1. Logické systémy – logické riadenie

Riadenie technologických procesov v sebe zahŕňa okrem regulácie aj automatické vykonávanie operácií v určitej postupnosti, rozhodovanie o ďalšom postupe podľa výsledku predošlého kroku a pod. Hovoríme o logickom riadení a tvrdí sa, že toto riadenie predstavuje v priemere až 80% funkcií realizovaných priemyselných riadiacich systémov. Napokon sa s ním stretávame aj v bežnom živote, príkladom je trebárs automatická pračka. Preto je dôležité mať aspoň základné znalosti o teoretickej podstate, možnostiach a technickej realizácii logických riadiacich systémov.

2. Teoretický základ logického riadenia

Rad základných pojmov a vzťahov matematickej (tzv. formálnej) logiky je analogických s pojmami a vzťahmi známymi z algebry. Preto sa tiež formálna logika často nazýva podľa jej zakladateľa Booleova algebra (George Boole, 1815-1864, írsky matematik, hlavné dielo Analysis of Logic). Základný rozdiel je v tom, že Booleova algebra nepracuje s číslami ale iba s veličinami (premennými), ktoré nadobúdajú iba dve hodnoty: áno/ne, pravda/nepravda, anglicky true/false, alebo tiež logická 1/logická 0. Najčastejšie sú používané hodnoty 1 a 0.

2.1. Logické premenné a logické funkcie

Logická premenná je analógiou číselnej premennej, ale zatiaľ čo číselná premenná môže nadobúdať ľubovoľné hodnoty z nejakej množiny (napr. reálnych čísiel), **logická premenná môže nadobúdať iba spomenuté dve hodnoty**. Z hľadiska riadenia to zodpovedá určitým medzným stavom reálnych zaradení, napr. „otvorené“/„zatvorené“, „zapnuté“/„vypnuté“, „chod“/„pokoj“, ale napríklad tiež „teplota<20°C“/„teplota≥20°C“. Tieto stavy môžeme ľahko popísať pomocou logických premenných, napr. takto:

čerpadlo	*log. premenná C*
Zapnuté	1
vypnuté	0

teplota	*log. premenná T*
<20°C	0
≥20°C	1

Logická funkcia je opäť obdobou číselnej matematickej funkcie s tým rozdielom, že jej argumentmi sú logické premenné a jej výsledkom je buď (logická) 1 alebo (logická) 0. **Logická funkcia je teda akékoľvek jednoznačné priradenie hodnôt závisle premennej (výstupnej veličiny) *y* ku stavom logických nezávisle premenných (vstupných veličín) *a*, *b*, *c*, ...** Ak je priradenie určené pre všetky možné kombinácie nezávisle premenných (2^n), hovoríme o *určitej logickej funkcii*, v inom prípade o *neurčitej logickej funkcii*.

Matematická funkcia je vždy nejako definovaná, obvykle matematickou formulou. Pre definíciu logickej funkcie sa používajú spôsoby:

- **slovne, výrokovo** - akýkoľvek zmysluplný text popisujúci súvislosť vstupných a výstupných dvojhodnotových veličín,
- **pravdivostnou tabuľkou**, ktorá udáva hodnotu logickej funkcie pre každú možnú kombináciu hodnôt jej argumentov (vstupov),
- **Karnaughovou mapou** (K-mapou),
- **logickým výrazom**, ktorý sa skladá z logických premenných spojených operátormi elementárnych logických funkcií.

Z popísaných spôsobov definovania si najprv všimnime pravdivostnú tabuľku. Tabuľka priraďuje kombináciám vstupných premenných hodnotu logickej funkcie, t.j. buď 1 alebo 0. Počet kombinácií prislúchajúcich k počtu *n* vstupných premenných je

$$\boxed{N = 2^n}$$

Napríklad pre tri vstupné premenné (a, b, c) je $2^3 = 8$ kombinácií hodnôt.

Tabuľka 1 – Zadanie logickej funkcie tabuľkou

Poradie, stavový index ***s***	Nezávisle premenné (vstupy)			Závisle premenné (výstupy)	
	a	***b***	***c***	***y***	(y_s)
0	0	0	0	1	(y_0)
1	0	0	1	1	(y_1)
2	0	1	0	0	(y_2)
3	0	1	1	1	(y_3)
4	1	0	0	0	(y_4)
5	1	0	1	–	(y_5)
6	1	1	0	1	(y_6)
7	1	1	1	0	(y_7)

V tabuľke prvý stĺpec ***s*** – *stavový index* udáva poradie kombinácie, aby bol dokumentovaný ich počet, je to vlastne dekadický zápis dvojkového čísla, ktoré tvoria kombinácie vstupných premenných. Hodnoty logickej funkcie dopĺňa návrhár logického obvodu alebo automatu pričom sa riadi jeho požadovanou funkciou. Je dobré zdôrazniť, že funkčné hodnoty nemusia byť definované pre všetky kombinácie vstupných premenných. Z definície môžu byť vyňaté kombinácie, ktoré nemôžu nastať (zaznamenané pomlčkou).

Pre funkciu v tabuľke 1 môžeme použiť aj skrátený zápis: $\boldsymbol{y_s}$**=1 pre *s*=0, 1, 3, 6, (5)**.

Rovnako ako v matematike sú definované základné výpočtové operácie, tak aj tu sú definované **základné operácie** s logickými premennými ako tzv. **elementárne logické funkcie** alebo tiež nazývané **booleove funkcie**. Tie sú tri:

- **negácia** — funkcia jednej premennej
  slovné vyjadrenie — Opak
  anglický názov: — NOT
  označenie: — $\bar{a}$
  pravdivostná tabuľka:

$\boldsymbol{a}$	$\boldsymbol{y \equiv f(a)}$
0	1
1	0

  zápis logickým výrazom: — $y = \bar{a} \quad (y = \text{NOT } a)$

- **logický súčet** — funkcia dvoch premenných
  slovné vyjadrenie — alebo, disjunkcia
  anglický názov: — OR
  označenie: — $a + b$ alebo $a \cup b$
  pravdivostná tabuľka:

$\boldsymbol{a}$	$\boldsymbol{b}$	$\boldsymbol{y \equiv f(a, b)}$
0	0	0
0	1	1
1	0	1
1	1	1

  zápis logickým výrazom: — $y = a + b \quad (y = a \text{ OR } b)$

- **logický súčin** funkcia dvoch premenných
  slovné vyjadrenie a, konjunkcia
  anglický názov: AND
  označenie: $a \cdot b$ alebo $a \cap b$
  pravdivostná tabuľka:

a	b	$y \equiv f(a,b)$
0	0	0
0	1	0
1	0	0
1	1	1

  zápis logickým výrazom: $y = a \cdot b \quad (y = a \text{ AND } b)$

Tieto funkcie predstavujú vlastne **logické operácie** obdobné operáciám algebrickým a preto analogicky pri vyhodnocovaní logických výrazov platia priority operácií postupne od najvyššej:

1. **negácia,** 2. **logický súčin,** 3. **logický súčet.**

Ak chceme priority operácií zmeniť, použijeme zátvorky.

Ľahko zistíme že úplný súbor logických funkcií dvoch premenných je 16. Tabuľka 2

a	b	y_1	y_2	y_3	y_4	y_5	y_6	y_7	y_8	y_9	y_{10}	y_{11}	y_{12}	y_{13}	y_{14}	y_{15}	y_{16}
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

Prakticky sa ale z tohto súboru využívajú len niektoré. Medzi elementárne (y_2, y_8, y_{13} resp. y_{11}) sa často zaraďujú logické funkcie, ktoré sú však definované na základe uvedených troch, takže vlastne už elementárne nie sú. Ide o nasledovné funkcie uvedené v zápise logickým výrazom:

- Shefferova funkcia (NAND, negácia logického súčinu) $f(a,b) = y_{15} = \overline{a \cdot b}$
- Pierceova funkcia (NOR, negácia logického súčtu) $f(a,b) = y_9 = \overline{a + b}$
- exclusive OR (XOR, neekvivalencia, súčet mod 2) $f(a,b) = y_7 = a\bar{b} + \bar{a}b$
- implikácia $f(a,b) = y_{14} = \bar{a} + b$
- ekvivalencia $f(a,b) = y_{10} = a \cdot b + \bar{a} \cdot \bar{b}$

Dôležité sú hlavne Schefferova funkcia (NAND) a Pierceova funkcia (NOR). Tieto dve spolu s elementárnymi funkciami tvoria tzv. **úplný súbor funkcií**, čo znamená, že pomocou nich môžeme vyjadriť ľubovoľnú funkciu.

Pri vytváraní logických funkcií napr. práve pre účely riadenia je veľmi prehľadné používať pravdivostnú tabuľku, avšak pri realizácii takto definovaných funkcií počítačovým programom je treba mať zápis v tvare logického výrazu. Ten sa vytvorí pomerne jednoducho.

Akúkoľvek logickú funkciu môžeme vyjadriť:

1. ***ú**plnou **d**isjunktnou (súčtovou) **n**ormálnou **f**ormou* (ÚDNF),

$$\boxed{y = \sum_{s=0}^{2^n-1} y_s \cdot K_s} \tag{2.1}$$

2. ***ú**plnou **k**onjunktnou (súčinovou) **n**ormálnou **f**ormou* (ÚKNF)

$$\boxed{y = \prod_{s=0}^{2^n-1} (y_s + \overline{K_s})} \tag{2.2}$$

Úplná znamená, že funkcia *obsahuje všetky* jednotkové mintermy (K_s) a každý minterm obsahuje *všetkých n* symbolov nezávisle premenných. *Disjunktná* preto, že mintermy viaže spolu operácia logického súčtu a *normálna forma* preto, že sa chápe ako dohovorený normatívny tvar. Pod pojmom ***minterm*** (K_s tiež *Mi*) rozumieme základný logický súčin nezávisle premenných zodpovedajúcich stavovému indexu *s* pričom v priamom vyjadrení sú tie premenné, ktoré pre príslušný stavový index nadobúdajú hodnotu 1 a v negácii sú tie, ktoré majú hodnotu 0. Negovaný minterm ($\overline{K_s}$ tiež *Ma*) nazývame ***maxterm***. Je to základný logický súčet všetkých nezávisle premenných. V priamom vyjadrení sú tie premenné, ktoré pre príslušný stavový index nadobúdajú hodnotu 0 a v negácii sú tie, ktoré majú hodnotu 1.

2.2.Zákony formálnej logiky

Pre operácie s logickými premennými a výrazmi platí podobne ako pri aritmetických operáciách určité zákony. Hovoríme o zákonoch formálnej logiky, alebo tiež o zákonoch **Booleovej algebry**. Sú to tieto zákony (*a, b, c* sú logické premenné):

- zákon agresivity 0 v súčine a 1 v súčte (zákon dominancie):

$$a \cdot 0 = 0 \qquad a + 1 = 1 \tag{2.3}$$

- zákon neutrality 0 v súčte a 1 v súčine (axióm):

$$a + 0 = a \qquad a \cdot 1 = a \tag{2.4}$$

- zákon o vylúčení tretieho (axióm):

$$a + \bar{a} = 1 \qquad a \cdot \bar{a} = 0 \tag{2.5}$$

- zákon komutatívny v súčte a súčine (axióm):

$$a + b = b + a \qquad a \cdot b = b \cdot a \tag{2.6}$$

- zákon asociatívny v súčte a súčine (axióm):

$$a + (b + c) = (a + b) + c \qquad a \cdot (b \cdot c) = (a \cdot b) \cdot c \tag{2.7}$$

- zákon distributívny (axióm):

$$a \cdot (b + c) = a \cdot b + a \cdot c \qquad a + (b \cdot c) = (a + b) \cdot (a + c) \tag{2.8}$$

- zákon opakovania:

$$a + a = a \qquad a \cdot a = a \tag{2.9}$$

- zákon absorpcie:

$$a + a \cdot b = a \qquad a \cdot (a + b) = a \tag{2.10}$$

- zákon dvojitej negácie:

$$\bar{\bar{a}} = a \tag{2.11}$$

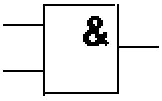
- de Morganove zákony:

$$\overline{a + b} = \bar{a} \cdot \bar{b} \qquad \overline{a \cdot b} = \bar{a} + \bar{b} \tag{2.12}$$

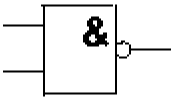
Tieto zákony používame pri úpravách a zjednodušovaní logických výrazov. Okrem nich existuje ešte jeden užitočný vzťah, ktorý sa pri zjednodušovaní výrazov môže hodiť:

- $a \cdot V + \bar{a} \cdot V = V$ (*V*... ľubovoľný logický výraz) (2.13)

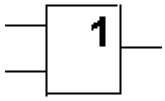
Pre informáciu ešte uvádzam štyri základné grafické symboly zodpovedajúce logickým funkciám dvoch premenných. Používajú sa pri kreslení funkčných blokových schém logických obvodov:



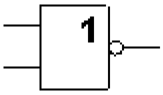
AND
logický súčin



NAND
negovaný logický súčin



OR
logický súčet



NOR
negovaný logický súčet

Teraz máme všetky potrebné informácie a tak sa môžeme vrátiť k vytvoreniu logického výrazu prakticky. Pre funkciu zapísanú pravdivostnou tabuľkou 1 dostaneme nasledujúce logické výrazy.

s	***a***	***b***	***c***	***y***	K_s	$\overline{K_s}$
0	0	0	0	1	$\bar{a}\cdot\bar{b}\cdot\bar{c}$	$a+b+c$
1	0	0	1	1	$\bar{a}\cdot\bar{b}\cdot c$	$a+b+\bar{c}$
2	0	1	0	0	$\bar{a}\cdot b\cdot\bar{c}$	$a+\bar{b}+c$
3	0	1	1	1	$\bar{a}\cdot b\cdot c$	$a+\bar{b}+\bar{c}$
4	1	0	0	0	$a\cdot\bar{b}\cdot\bar{c}$	$\bar{a}+b+c$
5	1	0	1	-	$a\cdot\bar{b}\cdot c$	$\bar{a}+b+\bar{c}$
6	1	1	0	1	$a\cdot b\cdot\bar{c}$	$\bar{a}+\bar{b}+c$
7	1	1	1	0	$a\cdot b\cdot c$	$\bar{a}+\bar{b}+\bar{c}$

Poznámka: pre skrátenie zápisu v ďalšom texte nie sú v mintermoch uvedené symboly násobenia.

podľa (2.1) ÚDNF

$$y = 1\cdot\bar{a}\bar{b}\bar{c} + 1\cdot\bar{a}\bar{b}\bar{c} + 0\cdot\bar{a}b\bar{c} + 1\cdot\bar{a}b\bar{c} + 0\cdot a\bar{b}\bar{c} + y_5\cdot a\bar{b}\bar{c} + 1\cdot ab\bar{c} + 0\cdot ab\bar{c}$$

podľa (2.2) ÚKNF

$$y = (1+a+b+c)\cdot(1+a+b+c)\cdot\left(0+a+\bar{b}+c\right)\cdot\left(1+a+\bar{b}+c\right)\cdot$$
$$\cdot(0+\bar{a}+b+c)\cdot(y_5+\bar{a}+b+c)\cdot\left(1+\bar{a}+\bar{b}+c\right)\cdot(0+\bar{a}+\bar{b}+\bar{c})$$

Úpravou takto získaných výrazov s využitím zákonov booleovej algebry (2.3, 2.4), resp. použitím mechanického postupu z pravdivostnej tabuľky, vytvoríme logický výraz:

Mechanický postup pre zápis výrazu <u>ÚDNF</u>:

1. z pravdivostnej tabuľky vyberieme riadky s hodnotou funkcie rovnajúcou sa 1,
2. každý takýto riadok zapíšeme ako logický súčin argumentov funkcie, pričom ak je hodnota argumentu = 1, zapíšeme jeho symbol v priamom tvare a ak je hodnota argumentu = 0, zapíšeme jeho symbol negovaný,
3. výsledný logický výraz (zápis funkcie) je logickým súčtom všetkých takto vytvorených logických súčinov.

Mechanický postup pre zápis výrazu <u>(ÚKNF)</u>:

1. z pravdivostnej tabuľky vyberieme riadky s hodnotou funkcie rovnajúcou sa 0,
2. každý takýto riadok zapíšeme ako logický súčet argumentov funkcie, pričom ak je hodnota argumentu = 0, zapíšeme jeho symbol v priamom tvare a ak je hodnota argumentu = 1, zapíšeme jeho symbol negovaný,
3. výsledný logický výraz (zápis funkcie) je logickým súčinom všetkých takto vytvorených logických súčtov.

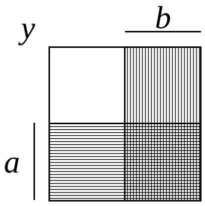
Definitívna podoba logického výrazu ma tvar:

ÚDNF: $y = \bar{a}\bar{b}\bar{c} + \bar{a}\bar{b}\bar{c} + \bar{a}b\bar{c} + y_5\cdot a\bar{b}\bar{c} + ab\bar{c}$

ÚKNF: $y = \left(a+\bar{b}+c\right)\cdot(\bar{a}+b+c)\cdot(y_5+\bar{a}+b+c)\cdot(\bar{a}+\bar{b}+\bar{c})$

Zápis logickej funkcie v mape

Mapa je buď obdĺžnik alebo štvorec, ktorý je pre *n* nezávisle premenných rozdelený do 2^n štvorcových políčok. Každému políčku je priradený jeden stavový index *s* a je v ňom vpísaná zodpovedajúca hodnota logickej závisle premennej (0 alebo 1), pre neurčité stavy pomlčka. Najčastejšie používanou je ***Karnaughova mapa***. Priradenie nezávisle premenných je urobené graficky, pomocou úsečiek, ktoré sú nad mapou a na jej boku a prekrývajú vždy polovicu políčok. Úsečka predstavuje hodnotu nezávisle premennej 1. V políčkach mapy pod a vedľa úsečiek nadobúda teda príslušná premenná hodnotu 1, inde 0 tak ako je názorne zobrazené v mape pre dve premenné. Úsečky sú rozmiestňované tak, aby boli pokryté všetky kombinácie nezávisle premenných.



0	1	3	2
4	5	7	6

KM pre 3 vstupné premenné so zápisom *s*

0000	0001	0011	0010
0100	0101	0111	0110
1100	1101	1111	1110
1000	1001	1011	1010

KM so zápisom kombinácií štyroch vstupných premenných

3. Klasifikácia logických systémov

- ak je výstup daný okamžitou kombináciou vstupov, potom sa systém nazýva **kombinačný** logický systém (KLS).
- ak je výstup daný nielen okamžitými kombináciami vstupov ale aj ich minulými hodnotami teda postupnosťou signálov, potom je systém **sekvenčný** logický systém (SLS). SLS je pamäťový logický systém
  - ak systém reaguje na zmeny vtedy, keď nastali, potom je to **asynchrónny** LS
  - ak systém reaguje iba vtedy, keď prichádzajú synchronizačné impulzy, potom je to **synchrónny** LS.

4. Činnosti pri použití logických systémov

- analýza LS - máme daný obvod (štruktúru) a zisťujeme akú funkciu (chovanie) plní,
- syntéza LS - máme danú funkciu (chovanie) a navrhujeme obvod (štruktúru),
- modelovanie a simulácia,
- diagnostika - hľadanie porúch,
- realizácia (výroba) LS,
  - pevná logika – hardvérová realizácia z hradiel,
  - pružná (programová) logika (PC, PLC),
- kontrola LS.