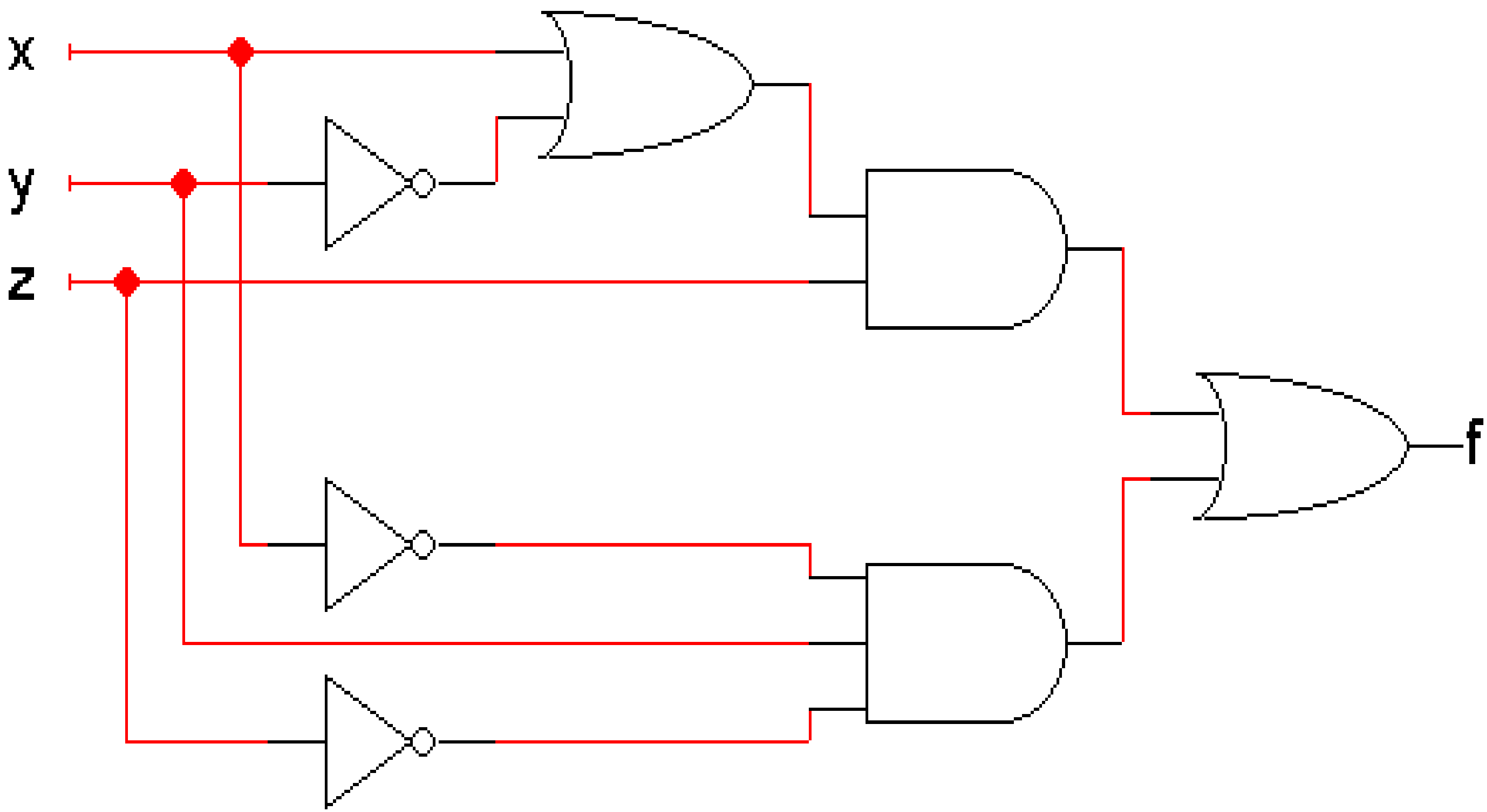
x	y	x+y
0	0	0
0	1	1
1	0	1
1	1	1

x	NOTx
0	1
1	0

Porți și expresii

- ***Orice*** expresie booleană poate fi convertită într-un circuit prin combinarea porților elementare

$$\left(\left(x+\overline{y}\right)\cdot z\right)+\left(\overline{x}\cdot y\cdot \overline{z}\right)$$



Expresii booleene

- Operațiile de bază por fi utilizate deci pentru a forma expresii complicate:

$$f(x,y,z) = (x + y')z + x'$$

- <u>Notă</u>:
  - f este numele funcției
  - (x,y,z) sunt variabilele de intrare (**1** sau **0**)
  - complementul se notează $x' =$ NOT x
- Ordinea operațiilor:
  - NOT urmat de AND, apoi OR.
  - paranteze (de ex. expresia de mai sus):

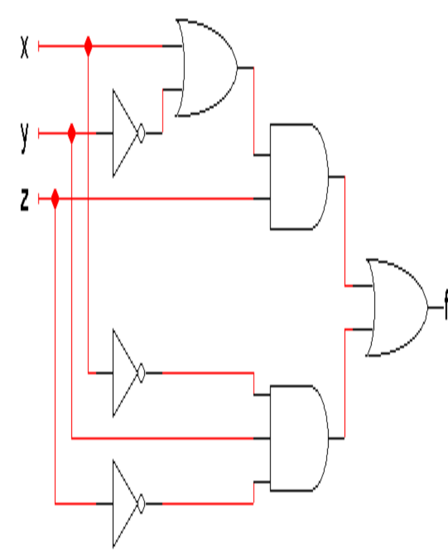
$$f(x,y,z) = (((x +(y'))z) + x')$$

Tabele de adevăr

- O tabelă de adevăr prezintă toate valorile posibile pentru intrările şi ieşirile funcţiei
  - Deoarece există doar un număr finit de valori (1 and 0), tabela de adevăr este şi ea finită
  - O funcţie cu *n* variabile are 2^n combinaţii posibile are intrărilor
- Intrările sunt listate în ordine binară (de ex. de la 000 la 111)

$f(x,y,z) = (x + y')z + x'$

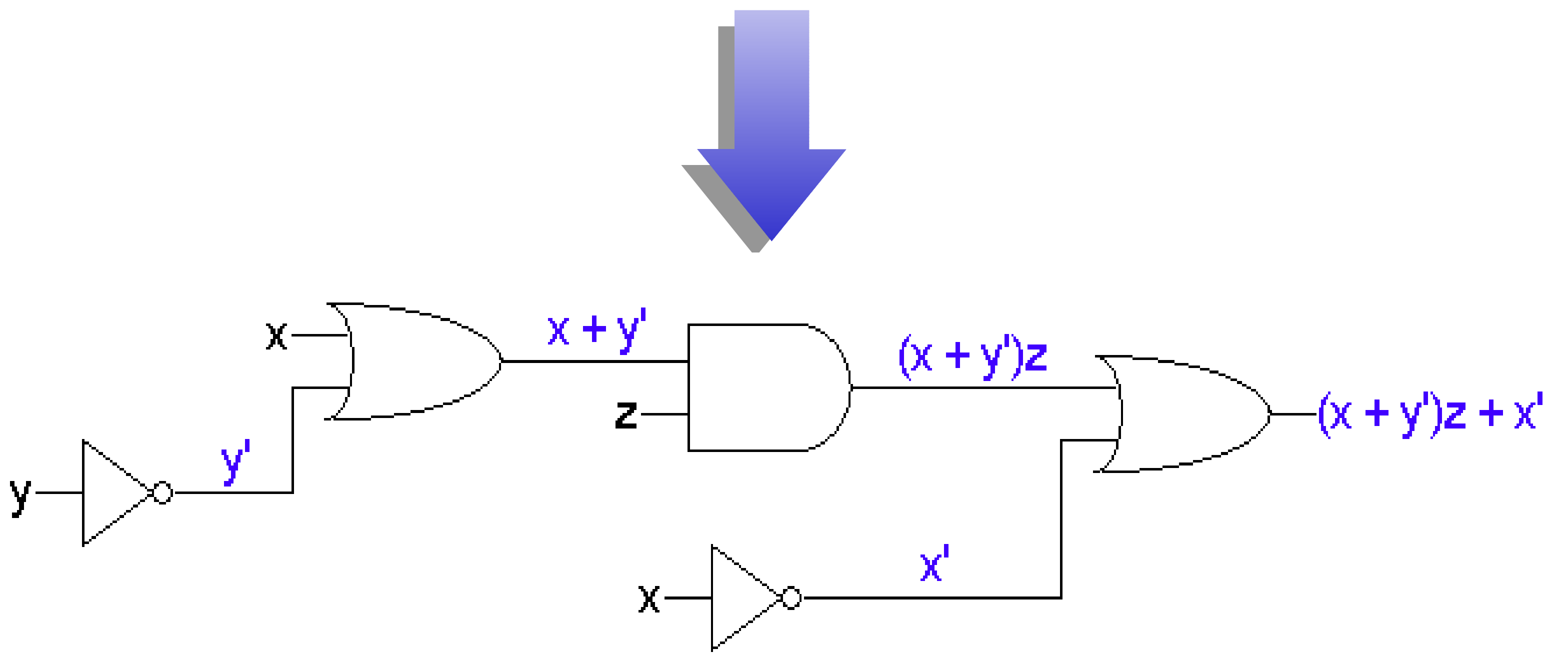
```
f(0,0,0) = (0 + 1)0 + 1 = 1
f(0,0,1) = (0 + 1)1 + 1 = 1
f(0,1,0) = (0 + 0)0 + 1 = 1
f(0,1,1) = (0 + 0)1 + 1 = 1
f(1,0,0) = (1 + 1)0 + 0 = 0
f(1,0,1) = (1 + 1)1 + 0 = 1
f(1,1,0) = (1 + 0)0 + 0 = 0
f(1,1,1) = (1 + 0)1 + 0 = 1
```



Expresii şi circuite

- Expresia booleeană este convertită într-un circuit cu porţi logice
- Diagrama de mai jos prezintă intrările şi ieşirile pentru fiecare poartă
- Ordinea operaţiilor este explicită în circuit

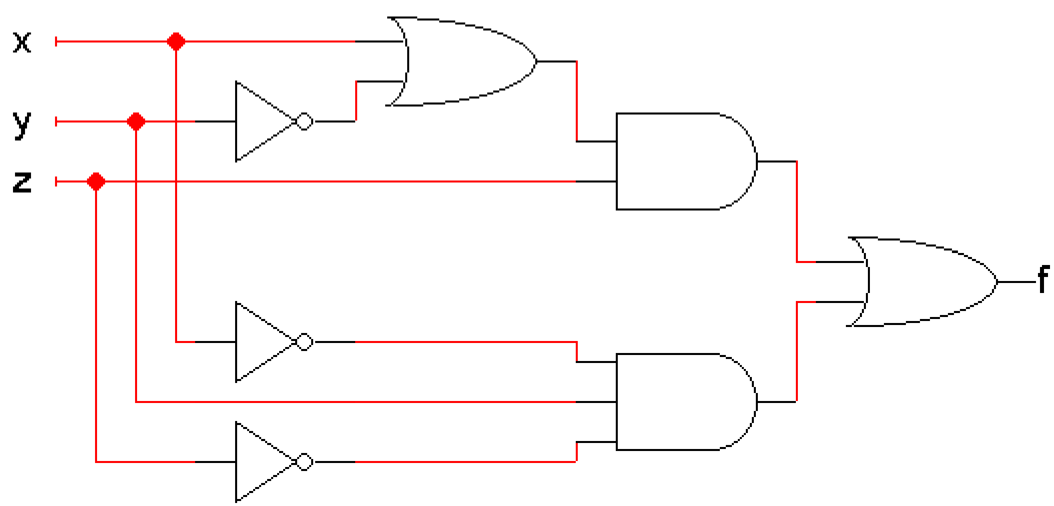
$$(x + y')z + x'$$



Analiza circuitului...

- Analiza de circuit explică modul de funcționare a circuitelor din punct de vedere logic
  - Fiecare circuit calculează o funcție care poate fi descrisă ca o expresie booleeană sau prin tabela de adevăr
  - Scopul este deci de a găsi o expresie echivalentă sau tabela de adevăr a circuitului

1. În primul rând trebuie stabilite clar mărimile de intrare şi de ieşire pentru circuit

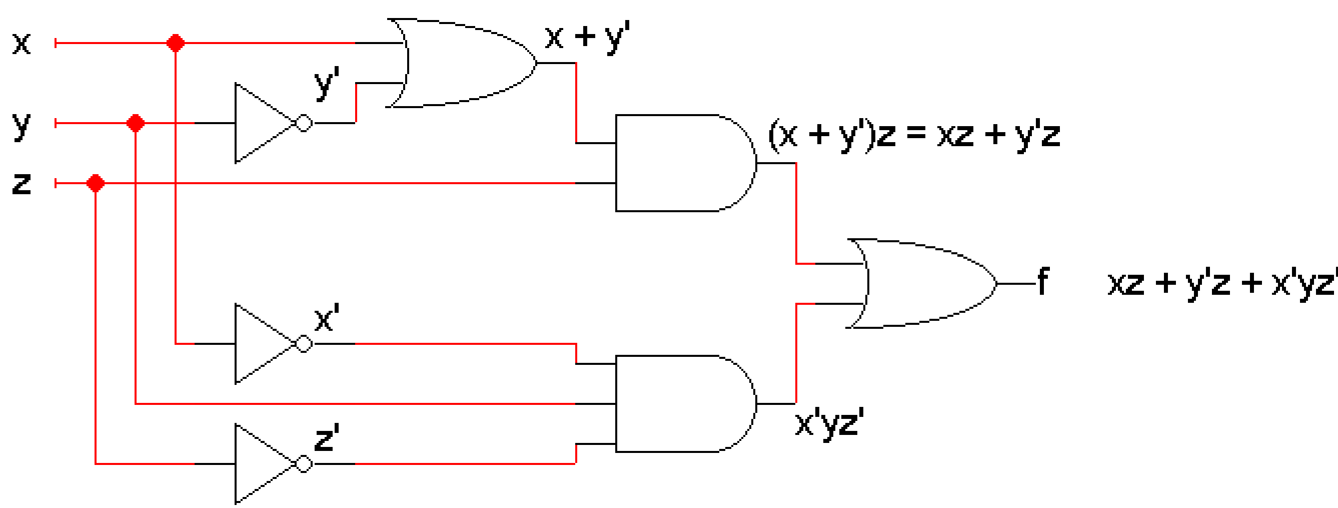


...ecuațiile algebrice...

2. Se scriu expresiile la ieşirea fiecărei porți, în funcție de intrările sale

    - Se începe de la inputs şi se termină la outputs
    - E preferabilă efectuarea de simplificări algebrice

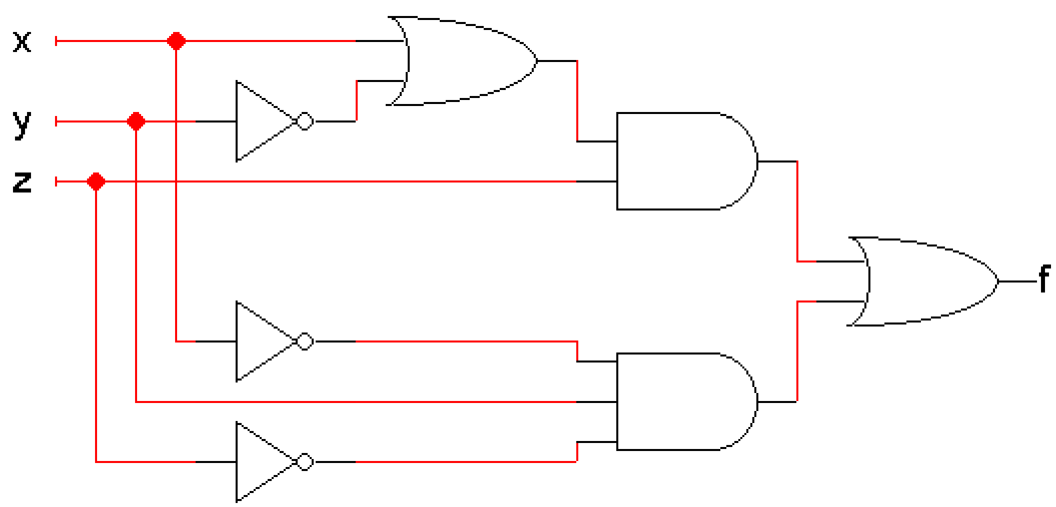
- Exemplu:
    - Apare o mică simplificare la poarta AND de sus
    - Se observă că circuitul calculează $f(x,y,z) = xz + y'z + x'yz'$



...sau tabela de adevăr

3. Este posibilă obținerea tabelei de adevăr direct din circuit

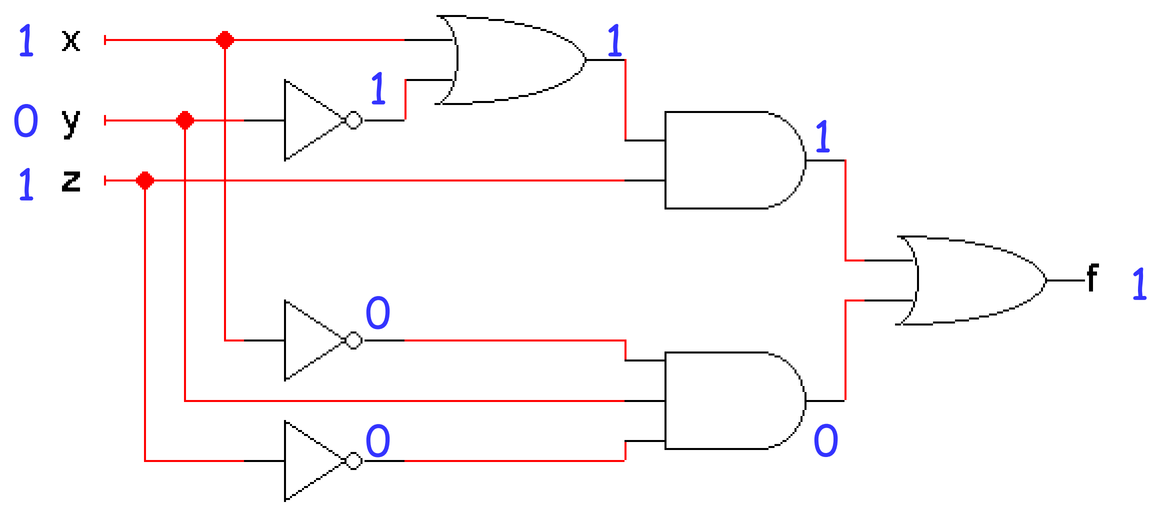
- Cunoscând numărul de inputs şi outputs, se listează toate combinațiile posibile în tabela de adevăr
  - Un circuit cu *n* inputs va avea o tabelă de adevăr cu 2^n linii
  - În exemplul nostru cu 3 inputs, tabela de adevăr are $2^3 = 8$ linii



x	y	z	f
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

Simularea circuitului...

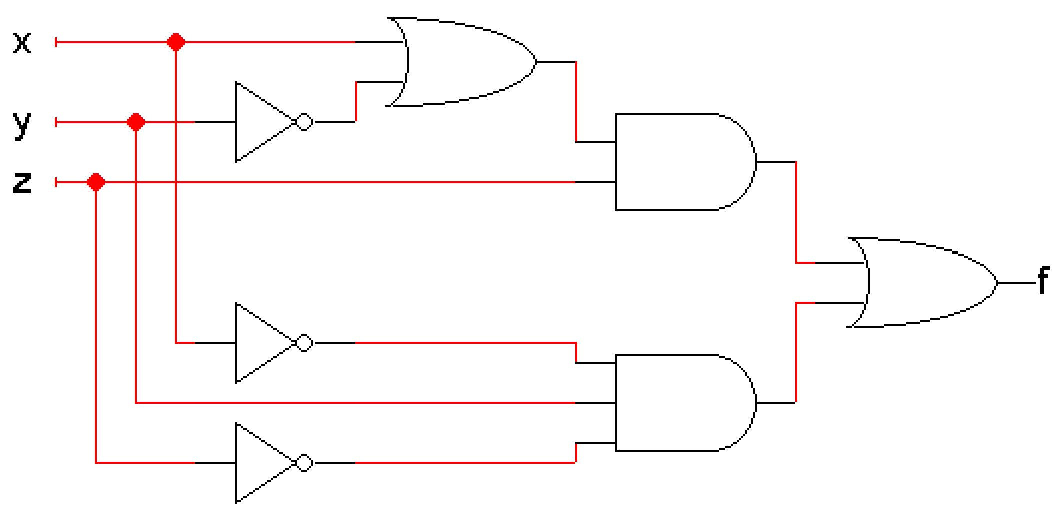
- Circuitul poate fi simulat „la mână” sau cu un program de simulare şi se găsesc stările ieşirilor pentru fiecare combinație posibilă a intrărilor
- De exemplu, când *xyz* = 101, ieşirile porților vor arăta astfel:
  - Se utilizează tabelele de adevăr pentru AND, OR şi NOT pentru a găsi ieşirile porților
  - Ieşirea finală este $f(1,0,1) = 1$



x	y	z	f
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	1
1	1	0	
1	1	1	

...şi finalizarea tabelei de adevăr

- În mod analog se procedează cu toate combinațiile posibile şi se completează tabela de adevăr
- Procesul este foarte simplu, dar îndelungat



x	y	z	f
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

Expresii şi tabele de adevăr

- Dacă deja avem expresia booleeană, se poate calcula uşor tabela de adevăr
- În exemplul nostru am găsit că circuitul calculează funcția $f(x,y,z) = xz + y'z + x'yz'$, wcu care se completează tabela (mai întâi termenii intermediari xz, $y'z$, $x'yz'$):

x	y	z	xz	y'z	x'yz'	f
0	0	0	0	0	0	0
0	0	1	0	1	0	1
0	1	0	0	0	1	1
0	1	1	0	0	0	0
1	0	0	0	0	0	0
1	0	1	1	1	0	1
1	1	0	0	0	0	0
1	1	1	1	0	0	1

Tabele de adevăr şi expresii

- Şi reciproca este valabilă: dacă avem tabela de adevăr se poate găsi uşor expresia booleană
- Tabela de adevăr poate fi convertită într-o sumă de minitermeni care corespund liniilor din tabelă a căror valoare la ieşire este **1**

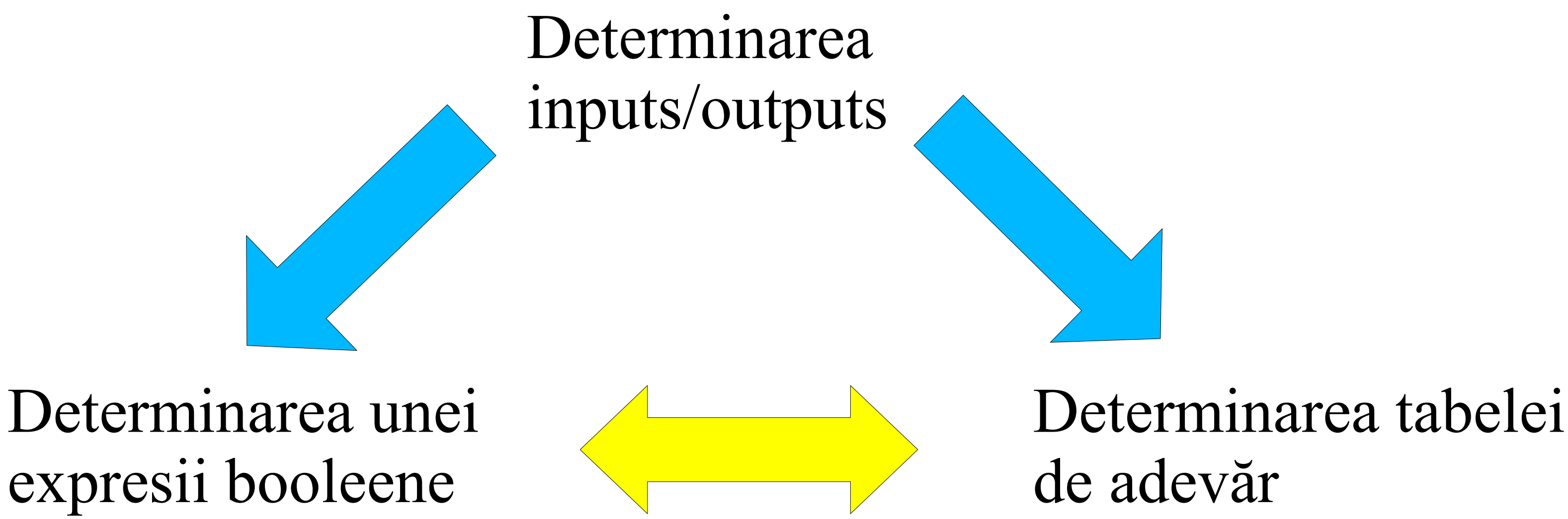
x	y	z	f
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

$$f(x,y,z) = x'y'z + x'yz' + xy'z + xyz$$
$$= m_1 + m_2 + m_5 + m_7$$

- Suma de minitermeni poate fi simplificată – cu K-map de exemplu

Analiza de circuit: concluzii

- După determinarea intrărilor şi ieşirilor unui circuit se poate găsi o expresie sau o tabelă de adevăr referitoare la comportamentul circuitului



Simplificarea expresiilor

- Expresia booleană se poate simplifica:

$x'y' + xyz + x'y$

$= x'(y' + y) + xyz$ (Distributivitatea: $x'y' + x'y = x'(y' + y)$)

$= \mathrm{x}'\cdot 1 + xyz$ ($y' + y = 1$)

$= x' + xyz$ ($x'\cdot 1 = x'$]

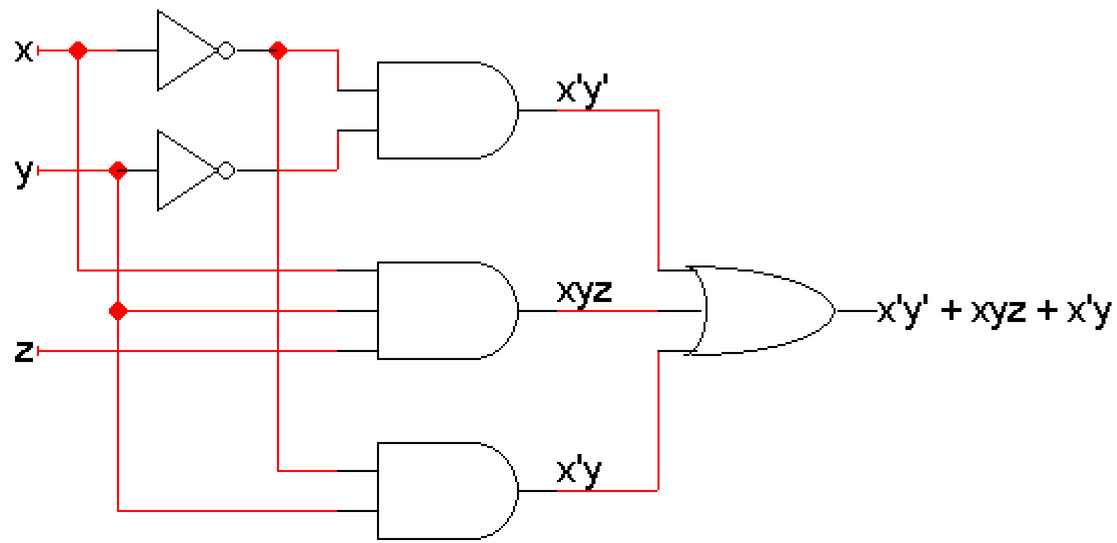
$= (x' + x)(x' + yz)$ (Distributivitate)

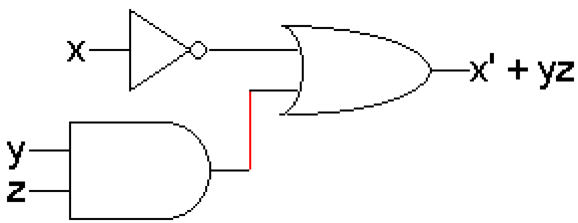
$= 1 \cdot (x' + yz)$ ($x' + x = 1$)

$= x' + yz$

Circuite echivalente

- Expresia booleană are aşadar două circuite **echivalente**
- Circuitul optim este circuitul cu cel mai puține porți:
  - este mai ieftin
  - consumă mai puțină energie
  - este mai fiabil





Complementul unei funcții

- Complementul unei funcții este negarea rezultatului funcției
- În tabela de adevăr, **0**-urile şi **1**-urile se interschimbă în coloana output

$f(x,y,z) = x(y'z' + yz)$

x	y	z	f(x,y,z)
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

x	y	z	f'(x,y,z)
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

Complementul unei funcții algebrice

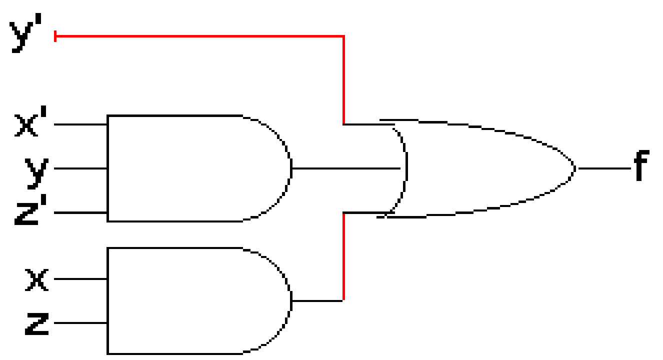
- Se pot utiliza legile lui de Morgan:
  - $f(x,y,z) = x(y'z' + yz)$
  - $f'(x,y,z) = (\, x(y'z' + yz)\,)'$ [complementare în ambii membri]
  - $= x' + (y'z' + yz)'$ [(xy)' = x' + y']
  - $= x' + (y'z')'\,(yz)'$ [(x + y)' = x' y']
  - $= x' + (y + z)(y' + z')$ [(xy)' = x' + y']
- Se poate negarea fiecărui termen din duala funcției:
  - $f(x,y,z) = x(y'z' + yz)$
  - duala funcției f este $x + (y' + z')(y + z)$
  - complementarea fiecărei variabile conduce la $x' + (y + z)(y' + z')$
  - și deci $f'(x,y,z) = x' + (y + z)(y' + z')$

Forme standard ale unei expresii

- O expresie poate fi scrisă în mai multe moduri, dar unele sunt mai folositoare decât altele
- Σ-notația sau suma de produse (sum of products SOP) conține:
  - Numai operații OR la nivelul cel mai apropiat de ieşire
  - Fiecare termen al sumei este un produs de variabile

$$f(x,y,z) = y' + x'yz' + xz$$

- Avantajul major al notației Σ este că implementarea se face pe un circuit pe două nivele
  - nivel 0: variabilele şi complementele lor
  - nivel 1: porți AND
  - nivel 2: o singură poartă OR



- Notă: porțile NOT sunt implicite iar variabilele apar de mai multe ori la intrare

Mintermeni

- Un mintermen este un produs special de variabile în care fiecare variabilă apare o singură dată
- O funcție cu *n* variabile are 2^n mintermeni
- Fiecare mintermen este adevărat pentru o singură combinație de intrări:

Mintermen	Adevărat dacă	Notație
x'y'z'	x=0, y=0, z=0	m_0
x'y'z	x=0, y=0, z=1	m_1
x'yz'	x=0, y=1, z=0	m_2
x'yz	x=0, y=1, z=1	m_3
xy'z'	x=1, y=0, z=0	m_4
xy'z	x=1, y=0, z=1	m_5
xyz'	x=1, y=1, z=0	m_6
xyz	x=1, y=1, z=1	m_7

Suma de mintermeni

- Orice funcție poate fi scrisă ca o sumă de minitermeni
- Notația Σ este unică pentru o funcție
- Dacă avem tabela de adevăr pentru o funcție, suma de mintermeni poate fi scrisă luând în considerare doar liniile care au 1 la output

x	y	z	f(x,y,z)	f'(x,y,z)
0	0	0	1	0
0	0	1	1	0
0	1	0	1	0
0	1	1	1	0
1	0	0	0	1
1	0	1	0	1
1	1	0	1	0
1	1	1	0	1

$$f = x'y'z' + x'y'z + x'yz' + x'yz + xyz'$$
$$= m_0 + m_1 + m_2 + m_3 + m_6$$
$$= \Sigma\, m(0,1,2,3,6)$$
$$f' = xy'z' + xy'z + xyz$$
$$= m_4 + m_5 + m_7$$
$$= \Sigma\, m(4,5,7)$$

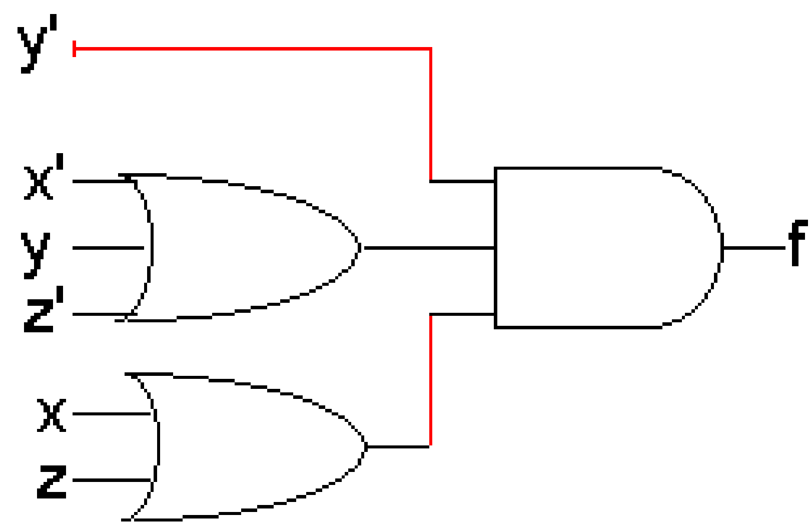
f' conține mintermenii care nu se găsesc în *f*

Produsul de sume – notația Π

- Există şi posibilitatea duală, produsul de sume, care conține
  - Numai operații AND la nivelul final
  - Fiecare termen este o sumă de variabile

$$f(x,y,z) = y' (x' + y + z') (x + z)$$

- Implementarea se realizează cu un circuit pe două nivele
  - nivel 0: inputs şi complementele lor
  - nivel 1: porți OR
  - nivel 2: o singură poartă AND



Maxtermeni

- Un maxtermen este o sumă de variabile în care fiecare input apare o singură dată
- O funcție cu *n* variabile are 2^n maxtermeni
- Fiecare maxtermen este fals pentru o singură combinație de inputs:

Maxtermen	Fals dacă	Notație
$x + y + z$	$x=0, y=0, z=0$	M_0
$x + y + z'$	$x=0, y=0, z=1$	M_1
$x + y' + z$	$x=0, y=1, z=0$	M_2
$x + y' + z'$	$x=0, y=1, z=1$	M_3
$x' + y + z$	$x=1, y=0, z=0$	M_4
$x' + y + z'$	$x=1, y=0, z=1$	M_5
$x' + y' + z$	$x=1, y=1, z=0$	M_6
$x' + y' + z'$	$x=1, y=1, z=1$	M_7

Produs de maxtermeni

- Orice funcție poate fi scrisă ca un produs unic de maxtermeni
- Dată tabela de adevăr a unei funcții, expresia funcției în notația Π se obține luând liniile din tabelă care au **0** la output

x	y	z	f(x,y,z)	f'(x,y,z)
0	0	0	1	0
0	0	1	1	0
0	1	0	1	0
0	1	1	1	0
1	0	0	0	1
1	0	1	0	1
1	1	0	1	0
1	1	1	0	1

$$f = (x' + y + z)(x' + y + z')(x' + y' + z')$$
$$= M_4\, M_5\, M_7$$
$$= \Pi\, M(4,5,7)$$
$$f' = (x + y + z)(x + y + z')(x + y' + z)$$
$$(x + y' + z')(x' + y' + z)$$
$$= M_0\, M_1\, M_2\, M_3\, M_6$$
$$= \Pi\, M(0,1,2,3,6)$$

f' conține maxtermenii care nu se găsesc în f

Mintermeni şi maxtermeni; conversii

- Mintermenul m_i este complementul maxtermenului M_i
- De exemplu, $m_4{}' = M_4$ deoarece $(xy'z')' = x' + y + z$
- Notația Σ se poate converti în Π:
  - $f = \Sigma m(0,1,2,3,6)$ şi $f' = \Sigma m(4,5,7) = m_4 + m_5 + m_7$
  - prin complementare $(f')' = (m4 + m5 + m7)'$ şi deci
  - $f = m_4{}' m_5{}' m_7{}'$ [DeMorgan]
  - $= M_4 M_5 M_7 = \Pi M(4,5,7)$
- În general, se înlocuiesc mintermenii cu maxtermenii, utilizând numerele de maxtermeni care nu apar în suma de mintermeni
- În mod analog se procedeză la conversia din produse de maxtermeni la sumă de mintermeni

Concluzii

- Algebra Bool ajută la simplificare expresiilor şi circuitelor, garantând obținerea unui circuit echivalent cu cel original
- Este necesară o metodă de optimizare mai simplă