12. Booleova algebra, logická funkce určitá a neurčitá, realizace logických funkcí, binární kódy pro algebraické operace.

Logická proměnná - proměnná nesoucí logickou hodnotu
Logická funkce - funkce přiřazující každému vstupnímu písmenu právě jedno výstupní písmeno
Vyjádření log. fce. - mapou (Karnaghova, Svobodova), tabulkou pravdivostních hodnot, N – rozměrným tělesem
Funkce určitá - nabývá předem definované hodnoty
Funkce neurčitá - nemá předem definovanou hodnotu, může nabývat všech stavů (určeno vzorcem nebo ponecháno bez určení)

Základní operace v Boolově algebře:

a´ negace (NOT) *
a+b konjunkce (OR)
a.b disjunkce (AND)
a⊗b nonekvivalence (exklusive OR)
a⇒b implikace
a⇔b ekvivalence

(celkem 16 možných kombinací – uvádíme jen ty nejdůležitější)

a	**b**	**a+b**	**a.b**	**a⊗b**	**a⇒b**	**a⇔b**
0	0	0	0	0	1	1
0	1	1	0	1	1	0
1	0	1	0	1	0	0
1	1	1	1	0	1	1

---

*Pozn. Značení negace je různé, většinou čárou nad písmenem, zde bude z důvodu jednoduššího zápisu značena apostrofem

Zákony platící v Booleově algebře:

- komutativní zákony a + b = b + a
  a.b = b.a
- asociativita zákony (a + b) + c = a + (b + c)
  (a.b).c = a.(b.c)
- distributivita zákony (a + b).c = a.c + b.c
  a.b + c = (a + c).(b + c)
- zákon o vyloučeném třetím a + a' = 1
  a.a' = 0
- zákon o neutrálnosti nuly a + 0 = a
- zákon o neutrálnosti jedničky a·1 = a
- zákon agresivity nuly a·0 = 0

- zákon agresivity jedničky — $a + 1 = 1$
- zákon o idempotenci prvků — $a + a = a$; $a \cdot a = a$
- zákon absorpce — $a + a \cdot b = a$
- zákon absorpce negace — $a + a'.b = a + b$; $a.(a' + b) = a.b$
- zákon dvojité negace — $(a')' = a$
- De Morganovy zákony — $a'.b' = (a + b)'$; $a' + b' = (a.b)'$

Realizace logických funkcí:

Logický člen je základní stavební prvek logických obvodů. Logické členy nazývaná též hradla typicky mají jeden či více vstupů a jeden výstup. Hodnota na výstupu log. členu je funkcí hodnot vstupních: $y = f(x_1, x_2, ..., x_n)$

Existují 2 způsoby značení logických členů:

- Čtvercové značky (DIN)
  - standardizovány dle normy IEEE
  - funkce logického členu je označena znaky
  - „&“ je pro funkci AND
  - „≥1“ pro funkci OR
  - negovaný výstup je označen kolečkem (funkce NOT je proto znázorněna jako jednovstupový OR s negovaným výstupem)

- Značky složené z křivek (ANSI)
  - rozšířené ve většině profesionálních systémů pro návrh logických obvodů

Pomocí základních log. členů AND, OR a NOT lze realizovat libovolný logický obvod a tedy i číslicový systém. Lze ukázat, že funkce AND a OR jsou, za pomoci funkce NOT, komplementární, což znamená, že je lze vhodným způsobem vzájemně nahradit. Lze implementovat jakýkoliv číslicový systém např. pouze za pomoci log. členů AND a NOT (NAND), či OR a NOT (NOR). Stačí použít log. členy AND a OR pouze se dvěma vstupy.

Reálné log. členy mohou být realizovány řadou technologií:

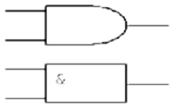
- pomocí relé
- hydraulických ventilů
- elektronek
- rezistorů a diod
- nejčastěji však pomocí tranzistorů

Při práci s reálnými členy je třeba přihlížet k jejich fyzikální podstatě. Musíme uvažovat jejich skutečné rozměry, zabezpečit dodávku energie nezbytné pro jejich činnost, znát rychlost s jakou jsou schopny pracovat, spolehlivost, cenu apod.

AND

Logický součin, značí se "&" nebo "·".

Značení



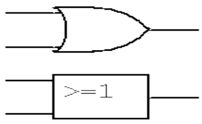
Pravdivostní tabulka

X_1	X_2	Y
0	0	0
0	1	0
1	0	0
1	1	1

OR

Logický součet, značí se "|" nebo "+".

Značení



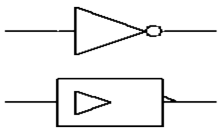
Pravdivostní tabulka

X_1	X_2	Y
0	0	0
0	1	1
1	0	1
1	1	1

NOT

Logická negace, značí se "~", "¬", nadtržítkem, někdy vykřičníkem.

Značení



Pravdivostní tabulka

X	Y
0	1
1	0

Připomenutí minimalizace log. funkcí: (známe všichni ze ZCT ☺**)**

Základní pojmy

- Minterm: a.b.c obsahující všechny vstupní proměnné
- Maxterm: a+b+c obsahující všechny vstupní proměnné
- Disjunktní forma: abc+abc+abc
- Konjunktivní forma: (a+b+c)(a+b+c)(a+b+c)
- Úplná disjunktní forma: výraz ve tvaru součtu mintermů
- Úplná konjunktivní forma: výraz ve tvaru součinu maxtermů

Minimalizace pomocí map

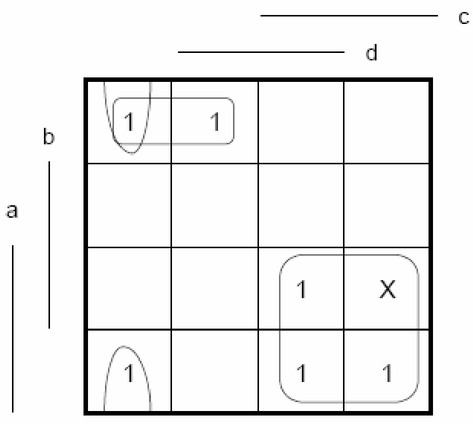
<u>Svobodova mapa</u> - Založena na binárním kódu
- Sousední stavy se nezobrazují do sousedních políček

<u>Karnaughova mapa</u> - Založena na Grayově kódu
- Vytvoření co nejmenšího počtu co největších podmap obepínajících 2i políček (i = celá přirozená čísla)

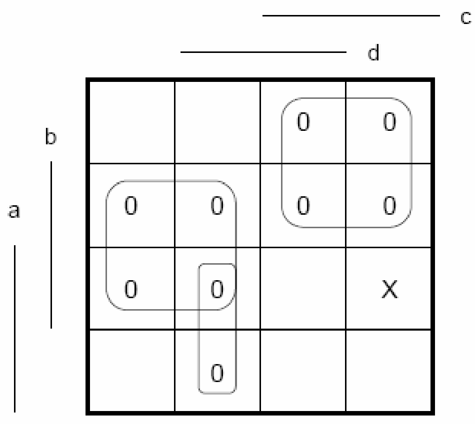
Karnaughova mapa

Funkce: 0, 1, 8, 10, 11, (14), 15

- Minimalizace disjunktní formy
- Minimalizace konjunktivní formy



$\bar{a}\bar{b}\bar{c} + \bar{b}\bar{c}\bar{d} + ac$

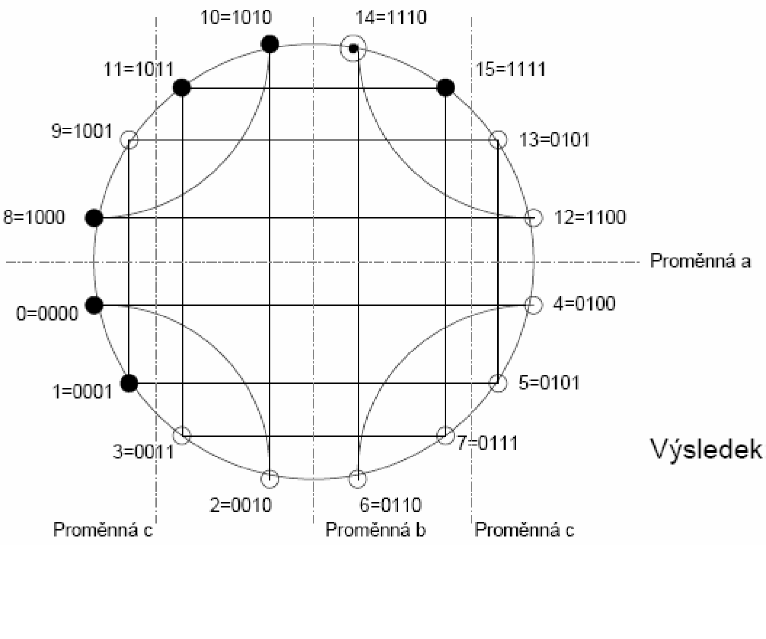


$\overline{b\bar{c}+\bar{a}c+a\bar{c}d}=(\bar{b}+c)(a+\bar{c})(\bar{a}+c+\bar{d})$

N-rozměrná tělesa

- V Grayově kódu – při přechodu po hraně z jednoho vrcholu do druhého změna pouze v jednom bitu (signálu)
- Značení: ●…1 ○…0 ⊙…X

Funkce: 0, 1, 8, 10, 11, (14), 15



	a	b	c	d
10	1	0	1	0
11	1	0	1	1
14	1	1	1	0
15	1	1	1	1

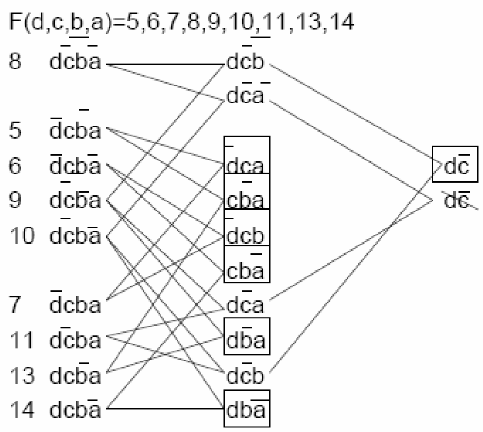
	a	b	c	d
0	0	0	0	0
1	0	0	0	1

	a	b	c	d
0	0	0	0	0
8	1	0	0	0

Výsledek: $ac + \bar{a}\bar{b}\bar{c} + \bar{b}\bar{c}\bar{d}$

Quine Mc Cluskey

1. Sdružení implikantů podle počtu 1 do skupin
2. Výběr dvojic implikantů lišících se právě v jednom bitu
3. Sloučení dvojic – vynecháním bitu v zápisu, kterým se lišily
4. Přidání zbylých implikantů – prosté
5. Tabulka pokrytí



	5	6	7	8	9	10	11	13	14
$d\bar{c}$				x	x	x	x		
$\bar{d}ca$	x		x						
$c\bar{b}a$	x							x	
$\bar{d}cb$		x	x						
$cb\bar{a}$		x							x
$d\bar{b}a$					x			x	
$db\bar{a}$						x			x

Binární kódy pro algebraické operace *:

<table>
<tr><th>Kód</th><th>Reprezentace</th><th>Vlastnosti</th></tr>
<tr><td>Desítkový zápis</td><td>11_d</td><td>Přirozený, hodně logických úrovní</td></tr>
<tr><td>Binární kódování</td><td>1*8+ 0*4 + 1*2+ 1*1
11_d -> 1011_b

12_d -> 1100_b</td><td>Přirozený pro počítač, problém se sousedními stavy</td></tr>
<tr><td>Grayův kód</td><td>$g_{n-1} = b_{n-1}$
g_i = XOR (b_i,b_{i+1})
11_d -> 1110
12_d -> 1010</td><td>Sousední stavy se liší právě v jednom bitu, obtížné aritmetické operace</td></tr>
<tr><td>desítkový kód BCD</td><td>Jako binární kódování</td><td rowspan="4">Desítkové kódy</td></tr>
<tr><td>Stibitzův kód +3</td><td>2_d + 3 = 5 -> 1001</td></tr>
<tr><td>Aikenův kód 2421</td><td>3_d -> 0011
7_d -> 1101</td></tr>
<tr><td>Rubinoffův kód 8,4,-2,-1</td><td>3_d -> 0101
7_d -> 1001</td></tr>
</table>

*Pozn. Kromě kódů pro algebraické operace (výpočty apod.) existuje ještě celá řada kódů pro přenos signálu (např. linkové kódy AMI, Manchester,…). (takže neplést ☺)