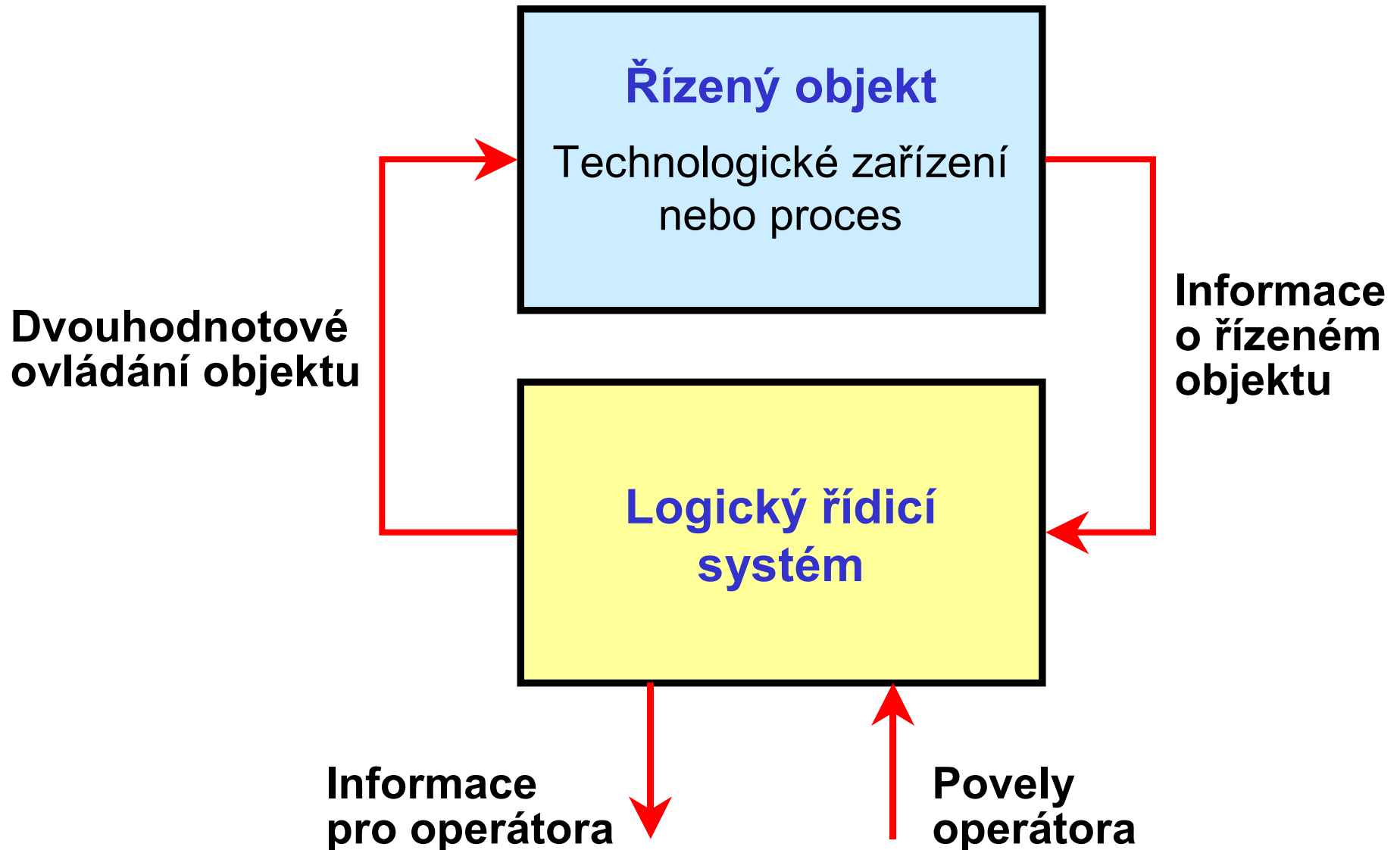


Základní schéma logického řízení



Logické řízení

- pracuje s nespojitými **dvouhodnotovými signály** (hodnota překročena/nepřekročena, zapnuto/vypnuto, otevřeno/zavřeno, chod/klid)
- Pozn.: vstupní signály mohou být spojité, protože mohou být komparovány s požadovanou mezní hodnotou až obvody logického řídicího systému



- mezní dvouhodnotová čidla
- ovládací spínače, tlačítka
- koncové spínače (dopravník)
- uzavírací ventily
- signalizační prvky
- stykače servopohonů

Teoretický základ logického řízení

Logická proměnná může nabývat pouze dvou hodnot (pravda/nepravda, logická jednička/logická nula, “1”/ “0”)

stav ventilu	log. proměnná <i>V</i>
otevřen	1
uzavřen	0

teplota	log. proměnná <i>T</i>
< 35 °C	0
≥ 35 °C	1

Logická funkce

- funkce, jejíž argumenty jsou logické proměnné
- může nabývat hodnot logické 1 nebo 0
- může být definována:
 - pravdivostní tabulkou
 - logickým výrazem
 - Karnaughovou nebo Svobodovou mapou
 - přechodovým nebo sekvenčním diagramem
 - atd.

Základní logické funkce

Logický výraz

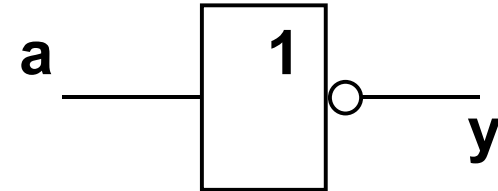
Negace: **NOT**

$$y = \bar{a}$$

Pravdivostní tabulka

a	y
0	1
1	0

Grafický symbol

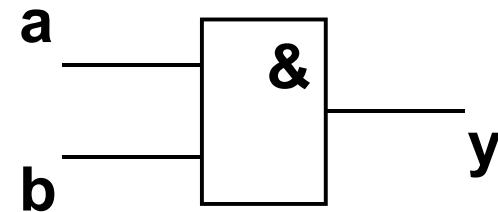


Log. součin:

AND

$$y = a.b$$

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

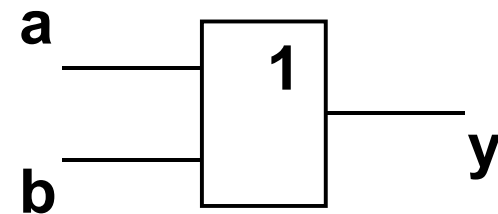


Log. součet:

OR

$$y = a + b$$

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



Univerzální logické funkce

Logický výraz

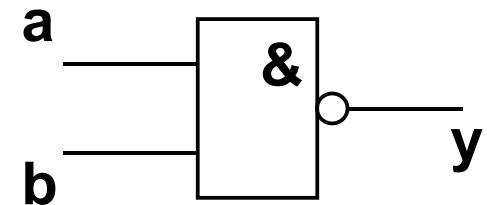
Pravdivostní tabulka

Grafický symbol

Negace logického
součinu: **NAND**

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

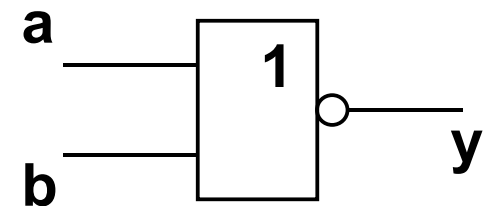
a	b	y
0	0	1
0	1	1
1	0	1
1	1	0



Negace logického
součtu: **NOR**

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

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



Základní zákony formální logiky

Booleova algebra

Zákon agresivity: $a.0 = 0$ $a + 1 = 1$

Zákon neutrality: $a.1 = a$ $a + 0 = a$

Zákon absorpce: $a.a = a$ $a + a = a$

Zákon vyloučení třetího: $a.\bar{a} = 0$ $a + \bar{a} = 1$

Zákon dvojí negace: $a = \bar{\bar{a}}$

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: $a.(b + c) = a.b + a.c$

$$(a + b).(a + c) = a + (b.c) \quad !!$$

De Morganovy zákony : $\overline{a.b} = \bar{a} + \bar{b}$ $\overline{a + b} = \bar{a}.\bar{b}$

- zákony a pravidla se používají při úpravách logických funkcí
- postupným zjednodušováním se **logická funkce minimalizuje**
- užitečná pravidla pro zjednodušování:

X je logický výraz

$$a.X + \bar{a}.X = (a + \bar{a}).X = X$$

$$(a + X).(\bar{a} + X) = a.\bar{a} + X = X$$

Příklad:

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

Vytvoření logického výrazu z pravdivostní tabulky:

Každou úplnou logickou funkci můžeme vyjádřit výrazem ve tvaru úplné normální disjunktivní nebo úplné normální konjunktivní formy.

$$\text{disjunktivní} \quad f(x_1, x_2, \dots, x_n) = \sum_{i=0}^{2^n-1} f(x_i) \cdot M_i(x_i)$$

$$\text{konjunktivní} \quad f(x_1, x_2, \dots, x_n) = \prod_{i=0}^{2^n-1} [f(x_i) + Ma(x_i)]$$

kde x_1, \dots, x_n - vstupní proměnné
 $f(x_i)$ - hodnota logické funkce f pro vstupní vektor x_i
 $M_i(x_i)$ - minterm nabývající hodnotu 1 pro vstupní vektor x_i
 $Ma(x_i)$ - maxterm nabývající hodnotu 0 pro vstupní vektor x_i

Definice některých používaných termínů :

Term je člen logického výrazu neboli boolský výraz. Rozeznáváme **term**:

- součinnový - **boolský výraz obsahující pouze operátor součinu (*)** - př.: $x_1 * x_2 * x_3$
- součtový - **boolský výraz obsahující pouze operátor součtu (+)** - př.: $x_1 + x_2 + x_3$
- **Minterm** je součinnový term, který obsahuje všechny proměnné v přímé nebo negované formě
- **Maxterm** je součtový term, který obsahuje všechny proměnné v přímé nebo negované formě
- Každý minterm nebo maxterm nabývá logické hodnoty 1 nebo 0 právě pro jeden vstupní vektor.

Vytvoření logického výrazu z pravdivostní tabulky:

Zápisem ve tvaru **úplné disjunktí normální formy - ÚDNF**

- z pravdivostní tabulky vybereme řádky s výstupem = 1
- každý vybraný řádek zapíšeme jako logický součin vstupních proměnných
- je-li proměnná = 1, zapíšeme symbol v přímém tvaru, je-li proměnná = 0, zapíšeme symbol negovaný
- výsledný tvar funkce je logický součet všech logických součinů

Zápisem ve tvaru **úplné konjunktí normální formy - ÚKNF**

- z pravdivostní tabulky vybereme řádky s výstupem = 0
- každý vybraný řádek zapíšeme jako logický součet vstupních proměnných
- je-li proměnná = 0, zapíšeme symbol v přímém tvaru, je-li proměnná = 1, zapíšeme symbol negovaný
- výsledný tvar funkce je logický součin všech logických součtů

Minimalizační metody

Standardní postup při minimalizaci je následující:

1. nalezení množiny primých implikantů
2. výběr minimálního počtu implikantů nutného k pokrytí logické funkce f

Principem je vyhledání dvou implikantů, které se liší pouze v jedné proměnné.

$$\begin{array}{l} a\bar{b}c \\ a\bar{b}\bar{c} \end{array} \rightrightarrows a\bar{b}$$

a nalezení přímého implikantu.

Minimalizace výrazů s využitím map

Jedno políčko v mapě představuje minterm.

Vzájemná poloha jedniček v mapě nese informaci o výskytu sousedních implikantů.

Snažíme se sestavit množinu implikantů, které představují přímý implikant.

Disjunktivní tvar těchto přímých implikantů tvoří minimalizovaný výraz logické funkce.

Hledáme minimalizační smyčku t.j. takovou oblast mapy, která obsahuje $2K$ políček, přičemž ke každému políčku přísluší v této smyčce K sousedních políček (sousední mintermy).

Snažíme se získat co nejmenší počet co největších minimalizačních smyček, které zahrnují všechny jedničky v mapě.

Smyčky se mohou překrývat t.j. mohou zahrnovat jedničky (mintermy) vícekrát do různých termů .

Mapa logické funkce

A 4x4 grid representing a binary map. The columns are labeled 0, 1, 2, 3 at the top. The rows are labeled 4, 5, 6, 7 on the left, 8, 9, 10, 11 on the right, and 12, 13, 14, 15 at the bottom. Above the grid, there are two horizontal lines labeled x1 and x0. To the left of the grid, there are two vertical lines labeled x3 and x2.

	0	1	2	3
4				
5				
6				
7				
8				
9				
10				
11				
12				
13				
14				
15				

Svobodova (binární) mapa

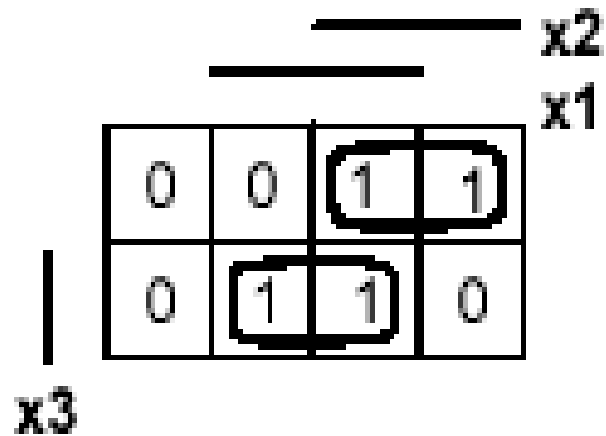
A 4x4 grid representing a Gray code map. The columns are labeled 0, 1, 3, 2 at the top. The rows are labeled 4, 5, 7, 6 on the left, 12, 13, 15, 14 on the right, and 8, 9, 11, 10 at the bottom. Above the grid, there are two horizontal lines labeled x1 and x0. To the left of the grid, there are two vertical lines labeled x3 and x2.

	0	1	3	2
4				
5				
7				
6				
12				
13				
15				
14				
8				
9				
11				
10				

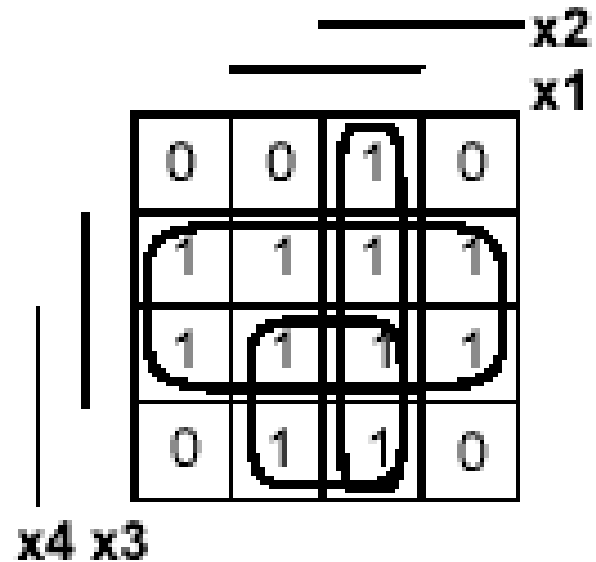
Karnaughova (gray-kód) mapa

Vznik : proměnné logické funkce rozdělíme na dvě skupiny, přičemž každé kombinaci jedné skupiny přiřadíme sloupec a každé kombinaci druhé skupiny přiřadíme řádek. Průnik řádků a sloupců (políčko) odpovídá kombinaci všech proměnných (minterm - vektor) a odpovídá jednomu řádku v pravdivostní tabulce. Do čtverečku vepisujeme hodnotu logické funkce pro dané vstupní písmeno (vektor). Záznamy jsou 1, 0, x .

Karnaughova mapa zajišťuje vzájemnou "sousednost " políček - mezi sousedními políčky mapy je změna pouze v jedné proměnné (totéž platí i o vnějších hranicích mapy - mapa je jako balón).



Disjunktivní (součtový) tvar
funkce $f = x_2 \cdot \neg x_3 + x_1 \cdot x_3$



Disjunktivní tvar funkce
 $f = x_3 + x_1 \cdot x_2 + x_1 \cdot x_4$

Logické řídicí systémy

- **Kombinační logické řízení**
 - hodnoty výstupů řídicího systému jsou dány pouze okamžitými hodnotami vstupních proměnných
- **Sekvenční logické řízení**
 - hodnoty výstupů řídicího systému závisí nejen na hodnotách vstupů, ale i na **stavu** řízeného systému, tj. na **předchozích hodnotách** vstupů a výstupů

Logické řízení – kombinační automat

Dělení logických systémů

Při dělení logických systémů nás zajímá model, u kterého se zabýváme pouze jeho vnějším dynamickým chováním. Podle chování dělíme logické systémy na:

Kombinační - jeho odezva v určitém časovém okamžiku je podmíněna výhradně okamžitými hodnotami na vstupech kombinačního systému.

chování lze popsat funkcí:

$$f: X \rightarrow Y$$

kde

X - množina vstupních vektorů

Y - množina výstupních vektorů

Tato funkce přiřazuje vektoru $x \in X$ určitý výstupní vektor $y = f(x)$, kde $y \in Y$. Tomuto přiřazení se také říká kombinační zobrazení, které přiřadí každému vektoru x pouze jeden vektor y a má vlastnosti funkce.

Zobrazení f můžeme rozložit na m dílčích zobrazení (funkcí), které definují hodnoty jednotlivých výstupních proměnných:

$$y_1 = f_1(x)$$

$$y_2 = f_2(x)$$

:

$$y_m = f_m(x)$$

kde $x \in X$



Obr.2.1. Kombinační systém

Oborem hodnot funkcí f_1, \dots, f_m je množina $\{0,1\}$. Funkce se označují pojmem logická funkce.

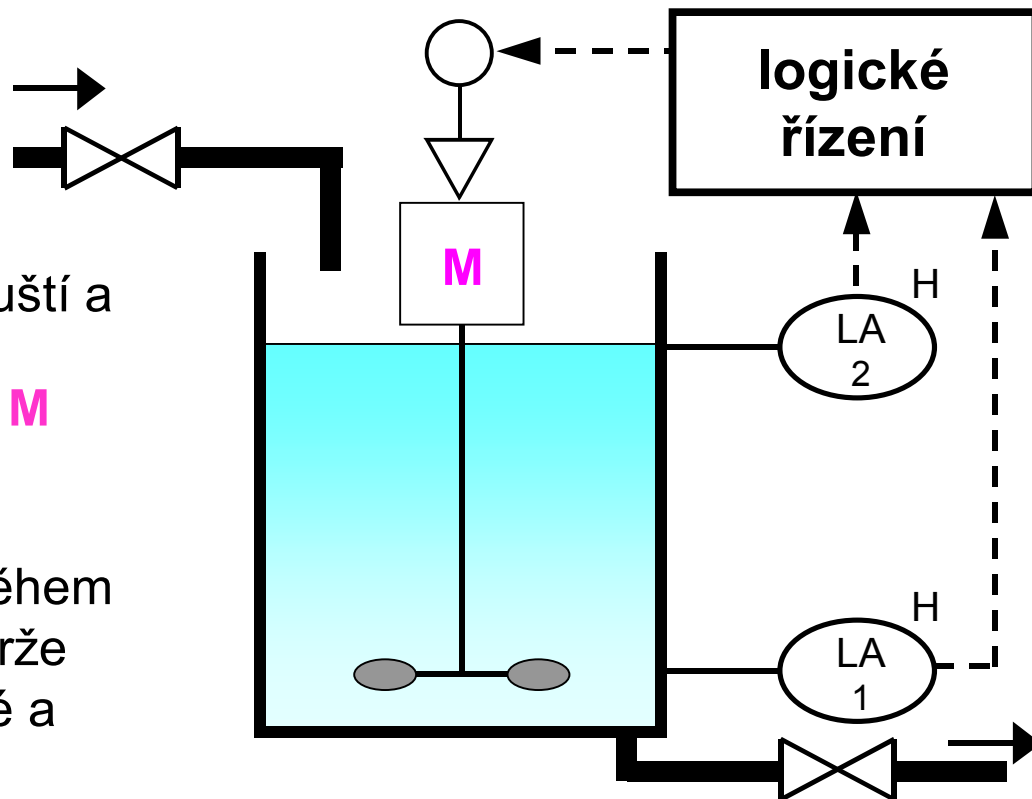
Návrh logického řídicího systému

- **slovní zadání**
 - při návrhu logického řídicího systému je důležitá spolupráce technologa a odborníka - projektanta řídicího systému
 - proces návrhu bývá několikastupňový (postupné zpřesňování slovního zadání)
- **sestavení pravdivostní tabulky**
- **vytvoření logického výrazu**
- **návrh technického řešení řídicího systému**

Příklad kombinačního logického řízení

Zadání:

- nádrž, která se cyklicky napouští a vypouští
- nádrž je opatřena míchadlem **M**
- 2 dvoupolohová čidla hladiny **LA1** a **LA2**
- míchadlo musí být zapnuto během napouštění či vypouštění nádrže
- míchadlo nesmí běžet při plné a prázdné nádrži



Logické proměnné:

- hladina je pod čidlem 1: **LA1 = 0**
hladina je nad čidlem 1: **LA1 = 1**
hladina je pod čidlem 2: **LA2 = 0**
hladina je nad čidlem 2: **LA2 = 1**
míchadlo je zapnuto: **M = 1**
míchadlo je vypnuto: **M = 0**

Pravdivostní tabulka:

vstupy		výstup
LA1	LA2	M
0	0	0
0	1	-
1	0	1
1	1	0

Dokončení příkladu:

Vytvoření logického výrazu z pravdivostní tabulky:

Pravdivostní tabulka:

vstupy		výstup
LA1	LA2	M
0	0	0
0	1	-
1	0	1
1	1	0

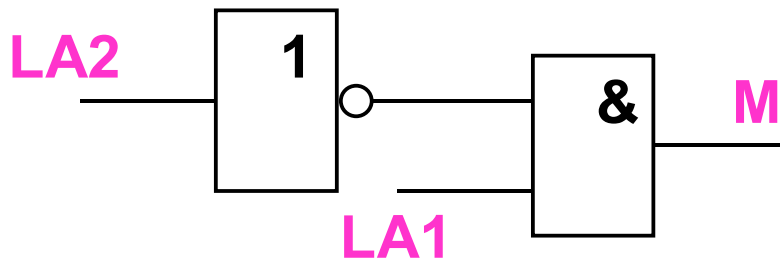
Podle ÚDNF:

$$M = LA1 \cdot \overline{LA2}$$

Podle ÚKNF:

$$M = (LA1 + LA2) \cdot (\overline{LA1} + \overline{LA2})$$

Realizace logického obvodu:



V následujícím textu jsou uvedeny příklady návrhu typických kombinačních obvodů.

Příklad 1. Zapínání světla ze dvou míst **Vstupy** - ovládače v místech **A a B**

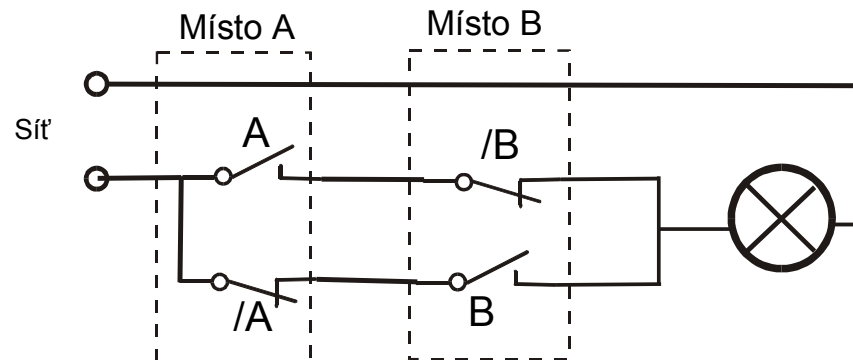
Výstup - žárovka **Y**

Řešení popisuje pravdivostní tabulka a z ní odvozený minimální výraz funkce

Vstupy		Výstup
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

1) Realizace kontaktními prvky

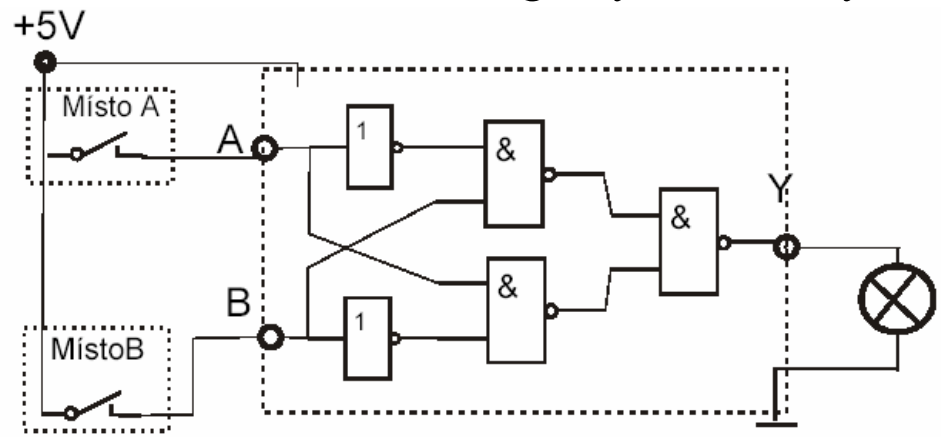
Dvupolové přepínače (schodišťové)



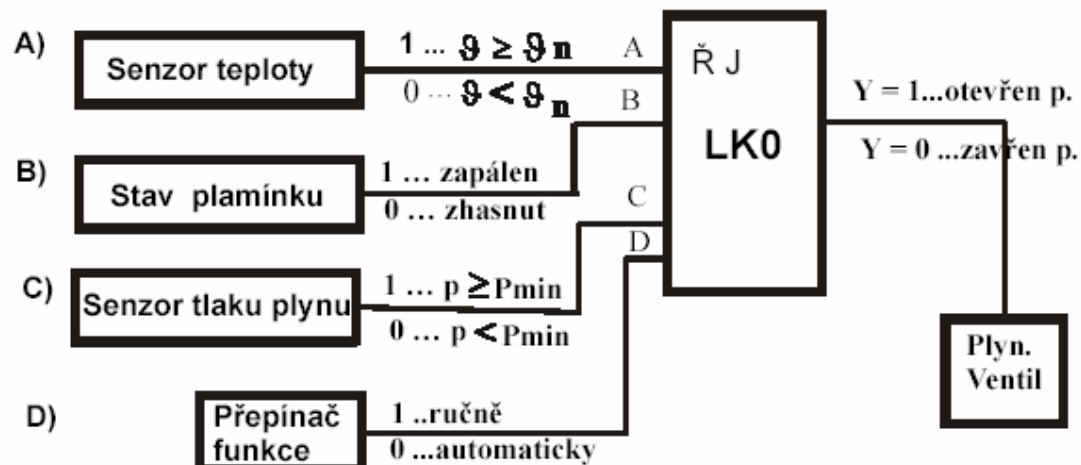
2) Realizace základními logickými obvody

Logický výraz
odvozený z tabulky

$$Y = A * /B + /A * B$$



Příklad 2. Navrhněte logický automat pro řízení hořáku plynového kotle



Vstupy: A, B, C, D

Výstup: Y

(tabulka 16 řádků,
 K-map 16 políček)

	D	C	B	A	Y
1	0	1	1	0	1
2	1	1	1	0	1
3	1	1	1	1	1
4					0
5					0
...					0
16					0

$$Y = \neg A * B * C * \neg D + \neg A * B * C * D + A * B * C * D$$

$$Y = \neg A * B * C + B * C * D$$

