# Struktura a architektura počítačů

Katedra číslicového návrhu
Fakulta informačních technologií
České vysoké učení technické



# Kombinační obvody, logická syntéza

BI-SAP, únor 2021

# Obsah

- Logické obvody kombinační
- Formy popisu a jejich vzájemné vtahy a vlastnosti
- Mapy, příklady a opět sčítačka

Logická syntéza

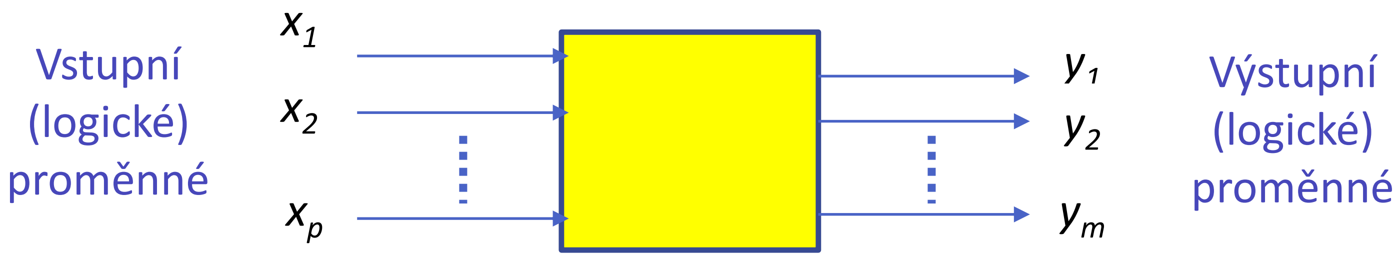
## Cíle přednášky:

- Najít takové vyjádření funkce, které vede k nejlepší implementaci v použité technologii,

protože

**na velikosti a rychlosti záleží**

# Logický obvod




Vstupy a výstupy nabývají pouze hodnot 0 nebo 1

**Kombinační obvod – popsán kombinační funkcí**

Hodnoty všech výstupních proměnných jsou v každém časovém okamžiku určeny pouze hodnotami vstupních proměnných v témže časovém okamžiku

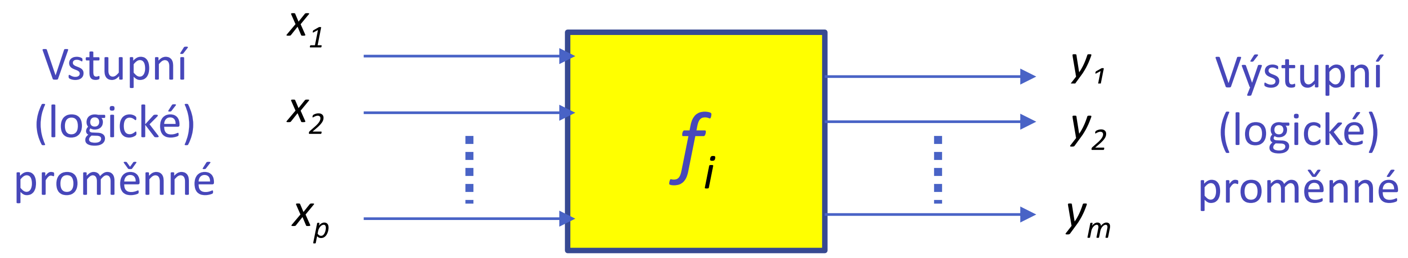
# Charakteristiky

- Dvojkové signály: 0 a 1
- Číslicový návrh
- Číslicové obvody = logické obvody
- Realizace základních bloků číslicového počítače, obecněji: číslicových systémů a jejich komunikace
- Kombinační obvody x sekvenční obvody
- Laboratoře s přípravky s programovatelným hardwarem (FPGA)
- Práce s moderními návrhovými systémy
- Čím jednodušší popis, tím rychlejší realizace

# Problémy k řešení při (jakémkoli) návrhu

- Specifikace – co chceme realizovat
- Optimalizace z různých hledisek
  - Velikost
  - Rychlost
  - Příkon
  - Spolehlivost (*obvykle na úkor velikosti, redundance v prostoru, v čase*)
  - Cena (včetně návrhových prostředků a celého návrhového procesu)
  - Rychlost návrhu *(„time-to-market“) ...* konkurenceschopnost
- Testovatelnost (*DFT = design for testability*)
- Hlavně aby to fungovalo ... ale jak zjistit, že to funguje a za všech okolností (tedy vždy)?

# Kombinační obvod




## Kombinační (logické) funkce:

$$y_1 = f_1(x_1, x_2, x_3, \ldots x_p),$$
$$y_2 = f_2(x_1, x_2, x_3, \ldots x_p),$$
$$\vdots$$
$$y_m = f_m(x_1, x_2, x_3, \ldots x_p)$$

Implementace – realizace (a optimalizace) všech funkcí $f_i$ „najednou“

# Fáze návrhového procesu pro kombinační obvody

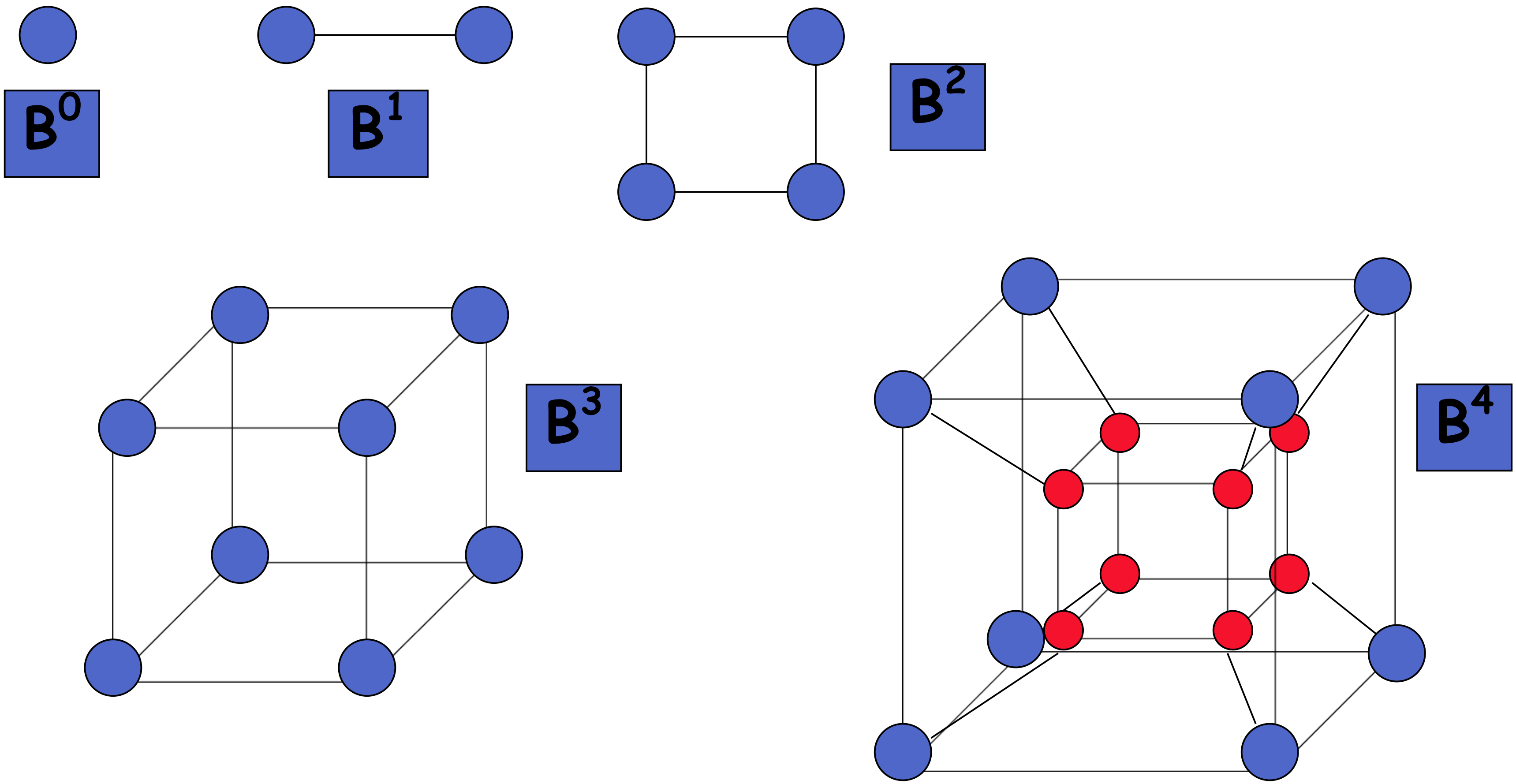
- Specifikace
- Určení vstupů a výstupů
- Pravdivostní tabulky
- Booleovské rovnice
- Návrh realizace na úrovni hradel
- Simulace na úrovni hradel
- Realizace číslicového obvodu
- Ověření návrhu

# Logická syntéza

- Logické funkce a jejich reprezentace, formy popisu a jejich vzájemný převod:
    - tabulka
    - n-rozměrná krychle
    - algebraický zápis
    - mapy
- Logická minimalizace
    - Karnaughova mapa
    - existují další metody ....
- Realizace na úrovni hradel

# Booleovská *n*-krychle (cube) $B^n$

- $B^1 = \{0,1\}$
- $B^2 = \{0,1\} \times \{0,1\} = \{00, 01, 10, 11\}$



# Vyjádření funkce v krychli

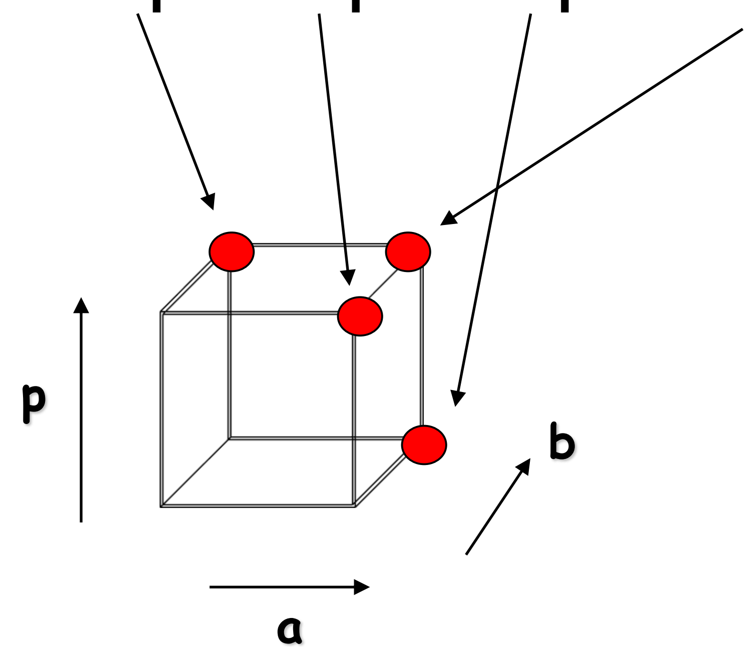
zde sčítačka a jen přenos

| | a | b | p | q | S |
|---|---|---|---|---|---|
| | 0 | 0 | 0 | **0** | 0 |
| | 0 | 0 | 1 | **0** | 1 |
| | 0 | 1 | 0 | **0** | 1 |
| | 0 | 1 | 1 | **1** | 0 |
| | 1 | 0 | 0 | **0** | 1 |
| | 1 | 0 | 1 | **1** | 0 |
| | 1 | 1 | 0 | **1** | 0 |
| | 1 | 1 | 1 | **1** | 1 |

SOP – Úplná normální disjunktivní forma (tvar)

(„Sum of Products“)

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




disjunktivní: “součty součinů“, popisujeme „**1**“

# Logická funkce

$f(x): B^n \longrightarrow B, \quad B = \{0, 1\}$

- $x_1, x_2, \ldots$ proměnné - variables
- $x_1, \overline{x_1}, x_2, \overline{x_2}, \ldots$ literály - literals
- Každému vrcholu $B^n$ je přiřazena „0“ nebo „1“
  - **onset** $f$: $\{x \mid f(x)=1\} = f^1$
  - **offset** $f$: $\{x \mid f(x)=0\} = f^0$
- 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

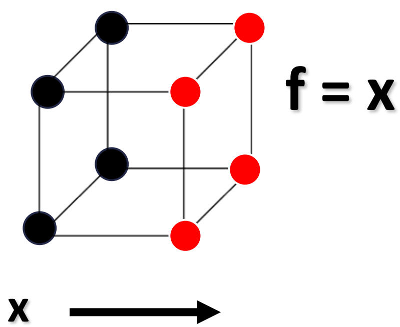
**Literál** je proměnná nebo její negace

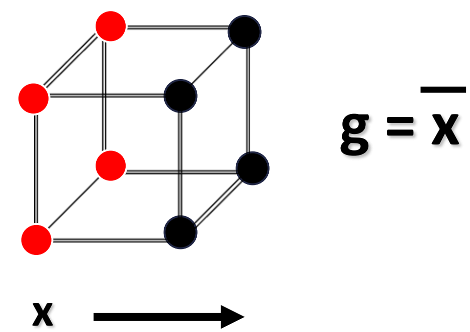
$x_1, \overline{x_1}, a, z, y, \overline{y}, \ldots.$ *input, out*

Literál může reprezentovat logickou funkci.

Příklad: Literál *x* reprezentuje logickou funkci *f*, kde

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




Příklad: Literál *x* reprezentuje logickou funkci *g*, kde

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

Zde vždy polovina vrcholů ....

# Booleovské formule, výrazy

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

- závorek ( , )
- literálů, např. $x, y, z, \overline{x}, \overline{y}, \overline{z}, x_1, x_2, \ldots .$
- Booleovských operátorů $+$ (OR), $\bullet$ (AND)
- negace, např. $\overline{x + y}$

Příklady

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

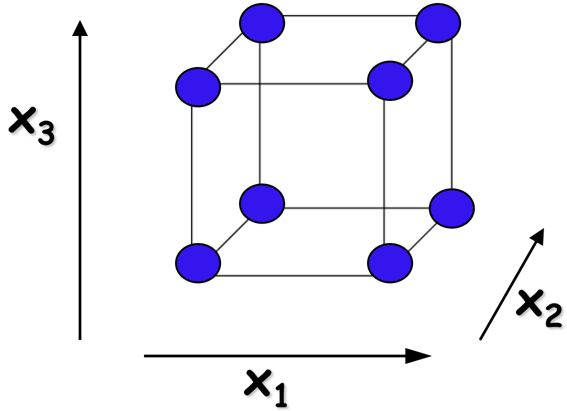
$$h = a + b . c = \overline{\overline{a} . (\overline{b} + \overline{c})}$$

Obvykle nahrazujeme $\bullet$ jen zřetězením, $a \bullet b \to ab$

# Logické funkce … kvantita

$2^n$ vrcholů v prostoru $\boldsymbol{B}^n$.

Zde $n$=3:




$x_1x_2x_3$
000
001
010
011
100
101
110
111

$2^8$ = 256 sloupců
→ 256 různých
funkcí!

Každá podmnožina vrcholů tvoří logickou funkci:

$$f \subseteq \boldsymbol{B}^n$$

Existuje $2^{(2^n)}$ různých logických funkcí ….. *tedy pro n=2 již 16*

# Logické formule

- Ale existuje nekonečně logických **formulí**

$$f = x + y$$
$$= xy + \bar{x}y + x\bar{y}$$
$$= x\bar{y} + \bar{x}y + y$$
$$= (x + \bar{y})(x + y) + \bar{x}y$$

(Tzn., že stejnou funkci, tedy stejnou podmnožinu vrcholů v $\boldsymbol{B}^n$, lze vyjádřit různými způsoby)

- Logická syntéza – nalezení *„nejlepší"* formule (reprezentace) z hlediska cílové platformy

# Úpravy algebraických výrazů

**Shannonův expanzní teorém**:

$$f(a,b,\dots,c)=a.f(1,b,\dots,c)+\bar{a}.f(0,b,\dots,c)$$

***Důkaz***: platí pro všechna *a* (dosaďte) $\forall a \in \{0,1\}$

***Důsledek:***

$$f(a,b,\dots,c)=a.g(b,\dots,c)+\bar{a}.h(b,\dots,c)$$

***Každá logická funkce se dá zapsat pomocí logického součtu, součinu a negace***

Otázka: **Proč je to důležité?**

# Zákony Booleovy algebry

**Booleova algebra *BA*:**

$$BA = \{B, +, \bullet, \bar{\ }, 0, 1\}$$

$$B = \{0,1\}$$

$$x, y, z \in B$$

a platí Huntingtonovy axiomy

*nebo z Matematiky*: Booleova algebra je komplementární distributivní svaz

*Viz přednáška 11 BI-MLO*

| | | |
|---|---|---|
| $x+y=y+x$ | komutativní | $x\cdot y=y\cdot x$ |
| $(x+y)+z=x+(y+z)$ | asociativní | $(x\cdot y)\cdot z=x\cdot(y\cdot z)$ |
| $(x+y)\cdot z=x\cdot z+y\cdot z$ | distributivní | $(x\cdot y)+z=(x+z)\cdot(y+z)$ |
| $x+0=x$ | o neutrálitě 0 a 1 | $x\cdot 1=x$ |
| $x\cdot 0=0$ | o agresivitě 0 a 1 | $x+1=1$ |
| $x+x=x$ | o idempotenci prvků | $x\cdot x=x$ |
| $x+\overline{x}=1$ | vyloučeného třetího | $x\cdot\overline{x}=0$ |
| | dvojí negace | $\overline{\overline{x}}=x$ |
| $\overline{x}+\overline{y}=\overline{x\cdot y}$ | De Morganova pravidla | $\overline{x}\cdot\overline{y}=\overline{x+y}$ |

tyhle zelené jsou Huntingtonovy axiomy

# Princip duality

**platí zákon Z → platí zákon Z***

**Z → Z* :**

+ → •

• → +

0 → 1

1 → 0

Př.: viz. pravidla Booleovy algebry

## Další zákony Booleovy algebry

**absorbce**

$$\boxed{x + x \cdot y = x}$$

$$\boxed{x \cdot (x + y) = x}$$

**absorbce negace**

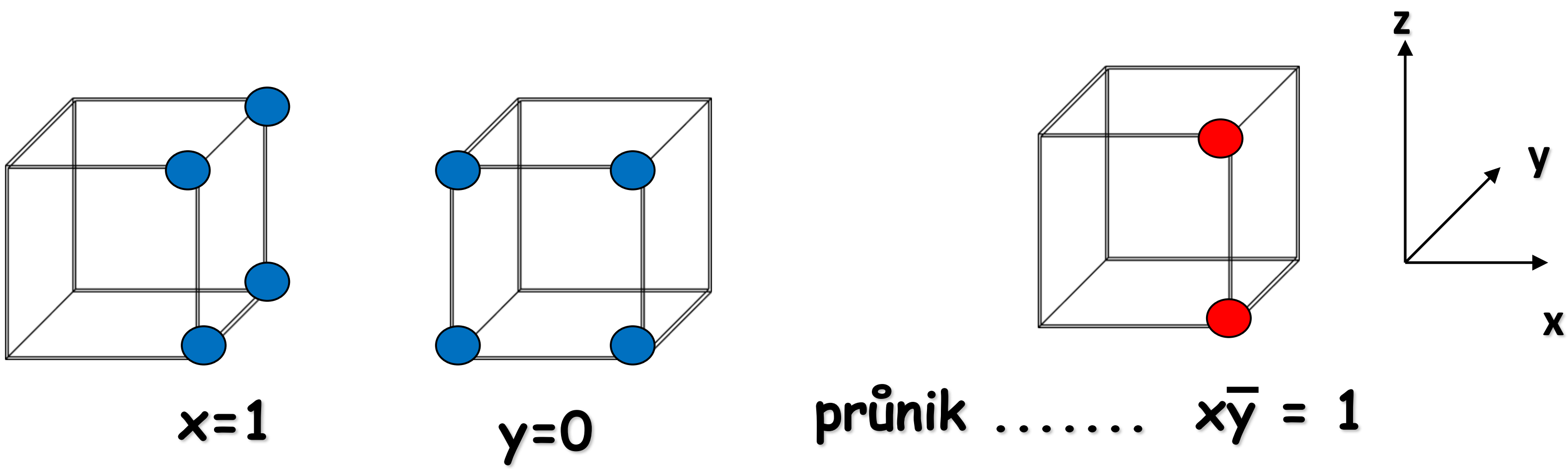
$$\boxed{x + \bar{x} \cdot y = x + y}$$

$$\boxed{x \cdot (\bar{x} + y) = x \cdot y}$$

# Pojem krychle

- Logický součin (AND) literálů *("conjunction/konjunkce")* je tzv. ***krychle*** *(cube, podkrychle, sub-cube)*

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




*Samotný literál je také krychlí*

# Implikant, minterm

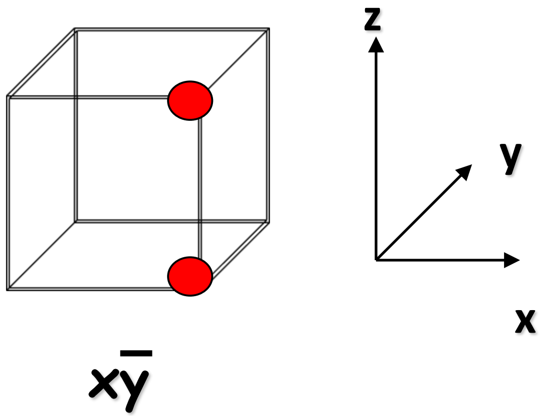
- Jestliže $C \subseteq f$, $C$ je krychle, pak $C$ je implikant $f$.
- Když $C \subseteq B^n$ a $C$ má $k$ literálů, pak $C$ má $\mathbf{2}^{n-k}$ vrcholů.

Příklad $C = x.\bar{y} \subseteq B^3$.

$k = 2$ , $n = 3$.

$C = \{100, 101\}$ .....$(xyz)$

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




- Jestliže $k=n$, pak krychle je minterm (obsahuje všechny proměnné v přímé nebo negované formě) ....

$k = 3$ , $n = 3$: $|C| = 2^{3-3} = 2^0 = 1$ (jeden vrchol)

*Pozor!!! ve strších přednáškách MLO (2016) je pojem minterm totožný s pojmem krychle (tedy s implikantem) a s atomem*

# Pokrytí

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

  $$f = ab + ac + bc$$

  Každá krychle je součin literálů, mluvíme tedy o reprezentaci "**S**um **O**f **P**roducts" – součet součinů SOP …. DNF (disjunktivní normální forma)

- SOP ... je množina krychlí *F*

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

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

# Kanonická reprezentace Booleovských funkcí

- Pravdivostní tabulka funkce $f : B^n \rightarrow B$ je vyjádření jejích hodnot pro všech $2^n$ vrcholů z $B^n$.
- Pro funkci $f$ (zde *úplná* DNF, součet mintermů)

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

| | 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 |

**Pravdivostní tabulka** (truth table):

- nepoužitelná pro velká n,
- ale je kanonická

„Kanonická“ znamená: když jsou dvě funkce stejné, je jejich kanonická reprezentace „stejná“

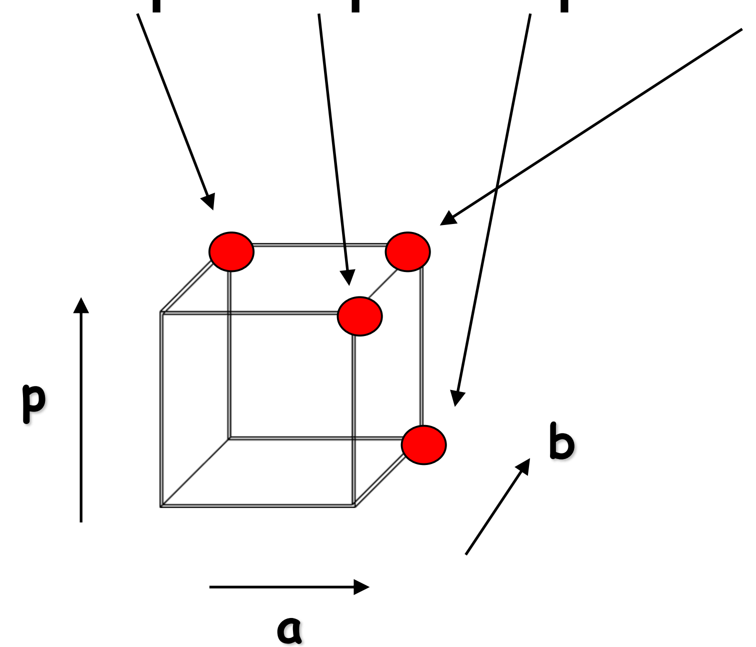
$\rightarrow$ ÚNDF je také kanononická

# Stavový index

| $s_i$ | 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
(Sum of Products)

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




$s_i$ – stavový index:

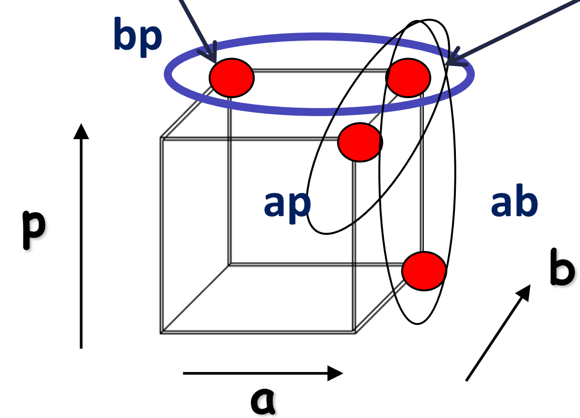
jednoznačně určí funkci i bez tabulky: $q = \sum(3, 5, 6, 7)$

# Součin literálů, (pod)krychle, minimalizace

| $s_i$ | 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**

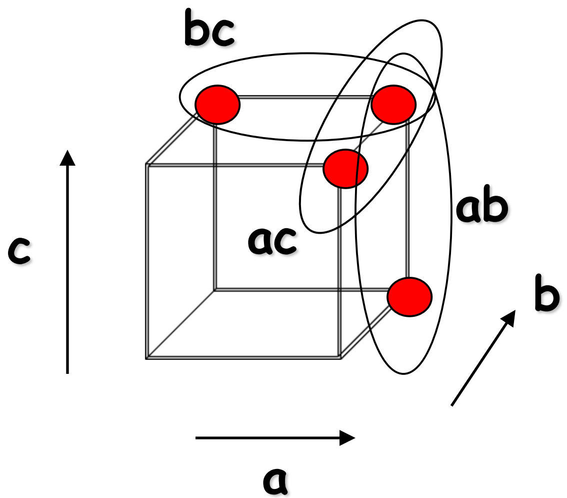
$$q = \bar{a}bp + a\bar{b}p + ab\bar{p} + abp$$




$q$ = ***bp + ap +ab***

SOP – Minimální normální disjunktivní forma MNDF

# Dvouúrovňová minimalizace, pokrytí




**= onset minterm**

**Cíl: každý onset vrchol je "pokrytý" nejméně jednou krychlí a krychle nepokrývá žádný offset (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*, ale co největších ... *přímost*).**

# Neredundance

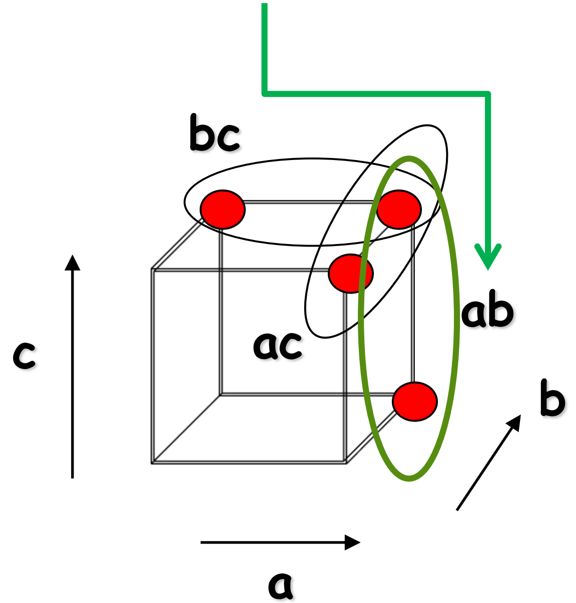
- Nechť $F = \{c_1, c_2, \ldots, c_k\}$ je pokrytí pro $f$.

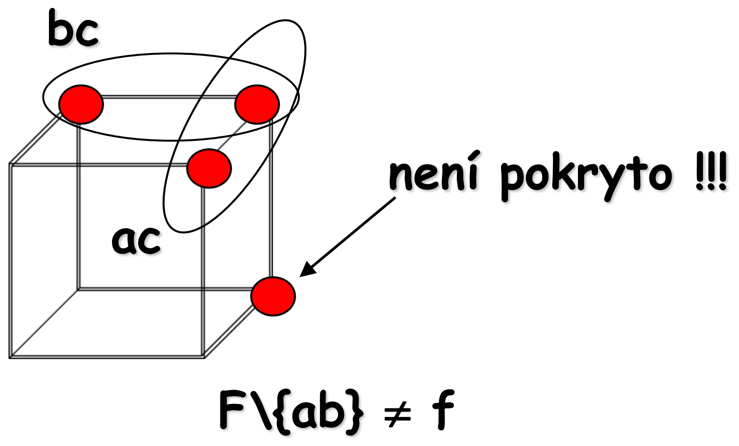
$$f = \sum_{i=1}^{k} c_i$$

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

**(tzn. když ji nezahrnu do řešení, nepokryji všechny onset vrcholy)**

**Příklad 2:** $f = ab + ac + bc$



# 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 je přímá, když všechny její literály jsou přímé (nemohou být vypuštěny).

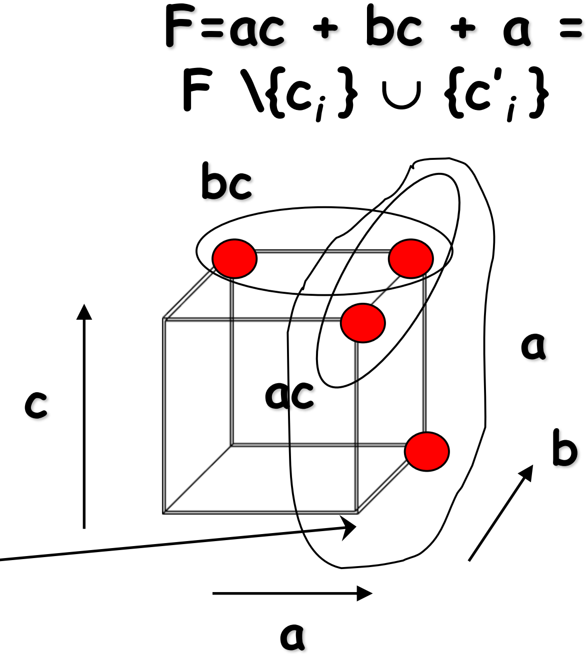
**Příklad**

$f = ab + ac + bc$

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

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

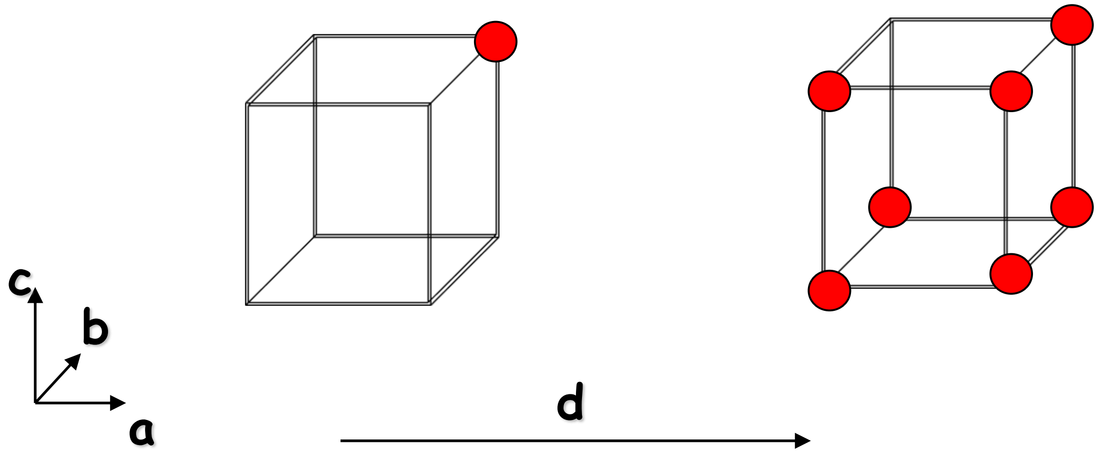
**Nerovná se *f*, protože je pokrytý offsetový vrchol**



# Podstatná krychle

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

Příklad:



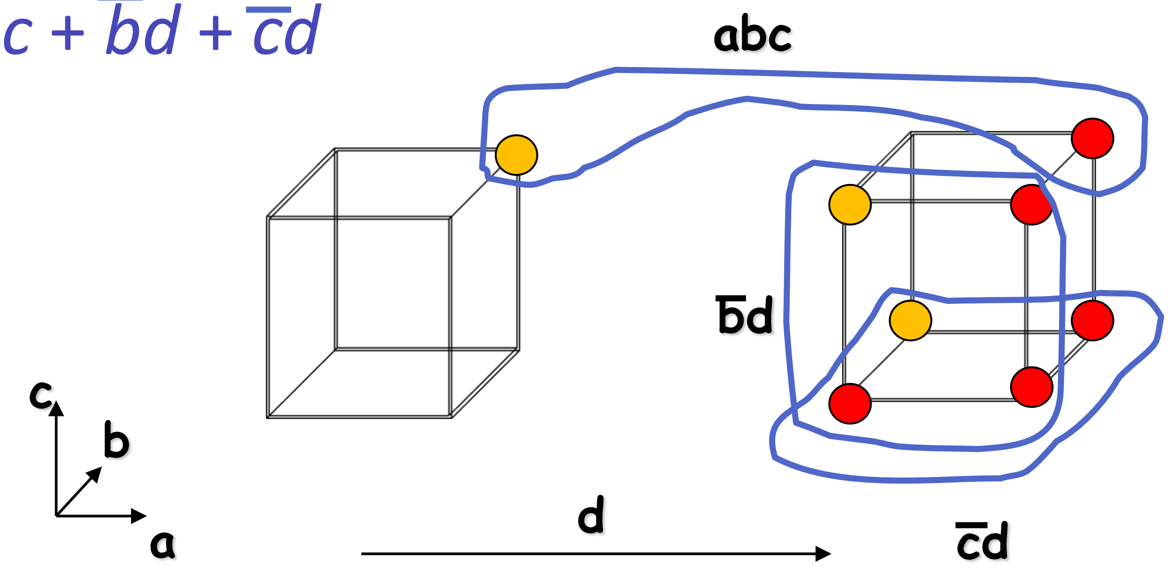
# ... podstatná krychle

Co je přímé a neredundantní pokrytí?

Existují 4 *přímé* krychle: $abc, \overline{b}d, \overline{c}d, ad$

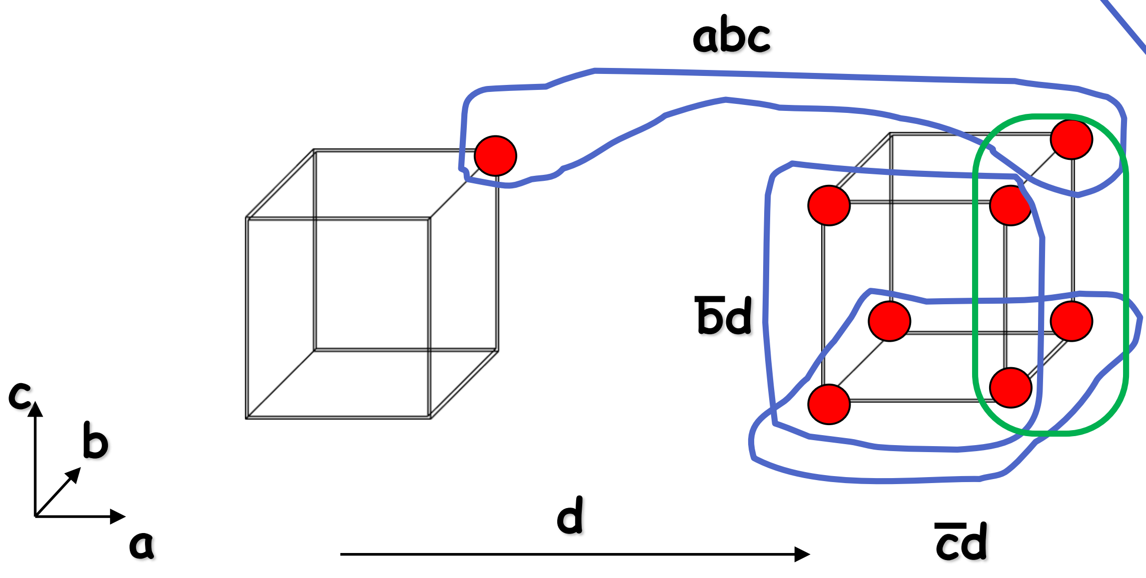
Ale jen 3 jsou *podstatné* a pokrývají $f$ (žluté vrcholy jsou pokryté jen jednou přímou krychlí):

$f = abc + \overline{b}d + \overline{c}d$



# ... podstatná krychle

Existují 4 *přímé* krychle: $abc, \overline{b}d, \overline{c}d, ad$




**krychle *ad* je přímá, ale ne podstatná**

# Mapy

- Formy popisu logických funkcí (jejich vzájemný převod):
  - tabulka + výčet stavových indexů
  - n-rozměrná krychle
  - **mapy**
  - „optimalizovaný“ algebraický zápis

Mapa: převod z n-rozměrné krychle n-D do 2-D

Optimalizace: Hledáme minimální počet co „největších“ krychlí

(největší = popsané minimálním počtem literálů)

# Vztah k algebraickým výrazům

- zviditelnění zákonů Booleovy algebry viz:

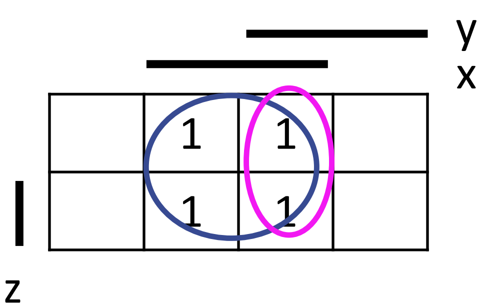
## Další zákony Booleovy algebry

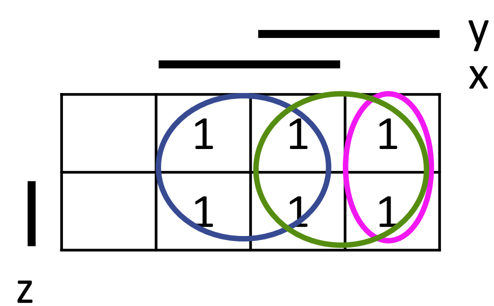
**absorbce**

$$x + x \cdot y = x \qquad x \cdot (x + y) = x$$

**absorbce negace**

$$x + \overline{x} \cdot y = x + y \qquad x \cdot (\overline{x} + y) = x \cdot y$$



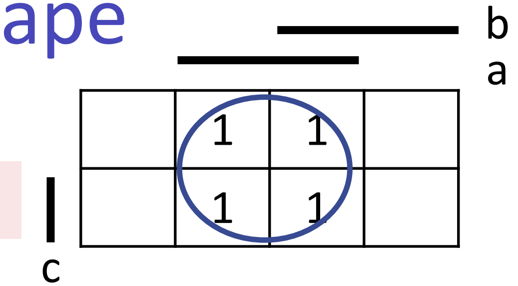



# Minimalizace v Karnaughově mapě

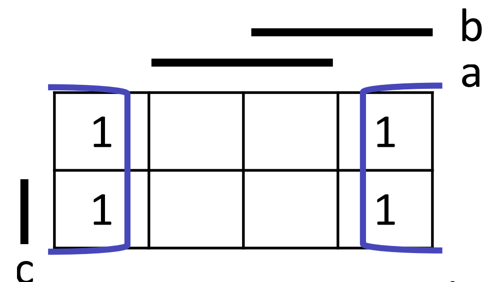
**Zápis do mapy:** podle přiřazení proměnných, tzn. proužků, které vyjadřují kdy nabývá proměnná hodnotu 1:

## POZOR!!

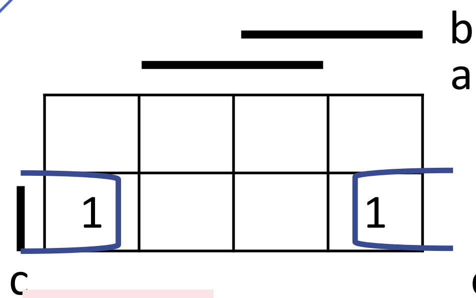
na přiřazení záleží:



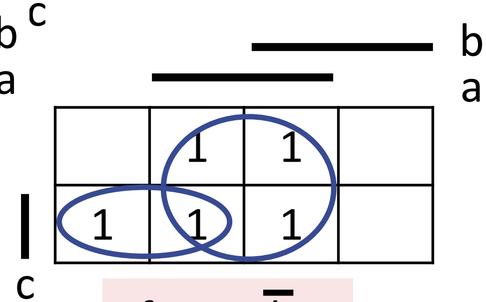

$f = a$



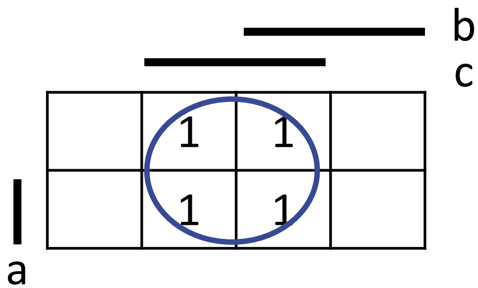

$f = \overline{a}$



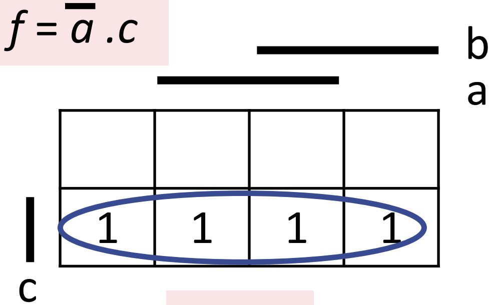

$f = \overline{a}.c$



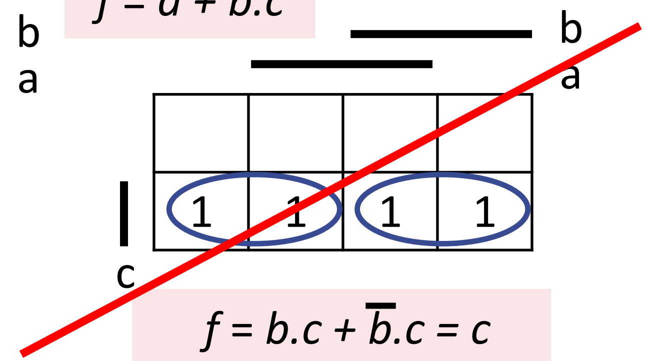

$f = a + \overline{b}.c$




$f = c$



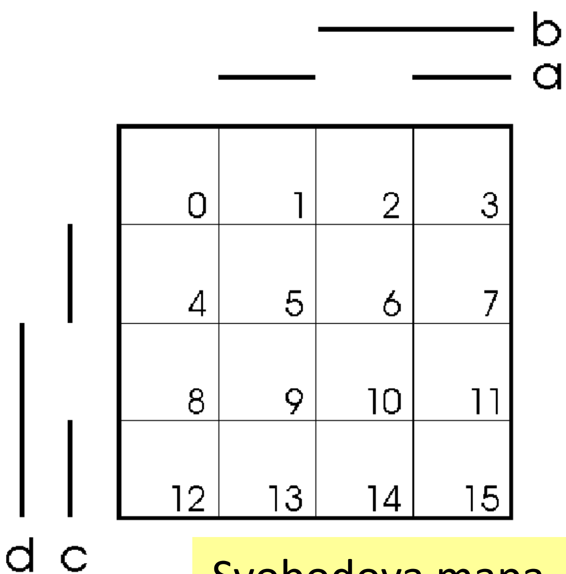

$f = c$




$f = b.c + \overline{b}.c = c$

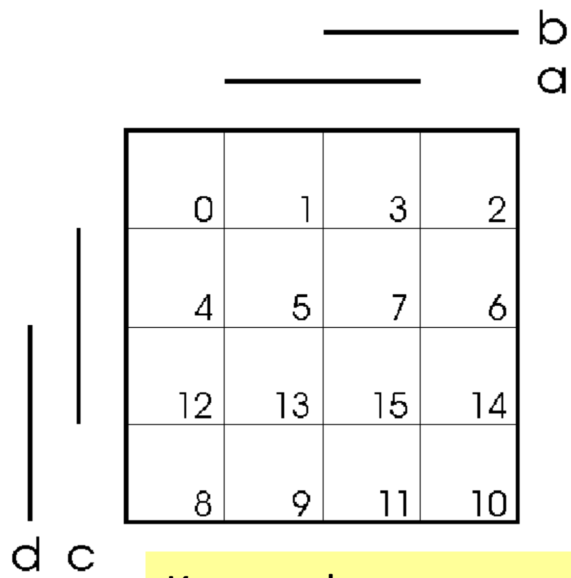
# Mapy a stavové indexy

Typicky: nejnižší váhu má proměnná a (podle abecedy, nebo nejnižší index.
V případě nejasností nebo jiných názvů, je třeba to určit a dodržet konzistenci.




Svobodova mapa

a)

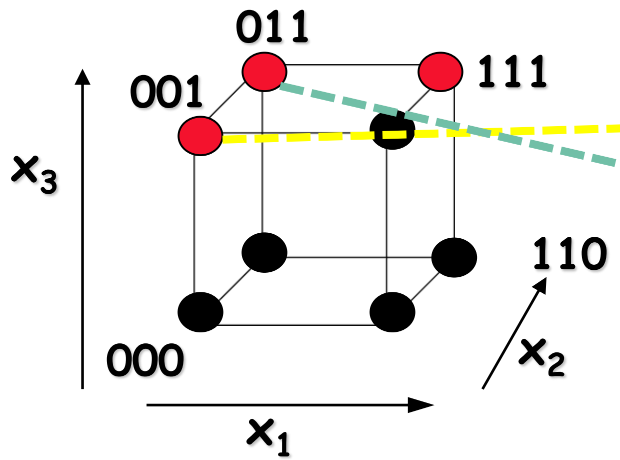



Karnaughova mapa

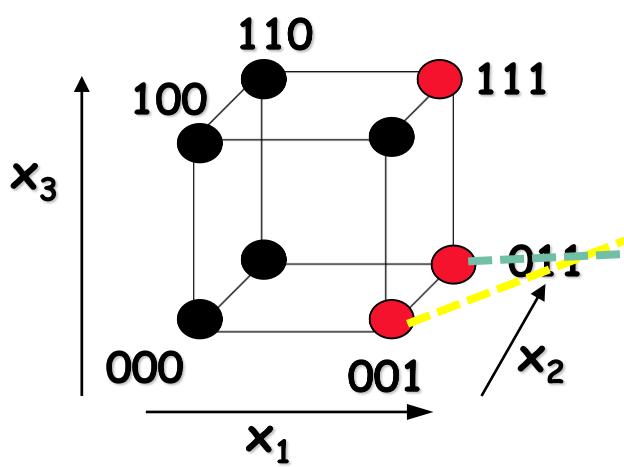
b)

# Konzistence popisu

## Krychle, tabulka $\boldsymbol{B}^n$ zde $n$=3:




| $x_1x_2x_3$ | f |
|---|---|
| 000 | 0 |
| 001 | 1 |
| 010 | 0 |
| 011 | 1 |
| 100 | 0 |
| 101 | 0 |
| 110 | 0 |
| 111 | 1 |




| $x_3x_2x_1$ | g |
|---|---|
| 000 | 0 |
| 001 | 1 |
| 010 | 0 |
| 011 | 1 |
| 100 | 0 |
| 101 | 0 |
| 110 | 0 |
| 111 | 1 |

**Pozor! na pořadí proměnných záleží!!!**

stejná tabulka, různé funkce:

***f*** a ***g***

protože

$x_1x_2x_3$ … f

$x_3x_2x_1$ … g

# Funkce neúplně specifikované

$F = (f, d, r)\colon B^n \to \{0, 1, x\}$

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 ... $d(a)=1 \leftrightarrow F(x)=x$

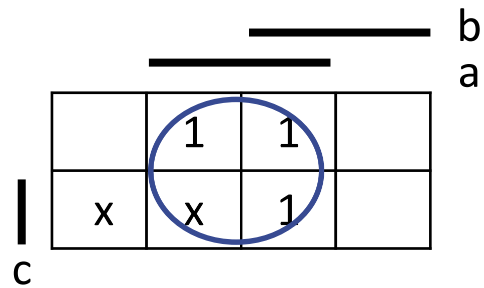
$(f, d, r)$ tvoří rozdělení $B^n$ tzn.

$f + d + r = B^n$

$f \cap d = f \cap r = d \cap r = \emptyset$

# Neurčené stavy

Cíl je pokrýt všechny „1“.
Neurčené stavy pokrýt nemusíme.
Neurčené stavy využíváme,
abychom získali větší krychle.
Může být větší počet řešení.

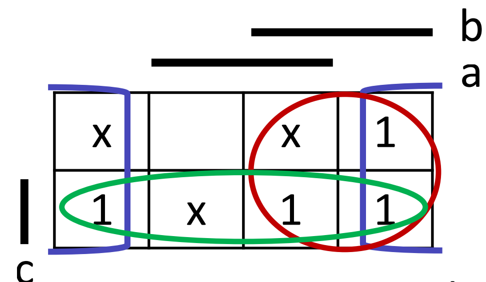


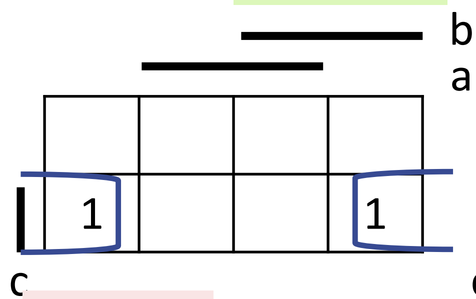
$f = a$

$f = b + c$

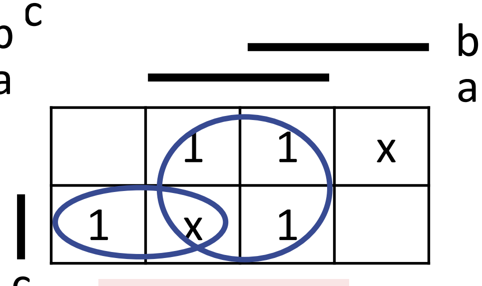
$f = \bar{a} + b$

$f = \bar{a} + c$

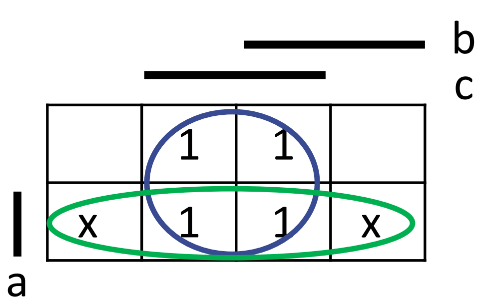




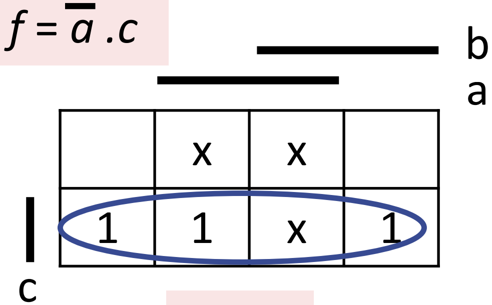
$f = \bar{a}.c$



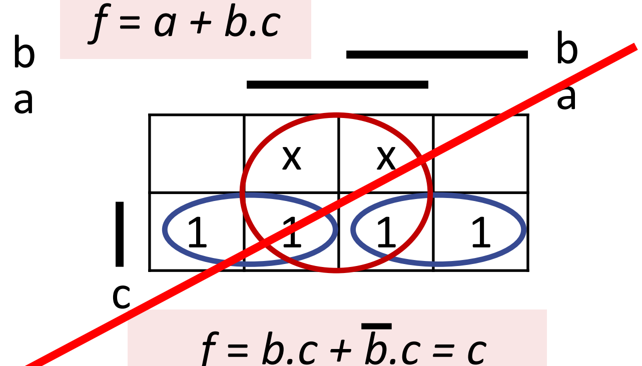
$f = a + \bar{b}.c$



$f = c$

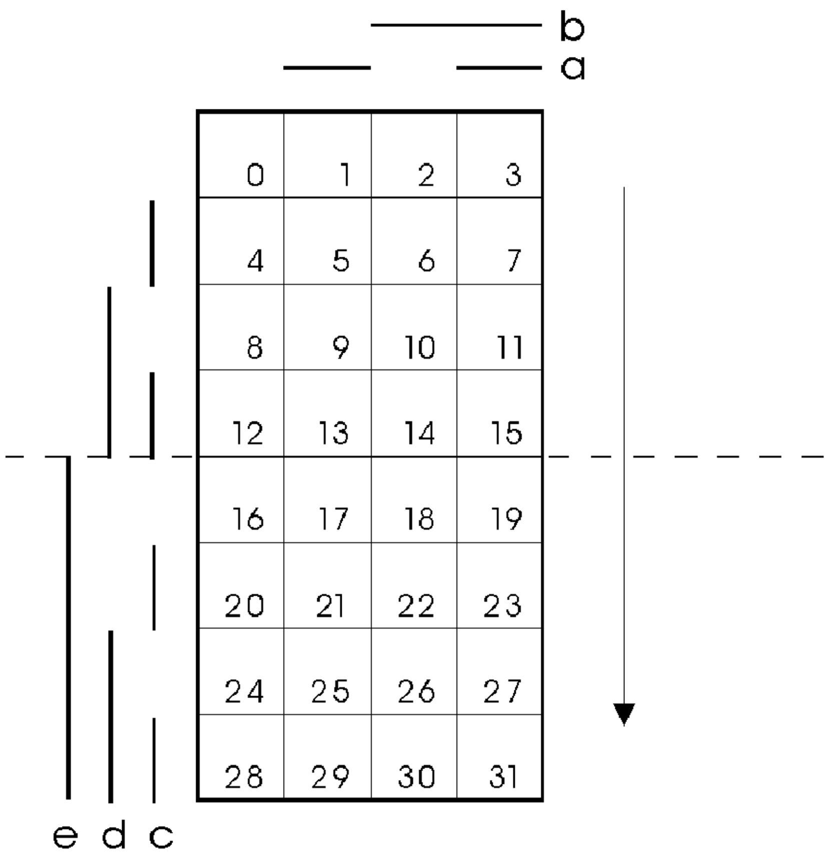


$f = c$

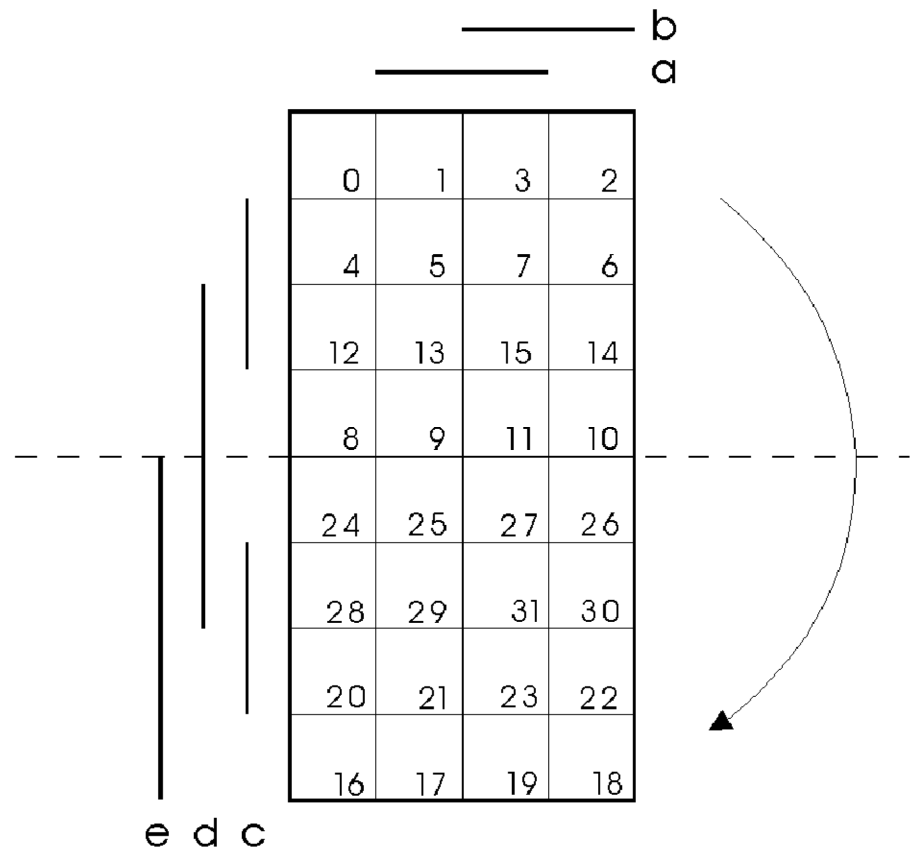


$f = b.c + \bar{b}.c = c$

# Změna velikosti mapy – zvyšování počtu proměnných

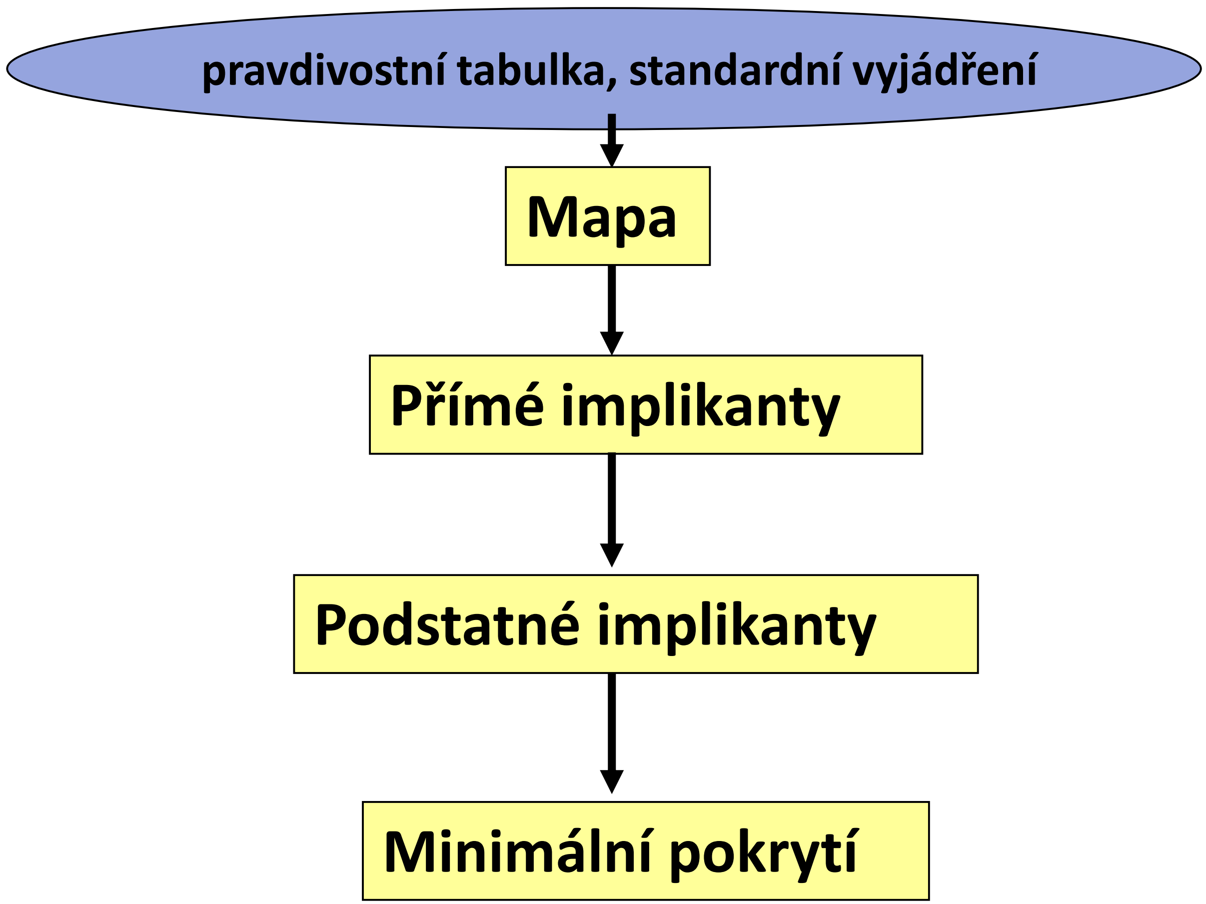



Svobodova: posun




Karnaughova: otočení

# Algoritmus minimalizace v mapě



# Kombinační x sekvenční obvody

- Kombinační – výstup 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

# Programovatelné obvody

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

→ **automatizováno**