

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA STROJNÍHO INŽENÝRSTVÍ
ÚSTAV AUTOMATIZACE A INFORMATIKY

FACULTY OF MECHANICAL ENGINEERING
INSTITUTE OF AUTOMATION AND COMPUTER SCIENCE

UKÁŽKOVÉ ÚLOHY PRE VÝUKU AUTOMATIZÁCIE

SAMPLE TASKS FOR AUTOMATION EDUCATION

BAKALÁRSKA PRÁCA
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

SILVIA SILESKÁ

VEDÚCÍ PRÁCE
SUPERVISOR

ING. OLGA DAVIDOVÁ, PH.D.

BRNO 2010

Zadanie závěrečné práce

(na místo tohoto listu vložte originál nebo kopii zadání Vaše práce)

Licenčná zmluva

(na miesto tohoto listu vložte vyplnený a podpísaný list formuláru licenčného ujednanja)

Abstrakt

Táto bakalárska práca sa zaoberá návrhom a riešením ukázkových úloh kombinačných a sekvenčných logických obvodov. K vyriešeniu zadania je nutné sa oboznámiť s teóriou logického riadenia, s programom LOGO!Soft Comfort a problematikou programovateľných automatov. Celkovo šesť navrhnutých úloh je riešených pomocou základných pravidiel využívaných v logickom riadení.

Na záver bolo potrebné overiť správnu funkčnosť navrhnutých úloh, ktorá sa dala zrealizovať pomocou programu LOGO!Soft Comfort a programovateľného automatu. Navrhnuté ukázkové úlohy poslúžia ako pomôcka pre výuku predmetu Automatizácia.

Abstract

This bachelor's thesis deals with the design and demonstration of tasks addressing combinatorial and sequential logic circuits. To resolve the assignment must be familiar with the theory of logical management, with program LOGO!Soft Comfort and issue of programmable controllers. Overall, the proposed six tasks are solved by basic rules for use in a logical management.

Finally, it was necessary to verify the proper functioning of the proposed tasks, which it realized by using LOGO!Soft Comfort and programmable logic computers. The proposed tasks will serve as a demonstration tool for teaching the subject of automation.

Klíčová slova

Programovateľný automat, LOGO! Soft Comfort, Kombinačný logický obvod, Sekvenčný logický obvod

Keywords

Programmable logic computer, LOGO! Soft Comfort, Combinational logic circuit, Sequential logic circuit

Pod'akovanie

Touto cestou by som sa chcela poďakovať vedúcej mojej bakalárskej práce pani Ing. Olge Davidovej, Ph.D. za jej cenné rady a vecné pripomienky.

Obsah:

Zadanie záverečnej práce	3
Abstrakt	7
Pod'akovanie	9
1 Úvod	13
2 Logické riadenie	15
2.1 Logické funkcie	15
2.2 Booleova algebra	17
2.3 Vyjadrenie logickej funkcie.....	17
2.4 Minimalizácia logických funkcií	17
2.5 Logické obvody	18
2.5.1 Kombinačné obvody	18
2.5.2 Sekvenčné obvody	18
3 Programovateľné automaty	21
3.1 Zloženie programovateľného automatu.....	21
3.2 Výhody a nevýhody programovateľných automatov	22
3.3 Vstupy a výstupy PLC	22
4 Logo! Soft Comfort	23
4.1 Modul LOGO!	23
4.2 Princíp činnosti LOGO!.....	24
4.3 Vývojový software	25
5 Ukážkové úlohy	27
5.1 Plniaca linka	27
5.2 Signalizácia spotreby	29
5.3 Osvetlenie v obytnom dome	32

5.4	Nádrž s regulovanou hladinou	36
5.5	Výhybka.....	40
5.6	Železničný prejazd.....	43
6	Záver	45
	Zoznam použitej literatúry	47

1 Úvod

Úlohou bakalárskej práce je oboznámenie sa s programom LOGO!Soft Comfort a vytvorenie ukážkových úloh kombinačného a sekvenčného logického obvodu. Na zvládnutie splnenia daných požiadavkov, ktoré sa týkajú navrhnutia a celkového vyriešenia ukážkových úloh je nutné poznať základy logického riadenia, preto je začiatok práce venovaný danej problematike. Práca približuje tvary logických funkcií, ich algebraické zápisy, zápisy do tabuliek a napokon ich schématické značky, ktoré budú použité pri zapojovaní vzniknutých logických obvodov, možnosť zápisu daných logických funkcií a ich minimalizáciu. Nakoľko sa práca bude zaoberať kombinačnými a sekvenčnými logickými obvodmi, je nutné tieto obvody zahrnúť do teoretickej časti práce. Pre úplnosť sa práca zmieňuje všeobecne o programovateľných automatoch, ale aj o programovateľnom automate LOGO! od firmy Siemens a programe LOGO!Soft Comfort.

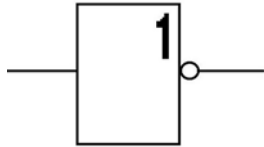
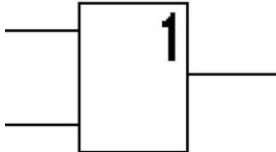
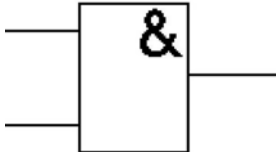
V praktickej časti bakalárskej práce budú navrhnuté už spomínané ukážkové úlohy kombinačného a sekvenčného logického obvodu. Tieto úlohy sa budú riešiť za pomoci pravidiel používaných v logickom riadení.

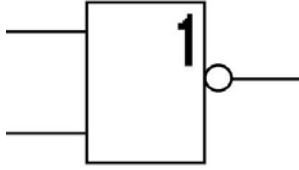
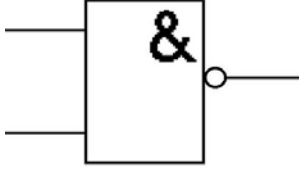
2 Logické riadenie

Najjednoduchší spôsob ako technicky uskutočniť riadenie je založený na tom, že v každej riadiacej činnosti alebo v hodnotení stavu systému rozlišujeme vždy iba dve možnosti, ktoré majú v poňatí riadiacej funkcie protikladný charakter. Tým je umožnené použiť pre popis funkcie jednoduchý aparát vzťahov medzi tzv. binárnymi veličinami, Booleovu algebru, Karnaughovu mapu a ďalšie prostriedky príbuzné výrokovej logike. Vďaka dvojhodnotovému princípu má každé takto popísané zariadenie len konečný počet rôznych stavov. Zásadne rozlišujeme dva typy logického riadenia, a to riadenie kombinačné, a riadenie sekvenčné. U kombinačného riadenia je pri kombinácii vstupov jednoznačne daný príslušný výstup. Pri sekvenčnom riadení je výstup daný nielen vstupmi ale aj časovým usporiadaním stavov ako sa na zariadení uskutočňujú.[1]

2.1 Logické funkcie

Existujú funkcie jednej premennej, dvoch premenných, štyroch až 16 premenných. Najčastejšie sú však využívané funkcie dvoch premenných.

Názov	Algebraický zápis	Tabuľka	Schematická značka															
Negácia	$y = \bar{x}$	<table border="1"> <thead> <tr> <th>x</th> <th>y</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> </tr> </tbody> </table>	x	y	0	1	1	0										
x	y																	
0	1																	
1	0																	
Logický súčet	$y = x_1 + x_2$	<table border="1"> <thead> <tr> <th>x₁</th> <th>x₂</th> <th>y</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> </tbody> </table>	x ₁	x ₂	y	0	0	0	0	1	1	1	0	1	1	1	1	
x ₁	x ₂	y																
0	0	0																
0	1	1																
1	0	1																
1	1	1																
Logický súčin	$y = x_1 \cdot x_2$	<table border="1"> <thead> <tr> <th>x₁</th> <th>x₂</th> <th>y</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> </tbody> </table>	x ₁	x ₂	y	0	0	0	0	1	0	1	0	0	1	1	1	
x ₁	x ₂	y																
0	0	0																
0	1	0																
1	0	0																
1	1	1																

Negovaný logický súčet	$y = \overline{x_1 + x_2}$	<table border="1"> <thead> <tr> <th>x_1</th> <th>x_2</th> <th>y</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> </tr> </tbody> </table>	x_1	x_2	y	0	0	1	0	1	0	1	0	0	1	1	0	
x_1	x_2	y																
0	0	1																
0	1	0																
1	0	0																
1	1	0																
Negovaný logický súčin	$y = \overline{x_1 \cdot x_2}$	<table border="1"> <thead> <tr> <th>x_1</th> <th>x_2</th> <th>y</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> </tr> </tbody> </table>	x_1	x_2	y	0	0	1	0	1	1	1	0	1	1	1	0	
x_1	x_2	y																
0	0	1																
0	1	1																
1	0	1																
1	1	0																

Obr. 1 Základné logické funkcie

Negácia ($y = \bar{x}$) je funkcia jednej premennej, u ktorej má závisle premenná vždy opačnú hodnotu než nezávisle premenná. V technickej praxi to znamená, že činnosť jedného prvku je daná nečinnosťou druhého prvku. Člen, ktorým sa realizuje logická negácia sa nazýva invertor alebo negátor. Má jeden vstup a jeden výstup. Na výstupe je vždy opačná hodnota logickej premennej privedenej na vstup.

Logický súčet ($y = x_1 + x_2$), nazývaný aj ako disjunkcia, nadobúda hodnotu logická jednička vtedy, keď aspoň jedna zo vstupných veličín má hodnotu 1. Táto funkcia je realizovaná logickým členom OR.

Logický súčin ($y = x_1 \cdot x_2$), nazývaný aj ako konjunkcia, nadobúda hodnotu logická jednička len v prípade, že všetky vstupné premenné majú hodnotu 1. Táto funkcia je realizovaná logickým členom AND.

Negovaný logický súčet ($y = \overline{x_1 + x_2}$), nazývaný aj ako negácia disjunkcie alebo Pierceova funkcia, táto funkcia nadobúda hodnoty $y = 1$ len vtedy, ak majú súčasne oba vstupy hodnotu logická nula. Funkcia je realizovaná logickým členom NOR.

Negovaný logický súčin ($y = \overline{x_1 \cdot x_2}$), nazývaný aj ako negácia konjunkcie alebo Shefferova funkcia, táto funkcia nadobúda hodnoty $y = 0$ vtedy, ak majú súčasne oba vstupy hodnotu logická jednička. Vo všetkých ostatných prípadoch je $y = 1$. Funkcia je realizovaná pomocou logického členu NAND.

2.2 Booleova algebra

Booleova algebra je dvojhodnotová logická algebra, ktorá pracuje s tromi základnými logickými funkciami. Tými funkciami sú: negácia, logický súčet a logický súčin. Základnou požiadavkou je každú logickú funkciu minimalizovať, to znamená vyjadriť ju čo najmenším počtom základných logických funkcií. Tým sa pri realizácii spotrebuje najmenší počet logických prvkov a technická realizácia vyjde najjednoduchšia.[2]

2.3 Vyjadrenie logickej funkcie

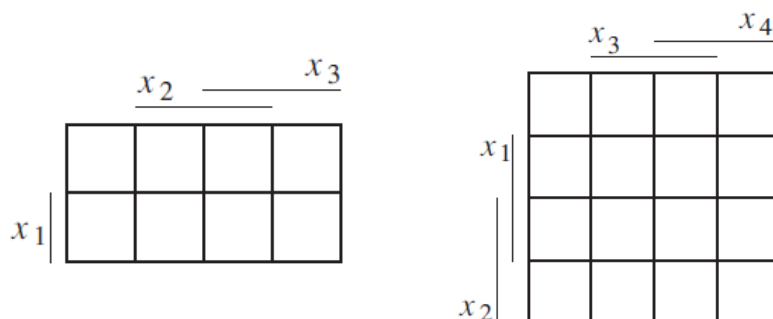
Logická funkcia sa dá vyjadriť niekoľkými spôsobmi. Môžeme hovoriť o slovnom vyjadrení, jedná sa o zmysluplný text, ktorý popisuje súvislosť medzi vstupnými a výstupnými dvojhodnotovými veličinami. Ďalšie vyjadrenia logickej funkcie sa dajú realizovať pomocou:

- pravdivostnej tabuľky
- algebraickým zápisom
- karnaughovou mapou
- blokovým schématom

2.4 Minimalizácia logických funkcií

Pre minimalizáciu logických funkcií existuje niekoľko metód. Jednou z metód je algebraická minimalizácia, kde sa funkcia zjednodušuje pomocou rôznych pravidiel Booleovej algebry až na minimálny výraz. Táto metóda však nie je zárukou, že danú logickú funkciu sme previedli na minimálny tvar. Ďalšou možnou metódou minimalizácie sú Karnaughove mapy.

Karnaughova mapa je tabuľka, ktorá je zostavená z takých políčok, koľko je kombinácií premenných vyšetrovanej funkcie. Funkciu s n dvojhodnotovými premennými vyjadríme mapou s 2^n políčkami. Každé políčko odpovedá jednej z možných kombinácií a zapisujeme doňho odpovedajúcu funkčnú hodnotu.[2]



Obr. 2 Ukážka Karnaughovej mapy pre tri a štyri premenné

2.5 Logické obvody

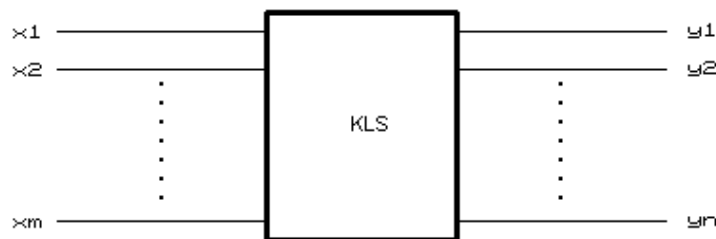
Logický obvod je obvod, ktorý vykonáva určitú logickú funkciu a uskutočňuje transformáciu vstupných signálov na výstupné. Logické obvody delíme na dve základné skupiny:

- Kombinačné obvody
- Sekvenčné obvody

Spoločná teória kombinačných a sekvenčných logických obvodov sa opiera o Booleovu algebru. Rozdiel medzi jednotlivými obvodmi je viditeľný vo výstupoch daných obvodov. Kým u kombinačných obvodov sú výstupy závislé len na kombinácii vstupných signálov, tak u sekvenčných obvodov sledujeme kombináciu vstupných signálov, ale aj ich poradie v čase.

2.5.1 Kombinačné obvody

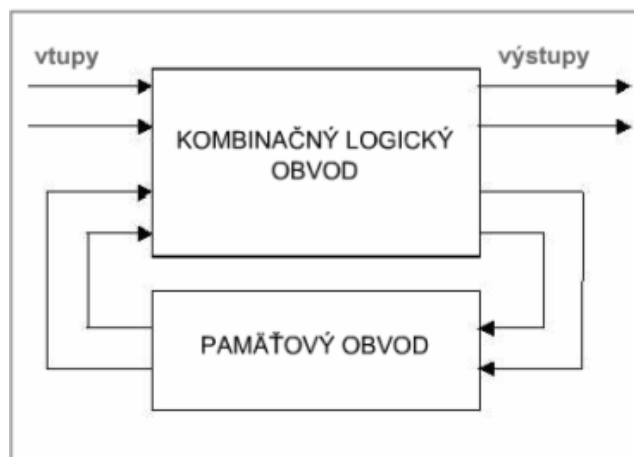
Pri kombinačných obvodoch sledujeme stav výstupov, ktorý závisí od momentálneho stavu vstupných signálov. V každom čase je možné priradiť akejkoľvek kombinácii vstupov vždy tú istú príslušnú výstupnú kombináciu. Navrhujú sa, v jednoduchých prípadoch cez Karnaughovu mapu, v zložitejších prípadoch sa využíva McCluskey/Quineho algoritmus.[]



Obr. 3 Blokové schéma kombinačného logického obvodu

2.5.2 Sekvenčné obvody

Sekvenčné logické obvody sú obvody, u ktorých výstupná funkcia je závislá nielen na okamžitých kombináciách pravdivostných hodnôt vstupných premenných, ale tiež na predchádzajúcich premenných t.j. na postupnosti kombinácii pravdivostných hodnôt vstupných premenných. Základným charakteristickým znakom rovnice sekvenčnej logickej funkcie je výskyt závisle premennej na oboch stranách logickej rovnice. Rovnica sekvenčnej logickej funkcie umožňuje navrhnuť alebo analyzovať logický obvod podľa zásad ako pri kombinačnej logickej rovnici. Pod pojmom sekvenčný obvod sa rozumie pamäťový obvod doplnený nezbytnými kombinačnými obvodmi pre záznam a mazanie pamäte. Logické premenné, ktoré sa podieľajú na vonkajšom zázname a mazaní pamäte, sa nazývajú vonkajšie premenné. Logické premenné, ktoré sa podieľajú na vnútornom nastavení pamäte, sa nazývajú vnútorné premenné. Sekvenčný logický obvod sa líši od kombinačného tým, že niektoré výstupné signály obvodu sú zavedené na niektoré vstupy obvodu ako vnútorné premenné. Znázorňuje to bloková štruktúra na obr. 4 []



Obr. 4 Blokové schéma sekvenčného logického obvodu

Sekvenčné logické obvody delíme na synchronné a asynchrónne, ďalej na:

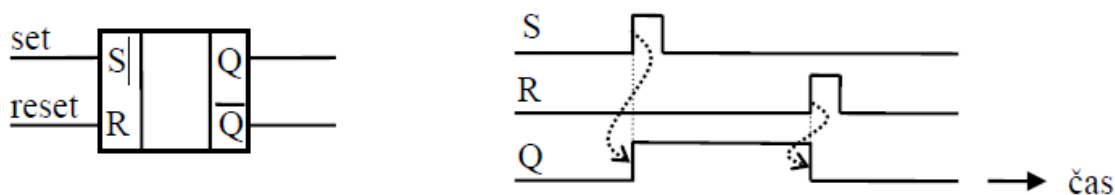
- Klopné obvody
- Čítače
- Posuvné registre

Klopné obvody sa ďalej delia na stabilné, astabilné a bistabilné obvody. Na vyriešenie ukážkovej úlohy železničného prejazdu je použitý RS klopný obvod, ktorý zaraďujeme medzi bistabilné klopné obvody. Podrobnejšie delenie sekvenčných logických obvodov sa spomína v literatúre [6].

RS Klopný obvod

Obvod tohto typu má dva vstupy, a to vstup nastavovací, ktorý sa značí S (Set) a resetovací vstup R (Reset). Signál privedený na vstup S klopného obvodu spôsobí preklopenie zo stavu $Q = 0$ do $Q = 1$. Ak už je na výstupe logická jednotka, tak v tom prípade sa nič nezmení. Účinok signál na vstupe R je opačný, to znamená, že ak je na vstup R privedená logická jednotka $Q = 1$, tak sa preklopí na stav $Q = 0$. Ak v tomto stave klopný obvod bol, tak nenastane žiadna zmena.

Vlastnosti klopných obvodov sa popisujú časovými diagramami, ktoré znázorňujú časový priebeh vstupných signálov a zmeny stavu klopného obvodu. Obvodovú značku a časový diagram funkcie klopného obvodu RS môžeme vidieť na obrázku 5. [6]



Obr. 5 Obvodová značka a časový diagram KO RS

S	R	Q
0	0	Predchádzajúci stav
0	1	0
1	0	1
-	-	Dané zapojením

Obr. 6 Pravdivostná tabuľka RS klopného obvodu

Nedefinovaný výstup je pre situáciu, keď vstupy R a S nadobúdajú hodnotu logickej jednotky. To odpovedá prípadu, keď je súčasne požadované nastavenie a resetovanie klopného obvodu.[6]

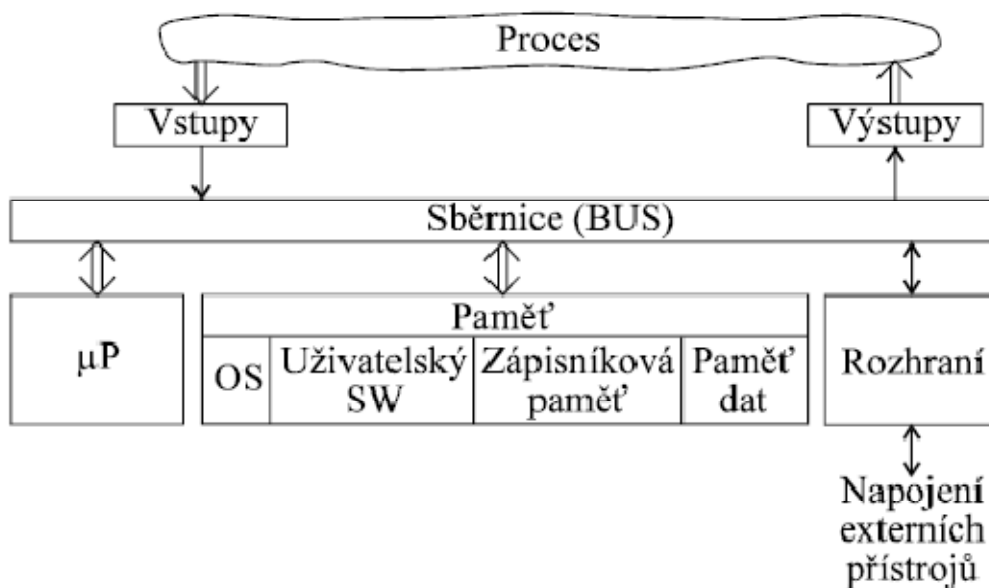
3 Programovateľné automaty

Programovateľný automat, označovaný ako PLC (Programmable Logic Controller) je voľne programovateľný riadiaci systém prispôsobený pre riadenie priemyslových a technologických procesov, mnohokrát špecializovaný na úlohy prevažne logického typu. Menšie typy bývajú riešené ako kompaktné celky, väčšie sa konštruujú ako modulárne.

Pôvodne boli určené na riadenie strojov ako náhrada za pevnú reléovú logiku, teraz sa využívajú nielen v tradičnej strojárskvej výrobnjej technológii vrátane manipulačnej a dopravnej techniky, ale aj v energetike (regulácia v elektrárnach, v kotoľniach, v klimatizačných jednotkách aj v chladiacich zariadeniach), v chemických výrobach, v poľnohospodárskej a spracovateľskej výrobe atď. [2]

3.1 Zloženie programovateľného automatu

Ako celok je programovateľný automat zložený z rady spolupracujúcich funkčných blokov prepojených spravidla prostredníctvom dvoch zberníc. Z funkčného hľadiska je to zložitý blok s mnohými vstupmi a výstupmi, medzi ktorými sa dá vytvoriť väčší počet kauzálnych logických vzťahov. Tieto vzťahy môžu mať povahu booleovských kombinačných alebo sekvenčných funkcií, ale môžu byť vyjadrené aj ďalšími reláciami ako sú časovače, čítače, spoždovače a iné.[1]



Obr. 7 Blokové schéma modulárneho programovateľného automatu

Modulárne prevedenie programovateľného automatu ma variabilný počet vstupných aj výstupných jednotiek a ďalších zariadení. Funkčné bloky sú prepojené prostredníctvom jednej alebo dvoch zberníc. Bežne osadzované modulové jednotky v programovateľných automatoch sú centrálna procesorová jednotka, systémová pamäť, užívateľská pamäť, interface umožňujúci spojenie s PC a množstvo modulov pre analógové, digitálne a binárne vstupy. V krajných prípadoch môže mať programovateľný automat dvojhodnotové vstupy a výstupy a byť vystavený ako čisto binárny systém alebo naopak môže byť koncipovaný ako analógový. [2]

3.2 Výhody a nevýhody programovateľných automatov

Výhody:

- rýchle preprogramovanie úlohy
- možnosť vystavania veľkej hierarchickej štruktúry podľa potreby
- flexibilita
- modularita
- hospodárnosť (lacné veľmi malé a malé kompaktné automaty)
- vstavaná diagnostika vlastného PLC
- možnosť tvorby vonkajšej diagnostiky
- jednoduché programovanie
- možnosť použitia vyšších programovacích jazykov u nových automatov
- jednoduchý a tým spoľahlivý OS reálneho času
- veľká ponuka kvalitných prístrojov od rôznych výrobcov

Nevýhody:

- nižší programátorský komfort než u minipočítačov
- menšia flexibilita v porovnaní s IPC
- využitie nedostatočne štandardizovaných sériových zberníc pre prepojenie automatov do komunikačných sietí [7]

3.3 Vstupy a výstupy PLC

Vstupná jednotka zaisťuje prevzatie informácií z riadeného procesu do PLC. Signály vstupujú do počítača prostredníctvom tzv. číslicových vstupov. Každý vstupný dvojhodnotový (binárny) signál obsahuje informáciu o veľkosti jedného bitu. Číslicové vstupy zachytávajú nielen daný stav hodnôt logických signálov, ale niektoré z nich sú tzv. číslicového typu, tj. Počítajú počet prichádzajúcich impulzov. Okrem číslicových vstupov je u PLC obmedzené množstvo tzv. analógových vstupov, ktoré merajú fyzikálne veličiny ako teplota, tlak a pod, tj. Výsledkom merania je číslo. Prevod signálu, najčastejšie napätového, na číslo sa uskutočňuje v tzv. analogovo-číslivom prevodníku ADC (A/Č).[7]

4 Logo! Soft Comfort

Kapitola nás oboznámi s programovateľným automatom LOGO! Soft Comfort od firmy Siemens a aj s jeho vývojovým prostredím.

4.1 Modul LOGO!

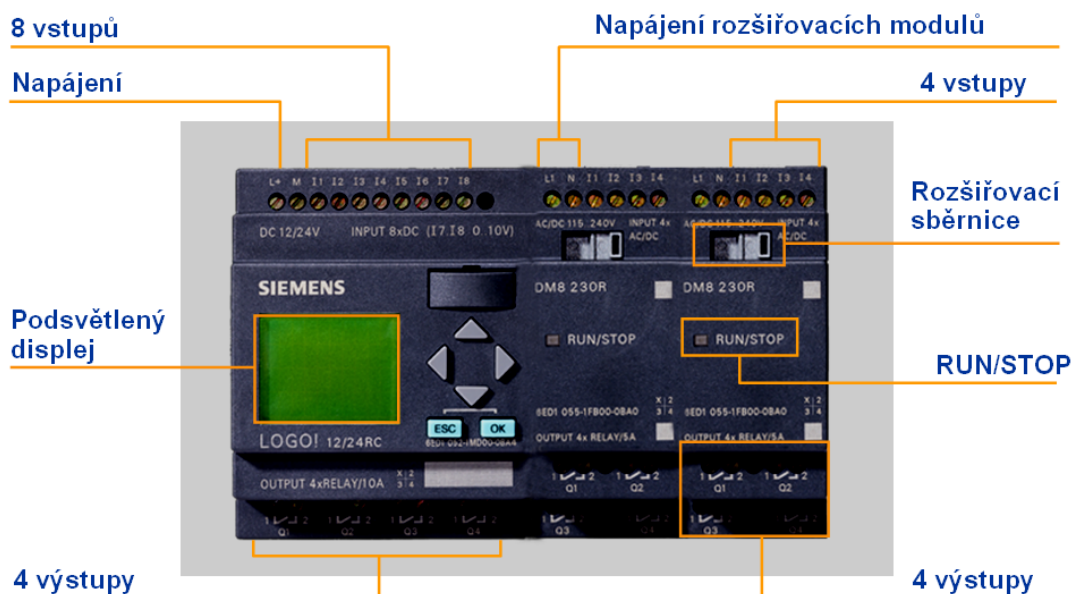
Základnou myšlienkou modulu je spojiť logické, časové a ostatné špeciálne funkcie do jedného modulu, a tým ušetriť prácu pri projekcii a pri realizácii celej aplikácie. Modul LOGO! dokáže ovládať rôzne typy úloh, a to od jednoduchších až po tie zložitejšie. Ovládanie čerpadiel, kompresorov, riadenie osvetlenia patrí medzi úlohy jednoduchšie, ktoré LOGO! zvláda. Medzi tie zložitejšie úlohy patrí sledovanie teplôt, tlaku a ostatných analogových veličín. Pri riešení jednoduchších priemyslových aplikácií sa často využívajú časovače, čítače, komparátory, prúdové pulzné relé apod. Pomocou logického modulu sa dá projekt naprogramovať do jedného zariadenia, ktoré sleduje stav vstupov a podľa potreby spína výstupy. LOGO! sa programuje priamo na displeji alebo je možné použiť prehľadný a jednoduchý vývojový software. Podsvietený displej, ktorým je tento modul vybavený, dokáže zobrazit' 4 x 12 znakov. Modul má aj mnoho výhod, jednou z nich je aj úspora nákladov (menšie náklady na hardware, menej priestoru v rozvád'ačoch, jednoduché programovanie), ďalšia z výhod je flexibilita, modul má rôzne napájacie napätia (vstupy DC12V, AC/DC24V, AC/DC115/230V, analógové vstupy 0...10V, 4...20mA, PT100), výstupy sú napätovo oddelené. Flexibilita modulu je vyznačovaná aj jednoduchým preprogramovaním funkcie priamo na module, na PC alebo pomocou pamäťového modulu.[8]



Obr. 8 Názorný model LOGO!

4.2 Princíp činnosti LOGO!

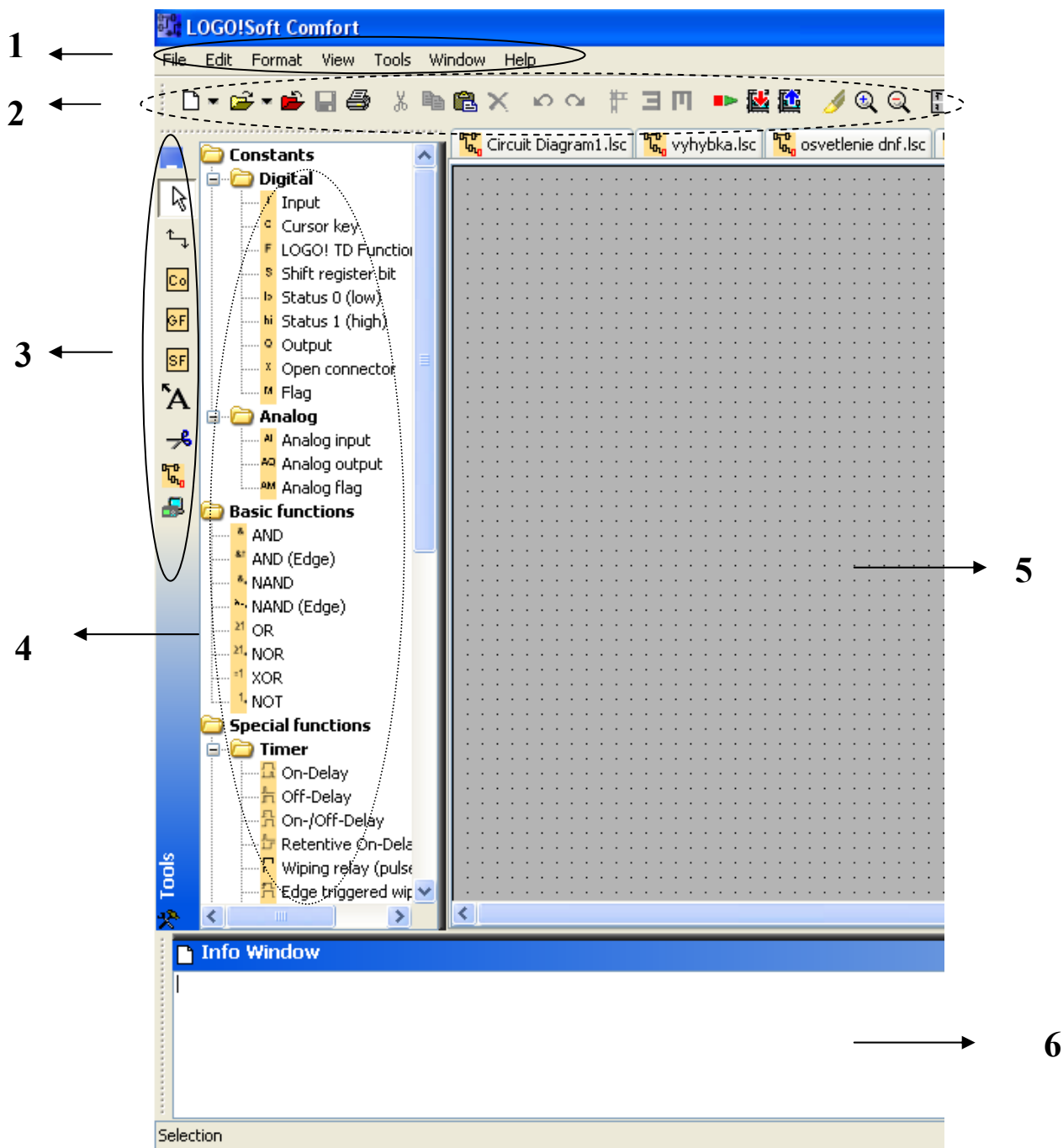
LOGO! obsahuje základné logické a časové funkcie, ale aj špeciálne funkcie. Môže zabezpečiť funkciu niekoľkých časových alebo impulzných relé, má čítače, logické funkcie, komparátory a mnoho ďalších špeciálnych funkcií, ako napríklad pre spracovanie analogových signálov, pre zobrazenie textových správ atď. Základný modul LOGO! má integrovaných 8 vstupov a 4 výstupy, pritom dva vstupy je možné použiť ako vstupy digitálne alebo analógové (0 – 10V). Je tiež možná voľba výstupov, a to či ide o výstupy tranzistorové alebo reléové. Ak nastane prípad, že aplikácia vyžaduje viac vstupov alebo výstupov ako LOGO! ponúka, je možné modul rozšíriť o ďalšie moduly so vstupmi a výstupmi. Maximálna konfigurácia je 24 digitálnych vstupov, 16 digitálnych výstupov, 8 analogových vstupov a 2 analogové výstupy. Rozširovací modul ponúka aj priame pripojenie odporových teplomerov Pt100 bez použitia žiadnych iných prevodníkov.[8]



Obr. 9 Modul LOGO! s popisom jednotlivých častí

4.3 Vývojový software

Vývojové prostredie LOGO!Soft Comfort je jednoduché a prehľadné. Na výber sú tu dva editory, a to funkčné bloky (FBD) alebo kontaktné schéma (LAD). Celý proces programovania znamená len spájanie jednotlivých blokov a tým za niekoľko málo minút sa dá naprogramovať aj pomerne zložitá aplikácia. V starších verziách LOGO! bolo možné vytvorený program len simulovať, dnes je už software rozšírený aj o monitorovanie a testovanie programu.[8]



Obr. 10 Vývojové prostredie LOGO!Soft Comfort

Popis obr. 3.3:

- 1 – Základná lišta Menu
- 2 – Nástrojová lišta
- 3 – Panel nástrojov programovania
- 4 – Konstanty, základné funkcie a špeciálne funkcie
- 5 – Programovacie rozhranie
- 6 – Informačné okno

Základná lišta Menu sa nachádza v hornej časti programu. Sú v nej obsiahnuté príkazy pre editáciu a správu programu.

Nástrojová lišta sa nachádza pod základnou lištou Menu a nad programovacím rozhraním. Po spustení programu sa lišta zobrazí v zjednodušenej forme a ponúka základné funkcie.

Panel nástrojov programovania je viditeľný na ľavej strane programovacieho okna. Ikony sú používané k tvorbe a editácii programu.

Konstanty, základné funkcie a špeciálne funkcie sa nachádzajú hneď vedľa panelu nástrojov programovania. V tejto časti sú obsiahnuté všetky logické funkcie, ktoré je možné využiť pri vytváraní programu.

Programovacie rozhranie je šedá plocha, kde sa vytvára samotný program. Do programovacieho rozhrania sa vkladajú konstanty, základné a špeciálne funkcie, na základe ktorých je program vytvorený.

Informačné okno sa nachádza pod programovacím rozhraním. Toto okno poskytuje informácie a poznámky o stave programu. [11]

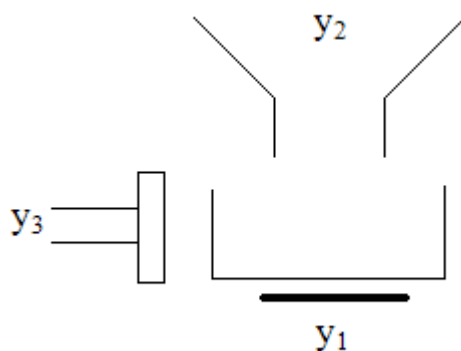
5 Ukázkové úlohy

V praktickej časti tejto bakalárskej práce som navrhla celkovo 6 ukázkových úloh. Prvé tri príklady (Plniaca linka, Signalizácia spotreby, Osvetlenie bytového domu) sú riešené prostredníctvom kombinačných logických obvodov. Nasledujúce príklady (Nádrž s regulovanou hladinou, Výhybka, Železničný prejazd) sú zamerané na sekvenčné logické obvody. Všetky navrhnuté úlohy sa snažia priblížiť princípy riešení používaných v automatizácii.

5.1 Plniaca linka

Zadanie:

Plnenie kelímkov je jedna z operácií na linke (obr. 11). Toto stanovisko je vybavené čidlami pre kontrolu prítomnosti kelímku a kontrolu naplnenia kelímku. Keď je kelímok neprítomný, tak sa prisunie prázdny kelímok. Po jeho naplnení sa odošle k ďalšiemu zpracovaniu.



Obr. 11 Schéma plniacej linky

Popis parametrov:

Vstupy $x_1 = 1$ – kelímok prítomný / 0 – kelímok neprítomný

$x_2 = 1$ – kelímok naplnený / 0 – kelímok nenaplnený

Výstupy y_1 – prisunutie prázdneho kelímku

y_2 – plnenie kelímku

y_3 – odsunutie plného kelímku

Riešenie:

Zo sloveného zadania si zhotovíme pravdivostnú tabuľku (Obr. 12), ktorá sa skladá zo vstupov x_1 a x_2 a výstupov y_1 až y_3 . Pri tvorení pravdivostnej tabuľky kladieme dôraz na to, aby plniaca linka splňovala dané požiadavky.

x_1	x_2	y_1	y_2	y_3
0	0	1	0	0
0	1	-	-	-
1	0	0	1	0
1	1	0	0	1

Obr. 12 Pravdivostná tabuľka Plniacej linky

Každá logická funkcia sa dá previesť na algebraický zápis, ktorý je vyjadrenie logického súčtu logických súčinov. V pravdivostnej tabuľke prechádzame jednotlivé riadky a pri disjunktívnej forme uvažujeme iba tie, v ktorých funkčná hodnota výstupného signálu nadobúda hodnotu 1. Pri konjunktívnej forme sa jedná o zápis logického súčinu logických súčtov. Z tabuľky sa následne vypíšu výstupné signály s nadobudnutou hodnotou 0:

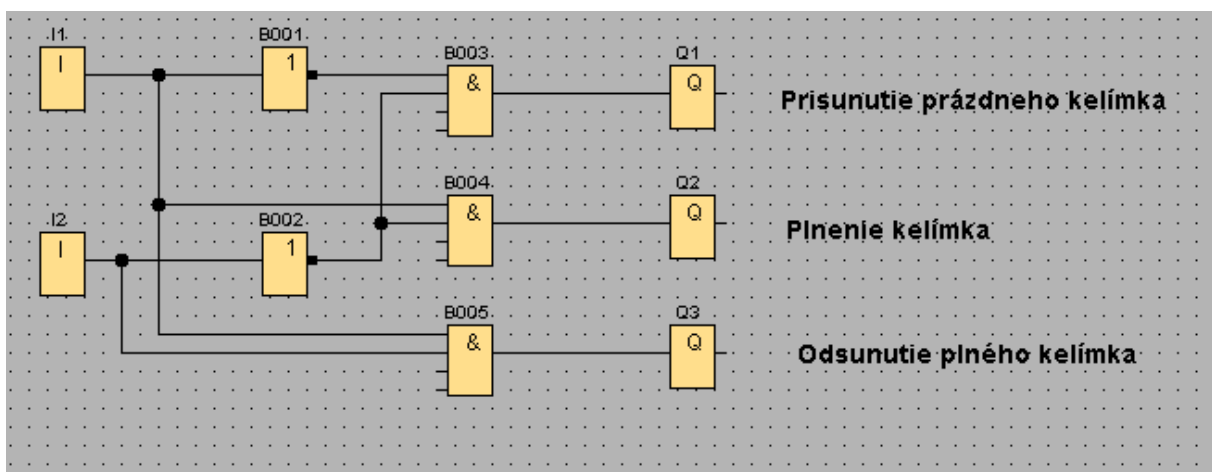
DNF:

$$y_1 = \bar{x}_1 \cdot \bar{x}_2$$

$$y_2 = x_1 \cdot \bar{x}_2$$

$$y_3 = x_1 \cdot x_2$$

V nasledujúcom kroku by sme použili riešenie pomocou karnaughovej mapy, ale nakoľko sa jedná o minimálny tvar algebraického zápisu, dané rovnice nejde už viac zminimalizovať a preto sa dajú rovno použiť na zakreslenie do programu pomocou funkčných blokov (obr. 13).



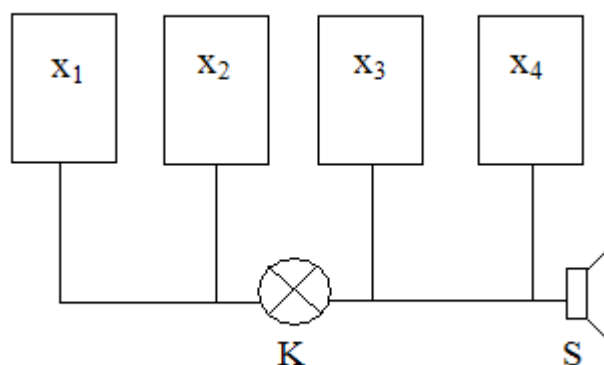
Obr. 13 Blokové schéma Plniacej linky

Na zrealizovanie danej úlohy boli použité dve čidlá, ktoré sledovali prítomnosť a plnosť kelímka.

5.2 Signalizácia spotreby

Zadanie:

V bytovacom zariadení majú v prevádzke 4 kotle na ohrev vody (obr. 14). Počet zopnutých kotlov je závislý na množstve odberu teplej vody. Pri normálnej prevádzke je zopnutý jeden alebo dva kotle. Pri zvýšenom odbere teplej vody sa zopne tretí pomocný kotol a rozsvieti sa kontrolka zvýšeného odberu vody. Pri nadmernom odbere teplej vody sa zopne aj štvrtý záložný kotol. Nadmerný odber teplej vody je ohlásený zvukovou sirénou. Zopnutie kotla je ovládané čidlom.



Obr. 14 Schéma Signalizácie spotreby

Popis parametrov:

Vstupy: Kotel 1 – $x_1 = 1$ - Kotel je zapnutý/ 0 - Kotel je vypnutý
 Kotel 2 – $x_2 = 1$ - Kotel je zapnutý/ 0 - Kotel je vypnutý
 Kotel 3 – $x_3 = 1$ - Kotel je zapnutý/ 0 - Kotel je vypnutý
 Kotel 4 – $x_4 = 1$ - Kotel je zapnutý/ 0 - Kotel je vypnutý

Výstupy: Kontrolka K = 1 – Kontrolka svieti/ 0 – Kontrolka nesvieti
 Siréna S = 1 – Siréna húka/ 0 – Siréna nehúka

Postup riešenia:

Zo slovného zadania si zostavíme pravdivostnú tabuľku obr. 15, v ktorej sú obsiahnuté vstupy x_1 až x_4 a výstupy K a S.

x_1	x_2	x_3	x_4	K	S
0	0	0	0	0	0
0	0	0	1	0	0
0	0	1	0	0	0
0	0	1	1	0	0
0	1	0	0	0	0
0	1	0	1	0	0
0	1	1	0	0	0
0	1	1	1	1	0
1	0	0	0	0	0
1	0	0	1	0	0
1	0	1	0	0	0
1	0	1	1	1	0
1	1	0	0	0	0
1	1	0	1	1	0
1	1	1	0	1	0
1	1	1	1	1	1

Obr. 15 Pravdivostná tabuľka signalizácie spotreby

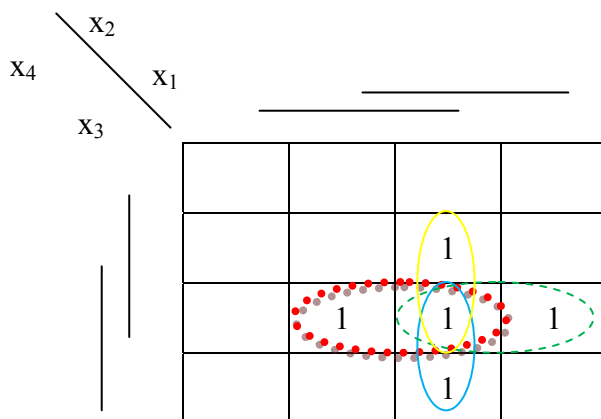
DNF:

$$K = \bar{x}_1 \cdot x_2 \cdot x_3 \cdot x_4 + x_1 \cdot \bar{x}_2 \cdot x_3 \cdot x_4 + x_1 \cdot x_2 \cdot \bar{x}_3 \cdot x_4 + x_1 \cdot x_2 \cdot x_3 \cdot \bar{x}_4 + x_1 \cdot x_2 \cdot x_3 \cdot x_4$$

$$S = x_1 \cdot x_2 \cdot x_3 \cdot x_4$$

Algebraickú rovnicu K zminimalizujeme pomocou zápisu do Karnaughovej mapy (Obr. 16). Rovnica S nejde minimalizovať, nakoľko sa v tomto prípade jedná o minimálny tvar.

Výstup K:

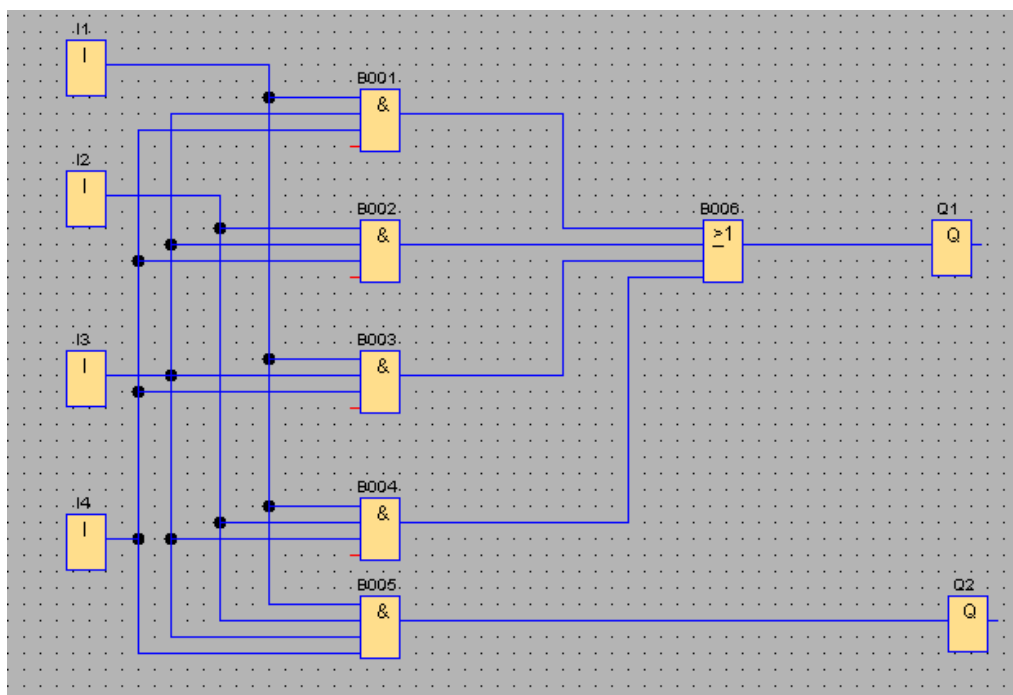


Obr. 16 Karnaughova mapa

Z Karnaughovej mapy získame funkciu v minimálnom tvare:

$$K = x_1 \cdot x_3 \cdot x_4 + x_2 \cdot x_3 \cdot x_4 + x_1 \cdot x_2 \cdot x_3 + x_1 \cdot x_2 \cdot x_4$$

Takto zminimalizovaná funkcia sa využíva pre realizáciu blokového schématu, ktoré sme vytvorili pomocou funkčných blokov v programe LOGO! Soft Comfort.



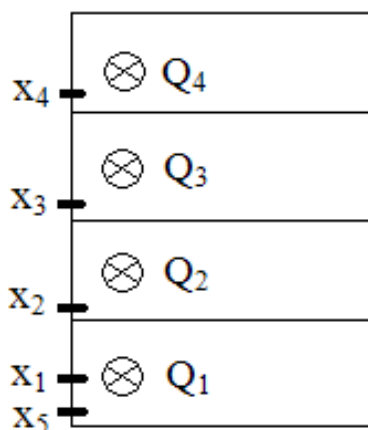
Obr. 17 Schéma zapojenia v LOGO! Soft Comfort

K vyriešeniu úlohy boli použité štyri čidlá. Každé jedno poskytovalo informáciu o stave kotla, tzn. či je kotol v prevádzke alebo nie. Po zopnutí troch čidiel sa rozsvietila svetelná signalizácia, po zopnutí štyroch sa rozorvučala zvuková siréna.

5.3 Osvetlenie v obytnom dome

Zadanie:

Obytný dom má štyri poschodia. Osvetlenie je realizované pomocou čidiel, ktoré slúžia ako senzory pohybu (obr. 18). Na každom poschodí je umiestnené takéto čidlo, ktoré rozsvieti svetlo na poschodí, kde je zaznamenaný pohyb. Svetlo sa nerozžne iba na poschodí kde senzor zaznamenal pohyb, ale aj v najbližších poschodiach, čiže poschodie pod a poschodie nad. Osvetlenie funguje pri určitej intenzite osvetlenia a za tmy.



Obr. 18 Schéma Osvetlenia bytového domu

Popis parametrov:

Vstupy: x_1 – Senzor pohybu na 1. poschodí
 x_2 – Senzor pohybu na 2. poschodí
 x_3 – Senzor pohybu na 3. poschodí
 x_4 – Senzor pohybu na 4. poschodí
 $x_1, x_2, x_3, x_4 = 1$ – Prítomnosť pohybu / 0 – Neprítomnosť pohybu

Výstupy: $Q_1, Q_2, Q_3, Q_4 = 1$ – Svetlo svieti / 0 – Svetlo nesvieti

Postup riešenia:

Zhotovíme si pravdivostnú tabuľku (Obr. 19) :

x_1	x_2	x_3	x_4	Q_1	Q_2	Q_3	Q_4
0	0	0	0	0	0	0	0
0	0	0	1	0	0	1	1
0	0	1	0	0	1	1	1
0	0	1	1	0	1	1	1
0	1	0	0	1	1	1	0
0	1	0	1	1	1	1	1
0	1	1	0	1	1	1	1
0	1	1	1	1	1	1	1
1	0	0	0	1	1	0	0
1	0	0	1	1	1	1	1
1	0	1	0	1	1	1	1
1	0	1	1	1	1	1	1
1	1	0	0	1	1	1	0
1	1	0	1	1	1	1	1
1	1	1	0	1	1	1	1
1	1	1	1	1	1	1	1

Obr. 19 Pravdivostná tabuľka realizácie osvetlenia

Z tabuľky si vypíšeme výstupy, ktoré nadobúdajú logickú hodnotu 1, a tým použijeme disjunktívnu formu zápisu. Následne zhotovíme blokové schéma, ktoré je zobrazené na obr. 24.

$$\begin{aligned}
 Q_1 = & (\bar{x}_1 \cdot x_2 \cdot \bar{x}_3 \cdot \bar{x}_4) + (\bar{x}_1 \cdot x_2 \cdot \bar{x}_3 \cdot x_4) + (\bar{x}_1 \cdot x_2 \cdot x_3 \cdot \bar{x}_4) + \\
 & + (\bar{x}_1 \cdot x_2 \cdot x_3 \cdot x_4) + (x_1 \cdot \bar{x}_2 \cdot \bar{x}_3 \cdot \bar{x}_4) + (x_1 \cdot \bar{x}_2 \cdot \bar{x}_3 \cdot x_4) + \\
 & + (x_1 \cdot \bar{x}_2 \cdot x_3 \cdot \bar{x}_4) + (x_1 \cdot \bar{x}_2 \cdot x_3 \cdot x_4) + (x_1 \cdot x_2 \cdot \bar{x}_3 \cdot \bar{x}_4) + \\
 & + (x_1 \cdot x_2 \cdot \bar{x}_3 \cdot x_4) + (x_1 \cdot x_2 \cdot x_3 \cdot \bar{x}_4) + (x_1 \cdot x_2 \cdot x_3 \cdot x_4)
 \end{aligned}$$

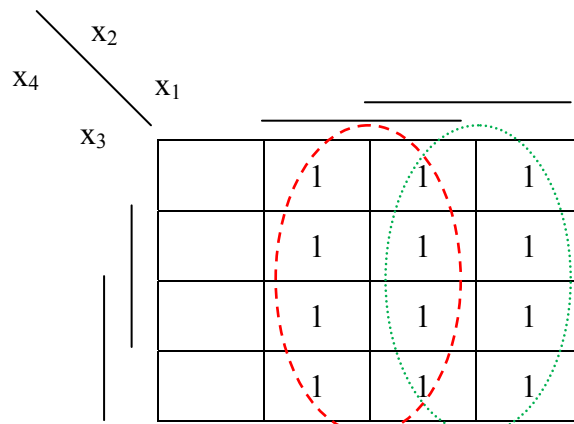
$$\begin{aligned}
 Q_2 = & (\bar{x}_1 \cdot \bar{x}_2 \cdot x_3 \cdot \bar{x}_4) + (\bar{x}_1 \cdot \bar{x}_2 \cdot x_3 \cdot x_4) + (\bar{x}_1 \cdot x_2 \cdot \bar{x}_3 \cdot \bar{x}_4) + \\
 & + (\bar{x}_1 \cdot x_2 \cdot \bar{x}_3 \cdot x_4) + (\bar{x}_1 \cdot x_2 \cdot x_3 \cdot \bar{x}_4) + (\bar{x}_1 \cdot x_2 \cdot x_3 \cdot x_4) + \\
 & + (x_1 \cdot \bar{x}_2 \cdot \bar{x}_3 \cdot \bar{x}_4) + (x_1 \cdot \bar{x}_2 \cdot \bar{x}_3 \cdot x_4) + (x_1 \cdot \bar{x}_2 \cdot x_3 \cdot \bar{x}_4) + \\
 & + (x_1 \cdot \bar{x}_2 \cdot x_3 \cdot x_4) + (x_1 \cdot x_2 \cdot \bar{x}_3 \cdot \bar{x}_4) + (x_1 \cdot x_2 \cdot \bar{x}_3 \cdot x_4) + \\
 & + (x_1 \cdot x_2 \cdot x_3 \cdot \bar{x}_4) + (x_1 \cdot x_2 \cdot x_3 \cdot x_4)
 \end{aligned}$$

$$\begin{aligned}
 Q_3 = & (\bar{x}_1 \cdot \bar{x}_2 \cdot \bar{x}_3 \cdot x_4) + (\bar{x}_1 \cdot \bar{x}_2 \cdot x_3 \cdot \bar{x}_4) + (\bar{x}_1 \cdot \bar{x}_2 \cdot x_3 \cdot x_4) + \\
 & + (\bar{x}_1 \cdot x_2 \cdot \bar{x}_3 \cdot \bar{x}_4) + (\bar{x}_1 \cdot x_2 \cdot \bar{x}_3 \cdot x_4) + (\bar{x}_1 \cdot x_2 \cdot x_3 \cdot \bar{x}_4) + \\
 & + (\bar{x}_1 \cdot x_2 \cdot x_3 \cdot x_4) + (x_1 \cdot \bar{x}_2 \cdot \bar{x}_3 \cdot x_4) + (\bar{x}_1 \cdot \bar{x}_2 \cdot x_3 \cdot \bar{x}_4) + \\
 & + (x_1 \cdot \bar{x}_2 \cdot x_3 \cdot x_4) + (x_1 \cdot x_2 \cdot \bar{x}_3 \cdot \bar{x}_4) + (x_1 \cdot x_2 \cdot \bar{x}_3 \cdot x_4) + \\
 & + (x_1 \cdot x_2 \cdot x_3 \cdot \bar{x}_4) + (x_1 \cdot x_2 \cdot x_3 \cdot x_4)
 \end{aligned}$$

$$\begin{aligned}
 Q_4 = & (\bar{x}_1 \cdot \bar{x}_2 \cdot \bar{x}_3 \cdot x_4) + (\bar{x}_1 \cdot \bar{x}_2 \cdot x_3 \cdot \bar{x}_4) + (\bar{x}_1 \cdot \bar{x}_2 \cdot x_3 \cdot x_4) + \\
 & + (\bar{x}_1 \cdot x_2 \cdot \bar{x}_3 \cdot x_4) + (\bar{x}_1 \cdot x_2 \cdot x_3 \cdot \bar{x}_4) + (\bar{x}_1 \cdot x_2 \cdot x_3 \cdot x_4) + \\
 & + (x_1 \cdot \bar{x}_2 \cdot \bar{x}_3 \cdot x_4) + (x_1 \cdot \bar{x}_2 \cdot x_3 \cdot \bar{x}_4) + (x_1 \cdot \bar{x}_2 \cdot x_3 \cdot x_4) + \\
 & + (x_1 \cdot x_2 \cdot \bar{x}_3 \cdot x_4) + (x_1 \cdot x_2 \cdot x_3 \cdot \bar{x}_4) + (x_1 \cdot x_2 \cdot x_3 \cdot x_4)
 \end{aligned}$$

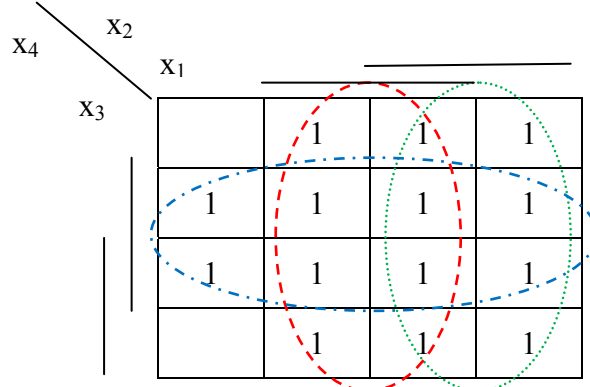
Tvary pre výstupy vypísané z tabuliek sa minimalizujú za pomoci Karnaughových máp (obr. 20, obr. 21, obr. 22, obr. 23):

Výstup Q₁:



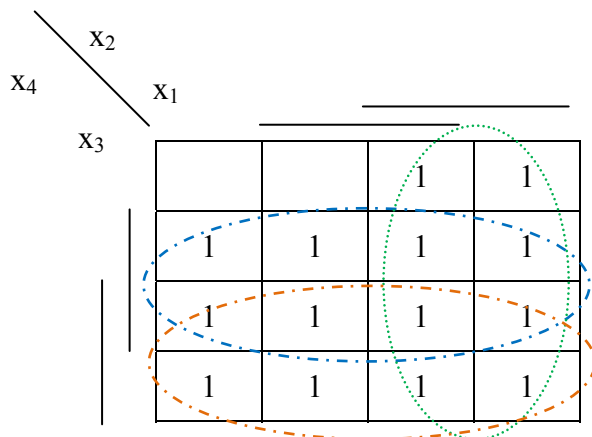
Obr. 20 Karnaughova mapa pre výstup Q₁

Výstup Q₂:



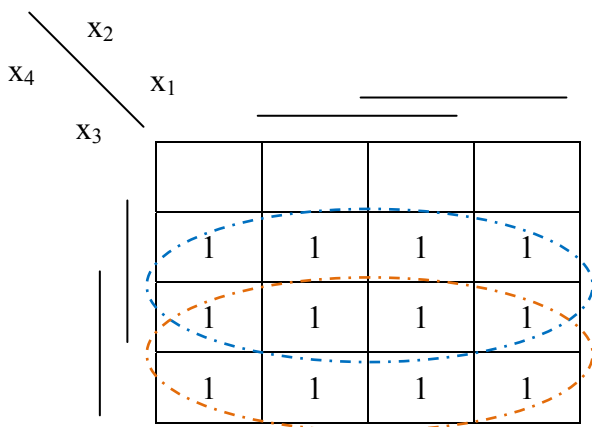
Obr. 21 Karnaughova mapa pre výstup Q₂

Výstup Q_3 :



Obr. 22 Karnaughova mapa pre výstup Q_3

Výstup Q_4 :



Obr. 23 Karnaughova mapa pre výstup Q_4

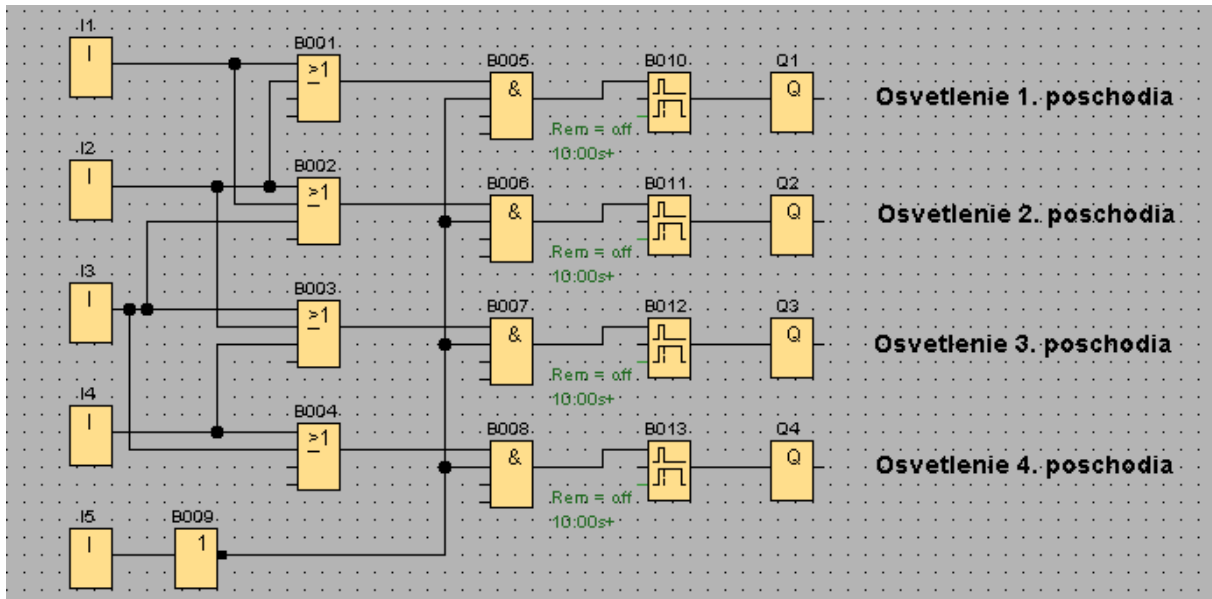
Zo všetkých štyroch karnaughových máp sme získali zminimalizované výrazy pre jednotlivé výstupy a za pomoci týchto rovníc sme zostavili blokové schéma osvetlenia obr. 24:

$$Q_1 = x_1 + x_2$$

$$Q_2 = x_1 + x_2 + x_3$$

$$Q_3 = x_2 + x_3 + x_4$$

$$Q_4 = x_3 + x_4$$



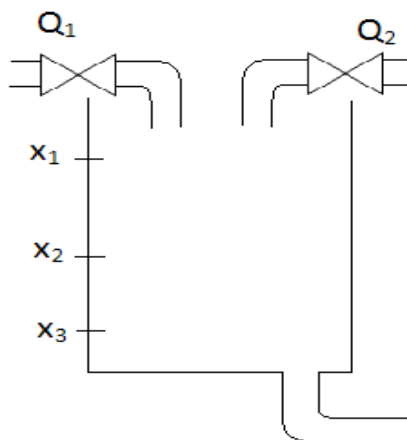
Obr. 24 Blokové schéma Osvetlenia bytového domu

Na vyriešenie tejto úlohy bolo potrebných celkovo päť čidiel, z toho štyri nám slúžili ako senzory pohybu a jedno bolo využité ako blokácia svetiel za denného svetla. V zapojení je využitý časovač pre vypnutie svetiel po určitom časovom intervale, aby úloha nadobudla realnosť.

5.4 Nádrž s regulovanou hladinou

Zadanie:

Vodáreň zásobuje obytný dom pitnou vodou. Zásobník na vodu je osadený čidlami pre kontrolu hladiny vody (obr. 25). Nádrž je napúšťaná vodou za pomoci prítokových čerpadiel Q_1 a Q_2 . Pri nízkom odbere sa hladina vody udržiava na úrovni čidla x_1 pomocou prítokového čerpadla Q_1 . Toto čerpadlo je samostatne v prevádzke až kým nedosiahne úroveň čidla x_3 . Pri zvýšenom odbere vody nestíha dopúšťať čerpadlo Q_1 a hladina klesne pod úroveň čidla x_3 , tak sa zopne pomocné prítokové čerpadlo Q_2 , ktoré dopúšťa vodu až po úroveň čidla x_2 .



Obr. 25 Schéma Nádrže s regulovanou hladinou

Popis parametrov:

Vstupy čidlo $x_1 = 1$ – Hladina vody nad čidlom / 0 – Hladina vody pod čidlom
 čidlo $x_2 = 1$ – Hladina vody nad čidlom / 0 – Hladina vody pod čidlom
 čidlo $x_3 = 1$ – Hladina vody nad čidlom / 0 – Hladina vody pod čidlom

Výstupy čerpadlo $Q_1 = 1$ – čerpadlo napúšťa / 0 – čerpadlo nenapúšťa
 čerpadlo $Q_2 = 1$ – čerpadlo napúšťa / 0 – čerpadlo nenapúšťa

Postup riešenia:

x_1	x_2	x_3	Q_1	Q_2
0	0	0	1	1
0	0	1	1	x
0	1	0	-	-
0	1	1	1	0
1	0	0	-	-
1	0	1	-	-
1	1	0	-	-
1	1	1	0	0

Obr.26 Pravdivostná tabuľka nádrže

- /Stav, ktorý nemôže nastať/

x /Zakázaný stav/

V tomto prípade môžeme vidieť, že na vyriešenie úlohy potrebujeme poznať predchádzajúci stav, nakoľko nie sme schopní rozhodnúť akú hodnotu nadobudne výstupný signál. Preto si zostavíme pravdivostnú tabuľku už aj so spomínaným predchádzajúcim stavom.

x_1	x_2	x_3	Q'_2	Q_1	Q_2
0	0	0	0	1	1
0	0	0	1	1	1
0	0	1	0	1	0
0	0	1	1	1	1
0	1	0	0	-	-
0	1	0	1	-	-
0	1	1	0	1	0
0	1	1	1	1	0
1	0	0	0	-	-
1	0	0	1	-	-
1	0	1	0	-	-
1	0	1	1	-	-
1	1	0	0	-	-
1	1	0	1	-	-
1	1	1	0	0	0
1	1	1	1	0	0

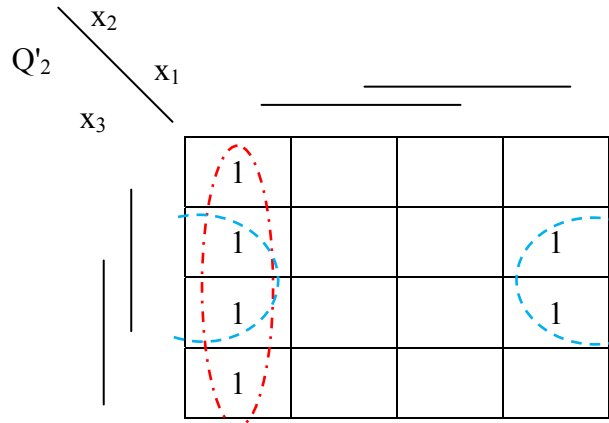
Obr. 27 Pravdivostná tabuľka nádrže (s predchádzajúcim stavom)

DNF:

$$Q_1 = \bar{x}_1 \cdot \bar{x}_2 \cdot \bar{x}_3 \cdot \bar{Q}'_2 + \bar{x}_1 \cdot \bar{x}_2 \cdot \bar{x}_3 \cdot Q'_2 + \bar{x}_1 \cdot \bar{x}_2 \cdot x_3 \cdot \bar{Q}'_2 + \bar{x}_1 \cdot \bar{x}_2 \cdot x_3 \cdot Q'_2 + \bar{x}_1 \cdot x_2 \cdot x_3 \cdot \bar{Q}'_2 + \bar{x}_1 \cdot x_2 \cdot x_3 \cdot Q'_2$$

$$Q_2 = \bar{x}_1 \cdot \bar{x}_2 \cdot \bar{x}_3 \cdot \bar{Q}'_2 + \bar{x}_1 \cdot \bar{x}_2 \cdot \bar{x}_3 \cdot Q'_2 + \bar{x}_1 \cdot \bar{x}_2 \cdot x_3 \cdot Q'_2$$

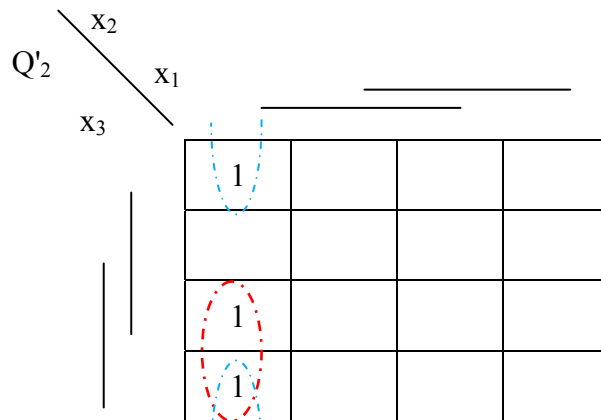
Výstup Q₁:



Obr.28 Karnaughova mapa pre výstup Q₁

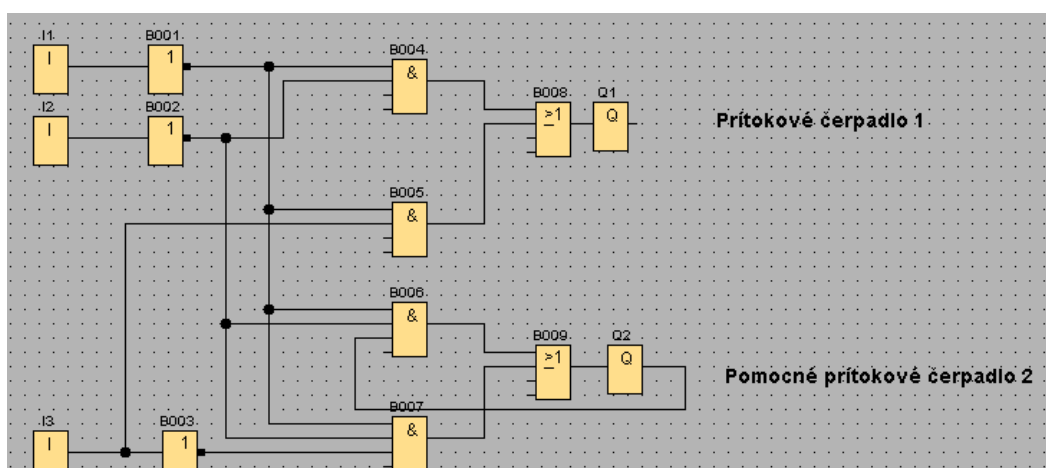
$$Q_1 = \bar{x}_1 \cdot \bar{x}_2 + \bar{x}_1 \cdot x_3$$

Výstup Q₂:



Obr. 29 Karnaughova mapa pre výstup Q₂

$$Q_2 = \bar{x}_1 \cdot \bar{x}_2 \cdot Q'_2 + \bar{x}_1 \cdot \bar{x}_2 \cdot \bar{x}_3$$



Obr. 30 Blokové schéma nádrže s regulovanou hľadinou

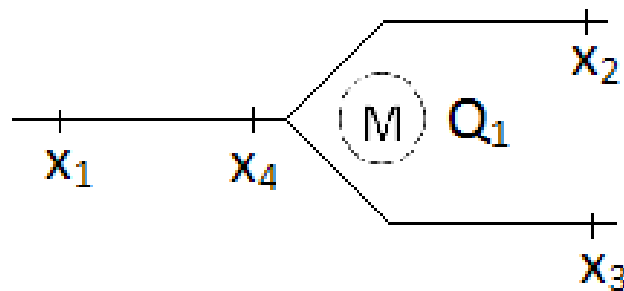
Riešenie úlohy spočívalo v použití troch čidiel, ktoré spĺňali funkciu sledovania výšky hladiny vody. Ak výška hladiny vody klesla pod najnižšie uložené čidlo, v tomto prípade sa jedná o vstup I3, tak sa zoplo pomocné prítokové čerpadlo. Na blokovom schémate je možné

vidieť, že z výstupu Q_2 je vedená spätná väzba na jeden z logických členov použitých v zapojení, čo značí, že v úlohe sa prejavuje sekvenčnosť.

5.5 Výhybka

Zadanie:

Táto úloha je zameraná na ovládanie výhybky na koľajisti. Pri príchode električky z niektorého z nástupíšť sa výhybka automaticky prekllopí na požadovaný smer, aby sa električka mohla dostať na spoločnú koľaj. Pri prízjazde električky zľava je detekovaný signál pre určenie nástupišťa, ku ktorému má prísť (obr. 31).



Obr. 31 Schéma Výhybky

Popis parametrov:

Vstupy $x_1 = 1$ – Príchod z ľava / 0 – Neprítomnosť električky
 $x_2 = 1$ – Príchod zhora / 0 – Neprítomnosť električky
 $x_3 = 1$ – Príchod zdola / 0 – Neprítomnosť električky
 $x_4 = 1$ – Signál pre výhybku dohora / 0 – Signál pre výhybku dole

Výstupy $Q_1 = 1$ – Výhybka hore / 0 – Výhybka dole

Riešenie:

x_1	x_2	x_3	x_4	Q'_1	Q_1
0	0	0	0	0	0
0	0	0	0	1	1
0	0	0	1	0	0
0	0	0	1	1	1
0	0	1	0	0	0

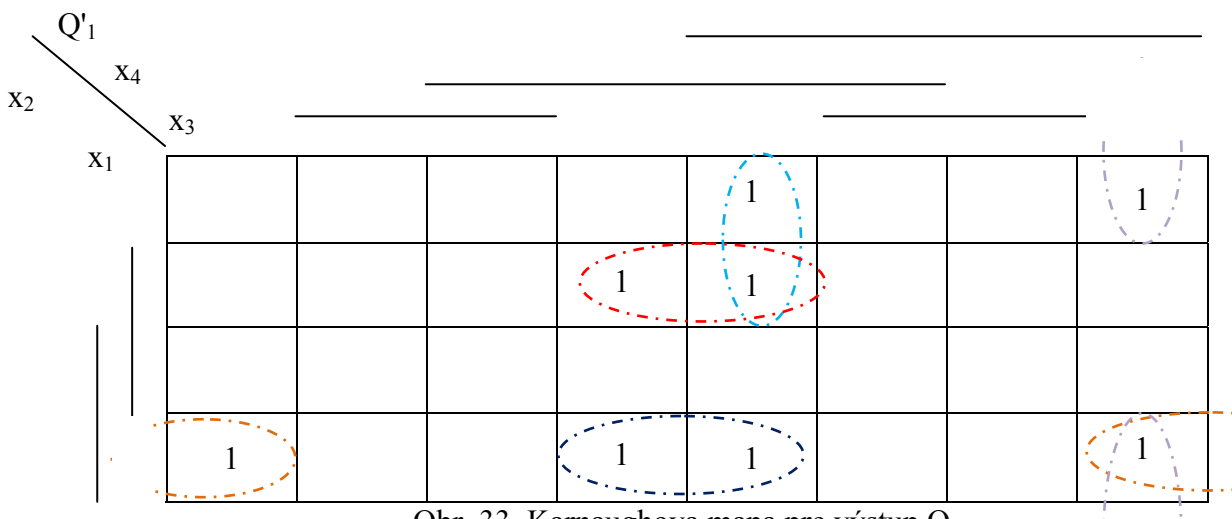
0	0	1	0	1	0
0	0	1	1	0	0
0	0	1	1	1	0
0	1	0	0	0	1
0	1	0	0	1	1
0	1	0	1	0	1
0	1	0	1	1	1
0	1	1	0	0	-
0	1	1	0	1	-
0	1	1	1	0	-
0	1	1	1	1	-
1	0	0	0	0	0
1	0	0	0	1	0
1	0	0	1	0	1
1	0	0	1	1	1
1	0	1	0	0	-
1	0	1	0	1	-
1	0	1	1	0	-
1	0	1	1	1	-
1	1	0	0	0	-
1	1	0	0	1	-

1	1	0	1	0	-
1	1	0	1	1	-
1	1	1	0	0	-
1	1	1	0	1	-
1	1	1	1	0	-
1	1	1	1	1	-

Obr. 32 Pravdivostná tabuľka výhybky

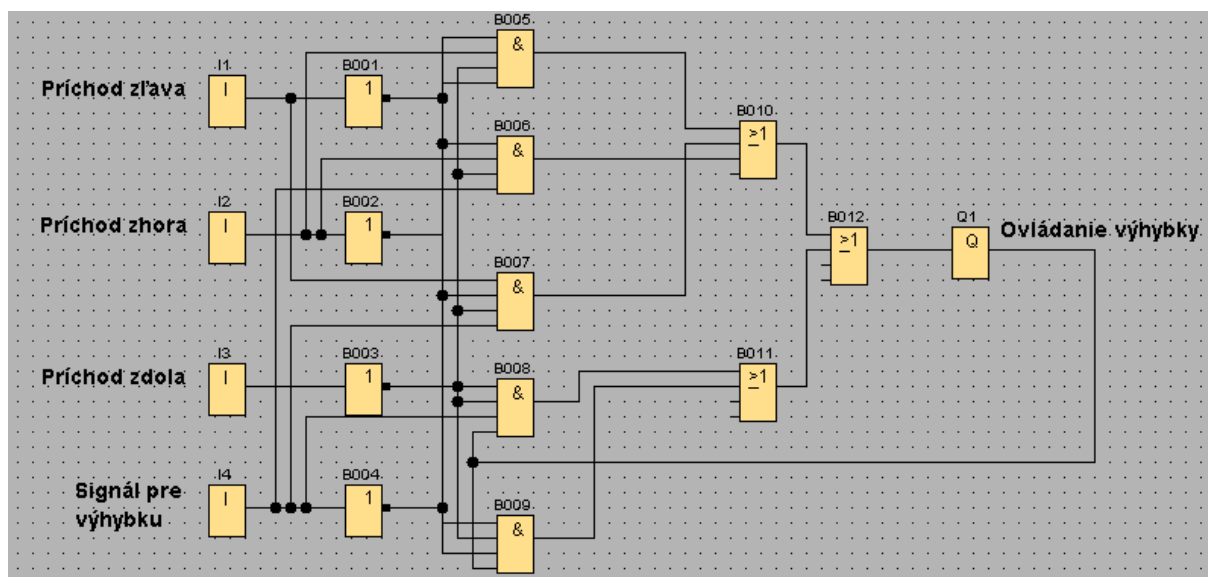
DNF

$$Q_1 = \bar{x}_1 \cdot \bar{x}_2 \cdot \bar{x}_3 \cdot \bar{x}_4 \cdot Q'_1 + \bar{x}_1 \cdot \bar{x}_2 \cdot \bar{x}_3 \cdot x_4 \cdot Q'_1 + \bar{x}_1 \cdot x_2 \cdot \bar{x}_3 \cdot \bar{x}_4 \cdot \bar{Q}'_1 + \bar{x}_1 \cdot x_2 \cdot \bar{x}_3 \cdot \bar{x}_4 \cdot Q'_1 + \bar{x}_1 \cdot x_2 \cdot \bar{x}_3 \cdot x_4 \cdot \bar{Q}'_1 + \bar{x}_1 \cdot x_2 \cdot \bar{x}_3 \cdot x_4 \cdot Q'_1 + x_1 \cdot \bar{x}_2 \cdot \bar{x}_3 \cdot x_4 \cdot \bar{Q}'_1 + x_1 \cdot \bar{x}_2 \cdot \bar{x}_3 \cdot x_4 \cdot Q'_1$$



Obr. 33 Karnaughova mapa pre výstup Q_1

$$Q_1 = \bar{x}_1 \cdot x_2 \cdot \bar{x}_3 \cdot \bar{x}_4 + \bar{x}_1 \cdot x_2 \cdot \bar{x}_3 \cdot x_4 + x_1 \cdot \bar{x}_2 \cdot \bar{x}_3 \cdot x_4 + \bar{x}_2 \cdot \bar{x}_3 \cdot x_4 \cdot Q'_1 + \bar{x}_1 \cdot \bar{x}_3 \cdot \bar{x}_4 \cdot Q'_1$$



Obr. 34 Blokové schéma výhybky

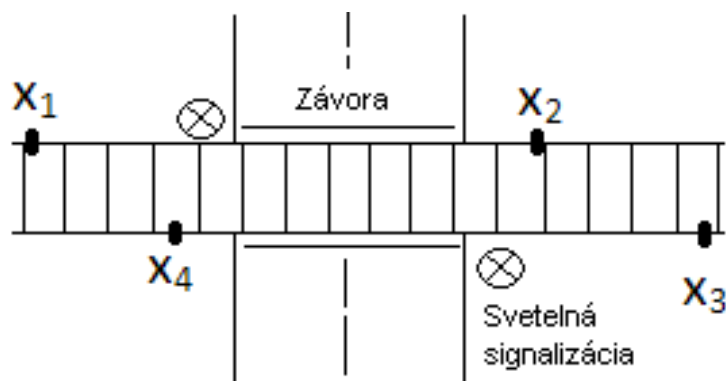
Úloha je vyriešená za pomoci štyroch snímačov. Tri slúžia na určenie prítomnosti zariadenia na koľaji, štvrtý snímač je použitý pre určenie smeru jazdy pri príchode zľava a spĺňa funkciu dávania povelu ovládaciemu motoru na preklopenie výhybky. V úlohe bolo riešené preklápanie koľajovej výhybky na základe signálu, ktorý obdrží príslušný snímač.

5.6 Železničný prejazd

Zadanie:

Železničný prejazd je zabezpečený závorami a svetelnou signalizáciou. Vlak približujúci sa k prejazdu zaznamená čidlo x_1 (príchod zľava) alebo čidlo x_2 (príchod zprava), v tomto okamžiku sa zapne signalizácia červených svetiel a následne po 10 sekundách sa spustí závora. Akonáhle vlak opustí železničný prejazd, čidlo umiestnené za prejazdom dá povel motoru k zdvihnutiu závor.

Základ úlohy je použitý z literatúry [11], kde je uvedené aj jej vyriešenie. Použitá úloha je vyriešená z hľadiska prejazdu vlaku z jednej strany, v tejto bakalárskej práci je úloha rozšírená o prejazd vlaku z oboch strán.



Obr. 35 Schéma Železničného prejazdu

Popis parametrov:**Vstupy**

x_1, x_2, x_3, x_4 = 1 – Čidlo zaznamenalo prejazd vlaku / 0 – Prejazd vlaku nezaznamenaný

Výstupy

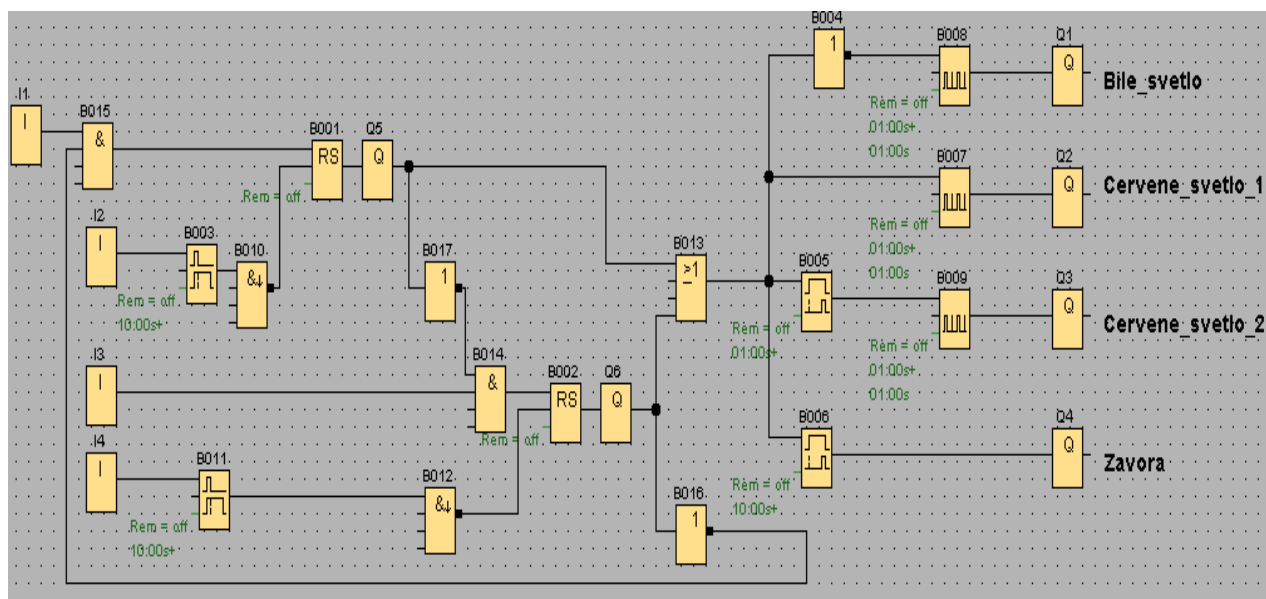
Q_1 = 1 – Biele svetlo bliká / 0 – Biele svetlo neblinká

Q_2, Q_3 = 1 – Červené svetlo bliká / 0 – Červené svetlo neblinká

Q_4 = 1 – Závora spustená dolu / 0 – Závora zdvihnutá hore

Popis riešenia obojsmerného prejazdu vlaku:

Pokiaľ je prejazd prázdny, tak na výstupe čidiel Q_5 a Q_6 je logická nula. Pri príchode vlaku z ľavej strany sa zopne čidlo I_1 . Signál prejde cez blokáciu ľavej strany ($B015$), pretože prejazd bol predtým prázdny a nebolo nutné čidla blokovať. Tento signál prekopí RS klopný obvod ($B001$) do logickej jednotky a tým sa cez súčtové hradlo ($B013$) pošle signál o prítomnosti vlaku a prejazd sa zabezpečí výstražnými svetlami a spustením závory. Signál logickej jednotky z tohto RS obvodu ($B001$) zároveň zablokuje čidla pre prejazd z pravej strany. Táto blokácia sa prevádza pomocou hradla NOT ($B017$), kde sa logická jednotka z RS klopného obvodu ($B001$) prevedie na logickú nulu a tým zablokuje hradlo AND ($B014$).



Obr. 36 Blokové schéma železničného prejazdu

V tejto úlohe je vyriešené ovládanie závory a svetelnej signalizácie pre obojsmerný prejazd vlaku cez priecestie. Ovládanie požadovaných výstupov je zabezpečené dvomi čidlami (I_1, I_2) na prejazd vlaku zľava a dvomi čidlami na prejazd vlaku zprava (I_3, I_4).

6 Záver

Cieľom bakalárskej práce bolo zoznámiť sa s programom LOGO!Soft Comfort, následne vytvoriť ukážkové úlohy kombinačných logických obvodov a sekvenčných logických obvodov.

V teoretickej časti práce sú poskytnuté základné informácie o logickom riadení, kde je priblížené stručné zhrnutie použitej teórie, ktorá bola aplikovaná pri tvorbe ukážkových úloh. Táto kapitola popisuje Booleovu algebru, možnosti vyjadrenia logickej funkcie a jej následnú minimalizáciu. V práci sú charakterizované kombinačné a sekvenčné logické obvody. Pri riešení úlohy železničného prejazdu bol použitý RS klopný obvod, ktorý patrí do skupiny sekvenčných obvodov. Z tohto hľadiska je podrobne rozobraný v teoretickej časti práce. Nasledujúca kapitola s názvom Programovateľné automaty oboznamuje s možnosťami využitia PLC, so štruktúrou tohto automatu a s jeho výhodami a nevýhodami. Teoretická časť ďalej približuje modul LOGO! a vývojový softvér LOGO! Soft Comfort.

V praktickej časti bakalárskej práce bolo navrhnutých šesť ukážkových úloh. Ukážkové úlohy sú rozdelené na tri úlohy kombinačného logického obvodu a tri úlohy sekvenčného logického obvodu. Pri vypracovávaní každej úlohy bol základ riešenia vždy rovnaký. Zhotovenie pravdivostnej tabuľky na základe slovného popisu úlohy nám určilo nastanie možných stavov, ktoré boli požadované. Z pravdivostnej tabuľky boli vypísané výsledné funkcie, ktoré boli následne zminimalizované za pomoci karnaughovej mapy. Tieto zminimalizované výrazy slúžili na zhotovenie blokového schémata v programe LOGO!Soft Comfort za pomoci funkčných blokov. Navrhnuté úlohy sú v práci zoradené podľa obtiažnosti zadania. Prvé tri zadania sú úlohy kombinačného logického obvodu.

Prvá úloha má názov Plniaca linka, kde plní funkciu plnenia kelímok. Druhá úloha s názvom Signalizácia spotreby poskytovala informáciu ohľadom prevádzky kotlov v zariadení. Tretia úloha kombinačného logického obvodu má názov Osvetlenie v bytovom dome.

Nasledujúce tri zadania sú úlohy sekvenčného logického obvodu. Rovnako ako pri zadaniach kombinačného logického obvodu, tieto úlohy sú radené v práci podľa obtiažnosti. Prvá úloha sekvenčného logického obvodu sa zaoberá s reguláciou výšky hladiny vody v nádrži. Druhá úloha s názvom Výhybka rieši preklápanie koľajovej výhybky. Tretia úloha a zároveň posledná je úloha železničného prejazdu, kde je ovládaná svetelná signalizácia a závora.

Overenie správnej funkčnosti všetkých navrhnutých úloh bolo zrealizované prostredníctvom programovateľného automatu. Tieto navrhnuté úlohy by mali poslúžiť ako výuková pomocka pre študentov do predmetu Automatizácia.

Zoznam použitej literatúry

[1] ZÍTEK, P.; HOFREITER, M.; HLAVA, J. *Automatické řízení*. Praha : Vydavatelství ČVUT, 2001. 148s. ISBN 80-01-02044-4

[2] ŠVARC, I.; ŠEDA, M.; VÍTEČKOVÁ, M. *Automatické řízení*. Brno : Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2007. 324s. ISBN 978-80-214-3491-2

[3] ŠVARC, I. *AUTOMATIZACE – Automatické řízení*. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2005. 262 s. ISBN 80-214-2943-7

[4] ZÍTEK, P.; HOFREITER, M.; HLAVA, J. *Příklady a návody z automatického řízení*. Praha : Vydavatelství ČVUT, 2002. 139s. ISBN 80-01-02447-4

[5] WIKIPEDIA. *Sekvenčný obvod* [online]. 4.4.2010 [cit. 6.5.2010]. Dostupné z: http://sk.wikipedia.org/wiki/Sekven%C4%8Dn%C3%BD_obvod

[6] TŮMA, J. *Logické řízení*, [online]. [cit. 25.3.2010] Dostupné z: <http://homel.vsb.cz/~tum52/download/LogickeRizeni.pdf>

[7] ZEZULKA, F.; BRADÁČ, Z.; FIEDLER, P.; KUČERA, P.; ŠTOHL, R. *Programovatelné automaty* [online]. 1.10.2003 [cit. 16.4.2010]. Dostupné z: http://www.vaeprosyst.cz/Dokumentace/Programovateln%C3%A9_automaty/Programovateln%C3%A9_automaty-Skripta_FEKT_VUT_Brno.pdf

[8] SIEMENS s. r. o. *LOGO! Manuál* [online]. [cit. 13.4.2010] Dostupné z: www1.siemens.cz/ad/current/file.php?fh=6f545eaa89&aid=2024986

[9] SIEMENS s. r. o. *Prezentace logického modulu LOGO!* [online]. 14.8.2007 [cit. 14.4.2010]. Dostupné z: http://www1.siemens.cz/ad/current/index.php?msite=file_details&fnh=563833f24b

[10] Bilišňanský, M. *Schematické znacky logických obvodov* [online]. [cit. 15.4.2010] Dostupné z: <http://logicky.blogspot.com/2010/01/schematicke-znacky-logicky-obvodov.html>

[11] KOLÁŘ, S. *Vytvoření výukových podkladu pro práci ve vývojovém prostředí LOGO!Soft Comfort*. Brno, 2008. 53s., Bakalářská práce na Fakultě strojního inženýrství VUT v Brně. Vedúci bakalářskej práce Ing. Tomáš Marada Ph.D.

