Logické řízení

Michal Šerý

Automatizace

Obsah

Výrokový počet

Výrok

– tvrzení, kterému lze jednoznačně přiřadit pravdivostní hodnotu. Při zápisu označujeme obvykle velkým písmenem A, B, ...X, Y, Z.

Pravdivostní hodnota

- F/T, FALSE/TRUE, Pravda/Nepravda
- 0/1, L/H, Low/High

Příklad

- Auto je dopravní prostředek. (TRUE) - je výrok
- Meloun není jedlý. (FALSE) - je výrok
- Jaké je počasí? (???) - není výrok

Booleova algebra

Booleova algebra je univerzální algebra typu $\{\vee, \wedge, \bar{}, 0, 1\}$ $(\{+, \bullet, \bar{}, 0, 1\}$; {OR, AND, NOT, 0, 1}).

Zákony Booleovy algebry

Zákon asociativní

$$A + B + C = A + (B + C) = (A + B) + C = (A + C) + B$$

$$A \cdot B \cdot C = A \cdot (B \cdot C) = (A \cdot B) \cdot C = (A \cdot C) \cdot B$$

Zákon komutativní

$$A + B = B + A$$

$$A \cdot B = B \cdot A$$

Zákon distributivní

$$A \cdot (B + C) = (A \cdot B) + (A \cdot C)$$

$$A + (B \cdot C) = (A + B) \cdot (A + C)$$

Zákon dvojité negace

$$\overline{\overline{A}} = A$$

Zákon vyloučeného třetího

$$A + \overline{A} = 1$$

$$A \cdot \overline{A} = 0$$

Zákony Booleovy algebry

Zákon agresivnosti

$$A + 1 = 1$$

$$A \cdot 0 = 0$$

Zákon neutrálnosti

$$A + 0 = A$$

$$A \cdot 1 = A$$

Zákon absorbce

$$A + A = A$$

$$A \cdot A = A$$

také

$$A + A \cdot B = A$$

$$A \cdot (A + B) = A$$

$$(A + B) \cdot (A + \overline{B}) = A$$

$$A \cdot B + A \cdot \overline{B} = A$$

Zákony Booleovy algebry

Zákon absorbce negace

$$A + \overline{A} \cdot B = A + B$$

$$A \cdot (\overline{A} + B) = A \cdot B$$

Zákon o vytvoření negace - De Morganovy zákony

$$\overline{(A + B)} = \overline{A} \cdot \overline{B}$$

$$\overline{(A \cdot B)} = \overline{A} + \overline{B}$$

Co je logická funkce

Je funkce, která přiřazuje výstupní logickou hodnotu kombinaci vstupních logických proměnných:

$$Y = f(A, B, C, D, ...)$$

Počet logických funkcí

Pro n vstupních logických proměnných lze definovat 2^{2^n} logických funkcí.

Logické funkce pro dvě proměnné

A	0	1	0	1	
B	0	0	1	1	
f_0	0	0	0	0	Nulová funkce $f_0 = 0$
f_1	1	0	0	0	NOR - Pierceova funkce $f_1 = \overline{(A + B)}$
f_2	0	1	0	0	Zpětná inhibice $f_2 = \overline{A}B$
f_3	1	1	0	0	Negace B $f_3 = \overline{B}$
f_4	0	0	1	0	Přímá inhibice $f_4 = \overline{B}A$
f_5	1	0	1	0	Negace A $f_5 = \overline{A}$
f_6	0	1	1	0	Nonekvivalence $f_6 = \overline{A}B + A\overline{B}$
f_7	1	1	1	0	NAND - Shafferova funkce $f_7 = \overline{AB}$
f_8	0	0	0	1	AND $f_8 = AB$
f_9	1	0	0	1	Ekvivalence $f_9 = \overline{A}\overline{B} + AB$
f_{10}	0	1	0	1	Opakování A $f_{10} = A$
f_{11}	1	1	0	1	Přímá implikace $f_{11} = A + \overline{B}$
f_{12}	0	0	1	1	Opakování B $f_{12} = B$
f_{13}	1	0	1	1	Zpětná implikace $f_{13} = \overline{A} + B$
f_{14}	0	1	1	1	OR $f_{14} = A + B$
f_{15}	1	1	1	1	Jedničková funkce

Vyjádření logické funkce

- slovně
- pravdivostní tabulkou
- množinou indexů stavu
- algebraickým výrazem
- mapou

Generátor liché parity

Vytvořte obvod, který bude generovat paritní bit.

Slovní zadání logické funkce

Generátor liché parity

Vytvořte obvod, který bude generovat paritní bit.

Bezpečnostní obvod

Pokud je zmáčknuto více jak jedno tlačítko spusť poplach.

Slovní zadání logické funkce

Generátor liché parity

Vytvořte obvod, který bude generovat paritní bit.

Bezpečnostní obvod

Pokud je zmáčknuto více jak jedno tlačítko spusť poplach.

Sčítačka

Navrhni bitovou sčítačku.

Struktura pravdivostní tabulky

index stavu | vstupní proměnné | výstupní hodnota

i	C	B	A	Y
0	0	0	0	
1	0	0	1	
2	0	1	0	
3	0	1	1	
4	1	0	0	
5	1	0	1	
6	1	1	0	
7	1	1	1	

Definice

Index stavu je dekadické vyjádření „binární“ kombinace vstupních hodnot.

Vyjádření logické funkce jako množinu indexů stavu

Je to množina, do které zahrneme ty indexy, pro které funkce nabývá hodnoty **1**.

Logická funkce - příklad

Lichá parita - pravdivostní tabulka

i	C	B	A	Y
0	0	0	0	1
1	0	0	1	0
2	0	1	0	0
3	0	1	1	1
4	1	0	0	0
5	1	0	1	1
6	1	1	0	1
7	1	1	1	0

Lichá parita - množina indexů stavu

i	C	B	A	Y
0	0	0	0	1
1	0	0	1	0
2	0	1	0	0
3	0	1	1	1
4	1	0	0	0
5	1	0	1	1
6	1	1	0	1
7	1	1	1	0

$$Y = \{0; 3; 5; 6\}$$

Libovolnou funkci $f(a, b, c, \ldots)$ lze zapsat ve dvou základních tvarech.

Základní součtový tvar (též úplná normální disjunktivní forma)

Funkci zapsanou v základním součtovém tvaru dostaneme jako součet základních součinů přímých nebo negovaných proměnných. Zapisujeme pouze základní součiny (mintermy) u těch kombinací přímých nebo negovaných proměnných, u kterých má funkce hodnotu **1**; v mintermu zapisujeme proměnné, které nabývají v příslušné kombinaci hodnoty **1** jako přímé, a proměnné, které nabývají hodnoty **0** jako negované.

Například

$$f = A\overline{B}C + AB\overline{C}$$

Základní součinový tvar (též úplná normální konjunktivní forma)

Funkci zapsanou v základním součinovém tvaru dostaneme jako součin základních součtů přímých nebo negovaných proměnných. Zapisujeme pouze základní součty (maxtermy) u těch kombinací přímých nebo negovaných proměnných, u kterých má funkce hodnotu **0**; v maxtermu zapisujeme proměnné, které nabývají v příslušné kombinaci hodnoty **0** jako přímé, a proměnné, které nabývají hodnoty **1** jako negované.

Například

$$f = (A + \overline{B} + C)(A + B + \overline{C})$$

Logická funkce - příklad

Lichá parita - algebraický výraz (disjunktivní forma)

C	B	A	Y
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

$$Y = \bar{A}\bar{B}\bar{C} + AB\bar{C} + A\bar{B}C + \bar{A}BC$$

Tato forma se používá se nejčastěji.

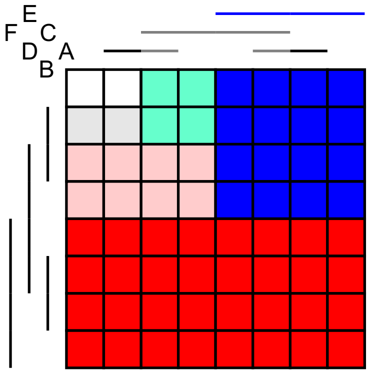
Lichá parita - algebraický výraz (konjunktivní forma)

C	B	A	Y
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

$$Y = (\bar{A}+B+C)(A+\bar{B}+C)(A+B+\bar{C})(\bar{A}+\bar{B}+\bar{C})$$

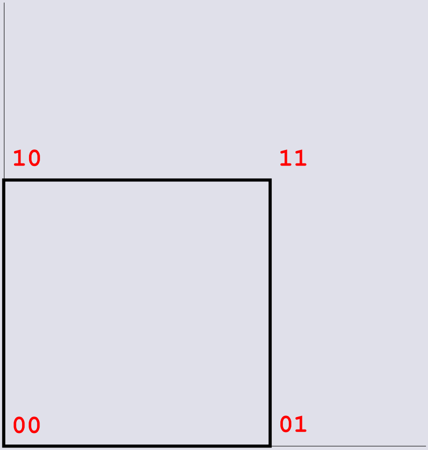
Karnaughova mapa

Vznik mapy



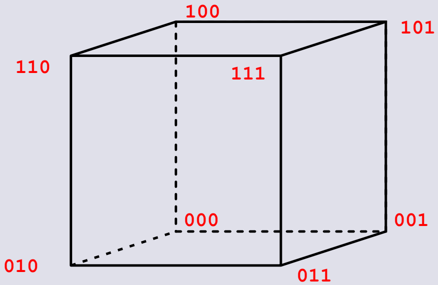
Kódová vzdálenost 2 proměnné

Kódová krychle 2D

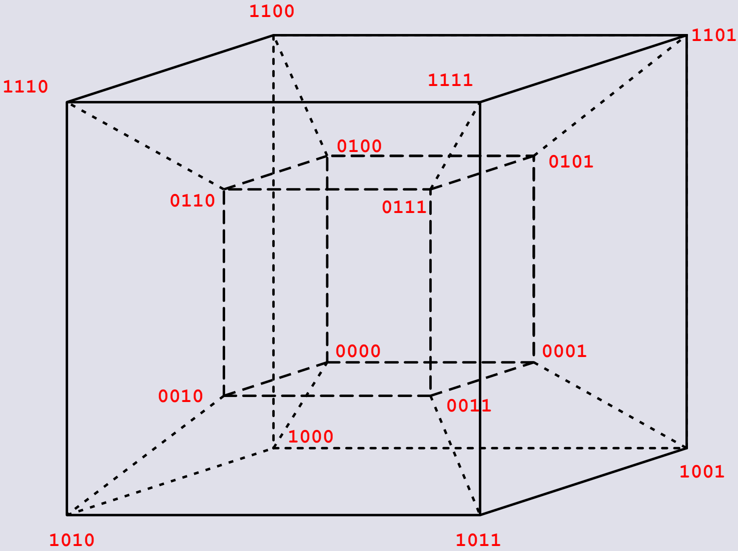


Kódová vzdálenost 3 proměnné

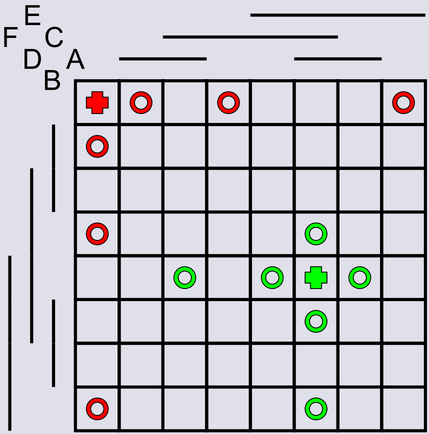
Kódová krychle 3D



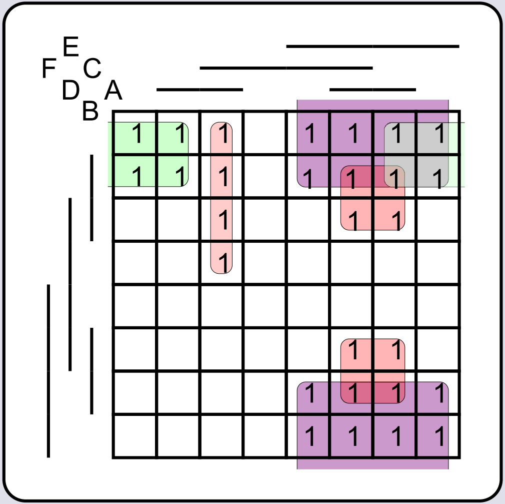
Kódová krychle 4D



Sousedi v mapě



Smyčky



$$f = \bar{C}\bar{D}\bar{E}\bar{F} + A C \bar{E}\bar{F} + A B E + E \bar{D}$$

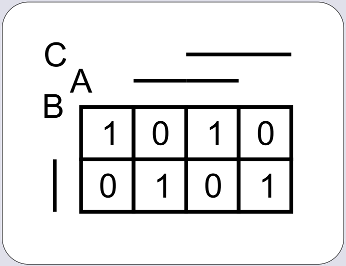
Popis smyček

$$f = \bar{C}\bar{D}\bar{E}\bar{F} + AC\bar{E}\bar{F} + ABE + E\bar{D}$$

Logická funkce - příklad

Lichá parita - Karnaughova mapa

C	B	A	Y
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0



Základní logické funkce

Negace

Pravdivostní tabulka:

A	Y
0	1
1	0

Algebraický výraz: $Y = \overline{A}$

Logický součet

Pravdivostní tabulka:

B	A	Y
0	0	0
0	1	1
1	0	1
1	1	1

Algebraický výraz: $Y = A + B$

Logický součin

Pravdivostní tabulka:

B	A	Y
0	0	0
0	1	0
1	0	0
1	1	1

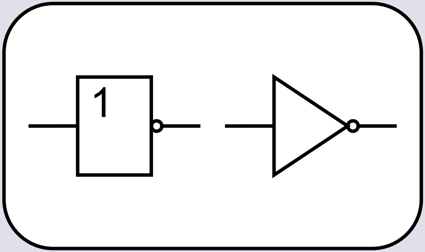
Algebraický výraz: $Y = A \cdot B$

Realizace logické funkce

K realizaci logických funkcí se používají tzv. logické obvody. Ty mohou být například:

- releové
- pneumatické
- diodové
- tranzistorové (bipolární, CMOS)
- na bázi IO

Negace: NOT



Logický součet: OR

fig/OR-eps-converted-to.pdf

Schématické značky logických funkcí

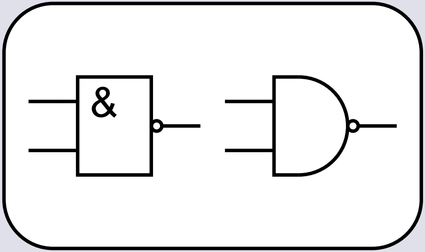
Logický součin: AND

fig/AND-eps-converted-to.pdf

Negovaný logický součet - Peirceova funkce: NOR

fig/NOR-eps-converted-to.pdf

Negovaný logický součin - Shefferova funkce: NAND



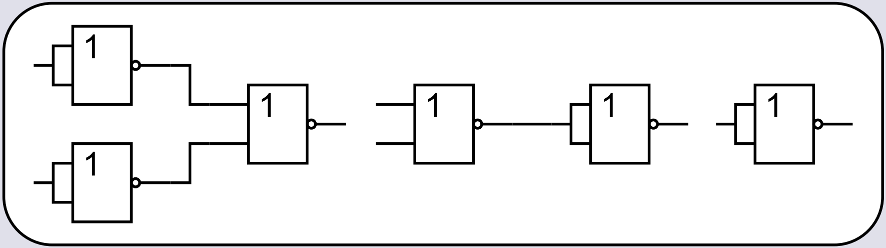
Shefferova funkce: NAND

AND OR NOT

fig/SHEF-eps-converted-to.pdf

Shefferova funkce: NOR

AND OR NOT



(LD, Ladder Diagram, jazyk kontaktních schémat) a jazyk funkčního blokového schématu (FBD, Function Block Diagram).

This text is in a normal font. This text is large. **This text is large and bold. This text is still large and bold. This text is only bold, but not large.** This text is small and bold.