LOGICKÉ OBVODY 2 – kombinační obvody, minimalizace

- logické obvody kombinační
- logické funkce a jejich reprezentace
- formy popisu –
  - tabulka,
  - n-rozměrné krychle
  - algebraický zápis
  - mapy

Kombinační x sekvenční obvody

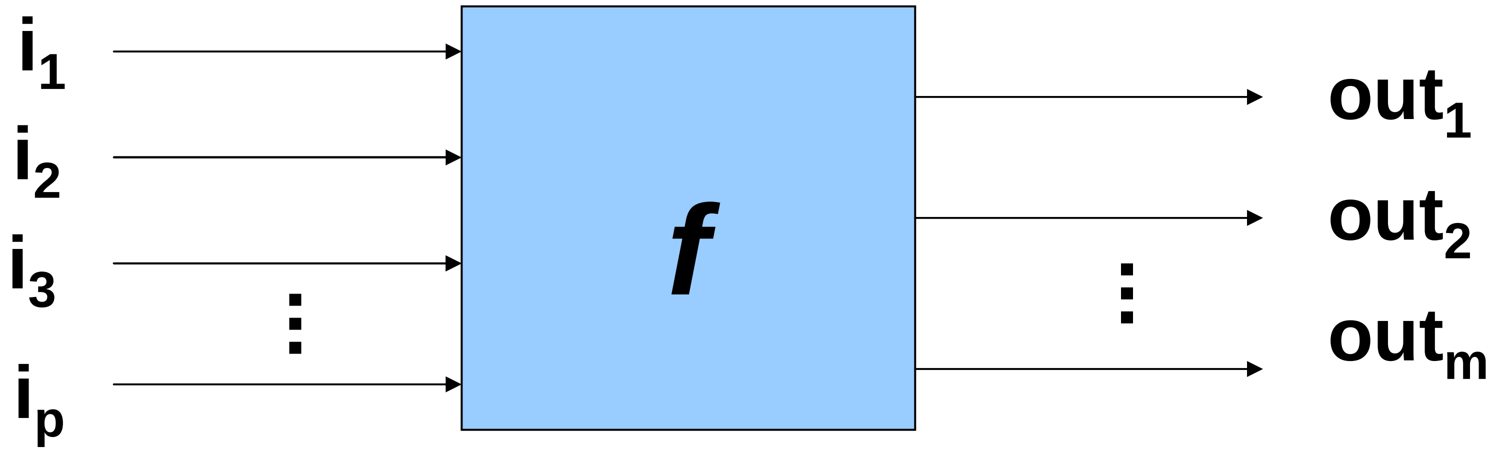
- Kombinační – vystup je dán kombinací vstupů, nezáleží na čase
- Sekvenční – výstup závisí na posloupnosti (sekvenci) hodnot na vstupech, realizuje se tzv. zpětnou vazbou

- Vše lze matematicky popsat
  - Logická funkce
  - Konečný automat - FSM

Kombinační funkce

Kombinační funkce:

$$out_k = f(i_1, i_2, i_3, \ldots\ i_p),\ k=1,2,\ldots,m$$



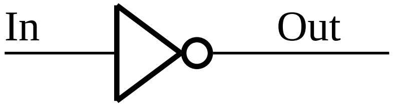
Základní kombinační prvky - hradla

Wire

Out = In

In	Out
0	0
1	1

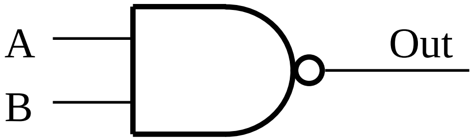
Inverter



$Out = \overline{In}$

In	Out
0	1
1	0

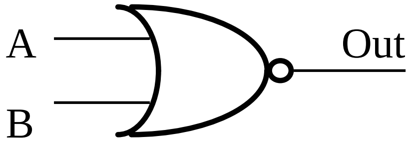
NAND Gate



A	B	Out
0	0	1
0	1	1
1	0	1
1	1	0

$Out = \overline{A \bullet B} = \overline{A} + \overline{B}$

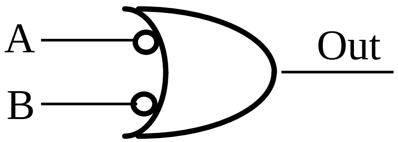
NOR Gate



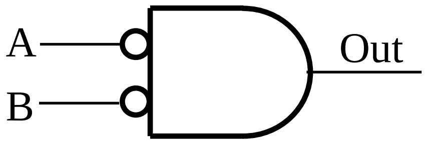
A	B	Out
0	0	1
0	1	0
1	0	0
1	1	0

$Out = \overline{A + B} = \overline{A} \bullet \overline{B}$

DeMorgan's Theorem

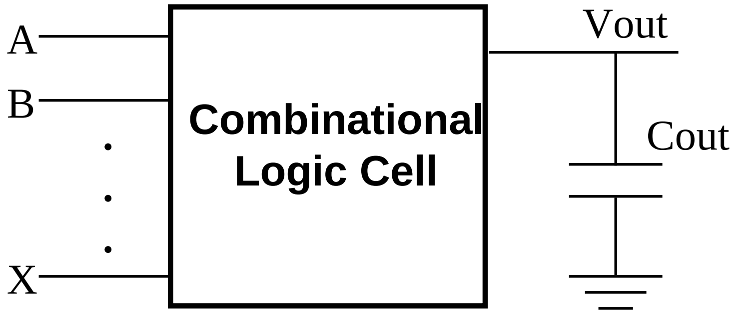


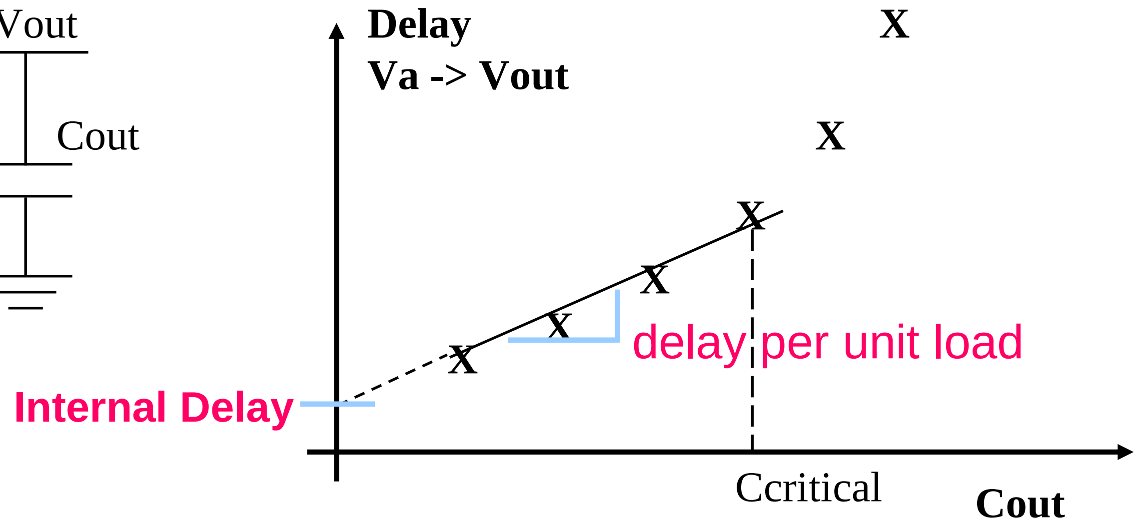
A	B	$\overline{A}$	$\overline{B}$	Out
0	0	1	1	1
0	1	1	0	1
1	0	0	1	1
1	1	0	0	0



A	B	$\overline{A}$	$\overline{B}$	Out
0	0	1	1	1
0	1	1	0	0
1	0	0	1	0
1	1	0	0	0

Obecná kombinační logická buňka, zpoždění





Kombinační buňka (symbol) je plně určena:

- Funkčním chováním (input -> output)
  - Pravdivostní tabulka, logická rovnice, ….
- Zatížením vstupů
- Propagačním zpožděním z každého vstupu na výstup a pro každou změnu signálu

Návrhový proces

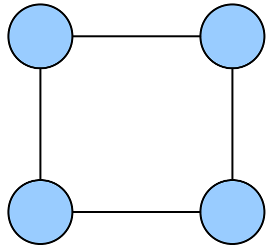
- Specifikace
- Určení vstupů a výstupů
- Pravdivostní tabulky
- Boolovské rovnice
- Návrh realizace na úrovni hradel
- Simulace na úrovni hradel
- Realizace číslicového obvodu
- Ověření návrhu

Základní pojmy logické syntézy

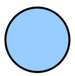
1. Logické funkce a jejich reprezentace, formy popisu a jejich vzájemný převod
   - tabulka (… měli jsme minule)
   - n-rozměrné krychle
   - algebraický zápis
   - mapy
2. Dvouúrovňová logická minimalizace –
   - terminologie
   - Karnaughova mapa
   - metoda Quine-McCluskey
3. Realizace na úrovni hradel

Boolovská n-krychle (cube) B^n

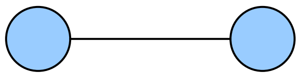
- **B = { 0,1}**
- **B^2 = {0,1} X {0,1} = {00, 01, 10, 11}**



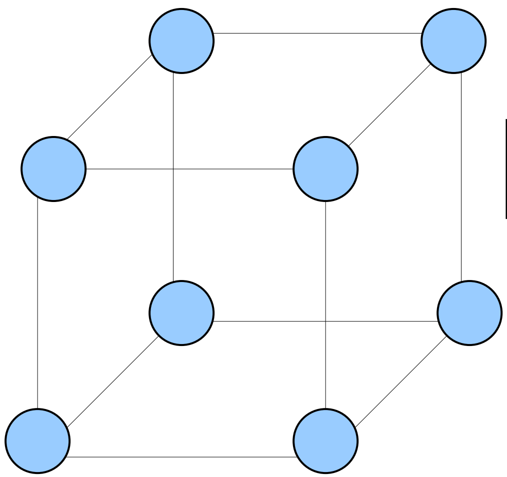
B^2



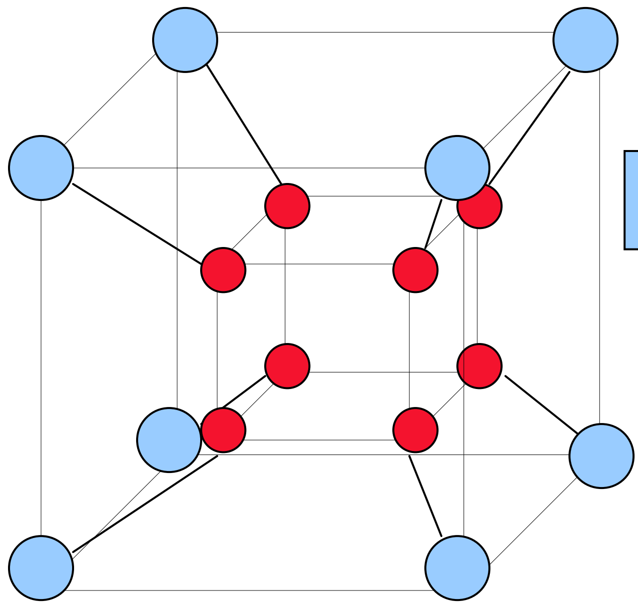
B^0



B^1



B^3



B^4

Booleovské funkce

$f(\mathrm{x}) : B^n \longrightarrow B$

$B = \{0, 1\}$, $x = (x_1, x_2, \ldots, x_n)$

- $x_1, x_2, \ldots$ jsou proměnné - variables
- $x_1, \overline{x}_1, x_2, \overline{x}_2, \ldots$ jsou literály - literals
- Každému vrcholu B^n je přiřazena 0 nebo 1
  - onset f je $\{x|f(x)=1\} = f^{\,1} = f^{\,-1}(1)$
  - offset f je $\{x|f(x)=0\} = f^{\,0} = f^{\,-1}(0)$
- jestliže $f^{\,1} = B^n$, f je tautologie, tzn. $f \equiv 1$
- jestliže $f^{\,0} = B^n$ ($f^{\,1} = \emptyset$), f není splnitelná
- jestliže $f(x) = g(x)$ pro všechna $x \in B^n$, pak f a g jsou ekvivalentní

Obvyklé zjednodušení: f namísto $f^{\,1}$

Literály

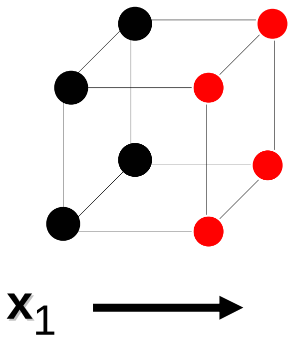
Literál je proměnná nebo její negace

$x_1, \overline{x_1}, a, \overline{z}, y$

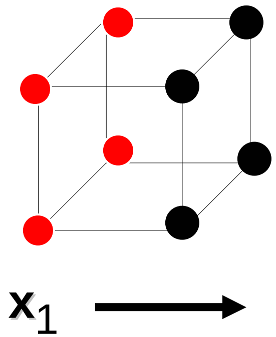
Literál reprezentuje logickou funkci.

Literál x_1 reprezentuje logickou funkci f, kde

$f = \{x \mid x_1 = 1\}$



$f = x_1$



$g = \overline{x_1}$

Literál $\overline{x_1}$ reprezentuje logickou funkci g, kde

$g = \{x \mid x_1 = 0\}$

Boolovské formule - výrazy

Boolovské formule (Boolean formulas) mohou být reprezentovány formulemi definovanými jako zřetězení

- závorek (,)
- literálů x, y, z, $\overline{x}$, $\overline{y}$, $\overline{z}$
- Boolovských operátorů + (OR), **.** (AND)
- komplementace, např. $\overline{x + y}$

Příklady

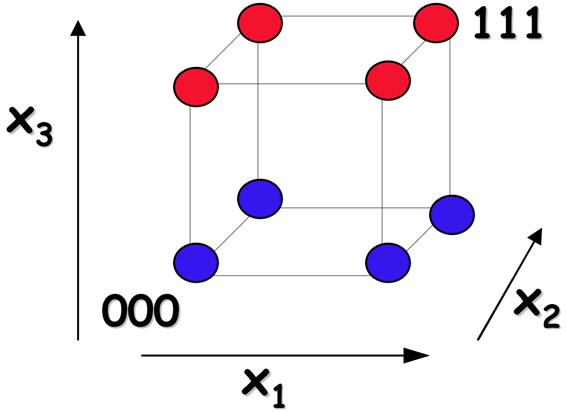
$$f = x_1 \cdot \overline{x_2} + \overline{x_1} \cdot x_2 = (x_1+x_2) \cdot (\overline{x_1}+\overline{x_2})$$

$$h = a + b \cdot c = \overline{\overline{a} \cdot (\overline{b} + \overline{c})}$$

Obvykle nahrazujeme **.** jen zřetězením, a **.** b $\rightarrow$ ab

Logické funkce

Existuje 2^n vrcholů v prostoru B^n



000		1
001		0
010		1
011		0
100	$\Rightarrow$	1
101		0
110		1
111		0

Existuje 2^{2^n} různých logických funkcí

Každá podmnožina vrcholů tvoří jinou logickou funkci:

$$f \subseteq B^n$$

Logické funkce

- Ale existuje nekonečně logických **formulí**

$$
\begin{aligned}
f &= x + y \\
&= xy + x\overline{y} + xy \\
&= x\overline{x} + x\overline{y} + y \\
&= (x + \overline{y})(x + y) + x\overline{y}
\end{aligned}
$$

- Syntéza – nalezení **"nejlepší"** formule (nebo “reprezentace”)

Boolovské operace - AND, OR, KOMPLEMENT

$$f : B^n \rightarrow B$$

$$g : B^n \rightarrow B$$

- AND - **fg = h** kde
  h = {x| f(x)=1 and g(x)=1}

- OR - **f + g = h** kde
  h = {x| f(x)=1 or g(x)=1}

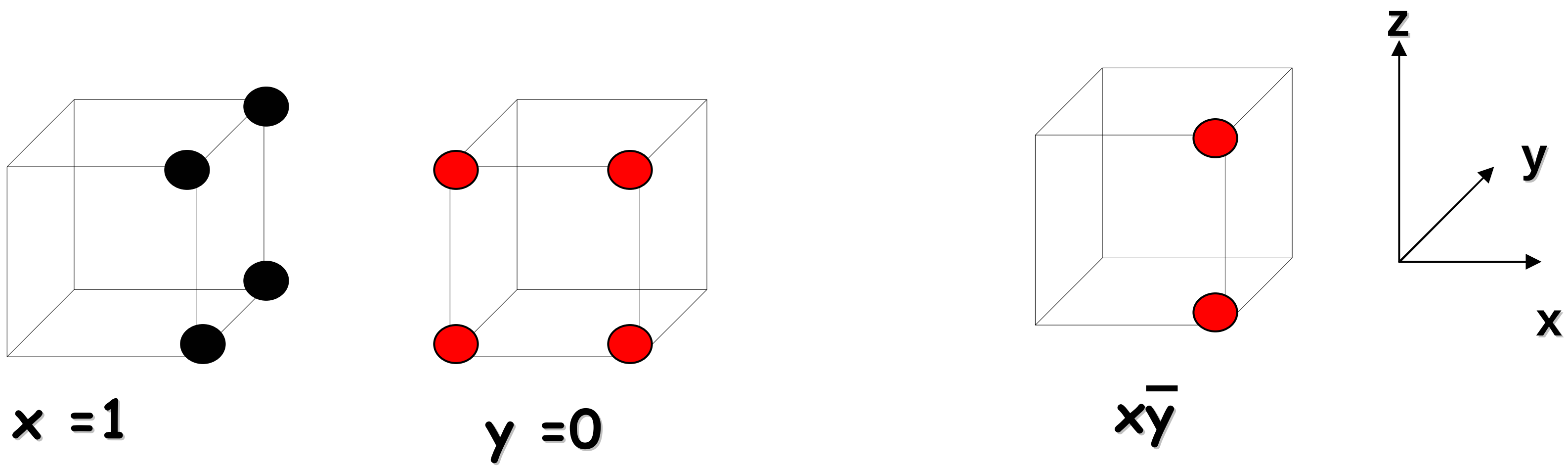
- KOMPLEMENT - $\overline{\mathbf{f}}$ **= h** kde
  h = {x| f(x) = 0}

Krychle - cube

- Logický součin (AND) množiny literálů („conjunction - konjunkce" literálů) je krychle

$$C = x.\overline{y} \qquad\qquad C = (x=1)(y=0)$$

ale může to být i samotný literál



Krychle - cube

- Jestliže $C \subseteq f$, C je krychle, pak C je implikant f.
- Když $C \subseteq B^n$ a C má k literálů, pak $|C|$ má **2^{n-k}** vrcholů.

Příklad1 $C = x\,y \subseteq B^3$.

k = 2 , n = 3.

$C = \{100, 101\}$.

$|C| = 2 = 2^{3-2}$.

- Jestliže k=n, pak krychle je minterm

(obsahuje všechny literály)

Reprezentace Boolovských funkcí

- Pravdivostní tabulka funkce $f_n: B^n \rightarrow B$ je vyjádření hodnot všech 2^n vrcholů z B^n.
- Pro

  $f = \bar{a}\bar{b}\bar{c}d + \bar{a}\bar{b}cd + \bar{a}b\bar{c}d + a\bar{b}\bar{c}d + a\bar{b}cd +$
  $ab\bar{c}d + abc\bar{d} + abcd$

  Pravdivostní tabulka (truth table):

  Nepoužitelná pro velká n
  (ale je kanonická - canonical)

  Kanonická znamená: když jsou dvě funkce stejné, je jejich kanonická reprezentace izomorfní.

	abcd	f
0	0000	0
1	0001	1
2	0010	0
3	0011	1
4	0100	0
5	0101	1
6	0110	0
7	0111	0
8	1000	0
9	1001	1
10	1010	0
11	1011	1
12	1100	0
13	1101	1
14	1110	1
15	1111	1

Sum-of-Products – SOP (Disjunktivní forma)

- Funkce může být reprezentována jako součet krychlí (součinů):

$$f = ab + ac + bc$$

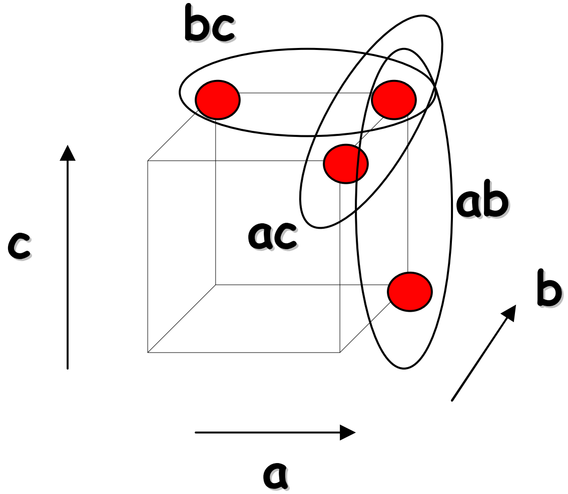
Každá krychle je součin literálů, mluvíme tedy o reprezentaci "sum of products" – součet součinů SOP …. DNF

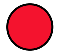
- SOP Můžeme považovat za množinu krychlí F

$$F = \{ab, ac, bc\} = C$$

- Množinu krychlí reprezentující f nazýváme pokrytí (cover) *f*.
- $F=\{ab, ac, bc\}$ je pokrytí fukce $f = ab + ac + bc$.

SOP




= onset minterm

Každý onset minterm je "pokrytý" nejméně jednou krychlí a nepokrývá žádný offset minterm (nulový vrchol).

Pokrytí (SOP's) mohou efektivně reprezentovat mnoho logických funkcí.
Dvouúrovňová minimalizace (two-level minimization) hledá pokrytí o minimální velikosti (nejmenší počet krychlí)

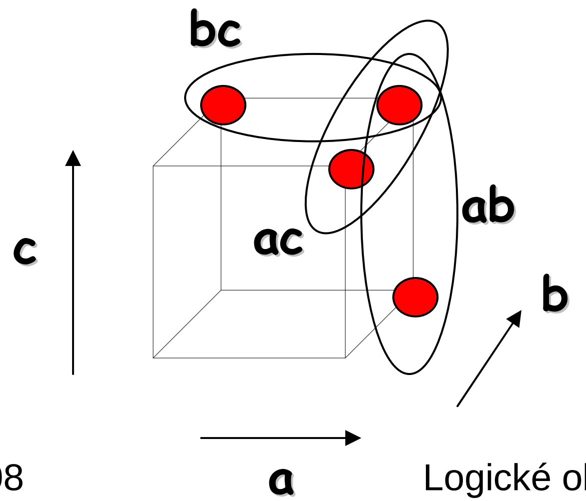
Neredundance

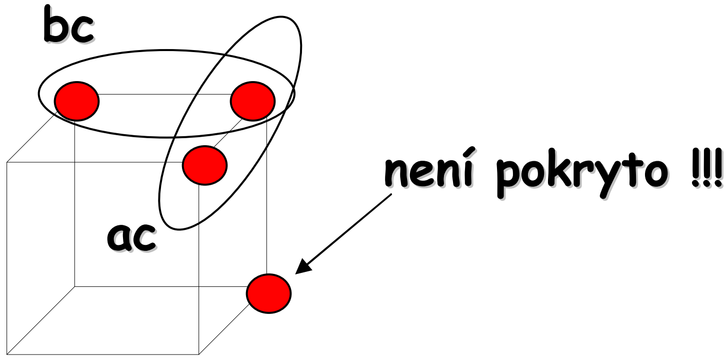
- Nechť $F = \{c_1, c_2, \ldots, c_k\}$ je pokrytí pro f.

$$f = \sum_{i=1}^{k} c_i$$

Krychle $c_i \in F$ je neredundantní, jestliže $F \setminus \{c_i\} \neq f$

Příklad 2: $f = ab + ac + bc$





$F \setminus \{ab\} \neq f$

Prime – přímost

- Literál j krychle $c_i \in F$ ($=f$) je přímý (prime) jestliže:

$$(F \setminus \{c_i\}) \cup \{c'_i\} \neq f$$

kde c'_i je c_i ve kterém je literál j z c_i vypuštěn.

- Krychle z F je přímá když všechny její literály jsou přímé (nemohou být vypuštěny)

Příklad 3

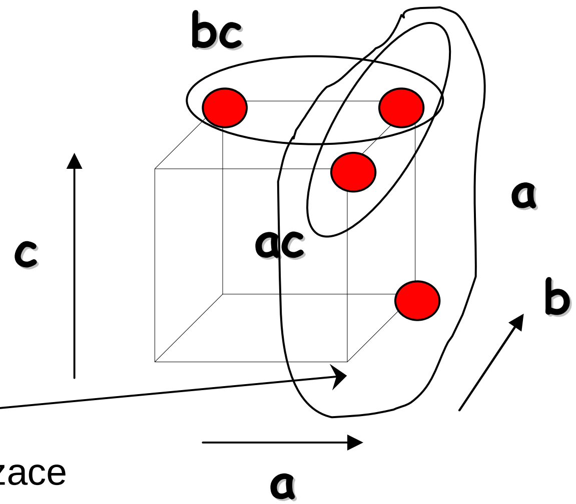
$f = ab + ac + bc$

$c_i = ab$; $c'_i = a$ (literál b odtraněn)

$F \setminus \{c_i\} \cup \{c'_i\} = a + ac + bc$

F=ac + bc + a = F \{c_i} ∪ {c'_i}

Nerovná se f protože je pokrytý offsetový vrchol



Podstatná krychle

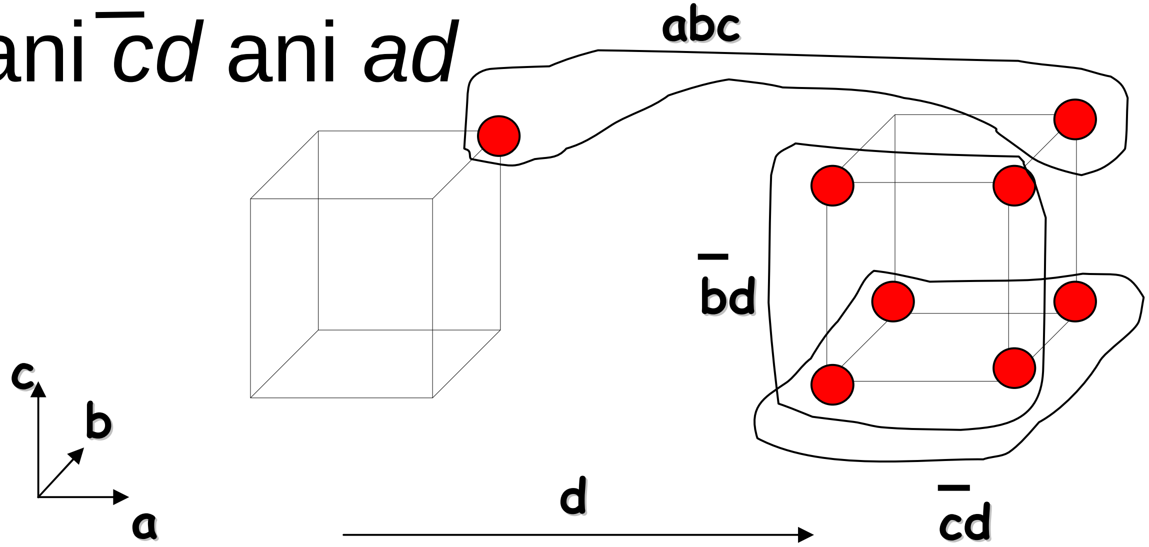
- Přímá krychle z f je **nesporná (podstatná, essential** - essential prime), jestliže obsahuje minterm (jedničkový vrchol), který není obsažen v jiné přímé krychli.

Přímé a neredundantní pokrytí, podstatná krychle

Příklad 4

$f = abc + \bar{b}d + \bar{c}d$ je přímé a neredundantní.

abc je podstatná protože $ab\bar{c}d \in abc$, a $\notin \bar{b}d$ ani $\bar{c}d$ ani ad



Funkce neúplně specifikované

F = (f, d, r) : $B^n \rightarrow \{0, 1, x\}$
kde x reprezentuje “don’t care” (neurčený stav)

- f = onset funkce - f(a)=1 $\leftrightarrow$ F(x)=1
- r = offset funkce - r(a)=1 $\leftrightarrow$ F(x)=0
- d = don’t care funkce - d(a)=1 $\leftrightarrow$ F(x)=x

(f,d,r) tvoří rozdělení (partition) B^n tzn.

- f + d + r = B^n
- fd = fr = dr = $\emptyset$ (pairwise disjoint)

…pokračování ..

Úplně určená funkce g je pokrytí pro $F=(f,d,r)$, jestliže $f \subseteq g \subseteq f+d$

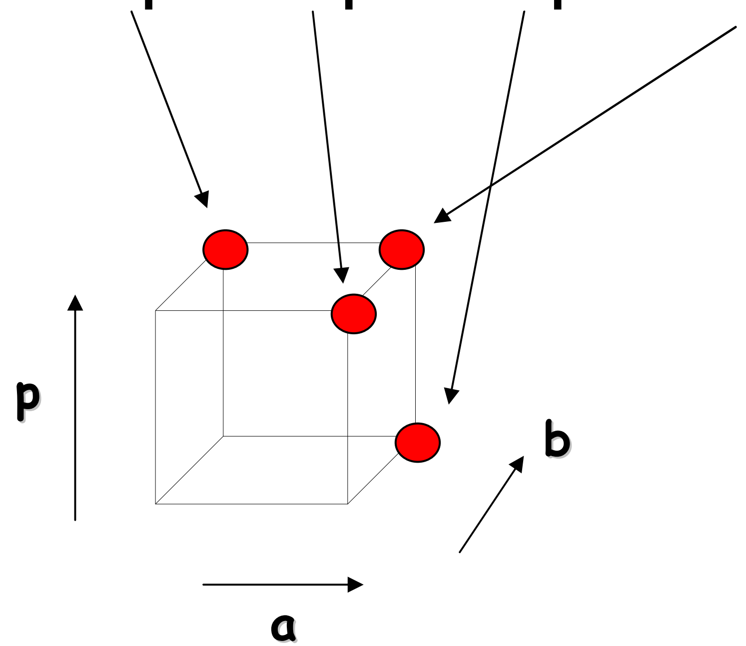
- Když $x \in d$ (tj. $d(x)=1$), potom $g(x)$ je 0 nebo 1,
- Ale když $x \in f$, potom $g(x)=1$
- A když $x \in r$, potom $g(x)=0.$

Příklady

s_i	a	b	p	q	S
0	0	0	0	0	0
1	0	0	1	0	1
2	0	1	0	0	1
3	0	1	1	1	0
4	1	0	0	0	1
5	1	0	1	1	0
6	1	1	0	1	0
7	1	1	1	1	1

SOP – Úplná normální disjunktivní forma

$$q = \bar{a}bp + a\bar{b}p + ab\bar{p} + abp$$



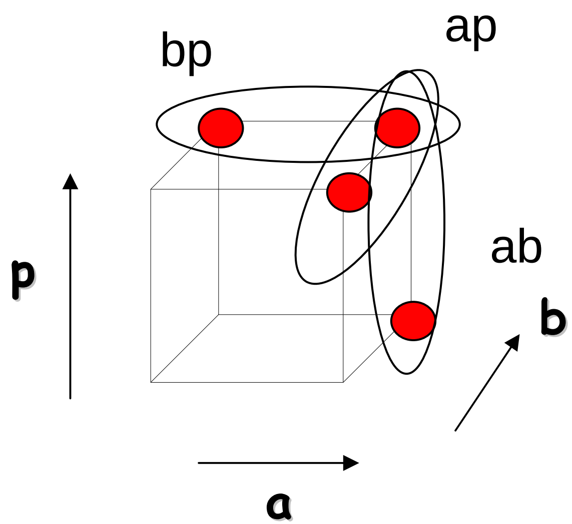
s_i – stavový index

Příklady

s_i	a	b	p	q	S
0	0	0	0	0	0
1	0	0	1	0	1
2	0	1	0	0	1
3	0	1	1	1	0
4	1	0	0	0	1
5	1	0	1	1	0
6	1	1	0	1	0
7	1	1	1	1	1

SOP – Úplná normální disjunktivní forma ÚNDF

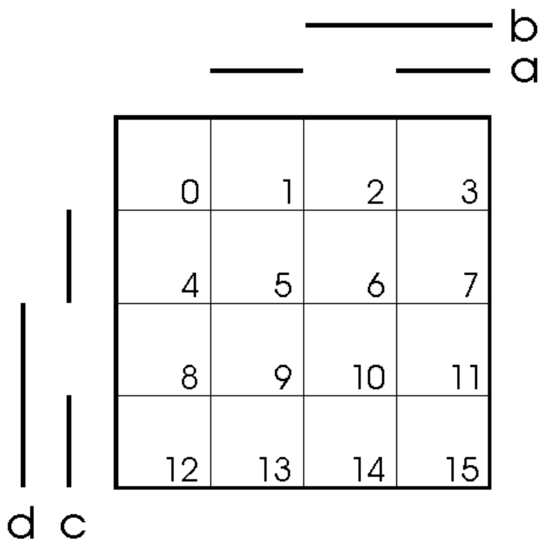
$$q = \bar{a}bp + a\bar{b}p + ab\bar{p} + abp$$



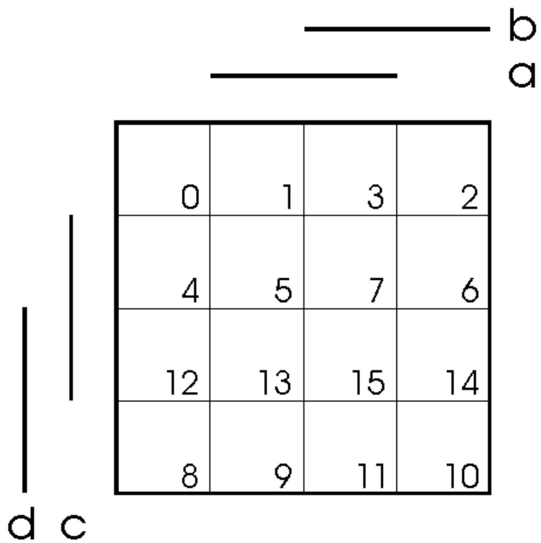
SOP – Minimální normální disjunktivní forma MNDF

$$q = bp + ap + ab$$

Mapy

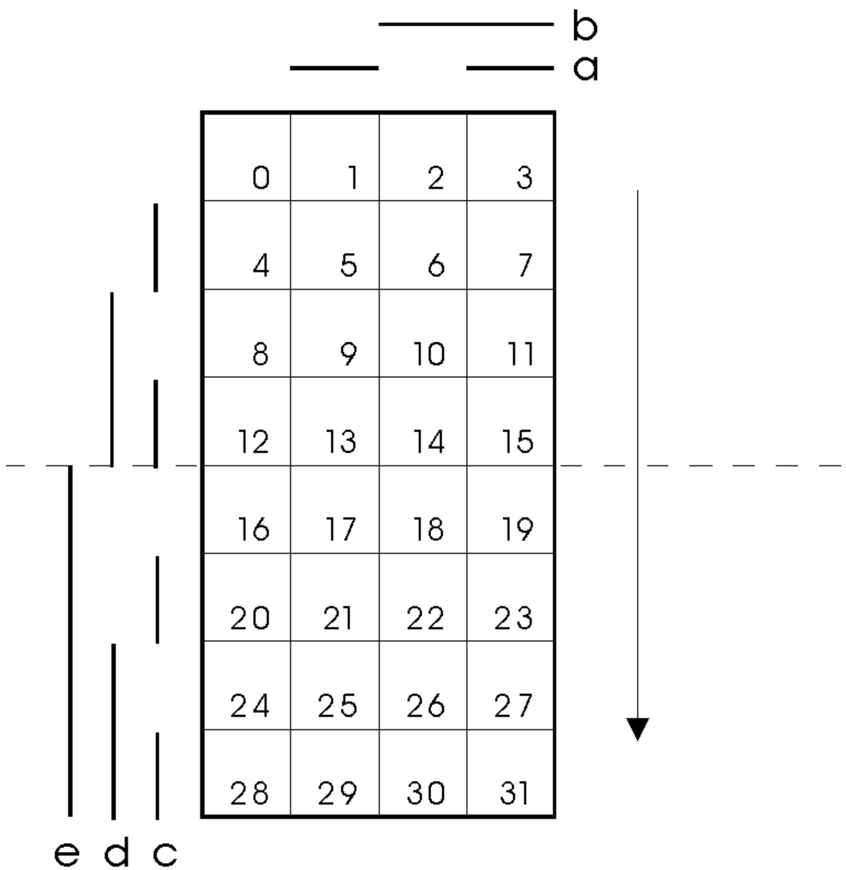


Svobodova mapa

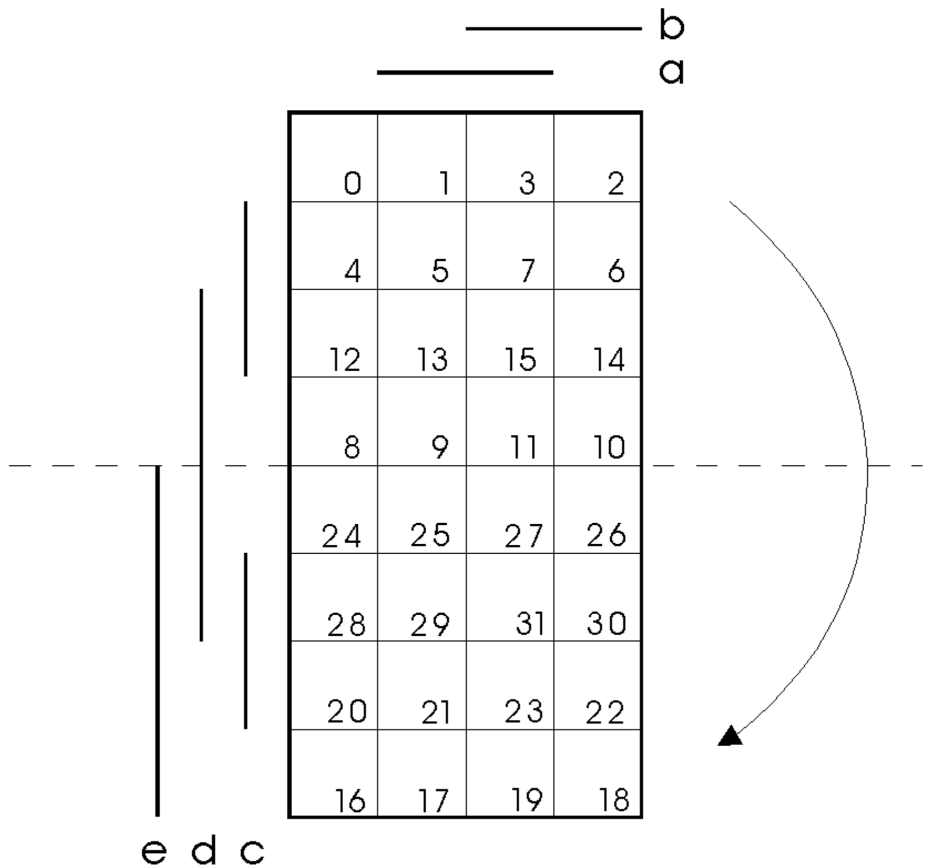


Karnaughova mapa

Změna velikosti mapy – zvyšování počtu proměnných



Svobodova



Karnaughova

Příklady - tabule

- Minimalizace v mapě pro přenos q a součet (nelze, nejsou sousední stavy)
- Příklad funkce určené onset a don´t care