

Logické řízení

Michal Šerý

Automatizace

- 1 **Výrokový počet**
 - Booleova algebra
- 2 **Základní logické funkce**
- 3 **Zákony Booleovy algebry**
- 4 **Logické funkce**

5 Vyjádření logické funkce

- Slovní zadání logické funkce
- Pravdivostní tabulka
- Index stavu
 - Příklad
- Algebraický výraz
 - Normální forma disjunktivní
 - Normální forma konjunktivní
 - Příklad
- Mapy
 - Kódová vzdálenost
 - Smyčky

6 Logické obvody

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}).

Logický součet

Logický součet proměnných nabývá hodnoty **1** pokud **alespoň jedna** z proměnná nabývá hodnoty **1**.

Logický součin

Logický součin proměnných nabývá hodnoty **1** pokud **všechny** proměnné nabývají hodnoty **1**.

Negace

Logická negace proměnné nabývá hodnoty **1** pokud proměnná nabývá hodnoty **0** a nabývá hodnoty **0** pokud proměnná nabývá hodnoty **1**.
Vrací opačnou hodnotu.

Negace

Pravdivostní tabulka:

A	Y
0	1
1	0

Algebraický výraz: $Y = \bar{A}$

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: $Y = A \cdot B$

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

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

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

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

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

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

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

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

Zákon absorbce negace

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

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

Zákon o vytvoření negace - De Morganovy zákony

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

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

Zákon absorbce negace

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

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

Zákon o vytvoření negace - De Morganovy zákony

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

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

Axiomy a odvozené vlastnosti

- idempotence: $A + A = A, A \cdot A = A$
- komutativita: $A + B = B + A, A \cdot B = B \cdot A$
- asociativita: $A + (B + C) = (A + B) + C, A \cdot (B \cdot C) = (A \cdot B) \cdot C$
- absorpce: $A \cdot (A + B) = A, A + (A \cdot B) = A$
- distributivita: $A \cdot (B + C) = (A \cdot B) + (A \cdot C), A + (B \cdot C) = (A + B) \cdot (A + C)$
- neutralita 0 a 1: $A + 0 = A, A \cdot 1 = A$
- agresivita 0 a 1: $A + 1 = 1, A \cdot 0 = 0$
- komplementarita: $\bar{A} + A = 1, \bar{A} \cdot A = 0$
- absorpce negace: $A \cdot (\bar{A} + B) = A \cdot B, A + (\bar{A} \cdot B) = A + B$
- DeMorganovy zákony: $\overline{A + B} = \bar{A} \cdot \bar{B}, \overline{A \cdot B} = \bar{A} + \bar{B}$
- dvojitá negace: $\overline{\bar{A}} = A$

Co je logická funkce

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

$$Y = f(A, B, C, D, \dots)$$

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 $f_2 = \overline{AB}$
f_3	1	1	0	0	Negace B $f_3 = \overline{B}$
f_4	0	0	1	0	Přímá inhibice $f_4 = \overline{BA}$
f_5	1	0	1	0	Negace A $f_5 = \overline{A}$
f_6	0	1	1	0	Nonekvivalence $f_6 = \overline{AB} + A\overline{B}$
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 $f_9 = \overline{AB} + A\overline{B}$
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

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

Generátor liché parity

Vytvořte obvod, který bude generovat paritní bit.

Generátor liché parity

Vytvořte obvod, který bude generovat paritní bit.

Bezpečnostní obvod

Pokud je zmáčknuto více jak jedno tlačítko spust' poplach.

Generátor liché parity

Vytvořte obvod, který bude generovat paritní bit.

Bezpečnostní obvod

Pokud je zmáčknuto více jak jedno tlačítko spust' poplach.

Sčítačka

Navrhni bitovou sčítačku.

Struktura pravdivostní tabulky

index vstupní výstupní
stavu proměnné 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.

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 $f(a, b, c, \dots)$ 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

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

Základní součinný tvar (též úplná normální konjunktivní forma)

Funkci zapsanou v základním součinném 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

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

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

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

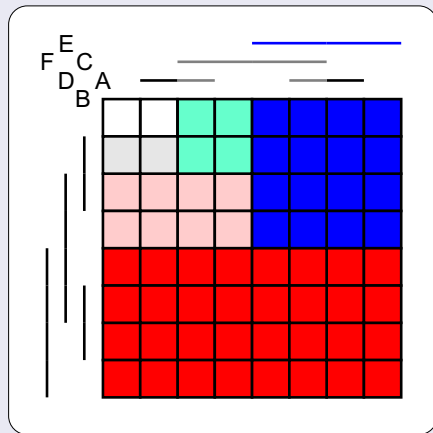
Tato forma se používá se nejčastěji.

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

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

Vznik mapy



Průmět pruhů

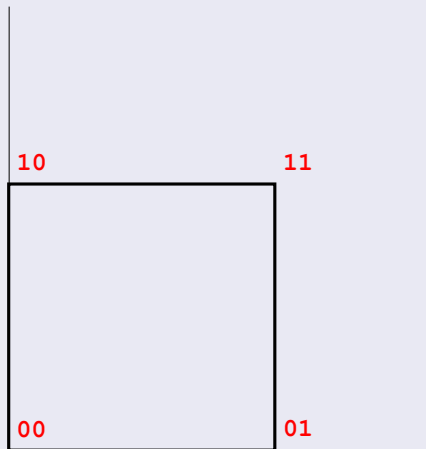
C

A

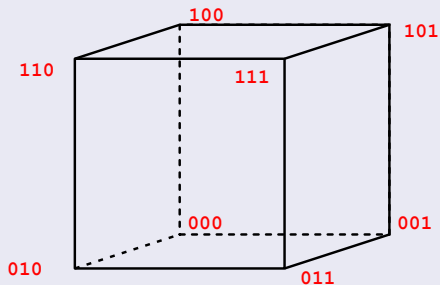
B

CBA	CBA	CBA	CBA
000	001	101	100
010	011	111	110

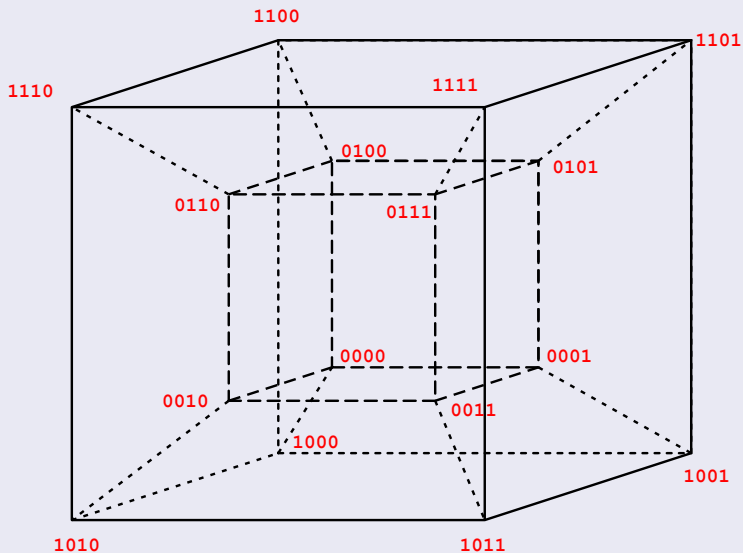
Kódová krychle 2D



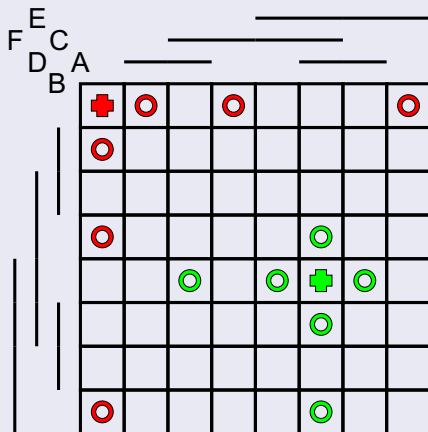
Kódová krychle 3D



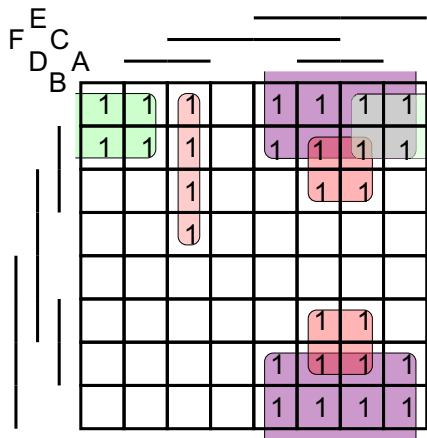
Kódová krychle 4D



Sousedí v mapě



Smyčky



$$f = \bar{C}\bar{D}\bar{F} +$$

$$AC\bar{E}\bar{F} +$$

$$ABE +$$

$$E\bar{D}$$

Pravidla pro smyčky

- smyčky se tvoří pro **1**
- na konci každá **1** musí být alespoň v jedné smyčce
- zahrnuje pouze **1** nebo neurčité stavy
- velikost smyčky 2^n to jest 1, 2, 4, 8, 16
- zahrnuje pouze nejbližší sousedy

Popis smyček

Do popisu zahrneme pouze takové proměnné, které na všech polích smyčky mají stejnou hodnotu. Ty proměnné, které nabývají hodnoty **1** píšeme do součinu přímo a ty které **0** píšeme s negací.

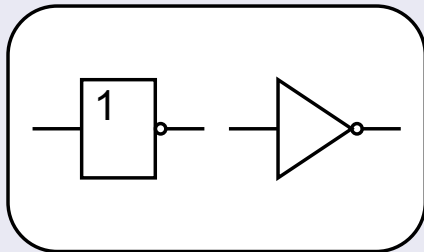
Popis smyček

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

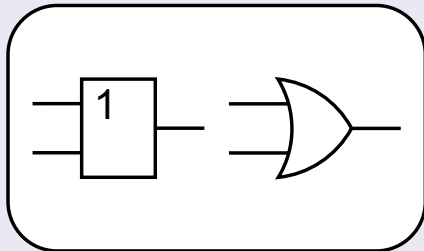
K realizaci logických funkcí se používají tzv. logické obvody. Ty mohou být například:

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

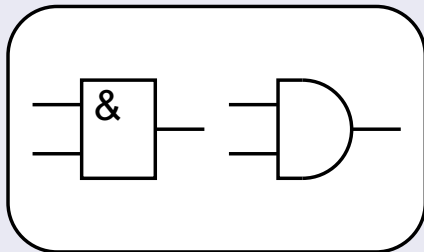
Negace: NOT



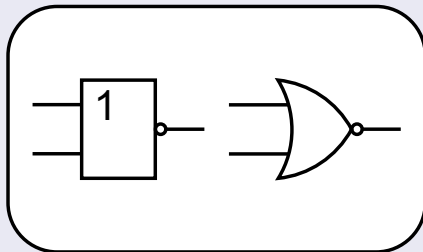
Logický součet: OR



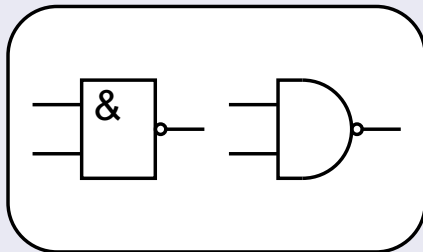
Logický součin: AND



Negovaný logický součet - Peirceova funkce: NOR



Negovaný logický součin - Shefferova funkce: NAND

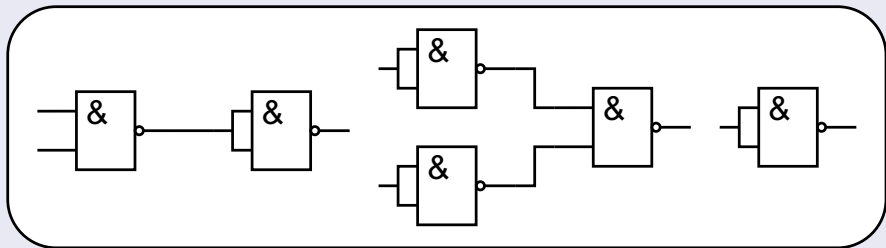


Shefferova funkce: NAND

AND

OR

NOT

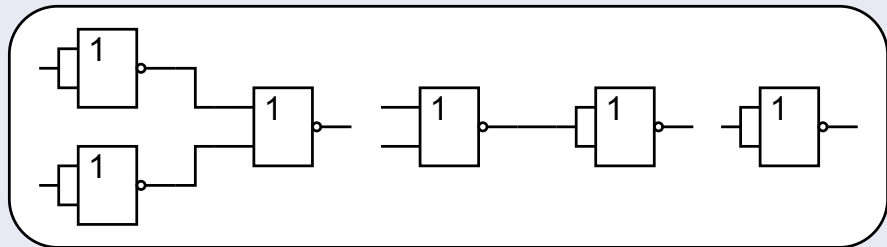


Shefferova funkce: NOR

AND

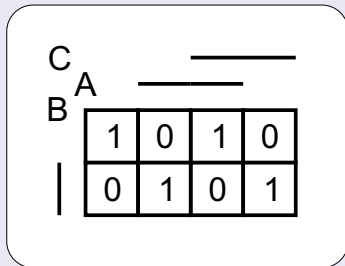
OR

NOT



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

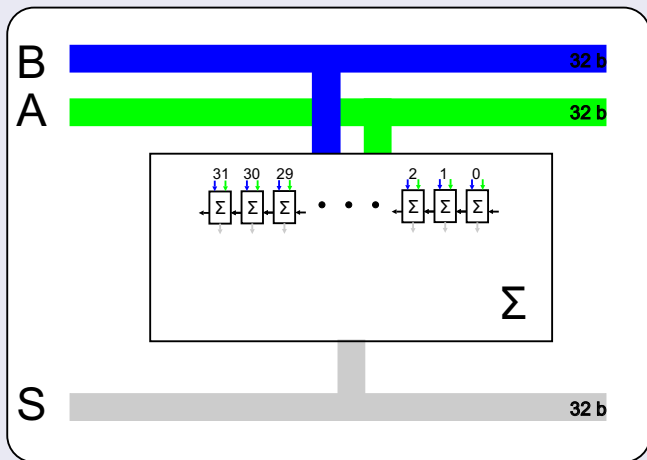


Bitová sčítačka

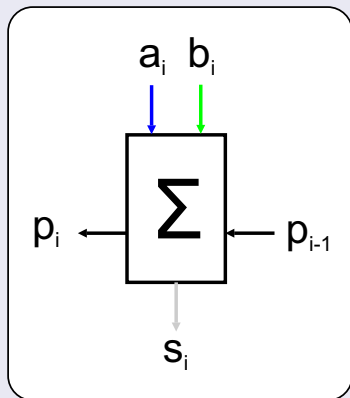
Navrhněte obvod, který bude realizovat bitovou sčítačku.

	1 1 1	
1011011010		730
100011010		282
<hr/>		<hr/>
1111110100		1012

Bitová sčítáčka - rozbor

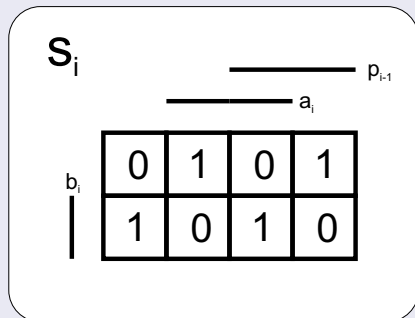


Bitová sčítačka - rozbor



Bitová sčítáčka - Karnaughova mapa - součet

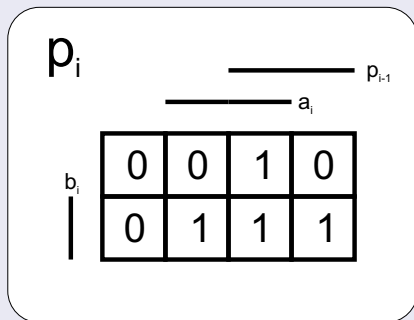
p_{i-1}	b_i	a_i	s_i
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1



$$s_i = a_i \bar{b}_i \bar{p}_{i-1} + \bar{a}_i b_i \bar{p}_{i-1} + \bar{a}_i \bar{b}_i p_{i-1} + a_i b_i p_{i-1}$$

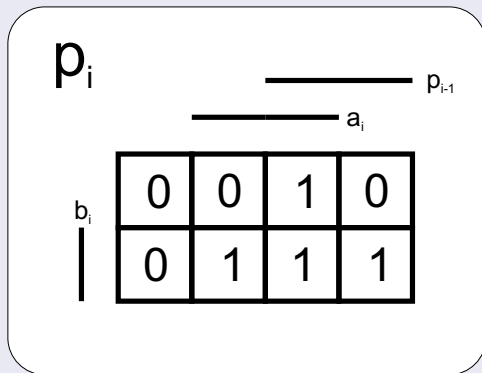
Bitová sčítačka - Karnaughova mapa - přenos do vyššího řádu

p_{i-1}	b_i	a_i	p_i
0	0	0	0
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	1

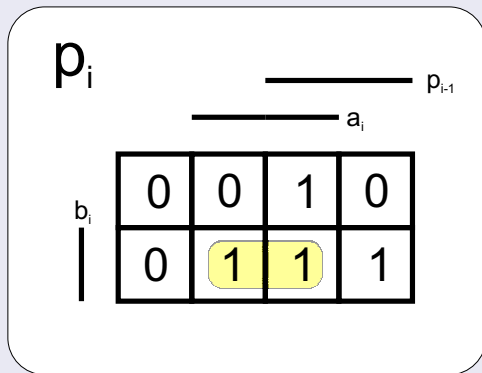


$$p_i = a_i b_i \overline{p_{i-1}} + a_i \overline{b_i} p_{i-1} + \overline{a_i} b_i p_{i-1} + a_i b_i p_{i-1}$$

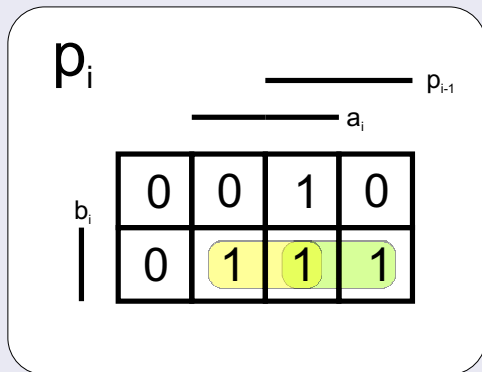
Bitová sčítáčka - Karnaughova mapa - přenos do vyššího řádu



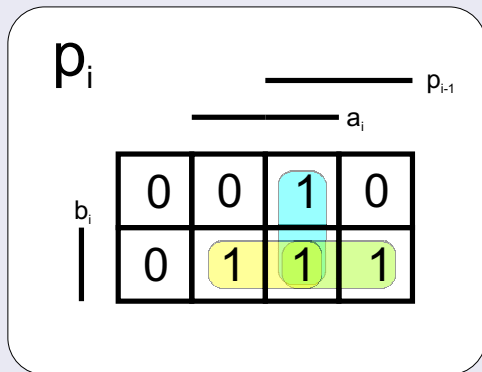
Bitová sčítáčka - Karnaughova mapa - přenos do vyššího řádu



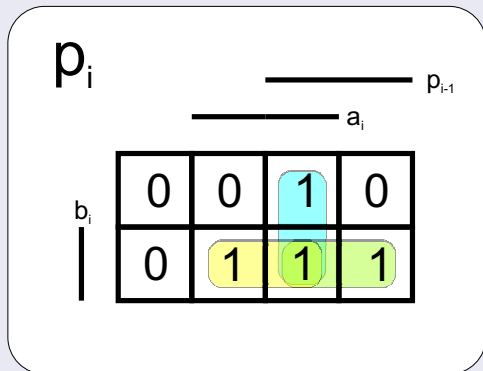
Bitová sčítáčka - Karnaughova mapa - přenos do vyššího řádu



Bitová sčítáčka - Karnaughova mapa - přenos do vyššího řádu

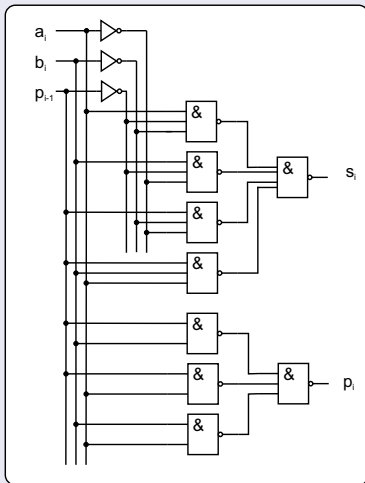


Bitová sčítačka - Karnaughova mapa - přenos do vyššího řádu



$$p_i = a_i b_i + a_i p_{i-1} + b_i p_{i-1}$$

Bitová sčítačka - realizace



- Norma IEC 61131-3
- jazyky:
 - jazyk kontaktních schémat (reléových schémat)
LD - Ladder Diagram (USA)
 - jazyk logických schémat (jazyk funkčních bloků)
FBD - Function Block Diagram (Evropa)
 - jazyk mnemokódů (assembler – j. symbolických adres)
IL - Instruction List
 - strukturovaný textový jazyk (obdoba PASCALu)
ST - Structured Text
 - jazyk sekvenčních diagramů
SFC – Sequential Function Chart

(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.