Kombinační logické obvody

Booleovská proměnná a funkce

Definice 1: Booleovská proměnná je proměnná, která nabývá pouze hodnot 0 nebo 1.

Přímé proměnné: $a, b, c, \ldots$

Negované proměnné: $\bar{a}, \bar{b}, \bar{c}, \ldots$

Definice 2: Booleovská funkce n proměnných je funkce $f : \{0,1\}^n \rightarrow \{0,1\}$

Booleovská funkce n proměnných $y = f(x_1, x_2, \ldots, x_n)$,
kde $y, x_1, x_2, \ldots, x_n$ jsou booleovské proměnné.

Pro n booleovských proměnných existuje 2^{2^n} booleovských funkcí

Funkce dvou proměnných

a	b	0	NOR		$\overline{a}$		$\overline{b}$	XOR	NAND	AND	⇔	b	=>	a		OR	1
0	0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	0	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

Minterm a maxterm

a	b	c	Minterm	Maxterm
0	0	0	$m_0 = \bar{a}\bar{b}\bar{c}$	$M_0 = a + b + c$
0	0	1	$m_1 = \bar{a}\bar{b}c$	$M_1 = a + b + \bar{c}$
0	1	0	$m_2 = \bar{a}b\bar{c}$	$M_2 = a + \bar{b} + c$
0	1	1	$m_3 = \bar{a}bc$	$M_3 = a + \bar{b} + \bar{c}$
1	0	0	$m_4 = a\bar{b}\bar{c}$	$M_4 = \bar{a} + b + c$
1	0	1	$m_5 = a\bar{b}c$	$M_5 = \bar{a} + b + \bar{c}$
1	1	0	$m_6 = ab\bar{c}$	$M_6 = \bar{a} + \bar{b} + c$
1	1	1	$m_7 = abc$	$M_7 = \bar{a} + \bar{b} + \bar{c}$

Minterm je součinem všech proměnných (buď přímých nebo negovaných).
Pokud pro daný řádek proměnná nabývá log. 1, použije se přímá proměnná. V opačném případě negovaná proměnná.

Maxterm je součtem všech proměnných (buď přímých nebo negovaných). Pokud pro daný řádek proměnná nabývá log. 1, použije se negovaná proměnná. V opačném případě přímá proměnná.

Pozn.: žádná proměnná se v mintermu a maxtermu neopakuje dvakrát, a to ať v přímé nebo negované podobě.

Úplné normální formy

Úpná normální disjunktní forma (ÚNDF) je součtem (disjunkcí) mintermů.
Úpná normální konjunktivní forma (ÚNKF) je součinem (konjunkcí) maxtermů.

a	b	c	y		
0	0	0	0		M0
0	0	1	1	m1	
0	1	0	0		M2
0	1	1	0		M3
1	0	0	0		M4
1	0	1	1	m5	
1	1	0	1	m6	
1	1	1	0		M7

$$y = m1 + m5 + m6 = \bar{a}.\bar{b}.c + a.\bar{b}.c + a.b.\bar{c}$$

$$y = M0.M2.M3.M4.M7 = (a+b+c).(a+\bar{b}+c).(a+\bar{b}+\bar{c}).(\bar{a}+b+c).(\bar{a}+\bar{b}+\bar{c})$$

Pravdivostní tabulka -> booleovský výraz (formule)

ÚNDF (Úplná normální disjunktní forma)

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

Minterm

a	b	c	y
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

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

a	b	c	y
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

Když je více jedniček než nul, je výhodnější popsat výrazem negaci funkce. Pak se zaměřujeme na nuly.

Booleova algebra

Zákony	+(or)	.(and)
Neutrality nuly a jedničky	$x + 0 = x$	$x.1 = x$
Komutativní	$x + y = y + x$	$x.y = y.x$
Asociativní	$(x + y) + z = x + (y + z)$	$(x.y).z = x.(y.z)$
Distributivní	$(x + y).(x + z) = x + y.z$	$x.(y + z) = x.y + x.z$
Idempotence	$x + x = x$	$x.x = x$
Agresivity nuly a jedničky	$x + 1 = 1$	$x.0 = 0$
Absorbce	$x + xy = x$	$x.(x + y) = x$
Absorbce negace	$x + \bar{x}.y = x + y$	$x.(\bar{x} + y) = x.y$
Negace negace	$\bar{\bar{x}} = x$	
Vyloučeného třetího	$x + \bar{x} = 1$	$x.\bar{x} = 0$
De Morganovy	$\overline{x + y} = \bar{x}.\bar{y}$	$\overline{x.y} = \bar{x} + \bar{y}$

Algebraický důkaz zákonů absorbce negace

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

Důkaz:

$$x(\bar{x} + \mathrm{y}) = \underbrace{x\bar{x}}_{0} + \mathrm{xy} = \mathrm{xy}$$

$$x + \bar{x}\mathrm{y} = \mathrm{x} + \mathrm{y}$$

Důkaz:

$$x + \bar{x}\mathrm{y} = x\overbrace{(1 + y)}^{1} + \bar{x}y = x + xy + \bar{x}y =$$

$$= x + xy + \bar{x}y = x + y\underbrace{(x + \bar{x})}_{1} = x + y$$

Zákony booleovy algebry využijeme k minimalizaci booleovské funkce

$$y = a.b.c + a.\bar{b}.c + a.\bar{b}.\bar{c} = a.c(b + \bar{b}) + a.\bar{b}.\bar{c} =$$

$$= a.c.1 + a.\bar{b}.\bar{c} = a(c + \bar{b}.\bar{c}) = a.(c + \bar{b})$$

Minimalizace booleovské funkce je postup, kterým výraz zjednodušíme z pohledu počtu přímých a negovaných proměnných a počtu termů ve výrazu. Obecně řečeno snížíme počet operací, které je nutno provést.

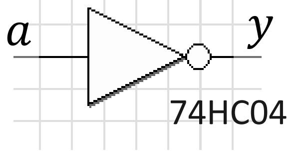
Metody:

- Algebraická minimalizace (úpravy výrazu využitím zákonů booleovy algebry)
- Karnaughovy mapy
- Algoritmus Quine-McCluskey

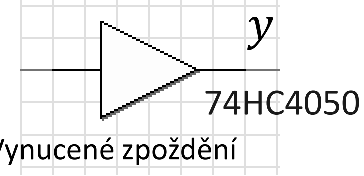
Logické obvody - logické (booleovské) funkce implementované v hardwaru

Přehled nejčastěji užívaných logických hradel

Invertor $y = \bar{a}$

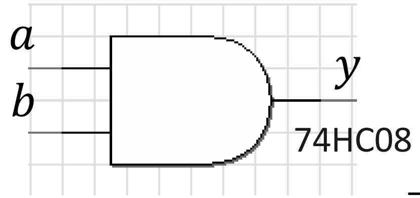


Buffer $y = a$

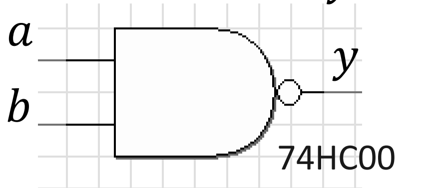


Vynucené zpoždění
nebo zesílení signálu

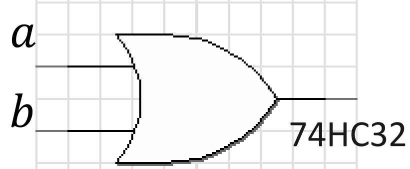
Hradlo AND $y = ab$



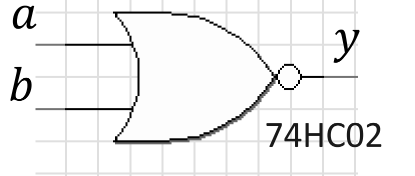
Hradlo NAND $y = \overline{ab}$



Hradlo OR $y = a + b$

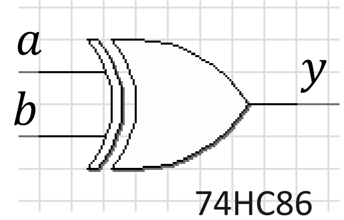


Hradlo NOR $y = \overline{a + b}$



Hradlo XOR

$y = a \oplus b$

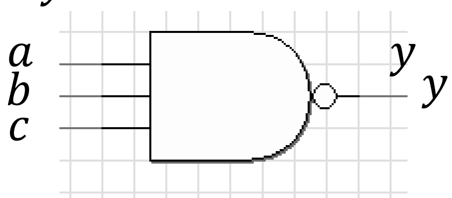


PRO ZVĚDAVCE:
U každého obvodu uvádíme číslo (např. 74HC00) pod kterým můžete výše uvedené hradlo zakoupit. Pro další informace použijte vyhledávač a dotaz např. „74HC00 datasheet pdf“ a získáte podrobné informace o daném obvodu.

Vícevstupá hradla, uvádíme pouze pro NAND, obdobně pro AND, OR, NOR, XOR

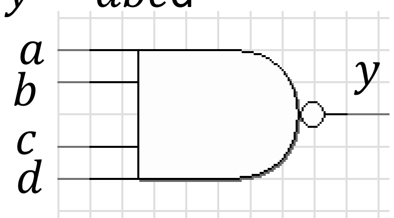
Třívstupový NAND

$y = \overline{abc}$



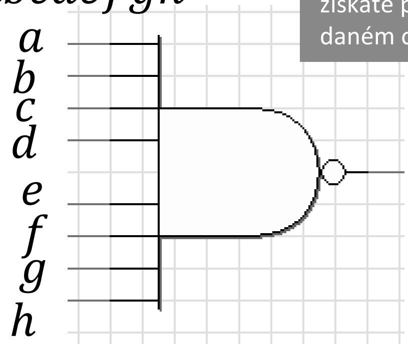
Čtyřvstupový NAND

$y = \overline{abcd}$



Osmivstupový NAND

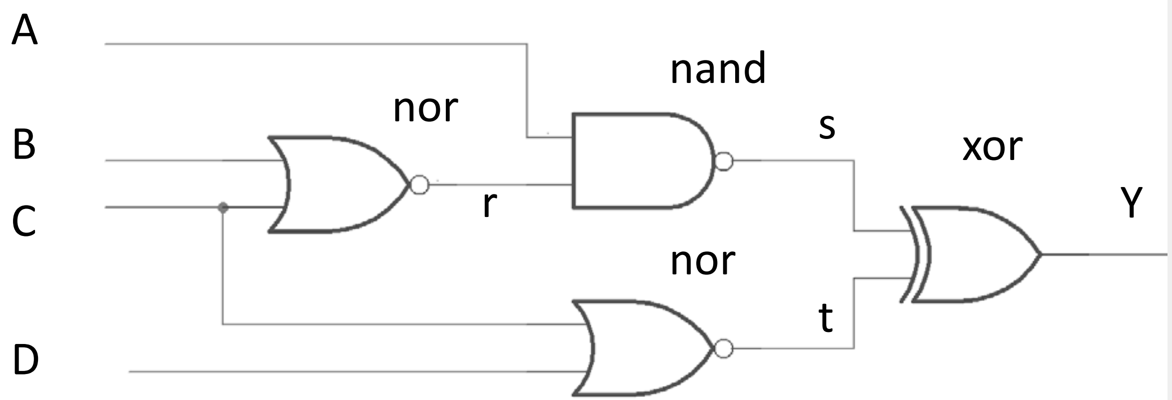
$y = \overline{abcdefgh}$



Kombinační logické obvody

Stavební prvky: logické obvody AND, OR, NOR (negovaný OR), AND, NAND (negovaný AND), XOR (exclusive or), invertor

Pospojováním se tvoří složitější logické funkce.



Sestrojte pravdivostní tabulku pro toto zapojení.

Návod: postupujte od vstupu k výstupu a postupně si vytvořte pravdivostní tabulky pro vnitřní signály a výstup, zde v pořadí r, t, s a nakonec Y.

Pro daný logický výraz nakreslete odpovídající schéma

$$y = a + \overline{b + d.(c + \overline{ab})}$$

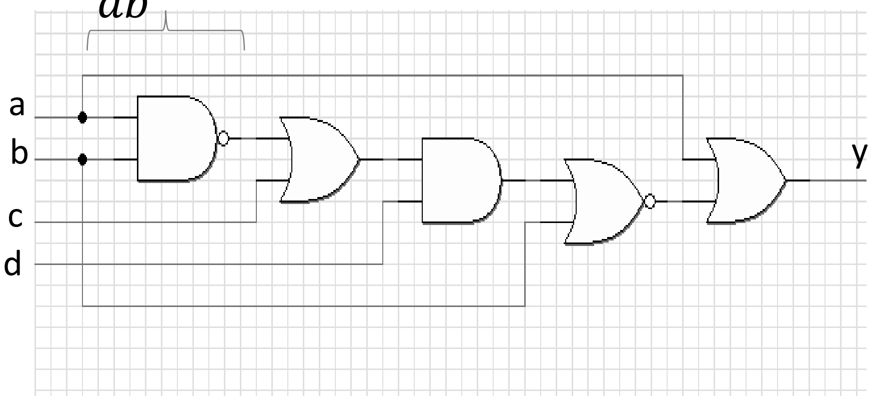
Řešení

$\overline{b + d.(c + \overline{ab})}$

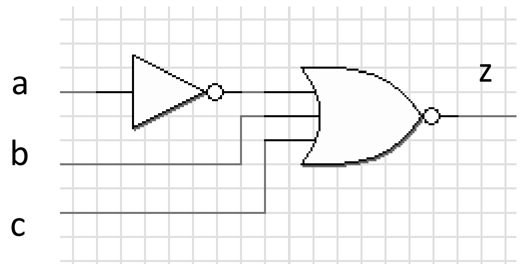
$d.(c + \overline{ab})$

$c + \overline{ab}$

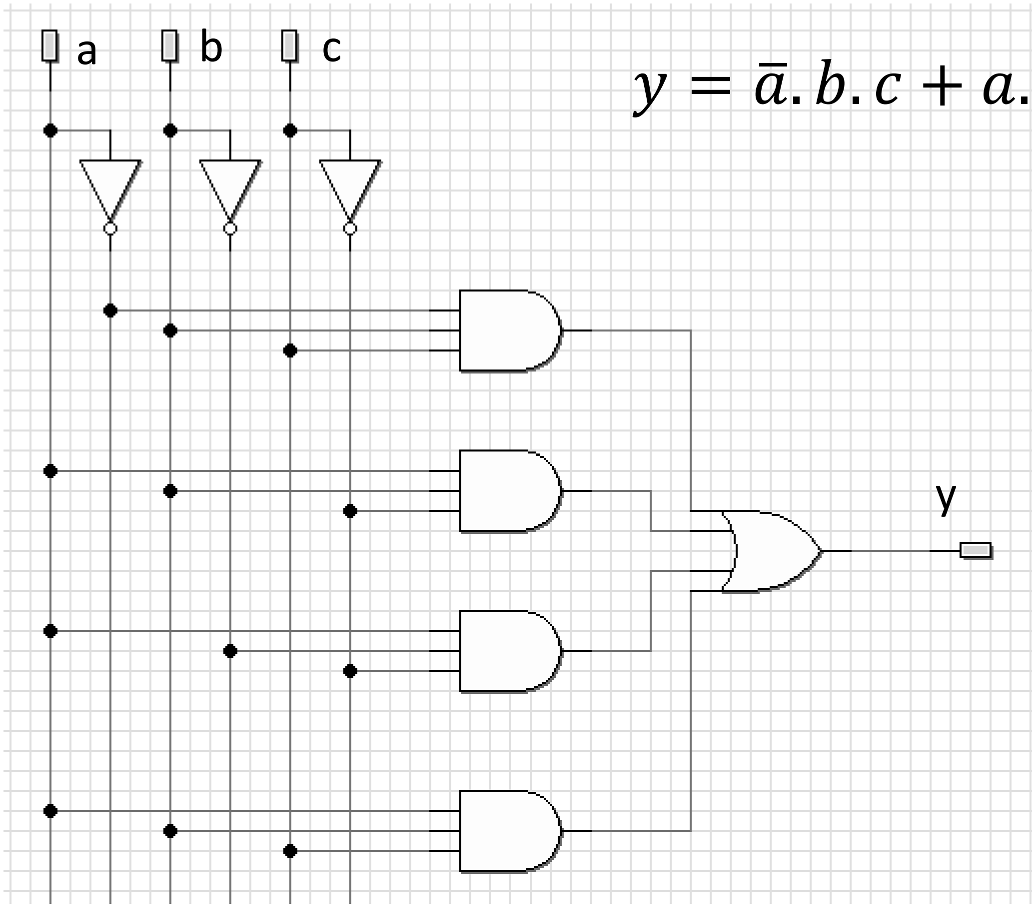
$\overline{ab}$



$$z = \overline{\bar{a}\text{+b+c}}$$



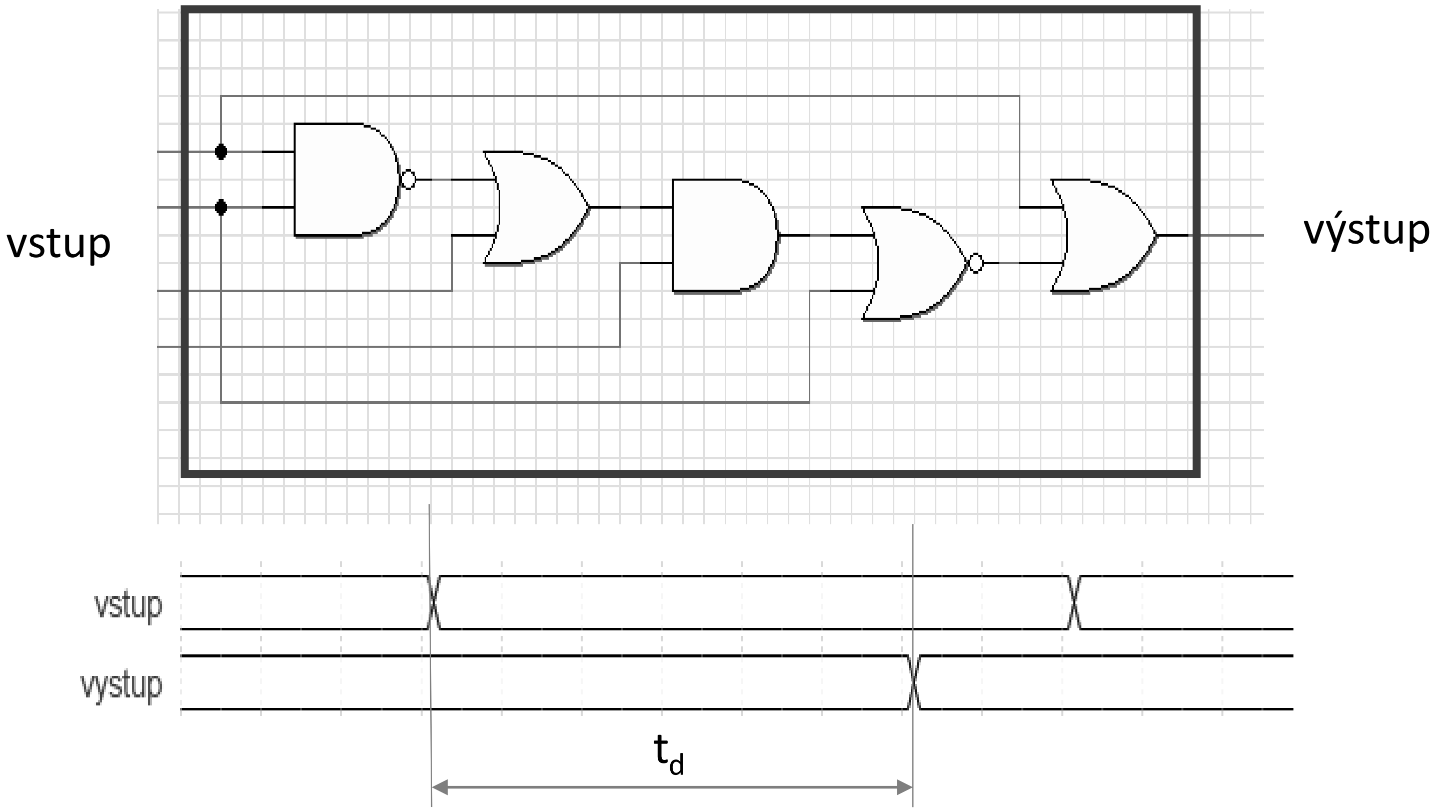
ÚNDF → schéma zapojení



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

a	b	c	y
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

Chování kombinačního obvodu v čase



T_d je zpoždění kombinačního obvodu. Je to doba mezi změnou vstupu a tomu odpovídající změně výstupu. Každé hradlo má určité zpoždění a zpoždění narůstá, jak se změna vstupu šíří obvodem. Celkové zpoždění t_d je tedy závislé na tom, které bity se mění na vstupu. V návrhu obvodů se musí vždy počítat s největším možným zpožděním.