

BCD kód (8421 kód)

Jedná se o dvojkově desítkový kód (Binary Coded Decimal). Tento kód se používá pro kódování desítkových číslic 0 až 9. V tomto kódu je každá desítková číslice D vyjádřena kódovým slovem se čtyřmi bity ve dvojkové soustavě, jak je znázorněno v tab.

D	b_3	b_2	b_1	b_0
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1

Binárně dvojkový kód BCD 8421

Příklad: Vyjádřete dekadické číslo 5794_{10} v BCD kódu.

Řešení: $5 \rightarrow 0101$

$7 \rightarrow 0111$

$9 \rightarrow 1001$

$4 \rightarrow 0100$

Výsledek: $5794_{10} = 0101\,0111\,1001\,0100_{BCD}$.

Znaménko čísla se zobrazuje takovou kombinací bitů, která se v BCD kódu nevyskytuje.

Znaménko plus bývá například zobrazeno jako 1010 a znaménko mínus jako 1011. Umístění znaménka ve slově může být libovolné a určuje ho programátor.

Logické funkce

Logický obvod je takový obvod, u něhož může každá veličina na vstupu i výstupu v ustáleném stavu nabývat s určenou přesností jen jedné ze dvou možných hodnot a který obsahuje takové prvky, jejichž vstupní a výstupní veličiny mohou nabývat také jen jedné ze dvou možných hodnot. Logický obvod je realizován skupinou logických členů vzájemně spojených tak, aby realizovaly požadované logické funkce. Podle druhu realizované logické funkce dělíme logické obvody na:

kombinační logické obvody - jedná se o systémy, jejichž odezva je v určitém časovém okamžiku podmíněna výhradně hodnotami, které panují na vstupech tohoto systému. Kombinační obvody můžeme popsat logickými rovnicemi

$$y_1 = f_1(x_1, x_2, \dots, x_n)$$

$$y_2 = f_2(x_1, x_2, \dots, x_n)$$

$$\vdots$$

$$y_m = f_m(x_1, x_2, \dots, x_n)$$

sekvenční logické obvody - jedná se o systémy, jejichž odezva je v určitém časovém okamžiku dána nejen hodnotami proměnných na vstupech tohoto systému, ale i posloupností (sekvencí) předcházejících vstupních hodnot. Sekvenční obvod je proto opatřen pamětí, která svým stavem definuje vnitřní stav tohoto systému v závislosti na posloupnosti signálů, které přicházely na vstup.

Logické funkce jedné proměnné

Logickou funkci jedné nezávisle proměnné lze vyjádřit čtyřmi způsoby, které jsou znázorněny v následující tabulce 4.1. Vstupní proměnná je zde označena a , logické funkce jsou f_1 , f_2 , f_3 a f_4 .

a	f_0	f_1	f_2	f_3
0	0	0	1	1
1	0	1	0	1

Význam uvedených logických funkcí je následující:

f_0 - triviální identita, nulová funkce

f_1 - identická funkce, její hodnota se shoduje s hodnotou proměnné a

f_2 - funkce negace, hodnota funkce se shoduje s hodnotou $\bar{a}$

f_3 - triviální identita, jednotková funkce.

Praktický význam mají pouze funkce identická a funkce negace. Negace logické proměnné a se označuje pruhem nad příslušnou veličinou, tj. $f_2 = \bar{a}$.

Logické funkce dvou proměnných

Pro dvě vstupní proměnné můžeme najít šestnáct různých logických funkcí. Jestliže označíme vstupní proměnné symboly a , b a logické funkce f_0 až f_{15} , můžeme sestavit následující tabulku. Jednotlivé sloupce tabulky vyznačují hodnoty logických funkcí pro všechny kombinace vstupních proměnných a, b (00, 01, 10, 11). V pravém sloupci je uvedeno označení příslušné funkce.

Logické funkce dvou proměnných

a	0	0	1	1	Význam jednotlivých funkcí
b	0	1	0	1	
f_0	0	0	0	0	$f_0 = 0$, triviální identita
f_1	0	0	0	1	$f_1 = a.b$, logický součin, AND
f_2	0	0	1	0	$f_2 = a.\bar{b}$, přímá inhibice
f_3	0	0	1	1	$f_3 = a$, funkce identická
f_4	0	1	0	0	$f_4 = \bar{a}.b$, zpětná inhibice
f_5	0	1	0	1	$f_5 = b$, funkce identická
f_6	0	1	1	0	$f_6 = a \oplus b$, nonekvivalence, XOR, součet modulo 2
f_7	0	1	1	1	$f_7 = a + b$, logický součet, OR
f_8	1	0	0	0	$f_8 = \overline{a+b}$, negovaný logický součet, NOR, Piercova funkce
f_9	1	0	0	1	$f_9 = \overline{a \oplus b}$, ekvivalence, XNOR
f_{10}	1	0	1	0	$f_{10} = \bar{b}$, negace
f_{11}	1	0	1	1	$f_{11} = a + \bar{b}$, zpětná implikace
f_{12}	1	1	0	0	$f_{12} = \bar{a}$, negace
f_{13}	1	1	0	1	$f_{13} = \bar{a} + b$, přímá implikace
f_{14}	1	1	1	0	$f_{14} = \overline{a.b}$, negovaný logický součin, NAND, Shefferova funkce,
f_{15}	1	1	1	1	$f_{15} = 1$, triviální identita

Základní logické funkce

- *logický součin* (funkce f_1) dvou proměnných a , b , označovaný jako AND. Označíme-li výstup logického obvodu y , pak logický součin zapíšeme jako

$$y = a \cdot b ,$$

- *logický součet* (funkce f_7), označovaný jako OR. Při stejném značení ho můžeme zapsat vztahem

$$y = a + b ,$$

- *negace* (funkce f_{10} a f_{12}), označovaná jako NOT, je dána vztahy

$$y = \bar{a} \quad \text{nebo} \quad y = \bar{b} .$$

Uvedené vztahy se nazývají úplný *soubor logických funkcí*, protože jejich kombinací můžeme navrhnout jakýkoliv číslicový obvod. Tak například dvě funkce AND a NOT, nebo OR a NOT nám stačí pro vyjádření jakékoliv logické funkce.

Dalšími významnými funkcemi z tabulky jsou:

- NAND negovaný logický součin (funkce f_{14})

$$y = \overline{a \cdot b},$$

- NOR negovaný logický součet (funkce f_8)

$$y = \overline{a + b}.$$

Rovněž pomocí funkcí NAND a NOR můžeme vyjádřit a realizovat každou logickou funkci.

Obě funkce XOR a XNOR jsou významné pro realizaci aritmetických obvodů a číslicových komparátorů.

- *nonekvivalence* (funkce f_6)

$$y = a \oplus b = \bar{a} \cdot b + a \cdot \bar{b}.$$

Tato funkce nabývá hodnoty 1, jsou-li hodnoty obou proměnných různé.
Tato funkce se též nazývá Exclusive OR a značí se XOR.

- *ekvivalence* (funkce f_9), XNOR

$$y = \overline{a \oplus b} = a \cdot b + \bar{a} \cdot \bar{b}.$$

Logické funkce realizujeme pomocí základních logických členů, jejichž spojováním získáme logický obvod. Pro kreslení schémat logických obvodů budeme používat *schematické značky* těchto členů, které jsou uvedeny v tabulce

$Y=A+B$		Log. součet OR	
$Y=\overline{A+B}$		Negovaný log. součet NOR	
$Y=A \cdot B$		Log. součin AND	
$Y=\overline{A \cdot B}$		Negovaný log. součin NAND	
$Y=\overline{A}$		Negace NOT	
$Y=\overline{A} \cdot B + A \cdot \overline{B} = A \oplus B$		Nonekvivalence XOR	
DIN	US	ČSN	IEC

Logické funkce

Booleova algebra

Booleova (logická) algebra je matematická disciplína, která je přímo aplikovatelná při návrhu číslicových obvodů. Zahrnuje pravidla a teoremy pro operace s logickými proměnnými a funkcemi. Při používání pravidel se využívají tři základní operace

- logický součin (konjunkce),
- logický součet (disjunkce),
- negace (inverze),

které tvoří teoretický prostředek pro návrh (syntézu) logických obvodů s požadovaným chováním. Vztahy mezi dvouhodnotovými proměnnými lze definovat několika matematickými zákony, tzv. *zákony Booleovy algebry*, které jsou uvedené v tabulce

Pro zjednodušení práce s rovnicemi v Booleově algebře byly De Morganovy zákony uvedeny do obecné podoby, které říkáme *Shannonův teorém*. Tento teorém umožňuje změnit každou rovnici tak, že znaménko logického součtu nahradíme znaménkem logického součinu a naopak. Rovnici v této podobě říkáme inverzní. Podle Shannonova teorému získáme inverzní tvar rovnice následujícím způsobem:

- všechny logické součty zaměníme za logické součiny a naopak
- každou jednotlivou proměnnou negujeme
- negujeme celou rovnici.

	Zákony součtu	Zákony součinu
Zákon idempotence	$a + a = a$	$a.a = a$
Zákon absorbce	$a + a.b = a$	$a.(a + b) = a$
Zákon absorbce negace	$a + \bar{a}.b = a + b$	$a.(\bar{a} + b) = a.b$
Zákon komutativní	$a + b = b + a$	$a.b = b.a$
Zákon asociativní	$a + (b + c) = (a + b) + c$	$a.(b.c) = (a.b).c$
Zákon distributivní	$a + (b.c) = (a + b).(a + c)$	$a.(b + c) = a.b + a.c$
Neutrálnost nuly a jedničky	$a + 0 = a$	$a.1 = a$
Agresívnost nuly a jedničky	$a + 1 = 1$	$a.0 = 0$
Zákon vyloučeného třetího	$a + \bar{a} = 1$	$a.\bar{a} = 0$
Zákon negace	$\bar{\bar{0}} = 1$	$\bar{\bar{1}} = 0$
Zákon dvojité negace	$\bar{\bar{a}} = a$	
De Morganovy zákony	$\overline{a + b} = \bar{a}.\bar{b}$	$\overline{a.b} = \bar{a} + \bar{b}$

Příklad: Upravte výraz $a + b \cdot c + \overline{b} \cdot \overline{c} + d$ pomocí De Morganových zákonů.

Řešení:

$$\begin{aligned} a + b \cdot c + \overline{b} \cdot \overline{c} + d &= \overline{\overline{a + b \cdot c + \overline{b} \cdot \overline{c} + d}} = \overline{\overline{a + b \cdot c} \cdot \overline{\overline{b} \cdot \overline{c}} \cdot \overline{d}} = \\ &= \overline{\overline{a + b \cdot c} \cdot (b + c) \cdot \overline{d}} = \overline{\overline{a + b \cdot c}} \cdot \overline{(b + c)} \cdot \overline{\overline{d}} = \\ &= a \cdot b \cdot c \cdot (b + c \cdot d) = a \cdot (b + c) \cdot (b + c \cdot d) \end{aligned}$$

Příklad: Upravte výraz $a + \overline{a} \cdot b$ pomocí De Morganových zákonů.

Řešení:

$$a + \overline{a} \cdot b = \overline{\overline{a + \overline{a} \cdot b}} = \overline{\overline{a} \cdot \overline{a + b}} = \overline{\overline{a} \cdot \overline{a + b}} = \overline{\overline{a} \cdot \overline{a + b}} = \overline{\overline{a} \cdot \overline{a + b}} = a + b$$

De Morganovy zákony umožňují realizaci logické funkce pomocí obvodů NAND nebo NOR.

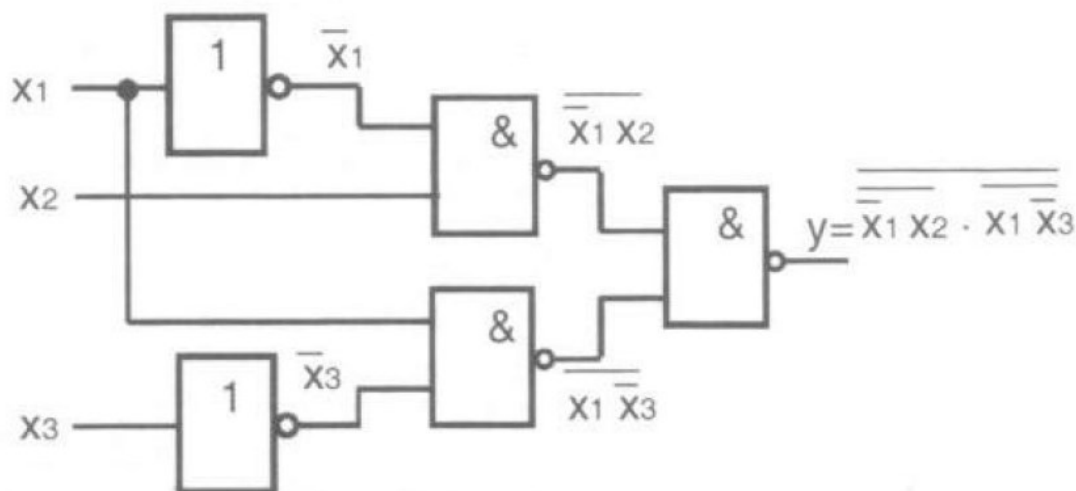
Příklad: Logickou funkci $y = \bar{x}_1 x_2 + x_1 \bar{x}_3$ realizujte pomocí obvodů typu NAND.

Řešení:

Logickou funkci upravíme pomocí De Morganova zákona. Aby nezměnila svou hodnotu, provedeme dvojitou negaci

$$y = \bar{x}_1 x_2 + x_1 \bar{x}_3 = \overline{\overline{\bar{x}_1 x_2 + x_1 \bar{x}_3}} = \overline{\overline{\bar{x}_1 x_2} \cdot \overline{x_1 \bar{x}_3}} .$$

Takto upravenou logickou funkci už můžeme realizovat pomocí základních členů NAND, protože se na pravé straně rovnice vyskytují pouze součiny a negace. Každý negovaný výraz odpovídá jednomu hradlu NAND. Na obr. je uvedeno schéma logického obvodu.



Pravdivostní tabulka určité hledané logické funkce f_1

N	c	b	a	Min.	Max.	f_1	f_2
0	0	0	0	$\bar{c}\bar{b}\bar{a}$	$c+b+a$	1	1
1	0	0	1	$\bar{c}\bar{b}a$	$c+b+\bar{a}$	1	0
2	0	1	0	$\bar{c}b\bar{a}$	$c+\bar{b}+a$	0	x
3	0	1	1	$\bar{c}ba$	$c+\bar{b}+\bar{a}$	0	1
4	1	0	0	$c\bar{b}\bar{a}$	$\bar{c}+b+a$	1	x
5	1	0	1	$c\bar{b}a$	$\bar{c}+b+\bar{a}$	0	0
6	1	1	0	$cb\bar{a}$	$\bar{c}+\bar{b}+a$	1	1
7	1	1	1	cba	$\bar{c}+\bar{b}+\bar{a}$	1	0

Z pravdivostní tabulky můžeme získat *logický výraz* pro jednotlivé logické funkce. Logický výraz funkce f_1 může být z pravdivostní tabulky získán dvěma způsoby:

- součtovou formou
- součinnovou formou,

podle toho, jestli použijeme k popisu logické funkce řádky, v nichž je funkce jedničková, nebo řádky, v nichž je nulová. Dříve, než popíšeme oba základní způsoby zápisu logické funkce, definujme několik základních pojmů:

- *Základní součinnový člen* je součin, který obsahuje všechny vstupní proměnné. Např. při vstupních proměnných a, b, c může mít základní součinnový člen tvar

$$a \cdot \bar{b} \cdot c.$$

Základní součinnový člen $a \cdot \bar{b} \cdot c = 1$, když $a = 1$, $b = 0$, $c = 1$.

Základní součinnový člen se označuje pojmem *minterm*.

- *Základní součtový člen* je součet, který obsahuje všechny vstupní proměnné. Při vstupních proměnných a, b, c může mít základní součtový člen např. tvar

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

Základní součtový člen $\bar{a} + b + \bar{c} = 0$, když $a = 1$, $b = 0$, $c = 1$.

Základní součet se někdy označuje pojmem *maxterm*.

- *Úplná součtová forma* logické funkce je dána součtem základních součinnových členů (mintermů), ve kterých logická funkce nabývá hodnoty 1.
- *Úplná součinnová forma* logické funkce je dána součinem základních součtových členů (maxtermů), ve kterých logická funkce nabývá hodnoty 0.

V tabulce je ve sloupci Min. uvedeno vyjádření všech mintermů funkce tří proměnných a ve sloupci Max. vyjádření všech maxtermů. Je na místě upozornit na skutečnost, že maxtermy jsou negací mintermů. Např.:

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

Příklad: Odvoďte úplnou součtovou formu logické funkce f_1 z pravdivostní tabulky

Řešení:

Úplná součtová forma vyjadřuje logickou funkci jako součet stavů, ve kterých nabývá hodnoty 1. Stavy, ve kterých je funkce $f_1 = 1$, jsou patrné z tabulky

vstupní proměnné			zákl. součiny (mintermy)
c	b	a	f_1
0	0	0	$\bar{c} \cdot \bar{b} \cdot \bar{a}$
0	0	1	$\bar{c} \cdot \bar{b} \cdot a$
1	0	0	$c \cdot \bar{b} \cdot \bar{a}$
1	1	0	$c \cdot b \cdot \bar{a}$
1	1	1	$c \cdot b \cdot a$

Úplná součtová forma funkce f_1 je dána jako součet mintermů

$$f_1(c, b, a) = \bar{c} \cdot \bar{b} \cdot \bar{a} + \bar{c} \cdot \bar{b} \cdot a + c \cdot \bar{b} \cdot \bar{a} + c \cdot b \cdot \bar{a} + c \cdot b \cdot a .$$

Pravdivostní tabulka

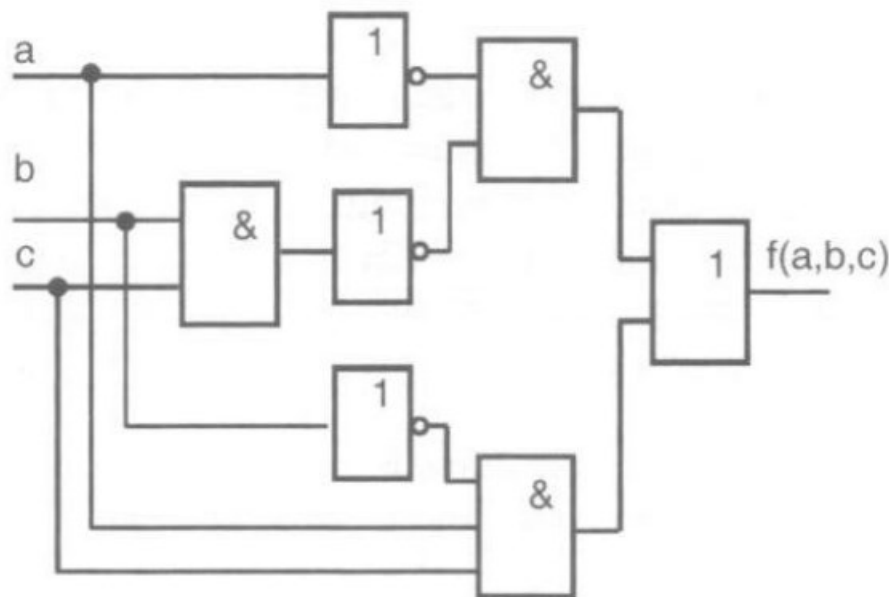
N	c	b	a	Min.	Max.	f ₁	f ₂
0	0	0	0	$\bar{c}\bar{b}\bar{a}$	$c+b+a$	1	1
1	0	0	1	$\bar{c}\bar{b}a$	$c+b+\bar{a}$	1	0
2	0	1	0	$\bar{c}b\bar{a}$	$c+\bar{b}+a$	0	x
3	0	1	1	$\bar{c}ba$	$c+\bar{b}+\bar{a}$	0	1
4	1	0	0	$c\bar{b}\bar{a}$	$\bar{c}+b+a$	1	x
5	1	0	1	$c\bar{b}a$	$\bar{c}+b+\bar{a}$	0	0
6	1	1	0	$cb\bar{a}$	$\bar{c}+\bar{b}+a$	1	1
7	1	1	1	cba	$\bar{c}+\bar{b}+\bar{a}$	1	0

Logický výraz

Příklad: Navrhněte logický obvod zadaný logickým výrazem
 $f(a,b,c) = \overline{a} \cdot \overline{b} \cdot c + a \cdot \overline{b} \cdot c$. Použijte základní členy AND, OR, NOT.

Řešení:

Jednotlivé logické operace ve výrazu jsme nahradili základními členy. Schéma navrženého logického obvodu je uvedeno na následujícím obrázku.



Realizační obvodová struktura funkce $f(a,b,c)$

Literatura:

M. Antošová, V. Davídek, Číslicová technika, 2013 Kopp

Richard C. Jaeger, Travis N. Blalock, Microelectronic circuit design,
2011 by The McGraw-Hill Companies, Inc.

P. Horowitz, W. Hill, The art of Electronics, Third edition, Cambridge University Press,
1980, 1989, 2015.

Donald A. Neamen, Microelectronics Circuit Analysis and Design, Published by
McGraw-Hill (2010).