

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ
УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

**І.Ш. Невлюдов, Л.О. Кривопляс-Володіна,
С.П. Новоселов, О.В. Сичова**

ПНЕВМАТИЧНІ ПРИСТРОЇ ТА ЗАСОБИ АВТОМАТИЗАЦІЇ МЕХАТРОННИХ СИСТЕМ

НАВЧАЛЬНИЙ ПОСІБНИК

Харків 2020

УДК 681.51
H40

*Рекомендовано Вченою радою
Харківського національного університету радіоелектроніки
(протокол № 4/8 від 29.05.2020 року)*

Невлюдов І.Ш. Пневматичні пристрої та засоби автоматизації мехатронних систем: Навчальний посібник / І.Ш. Невлюдов, Л.О. Кривопляс-Володіна, С.П. Новоселов, О.В. Сичова. – Харків: ФОП Панов А.М., 2020 . – 255 с.

ISBN 978-617-7859-58-0
DOI: 10.30837/978-617-7859-58-0

У навчальному посібнику подані відомості про сучасні технології організації виробництва, зокрема розглянуті особливості використання пневматичних систем, що разом з засобами автоматизації та ПЛК дозволяють створювати мехатронні роботизовані системи на виробництві. Розглянуті технології інтелектуальних приводів, датчиків і пневматичних пристроїв, а також системи управління за стандартами Індустрії 4.0. Дано практичні рекомендації щодо створення складних програмно-апаратних комплексів керування технологічними процесами (ТП).

Навчальний посібник дає можливість студентам ознайомитись із основами пневматики, навчитись виконувати розрахунки схем пневмоприводів, вивчити характеристики пневматичних приводів та оволодіти сучасними інструментами підготовки технологічних програм керування ТП, та перспективними мовами програмування промислових контролерів, такими як ST, LD, FBD, SFC.

Набуття теоретичних та практичних знань в галузі пневматики та мехатроніки, вивчення принципів програмування промислових контролерів, які застосовуються для функціонування АСУ ТП, є необхідним у процесі підготовки фахівців спеціальностей 151 «Автоматизація та комп'ютерно-інтегровані технології» та 172 «Телекомунікації та радіотехніка», практична діяльність яких орієнтована на виробництво, обслуговування, автоматизацію та створення комп'ютерно-інтегрованих технологій і промислової автоматизації та мехатроніки.

Навчальний посібник призначено для підготовки студентів денної та заочної форм навчання спеціальностей 151 «Автоматизація та комп'ютерно-інтегровані технології», 172 «Телекомунікації та радіотехніка». Може бути корисний аспірантам і фахівцям у промисловості, робота яких пов'язана з розробкою та організацією виробництв галузі радіоелектронного приладобудування.

УДК 681.51
H40

Рецензенти:

О.Ю. Панченко, д-р фізико-математичних наук, професор, завідувач кафедри проектування та експлуатації електронних апаратів, Харківський національний університет радіоелектроніки;

А.О. Андрусевич, д-р техн. наук, професор, директор Криворізького коледжу Національного авіаційного університету;

ISBN 978-617-7859-58-0
DOI: 10.30837/978-617-7859-58-0

© І.Ш. Невлюдов,
Л.О. Кривопляс-Володіна
С.П. Новоселов,
О.В. Сичова, 2020

ЗМІСТ

Перелік умовних познач, скорочень і термінів.....	5
Вступ	6
1 Пневматичні системи та основи пневматики.....	8
1.1 Основні параметри стисненого повітря. Одиниці вимірювання тиску.....	8
1.2 Виробництво та розподіл стисненого повітря	31
1.3 Блоки підготовки повітря.....	46
1.4 Пневмоциліндри.....	60
1.5 Пневморозподільники	68
1.6 Контрольні запитання та завдання.....	80
2 Характеристика пневматичних елементів, розробка пневматичних схем та електричних схем.....	81
2.1 Інтегровані рішення функціональних модулів в задачах синтезу ежекторів та пневматичних приводів	81
2.2 Реле, індикатори тиску, електропневмоперетворювачі в системі керування ежекторами	93
2.3 Контрольні запитання та завдання.....	115
3 Мехатронні пневматичні системи.....	116
3.1 Завдання механіки маніпуляторів.....	116
3.2 Кінематичний аналіз механізму маніпулятора	116
3.3 Динаміка маніпуляторів промислових роботів	124
3.4 Системи управління з програмованими логічними контролерами.....	130
3.5 Контрольні запитання та завдання.....	160
4 Технологія програмування ПЛК	161

4.1 Загальні елементи програмування мовами стандарту	
МЕК	161
4.2 ПЛК, як кінцевий автомат.....	203
4.3 Структурований текст (ST).....	206
4.4 Релейні діаграми (LD)	221
4.5 Функціональні діаграми FBD	230
4.6 Контрольні запитання та завдання.....	235
Перелік джерел посилань.....	236
ДОДАТОК А	238
ДОДАТОК Б	241

ПЕРЕЛІК УМОВНИХ ПОЗНАК, СКОРОЧЕНЬ І ТЕРМІНІВ

НМІ – Human Machine Interface;

АСУ – автоматизована система управління;

АСУТП – автоматизована система управління технологічними процесами;

АЦП – аналого-цифровий перетворювач;

ВП – виконавчий пристрій;

ГМС – газомодифіковане середовище;

ЕОМ – електронно-обчислювальна машина;

ЛВМ – локальна обчислювальна мережа;

ОЗП – оперативний запам'ятовувальний пристрій;

ООП – об'єктно-орієнтоване програмування;

ПГП – первинні генераторні перетворювачі;

ПЗ – програмне забезпечення;

ПЗП – постійний запам'ятовувальний пристрій;

ПЛК – програмований логічний контролер;

ПК – персональний комп'ютер;

ППП – первинні параметричні перетворювачі;

ЦП – центральний процесор;

ЦПП – центральний процесорний пристрій;

Командо-апарати – пристрої з перемикальними електричними контактами

Функціональний блок – програмний компонент, що відображає множину значень вхідних параметрів на множину вихідних.

ВСТУП

Зростання продуктивності праці, в тому числі шляхом її автоматизації, стає найважливішим джерелом розширення виробництва в сучасних умовах. Зазначені обставини висувають нові вимоги до масштабів використання пневматичних та мехатронних засобів автоматизації і технічному рівню АСУ ТП, до забезпечення їх надійності, точності, швидкодії, економічності, тобто до ефективності їх функціонування. Тому підготовка висококваліфікованих фахівців в галузі автоматизації технологічних процесів є актуальною задачею.

Таким чином набуття теоретичних та практичних знань в галузі пневматики та мехатроніки, вивчення принципів програмування промислових контролерів, які застосовуються для функціонування АСУ ТП, є необхідним у процесі підготовки фахівців спеціальностей 151 «Автоматизація та комп'ютерно-інтегровані технології» та 172 «Телекомунікації та радіотехніка», практична діяльність яких орієнтована на виробництво, обслуговування, автоматизацію та створення комп'ютерно-інтегрованих технологій і промислової автоматики та мехатроніки, що підтверджує актуальність написання даного навчального посібника.

Навчальний посібник є результатом співпраці Харківського національного університету радіоелектроніки, Національного університету харчових технологій та науково-навчального центру Samozzi в Україні в особі Володіна Сергія Олексійовича.

Навчальний посібник дає можливість студентам ознайомитись із основами пневматики, навчитись виконувати розрахунки схем пневмоприводів, вивчити характеристики пневматичних приводів та оволодіти сучасними інструментами підготовки технологічних програм керування ТП.

У навчальному посібнику подані відомості про сучасні технології організації виробництва, зокрема розглянуті особливості використання пневматичних систем, що разом з засобами автоматизації та ПЛК дозволяють створювати мехатронні роботизовані системи на виробництві. Розглянуті технології інтелектуальних приводів, датчиків і пневматичних пристроїв, а також системи управління за стандартами Індустрії 4.0.

Навчальний посібник складається з чотирьох розділів. Матеріал представлено в зручній та доступній для вивчення студентами формі з великою кількістю ілюстрацій та практичними прикладами.

Перший розділ навчального посібника присвячено висвітленню ролі пневматичних систем в автоматизації технологічних процесів на сучасному виробництві. Розкрито питання основ пневматики та розрахунку схему пневмоприводів.

В другому розділі розглядаються характеристики пневматичних елементів, показані приклади розробки пневматичних схем для різних задач використання. Надається інформація про взаємодію матеріальних і енергетичних потоків в пневматичних схемах.

В третьому розділі розглядаються мехатронні пневматичні системи. Приводяться відомості про кінематичний аналіз механізму маніпулятора та висвітлюються питання використання систем управління з програмованими логічними контролерами.

Четвертий розділ присвячено технології програмування ПЛК. Розглянуті основні технологічні мови програмування, які відповідають МЕК-61131 – ST, LD, FBD, SFC.

Зміст навчального посібника «Пневматичні пристрої та засоби автоматизації мехатронних систем» відповідає програмі підготовки бакалаврів, що навчаються за спеціальностями 151 «Автоматизація та комп'ютерно-інтегровані технології» і 172 «Телекомунікації та радіотехніка».

1 ПНЕВМАТИЧНІ СИСТЕМИ ТА ОСНОВИ ПНЕВМАТИКИ

1.1 Основні параметри стисненого повітря. Одиниці вимірювання тиску

У міжнародній системі одиниць СІ одиниці тиску присвоєно ім'я французького вченого Паскаля, автора відомого закону про рівномірне поширення статичного тиску в усіх напрямках.

Один «Паскаль» дорівнює тиску, який створюється силою в один «Ньютон», рівномірно розподіленою на площі в 1 м^2 (лат. $1 \text{ Pa} = 1 \text{ N} / \text{m}^2$). Це дуже малий тиск (уявіть собі, що на бутерброд площею 1 м^2 ви витратили всього 100 г масла).

Оскільки ця одиниця незручна, то для вимірювання стандартних рівнів тиску, які використовуються в техніці, введена спеціальна одиниця – бар, яка дорівнює $1/10$ мегапаскалей (лат. bar. $1 \text{ бар} = 0,1 \text{ МПа} = 10^5 \text{ Па}$).

Зручність цієї одиниці полягає в тому, що вона практично дорівнює традиційно прийнятій технічній атмосфері ($1 \text{ ат.} = 1 \text{ кгс/см}^2$). Одиниці вимірювання тиску зведені до таблиці 1.1.

Наведемо наступні співвідношення:

$$1 \text{ ат.} = 0,98067 \text{ бар}$$

$$1 \text{ бар} = 1,01972 \text{ ат}$$

$$1 \text{ бар} = 14,5 \text{ psi (фунт-сила/кв. дюйм)}$$

$$1 \text{ psi} = 0,06895 \text{ бар}$$

Для вимірювання невеликих рівнів тиску використовуються також міліметри водяного та ртутного стовпа.

Справедливі такі співвідношення:

$$1 \text{ ат.} = 10000 \text{ мм. вод. ст.} = 736 \text{ мм. рт. ст.}$$

$$1 \text{ мм. вод. ст.} = 9,80665 \text{ Па}$$

$$1 \text{ мм. рт. ст.} = 1,333224 \text{ Па}$$

Атмосфера Землі на поверхні має тиск в одну фізичну атмосферу:

$$1 \text{ атм.} = 760 \text{ мм. рт. ст.} = 1,01325 \text{ бар (нормальний тиск)}$$

$$1 \text{ бар} = 100\,000 \text{ Па}$$

Таблиця 1.1 – Одиниці вимірювання тиску

	Паскаль (Pa, Па)	Бар (bar, бар)	Технічна атмосфера (at, ат)	Фізична атмосфера (atm, атм)	Міліметр ртутного стовпчика (mm Hg, torr, мм рт.ст., торр)	Метр водяного стовпчика (m H ₂ O, м вод. ст.)	Фунт-сила на кв. дюйм (psi)
1 Па	1 Н/м ²	10 ⁻⁵	10,197·10 ⁻⁶	9,8692·10 ⁻⁶	7,5006·10 ⁻³	1,0197·10 ⁻⁴	145,04·10 ⁻⁶
1 бар	105	1·10 ⁶ дин/см ²	1,0197	0,98692	750,06	10,197	14,504
1 ат	98066,5	0,980665	1 кгс/см ²	0,96784	735,56	10	14,223
1 атм	101325	1,01325	1,033	1 атм	760	10,33	14,696
1 мм рт. ст.	133,322	1,3332·10 ⁻³	1,3595·10 ⁻³	1,3158·10 ⁻³	1 мм рт.ст.	13,595·10 ⁻³	19,337·10 ⁻³
1 м вод.ст.	9806,65	9,80665·10 ⁻²	0,1	0,096784	73,556	1 м вод. ст.	1,4223
1 psi	6894,76	68,948·10 ⁻³	70,307·10 ⁻³	68,046·10 ⁻³	51,715	0,70307	1 lbf/in ²

Тиск газу пропорційний його абсолютній температурі T і концентрації молекул n , яку можна визначити як відношення:

$$n = \frac{N}{V},$$

де N – число молекул, які знаходяться в посудині;

V – об'єм посудини.

Тиск P газу дорівнює:

$$P = n \cdot k \cdot T$$

та вимірюється у $\left[1 \text{ Па} = \frac{1}{\text{м}^3} \cdot \frac{\text{Дж}}{\text{К}} \cdot \text{К} \right]$.

Коефіцієнт пропорційності k – це постійна Больцмана, яка дорівнює:

$$k = 1,380662 \cdot 10^{-23} \frac{\text{Дж}}{\text{К}}.$$

Зазвичай відомий об'єм V робочої камери та маса m замкненого в ній повітря. Припустимо, що повітря є ідеальним газом, тиск P всередині робочої камери може бути визначено за формулою Менделєєва-Клапейрона:

$$P = m \frac{R \cdot T}{V},$$

де R – універсальна газова стала ($R = 287 \frac{\text{Дж}}{\text{кг} \cdot \text{К}}$ для повітря), константа, дорівнює роботі розширення 1 моля ідеального газу при постійному тиску і збільшенні температури на 1 К;

T – температура в градусах Кельвіна (абсолютна температура), °К.

Нормальна температура у фізиці $0 \text{ }^\circ\text{C} = 273,15 \text{ К}$.

Розрізняють:

– абсолютний тиск ($P_{\text{абс}}$) – це тиск, для вимірювання якого за початок відліку беруть нульове значення (теоретично – від абсолютного вакууму);

– надлишковий (манометричний) P_n – різниця між абсолютним і тиском навколишнього середовища, найчастіше – барометричним (атмосферним) тиском. Вимірюють манометром;

– розрідження (вакууметричний) $P_{\text{вак}}$ – тиск в системі, менший за атмосферний. Вимірюють вакууметрами.

– атмосферний тиск P_a – це тиск, викликаний атмосферним повітряним шаром, покриває поверхню землі (товщина атмосферного шару близько 33 км).

На рівні моря, в широті 45° , за температури $0 \text{ }^\circ\text{C}$ і вологості 65%, атмосферний тиск піднімається на 760 мм ртутного стовпчика або на 10,33 м водяного, що дорівнює 1 атмосфері (атм./atm).

На рис. 1.1 можна побачити співвідношення між абсолютним, надлишковим і вакуумним тисками.

Під надлишковим тиском розуміють різницю між абсолютним (повним) і місцевим атмосферним тиском.

Надлишковий тиск за загально прийнятим припущеннями є завжди позитивним.

«Негативний» відносний тиск визначається як вакуумний. Вакуумним називається тиск, який менше місцевого атмосферного тиску.

Місцевий атмосферний тиск може змінюватися в залежності від навколишньої температури, висоти над рівнем моря і місцевих погодних умов.

Існує також диференціальний тиск – це різниця між двома невідомими тисками. Цей тип вимірювання тиску зазвичай використовується, щоб отримати значення падіння тиску в системі при підведенні поточного середовища. При вимірюванні різниці тисків не важливо, яке з них вважати опорним.



$$\text{Абсолютний тиск} = \text{атмосферний тиск} + \text{надлишковий тиск}$$

Рисунок 1.1 – Співвідношення між абсолютним, надлишковим і вакуумним тисками

Якщо концентрація молекул дорівнює нулю, то абсолютний тиск в такій робочій камері також дорівнює нулю. Можна вважати, що на поверхні Землі P_a має деяку потенційну енергією, внаслідок того, що усе оточуюче повітря знаходиться під атмосферним тиском P_a і може виконувати роботу. Так працює більшість вакуумних пристроїв, наприклад вакуумні приводи, вакуумні присоски тощо. Припустимо, що ці пристрої працюють на розрідження. Робоча камера з повітрям буде під дією потенційної енергії, якщо тиск газу всередині неї буде більше атмосферного (тобто $P_i > 0$). Газ також може виконувати роботу, але вже при виході з робочої камери в атмосферу, тобто привести в дію пристрої, які працюють на нагнітання. Оскільки більшість пристроїв промислової пневмоавтоматики працює на P_n , а магістральний тиск істотно більше атмосферного, при розрахунку зусиль зручно використовувати надлишковий тиск. У термодинамічних розрахунках, наприклад у формулі Менделєєва-Клапейрона – абсолютний тиск.

Високий тиск – це тиск, що знаходиться в межах 3...10 бар. Використовується для живлення більшості промислових пристроїв.

Низький тиск використовується в датчиках різного виду та в деяких керуючих пристроях.

Низький тиск поділяється на три діапазони:

- тиск 0,5 бар та менше;
- тиск від 0,5 бар до 1 бара;
- тиск від 1 до 3 бар.

Витрати є одним з найважливіших параметрів, які характеризують роботу будь-якого пневматичного пристрою і визначаються як кількість речовини, яка проходить через даний переріз за одиницю часу. Оскільки можна вимірювати як об'єм речовини, так і його масу, існує поняття об'ємної (відносної, абсолютної) витрати Q і масової витрати G_m .

Витрати Q – це об'єм речовини, який проходить крізь поперечний переріз потоку за одиницю часу:

$$Q = \frac{V}{T},$$

де V – об'єм, м³;

t – час, с.

$$1 \text{ м}^3/\text{с} = 60000 \text{ л/хв} = 3600 \text{ м}^3/\text{год}.$$

Відносні витрати Q_r вимірюються в л/хв за манометричним надлишковим тиском.

Приклад: 6 л/хв за 6 бар.

Абсолютні витрати Q_a приведені за атмосферним тиском. Виражаються в літрах за хвилину A.N.R. або норма літрів за хвилину.

A – Atmosphere (Атмосферне)

N – Normal (Нормальне)

R – Reference (Співвідношення)

$Q_a = Q_r (P_n + 1,013)$ позначення A.N.R.

P_n – надлишковий тиск.

Приклад: 6 л/хв за 6 бар складаються з: $6 \cdot (6 + 1,013) \approx 42$ л/хв (A.N.R.) або 42 Нл/хв (читається як – нормальні літри за хвилину).

Масові витрати у кг/с:

$$G_m = \frac{m}{t}.$$

Співвідношення між об'ємними і масовими витратами:

$$G_m = Q \cdot \rho = S \cdot W \cdot \rho,$$

де Q – об'ємні витрати;

ρ – густина повітря;

W – швидкість потоку в перерізі;

S – площа поперечного перерізу потоку.

Коефіцієнт витрат C_V – вимірюється апаратом, який чисельно фіксує обсяг повітря в літрах/хвилину крізь його переріз, коли втрата тиску між входом і виходом становить 1 бар.

Кожен пневматичний елемент характеризується коефіцієнтом пропускної здатності C_V , який вказується в довідниках. Коефіцієнт пропускної здатності введений для полегшення роботи проектувальників пневматичних і

гідравлічних систем. За його допомогою можна без зусиль знайти витрати крізь клапан, або підібрати відповідний елемент для забезпечення витрат.

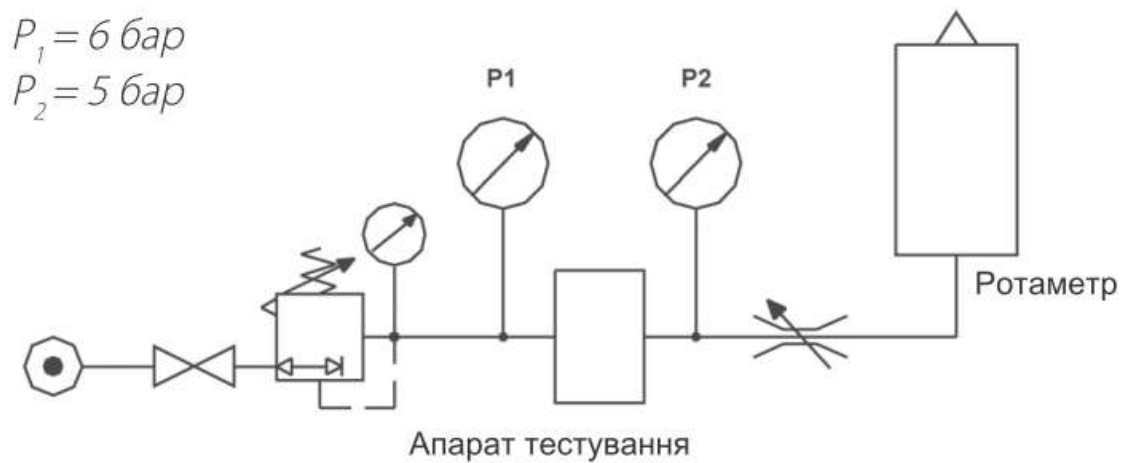


Рисунок 1.2 – Апарат для вимірювання коефіцієнту витрат C_V

Наведені нижче розрахункові формули мають рекомендаційний характер. Отримані результати повинні розглядатися тільки як одні з можливих.

Якщо $P_2 + 1 > 0,5 (P_1 + 1)$ то витрати для газового середовища розраховуються за формулою:

$$Q = 411 \frac{C_V \sqrt{(P_1 - P_2)(P_1 + 1)}}{\sqrt{S_r}},$$

де P_1 – тиск на вході, бар;

P_2 – тиск на виході, бар;

S_r – відносна густина газу (відносно повітря);

C_V – коефіцієнт пропускної здатності.

Якщо $P_2 + 1 < 0,5 (P_1 + 1)$ тоді витрати для газового середовища розраховуються таким чином:

$$Q = 411 \frac{C_V \sqrt{0,5(P_1 + 1)}}{\sqrt{S_r}}.$$

Якщо $P_2 + 1 > 0,5 \cdot (P_1 + 1)$, то мінімальний коефіцієнт C_V дорівнюватиме:

$$C_V = \frac{Q\sqrt{S_r}}{411\sqrt{(P_1 - P_2) \cdot (P_1 + 1)}},$$

Якщо $P_2 + 1 > 0,5 \cdot (P_1 + 1)$ тоді:

$$C_V = \frac{Q\sqrt{S_r}}{411\sqrt{0,5 \cdot (P_1 + 1)}}.$$

Густина – відношення робочої маси тіла до об'єму, який заповнено ними:

$$\rho = \frac{m}{V}, \text{ кг/м}^3.$$

Густина повітря залежить від температури і тиску. При нормальному (атмосферному) тиску ($P = 101325 \text{ Па}$):

$$\rho_{\text{повім}} = 1,29 \text{ кг/м}^3 \text{ за } T = 273,15 \text{ К } (t = 0 \text{ }^\circ\text{C});$$

$$\rho_{\text{повім}} = 1,20 \text{ кг/м}^3 \text{ за } T = 293,15 \text{ К } (t = 20 \text{ }^\circ\text{C}).$$

Замінімо відношення маси до об'єму на густину у формулі та отримаємо рівняння, що зв'язує густину із тиском і температурою газу:

$$\rho = \frac{P}{R \cdot T}, \frac{H \cdot \text{кг} \cdot \text{К}}{\text{м}^2 \cdot \text{Дж} \cdot \text{К}}.$$

Величина, обернена до густини називається питомим об'ємом – v :

$$v = \frac{1}{\rho}, \frac{\text{м}^3}{\text{кг}}.$$

Можна використовувати і питому вагу γ , під якою розуміють відношення ваги тіла G до об'єму V , що займає тіло:

$$\gamma = \frac{G}{V}, \frac{H}{\text{м}^3}, \quad \gamma = \frac{mg}{V} = \rho \cdot g.$$

Вплив температури на густину повітря наведені в табл. 1.2.

Таблиця 1.2 – Вплив температури на густину повітря на рівні моря

Температура, T, °C	+35	+30	+25	+20	+15	+10	+5	±0	-5	-10	-15	-20	-25
Густина повітря (з рівняння Клапейрона), ρ , кг/м ³	1,146	1,164	1,184	1,204	1,225	1,247	1,269	1,292	1,317	1,341	1,367	1,394	1,422

Залежність тиску, температури та щільності повітря від висоти в порівнянні зі стандартною атмосферою ($P_a = 101325$ Па; $T_0 = 288,15$ К; $\rho = 1,225$ кг/м³) наведено на рис. 1.3.

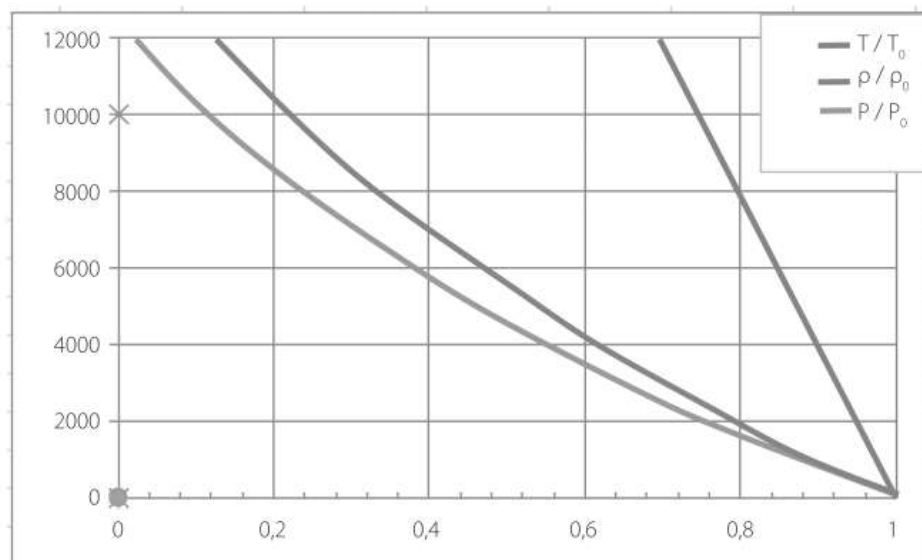


Рисунок 1.3 – Залежність тиску, температури та щільності повітря від висоти в порівнянні зі стандартною атмосферою

В'язкість газу характеризує сили внутрішнього тертя при русі потоку повітря і оцінюється коефіцієнтами динамічної і кінематичної в'язкості.

Динамічна в'язкість – характеризує внутрішнє тертя в рідині або газі.

Якщо переміщувати пластинку площею S у в'язкій рідині, то на неї буде діяти сила внутрішнього тертя (рис. 1.4):

$$F_{\text{тер}} = \mu_{\text{д}} \cdot S \frac{dW}{dL}.$$

де $F_{\text{тер}}$ – сила тертя пластинки та шару рідини (газу);

S – площа пластинки, м^2 ;

$\frac{dW}{dL}$ – градієнт швидкості, с^{-1} ;

$\mu_{\text{д}}$ – коефіцієнт динамічної в'язкості [$\text{Па} \cdot \text{с}$].

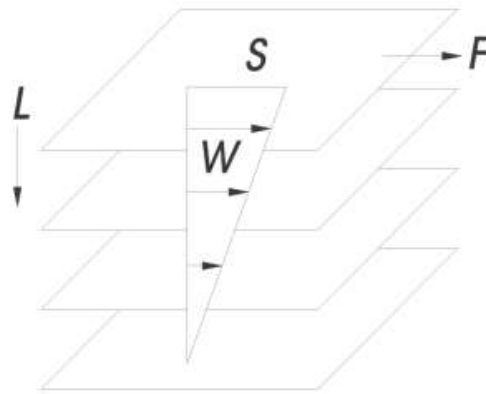


Рисунок 1.4 – Дія сили внутрішнього тертя на пластину у в'язкій рідині

Динамічна в'язкість води при температурі $20\text{ }^{\circ}\text{C}$ дорівнює $1\text{ МПа} \cdot \text{с}$, в'язкість повітря дорівнює $0,0182\text{ МПа} \cdot \text{с}$.

В'язкість рідин різко падає при підвищенні температури, а в'язкість повітря (газів) зростає.

Кінематична в'язкість рідини або газу називається співвідношенням динамічної в'язкості до густини:

$$\nu = \frac{\mu_{\text{д}}}{\rho}, \frac{\text{м}^2}{\text{с}}.$$

В якості одиниці кінематичної в'язкості застосовується також стокс (ст) і сантистокс (сст):

$$1 \text{ ст} = 10^{-4} \text{ м}^2/\text{с},$$

$$1 \text{ сст} = 10^{-6} \text{ м}^2/\text{с} = 1 \text{ мм}^2/\text{с}.$$

Під час проведення інженерних розрахунків зручніше використовувати наближені формули для визначення фізичних властивостей повітря.

Динамічна та кінематична в'язкість повітря в залежності від температури повітря наведена у табл. 1.3.

Таблиця 1.3 – Динамічна та кінематична в'язкість повітря за нормальним атмосферним тиском та різної температури повітря

Температура повітря, °С	Динамічна в'язкість повітря (Н·с/м ²) x 10 ⁻⁵	Кінематична в'язкість повітря (м ² /с) x 10 ⁻⁵
-20	1,63	1,17
0	1,71	1,32
5	1,73	1,36
10	1,76	1,41
15	1,80	1,47
20	1,82	1,51
25	1,85	1,56
30	1,86	1,60
40	1,87	1,66
50	1,95	1,76
60	1,97	1,86
70	2,03	1,97
80	2,07	2,07
90	2,14	2,20
100	2,17	2,29

Густина повітря, кг/м³:

$$\rho_{air}(T) = \frac{P_{air}}{287,4 \cdot T}.$$

Теплоємність повітря, Дж/кг·К:

$$c_{p,air}(T) = [1,0005 + 1,1904 \cdot 10^{-4} \cdot (T - 273)] \cdot 10^3$$

Теплопровідність повітря, Вт/(м·К):

$$\lambda_{air}(T) = 2,44 \cdot 10^{-2} \left(\frac{T}{273} \right)^{0,82}.$$

Коефіцієнт динамічної в'язкості повітря, Па·с:

$$\mu_{air}(T) = 1,717 \cdot 10^{-5} \left(\frac{T}{273} \right)^{0,683}.$$

Коефіцієнт кінематичної в'язкості повітря, м²/с:

$$\nu_{air}(T) = \frac{\mu_{air}(T)}{\rho_{air}(T)}.$$

Коефіцієнт температуропровідності, м²/с:

$$a_{air}(T) = \frac{\lambda_{air}(T)}{c_{P_{air}}(T) \cdot \rho(T) \cdot 1000}.$$

Навколишнє атмосферне повітря – це суміш деяких газів, водяної пари, твердих і рідких домішок. Кількість складових може змінюватися в залежності від умов оточуючого середовища.

Таблиця 1.4 – Основні складові сухого повітря (в робочому об'ємі)

Азот	78 %
Кисень	21 %
Інертний газ	0,94 % Аргон - Неон - Гелій
Вуглекислий газ	0,06 %

Характеристики пневматичних установок, які пов'язані із фізичними властивостями повітря, а саме:

- в'язкість (внутрішнє тертя) – одне із явищ переносу, властивість плинних тіл (рідин і газів) чинити опір переміщенню однієї їх частини щодо іншої. Може вимірюватись в сантистоксах (cst) або в мм²/с;

- стисливість – це одночасна зміна об'єму і тиску.

Стисливість – властивість робочого тіла змінювати свій об'єм під дією прикладеного тиску.

Коефіцієнт об'ємного стиснення β дорівнює, 1/Па:

$$\beta = -\frac{dV}{VdP} = \frac{d\rho}{\rho dP}.$$

Незважаючи на те, що молекули рідини рухомі, помітно стиснути рідину можна тільки приклавши до неї дуже великий тиск. Для практичних цілей можна вважати рідини нестисливими.

У таблиці 1.5 наведено основні характеристики в'язкості та стисливості при нормальних умовах.

Таблиця 1.5 – Основні характеристики в'язкості та стисливості при нормальних умовах

В'язкість	Стисливість
15,5 cst при 20 °С невелике значення	Споживання енергії тільки якщо є витрати
Швидкість дії завдяки високій швидкості переміщення	Контроль швидкості переміщення наблизений
стисненого повітря в трубопроводі від 50 до 100 м/с	Нерівномірність переміщення
Простота конструкції мереж	Складнощі використання при високому тиску
Централізоване виробництво	Продуктивність дуже низька
Велика розподільча мережа	Висока собівартість при високому тиску

Стисливість газів набагато вища і залежить від тиску, температури і об'єму які вони займають.

Бойлом в 1662 р. в Англії і незалежно від нього в 1676 р. Маріоттом у Франції було встановлено, що якщо газ займав деякий початковий обсяг V_0 і мав тиск P_0 , то після стиснення його до об'єму V_1 його тиск, за умови, що температура газу не змінюється (ізотермічний процес) підвищиться до величини P_1 . Добуток початкового об'єму і тиску буде дорівнювати добутку кінцевого об'єму і тиску:

$$\text{при } T = \text{const}, P_0 \cdot V_0 = P_1 \cdot V_1 \text{ або } \frac{P_0}{P_1} = \frac{V_1}{V_0}.$$

Важливим наслідком цього закону є те, що оскільки маса газу не змінюється, при підстановці значень об'ємів отримаємо, що відношення тисків дорівнює відношенню густини газу:

$$\frac{P_0}{P_1} = \frac{\rho_0}{\rho_1}.$$

Зміна об'єму може відбуватися також і при зміні температури T тіла, що враховується коефіцієнтом об'ємного розширення, 1/град.:

$$\alpha = \frac{dV}{Vdt}.$$

Більшість твердих тіл, рідин і всі гази збільшуються в об'ємі з нагріванням. Досліди Гей-Люссака в 1802 році показали, що коефіцієнт об'ємного розширення всіх газів однаковий, постійний і дорівнює:

$$\alpha = \frac{1}{273} = 0,366 \text{ (1/град).}$$

Шарль встановив, що якщо нагрівати деяку масу газу у закритій посудині, то на кожен градус підвищення температури, тиск P в посудині збільшується на $1/273$ частину тиску P_0 при 0°C . Застосовуємо рівняння:

$$P = P_0(1 + \gamma \cdot t),$$

де P_0 – тиск газу при $0\text{ }^\circ\text{C}$;

γ – термічний коефіцієнт тиску, дорівнює $1/273 = 0,00366$ (1/град);

t – температура газу в градусах Цельсія.

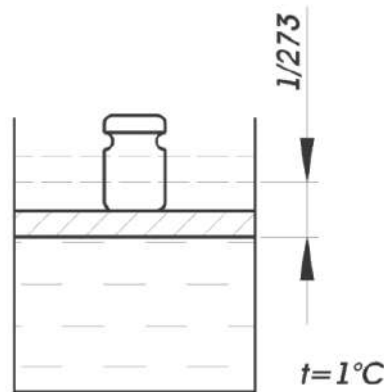


Рисунок 1.5 – Досліди Гей-Люссака

Рівність коефіцієнтів α і γ не випадкова, а обумовлена рівністю відношення тисків та об'ємів за законом Бойля-Маріотта.

Шкала Цельсія менш зручна для вимірювання температури газу, щоб забезпечити пряму пропорційність тиску і температури. Із закону Шарля можна отримати число $T = 273 + t\text{ }^\circ\text{C}$, яке можна розглядати як температуру, обраховану за новою, так званою абсолютною шкалою Кельвіна. Ціна градуса залишається сталою, але за нуль приймають точку, яка лежить на 273 градуси нижче точки танення льоду.

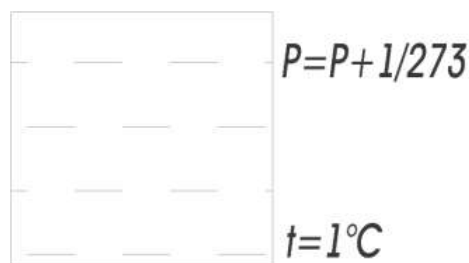


Рисунок 1.6 – Досліди Шарля

Рівняння стану ідеального газу описує всі три найважливіших параметри газу. Сенс його полягає в наступному: для даної кількості ідеального газу відношення добутку тиску на об'єм по відношенню щодо абсолютної температури – є величина постійна:

$$\frac{P \cdot V}{T} = \text{const}.$$

Можливі окремі випадки рівняння стану ідеального газу, коли один з параметрів залишається постійним (див. табл. 1.6).

У разі, якщо теплообмін із середовищем відсутній (наприклад, при хорошій теплоізоляції або при малому часі протікання) – процес називають адіабатичним. Стан параметрів газу при цьому виражається залежністю:

$$P \cdot V^n = \text{const},$$

де n – показник адіабати, що дорівнює відношенню $n = \frac{C_P}{C_V}$;

C_P – теплоємність газу за постійним тиском, Дж / кг·К;

C_V – теплоємність газу за постійним об'ємом, Дж / кг·К.

Таблиця 1.6 – Рівняння ідеального газу

Процес	Ізобаричний	Ізохоричний	Ізотермічний
Ознака	$P = \text{const}$	$V = \text{const}$	$T = \text{const}$
Запис	$\frac{V_1}{V_2} = \frac{T_1}{T_2}$	$\frac{P_1}{P_2} = \frac{T_1}{T_2}$	$\frac{P_1}{P_2} = \frac{V_2}{V_1}$
Закон	Гей-Люссака 1802 р.	Шарля 1787 р.	Бойля-Маріотта 1662 р.

Теплоємність речовини визначає кількість тепла, яку потрібно передати одному кілограму речовини, щоб підвищити його температуру на один градус.

При нагріванні газу за $P = \text{const}$ здійснюється зовнішня робота з підйому вантажу на деяку висоту, тому необхідно витратити додаткову кількість енергії. Тому $C_P > C_V$.

Для повітря показник адіабати n дорівнює:

$$n = \frac{C_P}{C_V} = 1,4.$$

Зважаючи на велику вільність руху молекул, газові закони, які описують зміну їх об'єму, універсальніші і точніші, ніж аналогічні співвідношення для твердих тіл і рідин.

Ефект падіння статичного тиску в струмені широко використовується у вакуумних перетворювачах, наприклад, для живлення вакуумних присосок (рис. 1.7).

$$\rho z_1 + \frac{P_1}{\rho} + \frac{v_1^2}{2} = \rho z_2 + \frac{P_2}{\rho} + \frac{v_2^2}{2} = \rho z_3 + \frac{P_3}{\rho} + \frac{v_3^2}{2},$$

де ρz – питома енергія місця положення ($\rho = 9,8 \text{ м/с}^2$ – прискорення вільного падіння);

P / ρ – питома енергія тиску;

$v^2 / 2$ – питома кінетична енергія.

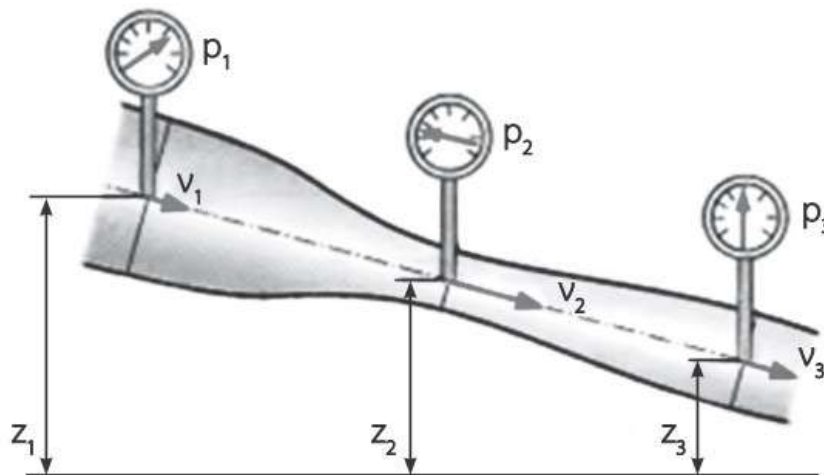


Рисунок 1.7 – Ефект падіння статичного тиску в струмені у вакуумному перетворювачі

Витрата стисненого повітря при видуванні його крізь форсунки в атмосферу наведено у таблиці 1.7.

Таблиця 1.7 – Витрата стисненого повітря при видуванні його крізь форсунки в атмосферу

Тиск (бар)	Діаметр форсунки (мм)									
	0,5	1	1,5	2	2,5	3	3,5	4	4,5	5
	Витрати стисненого повітря (л/хв)									
1	5	19	42	73	117	169	229	300	378	466
2	7	28	63	109	175	253	344	450	568	700
3	9	37	84	146	234	338	458	600	757	933
4	12	47	106	182	292	426	573	750	946	1166
5	14	56	127	218	350	507	687	900	1135	1400
6	16	65	148	255	409	592	802	1050	1324	1632
7	18	74	169	291	467	676	916	1200	1514	1866
8	21	84	190	328	526	761	1030	1350	1703	2099

Таблиця 1.7 (продовження)

Тиск (бар)	Діаметр форсунки (мм)									
	5,5	6	6,5	7	7,5	8	8,5	9	9,5	10
	Витрати стисненого повітря (л/хв)									
1	564	674	790	916	1052	1195	1350	1514	1685	1868
2	846	1010	1185	1374	1578	1792	2025	2271	2528	2803
3	1128	1347	1580	1832	2104	2390	2700	3028	3370	3737
4	1410	1684	1975	2290	2630	2987	3375	3785	4213	4671
5	1692	2021	2370	2748	3156	3584	4050	4542	5055	5605
6	1974	2358	2765	3206	3682	4182	4725	5300	5898	6539
7	2256	2694	3160	3664	4208	4779	5400	6056	6740	7474
8	2538	3031	3555	4122	4734	5377	6075	6813	7583	8408

Атмосферні забруднення чинять негативний вплив на навколишнє середовище. В середньому, випадає 8 тон пилу на 1км² у місяць. 80% цього пилу складається із найдрібніших частинок до 2 мікрон, тобто 112 мільйонів найдрібніших частинок до 2 мікрон на 1 м³ повітря.

Газоподібні вуглеводні, що виходять із труб заводів, вихлопи двигунів внутрішнього загоряння, досягають 0,004 грама на літр повітря.

Компресори також вносять свої забруднення:

- великі частинки від зношування, іржа;
- окислене мастило (яке формується у тверді частинки в повітрі).

У процесі роботи формується суміш у вигляді аерозолі, яку важко розділити (рис. 1.8).

РОЗМІРИ ЧАСТИНОК в мікронах					
0,001	0,01	0,3	0,5	10	100
Газоподібні молекули	Масляні аерозолі			Масляний туман	
	Сigaretний дим			Водяний туман	
Смог і пил					

Рисунок 1.8 – Розміри частинок у аерозолях

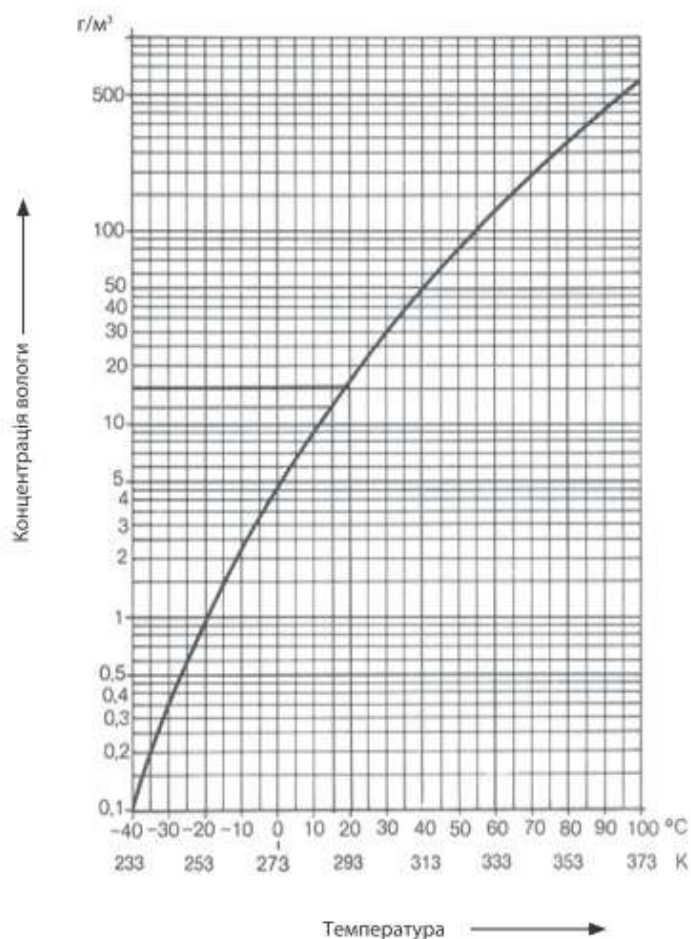


Рисунок 1.9 – Залежність концентрації вологи у повітрі від температури

Компресори підсилюють джерело атмосферного забруднення у процесі створення компресії. Стиснення повітря до 7 бар підвищує концентрацію до $112 \times 8 = 896$ мільйонів дрібних частинок розміром до 2 мікрон на 1 м^3 стисненого повітря, незважаючи на наявність фільтрації під час роботи компресора.

Коли повітря містить максимальну кількість водяної пари, яке відповідає гігromетричному ступеню 1, досягається 100% насичення. Це називається точка роси. Якщо певний об'єм повітря доповнити деякою кількістю водяної пари, то він негайно почне згущуватися.

Приклад.

У точці конденсації при 0 °С, 1 м³ повітря містить 4,868 г води.

Для того щоб отримати один кубічний метр стисненого повітря за умови тиску 10 бар, компресор повинен стиснути 11 кубічних метрів атмосферного повітря. Разом із цим повітрям він також всмоктує, подібно до пилососа, домішки, які містяться в повітрі: пил, вологу, пари мастила і хімікатів, а також природну вологість повітря.

Таблиця 1.8 – Вміст води у повітрі в залежності від температури

Точка конденсації, °С	Вміст води, г/м ³	Точка конденсації, °С	Вміст води, г/м ³	Точка конденсації, °С	Вміст води, г/м ³	Точка конденсації, °С	Вміст води, г/м ³	Точка конденсації, °С	Вміст води, г/м ³	Точка конденсації, °С	Вміст води, г/м ³	Точка конденсації, °С	Вміст води, г/м ³	Точка конденсації, °С	Вміст води, г/м ³
100	588,21	79	279,278	58	118,199	37	43,508	16	13,531	-4	3,513	-25	0,550	-46	0,060
99	569,07	78	268,806	57	113,130	36	41,322	15	12,739	-5	3,238	-26	0,510	-47	0,054
98	550,38	77	258,827	56	108,200	35	39,286	14	11,987	-6	2,984	-27	0,460	-48	0,048
97	532,13	76	248,840	55	103,453	34	37,229	13	11,276	-7	2,751	-28	0,410	-49	0,043
96	514,40	75	239,351	54	98,883	33	35,317	12	10,600	-8	2,537	-29	0,370	-50	0,038
95	497,21	74	230,142	53	94,483	32	33,490	11	9,961	-9	2,339	-30	0,330	-51	0,034
94	480,39	73	221,212	52	90,247	31	31,744	10	9,356	-10	2,156	-31	0,301	-52	0,030
93	464,12	72	212,648	51	86,173	30	30,078	9	8,784	-11	1,960	-32	0,271	-53	0,027
92	448,31	71	204,286	50	82,257	29	28,488	8	8,243	-12	1,800	-33	0,244	-54	0,024
91	432,89	70	196,213	49	78,491	28	26,970	7	7,732	-13	1,650	-34	0,220	-55	0,021
90	417,94	69	188,429	48	74,871	27	25,524	6	7,246	-14	1,510	-35	0,198	-56	0,019
89	403,38	68	180,855	47	71,395	26	24,143	5	6,790	-15	1,380	-36	0,178	-57	0,017
88	389,225	67	173,575	46	68,056	25	22,830	4	6,359	-16	1,270	-37	0,160	-58	0,015
87	375,47	66	166,507	45	64,848	24	21,578	3	5,953	-17	1,150	-38	0,144	-59	0,013
86	362,12	65	159,654	44	61,772	23	20,386	2	5,570	-18	1,050	-39	0,130	-60	0,011
85	340,19	64	153,103	43	58,820	22	19,252	1	5,209	-19	0,960	-40	0,117	-65	0,0064
84	336,66	63	146,771	42	55,989	21	18,191	0	4,868	-20	0,880	-41	0,104	-70	0,0033
83	324,47	62	140,659	41	53,274	20	17,148	-	-	-21	0,800	-42	0,093	-75	0,0013
82	311,62	61	134,684	40	50,672	19	16,172	-1	4,487	-22	0,730	-43	0,083	-80	0,0006
81	301,19	60	129,020	39	48,181	18	15,246	-2	4,135	-23	0,660	-44	0,075	-85	0,0003
80	290,02	59	123,495	38	45,593	17	14,367	-3	3,889	-24	0,600	-45	0,067	-90	0,0001

Незважаючи на високоякісні фільтри на всмоктуванні, всі компоненти, що містяться у робочому повітрі, потрапляють у стиснене повітря. Домішки, які розподілені в 11 кубічних метрах до стиснення, концентруються в одному кубічному метрі стисненого повітря, тобто їх концентрація збільшується в 11 разів.

Давайте розглянемо поняття «Домішки» і «Вологість» окремо.

Уявімо, атмосферне повітря у вигляді вологої губки. У спокійному стані вона може вміщувати деяку кількість води, без її просочування.

Якщо стиснути цю губку, частина води вилетіть з неї – але тільки частина. Навіть якщо вичавити її, не вся вода вийде з неї. Отже, частина вологи видаляється у вигляді конденсату, що виділився в процесі стиснення.

У воді домішки розчиняються, і в результаті цього утворюється агресивна суміш, яка може викликати корозію в компресорі та трубопроводах.

Агресивна суміш конденсату, оксидів та продуктів корозії переноситься по трубопроводах до машин, які споживають стиснене повітря. Очевидно, що ці машини вийдуть із ладу в результаті зношування, значно швидше тих, які приводяться в дію чистим стисненим повітрям. Це було доведено досвідом експлуатації.

Давайте проаналізуємо цей процес із фізичної точки зору.

Вологість – це наявність вологи (в загальному випадку) в сухому повітрі.

Тиск вологого повітря представляє суму P_A (тиск повітря) і P_M (парціальний тиск вологи).

Сухе повітря може поглинати вологу тільки до досягнення ним точки роси P_D .

Якщо парціальний тиск вологи перевищить тиск у точці роси ($P_M > P_D$), то зайва волога сконденсується у вигляді туману.

Здатність сухого повітря поглинати вологу змінюється в залежності від температури, але не залежить від тиску. В результаті, відносна вологість повітря:

$$\varphi_{\text{відн}} = \frac{P_M}{P_D}.$$

Наприклад, P_D за 3°C становить 0,007576 бар, але за 20°C , він становить 0,02337 бар. За відносною вологістю повітря 70% и 20°C , P_M становить 0,01636 бар. (Див. таблицю «Вміст води у повітрі в залежності від температури», в якій парціальний тиск води вже переведено у масові частки $\text{г}/\text{м}^3$).

З причини того, що здатність повітря поглинати вологу не залежить від тиску, ці співвідношення не зміняться для стисненого повітря.

Приклад.

Об'єм 8 м^3 вологого повітря при тиску 1 бар дорівнює 1 м^3 вологого повітря при тиску 8 бар абс., але парціальний тиск конденсації (точка роси) залишається незмінним при незмінній температурі. Початкові умови:

- 8 м^3 повітря,
- 20°C ,
- 1 бар (абс),
- $\varphi_{\text{відн}} = 70\%$,
- $\text{ТР} = 17,148\text{ г}/\text{м}^3$ при 20°C ,
- вміст води в повітрі: $17,148 \cdot 8 \cdot 0,7 = 96,029\text{ г}$.

Умови після стиснення:

- 1 м^3 повітря,
- 20°C ,
- 8 бар (абс),
- $\varphi_{\text{відн}} = 100\%$,
- $\text{ТР} = 17,148\text{ г}/\text{м}^3$ при 20°C ,
- вміст води в повітрі: $17,148 \cdot 1 \cdot 1 = 17,148\text{ г}$,
- стиснене повітря не може вміщувати води більше цього значення, і вода буде конденсуватися.

Результат: волога сконденсувалася.

Кількість конденсату складе: $96,029 - 17,148 = 78,881\text{ г}$

У процесі стиснення повітря в компресорі температура значно зростає, отже, волога не буде конденсуватися в компресорі.

Приклад.

Початкові умови:

- 8 м^3 повітря,

- 20 °C,
- 1 бар (абс),
- $\varphi_{\text{відн}} = 70\%$,
- $TR = 17,148 \text{ г/м}^3$ при 20 °C,
- вміст води в повітрі $17,148 \cdot 0,7 \cdot 8 = 96,029 \text{ г}$.

Умови після стиснення:

- 1 м³ повітря,
- 80 °C,
- максимальний вміст вологи в повітрі при 80 °C (TR) = 290,017 г/м³,
- фактичний вміст вологи в стислому повітрі становить 96,029 г/м³, що менше максимального значення вмісту вологи в стисненому повітрі при 80 °C, отже, волога не конденсується,
- $\varphi_{\text{відн}} = 96,029/290,017 = 33\%$.

Приклад.

Умови після охолодження стисненого повітря в кінцевому охолоджувачі компресора.

В кінцевому охолоджувачі компресора стиснене повітря охолоджується до температури 35 °C (на 10 °C вище температури навколишнього середовища).

Максимальний вміст вологи у повітрі при 35 °C (TR) = 39,286 г/м³.

Вміст вологи в стисненому повітрі становить 96,029 г/м³, що вище максимального значення вмісту вологи у стисненому повітрі при 35 °C, отже, волога буде конденсуватися в охолоджувачі.

Кількість конденсату складе: $96,029 - 39,286 = 56,743 \text{ г}$.

Якщо ми говоримо про компресор, який має продуктивність 8м³/хв, та стискає повітря до 7 бар надлишкового тиску, то кожен годину (разом із стисненим повітрям) з нього буде виходити 3,4 л конденсату, а в добу 82 л.

Примітки для проектування.

Найважливіше питання при проектуванні повинно звучати так: до якої мінімальної температури може бути охоложене стиснене повітря в трубопровідній мережі та біля споживачів?

Звертайте увагу на температуру. Пам'ятайте, охолодження в результаті розширення у споживачі, відбувається наступним чином: при кімнатній

температурі 20 °С, зниження тиску стисненого повітря у споживачі на 1 бар призводить до зниження температури до -15...-20 °С.

При виборі точки роси під тиском, звертайте увагу на температуру на 2...5 °С нижче від найнижчої температури.

1.2 Виробництво та розподіл стисненого повітря

Стиснене повітря є важливим енергоресурсом для промисловості, займає четверте місце після електроенергії, природного газу і води. Однак, на відміну від перших трьох енергоресурсів, стиснене повітря, виробляється на місці. Тому у споживача значно більше можливостей контролювати його споживання і витрати на його виробництві.

У пневматичних системах використовується найбільш поширений і найменш дорогий газ – повітря. Для забезпечення енергії, здатної виробляти механічну роботу, необхідно створити тиск повітря вище атмосферного.

Є три важливі причини, через які варто витратити час і сили на зниження витрат в системах стисненого повітря:

- виявлення та усунення витоків і нераціонального використання (заощаджує енергію та гроші);
- підвищення надійності і експлуатаційних параметрів систем стисненого повітря;
- зниження електроспоживання і, відповідно, зниження викидів вуглекислого газу (зменшує шкідливий вплив на навколишнє середовище).

Добре спроектована і належним чином введена в експлуатацію енергоефективна система стисненого повітря може приносити споживачеві десятки і навіть сотні тисяч гривень щорічної економії.

Застосування системного підходу

Енергоефективна система стисненого повітря забезпечується:

- при постійному підтриманні у справному стані;
- при регулярному техобслуговуванні всього обладнання і контролем за експлуатаційними параметрами;

- якісно спроектованою мережею (вірно обрані фітинги, фільтри, осушувачі, труби і трубні з'єднання) – для досягнення мінімальних втрат тиску;

- роботою при постійному або регулярному моніторингу із визначенням питомого енергоспоживання на основі отриманих даних;

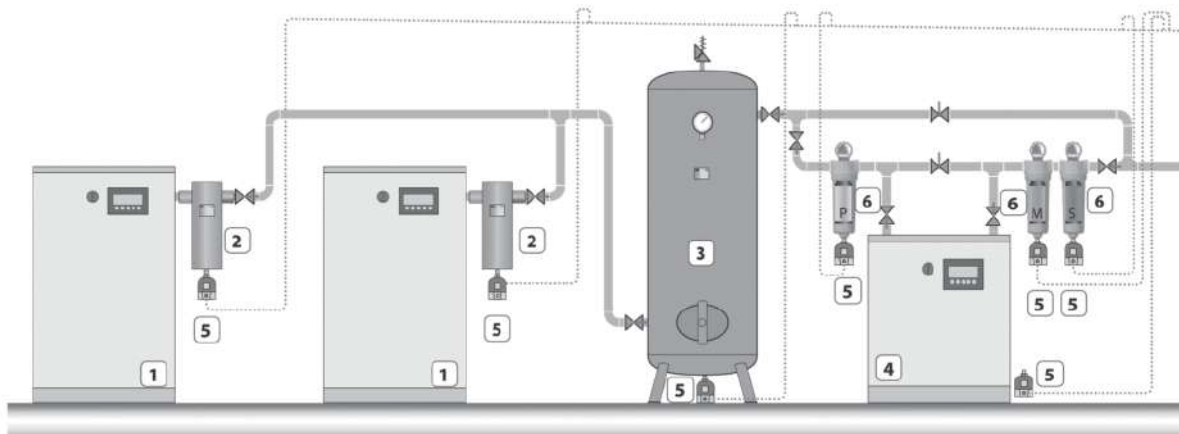
- експлуатацією персоналом, добре обізнаним про витрати на виробництві стисненого повітря (персонал пройшов навчання з ефективного використання обладнання, яке споживає стиснене повітря);

- постійним моніторингом енерговитрат.

Більшість систем стисненого повітря складаються покроково, а не розробляються в ході проектування.

При цьому, зазвичай, залученими виявляються кілька підрозділів підприємства, у тому числі:

- виробництво;
- технічна служба/управління підприємством;
- бухгалтерія/служба закупівель;
- енергетична служба/екологічний контроль.



- 1 – компресор; 2 – циклонний сепаратор для видалення конденсату;
3– ресивер; 4 – осушувач стисненого повітря; 5 – конденсатовідвідник;
6 – фільтр

Рисунок 1.10 – Схема типової системи виробництва і підготовки стисненого повітря

Основні типи втрат стисненого повітря:

- витоки;
- падіння тиску;
- робота компресора через відсутність потреби у стисненому повітрі.

Витоки існують у всіх системах стисненого повітря. Зниження витоків є найголовнішим енергозберігаючим заходом.

Рівень витоків у неправильно змонтованій керованій системі стисненого повітря може перевищувати 50% від обсягу виробництва стисненого повітря.

Витоки стисненого повітря також ведуть до додаткових витрат внаслідок:

- коливань тиску в системі, які можуть призвести до зниження ефективності роботи пневматичних інструментів та іншого обладнання з пневматичним приводом (потенційно може викликати зниження обсягів виробництва);

- скорочення терміну служби обладнання і позапланових ремонтів через непотрібність циклічної роботи компресора;

- надлишкової потужності компресора.

Причини витоків різноманітні, але найбільш частими є:

- залишені відкритими вентиля ручного відведення конденсату;
- залишені відкритими запірні вентиля;
- негерметичні гнучкі трубопроводи та з'єднання;
- негерметичні труби і трубні з'єднання;
- негерметичні регулятори тиску;
- увімкнене без потреби обладнання (яке споживає повітря, коли в його роботі немає необхідності).

1.2.1 Система розподілу повітря

Завдання системи розподілення повітря – доставляти стиснене повітря від випускного патрубку компресора до місць його споживання з мінімальними витоками, мінімальною втратою тиску і мінімальною зміною якості повітря.

Тертя та витоки викликають падіння тиску на ділянці від компресора до місця використання стисненого повітря. Ці втрати енергії в системі

розподілення повітря у великій мірі обумовлюються її конструкцією і проектною схемою.

У цьому розділі увагу буде акцентовано на правильному виборі трубопроводу, що надасть змогу зменшити падіння тиску в системі напірної магістралі.

1.2.2 Підбір діаметру трубопроводу підведення повітря

У структурі витрат, пов'язаних з магістральним трубопроводом стисненого повітря, більша частина відведена на частку початкових витрат на трубопроводи, монтаж системи стисненого повітря. Тому, вимагається щоб трубопровід меншого діаметру забезпечив економію капітальних витрат.

Однак, це лише уявна економія, оскільки опір, пов'язаний із меншим діаметром трубки, викликає більше падіння тиску в системі та призводить до зростання енергоспоживання.

Зростання енергетичних витрат може перевищити додаткову вартість трубопроводу більшого діаметра. На рисунку наведено, як змінюється (в залежності від діаметру трубки) величина потужності, яка є необхідною для перекачування 50 м³/год повітря з тиском 7 бар сталевим трубопроводом на відстань 100 м.

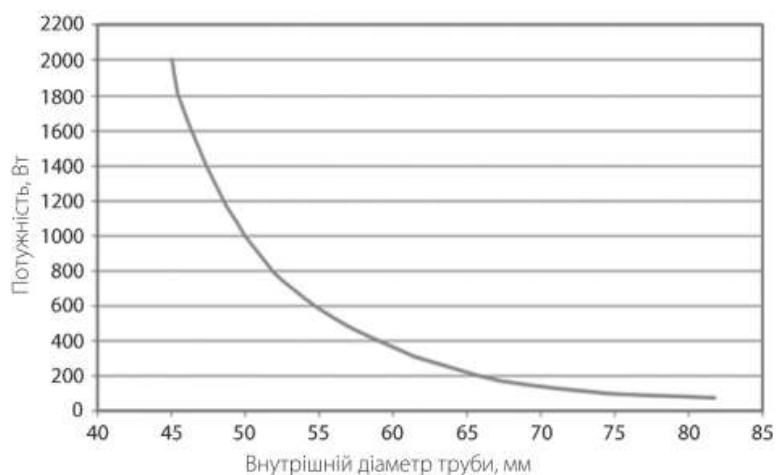


Рисунок 1.11 – Втрата потужності в трубопроводі різного діаметру

В цілому, необхідний діаметр трубопроводу повинен визначатися за максимальною швидкістю повітряного потоку в магістральній мережі 6 м/с. У відводах, загальною довжиною до 15 м допустима швидкість до 15 м/с.

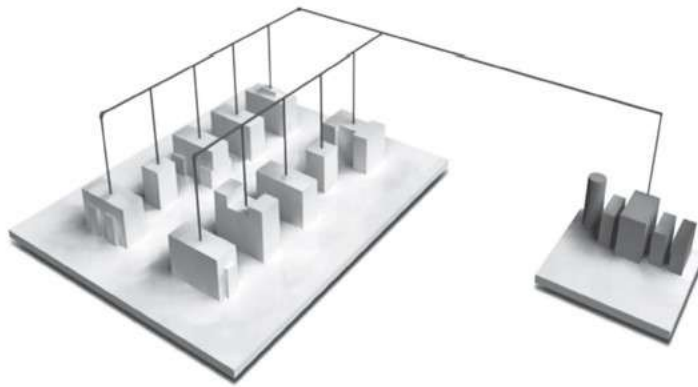


Рисунок 1.12 – Оцінка падіння тиску в трубопроводі і визначення оптимального діаметра трубопроводу

Наведена на рис. 1.13 діаграма може бути використана для визначення падіння тиску в існуючому трубопроводі стисненого повітря. За допомогою неї можна також обирати діаметр трубопроводу. Отримані за допомогою діаграми значення можуть бути використані тільки для оціночних розрахунків.

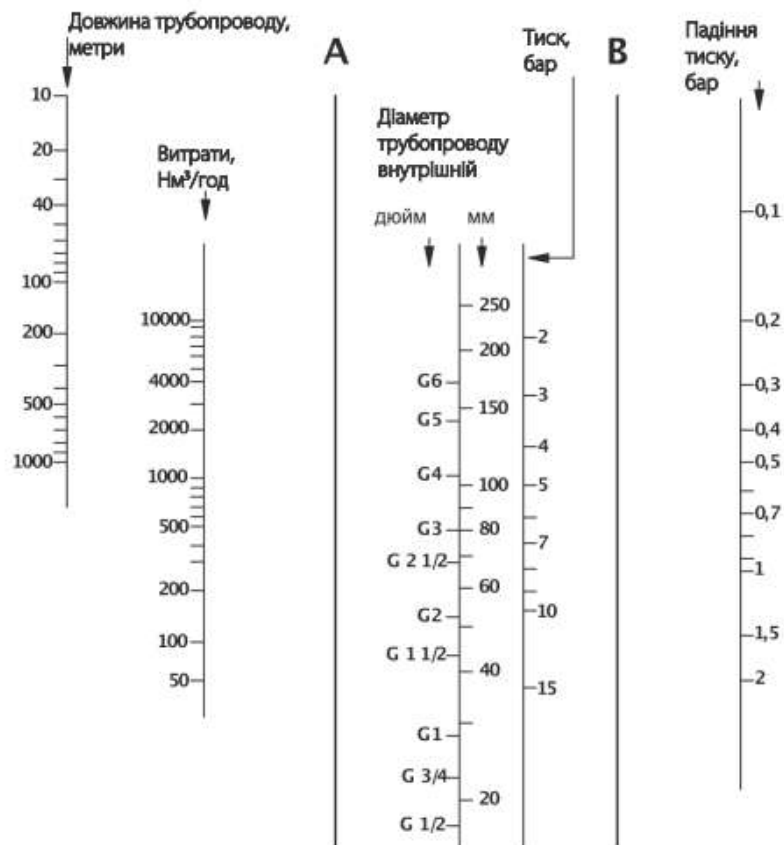


Рисунок 1.13 – Діаграма для визначення падіння тиску та діаметру трубопроводу

Для визначення параметрів внутрішнього діаметра первинної мережі можна застосувати два методи:

- аналітичний – враховуються втрати тиску в різних відгалуженнях мережі. Цей метод вимагає часу, але є точним;
- графічний – спрощений і практичний метод. Використовується номограма, в якій представлені всі величини, параметри яких потрібно визначити.

Примітка: номограма не приймає до уваги падіння тиску від фітингів, вентилів та ймовірних переходів, розташованих на первинній мережі розподілення стисненого повітря.

Всі трубопроводи стисненого повітря повинні проектуватися з урахуванням наступних моментів:

- підбір діаметрів труб повинен бути націлений на мінімізацію падіння тиску і при можливому розширенні системи;
- фітинги та вентиля повинні обиратися за принципом створення найменшого опору повітряному потоку. Наприклад, коліно труби великого діаметру має менші втрати стисненого повітря, ніж труба із різкою зміною напрямку повітряного потоку;
- кульові крани кращі за засувки;
- всі труби повинні мати надійні опори для мінімізації зсувів і провисання.

Це допоможе мінімізувати витрати, уникнути відкладення наносів і корозії та продовжити термін служби трубопроводу.

Існує два основних види систем розподілення стисненого повітря – одномагістральна і кільцева.

Одномагістральна мережа є найбільш зручною для простих установок, де місця використання та виробництва стисненого повітря знаходяться відносно недалеко один від одного. В добре спроектованій системі максимальне падіння тиску повинно бути не більше 0,2 бар. Бажано встановити магістральний повітропровід найбільшого діаметру, особливо, якщо в майбутньому можливе розширення системи.

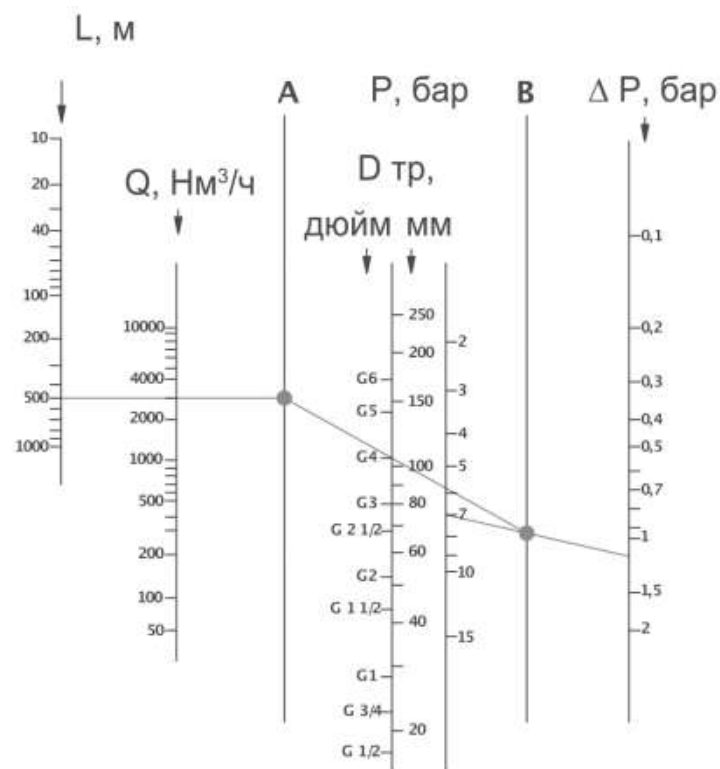
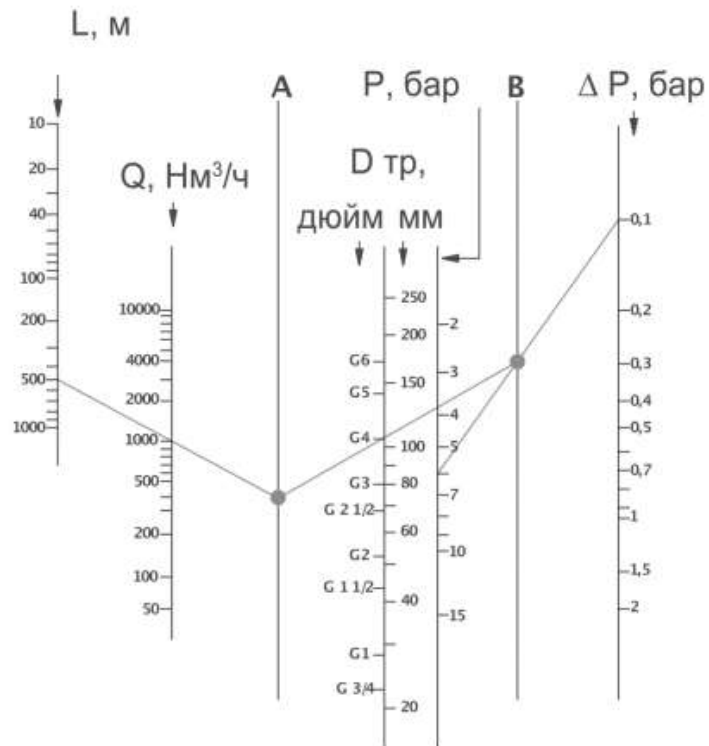


Рисунок 1.14 – Номограми для визначення внутрішнього діаметра магістрального трубопроводу

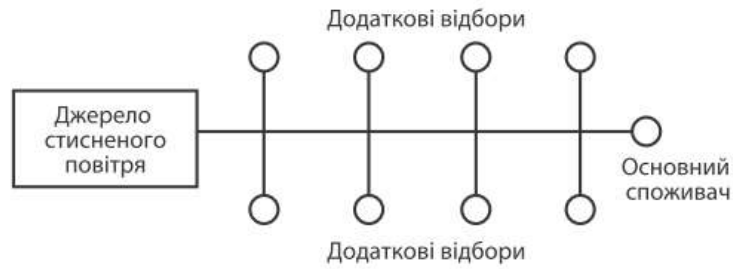


Рисунок 1.15 – Одномагістральна мережа

Кільцевій магістральний повітропровід є кращим варіантом для більших систем з численними точками відбору. Оскільки повітря подається на будь-яке обладнання за двома напрямками, швидкість скорочується вдвічі, падіння тиску зменшується. Інша перевага полягає у тому, що можна передбачити встановлення запірних клапанів. Також є можливість перекривати різні ділянки системи для проведення техобслуговування, не перериваючи підведення повітря іншим споживачам. Такі системи характеризуються більш високим рівнем енергоефективності.

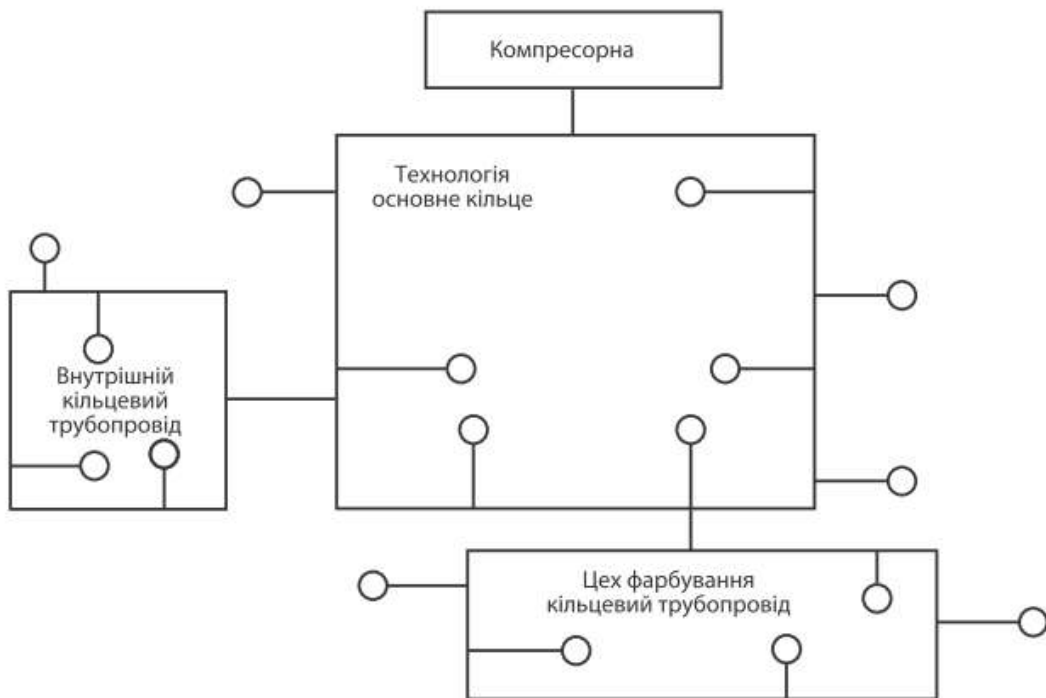


Рисунок 1.16 – Кільцевій магістральний повітропровід

1.2.3 Компресори

Компресор – пристрій для підвищення тиску (стиснення) і переміщення газоподібних речовин.

Робочий тиск і необхідний об'єм повітря є основними визначальними критеріями для вибору компресора.

Енергоефективність будь-якого компресора залежить від його:

- конструкції;
- установки;
- експлуатації;
- техобслуговування.

Ефективність компресорів найвища при роботі з повним навантаженням. Навіть, на холостому ходу електроспоживання компресорів може досягати 20...70% від рівня споживання (при повному навантаженні). Таким чином, для досягнення максимальної ефективності роботи компресора необхідно, щоб його виробіток відповідав потребі в стисненому повітрі.



Рисунок 1.17 – Види компресорів

Компресори повинні знаходитися у сухому, чистому, прохолодному і добре вентильованому приміщенні.

Для стиснення теплого і вологого повітря потрібно не тільки більше енергії, а й додаткове осушення, щоб волога не стала причиною корозії труб та інших проблем з обладнанням. Можливо, потрібно буде примусова

вентиляція в компресорному приміщенні для відведення теплоти. Повітрязбірник приміщення компресорної станції повинен, за можливістю, розташовуватися на стіні, яку орієнтовано на північ, або, в затіненому місці. Необхідно встановити решітку для захисту від сторонніх предметів. Фільтрувати пил і бруд з метою очищення повітря, яке подається, потрібно для мінімізації зношування та уникнення пошкоджень системи стисненого повітря. Слід регулярно перевіряти фільтри повітрязбірника і замінювати їх до того, як падіння тиску на них стане значним.

Задача вибору компресора найбільше підходить для конкретного застосування, має ґрунтуватися на низці факторів, головними з яких є наступні:

- рівень якості повітря, необхідний для конкретного застосування/технології (наприклад, чи є необхідність в стисненому повітрі, яке не містить мастила);
- дотримання витрат повітря та тиску;
- наявне фінансування та наступні експлуатаційні витрати.

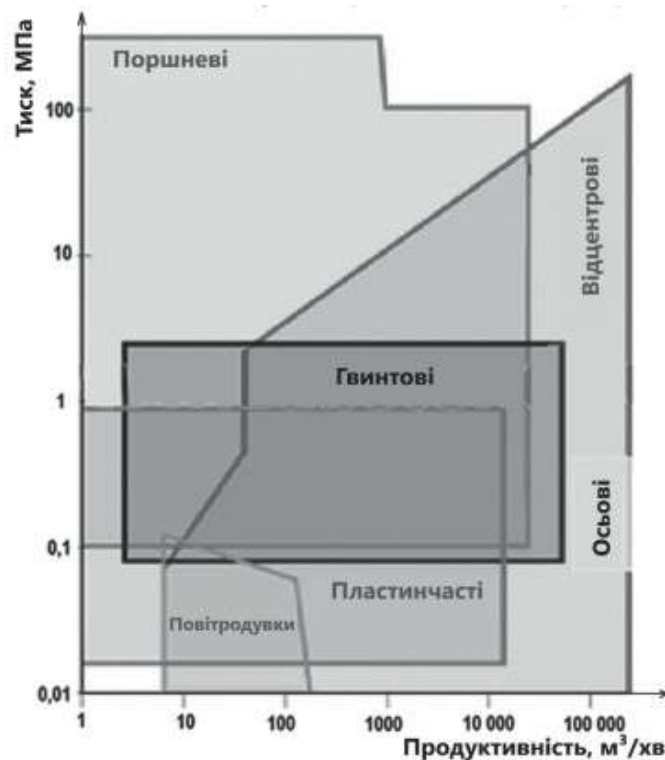


Рисунок 1.18 – Область застосування різних типів компресорів

У типових промислових системах, які працюють із тиском від 6 до 10 бар, зазвичай використовуються гвинтові, пластинчасті або поршневі компресори. Відцентрові компресори, як правило, застосовуються в системах із великими витратами.

Компресори, регулювання та акумулювання стисненого повітря.

1. Вимикайте компресори, коли немає потреби в стисненому повітрі (але переконайтесь, що це не призведе до зупинки потрібної частини обладнання).

2. Придивляйтесь і прислухайтесь чи працюють запобіжні клапани. Якщо так, то регулювання функціонує ненормально. Чи часто компресори включаються та відключаються?

3. Чи можна зменшити тиск на виході компресору? Чим вище вимоги до тиску на виході, тим вище енергоспоживання.

4. У запиленому приміщенні або за несприятливих умов роботи компресорів слід частіше міняти фільтри повітрязабірника.

5. Для підвищення ефективності роботи компресора використовуйте джерело повітря з максимально низькою температурою.

6. Подумайте про встановлення окремого компресора там, де є необхідність у стисненому повітрі з іншим тиском або в інші години роботи.

7. Вивчіть розмір ресиверів. Якщо компресори часто змінюють режим роботи з навантаженням і без неї (цикли), можливо, слід подумати про додатковий ресивер.

8. Вивчіть розташування повітроприймачів (допоміжний ресивер слід встановити поблизу від споживача з імпульсним високим споживанням стисненого повітря).

1.2.4 Ресивер

Для оптимальної роботи розмір ресивера повинен обиратися з таким розрахунком, щоб забезпечити найвище споживання повітря в системі.

У випадках, коли споживання дискретне (не потрібне), краще встановити ресивер ближче до такої виробничої ділянки або виду обладнання, ніж вибирати головний ресивер більшого розміру та встановлювати додатковий компресор, який буде простоювати більшу частину часу.

Для прийняття рішення про необхідність місцевого (допоміжного) ресивера стисненого повітря:

- оцінить загальний максимальний необхідний об'єм для основного ресивера, як описано вище;

- потім оцінить об'єм ресивера, необхідного для покриття пікового споживання. Якщо він перевищує 10% об'єму основного, доцільно встановити місцевий резервуар.

Наприклад, якщо загальна продуктивність компресора дорівнює 1200 л/хв, максимальний об'єм сховища становить 200 літрів. Якщо єдиний пік споживання становить 150 л/хв, максимальний обсяг сховища дорівнює 25 літрів. В цьому випадку рекомендується встановити місцевий ресивер.

Розмір будь-якого резервного ресивера залежить від споживання стисненого повітря в розрахунку на операцію, та допустимого падіння тиску (його можна оцінити за формулою):

$$\text{Необхідний об'єм ресивера} = \frac{\text{споживання в розрахунку на операцію (літрів повітря)}}{\text{допустиме падіння тиску (бар)}}$$

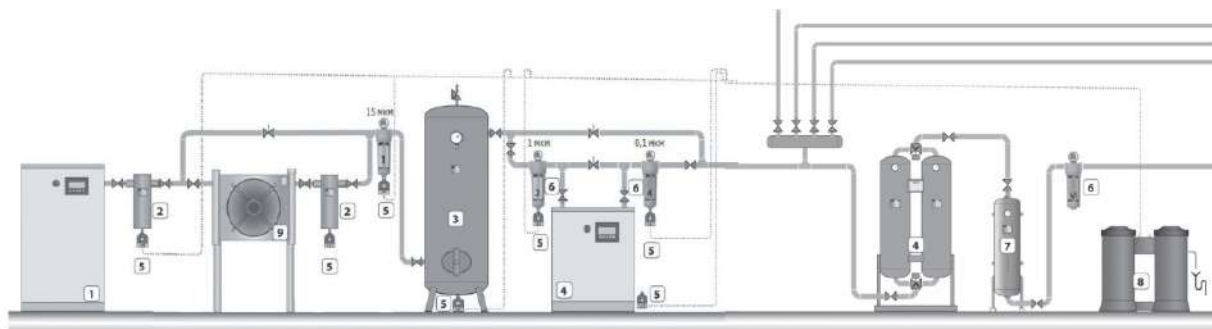


Рисунок 1.19 – Ресивер

Важливо переконатися, що компресор досить продуктивний, щоб наповнити ресивер повітрям до вихідного рівня тиску, до наступного періоду пікового споживання.

Виробництво стисненого повітря починається із установки, яка включає в себе:

- компресор, який приводиться в рух електродвигуном;
- резервуар (ресивер), як акумулятор енергії;
- прилади системи безпеки, регулювання (клапани безпеки, скидання, фільтри тощо);
- розподільчі схеми.



- 1 – компресор; 2 – вологовіддільник циклонного типу; 3 – основний ресивер;
4 – осушувачі різних типів; 5 – клапани зливу конденсату;
6 – фільтр грубого очищення; 7 – фільтри тонкого очищення різних типів;
8 – масло- та вологовідділювачі; 9 – доохолоджувач

Рисунок 1.20 – Установка з виробництва стисненого повітря

1.2.5 Вибір пристроїв для очищення повітря та схеми їх застосування

Очищення повітря на підприємстві – це комплекс заходів для раціонального вибору, розміщення, монтажу та експлуатації очисних пристроїв на трьох ділянках пневматичної мережі: на компресорній станції, на магістральних повітропроводах та у споживача. Нижче наведені дані про найбільш характерні схеми використання очисних та дренажних пристроїв на повітропроводах і споживачах.

1.2.6 Видалення забруднень із магістральних повітропроводів

Вибір схем і пристроїв очищення повітря у магістральних трубопроводах залежить від класу забрудненості стисненого повітря на виході

з компресорної станції (установки), від вимог до чистоти стисненого повітря у споживачів.

У тому випадку, коли на компресорній станції застосовано одну із схем холодильникового типу, в міру віддалення від компресорної станції концентрація забруднень в магістралях зростає внаслідок конденсації водяної пари та мастила, корозії трубопроводів із сполуками окалини, іржі, фарби та ін. Очисні пристрої в магістральних повітроводах не тільки захищають їх від надмірного забруднення та обводнення, а й захищають групові та індивідуальні очисні пристрої від перевантаження та зниження ефективності, можливих за надмірних концентраціях забруднень.



Рисунок 1.21 – Виробництво та підготовка стисненого повітря

Визначальним фактором при виборі раціонального місця, конструкції, розміру та числа очисних пристроїв у магістралі є характер процесу конденсації та накопичення вологи під час руху повітря в трубопроводах. Зазвичай у магістральних трубопроводах встановлюють очисні пристрої інерційного типу, тому що застосування пристроїв фільтруючого типу призводить до збільшення втрат тиску при проходженні повітря, що ускладнює експлуатацію магістральних очисних пристроїв. Воду та мастило у

рідкому стані видаляють з магістралей за допомогою водозбірників, або очисних пристроїв. Водозбірники встановлюють у низьких місцях і колінах трубопроводів, при великому вмісті конденсату, перед магістральними вологовідділювачами. Пристрої для уловлювання вологи розміщують в місцях максимального скупчення конденсату – на вході в цеху.

Для забезпечення нормальної роботи водозбірників і вологовіддільників необхідно вживати заходи проти замерзання в них конденсату за мінусових температур, за можливості розташовувати їх в приміщеннях або утеплювати.

Якщо на компресорній станції проводиться очищення стисненого повітря, яке виключає конденсацію води та мастила у трубопроводах з використанням адсорбційних осушувачів, або за допомогою глибокого охолодження – то необхідність в установці водозбірників і вологовіддільників відпадає.

1.2.7 Стандарти забрудненості стисненого повітря

За ДСТУ 4169:2003 регламентуються: розмір твердих частинок (d, мкм), концентрація твердих частинок (C), концентрація крапельних фракцій мастила (Oil) та концентрація води (W), яка вимірюється в мг/м³, точка роси водяної пари.

Таблиця 1.9 – Стандарти забрудненості стисненого повітря за ДСТУ 4169:2003

Клас, с	D, мкм	C, мг/м	Oil, мг/м	W, мг/м	Клас, с	D, мкм	C, мг/м	Oil, мг/м	W, мг/м
0	0,5	0,001	0	0	-	-	-	-	-
1	5	1	0	0	2	5	1	500	0
3	10	2	0	0	4	10	2	800	16
5	25	2	0	0	6	25	2	800	16
7	40	4	0	0	8	40	4	800	16
9	80	4	0	0	10	80	4	800	16
11	*	25	0	0	12	*	12,5	3200	25
13	*	40	0	0	14	-	255	10000	100
Для класів 0, 1, 3, 5, 7, 9, 11, 13 точка роси водяної пари – нижче мінімальної робочої температури, не менше ніж на 10 °С					Для класів 2, 4, 6, 8, 10, 12, 14 точка роси водяної пари не регламентується. * Значення не регламентується				

Приклад запису: «Повітря Кл. 7 ДСТУ 4169:2003»

Стандарт ISO 8573.1 обумовлює три види класів: які регламентують вміст твердих частинок за їх максимальним розміром (d , мкм) і концентрації (C , мг/м³); які регламентують вміст водяної пари за точкою роси (T , °C); які обумовлюють вміст крапельних фракцій мастила (Oil, мг/м³).

Приклад запису: “ISO 8573.1 клас 1.4.1” для повітря класу 1 за частинками, класу 4 за точкою роси та класу 1 за мастилом.

Приклад 2: Клас очищення повітря відповідно до ISO 8573-1:2010 [7:5:4] – повітря класу 7 за твердими частками, класу 5 за вологою та класу 4 за утримання мастила.

1.3 Блоки підготовки повітря

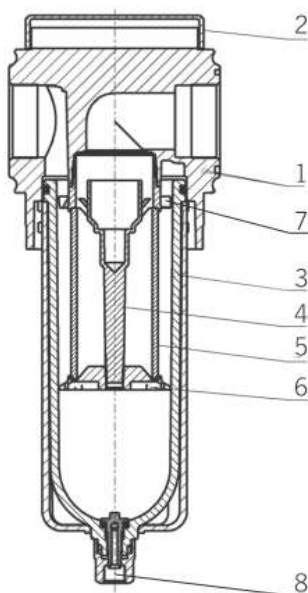
1.3.1 Фільтри

Компанія Samozzi виробляє фільтри для очищення повітря від краплинної вологи, мастила та твердих частинок. Всі фільтри Samozzi відносяться до фільтрів вихрового типу (відцентровий спосіб очищення). Конструкція фільтра даного типу зображена на рисунку.

Потрапляючи у фільтр, повітря контактує із вхідною перегородкою – це сприяє інтенсивному відділенню крапельної вологи та мастила. Проходячи далі, повітря завихрюється. Це досягається використанням у конструкції фільтра спеціальної крильчатки 7. Після завихрення крильчаткою, повітря рухається за дотичною до стінки склянки 3. На крапельки води при русі дотичній до кола (компресорне масло та тверді частинки) діє відцентрова сила, яка відкидає їх на стінки склянки, де конденсуються всі забруднення. Далі, вони стікають до клапана зливу конденсату 8 (на рис.1.22 зображено напівавтоматичний клапан). Всі ці процеси відбуваються у турбулентній зоні фільтра. Ця зона розташована вище гайки-стабілізатора 6. Стабілізатор у фільтрах вихрового типу виконує роль розподільника між турбулентною та ламінарною зонами. Турбулентна зона розташована вище стабілізатора. У ній відбувається інтенсивне відділення вологи і твердих частинок зі стисненого повітря. Ламінарна зона, утворюється нижче стабілізатора. В ній накопичуються всі забруднення. Відсутність завихрень повітря у цій зоні перешкоджає поверненню забруднень в пневмережі.



Рисунок 1.22 – Фільтр і фільтр-регулятор серії МХ компанії Samozzi



- 1 – корпус (алюміній); 2 – кришка (поліацеталь);
 3 – стакан з полімерним покриттям (полікарбонат/поліамід);
 4 – вісь фільтроелемента (поліацеталь); 5 – фільтруючий елемент (поліетілен); 6 – гайка-стабілізатор (поліацеталь); 7 – дефлектор (крильчатка) (поліацеталь); 8 – клапан зливу конденсату (латунь з технополімером)

Рисунок 1.23 – Фільтр з напівавтоматичним клапаном (ущільнення NBR)

В процесі експлуатації фільтрів із напівавтоматичним скиданням вологи необхідно стежити щоб рівень конденсату у склянці не піднімався вище гайки-стабілізатора 6.

У фільтрах компанії Camozzi застосовується два види клапанів для скидання конденсату: клапан для напівавтоматичного скидання конденсату (скидання конденсату відбувається з падінням тиску в мережі нижче 0,3 бар, після відключення компресора в кінці робочої зміни) та клапан, який забезпечено спеціальним механізмом для автоматичного скидання конденсату (з його підйомом до певного рівня).

Після відділення краплинної вологи у турбулентній зоні, повітря проходить через фільтруючий елемент 5. Фільтруючий елемент являє собою конус із пористого пластику, або циліндр із багат шарового пористого матеріалу.

Призначення його пов'язано із утриманням твердих частинок, не відділених під дією відцентрової сили.

Мінімальні розміри частинок, які утримуються, визначає тонкість очистки фільтруючого елемента. Фільтри Camozzi за тонкістю очищення діляться на 25; 5; 1; 0,01; 0,003 мкм. За ступенем очищення фільтри – від 99,9 до 99,999%.

За мірою забруднення фільтруючі елементи із пластику (25; 5 мкм) рекомендується промити у теплій мильній воді. А фільтруючі елементи субмікронного очищення замінити. Це допоможе уникнути зменшення витрати повітря через фільтр та погіршення ступеню очищення.

Правильно обраний фільтр забезпечує:

- необхідну тонкість очищення;
- необхідні витрати повітря через фільтр.

Вибір фільтра залежить від падіння тиску на фільтр-елементі та від витрат повітря. Для якісного очищення повітря падіння тиску на циклонному фільтрі повинно знаходитися у межах 0,2...0,3 бар, на коалісцентному фільтрі – у межах 0,1...0,3 бар.

Інформація для визначення необхідної точності очищення (для техніки) наведена в табл. 1.10.

Таблиця 1.10 – Класи очищення

Клас очищення ISO 8573-1:2010	Тверді частинки			Концентрація	Вода		Масило
	Гранично допустима кількість частинок в 1 куб. м.				Точка роси	Концентрація води в рідкій фазі	
	0,1...0,5 мкм	0,5...1 мкм	1...6 мкм	мг/м ³		г/м ³	мг/м ³
1	≤ 20,000	≤ 400	≤ 10	/	≤ -70 °C	/	0,01
2	≤ 400,000	≤ 6000	≤ 100	/	≤ -40 °C	/	0,1
3	/	≤ 90,000	≤ 1,000	/	≤ -20 °C	/	1
4	/	/	≤ 10,000	/	≤ +3 °C	/	5
5	/	/	≤ 100,000	/	≤ +7 °C	/	/
6	/	/	/	≤ 5	≤ +10 °C	/	/
7	/	/	/	5...10	/	≤ 0,5	/
8	/	/	/	/	/	0,5...5	/
9	/	/	/	/	/	5...10	/

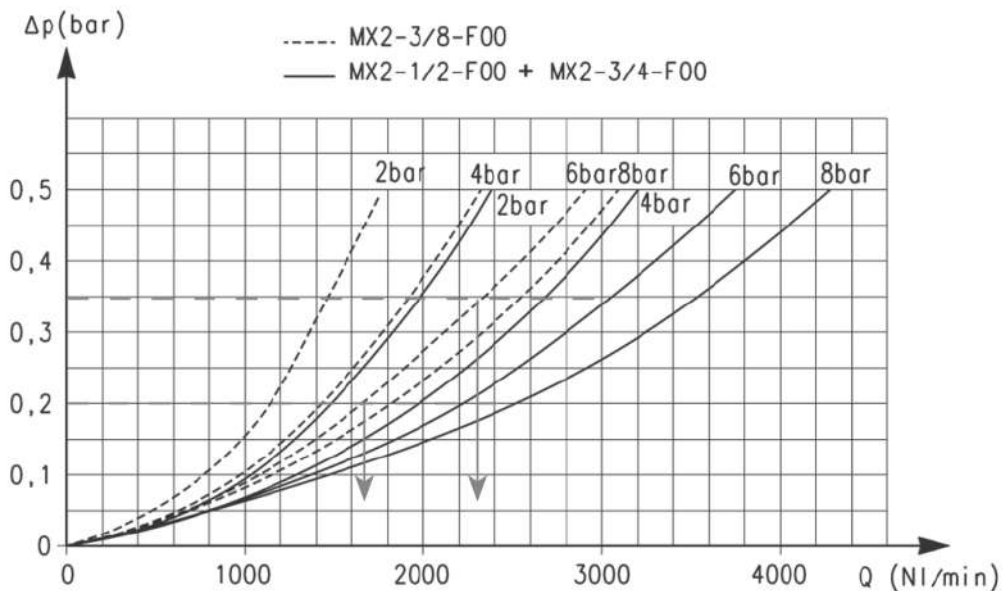


Рисунок 1.24 – Приклад діаграми витрат повітря фільтру серії MX2

Для якісного очищення повітря, падіння тиску на фільтрі повинно знаходитися у межах 0,2...0,3 бар. Це пов'язано із середньою швидкістю руху повітря у фільтрі. Якщо падіння тиску на фільтрі при запланованих витратах виявилось менше 0,2 бар – швидкість повітря у ньому буде невисока. Отже, відцентрова сила, яка діє на частинки вологи і мастила у турбулентній зоні невелика. Ефективність очищення падає. У такому випадку необхідно переходити на фільтр меншого типорозміру. При падінні тиску більше 0,3 бар – необхідно вибрати більший фільтр, тому що не виправдано збільшуються втрати тиску повітря при його транспортуванні від компресора до споживача. Це відзначається на вартості використання стисненого повітря.

При експлуатації фільтрів необхідно дотримуватися таких правил:

- фільтр повинен монтуватися строго вертикально;
- необхідно дотримуватися правильності підключення (конденсатовідділювачем униз) стисненого повітря до фільтру (правильний напрямок проходження повітря крізь фільтр показано стрілкою на його корпусі);

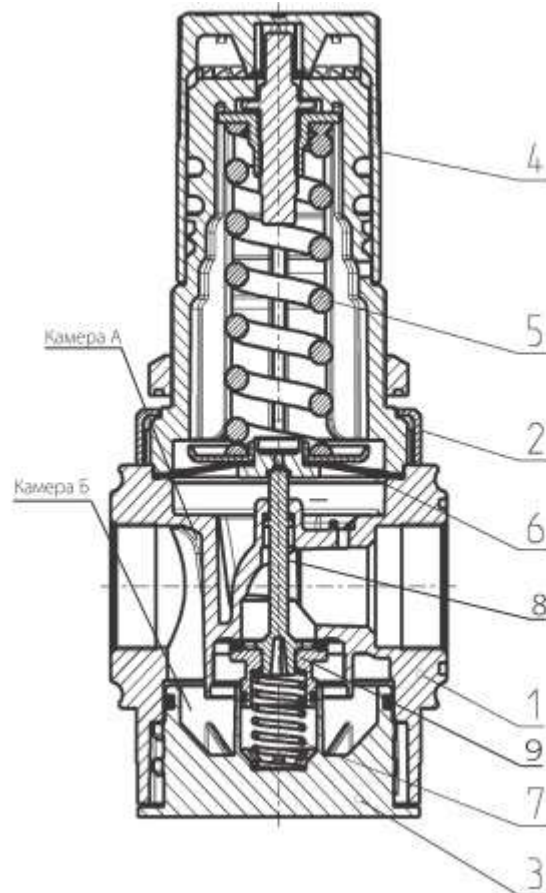
- при використанні фільтрів високої тонкості очищення, перед ними потрібно встановлювати фільтри з меншою тонкістю очищення.

1.3.2 Регулятори тиску

Регулятори тиску призначені для зменшення тиску у магістралі до рівня робочого тиску виконавчих елементів. А також для стабілізації вихідного тиску при компресії з боку споживача.

Основним елементом регулятора тиску є мембрана 6, закріплена у корпусі 1. Жорсткий центр мембрани зв'язано з одного боку пружиною 5, з регулюючим гвинтом та рукояткою 4 – а з іншого боку, штоком 8 із тарільчастим клапаном 9, який підтримується пружиною 7. Шток 8 має проточку, яка сполучає вихід регулятора із камерою Б. Пружина 5 впливає на мембрану 6 (зміна зусилля впливу проводиться рукояткою 4). Через мембрану на шток 8 та тарільчастий клапан 9 і підтримує пружину 7. Якщо зусилля, яке створюється регулювальною пружиною 1 перевищує зусилля підтримуючої пружини 7 – то клапан 9 відривається від сідла та пропускає стиснене повітря із входу регулятора на його вихід. Тарільчастий клапан 9 буде відкрито доти,

доки сумарне зусилля: яке створилося тиском у камері А на мембрану 6 (тиск в камері А дорівнює тиску на виході регулятора); зусилля підтримуючої пружини 7; зусилля підтискання тарільчастого клапану, яке створюється завдяки тиску в камері Б (тиск в камері Б дорівнює тиску на виході регулятора) – не перевищать зусилля, яке дає регулююча пружина 5.



- 1 – корпус; 2 – гайка; 3 – зовнішня кришка; 4 – рукоятка;
 5 – пружина з регулюючим гвинтом; 6 – мембрана; 7 – підтримуюча пружина; 8 – шток клапану зливу конденсату (латунь з технополімером);
 9 – тарільчастий клапан

Рисунок 1.25 – Конструкція регулятора тиску (ущільнення NBR)

Сумарне зусилля визначається вихідним тиском та зусиллям підтискаючої пружини 7, тобто як тільки тиск на виході регулятора перевищує налаштоване, тарільчастий клапан 9 відсікає вихід регулятора від його входу, тим самим перешкоджаючи подальше зростання вихідного тиску. Коли (через

споживання стисненого повітря) тиск на виході регулятора падає, нижче налаштованого, тарільчастий клапан 9 відкривається і здійснюється підняття тиску до налаштованого. Тоді відбудеться підтримка налаштованого тиску.

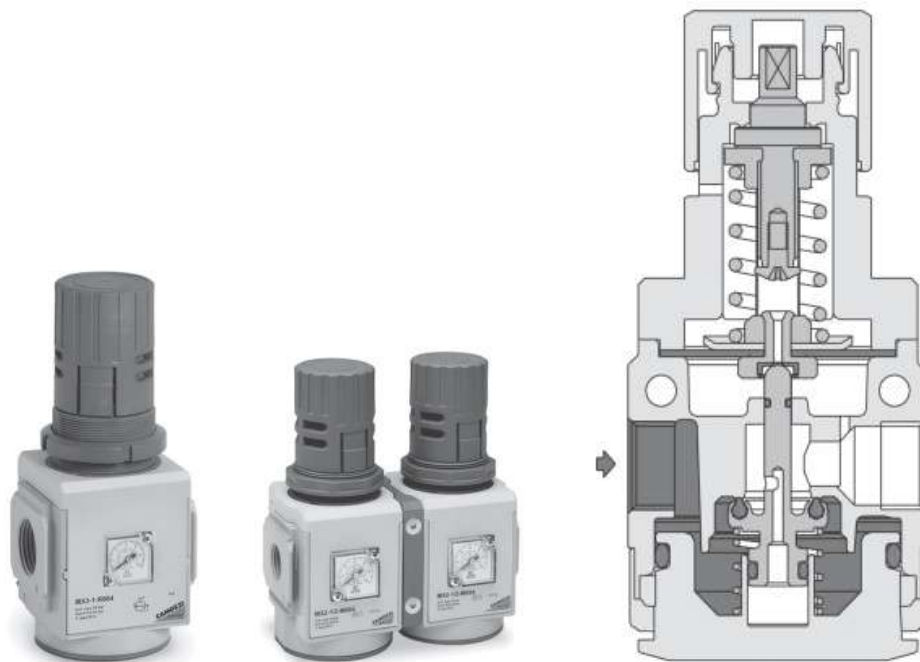


Рисунок 1.26 – Регулятори тиску Серія МХ

У разі значного перевищення вихідного тиску порівняно з налаштованим (це можливо, наприклад, при різкому впливі на пневмоциліндр будь-якої маси, компресії з боку споживача) відбувається наступне:

1. Високий тиск у камері А впливає на мембрану 4, внаслідок чого вона вигинається, стискаючи пружину 5.

2. Тарільчастий клапан 9 відсікає вихід регулятора від входу. Це відбувається тому що на шток 8 більше не діє зусилля з боку мембрани 6. Тарільчастий клапан закривається під дією зусилля, яке створюється пружиною 7 і тиском в камері Б.

3. Після того як мембрана 6 вигнулась, її жорсткий центр вийшов із контакту зі штоком 8, який перекривав отвір у жорсткому центрі. Через отвір, який відкрився, надлишки стисненого повітря з боку споживача виходять в атмосферу. Це триває доти, доки тиск (на виході регулятора) не стане рівним налаштованому.

У номенклатуру регуляторів компанії Camozzi входить шість серій регуляторів: N, M, MC, MX, MD і T. Для всіх серій максимальний вхідний тиск становить 16 бар.

Регулятори серії M. Серія регуляторів із найменшими витратами. Застосовуються для подачі повітря на машини, механізми та робочі місця, із невеликим споживанням стисненого повітря 600...800 Нл/хв. З'єднання в корпусі трубні, дюймові циліндричні із різьбою G 1/8 і G 1/4.

Регулятори серії N. Застосовується для подачі повітря на машини, механізми та робочі місця, із середнім споживанням стисненого повітря 800...1200 Нл/хв. З'єднання трубні дюймові циліндричні різьблення G 1/8 і G 1/4.

Регулятори серії MD, MC і MX. Застосовуються для подачі повітря на машини і механізми із великим споживанням повітря та для невеликих цехів.

Витрати повітря серії MD – 800...4000 Нл/хв; серії MC – 2000...6000 Нл/хв; серії MX – 6500...12000 Нл/хв. З'єднання трубні дюймові циліндричні із різьбою: G 1/8, G 1/4, G 3/8, G 1/2, G 3/4 і G 1.

При виборі регулятора основну увагу приділяють забезпеченню необхідної витрати стисненого повітря при заданому тиску на виході регулятора.

Регулятор необхідно підібрати за умови падіння тиску за середній очікуваній витраті стисненого повітря при падінні тиску не вище 1 бара.

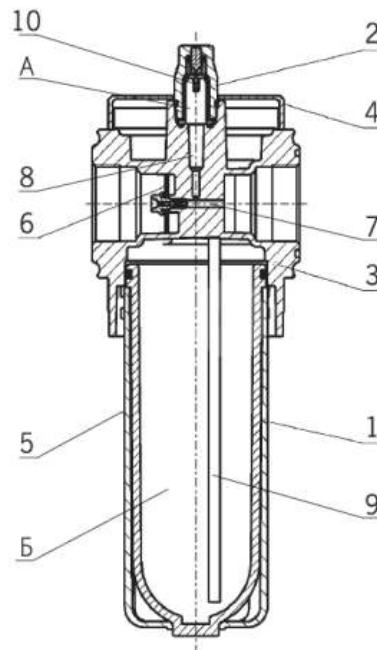
Якщо падіння тиску перевищує 1 бар, необхідно замінити регулятор на серії більшого типорозміру.

Фільтри-регулятори – це пристрої, які об'єднують можливості фільтру і регулятору в одному корпусі. Вони забезпечують більш бюджетне та компактне вирішення для систем підготовки повітря.

1.3.3 Маслорозпилювачі

Як правило, пневматика Camozzi використовується без маслорозпилювачів. У місця з'єднання рухомих елементів закладається мастило, розраховане на весь термін їх експлуатації. Якщо приводи використовуються в екстремальних умовах, наприклад, висока частота (до 40 і більше разів на хвилину) спрацьовування, швидкість переміщення (1 м/с і

більше), необхідність високої плавності переміщення, то застосовують повітря збагачене мастилом.



1 - стакан; 2 - прозорий ковпачок; 3 - корпус; 4 - кришка; 5 - кришка стакана;
6 - мембрана; 7 - ежекторуєчий канал; 8 - сопло; 9 - мірна трубка;
10 – капельниця

Рисунок 1.27 – Конструкція маслорозпилювача Серії МХ

Компанія Camozzi виробляє маслорозпилювачі з постійним розпиленням масла, які забезпечують отримання частинок мастила від 0,01 до 0,003 мм.

У маслорозпилювач необхідно подавати повітря попередньо очищене від компресорного мастила та вологи. На рисунку наводиться конструкція маслорозпилювача.

Якщо повітря подається на маслорозпилювач, а споживання повітря немає, то в порожнинах А і Б встановлюється рівний тиск. Маслорозпилювач при цьому не працює, розпилення мастила не відбувається.

З появою витрат повітря в маслорозпилювачі утворюються два повітряні потоки. Основний потік із входу, відгинаючи мембрану 6, проходить крізь вікна і надходить безпосередньо на вихід маслорозпилювача. Інша частина потоку проходить через отвір 7, обтікає сопло 8 і завдяки дії підсмоктування

основного потоку також надходить на вихід. При обтіканні сопла 8 допоміжний потік підсмоктує через нього повітря з порожнини А, знижуючи в ній тиск.

Під дією більш високого тиску порожнини Б (тиск в цій порожнині дорівнює магістральному) мастило зі склянки 1 піднімається по трубці 9, проходить через зворотний клапан і з входу 10 краплі падають в кінцевий отвір сопла 8. Частота падіння крапель при витраті повітря крізь маслорозпилювач регулюється дроселем, встановленому на лінії зв'язку між порожнинами А і Б. На частоту падіння крапель також впливають витрати повітря через маслорозпилювач і рівень мастила в склянці. Для зменшення впливу витрат стисненого повітря на інтенсивність падіння крапель олії використовується гнучка мембрана 6, яка автоматично регулює прохідний переріз крізь маслорозпилювач, тим самим підтримуючи необхідну швидкість потоку повітря в отворі 7. При падінні крапель з сопла 8 відбувається первинне розпорошення масла, і повітря з краплями мастила і захоплюється на вихід маслорозпилювача, де зустрічається з основним потоком повітря. Це призводить до повторного розпорошення мастила, що дозволяє отримати дрібні краплі олії, зважені в повітрі. Якщо витрат повітря крізь маслорозпилювач немає, тиск в порожнинах А і Б зрівнюється, падіння крапель олії із сопла 10 припиняється. При цьому кульковий зворотний клапан перешкоджає зниженню рівня масла в трубопроводі 9 (відбувається вирівнювання тиску в порожнинах А і Б). Тоді подача мастила на розпорошення починається відразу після виникнення витрат повітря крізь маслорозпилювач.

Маслорозпилювачі необхідно розташовувати максимально близько до механізмів. При цьому не допускається наявність перегинів або крутих поворотів трубопроводу на ділянці від маслорозпилювача до механізму. При проходженні різких поворотів на краплі олії в трубопроводі діє відцентрова сила, що при-тискає їх до зовнішньої стінки трубопроводу – це викликає осідання мастила на стінках трубопроводу і, як наслідок, зменшення кількості мастила, яке потрапляє у механізм.



Рисунок 1.28 – Маслорозпилювач серії MX

У номенклатурі Camozzi присутні маслорозпилювачі серій: MD, N, MC і MX. Маслорозпилювачі серії N, MD і MC1 застосовуються для змащення механізмів із малим до 3000 Нл/хв споживанням повітря. Мاستило заливають в маслорозпилювачі даної серії при відключеному вхідному тиску.

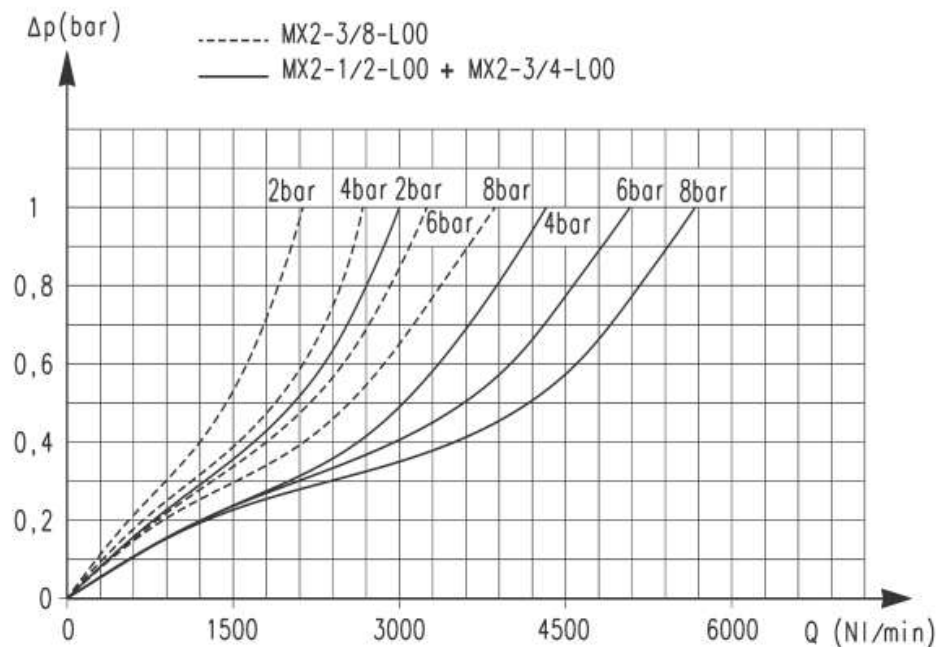


Рисунок 1.29 – Діаграма витрат повітря

Маслорозпилювачі серії MC2 і MX2, MX3 спеціально розроблені для роботи в тяжких умовах на механізмах безперервної роботи з великими витратами повітря до 10000 Нл/хв. Маслорозпилювачі даних серій з приєднувальними розмірами Серія MX від 3/8 до 1 дюйма забезпечені спеціальним пристроєм, що дозволяє здійснювати додавання масла без відключення вхідного тиску.

Для ефективної роботи маслорозпилювача він повинен обиратися таким чином, щоб падіння тиску на ньому при планованих витратах становило 0,2...0,5 бар. При падінні тиску нижче 0,2 бар необхідно обрати маслорозпилювач з меншим приєднувальним розміром і навпаки взяти маслорозпилювач з великим приєднувальним розміром при великому падінні тиску.



Рисунок 1.30 – Модульна збірка серія MX

1.3.4 Загальна рекомендаційна інформація щодо підготовки повітря

У той час як електрика, вода і газ зазвичай поставляються сторонніми організаціями, які гарантують їх норми, стиснене повітря постачається за рахунок коштів споживача пневмоапаратурою. Споживач повинен забезпечувати відповідну якість. Клас підготовки повітря має важливе значення для правильного функціонування пневматичних систем. Один кубічний метр повітря при атмосферному тиску може містити в собі:

- більше 150 млн. твердих частинок із розмірами від 0,01 мкм до 100 мкм;
- продукти згоряння у вигляді газів;

- водяна пара, кількість якої залежить від температури (при 30 °С біля 30 г/м³ води);
- мастило у вигляді незгорілих вуглеводнів (до 0,03 мг);
- мікроорганізми;
- різні хімічні забруднювачі, запахи тощо.

При стисненні повітря в N раз, концентрація забруднювачів також виростає в N раз. Для того щоб обмежити вплив цих речовин, на вході і виході з компресорів встановлюють фільтри, осушувачі та масловіддільники. Незважаючи на ці запобіжні заходи, повітря, під час його транспортування по трубах, або під час зберігання у ресіверах, може збирати частинки іржі.

Частина водяної пари, яка міститься у повітрі, при охолодженні, може переходити із газоподібного стану у рідкий. Також може відбуватися переродження масляних парів у смолообразні речовини, які не були утримані попередніми фільтрами. З цієї причини доцільно оснащувати систему блоками кінцевої підготовки повітря.

Блоки підготовки повітря можуть містити різні елементи: запірні клапани, регулятори тиску, клапани м'якого пуску і фільтри. Тільки у деяких обмежених застосуваннях ще використовуються і маслорозпилювачі.

Для кількісної оцінки допустимих домішок у повітрі існують стандарти, такі як ISO 8573-1: 2010, які класифікують повітря відповідно до його чистоти.

Цей стандарт визначає клас стисненого повітря відповідно до наявності трьох забруднюючих компонентів:

- тверді частинки;
- вода і водяна пара;
- концентрація мастила у фазах аерозолі і парів.

Ступінь очищення стисненого повітря слід приймати відповідно до класу ISO 8573-1:2010 [7:4:4]. Цей запис означає наступне:

Клас 7. Допускається максимальна концентрація твердих частинок 5 мг/м³, їх розмір не вказано.

Стандартні відцентрові фільтри Samozzi мають клас 7 при тому, що їх фільтруючі елементи виконані за технологією, яка дозволяє відокремити тверді частинки з розміром більше 25 мкм. Повітря, яке виходить з фільтрів та повітря, яке надходить на вхід інших компонентів, може містити тверді

частинки з максимальною концентрацією 5 мг/м³ максимальним розміром 25 мкм.

Клас 4. Температура не повинна опускатись нижче 3 °С для того, щоб водяна пара не почала сконденсовуватися і переходити у рідкий стан.

Класичні фільтри відцентрового типу мають характеристики, які дозволяють відокремлювати вологу з повітря, тільки за умови якщо вона знаходиться у рідкому стані або близькому до нього. Охолодження повітря призводить до конденсації, а потім до відбору води, який знаходиться у вигляді водяної пари. Потік повітря, що потрапляє в стакан фільтра, проходить мінімальну фазу розширення (коли газ різко розширюється, його температура знижується). Потім повітряний потік закручується на лопатях крильчатки. Відцентрова сила інерції направляє більш важкі частинки і водяний пар, який вже присутній у повітрі або конденсується та осідає у напрямку до дренажної системи. Деякі виконання фільтрів Samozzi мають клас очищення повітря 8, регламентуючи не температуру точки роси, а гранично допустимий вміст води у рідкій фазі. Це означає, що користувач додатково повинен забезпечити установку осушувачів у системі виробництва стисненого повітря для його охолодження.

Клас 4. Концентрація мастила не повинна перевищувати 5 мг/м³.

У компресорах застосовується мастило, яке в процесі роботи може потрапити в пневматичну систему у вигляді аерозолів, пари або рідини. Це масло, як і всі інші забруднюючі речовини, переміщується повітрям в пневматичній магістральній системі, входить у контакт з ущільненнями компонентів, а потім переміщується у навколишнє середовище через вихлопні отвори розподільників. Принцип роботи коалесцентних фільтрів і їх фільтруючих картриджів дозволяє накопичувати і зрощувати мікромолекули мастила, зважені у повітрі та видаляти їх. Коалесцентні фільтри Samozzi дозволяють досягти 2 і 1 класів очищення від мастил. Ефективна робота фільтрів тонкого очищення (коалесцентних) можлива тільки в тому випадку, якщо перед ним встановлені префільтри більш грубого очищення (відцентрові).

В Samozzi є кілька типів фільтрів з різними характеристиками, але кожен тип фільтра ефективний для певного забруднювача і не може

застосовуватися для відділення інших забруднюючих речовин. Фільтруючі елементи визначають клас фільтрів, ці елементи повинні бути замінені після певного періоду часу або після певної кількості годин роботи обладнання. Термін заміни фільтроелементів залежить від характеристик, які надходять на вхід фільтрів повітря, але не повинен перевищувати значень, наведених у таблиці 1.11.

Таблиця 1.11 – Терміни заміни фільтроелементів

Фільтруючий елемент 25 мкм	клас ISO 8573-1:2010 [7:8:4]	заміна кожні 18 місяців або 12 000 годин роботи
Фільтруючий елемент 5 мкм	клас ISO 8573-1:2010 [6:8:4]	заміна кожні 18 місяців або 12 000 годин роботи
Фільтруючий елемент 1 мкм	клас ISO 8573-1:2010 [2:8:2], потрібна установка префільтру, що забезпечує клас ISO 8573-1:2010 [6:8:4].	заміна кожні 12 місяців або 8 000 годин роботи
Фільтруючий елемент 0,01 мкм	клас ISO 8573-1:2010 [1:8:1], потрібна установка префільтру, що забезпечує клас ISO 8573-1:2010 [6:8:4].	заміна кожні 12 місяців або 8 000 годин роботи
Фільтруючий елемент на основі активованого вугілля	клас ISO 8573-1:2010 [1:7:1], потрібна установка префільтру, що забезпечує клас ISO 8573-1:2010 [1:8:1].	заміна кожні 6 місяців або 1 000 годин роботи

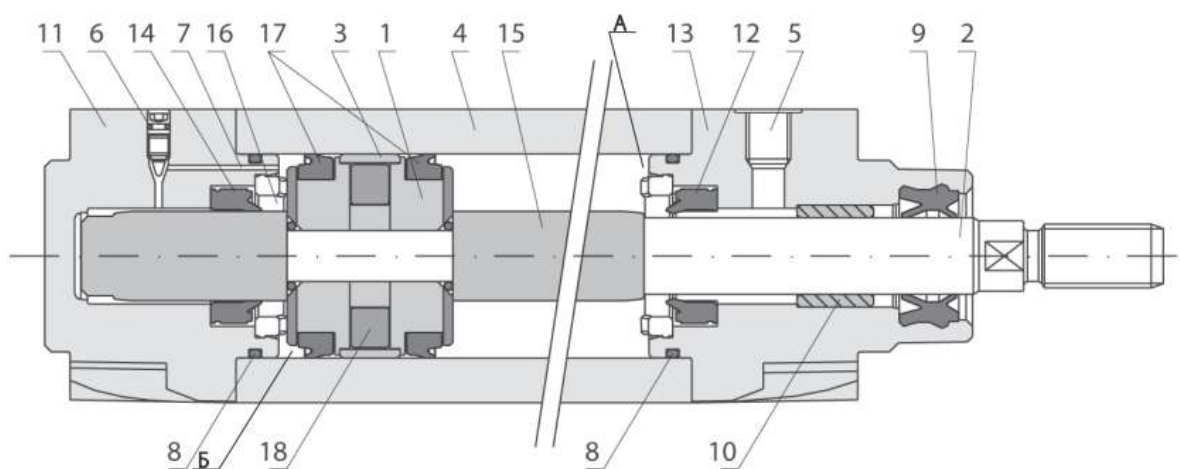
Всі кінематичні пари, які контактують у виробках Camozzi мають попередньо нанесене пастоподібне консистентне мастило. При коректній підготовці повітря і роботі обладнання його вистачає на весь життєвий цикл виробів. У разі необхідності, доцільно використовувати мастило із показником в'язкості ISO VG 32.

Кількість мастила, яке вводиться у систему, залежить від області застосування. Рекомендоване дозування – не більше трьох крапель у хвилину або трохи більше 5 крапель на 1 м³ стисненого повітря.

1.4 Пневмоциліндри

Пневматичні циліндри – найбільш поширені пневматичні виконавчі механізми. За конструкцією та робочими параметрами пневмоциліндри досить різноманітні. Прийнято розділяти їх на два типи – одно - і двосторонньої дії. В

пневмоциліндрах першого типу поршень зі штоком переміщується під дією зусилля, яке створюється тиском стисненого повітря тільки в одному напрямку. Повернення його у вихідне положення здійснюється механічно. Дані пневмоциліндри мають обмеження ходу 1...2,5 діаметра поршня. В пневмоциліндрах другого типу прямий і зворотний ходи поршня забезпечуються подачею стисненого повітря по черзі в обидві порожнини циліндра. Максимальний хід циліндрів двосторонньої дії може досягати декількох метрів. Конструкція пневмоциліндра двосторонньої дії представлена на рис. 1.31.



- 1 – поршень; 2 – шток; 3 – антифрикційне кільце; 4 – гільза; 5 – вихід;
 6 – регулюючий демпферний гвинт; 7 – демпферуючий канал;
 8 – гумові кільця; 9 – штокова манжета; 10 – направляюча втулка;
 11, 13 – кришки; 12, 14 – демпферна манжета;
 15 – втулка для гальмування; 16 – демпферна пластина;
 17 – манжети поршня; 18 – магніт

Рисунок 1.31 – Конструкція циліндру двосторонньої дії

В гільзі 4, яка закрита з двох сторін кришками 11 і 13, розташовано поршень 1, закріплений на штоку 2. Кришки стягуються шпильками. У кришках виконані різьбові отвори 5 для приєднання фітингів. У передній кришці 13 запресовано направляючу втулку 10 для штока і встановлено манжетні ущільнення 9 для герметизації штокової порожнини циліндра,

захисту її від попадання забруднень із зовнішнього середовища. Нерухоме з'єднання кришка-гільза ущільнюється гумовими кільцями 8 круглого перерізу. З'єднання поршень-гільза ущільнюється манжетами V-подібного перерізу із зустрічним розташуванням кромок. Для центрування поршня в гільзі використовується кільце із антифрикційного матеріалу 3. У більшості конструкцій пневмоциліндрів передбачено вбудовані демпферні пристрої, які дозволяють здійснювати плавну безударну зупинку в кінці ходу поршня.

1.4.1 Принцип роботи вбудованих демпферів

На основній ділянці ходу поршня повітря з порожнини протитиску (скидання) вільно надходить у вихідний отвір 5 через зазори у кришці. В кінці отвору 5 встановлено демпферну манжету 12, яка перекривається гальмівною втулкою поршня 16. У результаті чого утворюються дві камери А і Б. Камера А безпосередньо з'єднана із виходом 5 і стиснене повітря з неї виходить швидко. Камера Б з'єднана з вихлопним отвором через канал 7, який дроселюється. При подальшому русі поршня в сторону втягування, обсяг порожнини Б починає зменшуватися, відповідно в ній різко зростає тиск. Тиск у порожнині Б значно перевищує тиск в порожнині нагнітання – це є причиною інтенсивного гальмування поршня в кінці ходу. Інтенсивність гальмування поршня регулюється дроселем 6, який змінює витрати крізь канал 7 – з'єднує порожнину Б з вихлопним отвором 5.

Коли починається рух поршня відбувається наступне. Тиск подається у порожнину А, він діє на демпферну манжету 14, (яке має рифлену торцеву поверхню) і притискає її рифленим торцем до проточки. Тим самим утворюється можливість для повітря, огинаючи її, пройти із порожнини А у порожнину Б. Після того як тиск стисненого повітря в порожнинах А та Б створить на поршень 1 зусилля необхідне для його зсуву, він почне переміщуватись. У початковий момент руху (поки гальмівна втулка контактує з демпферною манжетою, стиснене повітря проходить в порожнину Б (огинаючи манжету), швидкість пересування поршня не велике. Витрати повітря у порожнину Б невеликі. Після виходу гальмівної втулки із контакту з демпферною манжетою, швидкість поршня значно збільшується, тому що витрати повітря в камері Б стають максимальними. При підході поршня до

передньої кришки та контакті гальмівної втулки 15 із демпферною манжетою 12 відбувається процес гальмування, аналогічний описаному вище.

Довжина гальмівного шляху в циліндрах двосторонньої дії лежить у межах від 10 до 50 мм. У циліндрів з малим ходом, можлива ситуація, коли поршень не покидатиме межі зони демпфірування. Тому швидкість лінійного переміщення таких циліндрів значно знижується по відношенню до циліндрів із великим ходом. Це необхідно враховувати при конструюванні та модернізації машин, які мають пневмопривід.

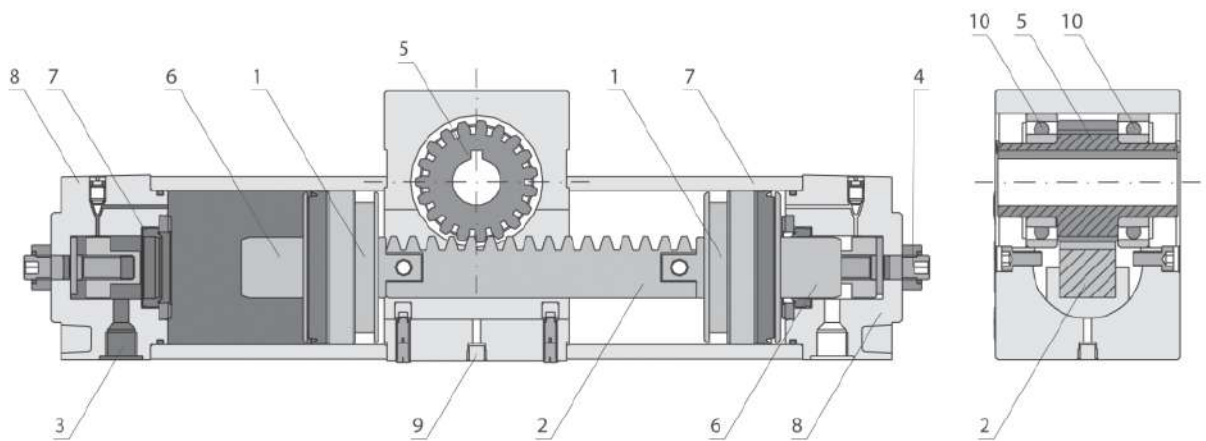
1.4.2 Поворотні пневмоциліндри

Обертальний рух вихідної ланки із обмеженим кутом повороту можна отримати за допомогою спеціальних – шибєрних (пластинчастих) пневмодвигунів, або пневмоциліндрів із лінійним переміщенням вихідної ланки у поєднанні з передавальним механізмом рейкового, важільного або гвинтового типів.

Найбільш поширеним за своєю надійністю та порівняльною простотою конструкції є привід з зубчато-рейкової передачею.

Застосовуються для перетворення зусилля, яке створюється тиском стисненого повітря під час обертального моменту на валу пневмоциліндра. Такі пневмоциліндри мають обмежений кут повороту, який лежить у межах від 0 до 360°. Типова конструкція поворотного пневмоциліндра представлена на рисунку.

У гільзі 7, закритою з двох сторін кришками 8, розташовано поршні 1, які закріплені на штоку із зубчастою рейкою 2. Кришки стягуються шпильками. У кришках 8 виконані різьбові отвори 3 для приєднання фітингів. У середній кришці на підшипниках кочення встановлено вал-шестерню 5, яка входить у зачеплення із зубчастою рейкою на штоці. Середня порожнина між поршнями не ущільнюється. Під час зборки пневмоциліндрів, у них вноситься мастило на весь термін служби. Але при роботі циліндра із великою частотою спрацьовувань необхідно внесення додаткового змащення у зубчасту пару.



- 1 – поршень; 2 – шток; 3 – різьбові отвори; 4 – гвинти для регулювання;
 5 – вал–шестерня; 6 – втулка; 7 – гільза; 8 – кришки; 9 – глушник;
 10 – кульковий радіально-упорний підшипник

Рисунок 1.32 – Конструкція поворотного циліндру

Для цього у середній кришці виконано спеціальний отвір, закритий заглушкою 9. У даній конструкції пневмоциліндра передбачені вбудовані демпферні пристрої, які дозволяють здійснити плавну зупинку поршня в кінці ходу. Їх робота аналогічна демпферам циліндра двосторонньої дії, описаного вище. Також у циліндрах даної конструкції передбачено підстроювання кута повороту у межах $\pm 5^\circ$, для цього в бічні кришки пневмоциліндра вбудовані гвинти для регулювання 4.

1.4.3 Безштокові пневмоциліндри

Дані пневмоциліндри використовуються під час великих ходів. Особливо у пристроях із обмеженими габаритами. У порівнянні із звичайними пневмоциліндрами, вони мають перевагу в габаритному розмірі по довжині, приблизно вдвічі, при тих самих робочих ходах.

Існує два різновиди подібних циліндрів: з магнітним і механічним зв'язком між поршнем і кареткою, довжина ходу таких циліндрів може досягати трьох метрів.

На рис. 1.33 представлено конструкцію безштокового пневмоциліндра з механічним зв'язком між поршнем і кареткою.

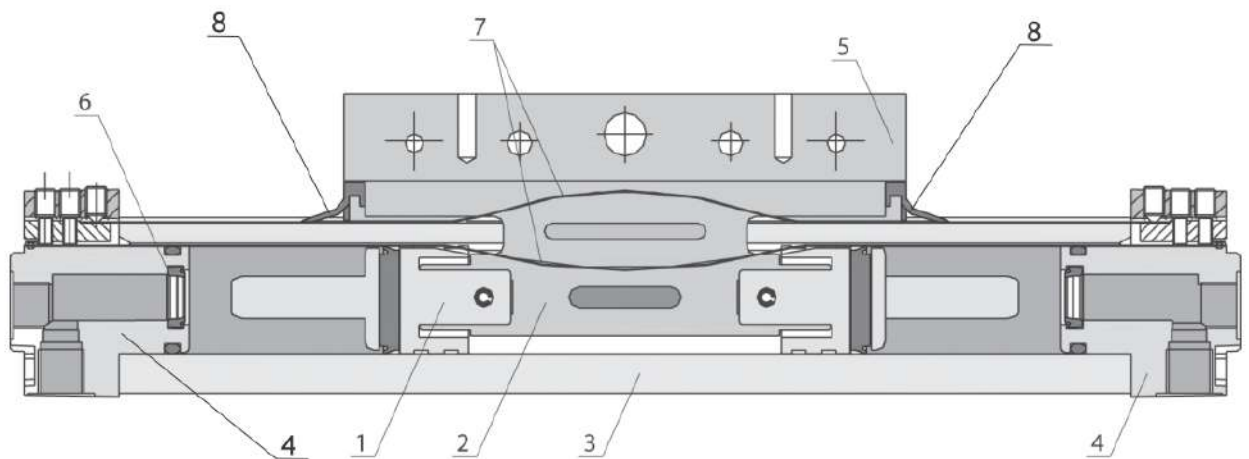


Рисунок 1.33 – Безштоковий пневмоциліндр

У корпусі циліндра 3 розташовано здвоєний поршень 1, з яким жорстко з'єднана каретка 5. Корпус циліндра має переріз по всій довжині, який закривається двома металевими стрічками 7, натягнутими уздовж корпусу і закріплених кришками 4. Нижня стрічка герметизує робочу порожнину циліндра (під робочою розуміється порожнина з надлишковим тиском). Герметизація відбувається підтисканням стрічки у циліндрі вздовж перерізу. Для можливості з'єднання каретки із здвоєним поршнем, ця стрічка пропускається під кареткою. Герметизація в цьому місці не потрібна, внаслідок застосування здвоєного поршня. Друга стрічка виконує захисну функцію – вона запобігає потраплянню пилу, вологи, забруднене з навколишнього середовища в порожнину циліндра. Для очищення верхньої стрічки від забруднень каретка оснащена спеціальними брудознімачами 8.

Для забезпечення безударної зупинки в кінці ходу поршня безштокові циліндри обладнані пристроєм демпфування. Аналогічні пристрої є і у циліндрах двосторонньої дії; поворотних пневмоциліндрах, які описано вище.

1.4.4 Пневматичні циліндри із подвоєними (потроєними) поршнями

Пневматичні циліндри із подвоєними (потроєними) поршнями, так звані тандеми і трайдеми, призначені для отримання великих притискних зусиль при обмежених габаритах пневмоциліндру. Конструкція такого циліндру є послідовне з'єднання двох циліндрів двосторонньої дії (рис. 1.34).

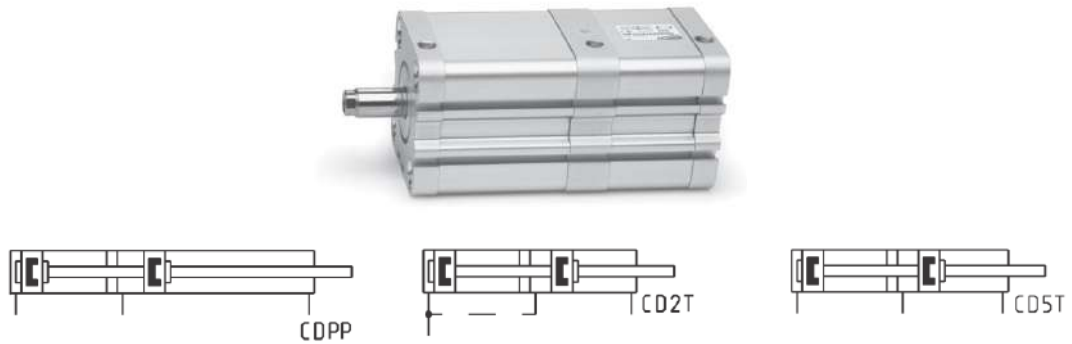


Рисунок 1.34 – Пневмоциліндри серії 32 Тандем

Дана конструкція здатна подвоїти зусилля тільки в напрямку висування штоку (рух вперед). Збільшення зусилля забезпечується поєднаннями навантажень на кожен із поршнів окремо. Повернення поршнів із штоками здійснюється шляхом подачі повітря у штокову порожнину циліндру двосторонньої дії.

1.4.5 Багатопозиційні пневмоциліндри

За своєю конструкцією та зовнішнім виглядом ці циліндри дуже подібні до циліндрів із здвоєними поршнями. Єдиними відмінностями даних циліндрів від вищеописаних є те, що хід з'єднаних послідовно циліндрів різний, обидва з'єднаних циліндра є циліндрами двосторонньої дії. Тому послідовно подаючи тиск у порожнини двопозиційного циліндра можна отримати 3 фіксовані положення штока. Умовне позначення подібних пневмоциліндрів наведено на рис. 1.35.

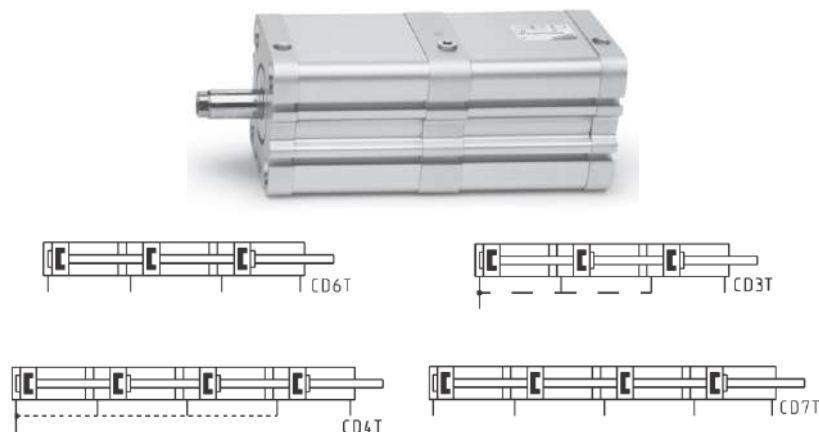


Рисунок 1.35 – Пневмоциліндри Серія 32. Багатопозиційне виконання

На рис. 1.36 зображено динаміку зміни тиску у порожнинах нагнітання і витиснення повітря в циліндрі двосторонньої дії при переміщенні поршня. У початковий момент часу, повітря подається у штокову порожнину циліндра, поршень повністю втягнуто, тиск у безштоковій порожнині дорівнює атмосферному. Після перемикавання розподільника починається подача стисненого повітря у безштокову порожнину та скидання із штокової порожнини.

Рух поршня циліндра починається тільки після того, як різниця зусиль створюваних тиском у безштоковій і штоковій порожнинах на поршень перевищить зусилля зовнішнього опору на штоці та силу тертя в ущільненнях.

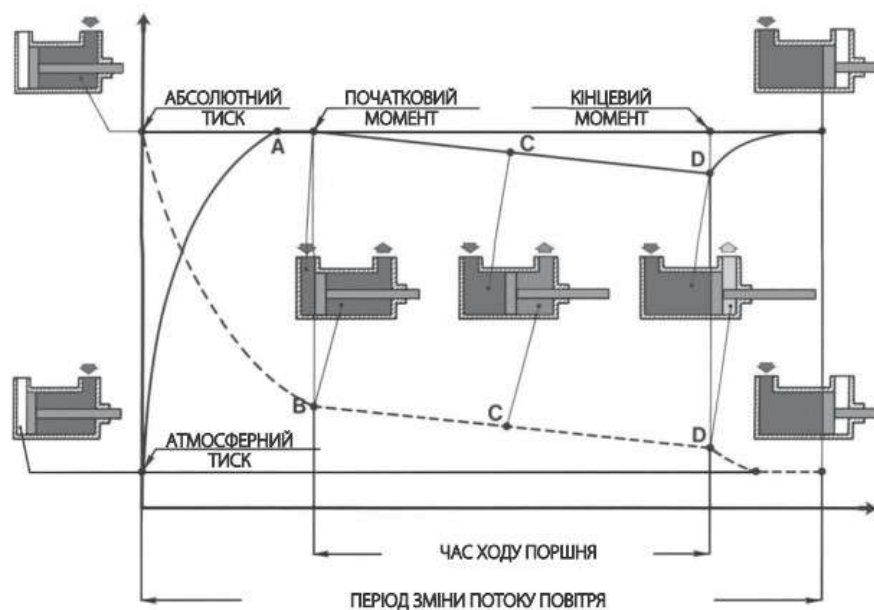


Рисунок 1.36 – Динаміка циліндра двосторонньої дії

Після початку переміщення, повітря із штокової порожнини ви-ходить швидше, ніж зменшується його об'єм, за рахунок переміщення поршня (поршень ще не розігнався). Тобто при зменшенні тиску у штоковій порожнині збільшується швидкість переміщення поршня, за рахунок збільшення різниці тисків в штоковій і безштоковій порожнинах.

Через деякий час швидкість скорочення обсягу штокової порожнини, за рахунок переміщення поршня, зрівняється зі швидкістю витікання повітря з цієї порожнини, сума всіх зусиль на поршні стане рівною нулю. Тоді встановиться постійна швидкість переміщення поршня. При попаданні

поршня циліндра в область демпфування його швидкість різко падає і він плавно підходить до передньої кришки циліндра. У штоковій порожнині встановлюється атмосферний тиск, а у безштоковій тиск, який дорівнює магістральному.

1.5 Пневморозподільники

Це пристрої, які призначені для управління потоками стисненого повітря в пневмосистемах. Вони використовуються для пуску, зупинки та зміни напрямку руху потоку повітря від джерела живлення в конкретну пневмолінію та випуску відпрацьованого повітря в атмосферу через вихлопні канали.



Рисунок 1.37 – Розподільник серії 4

За конструкцією пневморозподільники поділяються на золотникові та клапанні. Клапанні розподільники, в свою чергу, поділяються на розподільники зі сферичними та дисковими клапанами, а золотникові розподільники бувають із циліндричним, дисковим та плоским золотником.

У клапанних розподільниках прохідні перерізи відкриваються і закриваються за допомогою сферичних, плоских і конічних затворів, сідло яких зазвичай забезпечене пластмасовими або гумовими ущільнювачами. Такі розподільники мають просту конструкцію, не мають деталей, які швидко зношуються, тому довговічні, міцні та стійкі до повітря з поганою підготовкою. До недоліків розподільників даної конструкції відносяться: велике зусилля перемикання золотника та малі витрати стисненого повітря.

Золотникові розподільники, у порівнянні із клапанними, потенційно здатні забезпечити більші витрати повітря, завдяки більшій площі прохідного перерізу. Ущільнення між каналами у золотникових розподільниках забезпечується або за рахунок дуже маленьких зазорів між самим золотником і корпусом, або за рахунок установки у корпусі (або на золотнику) кільця круглого перерізу (манжет). Як найкраще зарекомендували себе золотникові розподільники із ущільнюючими кільцями, встановленими у корпусі розподільника. Такі розподільники менш вибагливі до якості повітря, більш зносостійкі та ремонтпридатні.

За видом керуючого впливу, розподільники поділяються на: розподільники із ручним, механічним, електричним, пневматичним або електропневматичним керуванням.

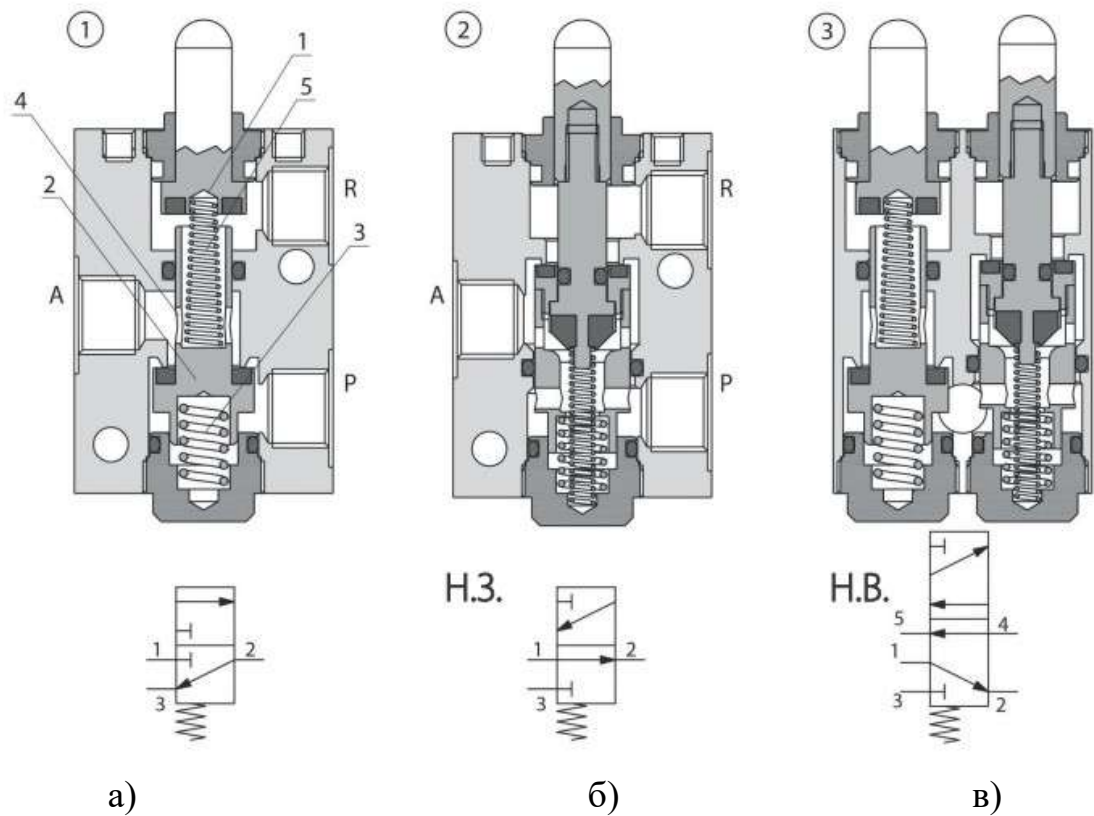
Керування може бути одно- і двостороннім. У першому випадку зовнішній керуючий вплив викликає перемикання розподільного елемента тільки в одному напрямку. Після зняття впливу, зворотне перемикання відбувається під дією механічної або пневматичної пружини. У другому випадку для прямого і зворотного перемикань розподільного елемента, потрібно змінювати напрямок керуючого впливу, тобто потрібно два протилежних за напрямком керуючих сигнали.

Залежно від числа фіксованих положень розподільного елемента, розрізняють дво-, три- і багатопозиційні пневморозподільники. У більшості випадків у пневматичних приводах використовують двопозиційні розподільники, що відповідає дискретному характеру роботи пневмоциліндрів та інших пристроїв. За кількістю зовнішніх ліній, пов'язаних із розподільником, визначається його лінійність. При цьому і виходи в атмосферу розглядаються як зовнішні лінії. У пневматичних приводах найчастіше застосовуються дво-, три- і п'ятилінійні розподільники, рідше – чотирилінійні.

Робота трилінійного двопозиційного нормально закритого розподільника полягає у наступному. Якщо немає механічного впливу на плунжер 1, клапан 2 притиснутий до сидла пружиною 3 і тиском стисненого повітря. Вихід А при цьому з'єднано з атмосферою через канал 4 у плунжері 2 та отвором R. При натисканні на поршень 1, спочатку стискається пружина 5,

яка має меншу твердість, плунжер 1 перекриває канал 4 і відсікає вихід А від атмосфери, далі стискається більш жорстка пружина 3, клапан 2 відривається від сідла і стиснене повітря з входу Р надходить на вихід А.

Трилінійний двопозиційний нормально відкритий розподільник працює аналогічно нормально закритому. Єдина відмінність в тому, що при механічному впливі на плунжер вихід А відсікається від входу Р і з'єднується з атмосферою. На рис. 1.38 зображено декілька умовних позначень розподільників.



Отвір Р – підведення стисненого повітря; отвір А – вихід до споживача;
отвір R – вихід в атмосферу

Рисунок 1.38 – Конструкція розподільників клапанного типу: а) трилінійний двопозиційний нормально закритий розподільник; б) трилінійний двопозиційний нормально відкритий розподільник; в) п'ятилінійний двопозиційний розподільник



Рисунок 1.39 – Розподільник серія А

На рис. 1.40 зображено нормально закритий розподільник, клапанного типу із безпосереднім електромагнітним керуванням. При відсутності керуючої напруги на котушці (соленоїді) розподільника, якір 3 підтиснено пружиною 5 до сідла у корпусі розподільника. При цьому отвір, який пов'язаний із споживачем (А) з'єднано з атмосферою через проточки 4 в якорі 3 у вихідний отвір R. З подачею напруги на котушку розподільника, утворюється магнітне поле, яке підштовхує золотник у корпус котушки. При переміщенні золотника вихідний отвір R перекривається підпружиненим клапаном 2, який підтискається до виходу пружиною під тиском повітря. Підведення стисненого повітря отвору Р з'єднується із виходом до споживача А.

Клапанні розподільники із безпосереднім електромагнітним керуванням найчастіше використовуються при необхідності забезпечення невеликих витрат стисненого повітря (до 100 Нл/хв) або в якості пілотних розподільників (розподільники, які використовуються для управління золотниковими розподільниками).

Розподільники клапанного типу з невеликими витратами стисненого повітря, частіше припливного виконання, які використовуються для управління золотниковими або клапанними розподільниками із великими витратами стисненого повітря, які вимагають порівняно великих зусиль перемикання і мають великий хід запірного елемента. Керуючі (пілотні) розподільники, виконуються в одному корпусі з основним розподільником і можуть мати як спільне з основним розподільником, так і роздільне підведення повітря.

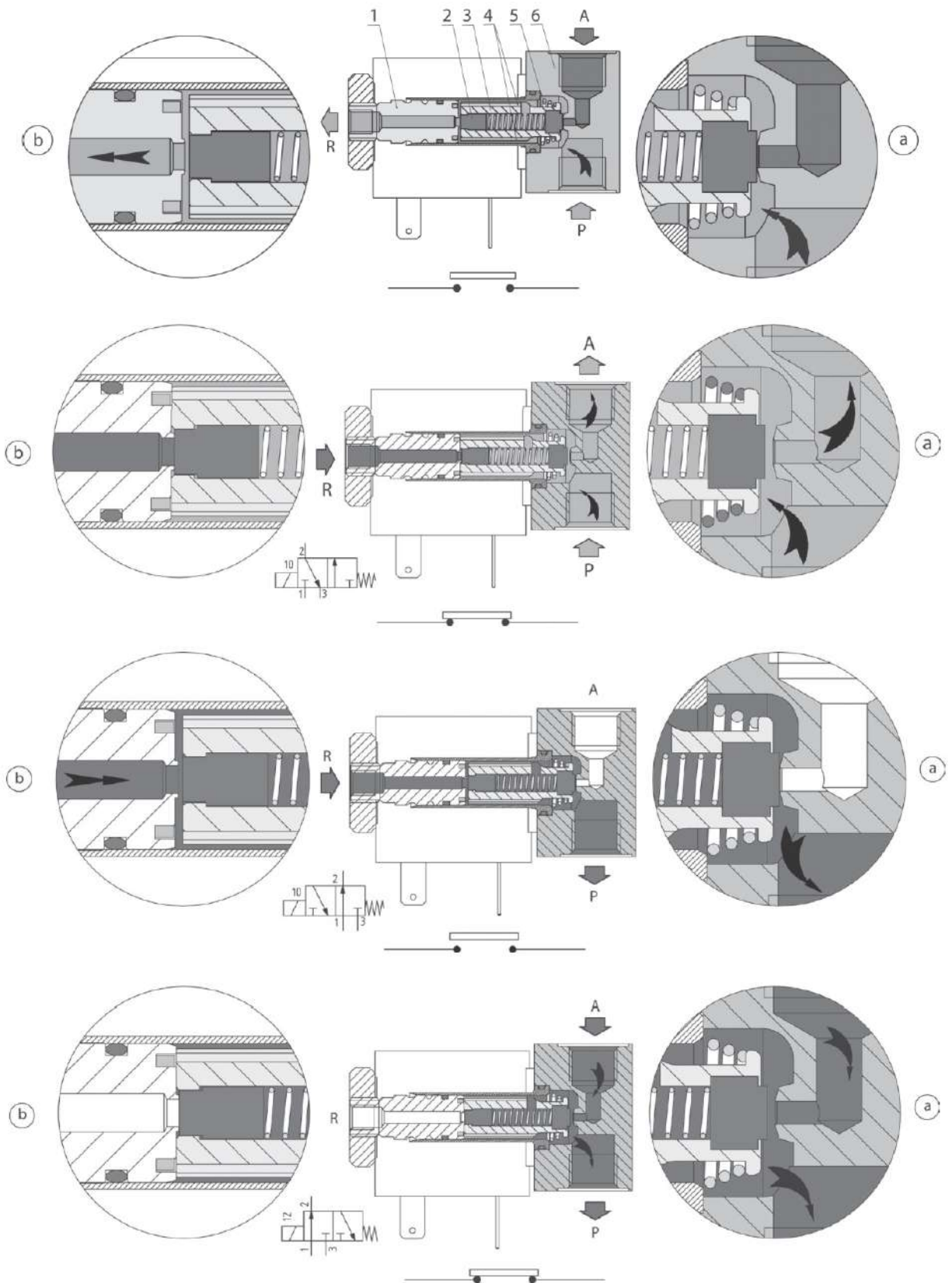


Рисунок 1.40 – Трилінійний двопозиційний розподільник із електромагнітним керуванням

На рисунку 1.41 зображені розподільники наступних типів:

а) з двостороннім електропневматичним управлінням (загальне підведення повітря). Особливості експлуатації: високий тиск (тиск живлення) повинен підводитись до входу Р – тільки цей вхід з'єднано зі пілотними розподільниками. Для стабільного перемикання дані розподільники необхідно використовувати на тисках у межах 0,4...1,4 бар. Це обумовлено мінімальним зусиллям необхідним для подолання тертя між золотником і кільцями ущільнювачів;

б) з одностороннім електропневматичним управлінням і пружинним поверненням. Особливості експлуатації: підведення стисненого повітря, як і у попередньому розподільнику, обов'язково здійснюється в порт Р. Мінімальний робочий тиск для розподільників даного способу управління становить 2,5 бар. Це обумовлено необхідністю долати тертя в ущільненнях і зусилля пружини;

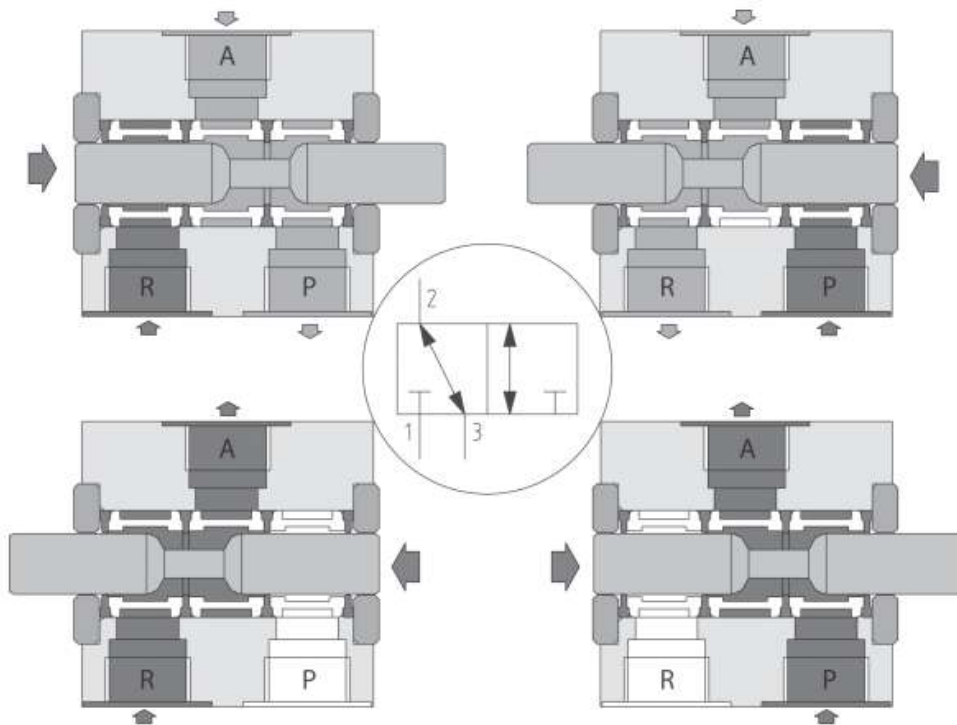
в) з одностороннім електропневматичним управлінням і пневматичним поверненням. Особливості експлуатації: підведення стисненого повітря, як і у всіх попередніх розподільниках, обов'язково здійснюється у порт Р. Мінімальний робочий тиск для розподільників даного способу управління становить 1,5 бар, це обумовлено необхідністю долати тертя в ущільненнях і опір пневматичної пружини;

г) з одностороннім електропневматичним управлінням і зовнішнім пневматичним поверненням. Особливості експлуатації: підведення стисненого повітря, як і у всіх попередніх розподільниках, обов'язково здійснюється у порт Р. Мінімальний робочий тиск для розподільників даного способу управління становить 1 бар, це обумовлено необхідністю долати тертя в ущільненнях.

Тобто всі золотникові розподільники з електропневматичним управлінням (із спільним підведенням повітря) можна використовувати тільки для роботи на надлишковому тиску.

При роздільному підводі повітря до пілота і основного розподільника, можливо використовувати основний розподільник на подачу вакууму, при підводі до пілота високого тиску.

На рис. 1.43 представлено трилінійний двопозиційний золотниковий розподільник із ущільнюючими кільцями, встановленими у корпусі.



Отвір Р – підведення стисненого повітря; отвори А – вихід до споживача;
отвори R – вихід в атмосферу

Рисунок 1.43 – Трилінійний двопозиційний золотниковий розподільник

Золотник в розподільнику даної конструкції може займати тільки два фіксованих положення в одному з яких отвір А зв'язано із атмосферою і друге положення при якому вихідний отвір А зв'язано із підведенням стисненого повітря. Однією із переваг розподільників даної конструкції також є те, що симетрична форма золотника дає можливість тиску стисненого повітря, яке підводиться, не створювати на нього дію. Тобто для утримання золотника в фіксованому положенні не потребує докладання постійно діючого зусилля (на відміну від розподільників клапанного типу). Для перемикання цього розподільника, до золотника потрібно докласти зусилля щоб подолати тертя в ущільненнях між золотником і направляючими втулками. Це зусилля буде діяти тільки на час переміщення золотника з одного крайнього положення в інше.

Аналогічну конструкцію мають трипозиційні золотникові розподільники, відмінністю яких є наявність механізму утримання золотника в середньому положенні (див. рис. 1.44).

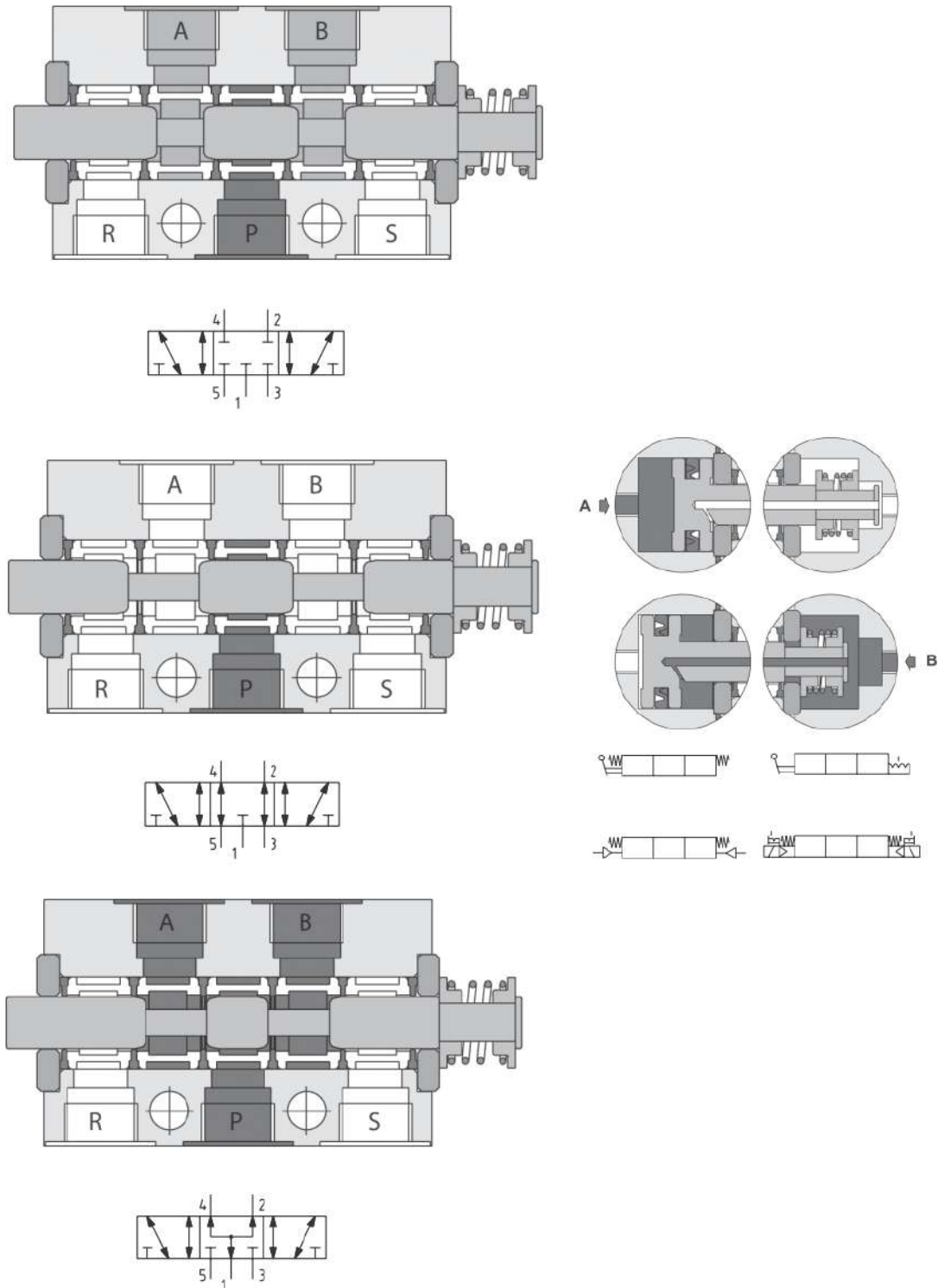


Рисунок 1.44 – П'ятилінійний трипозиційний золотниковий розподільник

За виконавчими функціями трипозиційні розподільники діляться на:

- розподільники із закритим центром – при відсутності керуючих сигналів обидва виходи до споживача перекриті, тобто повітря в них не надходить, та з них не виходить;

- розподільники із відкритим центром – при відсутності керуючих сигналів обидва виходи А і В з'єднані з атмосферою;

- розподільники із подачею повітря в обидві лінії – при відсутності керуючих сигналів на два виходи А і В подається стиснене повітря із магістралі Р.

Трипозиційні розподільники бувають із ручним, двостороннім електропневматичним і пневматичним управлінням.

1.5.1 Пневмологіка

Більшість систем, які застосовують пневмопривід – це системи дискретного типу. Тобто такі системи, які мають фіксовану кількість точок положення вихідних ланок.

Для управління дискретними системами, простіше і доцільніше використовувати пневматичні логічні елементи замість електричних або електронних. Особливо у тих випадках, коли застосування електрики небажано або неможливо.

За принципом роботи пневматичні логічні елементи не відрізняються від їх електронних аналогів.

Рівні логічних пневматичних сигналів для елементів реалізують функції однієї змінної. На вході у пневматичний логічний елемент тиск стисненого повітря нижче 0,3 бар. Це сприймається як логічний нуль. Вище 0,3 бар – відповідно як логічна одиниця. На виході пневматичного логічного елемента нуль представлений атмосферним тиском, а одиниця тиском живлення логічного елемента.

Рівні логічних пневматичних сигналів для елементів реалізують функції двох змінних. На вході у пневматичний логічний елемент атмосферний тиск повітря сприймається як логічний нуль, а будь-який тиск вище атмосферного сприймається як логічна одиниця. На виході пневматичного логічного

елемента нуль представлений атмосферним тиском, а одиниця тиском на одному із входів логічного елемента.



Рисунок 1.45 – Основні елементи пневмологіки

Логічні елементи, які реалізують функцію однієї змінної:

- Функція ТАК (Мод. 2LS-SB4-B) повторення.

X	1	0
F	1	0

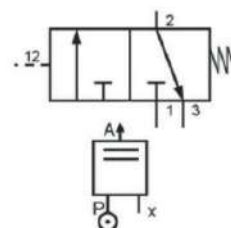


Рисунок 1.46 – Таблиця істинності та схематичне позначення функції ТАК

На виході логічного елемента ТАК з'являється високий тиск (що дорівнює тиску живлення) при тиску на вході більше 0,6 бар, і навпаки – при тиску на вході менше 0,6 бар вихід цього логічного елемента з'єднується з атмосферою.

Основним призначенням цього елемента є функція посилення вхідного сигналу (його повторення). Тобто якщо на виході, будь-якої схеми є пневматичний сигнал від будь-якого пристрою, але величини тиску цього вихідного сигналу недостатньо для перемикання цього пристрою, то в цьому випадку доцільно використовувати елемент ТАК, для посилення керуючого сигналу.

- Функція НІ (Мод. 2L T-SB4-B) заперечення.

На виході логічного елемента НІ з'являється високий тиск (що дорівнює тиску живлення) при тиску на вході 0,3 бар і менше, і навпаки – при тиску на вході більше 0,3 бар вихід цього логічного елемента з'єднується з атмосферою.

Основне призначення – інвертування сигналу. Даний логічний елемент використовується в заданих схемах, схемах автоколивань та інших лінійних і нелінійних схемах.

X	1	0
F	0	1

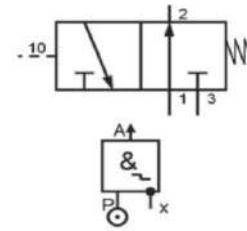


Рисунок 1.47 – Таблиця істинності та схематичне позначення функції НІ

Логічні елементи, що реалізують функцію двох змінних:

- Функція АБО (Мод. 2LR-SB4-B).

X1	1	0	1	0
X2	1	1	0	0
F	1	1	1	0

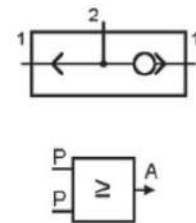


Рисунок 1.48 – Таблиця істинності та схематичне позначення функції АБО

Стиснене повітря з будь-якого з входів може пройти на вихід. При відсутності надлишкового тиску на входах, вихід цього елемента з'єднується із атмосферою.

Основне призначення – передача двох керуючих сигналів на один вихід. Даний логічний елемент використовується для установки в системах з автоматичним управлінням і ручним дублюванням, для об'єднання сигналів від автоматичних та дублюючих пристроїв і видачі їх на об'єкт управління.

- Функція І (Мод. 2LD-SB4-B).

X1	1	0	1	0
X2	1	1	0	0
F	1	0	0	0

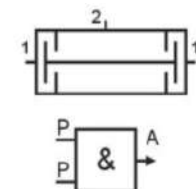


Рисунок 1.49 – Таблиця істинності та схематичне позначення функції І

Стиснене повітря з будь-якого із указаних в таблиці входів може пройти на вихід, тільки в тому випадку, якщо присутні обидва вхідних сигнали. При відсутності надлишкового тиску на будь-якому з входів, вихід цього елемента з'єднується з атмосферою.

Основне призначення – передача сигналу на вихід при присутності на вході одночасно основного і підтверджуючого сигналів.

1.6 Контрольні запитання та завдання

1. Які основні параметри стисненого повітря?
2. Яким чином розраховується пропускна здатність системи?
3. Як залежить динамічна та кінематична в'язкість повітря від температури повітря?
4. Як виробляється стиснене повітря?
5. Наведіть схему виробництва і підготовки стисненого повітря
6. Що таке компресор?
7. Які основні види компресорів ви знаєте?
8. Що таке ресивер?
9. Які існують компоненти блоку підготовки повітря?
10. Як працює регулятор тиску?
11. Поясніть конструкцію маслорозпилювача.
12. Поясніть конструкцію пневмоциліндра?
13. Які існують види пневмоциліндрів?
14. Поясніть конструкцію пневморозподільника.

2 ХАРАКТЕРИСТИКА ПНЕВМАТИЧНИХ ЕЛЕМЕНТІВ, РОЗРОБКА ПНЕВМАТИЧНИХ СХЕМ ТА ЕЛЕКТРИЧНИХ СХЕМ

2.1 Інтегровані рішення функціональних модулів в задачах синтезу ежекторів та пневматичних приводів

2.1.1 Приклади використання ежекторів в промисловому обладнанні

В технологічних лініях багатьох галузей промисловості під час виконання допоміжних технологічних операцій широко застосовуються пневмосоплові пристрої із системами ежекторів. Прикладами застосування є операції оброблення поверхонь матеріалів оброблення, споживчої упаковки, введення газомодифікованого середовища (ГМС) всередину упаковки, розділення потоку об'єктів на конвеєрі, створення вакууму, очищення клапанів картонної упаковки перед нанесенням клейового розчину, очищення деревообробної ділянки тощо. В таких системах найчастіше використовують пневматичні приводи у комбінуванні із вакуумними системами, а також електропневматичні прецизійні системи контролю за поточними операціями технологічного процесу.

Найбільш розповсюдженими конструкціями сьогодні є ежектори з L- та H-подібною компоновкою, із сопловим перерізом у межах 0,5; 0,7; 1; 2; 2,5; 3 мм. Приклади використання ежекторів у складі пневматичного приводу наведено на рис. 2.1. Робота вакуумних ежекторів заснована на принципі Вентурі, і для створення вакууму використовують стиснене повітря.

2.1.2. Опис конструкцій вакуумних ежекторів

Базові ежектори без рухомих частин, робота яких заснована на принципі Вентурі. Виконання «L» зі збільшеною продуктивністю. Виконання «H» для створення більш глибокого вакууму (85...90%).

Принцип Вентурі – це ефект, в основі якого лежить падіння тиску, коли потік рідини чи газу протікає через звужену частину труби. Цей ефект названий в честь італійського фізика Джованні Вентурі (рис. 2.2).

Газовий ежектор відноситься до струйних пристроїв, в яких повний тиск газового потоку підсилюється під дією тиску іншого, більш високонапірного

поток, забезпечує роботу соплових систем технологічних ліній в харчовій промисловості. Передача енергії від потоку повітря до речовини розпилювання, відбувається шляхом турбулентного змішування.

Введення газомодифікованого середовища



Дозування рідкого азоту для видалення кисню



Обробка поверхні споживчої упаковки перед маркуванням



Пневматичний сопловий зштовхувач



Формування потоку споживчої упаковки



Нанесення цукрової суміші (глазурування)



Рисунок 2.1 – Приклади використання ежекторів у функціональних технологічних пристроях

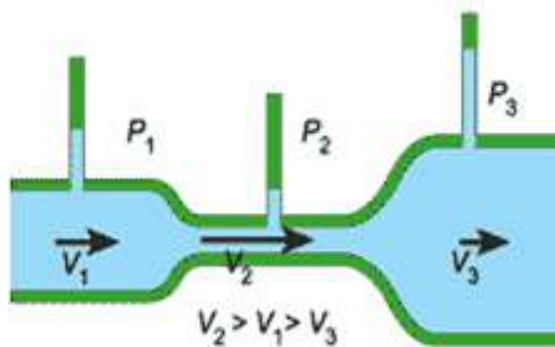
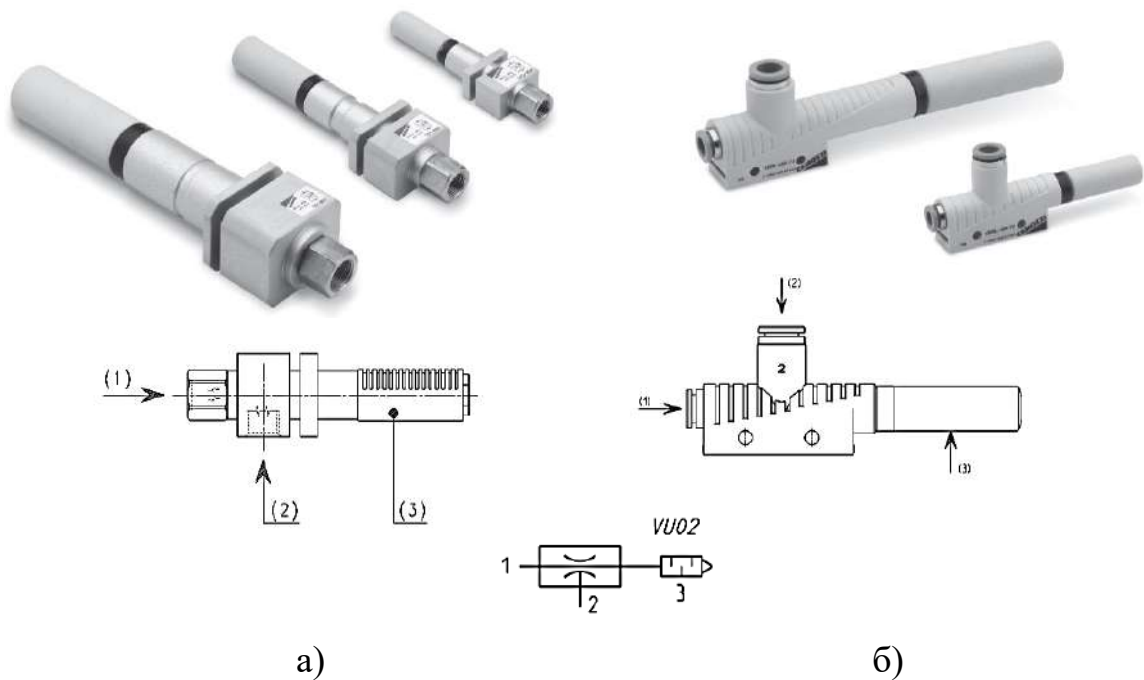


Рисунок 2.2 – Ефект інжекції потоку (ефект Вентурі) – прискорення ламінарного повітряного потоку при проходженні через плавне звуження, що призводить до зниження тиску в зоні звуження

Ефект Вентурі є наслідком закону Бернуллі. Підвищення тиску ежектуемого потоку без безпосередньої витрати механічної енергії є основною перевагою ежектора, порівняно із вакуумними насосами. Завдяки простоті конструкції, у порівнянні з механічними нагнітачами, газові ежектори знайшли широке застосування у технологічних лініях для харчових продуктів. Ежектор може використовуватися в ролі насоса, компресора, екстаустера для створення зниженого тиску в деякому обсязі, для інших цілей. В залежності від призначення ежектори мають різні конструктивні особливості.

Приклад вакуумних ежекторів серії VEB, VEBL Camozzi наведено на рис. 2.3, а серії VED, VEDL Camozzi – на рис. 2.4. Особливістю конструкцій є по різному сформовані вихлопні канали, які обумовлюють умови монтажу та особливості роботи вакуумних ежекторів при створенні ежекторів.

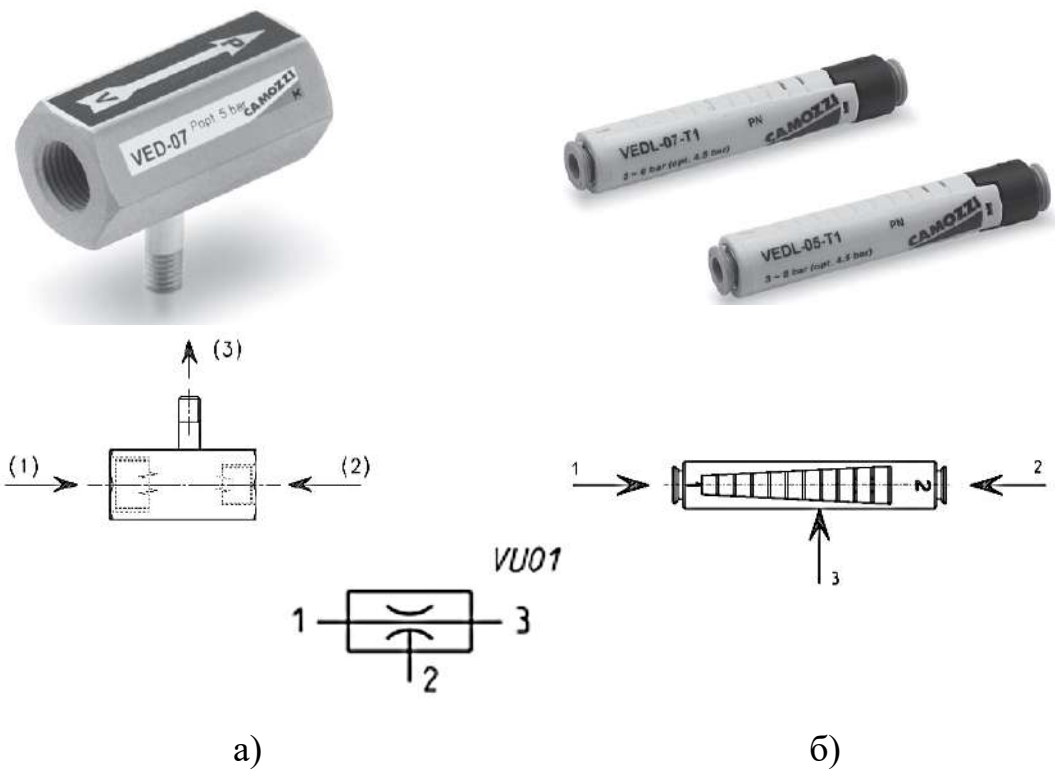


1 – вхід стисненого повітря; 2 – вхід вакууму; 3 – глушник

Рисунок 2.3 – Загальний вигляд вакуумних ежекторів з Н, L- подібною конструкцією: а) корпус виконано із алюмінію;
б) корпус виконано із композиційного полімеру

Технічні характеристики ежектора VEB-05H:

- діаметр сопла (місце звуження) 0,5 мм;
- глибина вакууму 82%;
- максимальна продуктивність 170 л/хв;
- витрати повітря 186 Нл/хв;
- робочий тиск в пневмережі на вході в ежектор 5 бар.



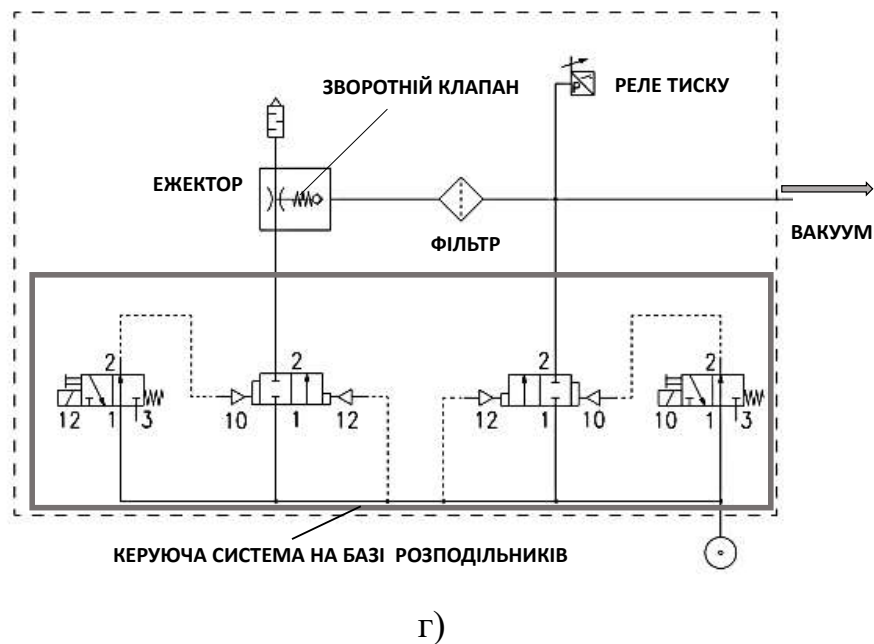
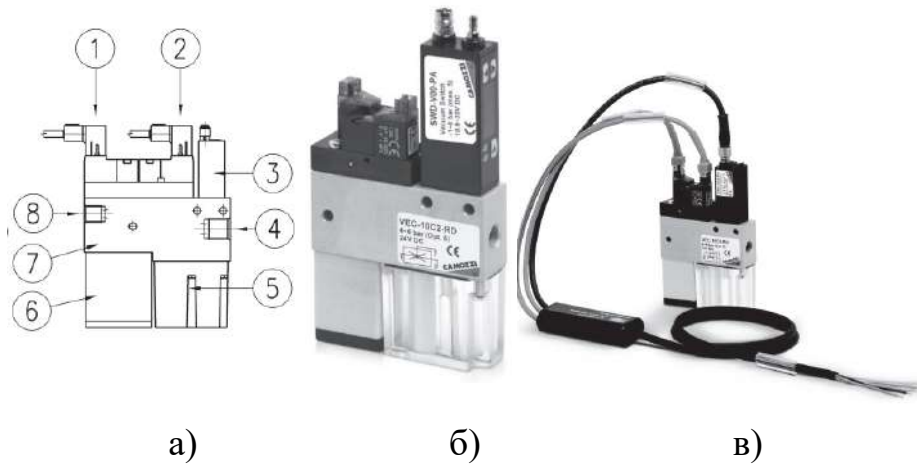
1 – вхід стисненого повітря; 2 – вхід вакууму; 3 – глушник

Рисунок 2.4 – Загальний вигляд вакуумних ежекторів VED з лінійною конструкцією: а) корпус виконано із алюмінію; б) корпус виконано із композиційного полімеру

Технічні характеристики ежектора VED-07:

- діаметр сопла (місце звуження) 0,7 мм;
- глибина вакууму 90%;
- максимальна продуктивність 170 л/хв;
- витрати повітря 14 Нл/хв;
- робочий тиск в пневмережі на вході в ежектор 5 бар.

Вакуумні генератори з вбудованими розподільниками, які дають можливість вакуумування і скидання тиску із вбудованим вакуумним реле називають компактні вакуумні ежектори (рис. 2.5). Така конструкція не вимагає зовнішніх розподільників.



- 1 - клапан вакуумування; 2 - клапан відпускання; 3 - реле вакууму;
 4 - вхід для вакууму; 5 - фільтр; 6 - глушник ; 7 – корпус;
 8 - підведення стисненого повітря

Рисунок 2.5 – Компактні вакуумні ежектори: а) загальна схема;
 б) загальний вигляд моделі VEC Camozzi;
 в) компоновка із роз'ємами і кабелями; г) принципова схема ежектора

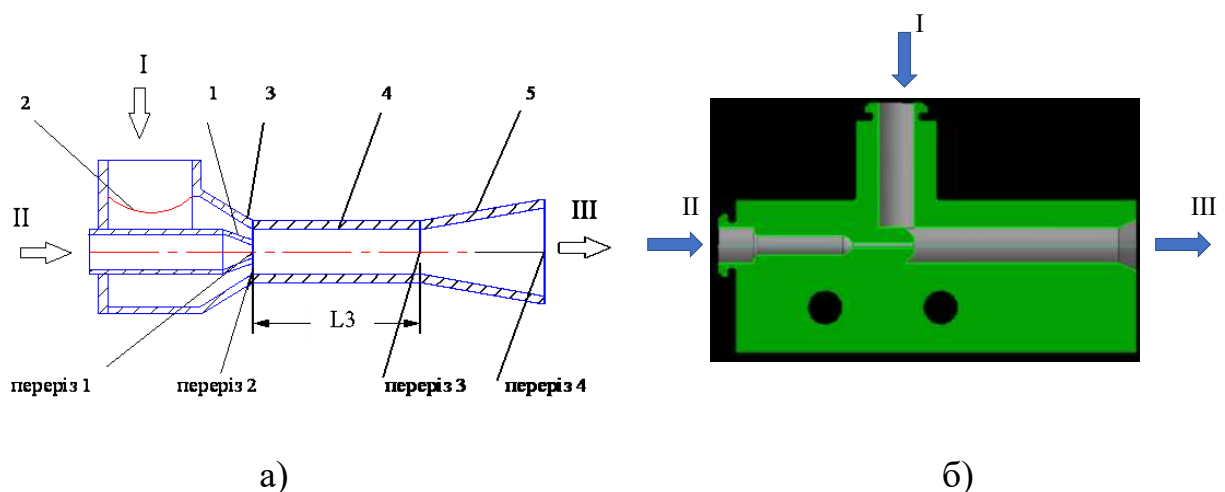
Зазвичай використовуються в повністю автоматизованих транспортних системах. Автоматична система економії стисненого повітря є невід'ємною частиною компактних вакуумних ежекторів. Під час захоплення об'єкту і вакуумування, ежектор залишається включеним до тих пір, поки не буде досягнута необхідна глибина вакууму. При досягненні необхідного вакууму ежектор вимикається. Ежектор залишається вимкненим, поки величина вакууму не зменшиться на величину гістерезису реле (в цей час не витрачається стиснене повітря). Коли глибина вакууму зменшується до зазначеного значення ежектор включається, поки знову не буде досягнута необхідна глибина вакууму. Технічні характеристики ежектора VEC-25:

- діаметр сопла (місце звуження) 2,5 мм;
- глибина вакууму 85%;
- максимальна продуктивність 161 л/хв;
- витрати повітря 310 Нл/хв;
- робочий тиск в пневмережі на вході в ежектор 5...6 бар;
- робочий температурний діапазон 0...45 °С.

Найбільш розповсюджена в технологічних машинах є геометрія L-подібного ежектора. Принципова схема дослідного L-подібного ежектора подана на рис. 2.6.

Він складається із активного сопла 1, яке подає робоче середовище до приймальної камери 2 (для підведення потоку ежекції), пасивного сопла 3 для підведення повітря, камери змішування 4, дифузора 5, призначеного для підвищення статичного тиску. В деяких компоновках дифузор або пасивне сопло можуть бути відсутні. Перерізи 1, 2, 3, 4 є основними зонами дослідження тиску p , температури T , швидкості потоку ω , масових витрат G , σ коефіцієнтів відновлення тиску.

Зупинимося на описі моделі роботи ежектора докладніше. Робочий процес ежектора у пакувальній машині для оброблення упаковок зводиться до наступного. Високо напірний (робочий) газ, що має повний тиск p^* і температуру гальмування T^* – підводиться крізь сопло 1 (рис. 2.6) у змішувальну камеру 4. У вхідному перерізі камери змішувача встановлюється статичний тиск p_2 , який нижче повного тиску низьконапірного газу p_2^* ежекції.



I – канал підведення робочого середовища в сопловий переріз;
 II – вхідний напірний канал; III – вивідний (вихлопний) канал

Рисунок 2.6 – Загальний вигляд конструкції L-подібних ежекторів соплових систем у технологічних лініях:

- а) класична модель із основними зонами (перерізами) зміни тиску;
- б) 3D розроблена модель із змінними прохідними перерізами

Під дією різниці тисків низьконапірний газ (повітря) спрямовується у камеру змішування. В кінці камери 4 після завершення процесу змішування газ має усереднені параметри p_3 , T_3 , ω_2 .

На виході дифузору статичний тиск газу p_4 більше статичного тиску в кінці камери змішування p_3 . Якщо позначити масову витрату робочого потоку G_1 , а масові витрати газу ежекції G_2 , то співвідношення $n = G_2/G_1$ є коефіцієнтом ежекції. Разом з цим відношення статичного тиску робочого газу перед соплом до статичного тиску газу ежекції p_1/p_2 буде ступенем розширення.

2.1.3 Розрахунок основних технологічних характеристик ежектора

Відношення статичного тиску на виході з дифузора до статичного тиску газу ежекції p_4/p_2 визначає ступінь стиснення, або ступінь підвищення тиску.

При моделюванні у просторі ПП FlowVision, основним геометричним параметром ежектора є відношення площин вихідних соплових перетинів для робочого потоку і потоку ежекції:

$$\alpha = \frac{F_1}{F_1} = \frac{F_1}{F_3 - F_1} \quad (2.1)$$

де F_1, F_2 – відповідно площа вихідних соплових перерізів для підведення робочого потоку і потоку ежекції;

F_3 – площа перетину циліндричної камери змішування (переріз 3).

Ежектор з великим значенням α вважається високо напірним, але має малий коефіцієнт ежекції. І навпаки, ежектор з малим α дозволяє отримати великий коефіцієнт ежекції, але характеризується малим напором.

Другий геометричний параметр при імітаційному моделюванні ежектора – відносна довжина камери змішування L_1/d_3 – також впливає на динамічні характеристики ежектора.

Третій геометричний параметр моделі ежектора – ступінь розширення дифузора $f = F_4/F_3$ – приймалась як відношення площі перерізу на виході з дифузора до площини на вході в нього.

Припущенням дослідної моделі Flow Vision приймалось що ежектор працює за заданим статичним тиском на виході із дифузора (наприклад, при виході в атмосферу, або всередину споживчої упаковки із постійним тиском). Ступінь розширення дифузору f , розглядався як параметр який суттєво впливає на режими роботи ежектора. Для проведення моделювання розроблено чотири 3D моделі ежектора з різними характеристиками прохідного робочого сопла (рис. 2.7).

При моделюванні ежектора також прийняті припущення про одномірність потоку. Закон збереження маси має вигляд:

$$G_3 = G_1 + G_2, \text{ тоді } G_3/G_1 = 1 + n \quad (2.2)$$

На основі закону збереження енергії можна записати:

$$G_3 \left(c_3 T_3 + \frac{w_3^2}{2} \right) = G_1 \left(c_1 T_1 + \frac{w_1^2}{2} \right) + G_2 \left(c_2 T_2 + \frac{w_2^2}{2} \right) + Q \quad (2.3)$$

де Q – загальна кількість тепла, яке підводиться до повітря (газу) в секунду через стінку змішувальної камери.

При розрахунках лінійних ежекторів приймається, що $Q = 0$ і гази, які змішуються, однорідні і мають однакову теплоємність.

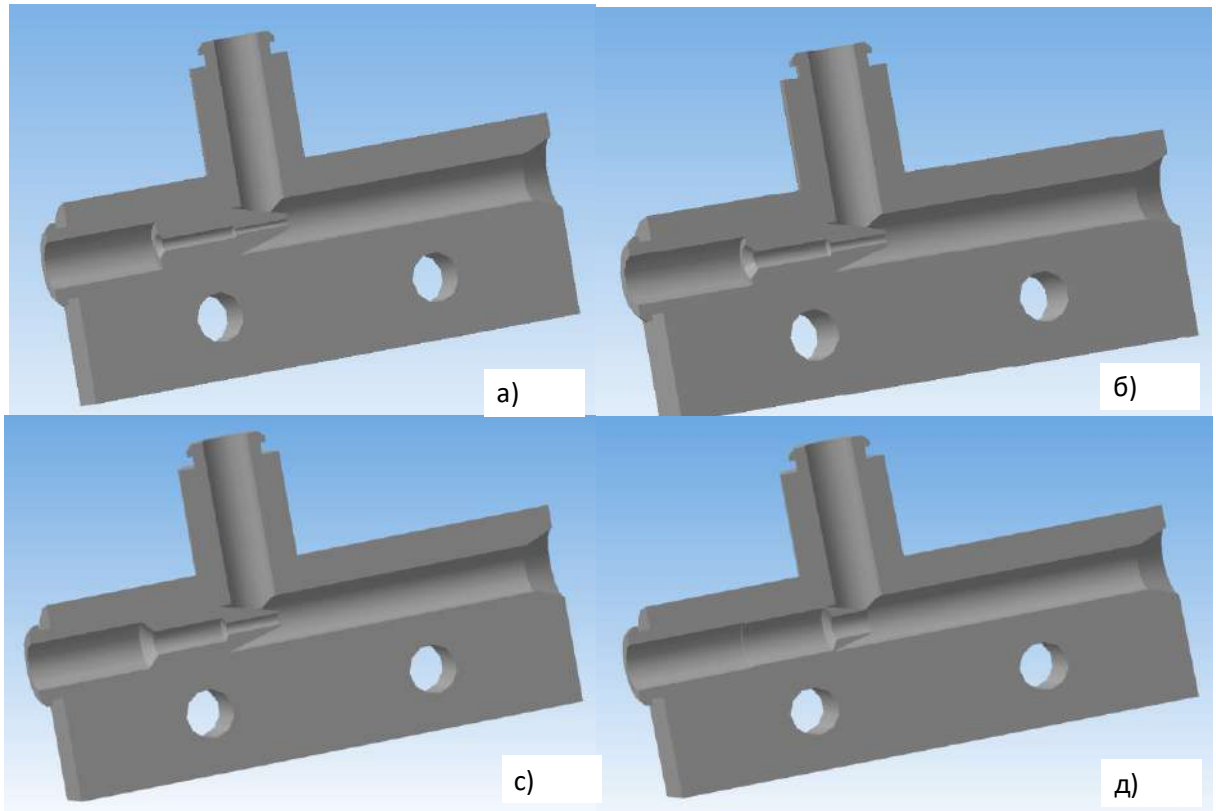


Рисунок 2.7 – Переріз імітаційних 3D моделей L-подібного ежектора:

а) діаметр сопла 0,5 мм; б) діаметр сопла 0,7 мм;

в) діаметр сопла 1 мм; г) діаметр сопла 2 мм

Припустимо, що

$$(n + 1) \frac{T_3^*}{T_1^*} = 1 + n \frac{T_2^*}{T_1^*} \quad (2.4)$$

Введені співвідношення $\Theta = \frac{T_2^*}{T_1^*}$, дозволили отримати зручні формули для розрахунку:

$$\frac{T_3^*}{T_1^*} = \frac{n\Theta + 1}{n + 1} \quad (2.5)$$

$$\frac{a_{k3}}{a_{k2}} = \sqrt{\frac{n\Theta + 1}{n + 1}} \quad (2.6)$$

Витрати повітря (газу) в будь-якому перетині сопла розраховувались за формулою:

$$G = m\sigma_1 \frac{p^* F q(\lambda)}{\sqrt{T^*}} \quad (2.7)$$

де m знаходиться за формулою і для повітря $m = 0,0404 \text{ м}^{-1} \cdot \text{с} \cdot \text{К}^{0,5}$; σ_1 – коефіцієнт відновлення тиску в активному соплі.

Визначення масових витрат у рівнянні нерозривності потоку за допомогою формули (2.7), дають можливість отримати формулу для зв'язку параметрів ежектора в перетинах ежекції 1 та 3 :

$$\frac{p_3^*}{p^*} = \sigma_1 \frac{\sqrt{\left(1+n\Theta\frac{c_2}{c_1}\right)(n+1)\frac{c_2}{c_1}}}{F_3/F_1 \cdot q(\lambda_3)} q(\lambda_1) \quad (2.8)$$

У випадку звукового або надзвукового режимах роботи ежектора приймаємо: $q(\lambda_1) = 1$. Рівняння руху робочого середовища має вигляд:

$$G_3\omega_3 + p_3F_3 = G_1\omega_1 + p_1F_1 + G_2\omega_2 + p_2F_2 - P_{mp} \quad (2.9)$$

де $G\omega + pF$ – повний імпульс у відповідному перерізі;

P_{mp} – сумарна сила тертя по бічній поверхні змішувальної камери.

Часто приймають $P_{mp} = 0$. В іншому випадку її можна визначити за формулою:

$$P_{mp} = \zeta \frac{L_3}{d_3} \frac{p_3\omega_3^2}{2} F_3 \quad (2.10)$$

де ζ – коефіцієнт опору.

Для турбулентного режиму приймаємо, що $\zeta = const$. Загальне значення $\zeta = 0,015 - 0,02$. Враховуючи значення приведеної швидкості потоку ежекції, отримаємо:

$$P_{mp} = \zeta \frac{L_3}{d_3} G_3 \lambda_3 a_{k3} \quad (2.11)$$

Рівняння імпульсів приймає наступний вигляд:

$$\left[z(\lambda_3) + \frac{1}{2} \chi \lambda_3 \right] \sqrt{(n+1)(1+n\theta)} = z(\lambda_1) + z(\lambda_2) n \sqrt{\theta} \quad (2.12)$$

Тут величина χ – довжина тертя, що обумовлена залежністю:

$$\chi = \zeta \frac{L_3}{d_3} \frac{2k}{k+1} \quad (2.13)$$

Так як в ежекторах відношення L_3/d_3 зазвичай не перевищує межу 10, то тертям можна знехтувати і використати формулу:

$$z(\lambda_3) \sqrt{(n+1)(1+n\theta)} = z(\lambda_1) + z(\lambda_2) n \sqrt{\theta} \quad (2.14)$$

Виразимо коефіцієнт ежекції через відношення витрат за допомогою залежності (7):

$$n = \frac{G_2}{G_1} = \frac{\sigma_2 p_n^* q(\lambda_2) F_2 \sqrt{T_1^*}}{\sigma_1 p^* q(\lambda_1) F_1 \sqrt{T_2^*}}$$

Після спрощення отримаємо:

$$n = \frac{p_2^* q(\lambda_2)}{p_1^* q(\lambda_1) \alpha \sqrt{\theta}} \quad (2.15)$$

де σ_2, σ_1 – коефіцієнти відновлення тиску відповідно в активному і пасивному соплі $p_1^* = p^* \sigma_1$ і $p_2^* = p_n^* \sigma_2$.

Аналогічно вище наведеній послідовності розрахунку, отримані рівняння, які зв'язують параметри повітря (газу) в перетині 1 і 3 і коефіцієнту ежекції:

$$\frac{p_3^* q(\lambda_3) F_3 \sqrt{T_1^*}}{p_1^* q(\lambda_1) F_1 \sqrt{T_3^*}} = n + 1 \quad (2.16)$$

Приведені рівняння дозволяють визначити основні параметри в перетинах 1, 2, 3 при заданих геометричних параметрах і початкових параметрах потоку. Для розрахунку дифузора використовують рівняння нерозривності в формі:

$$\frac{p_4^* F_4 q(\lambda_4)}{\sqrt{T_4^*}} = \frac{p_3^* F_3 q(\lambda_3)}{\sqrt{T_3^*}} \quad (2.17)$$

Коефіцієнт відновлення тиску в дифузорі σ_1 визначено за формулою:

$$\sigma_1 = \frac{p_4^*}{p_3^*} = \frac{p_4}{p_3^* \pi(\lambda_4)} \quad (2.18)$$

За умови $T_4^* = T_3^*$, рівняння нерозривності має вигляд:

$$q(\lambda_4) = \frac{F_3 q(\lambda_3)}{F_3 \sigma_1} \quad (2.19)$$

Якщо задати статичний тиск на виході p_4 із дифузора, рівняння нерозривності складатиме:

$$y(\lambda_4) = \frac{F_3 q(\lambda_3)}{F_4 \sigma_1} \quad (2.20)$$

Враховуючи формулу (2.1) для визначення повного тиску після змішування, матимемо узагальнене значення тиску:

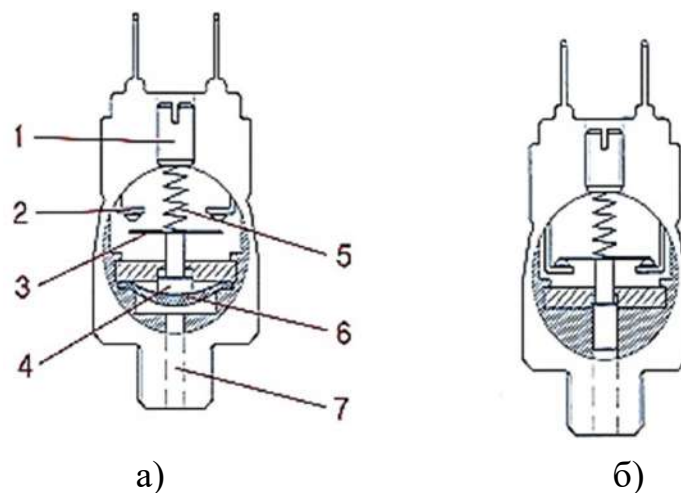
$$p_3^* = \frac{\alpha p_1^* + p_2^*}{1 + \alpha} \quad (2.21)$$

Прийmemo величину $\lambda_1, \lambda_2, \lambda_3$ у межах нормальних значень - (0,15-1,1) і відношення температур $0,3 < \Theta < 3,0$, для проведення імітаційного моделювання ежектора.

2.2 Реле, індикатори тиску, електропневмоперетворювачі в системі керування ежекторами

Реле тиску (англ. pressure switch, pressure relay) – пристрій, призначений для подання сигналу після того, як тиск робочого середовища досягне заданого значення. За принципом дії пристрій являє собою двоконтактне реле комутації електричних ланцюгів, що спрацьовує при досягненні відповідних значень тиску робочої речовини.

Назва реле визначила його призначення – управління поршневим компресором, для підтримки в ресивері робочого тиску. Часто його можна зустріти на компресорному блоці автоматики, або в системах керування ежекторів, пневмоприводів. Вибір проводиться відповідно до їх принципу дії. Існує декілька модифікацій (рис. 2.8): у першому варіанті механізм вимикає електромотор в момент перевищення норм рівня тиску стисненого повітря в пневмомережі – такі пристрої називаються нормально розімкненими.



1 – перетворювач тиску; 2 і 3 – контакти; 4 – поршень; 5 – пружина;
6 – мембрана; 7 – різьбове з'єднання

Рисунок 2.8 – Загальна схема реле тиску: а) мембранний вимикач тиску з нормально відкритим контактом (НВ); б) поршневий вимикач тиску з нормально закритим контактом (НЗ)

Прикладом вищеописаної схеми реле є елементи компанії Camozzi: регульоване реле тиску Серія РМ. Реле виробляється двох типів – нормально

закриті (НЗ, NC.) і нормально відкриті (НВ, NO). Необхідне значення тиску встановлюється регулювальним гвинтом за допомогою невеликої викрутки. Попередньо встановлений тиск перемикання – 5,5 бар. Загальний вигляд реле наведено на рис. 2.9.

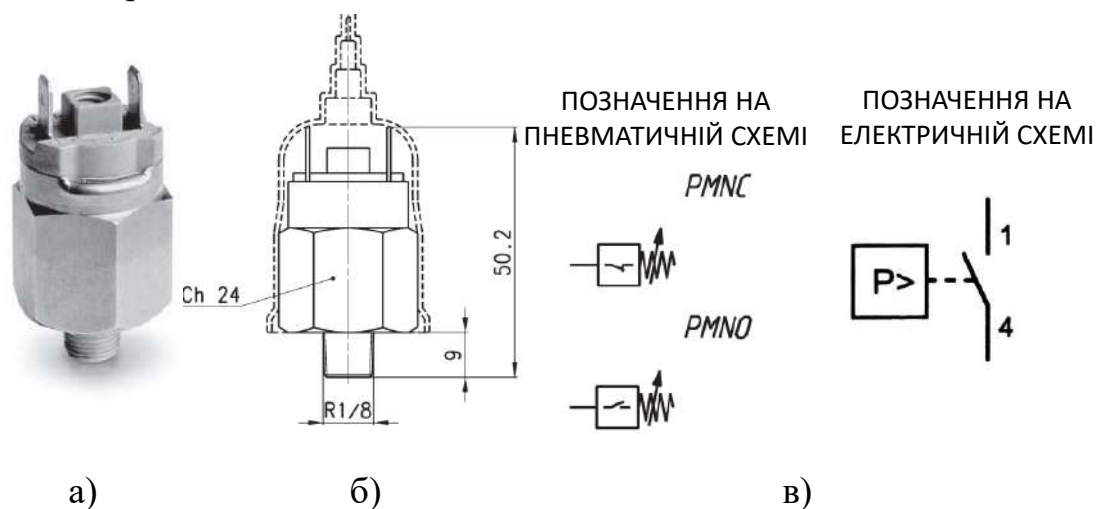


Рисунок 2.9 – Реле тиску серії РМ: а) загальний вигляд; б) ескіз реле із пілозахисним ковпачком; в) умовні позначення НЗ та НВ типів для принципової схеми керування

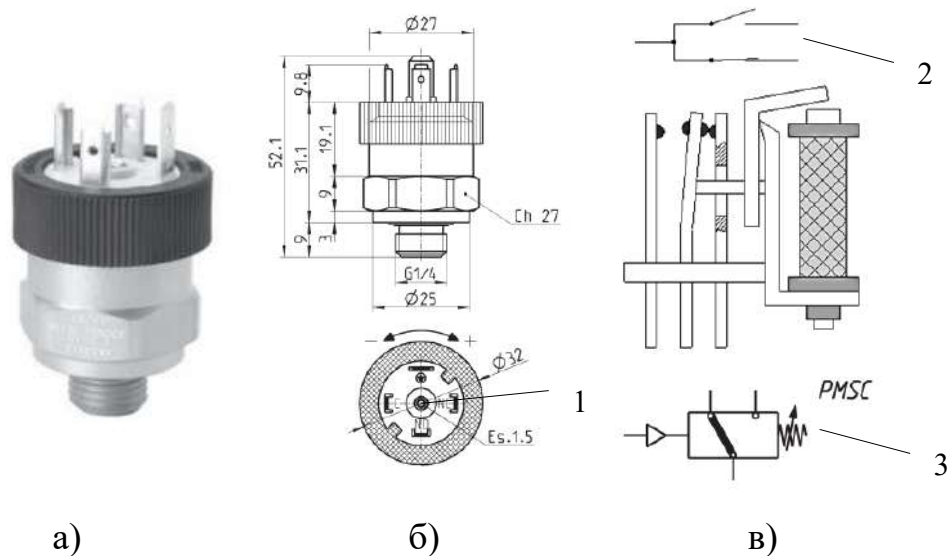
Також використовують реле, яке має два контакти з різними функціями: один з нормально відкритою, інший з нормально закритою.

На рис. 2.10, а котушка знеструмлена, електричний сигнал може проходити тільки за нормально закритому контакті. Тоді як на рис. 2.10, б – котушка збуджена і притягує механізм руху, який змінює стан контактів. Прикладом даного технічного рішення є реле тиску з перемикаючим контактом Мод. РМ11-SC Camozzi.

Нормально відкритий контакт закривається, тепер електричний сигнал може проходити, нормально закритий контакт відкривається, електричний сигнал переривається.

Необхідне значення тиску на реле РМ, встановлюється регулювальним гвинтом за допомогою невеликої викрутки. Попередньо встановлений тиск перемикання – 5,5 бар.

Такі реле використовуються в системах з дискретним керуванням пневмоприводу або ежектору.



- а) загальний вигляд; б) ескіз реле;
- 1 – регулювальний гвинт для налаштування порогового тиску;
 2 – умовне позначення перекидний контакт для електросхеми;
 3 – умовне позначення реле для пневмосхеми

Рисунок 2.10 – Реле тиску PM11-SC із перемикаючим контактом:

- а) загальний вигляд; б) ескіз реле;
 в) умовні позначення та схема роботи контактів

Технічні характеристики для реле тиску PM11-SC:

- діапазон регулювання 2...10 бар;
- конструкція з регульованою діафрагмою;
- допустимий струм:
 - а) 4 А / 12 В постійного струму;
 - б) 3 А / 24 В постійного струму;
 - в) 3 А / 110 В змінного струму;
 - г) 1 А / 220 В змінного струму;
- максимальний гістерезис 0,15 бар;
- механічний термін служби 1 млн. циклів.

2.2.1 Електронні реле вакууму / тиску

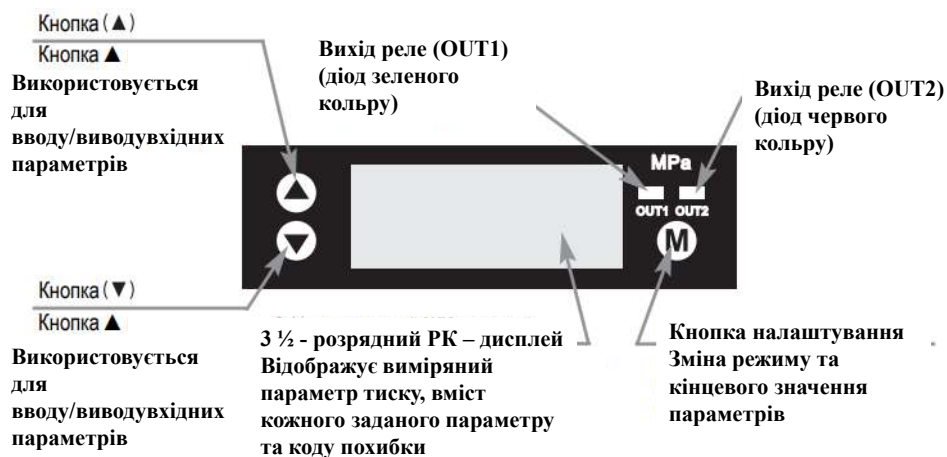
Для вирішення задач автоматизації технологічного процесу, де є позиційні та слідкуючі системи, тобто присутні умови зворотного зв'язку, застосовують електронні реле тиску. В таких пристроях вбудований цифровий

індикатор – прецизійна електронна схема із двома окремими дискретними виходами. Поріг спрацьовування і гістерезис програмується за допомогою мембранної клавіатури та із залученням програмованих логічних контролерів (ПЛК). Функція датчика тиску виконується із аналоговим виходом, та двома дискретними. Електронні реле вакууму / тиску використовуються в системах безпеки, для оптимізації часу циклу або в пристроях енергозбереження; можуть бути встановлені безпосередньо на виконавчих ланках захвату в маніпуляторах; для установки граничного значення вакууму, для постійного контролю за встановленим значенням тиску тощо. На рис. 2.11 наведено реле вакууму / тиску серії SWDN Camozzi.



1 – «+»; 2 – «OUT 2»; 3 – «-»; 4 – «OUT 1»

а)



б)

Рисунок 2.11 – Реле вакууму / тиску серії SWDN:

а) загальний вигляд; б) опис панелі керування

Зауважимо, що на рисунку вище, наведено приклад виконання електронного реле із дискретними виходами. В разі необхідності формування постійного віддаленого контролю за змінами фізичних параметрів та

автоматизації технологічної ділянки із мережею зворотного зв'язку, обирається реле із електричною схемою вихідного ланцюга на рис. 2.12, при цьому загальний опис налаштування приладу залишається.

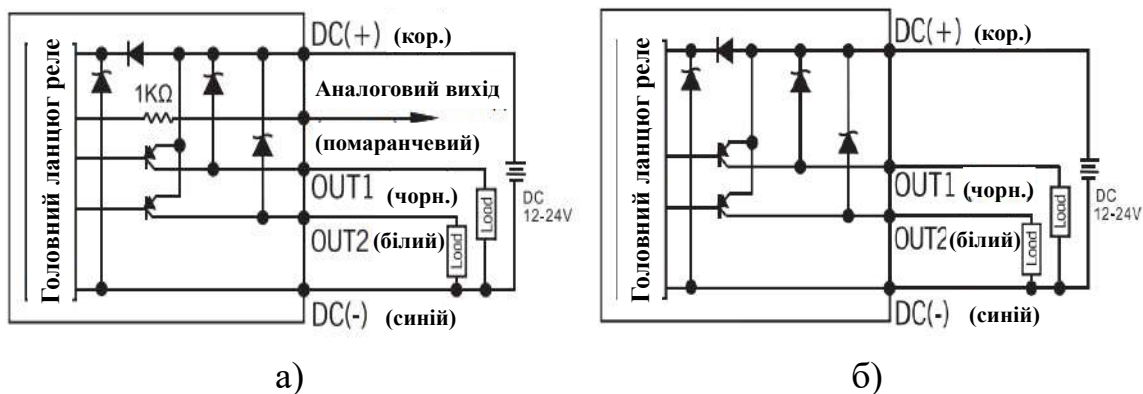


Рисунок 2.12 – Електричні схеми вихідного ланцюга:

а) аналогове виконання; б) дискретне виконання

Загальні технічні характеристики реле вакууму / тиску SWDN:

- діапазон розрахункового тиску -1..1 бар; 0...10 бар;
- робоче середовище – повітря, негорючі гази;
- крок встановленого тиску 0,1 кПа; 0,001 бар;
- напруга живлення 12...24 В постійного струму $\pm 10\%$, пульсація (P-P) 10% або менше;
- споживання струму ≤ 55 мА;
- вихід перемикача PNP;
- стабільність (вихід перемикача) $\leq \pm 0,2\%$ від повної шкали ± 1 знак;
- час відгуку $\leq 2,5$ мс (функція захисту від вібрації: 24 мс, опції 192 мс та 768 мс);
- аналоговий вихід – вихідна напруга від 1 до 5 В $\pm 5\%$ від повної шкали (у межах діапазону розрахункового тиску)
- лінійність $\leq \pm 1\%$ від повної шкали.

Зауважимо, що релейні елементи характеризуються параметрами, що відносяться до вхідних і вихідних впливів:

- спрацьовування – мінімальне значення впливу (на вході) при такому його зростанні, що релейний елемент змінює свій стан і одночасно впливає на виході відповідно до релейної характеристики;

- відпускання – мінімальне значення впливу на вході при такому його зменшенні, що релейний елемент повертається до свого початкового стану.

У зв'язку з не ідеальністю релейної характеристики ці величини зазвичай не збігаються одна з одною (гістерезис). У ряді випадків релейний елемент може мати властивості фіксації, тобто залишатися в зайнятому ним стані і після зняття впливу на вході. В цьому випадку релейний елемент повертається в первісний стан зазвичай після подачі впливу на інший його вхід (або вплив протилежного знаку впливу на той же вхід).

Поглянемо як це відбувається під час роботи реле вакууму / тиску серії SWDN на рис. 2.13.

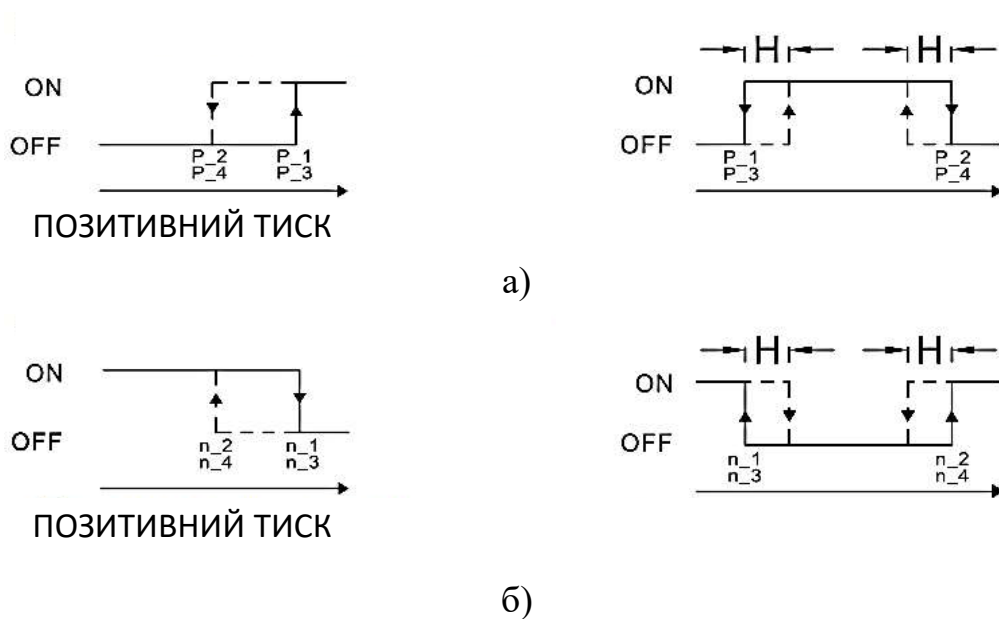


Рисунок 2.13 – Опис вихідних контактів реле вакууму / тиску серії SWDN, при налаштуванні двох діапазонів роботи приладу на OUT1 (точки P1...P2), OUT2 (точки P3...P4):

- а) стан контакту – нормально відкритий, Н – гістерезис;
- б) стан контакту – нормально закритий, Н – гістерезис

Приклад для реле тиску:

Реле налаштоване на два пороги спрацьовування:

- поріг включення 1,2 бар,
- поріг відключення 3 бар.

Гістерезис при цьому буде дорівнювати $(3 - 1,2) / 2 = 0,9$ бар.



Коли тиск становить 1 бар, реле замикає контакт. Коли тиск досягає 3 бар, реле розмикає контакт. Коли тиск досягає 1,2 бар, реле знову замикає контакт. Цикл повторюється. Якби тиск вмикання і вимикання мали одне значення, то гістерезису б не було. Тобто якщо поріг вмикання дорівнює порогу вимикання, то в такій системі відсутній гістерезис. А оскільки реле вакууму / тиску налаштовуються різними порогоми вмикання і вимикання, то така система має гістерезис. Гістерезис в свою чергу дозволяє зменшити кількість перемикання між двома положеннями: вмикання / вимикання. Але чим більше гістерезис, тим вище стрибкоподібна зміна тиску.

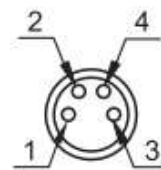
На рис. 2.15 наведений ще один приклад реле вакууму / тиску серії SWCN компанії Samozzi.

На рис. 2.15 наведено приклад виконання електронного реле із дискретними виходами. В разі необхідності формування постійного віддаленого контролю за змінами фізичних параметрів та автоматизації технологічної ділянки із мережею зворотного зв'язку, формується реле із електричною схемою вихідного ланцюга як на рис. 2.14, при цьому загальний опис налаштування приладу залишається.

Загальні технічні характеристики реле вакууму / тиску SWCN:

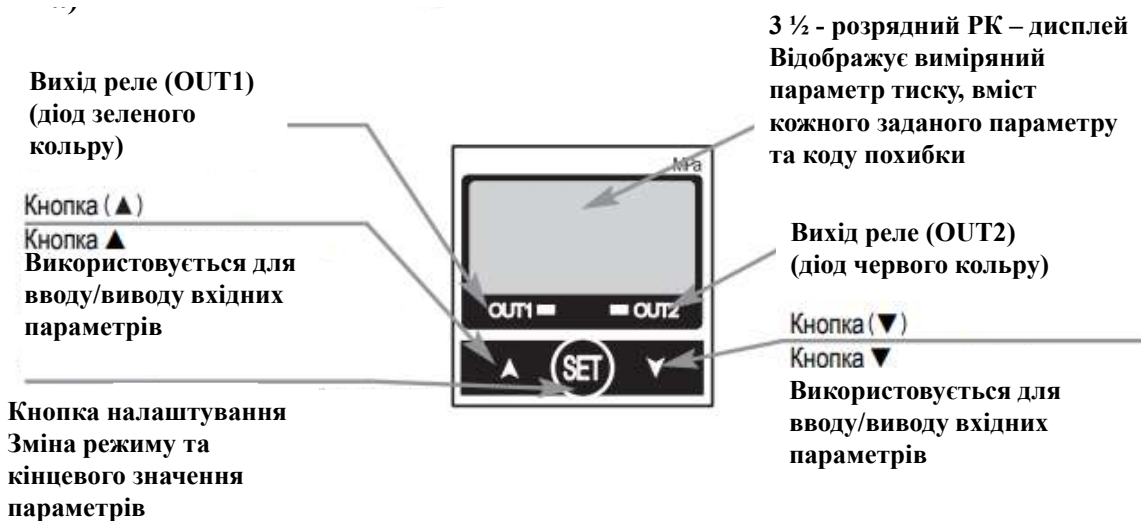
- діапазон розрахункового тиску -1..1 бар; 0...10 бар;
- робоче середовище – повітря, інертні гази;
- крок встановленого тиску 0,1 кПа; 0,001 бар;
- напруга живлення 12...24 В постійного струму $\pm 10\%$, пульсація (P-P) 10% або менше;
- споживання струму ≤ 55 мА;
- вихід перемикача PNP;
- стабільність (вихід перемикача) $\leq \pm 0,2\%$ від повної шкали ± 1 знак;

- час відгуку $\leq 2,5$ мс (функція захисту від вібрації: 24 мс, опції 192 мс та 768 мс);
- аналоговий вихід – вихідна напруга від 1 до 5 В $\pm 5\%$ від повної шкали (у межах діапазону розрахункового тиску)
- лінійність: $\leq \pm 1\%$ від повної шкали.



1 – «+»; 2 – «OUT 2»; 3 – «-»; 4 – «OUT 1»

а)



б)

Рисунок 2.14 – Реле вакууму/тиску серії SWCN:

а) загальний вигляд; б) опис панелі керування

Зауважимо, що наведені вище характеристики повністю збігаються із серією SWDN. Тобто, відмінністю є тільки форма корпусу та розташування

елементів на електронному дисплеї. Алгоритм налаштування вищеописаних електронних реле представлено на рис. 2.15.



Рисунок 2.15 – Алгоритм налаштування електронного реле вакууму / тиску

2.2.2 Характеристики реле в структурі пневматичних розподільників

Пневматичні розподільники і клапани призначені для зміни напрямку потоку стисненого повітря. Електропневматичний розподільник (клапан) – це комбінація двох основних функціональних вузлів: соленоїд (електромагніт) з сердечником (якорем) і безпосередньо розподільник (клапан) в якому встановлено золотник або поршень, щоб змінювати напрямок потоку стисненого повітря. Коли на котушку соленоїда подається живлення, магнітний сердечник втягується, викликаючи переміщення золотника і перемикання розподільника (клапана). Розрізняють електропневматичні розподільники з електричним (електромагнітним) управлінням і з

електропневматичним керуванням. У розподільниках (клапанах) з електричним керуванням золотник (поршень) механічно пов'язаний з сердечником електромагніту (рис. 2.16).

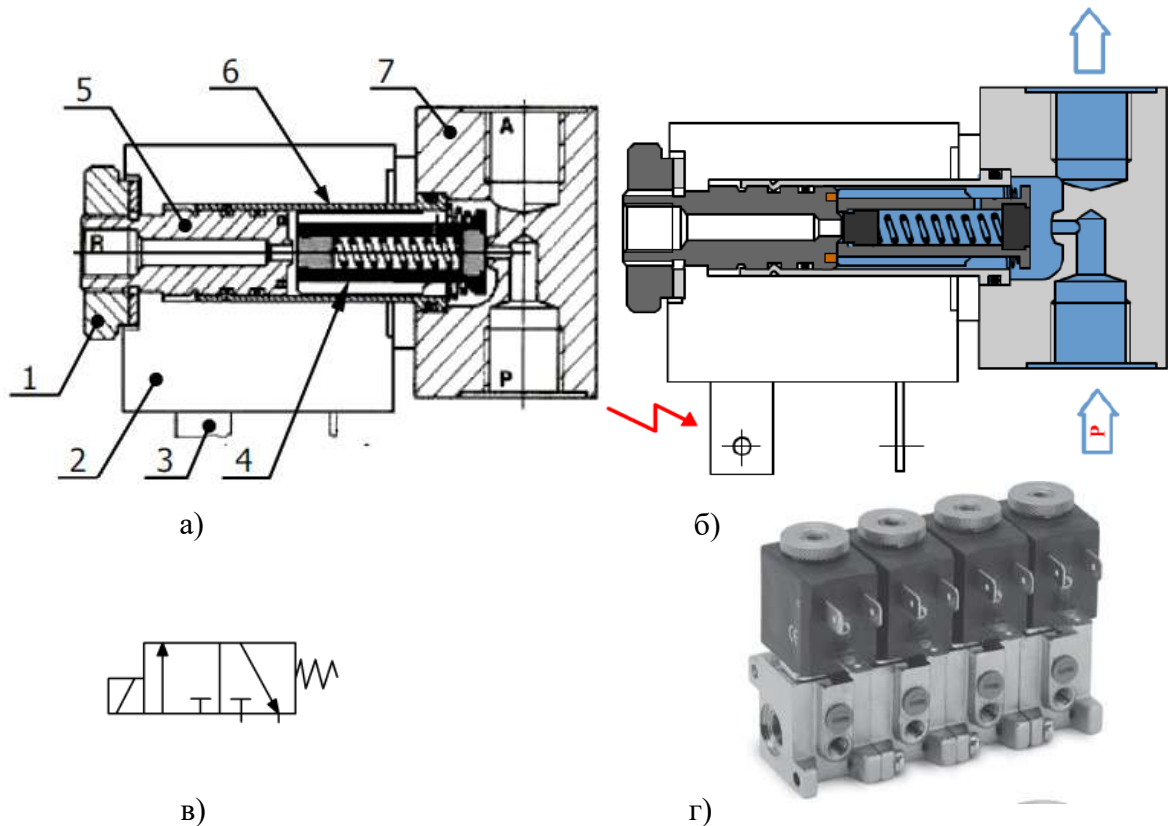


Рисунок 2.16 – Конструкція клапана з електронним управлінням:

- а) ескіз розподільника серії А (виробник Camozzi);
- б) робоче положення розподільника в момент перепуску стисненого повітря та спрацьовування ЕМР; в) умовне позначення на схемі керування;
- г)приклад групового монтажу серії А

У корпус 7 запресовано втулку 6, всередині якої знаходиться сердечник 5 і якір 4, одночасно є поршнем. Соленоїд (електромагніт) 2 насаджується на втулку і притискається до корпусу притисочною гайкою 1. За необхідності, соленоїд може бути замінено на інший. У початковому стані канал А з'єднаний з каналом R, а канал Р перекрито. З подачею напруги на контакти 3 соленоїда якір притягається до сердечника. При цьому стиснене повітря надходить з каналу Р в канал А, а канал R перекривається. Зв зняттям напруги, під дією пружини, сердечник повертається у вихідну позицію (моностабільний розподільник). У розподільниках з електропневматичним управлінням в

корпусі розподільника є додатковий керуючий канал 3, пов'язаний з каналом підведення стисненого повітря Р (рис. 2.17).

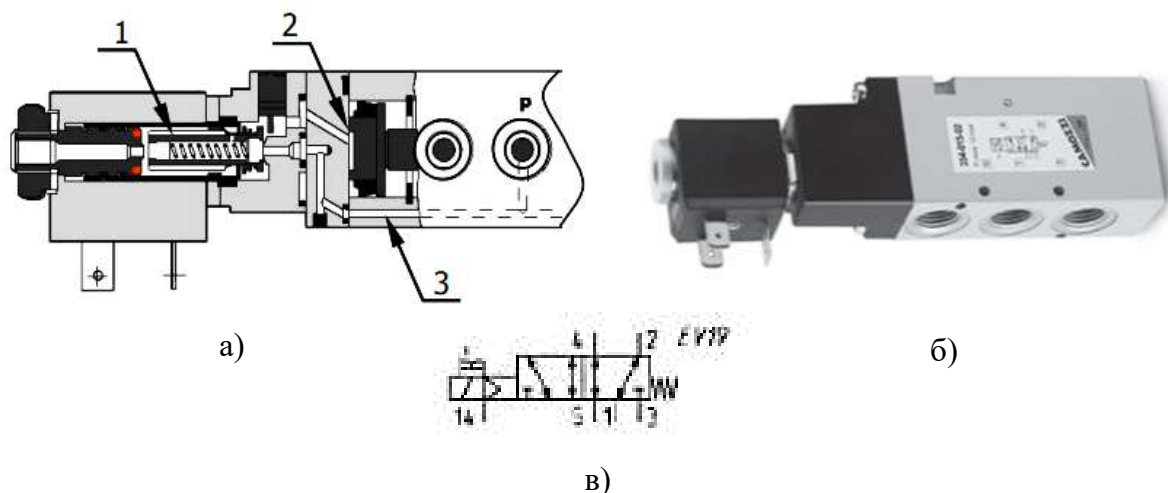


Рисунок 2.17 – Конструкція клапана з електропневматичним управлінням:
 а) ескіз розподільника серії 3 (виробник Camozzi);
 б) робоче положення розподільника в момент перепуску стисненого повітря та спрацьовування ЕМР; в) умовне позначення на схемі керування

З подачею напруги на контакти соленоїда якір 1 притягується до сердечника і відкриває керуючий канал. При цьому стиснене повітря впливає на поршень золотника і переміщує його в іншу позицію, викликаючи тим самим перекомутацію каналів. Розподільники з електропневматичним управлінням, розраховані на великі витрати повітря, можуть складатися з двох-трьох розподільників: основного, з пневматичним управлінням, і керуючих (пілотних), з електронним управлінням, приєднаних до входів управління.

Для управління електропневматичними розподільниками необхідно знати основні параметри соленоїдів – номінальну напругу і споживану потужність. Компанії-виробники пневматики випускають широкий спектр соленоїдів з різними параметрами. Наприклад, компанія Camozzi випускає соленоїди з номінальною напругою змінного струму 24 В, 48 В, 110 В, 230 В, 380 В і номінальною напругою постійного струму – 6 В, 12 В, 24 В, 48 В, 110 В.

Особливої уваги заслуговує соленоїд моделі G90. Він забезпечує можливість імпульсного управління клапаном. Досить короткого імпульсу –

20 мс, для того, щоб запірний елемент спрацював і залишився в цьому положенні. Для повернення його у вихідне положення необхідна подача короткого імпульсу за окремим входом. Таким чином, цей соленоїд дозволяє змінити принцип роботи клапана з моностабільного на бістабільний. Для регулювання в процесі налагодження пневмоприводу розподільники оснащуються ручним дублюванням, тобто можливістю переміщення золотника вручну (як правило, за допомогою викрутки). Варіанти умовних графічних позначень електропневматичних розподільників з різними видами управління наведені на рис. 2.18.

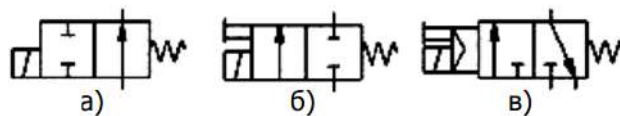


Рисунок 2.18 – Умовні графічні позначення пневморозподільвачів:
 а) з електричним керуванням; б) з електричним керуванням і ручним дублюванням; в) з електропневматичним керуванням і ручним дублюванням

Особливий вид пневморозподільвачів з електроуправлінням складають пневмоострови.

Пневмоострів – це декілька розподільників, встановлених на одній плиті, що має загальний канал підведення стисненого повітря для всіх розподільників і два загальних каналу вихлопу. За необхідності, можна комбінувати окремі плити таким чином, щоб розподільники працювали на різних тисках. Контакти електромагнітів об'єднуються в один загальний роз'єм і за допомогою кабелю підключаються до модуля управління.

Застосування пневмоостровів дозволяє зменшити кількість трубопроводів (і відповідно втрат тиску) і електричних проводів і полегшити монтаж. Зображений на рис. 2.19 пневмоострів компанії Samozzi, зібраний на основі Plug-In системи для електропневматичних розподільників, дозволяє об'єднати в один конструктивний модуль до 22 розподільників, що працюють з використанням трьох і більше різних номіналів тиску.



Рисунок 2.19 – Приклад модифікації Plug-In острова на базі електропневматичних розподільників серії 3 компанії Camozzi

Функції розподільників: 2x3/2, 5/2 і 5/3 (закритий центр, відкритий центр, подача тиску в обидві лінії). Багатоштиркова версія – підключення за допомогою роз'єму Sub-D 25-контактний. Шина Fieldbus: PROFIBUS-DP, DeviceNet, CANopen, EtherNET/IP, EtherCAT, PROFINET. Електрична частина острова оптимізована для видачі сигналів управління на моностабільні і бістабільні. Один електричний модуль може бути на 2 або 3 розподільника. Пневматична частина острова дозволяє здійснити роботу острова на різних тисках. Подача напруги на соленоїд розподільника здійснюється безпосередньо від джерела живлення за допомогою електричного перемикача (кнопки або тумблера) або через контакти електромагнітного реле.

2.2.3 Розрахунок та технічні характеристики електромагнітного реле

Електромеханічні елементи (разом з електромагнітними) є найбільш старими електричними елементами автоматики. Проте, видозмінюючись і вдосконалюючись, вони успішно конкурують із відносно новими магнітними елементами.

Електромагніт – найбільш простий перетворювач електричного сигналу в механічне зусилля й переміщення. Вхідний електричний сигнал подається на обмотку електромагніту, що притягає рухому частину, яка називається якорем.

За типом струму в обмотці розрізняють електромагніти постійного і змінного струму. Електромагніти постійного струму підрозділяють на нейтральні й поляризовані. Нейтральні притягають якір при будь-якій полярності струму в обмотці. У поляризованих електромагнітах напрямок зусилля, що діє на якір, змінюється зі змінною полярності струму в обмотці.

Часто електромагніти є приводними (тяговими) і слугують для переміщення таких виконавчих пристроїв, як клапани, заслінки та ін. Однак найбільше поширення отримали електромагніти, забезпечені контактною системою – електромагнітні реле.

Електромагнітні реле є одним з розповсюджених елементів багатьох систем автоматики, і випускається понад 200 типів тільки реле постійного струму.

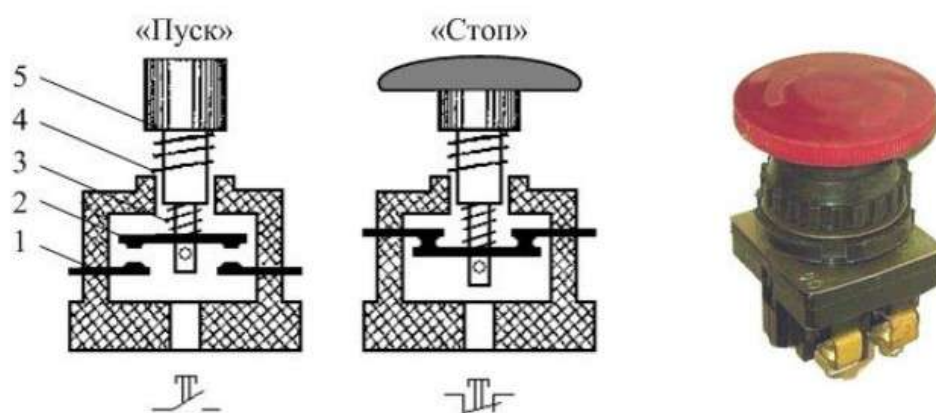
Реле призначено для виконання логічних операцій і безпосереднього керування силовими навантаженнями невеликої потужності, встановлюються в низьковольтних комплектних пристроях керування промисловими об'єктами, а також у пристроях торговельної, медичної та іншої техніки. За величиною споживаної при спрацьовуванні потужності реле можна підрозділити на високочутливі (до 10 мВт) і слабкострумові нормальної чутливості (до 1...5 Вт).

Реле можна розділити за часовими параметрами на нормальні, швидкодіючі й уповільнені, так звані реле часу.

До електромагнітного реле висувають різноманітні вимоги, які не завжди вдається задовольнити в одній конструкції. Насамперед задаються вимоги чутливості й потужності. Часто реле повинні мати малі габарити, велику кількість перемикаємих ланцюгів (контактів), мати більший строк служби й достатньою надійність роботи в умовах вібрації, при різких коливаннях температури й вологості, малим часом спрацьовування й відпускання, а іноді й значною витримкою часу при спрацьовуванні або відпусканні.

Як засоби автоматизації у всіх галузях промисловості широке застосування знаходять електромагнітні елементи автоматики, значну частку яких становлять різні електромагнітні механізми. У зв'язку із цим знання теорії, практики розрахунку й основ оптимального проектування останніх є

необхідним для інженерів різних спеціальностей, особливо інженерів-електриків і інженерів-електромеханіків. Застосування електромагнітних реле в апаратурі пред'являє ряд істотних вимог до технічної документації, до літератури й до знань розроблюваної апаратури. Однією з вимог є зменшення маси й габаритів, зниження вартості, підвищення стійкості до зовнішніх дестабілізуючих факторів, надійність і довговічність апаратури. Наприклад на рис. 2.20 представлено пристрій і зовнішній вигляд кнопкових перемикачів SB1 «Пуск» і SB2 «Стоп», що застосовуються для управління електромагнітним контактором постійного струму.



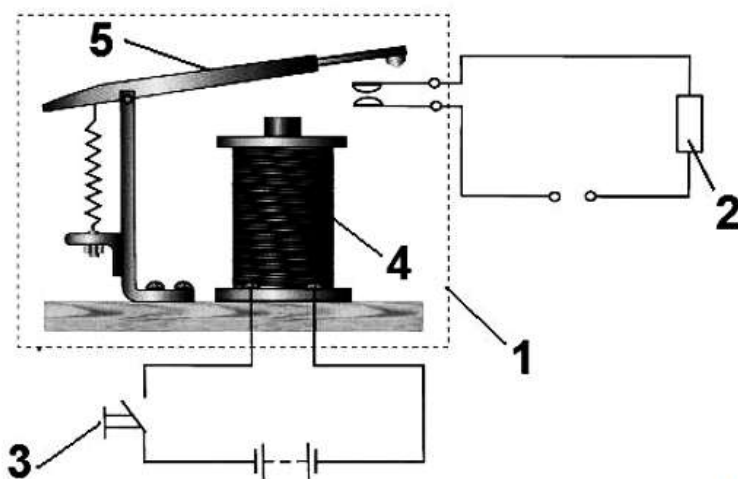
1 – нерухомий контакт; 2 – рухомий контакт (місток); 3 – натискна пружина;
4 – поворотна пружина; 5 – привод кнопки

Рисунок 2.20 – Пристрій кнопкових перемикачів «Пуск» і «Стоп»

Подача напруги на соленоїд розподільника здійснюється безпосередньо від джерела живлення за допомогою електричного перемикача (кнопки або тумблера) або через контакти електромагнітного реле (рис. 2.21).

Робота електромагнітних реле заснована на використанні електромагнітних сил, що виникають в металевому сердечнику з проходженням струму за витками його котушки – 4. Деталі реле монтуються на основі і закриваються кришкою. Над сердечником електромагніту встановлений рухомий якір (пластина) – 5 з одним або декількома контактами. Навпроти них знаходяться відповідні парні нерухомі контакти. У вихідному положенні якір утримується пружиною. З подачею напруги електромагніт

притягує якір, долаючи її зусилля, і замикає або розмикає контакти в залежності від конструкції реле. Після відключення напруги пружина повертає якір в вихідне положення. В деякі моделі, можуть бути вбудовані електронні елементи. Це резистор, підключений до обмотки котушки для більш чіткого спрацьовування реле, або (і) конденсатор, паралельний контактам для зниження іскріння і перешкод.



1 – реле; 2 – навантаження; 3 – кнопка; 4 – котушка з сердечником; 5 – якір

Рисунок 2.21 – Схема включення електричного навантаження через контакти електромагнітного реле

Керований ланцюг з навантаженням – 2 електрично ніяк не пов'язаний з керуючим елементом (кнопка – 3), більш того в керованому ланцюзі величина струму може бути набагато більша ніж в керуючій частині. Реле, по суті, виконує роль підсилювача струму, напруги або потужності в електричному ланцюзі. Електромагнітні реле, завдяки простим принципом дії і високої надійності, отримали широке застосування в системах автоматики і в схемах захисту електроустановок. За способом управління електромагнітні реле діляться на реле постійного і змінного струму. Реле постійного струму діляться на нейтральні і поляризовані. Нейтральні реле однаково реагують на постійний струм обох напрямків, що протікає за його обмоткою, а поляризовані реле реагують на полярність керуючого сигналу. Умовне графічне позначення реле на електричних і пневмоелектричних схемах наведено на рис. 2.22.

Електромагнітне реле має одну робочу обмотку (K1) і один або кілька контактів (K1.1-K1.3). Контакти бувають нормально-розімкнуті (K1.1), нормально замкнуті (K1.2) і парні (K1.3), що представляють комбінацію нормально розімкнутого і нормально-замкнутого контактів. Якщо на робочу обмотку реле подати напругу, то нормально-розімкнутий контакт замикається, нормально-замкнутий – розмикається, парний – перемикається. На електричній схемі контакти реле зображуються у вихідному стані (тобто коли на робочу обмотку реле напруга не подається). Дуже часто в релейних схемах використовується функція «Самоблокування» (або «самопідтримки»). Контакти показані в початковому стані. Стрілками показано напрямок перемикання з подачею напруги управління.

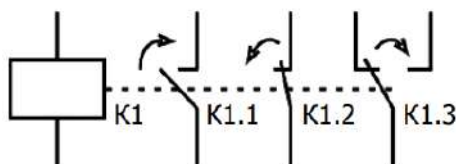


Рисунок 2.22 – Умовне графічне позначення електромагнітного реле

2.3 Синтез принципів схем керування із використанням реле та програмованих логічних контролерів

Незважаючи на те, що в даний час при побудові автоматичних систем керування (АСК) технологічними процесами широко використовуються програмовані логічні контролери (ПЛК), електричні системи управління як і раніше часто застосовуються завдяки простоті, зручності комутації і невисокою вартістю.

Перевагою релейно-контактних схем (РКС), які складають основу для електричних систем управління, є їх універсальність, тобто можливість використання одного і того ж схематичного рішення для управління силовими частинами приводів, побудованих як на базі пневматичних, так і гідравлічних пристроїв. Умовні графічні позначення елементів, що входять в релейно-контактну схему, викреслюються між двома паралельними горизонтальними прямими, які зображують електричні шини. У разі застосування джерела

постійного струму, на верхню шину подається живлення від його «+», а на нижню від «-». Поруч з шинами може проставлятися значення напруги в них (наприклад, 24 В).

Схема складається і читається зліва направо, причому струмопроводи нумеруються над верхньою шиною по порядку. Паралельно працюючим струмопроводам може бути присвоєно один і той самий порядковий номер, але при цьому його доповнюють буквеним індексом (наприклад, 2 і 2а). Напрямок проходження електричних сигналів приймається зверху вниз, тобто від «+» до «-». Позначення на схемі електромагнітних приводів розподільників, реле або інших елементів зображують безпосередньо приєднаними до нижньої шини («-»). У той же час керуючі даними елементами пристрою введення електричних сигналів розміщуються в струмопроводах, з'єднаних із верхньою шиною («+»). Це правило засноване на практичному досвіді монтажу та експлуатації релейно-контактних систем управління і вимогах безпечної роботи. Очевидно, що система буде працювати і в тому випадку, якщо одну з клем електромагнітного приводу (наприклад, приводу розподільника) з'єднати безпосередньо з «+» джерела живлення, а включення даного приводу здійснювати від «-».

Однак в цих умовах різко зростає ймовірність самовільного спрацювання, як розподільника, так і керованого ним виконавчого механізму. Виникнення в системі будь-якого дефекту (обрив проводу або пошкодження його ізоляції) може привести до зіткнення другої клеми приводу розподільника із заземленим шафи керування, що викличе спрацювання розподільника і, як наслідок, неконтрольоване переміщення виконавчого механізму і ведених вузлах технологічної установки (рис. 2.23).

Поруч з умовними позначеннями елементів релейно-контактних систем управління проставляють букви і цифри індекси відповідно до існуючих стандартів. Якщо різні контакти будь-якого пристрою розташовуються в поряд розташованих струмопроводах, то допускається зображення між ними механічного зв'язку. Під позначенням реле викреслюються таблиці з інформацією про тип наявних в даному реле контактів і струмопроводах, в яких ці контакти задіяні.

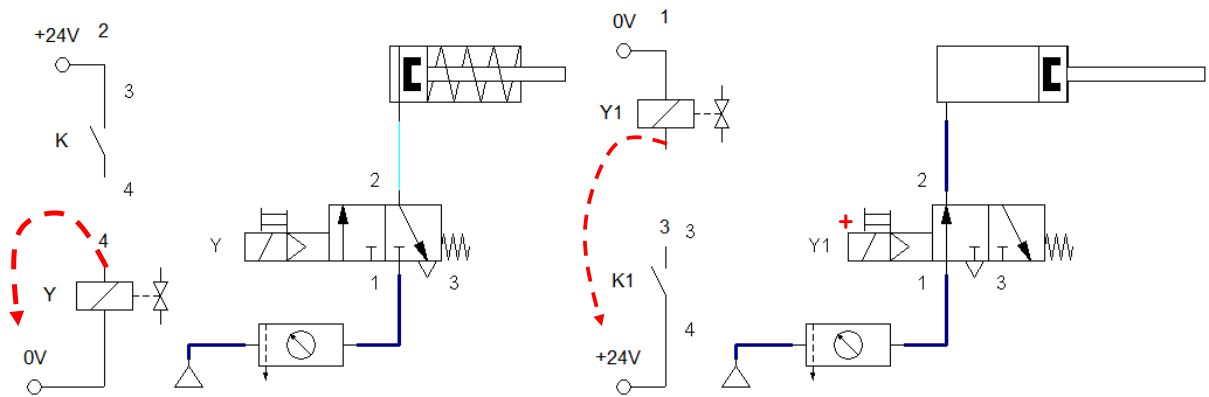


Рисунок 2.23 – Прояв дефектів в релейно-контактних системах управління

Всі елементи релейно-контактної схеми зображуються в початковому положенні, тобто за відсутності напруги в шинах живлення. Якщо в початковому положенні електромеханічні кінцеві вимикачі активні, то на схемі поряд з їх позначенням викреслюється вертикальна стрілка.

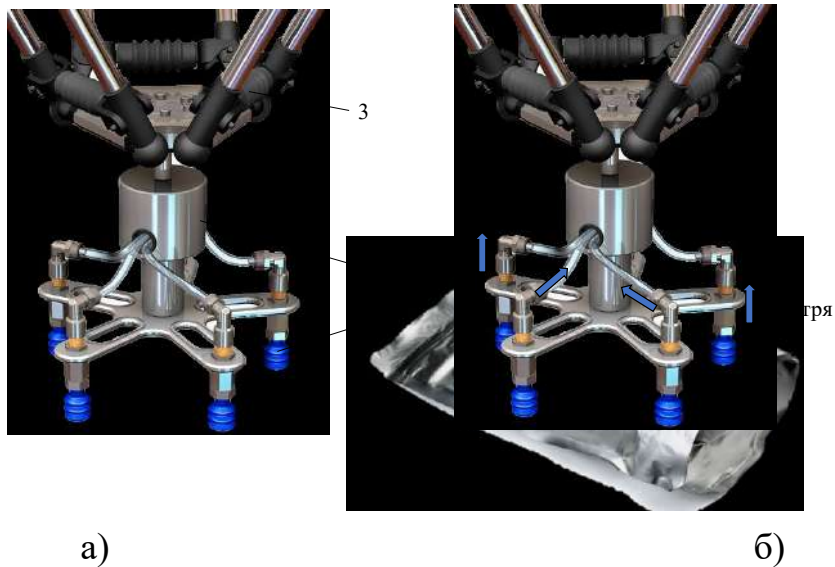
На принциповій пневматичній схемі місця розташування кінцевих вимикачів, які контролюють положення вихідних ланок виконавчих механізмів, позначаються вертикальними штрихами, які супроводжують буквено-цифрові індекси. Умовні позначення пневматичної та електричної схем наведено в додатках А та Б.

Розглянемо типову електропневмосистему на прикладі задачі керування функціональним технологічним модулем захвату, який активується від кнопки та є вихідною ланкою маніпулятора триподу рис. 2.24.

В системі робота-маніпулятора на основі трипода для здійснення операції захвату та транспортування об'єктів оброблення застосовується ежектор із вакуумними захватами (присосками).

Система передбачає декілька режимів керування (рис. 2.25): ручний (для запуску та перевірки роботи під час налагоджувальних та ремонтних робіт); автоматичний (оброблення об'єктів відповідно до умов технологічного циклу виробництва).

За допомогою електричної кнопки (або перемикача), оператор активує роботу вакуумного захвату, попередньо налаштувавши пороговий рівень робочого тиску на вході в систему (регулятор тиску).



а) б)
 1 – присоска сільфонна на 2,5 гофри; 2 – корпус для розміщення ежектора;
 3 – кінематичні ланки триподу

Рисунок 2.24 – Загальний вигляд вихідного функціонального вузла маніпулятора триподу: а) неробоче положення; б) положення під час здійснення технологічної операції

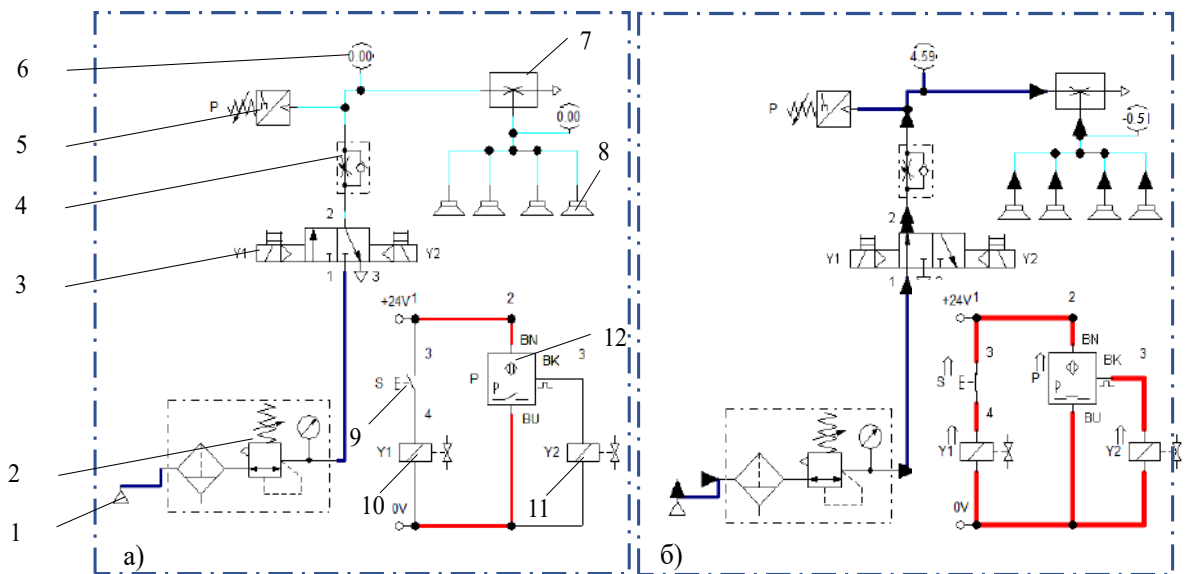


Рисунок 2.25 – Принципова електропневматична схема управління захватним пристроєм в ручному режимі (тестування):

- а) вихідне положення елементів при увімкненні живлення;
- б) робоче положення елементів при активації пускової кнопки

На рис. 2.25 позначено: 1 – джерело живлення; 2 – регулятор тиску; 3 – розподільник 5/2 з електропневматичним керуванням та ручним дублюванням; 4 – ежектор; 5 – реле тиску; 6 – манометр; 7 – ежектор; 8 – присоски; 9 – S кнопка пуск; 10, 11 – ЕМР керування розподільником; 12 – реле тиску.

Реле тиску при досягненні потрібного тиску розрідження 0,5 бар (0,05 МПа), автоматично вимикає ежектор з спрацьовуванням Y2. Для зручності привод оснащений двома манометрами 6: перший манометр на вході в ежектор фіксує значення магістрального тиску (надлишкового, бар); другий манометр – показує інформацію про рівень вакууму на присосках.

Максимальне зусилля захвату, що розвивається присосками, встановлюється за допомогою регулювання надлишкового тиску на вході в ежектор та відповідає обраним характеристикам ежектора та присосок. Цим запобігається можливе пошкодження об'єкта оброблення під час захоплення та транспортування. Наприклад, розглянемо характеристики ежектора VEB-30L (рис.2.26).

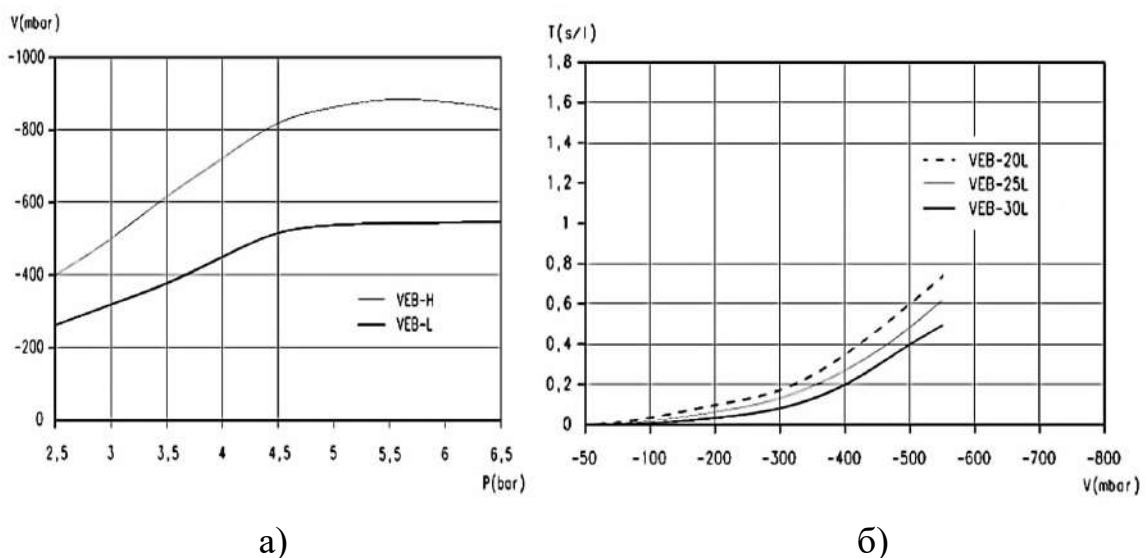


Рисунок 2.26 – Характеристики роботи ежектора VEB-30L:

а) залежність максимального розрідження (вакууму) від вхідного магістрального тиску; б) час створення вакууму в одному літрі об'єму при різних глибинах вакууму

За наведеними емпіричними залежностями, можна визначити межі розрідження на виході із ежектора при зміні тиску в магістралі (за допомогою регулятора тиску 2, рис. 2.25), а також порівняти швидкість (час) створення вакууму.

Таким чином, при створенні магістрального тиску на вході в ежектор 4,59 бар, отримаємо розрідження -500 мбар; за умови створення вакууму -0,4 с/л.

В разі необхідності вирішити дану задачу із застосуванням ПЛК, пневматична схема не зміниться. На рис. 2.27 подано електричну схему підключення елементів керування до ПЛК та програму керування на базі ПЛК -SIMATIC S7-200/-SIMATIC S7-1200.

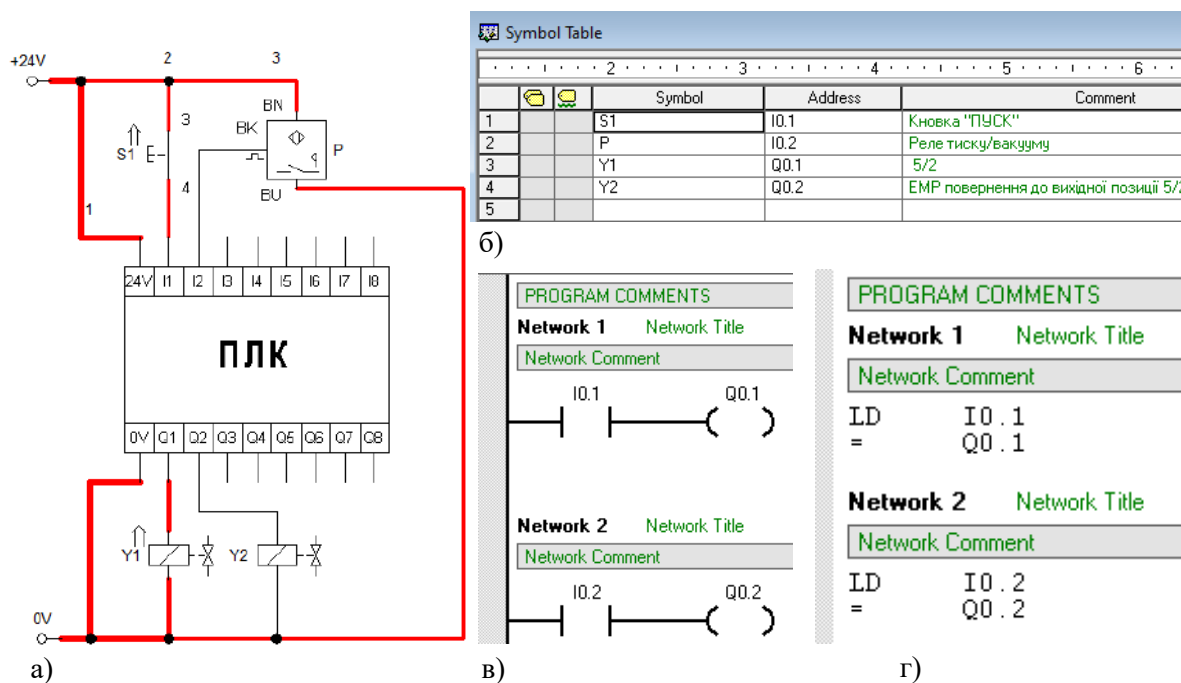


Рисунок 2.27 – Результати розробки керуючої системи на базі ПЛК:

- а) електрична схема під'єднання основних електричних компонентів схеми;
- б) символна таблиця для формування опису окремих елементів та коментарів їх роботи;
- в) код програми в мові LAD;
- г) код програми в мові STL

Вирішення другої частини задачі для організації автоматичного режиму керування захватом буде розглянуто в наступному розділі.

2.3 Контрольні запитання та завдання

1. Наведіть приклади використання ежекторів в промисловому обладнанні.
2. Опишіть конструкцію вакуумних ежекторів
3. Яким чином виконується розрахунок основних технологічних характеристик ежектора?
4. Наведіть загальну схему реле тиску
5. Яким чином працює реле тиску?
6. Наведіть схему та принцип роботи реле вакууму.
7. Поясніть особливості роботи реле в структурі пневматичних розподільників
8. Що таке пневмоострів?
9. Яким чином виконується синтез принципових схем керування із використанням реле та програмованих логічних контролерів?

3 МЕХАТРОННІ ПНЕВМАТИЧНІ СИСТЕМИ

3.1 Завдання механіки маніпуляторів

До основних завдань механіки маніпуляторів можна віднести:

- розробку методів синтезу і аналізу виконавчих механізмів (включаючи механізми приводів);
- програмування руху маніпулятора;
- розрахунок реакцій і зусиль, що управляють;
- урівноваження механізмів маніпуляторів;
- інші завдання.

Ці завдання вирішуються на базі загальних методів дослідження структури, геометрії, кінематики і динаміки систем з просторовими багаторухливими механізмами. Кожна з цих задач може бути сформульована як пряма (завдання аналізу) або як зворотна (завдання синтезу). При визначенні функцій положення механізму, в прямій задачі знаходять закон зміни абсолютних координат вихідної ланки за заданими законами зміни відносних або абсолютних координат ланок. У зворотній – за заданим законом руху захвату знаходять закони зміни координат ланок, зазвичай, лінійних або кутових переміщень в приводах. Рішення зворотної задачі або завдання синтезу складніше, оскільки часто вона має безліч допустимих рішень, з яких необхідно вибрати оптимальне. У зворотній задачі кінематики за необхідним законом зміни швидкостей і прискорень вихідної ланки визначаються відповідні закони зміни швидкостей і прискорень в приводах маніпулятора. Зворотна задача динаміки полягає у визначенні закону зміни сил, що управляють, і моментів у приводах, що забезпечують заданий закон руху вихідної ланки.

3.2 Кінематичний аналіз механізму маніпулятора

Перше і основне завдання кінематики – визначення функції положення. Для просторових механізмів найбільш ефективними методами рішення цієї задачі є векторний метод і метод перетворення координат. При рішенні прямої

задачі про положення захвату маніпулятора зазвичай використовують метод перетворення координат. З багатьох методів перетворення координат, які відрізняються один від одного правилами вибору осей локальних систем координат, для маніпуляторів зазвичай використовується метод Денавіта і Хартенберга.

Опишемо два типи матриць:

- матриці M , що визначають відношення між системами координат сусідніх ланок;

- матриці T , визначають положення і орієнтацію кожної ланки механізму в нерухомій або базовій системі координат.

Скористаємося однорідними координатами тривимірного проектного простору PR^3 , в яких рух евклідова простора R^3 можна представити лінійним перетворенням

$$\bar{r}_i = M_{ij} \cdot \bar{r}_j, \quad (3.1)$$

де M_{ij} – матриця 4×4 виду $\begin{vmatrix} U_{ij} & b \\ 000 & 1 \end{vmatrix}$.

Це перетворення еквівалентно перетворенню в евклідовому просторі $\bar{r}_i = U_{ij} \cdot \bar{r}_j + b$, де $[\bar{r}_i, \bar{r}_j] \in R^3$. Тобто перетворенню, яке містить поворот, який визначається матрицею U_{ij} розмірністю 3×3 , і паралельне перенесення, що задається вектором розмірністю 3. В однорідному просторі положення точки визначатимуть не три x, y і z , а чотири величини x', y', z' і t' , які задовольняють наступним співвідношенням

$$x=x'/t', \quad y=y'/t', \quad z=z'/t'. \quad (3.2)$$

Зазвичай приймають $t'=1$. У матриці повороту U_{ij} елементами є направляючі косинуси кутів між новою віссю i та старою віссю j . Вектор $\bar{b} = (x, y, z)$ – тривимірний вектор, що визначає положення початку нової системи координат i в старій системі j . Вибір розташування осей повинен відповідати задачі, яку потрібно розв'язати. При рішенні задачі про положення

необхідно: в прямому завданні визначити положення вихідної ланки, як функцію переміщень в приводах, в зворотній – задане положення вихідної ланки представити як функцію переміщень в приводах. Вибір розташування і орієнтації локальних систем координат повинен забезпечувати виконання цих завдань. При використанні методу Денавіта і Хартенберга осі координат розташовуються за наступними правилами.

Для ланки i вісь z_i спрямовується за віссю кінематичної пари, що утворюється ним з ланкою $i+1$. Початок координат розміщують в геометричному центрі цієї пари.

Вісь x_i спрямовується за загальним перпендикуляром щодо осей z_{i-1} і z_i з напрямом від z_{i-1} до z_i . Якщо осі z_{i-1} і z_i співпадають, то x_i перпендикулярна щодо них і спрямована довільно. Якщо вони перетинаються в центрі кінематичної пари, то початок координат розташовується в точці перетину, а вісь x_i спрямовується за правилом векторного добутку (найкоротший поворот осі z_i до поєднання з z_{i-1} при спостереженні з кінця x_i повинен відбуватися проти годинникової стрілки). Вісь y_i спрямовується таким чином, щоб система координат була правобічною.

У прямій задачі необхідно визначити положення захвату маніпулятора і пов'язаної з ним системи координат $M_{x_n y_n z_n}$ по відношенню до нерухомої або базової системи координат $K_{x_0 y_0 z_0}$. Це здійснюється послідовними переходами з системи координат ланки i у систему координат ланки $i-1$. Згідно з прийнятим методом, кожен перехід включає послідовність чотирьох рухів: двох поворотів і двох паралельних перенесень, які здійснюються у вказаній послідовності (див. рис. 3.1):

- поворот i -ої системи навколо осі x_i на кут $-\theta_i$ до паралельності осей z_i і z_{i-1} (позитивний напрям повороту при спостереженні з кінця вектору x_i проти годинникової стрілки);

- перенесення уздовж осі x_i на величину $-a_i$ до поєднання початку системи координат O_i з точкою перетину осей x_i і z_{i-1} (відлік за віссю x_i від точки перетину осі x_i і осі z_{i-1});

- перенесення уздовж осі z_{i-1} на величину $-s_i$, після якого початок системи координат O_i виявляється на початку координат O_{i-1} системи $i-1$

(відлічується за віссю z_{i-1} від її початку координат до O_{i-1} точки її перетину з віссю x_i);

- поворот навколо осі z_{i-1} на кут $-\varphi_i$, до тих пір, доки вісь x_i не стане паралельною щодо осі x_{i-1} (позитивний напрям повороту при спостереженні з кінця вектору z_{i-1} проти годинникової стрілки).

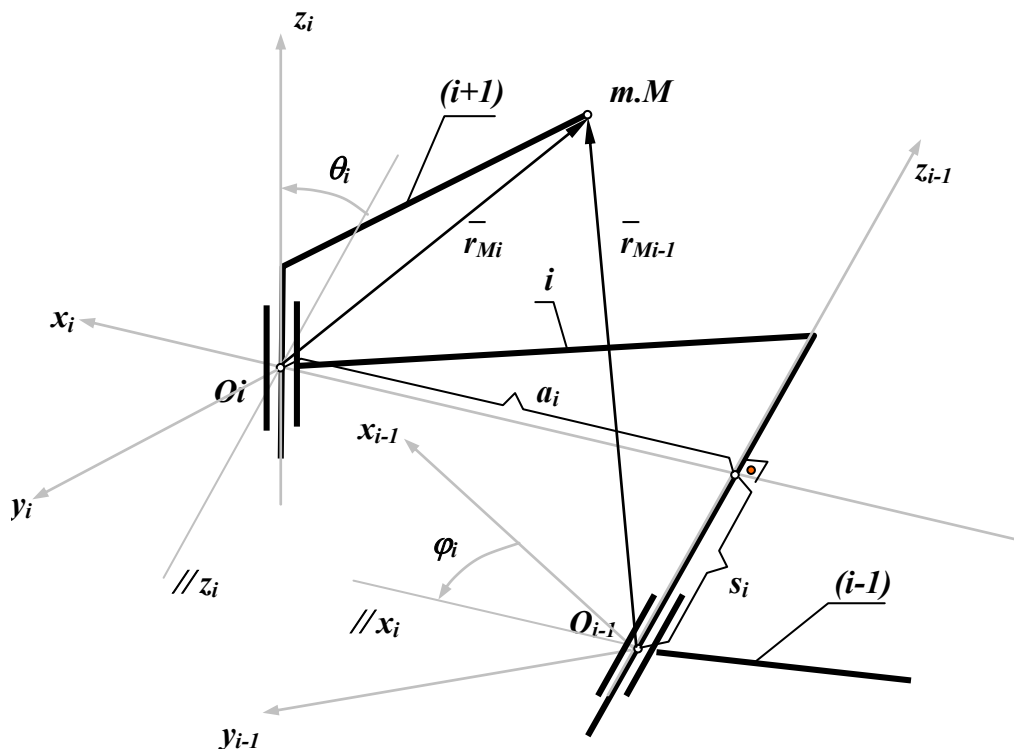


Рисунок 3.1 – Схема положення захвату

Необхідно відмітити, що знак кута повороту не має значення, оскільки в матрицях переходу використовуються направляючі косинуси (парні функції). Доцільно розглядати кут, що забезпечує найкоротший поворот осі старої системи i до поєднання (паралельності) з відповідною віссю нової $i-1$. Переміщення початку координат визначаються як координати початку старої системи O_i в новій O_{i-1} . В маніпуляторах зазвичай використовуються однорухливі кінематичні пари або обертальні, або поступальні. Обидві відносні рухи як обертальні, так і поступальні, реалізуються в циліндричних парах. Тому при загальному поданні механізму використовуються циліндричні пари.

Матриці переходу їх системи O_i в систему O_{i-1} можна записати наступним чином

$$M_i = M_i^\theta \cdot M_i^a \cdot M_i^s \cdot M_i^\varphi, \quad (3.3)$$

$$\text{де } M_i^\theta = \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \theta_i & -\sin \theta_i & 0 \\ 0 & \sin \theta_i & \cos \theta_i & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \text{ – матриця повороту навколо осі } x_i \text{ на кут } -\theta_i,$$

$$M_i^a = \begin{vmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \text{ – матриця перенесення уздовж осі } x_i \text{ на } -a_i,$$

$$M_i^s = \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & s_i \\ 0 & 0 & 0 & 1 \end{vmatrix} \text{ – матриця перенесення уздовж осі } z_{i-1} \text{ на } -s_i,$$

$$M_i^\varphi = \begin{vmatrix} \cos \varphi_i & -\sin \varphi_i & 0 & 0 \\ \sin \varphi_i & \cos \varphi_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \text{ – матриця повороту навколо осі } z_{i-1} \text{ на кут } -\varphi_i.$$

У цих матрицях змінні s_i і φ_i відповідають відносним переміщенням ланок в кінематичних парах і є узагальненими координатами маніпулятора, що визначають конфігурацію механізму в даному положенні. Змінні a_i і θ_i визначаються конструктивним виконанням ланок маніпулятора, в процесі руху вони залишаються незмінними.

Положення деякої довільної точки M в системі координат ланки i визначається вектором rM_i , а в системі координат ланки $i-1$ – вектором rM_{i-1} . Ці радіуси пов'язані між собою через матрицю перетворення координат M_i наступним рівнянням:

$$\bar{r}M_{i-1} = M_i \cdot \bar{r}M_i, \quad (3.4)$$

$$\text{де } M_i = \begin{vmatrix} \cos \varphi_i & -\cos \theta_i \cdot \sin \varphi_i & \sin \varphi_i \cdot \sin \theta_i & a_i \cdot \cos \varphi_i \\ \sin \varphi_i & \cos \theta_i \cdot \cos \varphi_i & -\cos \varphi_i \cdot \sin \theta_i & a_i \cdot \sin \varphi_i \\ 0 & \sin \theta_i & \cos \theta_i & s_i \\ 0 & 0 & 0 & 1 \end{vmatrix} \text{ – матриця переходу з } i\text{-ої}$$

системи координат в $(i - 1)$ - у.

Розглянемо шестирухливий маніпулятор в початковому положенні (рис.3.2). За початкове положення береться таке, в якому усі відносні узагальнені координати дорівнюють нулю. Перехід з системи координат будь-якої i -ої ланки до нерухомої або базової системи записується у вигляді:

$$\bar{r}_{M0} = M_1 \cdot M_2 \cdot \dots \cdot M_i \cdot \bar{r}_{Mi} \text{ або } \bar{r}_{M0} = T_i \cdot \bar{r}_{Mi}, \quad (3.5)$$

де $T_i = M_1 \cdot M_2 \cdot \dots \cdot M_i$ – матриця перетворення координат i -ої системи в координати базової системи координат.

Для схеми, поданої на рис. 3.2, радіус $r_{M6} = 0$, а радіус r_{M0} визначиться за формулою $\bar{r}_{M0} = T_n = T_6$, тобто положення вихідної ланки маніпулятора визначається матрицею T_n .

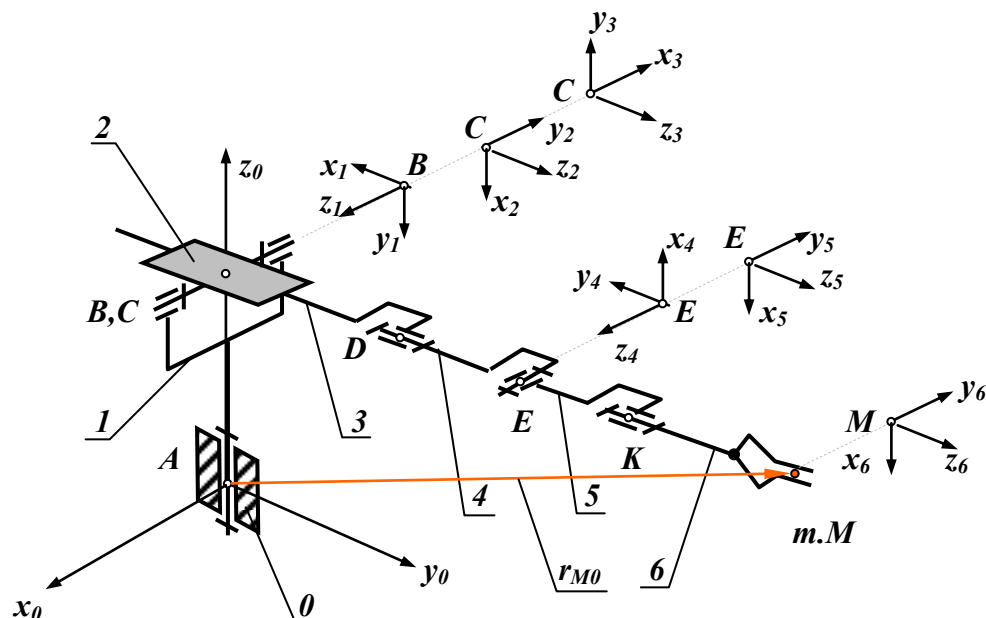


Рисунок 3.2 – Структурна схема маніпулятора

Елементи цієї матриці визначають положення центру захвату точки M і орієнтацію його в просторі. Четвертий стовпець визначає Декартові координати точки M (проекції вектору rM_0 на осі координат). Третій стовпець містить осі z_n системи координат, яка пов'язана з захватом, або вектором підходу \bar{A} , що направляють косинуси, який характеризує напрям губок захвату (рис. 3.3).

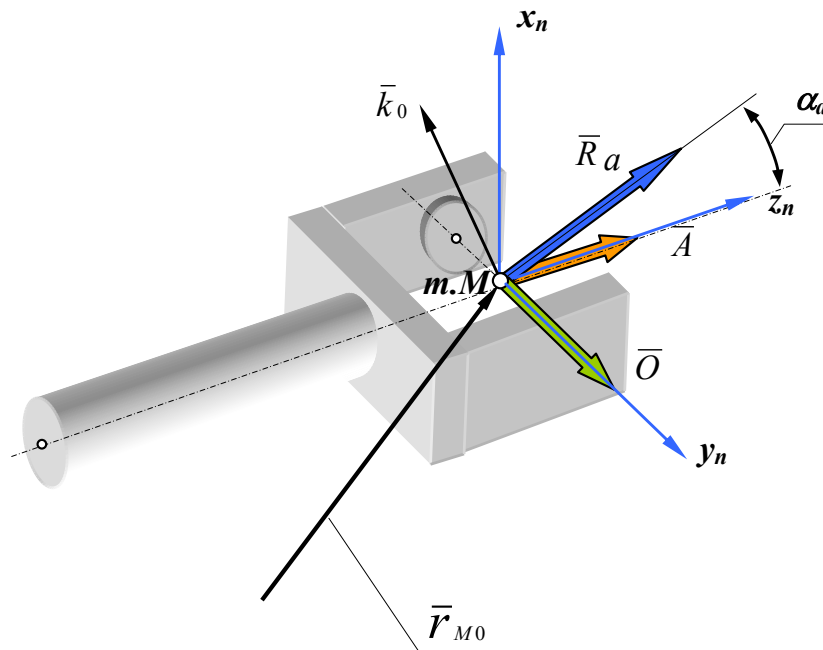


Рисунок 3.3 – Захват маніпулятора

Другий стовпець визначає напрям осі y_n або вектору орієнтації \bar{O} , який проходить через центр захвату за віссю, яка перпендикулярна робочим поверхням його губок. У першому стовпці містяться направляючі косинуси осі x_n або вектору $(\bar{O} \times \bar{A})$. Кутом підходу захвату α називається кут між вектором підходу і базовим вектором

$$R_\alpha = (\bar{O} \times \bar{k}_0), \quad (3.6)$$

де \bar{k}_0 – орт вектору \bar{z}_0 нерухомої або базової системи координат.

З урахуванням цього, матриця T_n може бути представлена в наступному вигляді

$$T_n = \begin{vmatrix} (\bar{O} \times \bar{A})_x & O_x & A_x & r_{nMx} \\ (\bar{O} \times \bar{A})_y & O_y & A_y & r_{nMy} \\ (\bar{O} \times \bar{A})_z & O_z & A_z & r_{nMz} \\ 0 & 0 & 0 & 1 \end{vmatrix} \quad (3.7)$$

В результаті матричних перетворень отримаємо радіус-вектор точки M захвату у функціях узагальнених координат. Зазвичай, за узагальнені координати приймають лінійні і кутові переміщення в кінематичних парах або на вихідних валах приводів маніпулятора. У механізмі з n рухомостями в загальному вигляді функцію положення захвату можна записати таким чином:

$$\bar{r}_{OM} = T_n \cdot \bar{A} = \bar{P}(q_1, q_2, \dots, q_n), \quad (3.8)$$

де q_1, q_2, \dots, q_n – узагальнені координати маніпулятора.

При кінематичному аналізі маніпулятора в прямій задачі необхідно визначити лінійні і кутові швидкості і прискорення захвату за заданих кутових і лінійних узагальнених швидкостях і прискореннях (зазвичай відносних швидкостях і прискореннях в кінематичних парах механізму). У зворотній задачі за заданим законом зміни швидкостей і прискорень захвату визначаються закони зміни швидкостей і прискорень в КП або на вихідних ланках приводів. Рішення прямої задачі кінематики для точки M захвату можна отримати продиференціювавши четвертий стовпець матриці T_n за часом:

$$r_{nM} = \begin{vmatrix} r_{nMx} \\ r_{nMy} \\ r_{nMz} \\ 1 \end{vmatrix}; \quad V_{nM} = \frac{dr_{nM}}{dt} = \begin{vmatrix} V_{nMx} \\ V_{nMy} \\ V_{nMz} \\ 1 \end{vmatrix}; \quad a_{nM} = \frac{d^2 r_{nM}}{dt^2} = \begin{vmatrix} a_{nMx} \\ a_{nMy} \\ a_{nMz} \\ 1 \end{vmatrix};$$

Кутову швидкість і кутове прискорення захвату можна визначити векторним підсумовуванням відносних кутових швидкостей в обертальних КП механізму. Оскільки вектору кутових швидкостей, при цьому виборі орієнтації осей координат, співпадають з віссю z , то кутова швидкість захвату

$$\bar{\omega}_n = \sum_{i=1}^m \bar{k}_{i-1} \cdot \omega_{i,i-1}, \quad (3.9)$$

де \bar{k}_{i-1} – орт осі z системи координат, розташованої в центрі КП, яка поєднує ланку i та ланку $i - 1$, m – число обертальних КП в механізмі.

Диференціюючи цей вираз за часом, отримаємо формулу для визначення кутового прискорення захвату

$$\bar{\varepsilon}_n = \sum_{i=1}^m \bar{k}_{i-1} \cdot \varepsilon_{i,i-1} + \sum_{j=2}^m \left[\left(\sum_{k=l-1}^{m-1} \bar{k}_{k-1} \cdot \omega_k \right) \times \left(\bar{k}_{j-1} \cdot \omega_j \right) \right]. \quad (3.10)$$

3.3 Динаміка маніпуляторів промислових роботів

З великої різноманітності задач динаміки маніпуляторів розглянемо дві: силовий розрахунок і розрахунок швидкодії промислових роботів. При силовому розрахунку маніпуляторів вирішується задача визначення зовнішніх силових дій, що управляють та забезпечують необхідний закон руху механізму, і розрахунку реакцій в кінематичних парах. Першу частину часто називають завданням синтезу управління. При силовому розрахунку зазвичай застосовується метод кінетостатики, ґрунтований на принципі Д'Аламбера. За цим методом до зовнішніх сил і моментів, які прикладені до ланок механізму, додаються розрахункові сили інерції, які забезпечують силову урівноваженість системи і дозволяють розглядати рухливу систему в квазістатичній рівновазі, тобто, як умовно нерухому. Силовий розрахунок виконується за заданим корисним навантаженням \bar{F}_n , відомих законах руху ланок \bar{a}_{si} і $\bar{\varepsilon}_i$ (з попереднього кінематичного розрахунку), відомих інерційних

характеристиках ланок: масах ланок m_i і їх моментах інерції I_{si} . За цими даними визначаються головні вектору $\bar{F}_{ui} = -m \cdot \bar{a}_{si}$ і головні моменти $\bar{M}_{ui} = -I_{si} \cdot \bar{\varepsilon}_i$ сил інерції для кожної з ланок механізму. Для відкритого кінематичного ланцюга рішення розпочинаємо з вихідної ланки – захвату. Відкинуті зв'язки ланки n з ланкою $n - 1$ і вихідним валом приводу ланки n замінюємо реакціями $\bar{M}_{n,n-1}$ і $\bar{F}_{n,n-1}$ і складаємо кінетостатичні векторні рівняння рівноваги сил і моментів для ланки n (рис. 3.4):

$$\bar{G}_o + \bar{G}_n + \bar{F}_{un} + \bar{F}_{uo} + \bar{F}_{n,n-1} + \bar{F}_n = 0, \quad (3.11)$$

$$\bar{M}(\bar{G}_n) + \bar{M}(\bar{G}_o) + \bar{M}(\bar{F}_{un}) + \bar{M}(\bar{F}_{uo}) + \bar{M}_{un} + \bar{M}_{n,n-1} + \bar{M}(\bar{F}_n) = 0, \quad (3.12)$$

де $\bar{M}_{n,n-1}$ – вектор моменту в кінематичній парі (проекція цього вектору на вісь $z \in$ рушійним моментом приводу в КП, тобто $\bar{M}_{z(n,n-1)} = \bar{M}_{\partial(n,n-1)}$).

Проектуючи векторні рівняння на вісь координат, отримуємо систему шести рівнянь алгебри, звідки чому визначимо шість невідомих

$$F_{x(n,n-1)}, F_{y(n,n-1)}, F_{z(n,n-1)}, M_{x(n,n-1)}, M_{y(n,n-1)}, M_{z(n,n-1)} = M_{\partial(n,n-1)}. \quad (3.13)$$

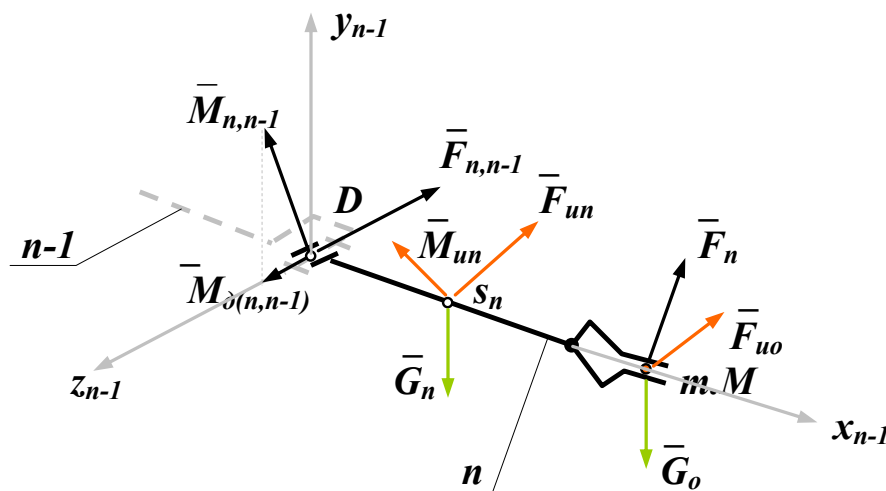


Рисунок 3.4 – Схема для проведення силового розрахунку важеля

Далі розглядається рівновага ланки $n - 1$. При цьому в місці його приєднання до ланки n прикладаються реакції з боку ланки n

$$F_{x(n-1,n)}, F_{y(n-1,n)}, F_{z(n-1,n)}, M_{x(n-1,n)}, M_{y(n-1,n)}, M_{z(n-1,n)} = M_{d(n-1,n)}. \quad (3.14)$$

рівні за величиною і протилежні за напрямом реакцій, визначеним на попередньому етапі розрахунку. Так послідовно складаються рівняння силової рівноваги для усіх n ланок механізму. З рішення отриманої системи $6n$ рівнянь визначаються реакції в кінематичних парах, рушійні сили і моменти.

3.3.1 Розрахунок швидкодії промислового робота

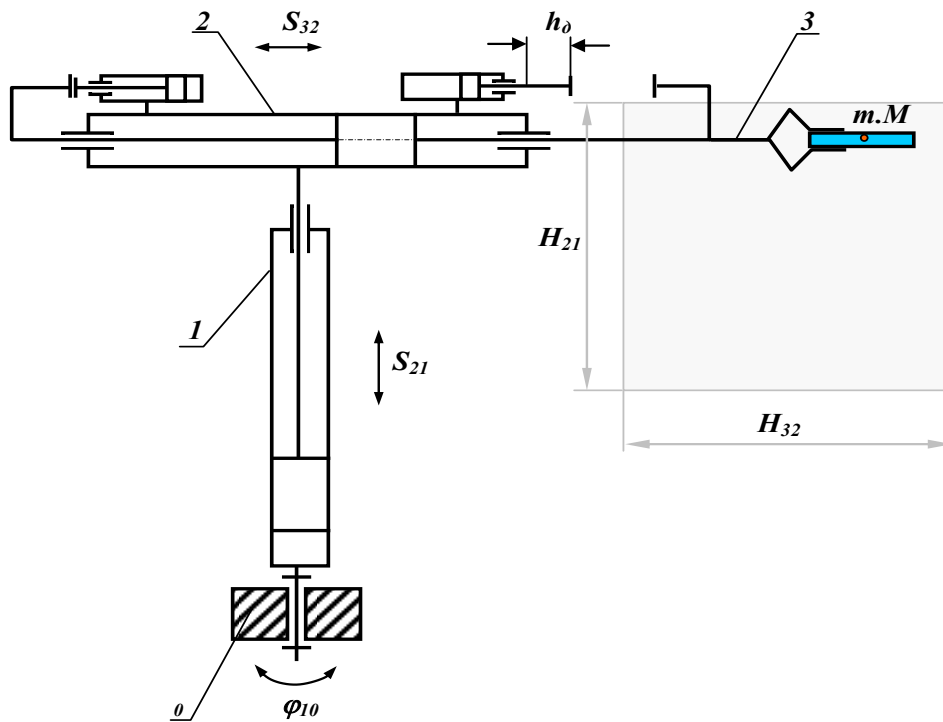
Час виконання роботом циклу переміщень деталі багато в чому визначає продуктивність усього роботизованого комплексу. Тому вимоги до швидкодії робота зазвичай досить високі. Час виконання роботом технологічної операції обумовлений законами зміни зовнішніх сил (рушійних і опори) і інертністю ланок механізму. Закон зміни сил, що управляють, залежить від типу використовуваного приводу і від виду системи управління. Існують роботи з гідравлічними, пневматичними, електричними і комбінованими приводами. У них застосовуються циклові, позиційні або контурні системи управління.

Розглянемо розрахунок швидкодії одного з приводів промислового робота з цикловою системою управління. При цикловій системі управління відносні переміщення ланок обмежуються пересувними упорами і кінцевими вимикачами.

На рис. 3.5 зображена кінематична схема трьохрухливого маніпулятора промислового робота.

На рис. 3.5, б наведена циклограма налаштування командо-апарата (суцільні лінії) і циклограма роботи промислового робота (пунктирні лінії). Загальний час робочого циклу $T_{\text{ц}}$ складається з часу простою в заданих положеннях (на циклограмі простій показаний прямими, які паралельні щодо горизонтальній осі t) і часі відносних переміщень ланок з одного заданого положення в інше t_{nx} і назад t_{ox} (похилі прямі на діаграмах). Час простою зазвичай заданий умовами технологічного процесу. Час виконання роботом

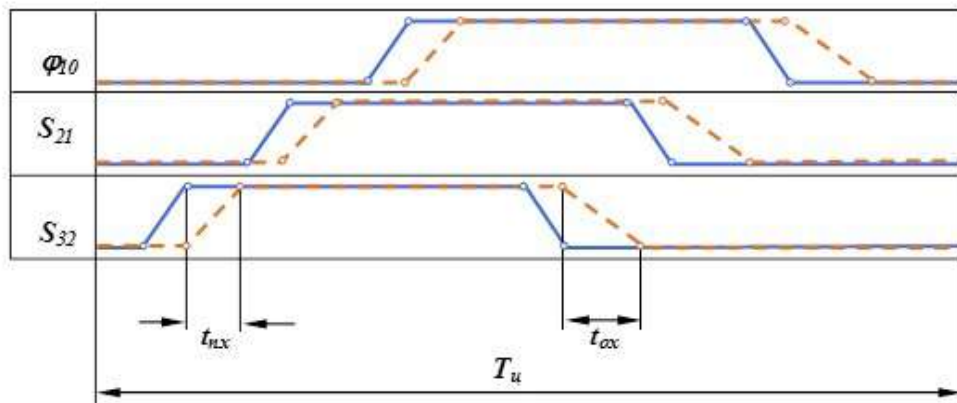
рухів визначається динамічними характеристиками приводів і маніпулятора – рушійними силами і силами опору, масами і моментами інерції ланок.



0 – нерухома ланка; 1, 2, 3 – рухливі ланки

а)

Циклограми командоапарата і промислового робота



б)

Рисунок 3.5 – Структурна схема привода маніпулятора промислового робота та циклограма його роботи

Розглянемо роботу пневмоприводу переміщення руки маніпулятора (рис.3.5). За сигналом від командо-апарата в праву порожнину

циліндра подається стисле повітря, яке діє на поршень з силою $F_{\partial 3} = pS_n$ де p – тиск повітря, S_n – активна площа поршня. Під дією цієї сили поршень і рука 3 переміщуються вліво з постійним прискоренням і з зростаючою швидкістю V_{32} (рис.3.6, а). Обмеження ходу поршня може здійснюватися або жорстким упором без демпфера, або упором з демпфером.

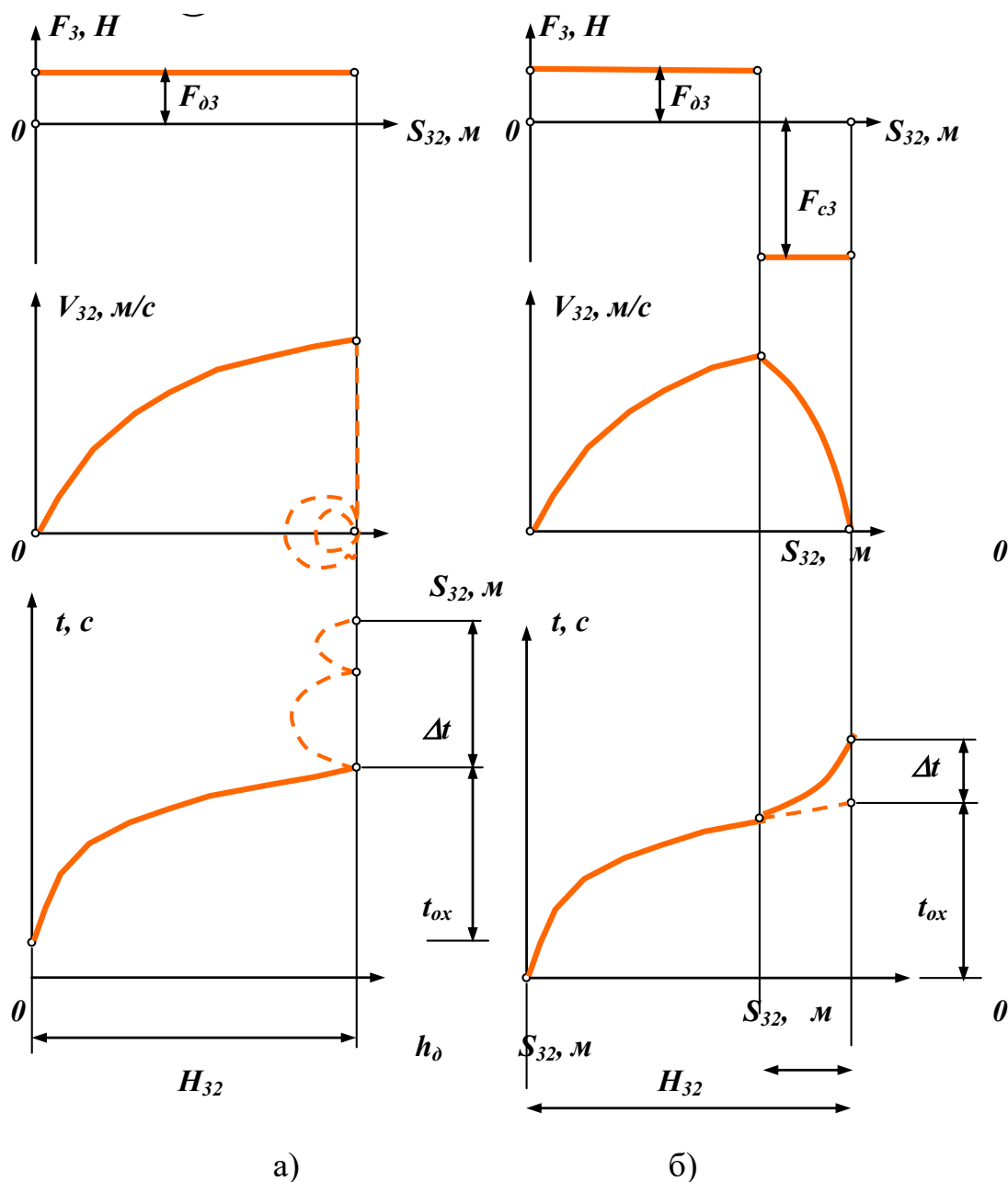


Рисунок 3.6 – Кінематична характеристика переміщення руки робота:
 а) упор без демпфера; б) упор з демпфером

З зупинкою на упорі без демпфера, швидкість ланки 3 повинна миттєво зменшитися з деякого кінцевого значення до нуля. За такою змінною швидкості прискорення $a_{32} \Rightarrow -\infty$. Така зупинка ланки називається жорстким ударом. Вона супроводжується великими динамічними навантаженнями на ланки механізму. Оскільки реальний маніпулятор є пружно-інерційною системою, то ці навантаження викликають відскік ланки 3 від упору, а також коливання усього механізму. Захват здійснюватиме коливання відносно заданого кінцевого положення. Час загасання цього процесу Δt (рис.3.6, а) значно знижує швидкодію промислового робота.

Зменшити ці коливання або взагалі виключити їх можна, якщо

$$V_{32n} = 0, \quad a_{32n} = 0; \quad (3.15)$$

де V_{32n}, a_{32n} – відносна швидкість і відносне прискорення ланок у момент зупинки. Проте це здійснено тільки в регульованому приводі за контурним управлінням. Крім того за без ударної зупинки у кінці ходу відносна швидкість близька до нуля, тому час переміщення захвату в необхідне положення значно зростає. Компромісним рішенням є зупинка з м'яким ударом, за якої відносна швидкість у кінці ходу $V_{32n} = 0$, а прискорення обмежене деяким допустимим значенням $a_{32n} \leq [a]$.

У механізмах з цикловим управлінням режим руху з м'яким ударом забезпечується встановленням упорів з демпферами, що гасять кінетичну енергію руки. Розрахунок демпфера ведеться з умови $A\Sigma n = 0$, яке забезпечується рівністю за цикл руху роботи рушійної сили $AF_{\partial 3}$ і роботи сили опору демпфера AF_c (рис. 3.6, б):

$$AF_{\partial 3} = -AF_c \quad \text{або} \quad F_{\partial 3} \cdot (H_{32} - h_{\partial}) = -F_c \cdot h_{\partial}. \quad (3.16)$$

У цьому виразі невідомі дві величини F_c і h_{∂} , однією з них задають, іншу – розраховують.

3.4 Системи управління з програмованими логічними контролерами

3.4.1 Автоматизовані виробничі системи

Автоматизація об'єкту управління, яким може виступати машина або її частина, окремий апарат або будь-який інший технічний пристрій, полягає в забезпеченні роботи цього об'єкту за допомогою облаштування управління. Створювана з цією метою автоматизована система функціонуватиме за заданим алгоритмом з урахуванням параметрів стану об'єкту управління. При цьому людина (оператор) не бере участі в безпосередньому управлінні роботою об'єкту, а тільки контролює хід процесу, маючи можливість втручання в процес управління. На рис. 3.7 наведена структурна схема автоматизованої системи управління (АСУ) автономним об'єктом.

Як впливає з рис. 3.7, об'єкт управління і облаштування управління утворюють замкнутий контур, за яким безперервно циркулює інформація різного функціонального призначення. З одного боку, в облаштування управління надходить інформація від датчиків про стан об'єкта, отримана шляхом виміру параметрів процесу, а з іншої – від облаштування управління на об'єкт поступають дії, які управляють, та за допомогою виконавчих механізмів змінюють стан об'єкту відповідно до алгоритму управління. Оператор, який знаходиться у пульта, може спостерігати за роботою об'єкту управління завдяки різним засобам відображення інформації (звукова і світлова сигналізація, мнемосхеми, дисплеї тощо) і за необхідністю втручатися в робочий процес (наприклад, з відхиленням від розрахункового режиму роботи).

Розглянута структурна схема АСУ носить назву однорівневої системи, вона реалізує тільки один рівень управління. Сучасні ж виробничі системи управління об'єднують в собі цілі групи об'єктів, усередині яких реалізуються закінчені технологічні процеси. Створення автоматизованих систем управління технологічними процесами стало можливим завдяки появі двох- і багаторівневих структур, що управляють, в яких обчислювальні ресурси розподіляються як між об'єктами одного рівня, так і між окремими рівнями управління, побудованими за ієрархічним принципом. При цьому

безпосереднє управління технологічним процесом і устаткуванням реалізується на самому нижньому рівні управління, де кожному об'єкту виділяється незалежна кінцева (термінальна) система управління. Зв'язок між рівнями зазвичай здійснюється за одним із стандартних інтерфейсів за допомогою мережевих комунікацій.



Рисунок 3.7 – Структурна схема автоматизованої системи управління автономним об'єктом

У загальному випадку необхідний алгоритм управління може бути реалізований системою управління з жорсткою структурою на основі електромагнітних пристроїв (реле) або електронних схем, елементи яких сполучені один з одним так, щоб безпосередньо забезпечують рішення поставленої задачі. Подібне рішення називають схмотехнікою або апаратним. Реалізація ж іншого алгоритму цією системою неможлива без зміни її структури і складу технічних засобів.

У системах управління з гнучкою структурою існує можливість зміни порядку взаємодії різних схем за допомогою дій, що управляють. Послідовність таких дій на кожному робочому кроці визначає функції, що виконуються відповідною апаратурою, і дає можливість забезпечити рішення будь-якої, навіть найскладнішої задачі. Таке рішення називають програмною

реалізацією алгоритму управління. Послідовність дій, що реалізують алгоритм управління, утворює програму.

Технічну основу сучасних систем управління виробничими процесами складають програмовані пристрої управління. При цьому широке поширення отримали системи управління, нижній рівень яких включає програмовані логічні контролери (ПЛК) (рис. 3.8).



ОУ – об'єкт управління; ПЛК1, ПЛК2 – програмовані логічні контролери;
Д1, Д2, ... – датчики; ВП1, ВП2, ... – виконавчі пристрої

Рисунок 3.8 – Нижній рівень ієрархічної системи управління

Згідно з наведеною схемою контролери ПЛК отримують сигнали безпосередньо від датчиків Д1, Д2, .., встановлених на об'єкті управління ОУ тим самим забезпечуючи збір і обробку даних, а також видають сигнали управління безпосередньо на виконавчі пристрої ВП1, ВП2, ..., які, у свою чергу, впливають на об'єкт управління для приведення його в потрібний стан (положення). В якості ОУ нижнього рівня зазвичай розглядають об'єкти, алгоритм роботи яких повністю визначений (наприклад, приводи, механізми, вимірювальні прилади тощо).

Системи управління виробничими процесами нижнього рівня виконують наступні загальні типові завдання:

- зберігання і реалізація загального алгоритму управління;
- локальне логічне управління (сигнали управління дискретні);
- локальне аналогове управління (формується дії, що управляють, із заданими динамічними показниками);
- зв'язок з оператором;
- комунікаційні обміни даними.

Виконання завдання зберігання і реалізації загального алгоритму управління забезпечується шляхом паралельного рішення сукупності підзадач, серед яких можна назвати наступні: прийом і зберігання даних і програм, що управляють; виконання системних і прикладних програм, що відображають алгоритми і закони управління; логічні і математичні обчислення; реалізація функцій таймера і лічильника; технічна діагностика тощо. У виробничих системах найчастіше зустрічаються логічні процеси управління, інформація в яких має дискретний або двійковий характер. Такі процеси формуються за логічним принципом:

ЯКЩО ...,

ТО ...,

ІНАКШЕ ...,

де замість точок слід підставляти умови і наслідки, наприклад:

ЯКЩО шток пневмоциліндра повністю висунений, ТО включити привід конвеєра,

ІНАКШЕ (тобто шток пневмоциліндра не зайняв потрібне положення) не вмикати привід конвеєра.

Локальне логічне управління технологічним устаткуванням дискретної дії зводиться до формування сигналів, що управляють, для виконавчих пристроїв з релейною характеристикою. Іншими словами, в цьому випадку контролер реалізує елементарні дії типу «увімкнути», «вимкнути», «реалізувати затримку часу» тощо. Подібні дії поступають на об'єкт управління в деякі дискретні моменти часу у відповідності з алгоритмом управління, утворюючи певну тимчасову послідовність. У якості прикладу на рис. 3.9, а наведено послідовність трьох керівників дій, забезпечують

ввімкнення деяких виконавчих пристроїв на певні проміжки часу: пристрій 1 вмикається на період $t_1...t_3$, пристрій 2 – на $t_2...t_4$ і пристрій 3 – на $t_5...t_6$.

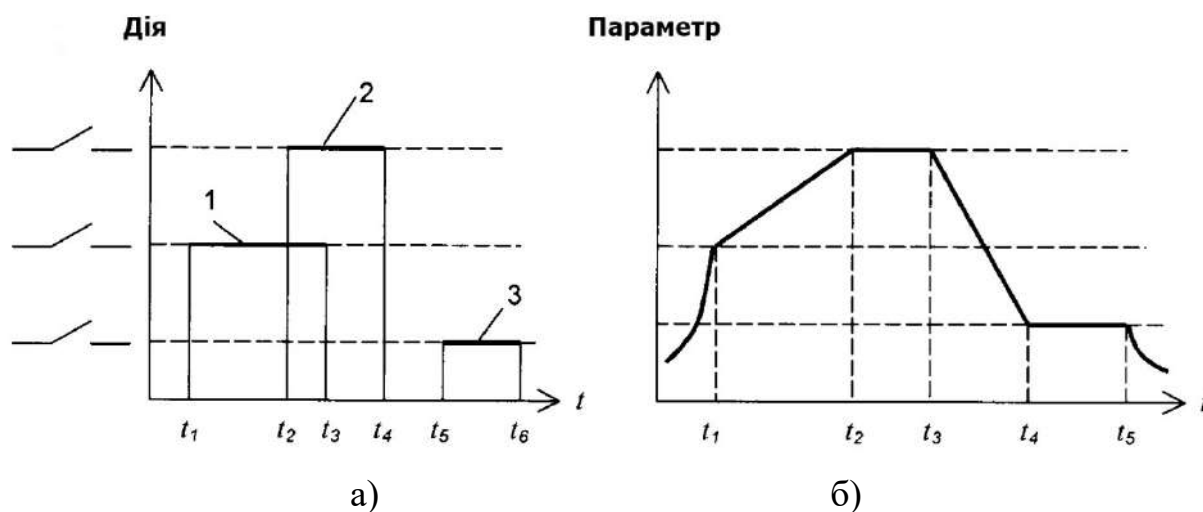


Рисунок 3.9 – Типові функції управління технологічним устаткуванням:

а) дискретної дії; б) безперервної дії

Локальне аналогове управління технологічним устаткуванням безперервної дії забезпечує виконавчі дії із заданими динамічними показниками: стійкістю, часом перехідного процесу, динамічними помилками тощо. Для безперервних технологічних процесів характерним є алгоритм регулювання параметрів, визначаючих процес. Програмований контролер забезпечує необхідну зміну регульованих параметрів в часі за допомогою прямого цифрового управління, як показано, приміром, на рис. 3.9, б: до настання моменту часу t_1 процес некерований, в період $t_1...t_2$ має місце контрольоване лінійне збільшення значення параметра, в період $t_2...t_3$ його величина підтримується на постійному рівні, впродовж часу $t_3...t_4$ здійснюється кероване лінійне зменшення значення параметра, далі знову йде період стабілізації $t_4...t_5$, і нарешті у момент часу t_5 процес управління припиняється. З метою реалізації функції безперервного регулювання контролер забезпечується модулями аналого-цифрового і цифро-аналогового перетворень.

Рішення задачі зв'язку з оператором зводиться до обирання засобів відображення і введення інформації. У сучасних системах управління зазвичай використовуються засоби людино-машинного інтерфейсу (НМІ, Human

Machine Interface) – текстові і графічні панелі оператора, які випускаються виробниками контролерів. Поширення отримали також системи на основі персональних комп'ютерів, що дозволяють за допомогою програмного інструментарію скласти віртуальні спеціалізовані пульти управління та індикації.

Завдання комунікаційних обмінів даними вирішується залежно від рівня обміну шляхом вибору одного з існуючих стандартних інтерфейсів. Для роботи в мережі, що становить один рівень, найчастіше використовується послідовний інтерфейс, наприклад RS-232 (за схемою «точка-точка») або RS-485 (за багатоточковою схемою). Для вирішення завдання зв'язку з обчислювальними ресурсами верхнього рівня застосовують магістральний інтерфейс, наприклад PROFIBUS, і відповідні програмно-апаратні засоби. Здатність програмованих контролерів працювати в мережі стрімко підвищує ефективність їх застосування з метою автоматизації виробничих процесів, дозволяючи створювати децентралізовані системи управління.

3.4.2 Будова і робота програмованих логічних контролерів

Програмований логічний контролер (ПЛК) є універсальним засобом автоматизації виробничих процесів. Його застосування в автоматизованих системах управління забезпечує високоефективну обробку інформації, що надходить від об'єкту управління, і реалізацію управляючих дій.

Контролером в широкому значенні цього слова називають будь-який пристрій управління яким-небудь технічним об'єктом. У більш вузькому сенсі під контролером розуміють пристрій управління, який працює за певним алгоритмом (програмою). Подібний пристрій, виконаний із застосуванням інтегральної технології, отримав назву мікроконтролер. Можливість застосування модулів пам'яті з метою зберігання програм дозволила змінювати алгоритм його роботи для вирішення конкретного завдання, що призвело до появи програмованих мікроконтролерів.

Програмований мікроконтролер є спеціалізованою мікроЕОМ, яка управляє, та працює в режимі реального часу за певною робочою програмою, розміщеної в її пам'яті. Ці пристрої, орієнтовані на реалізацію алгоритмів

логічного типу, пізніше стали називати програмованими логічними мікроконтролерами або просто програмованими логічними контролерами.

Сучасні ПЛК характеризуються невисокою вартістю, малими габаритами, невеликою споживаною потужністю і високою надійністю, а їх конструктивне виконання припускає використання контролерів у важких виробничих умовах безпосередньо в об'єкті управління. В порівнянні з традиційними релейними пристроями з жорсткою структурою управління (жорсткою логікою) ПЛК відрізняються розвиненими функціональними можливостями, що забезпечують простоту зміни алгоритму управління при зміні технологічного процесу.

У складі ЦПУ можна виділити наступні функціональні вузли (рис. 3.10):

- центральний процесор (ЦПУ);
- пристрій пам'яті програм, що запам'ятовує (ПЗП, постійний пристрій, що запам'ятовує);
- облаштування пам'яті даних, що запам'ятовує (ОЗП, оперативний пристрій, що запам'ятовує);
- контролери вводу/виводу;
- комунікаційний процесор.

Для різних за складністю й швидкодії ПЛК контролери вводу/виводу, а також комунікаційні процесори можуть бути реалізовані за допомогою як апаратних, так і програмних засобів. Центральний процесор є основним вузлом будь-якої програмованої системи управління технологічними процесами. Усі операції, що реалізуються в такій системі, ініціюються ЦПУ, хоча іноді вони можуть виконуватися і без його безпосередньої участі. Слід зазначити, що центральний процесор ПЛК має ряд особливостей, що відрізняють його, наприклад, від персонального комп'ютера.

ЦПУ перших моделей ПЛК будувалися як однобітовий логічний процесор, який реалізує в основному логічні операції над дискретними сигналами. Подібна побудова процесора дозволяла використати прості мови програмування типу мови релейно-контактних схем або мови логічних рівнянь булевої алгебри, що, у свою чергу, спрощувало програмування контролерів і не вимагало від користувача спеціальної підготовки в області програмування ЕОМ.

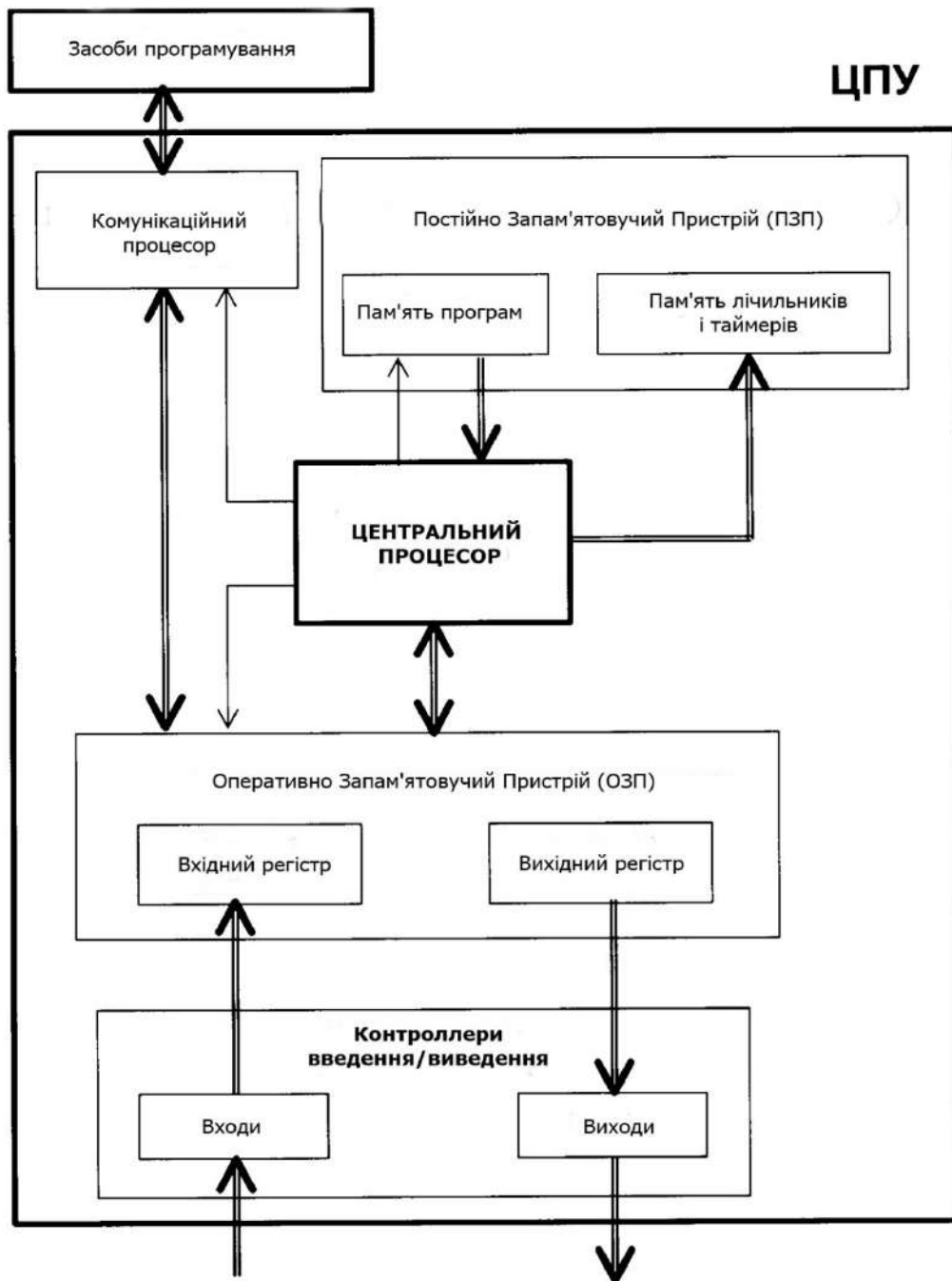


Рисунок 3.10 – Узагальнена структура програмованого логічного контролера (подвійні стрілки означають зв'язки за допомогою передачі даних, одинарні – за сигналами управління)

З іншого боку, сфера застосування контролерів обмежувалася тільки дискретними об'єктами. Сучасні ж ПЛК підтримують не лише логічні, але і арифметичні операції, що дозволяє здійснювати разом з однорозрядною послідовною (побітовою) обробкою вхідних даних також їх паралельну

обробку у вигляді машинних слів. Це надає можливість за допомогою додаткових зовнішніх модулів обробляти аналогові сигнали від об'єкту управління і реалізовувати прості системи автоматичного регулювання.

Постійний пристрій, що запам'ятовує (ПЗП), і має можливість перезапису, разом з оперативним пристроєм, що запам'ятовує (ОЗП), складають центральну пам'ять ЦПУ. У пам'яті програм контролера у вигляді послідовності команд записана програма управління об'єктом. Ця пам'ять організована в словах, оскільки вона повинна містити команди. У тій самій області пам'яті одночасно може бути записані декілька програм. У пам'яті даних зберігається стан однорозрядних (бітових) входів, виходів і внутрішніх елементів. На відміну від інших пристроїв програмованої логіки центральна пам'ять ПЛК спеціалізована. Вона розділена на фіксовані області, використовувані для зберігання конкретних видів інформації. Спеціалізація пам'яті дозволяє повністю звільнити користувача від реалізації операцій управління входами/виходами в створюваній ним програмі, що є значною перевагою ПЛК в порівнянні з іншими програмованими пристроями логічного управління.

Сигнали від дискретних датчиків на об'єкті управління у вигляді наборів двійкових величин надходять на контролер введення, який здійснює гальванічну розв'язку сигналів, нормалізацію, масштабування і передачу даних в оперативну пам'ять, де вони і зберігаються в процесі їх обробки. Сигнали управління також у вигляді дискретних сигналів фіксуються на виході контролера і через пристрій сполучення впливають на об'єкт. При роботі з аналоговими сигналами використовуються спеціальні модулі розширення, аналого-цифрові і цифро-аналогові перетворювачі. Обмін даними між центральним процесором і модулями розширення здійснюється під час використання загальних шин.

Програмуючий пристрій (програматор), за допомогою якого реалізуються завантаження програм в пам'ять ПЛК і налаштування системи управління, виконується автономним і часто переносним, тому що програмовані контролери зазвичай не вимагають постійної присутності обслуговуючого персоналу.

Комунікаційний процесор призначений для обслуговування каналів зв'язку ПЛК з програматорами, з провідною ЕОМ, з іншими контролерами або локальними обчислювальними мережами (ЛВМ). Інформація, що поступає після каналів зв'язку, розміщується безпосередньо в пам'яті.

Сучасні програмовані логічні контролери включають до свого складу таймери і лічильники, які виконують функції затримки часу і рахунку і є обов'язковим елементом будь-якої системи, що управляє, та працюючої в режимі реального часу. Зокрема, широко використовується можливість програмного зчитування потокового часу таймерів системи управління або завдання в програмі, що управляє часових інтервалів. Лічильники застосовують, наприклад, для підрахунку кількості різного роду подій, обліку готової продукції тощо. Зазвичай таймери і лічильники реалізуються програмно за допомогою базового програмного забезпечення контролера і у разі використання в програмі, яка використовується, конфігуруються користувачем.

Робота ПЛК здійснюється в циклічному режимі. Фази робочого циклу можна описати таким чином, як це представлено в таблиці. 3.1.

Таблиця 3.1 – Фази робочого циклу ПЛК

Фаза	Зміст процесу
1	Внутрішня діагностика, перевірка і тестування контролера
2	Опитування вхідних регістрів і введення даних
3	Виконання команд програми, яка використовується
4	Оновлення вихідних даних і їх виведення
5	Обслуговування периферійних пристроїв і каналів зв'язку

Виконання програми полягає у формуванні вихідних сигналів, що управляють, і передачі їх на відповідні виходи контролера. При цьому процесор відповідно до програм опитує усі свої вхідні регістри, здійснює логічну обробку даних і за її результатами видає вихідні сигнали управління на ті або інші виконавчі органи. Процесор послідовно, рядок за рядком, опитує (сканує) пам'ять програм, робить логічні обчислення відповідно до записаного алгоритму і заносить отримані значення в пам'ять даних. Після завершення

сканування усієї пам'яті процесор забезпечує обмін даними між вхідними і вихідними регістрами і пам'яттю даних, і потім сканування пам'яті програм повторюється знову від початку до кінця, тобто процеси сканування пам'яті і обміну даними періодично повторюються. Одноразовий прохід процесора за усією програмою називають циклом повного сканування пам'яті або просто циклом. Найбільш поширені цикли відпрацювання програм ПЛК наведені на рис. 3.11 (показані тільки фази прийому, обробки і виведення даних).

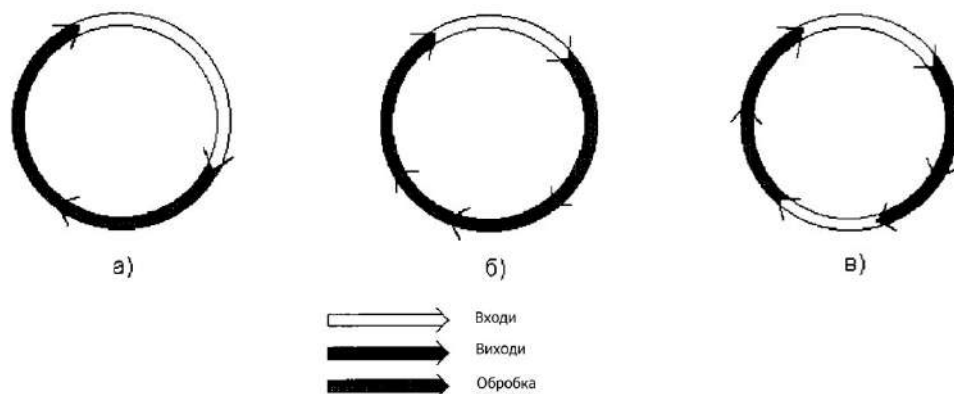


Рисунок 3.11 – Типові цикли ПЛК : а) елементарний цикл;
б) цикл з розподілом виходів; в) цикл з розподілом входів і виходів

Цикл роботи контролера може будуватися за наступними схемами:

а) на самому початку циклу робиться вибірка (опитування) усіх входів; виходи збуджуються (тобто формуються сигнали управління) лише після закінчення рішення усіх рівнянь;

б) на початку циклу робиться вибірка усіх входів; виходи збуджуються за запитом (наприклад, після закінчення рішення кожного рівняння);

в) в циклі по черзі здійснюються зчитування значень змінних, проведення пов'язаної з цією дією логічної обробки даних і збудження виходів.

Час, за який здійснюється цикл роботи, називають часом циклу сканування. Час сканування є одним з основних характеристик ПЛК, оскільки він визначає швидкість його реакції на зміну стану об'єкту управління. В сучасних програмованих контролерах цей час безперервно контролюється за допомогою сторожового таймера (Watch Dog Timer). Цикл рахунку

сторожового таймера встановлюється більшим, ніж цикл сканування контролера, і таймер програмно скидається і запускається в кожному черговому циклі. Під час збоїв у роботі програми або відсутності будь-якого вхідного сигналу, час сканування може перевищити час рахунку сторожового таймера, внаслідок чого останній подасть аварійний сигнал внутрішнього переривання, який використовується для зупинки контролера і повідомлення про аварійну ситуацію, що виникла.

3.4.3 Програмне забезпечення ПЛК

В процесі роботи автоматизованої системи управління об'єктом мають місце два види обміну інформацією:

- постійний обмін даними між об'єктом і комплексом засобів управління (через канали введення і виведення інформації);
- обмін даними між оператором і системою управління в міру необхідності (коригування операцій з обробки даних, зміна установок, перехід в режим ручного управління та ін.).

Для здійснення вказаних видів обміну інформацією слугує програмне забезпечення (ПЗ), яке зазвичай підрозділяється на базове (основне) і прикладне (функціональне).

Базове ПЗ включає комплекс програм, що поставляються разом з технічними засобами і які призначені для вирішення питань, пов'язаних з експлуатацією цих засобів. Зокрема, до базового ПЗ відноситься програма, що управляє усіма ресурсами контролера і забезпечує циклічний характер його роботи.

У функціональному ПЗ користувача міститься програмне забезпечення, необхідне для розробки і виконання програм управління. Воно представляється виключно важливим для користувача, оскільки визначає трудомісткість складання програм і ефективність їх роботи. Програмований контролер, як і будь-який інший пристрій на базі мікропроцесора, розуміє тільки мову машинних команд. Ця мова складна і незручна для розробки програм. Крім того, вона вимагає від користувача спеціальних знань в області інформатики. Тому експлуатація ПЛК пов'язана з використанням спеціальних мов, доступних фахівцям в області автоматички, а не інформатики. Ці мови є

проблемно-орієнтованими, тому що конструктивне виконання програмованого контролера спеціалізоване для вирішення конкретних завдань, і розроблялися вони таким чином, щоб максимально полегшити перехід користувачів від жорсткої (апаратної) логіки до програмної.

Розрізняють наступні типи мов програмування ПЛК:

- мова релейно-контактних схем;
- мнемонічна мова;
- мова логіки;
- мова логічних схем.

Мова релейно-контактних схем призначена для програмування алгоритмів, тотожних звичайним релейно-контактним схемам промислової автоматики. Тому вона дуже зручна у випадках модернізації устаткування, коли є готова принципова схема і, відповідно, потрібно замінити релейну схему на ПЛК. Ця мова є графічною мовою, в якій і функції, що управляють, і умови їх виконання зображуються за допомогою спеціальних графічних елементів, що нагадують елементи релейно-контактних схем (рис. 3.12).

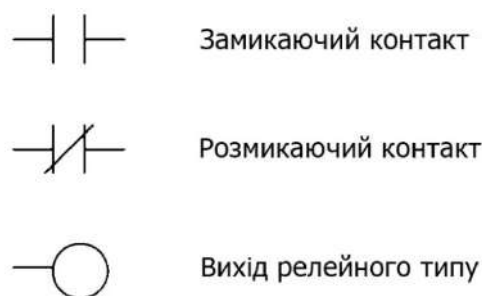


Рисунок 3.12 – Основні елементи релейно-контактних схем

При програмуванні кожен логічний ланцюжок електричних контактів, що визначає умови виконання тієї або іншої функції управління (умови включення відповідних реле), замінюється обчислювальним ланцюжком, що управляє, який є послідовними і паралельними з'єднаннями контактів (ключів) і закінчується виходом релейного типу.

Контакти обчислювальних ланцюжків є операндами логічних операцій (замикаючий контакт – пряме значення операнда, розмикаючий контакт –

інверсне), а вихід релейного типу визначає функцію, що управляє. Логічні операції реалізуються в обчислювальних ланцюжках за допомогою послідовних і паралельних з'єднань контактів. Послідовне з'єднання контактів дає функцію логічного добутку, паралельне – функцію логічного складання.

Нехай, наприклад, потрібно забезпечити програмну реалізацію релейно-контактної схеми, зображеної на рис. 3.13, якій відповідає логічне рівняння

$$Y = (X_1 + X_2) \cdot \bar{X}_3 \quad (3.17)$$

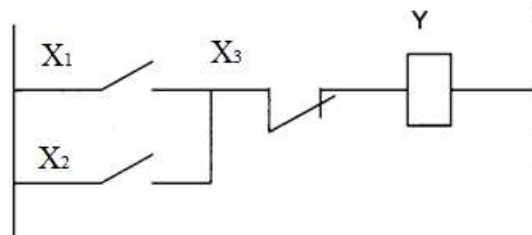


Рисунок 3.13 – Фрагмент релейно-контактної схеми

Ця схема працює таким чином. Якщо замкнутий контакт X_1 або контакт X_2 і не розімкнений контакт X_3 , то обмоткою реле Y піде струм і реле спрацює, виконуючи необхідні перемикання в електричних ланцюгах.

На рис. 3.14 наведено обчислювальний ланцюжок програми для контролера фірми «Siemens» (Німеччина), що моделює розглянутий фрагмент релейно-контактної схеми.

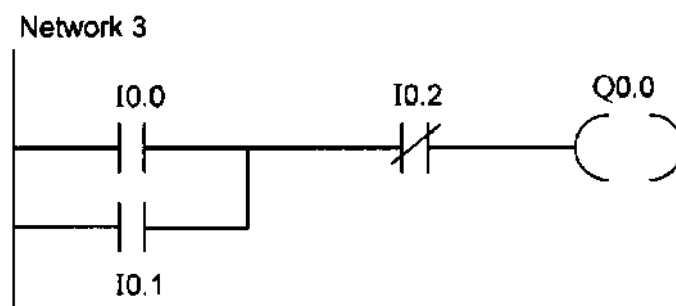


Рисунок 3.14 – Приклад обчислювального ланцюжка

У наведеному прикладі контакти є сигналами X_1 , X_2 і X_3 , що подаються на входи контролера, яким присвоєні адреси пам'яті I0.0, I0.1 і I0.2, а вихід ланцюжка відповідає дискретному виходу контролера з адресом Q0.0. Логіка роботи ланцюжка проста: якщо є сигнал на вході I0.0 ($X_1=1$) АБО на вході I0.0 ($X_2=1$) з одночасною відсутністю сигналу на вході I0.1 ($X_3=1$; I-NE), то формується одиничний сигнал на релейному виході Q0.0 ($Y=1$).

Призначена для користувача програма є послідовним набором подібних ланцюжків, нагадуючи за структурою сходи. Тому мова релейно-контактних схем має також назву Ladder Diagram («діаграма сходів», або мова «сходової логіки», скорочено LAD). Мова LAD – найбільш поширена мова програмування логічних контролерів.

Мнемонічна мова програмування ПЛК є набором записаних у вигляді мнемонік (певних символічних скорочень) команд і директив, чим нагадує мову асемблера. Команди складають виконувану процесором частину програми, а директиви слугують для підготовки програми і в процесі її роботи виконанню не підлягають. Як приклад подібної мови можна назвати мову програмування STL (Statement List – «перелік інструкцій»).

У мові STL використовується базовий набір команд, перерахованих нижче.

LOAD (LD): завантаження змінної, що означає контакт, в акумулятор процесора.

AND (A): логічний добуток однієї змінної (контакту) та іншої змінної (контакт) – результат дорівнює одиниці, якщо обидві величини дорівнюють одиниці.

OR (O): логічне складання однієї змінної (контакту) з іншою змінною (контактом) – результат дорівнює одиниці, якщо значення будь-якої зі змінних або обох одночасно дорівнює одиниці.

OUTPUT (=): привласнення результату операції певному виходу або інструкції.

Програмна реалізація на мові STL (SIMATIC STEP 7 фірми «Siemens», Німеччина) розглянутого вище фрагменту релейно-контактної схеми матиме наступний вигляд:

NETWORK

```
LD 10.0 // Завантажити значення входу I0.0
OR 10.1 // Логічно скласти зі значенням входу I0.1
AND 10.2 // Логічно помножити на заперечення значення входу I0.2
= Q0.0 // Присвоїти результат виходу Q0.0
```

Тут словом NETWORK позначається блок програми, що містить групу інструкцій для виконання заданих функцій. Інструкції в такій групі записуються і виконуються процесором в певній послідовності, утворюючи обчислювальний ланцюжок. Кожен рядок містить мнемоніку інструкції і операнд, який визначається адресом в пам'яті контролера (у цьому прикладі в якості операндів фігурують входи I0.0, I0.1 і I0.2, а також результат операції, який привласнюється виходу Q0.0). Додаткові поля рядків, відокремлені двома похилими рисами, є коментарями.

У загальному випадку STL-програма є набором подібних блоків і у функціональному відношенні повністю еквівалентна LAD-програмі того ж самого процесу.

Мова логіки дозволяє формувати програму для контролера шляхом використання звичних понять логічних послідовностей. Прикладом може бути мова програмування логічних контролерів типу FPS фірми «Festo» (Німеччина), команди якого нагадують оператори відомої мови програмування BASIC і включають такі слова, як ЯКЩО, ТОДІ, ІНАКШЕ, КРІМ ТОГО та ін. Програма на подібній мові логіки для фрагмента схеми на рис. 3.14 виглядатиме таким чином:

```
ЯКЩО вхід X1
ЧИ X2
І-НЕ X3
ТОДІ включити Y
ІНАКШЕ вимкнути Y
```

Незважаючи на те що подібні мови прості в освоєнні, широкого поширення вони не набули внаслідок громіздкості програм, особливо у реалізації складних алгоритмів.

На відміну від мови релейно-контактних схем мова логічних схем призначена для переходу від безконтактних схем промислової автоматики до їх програмної реалізації. Він також виявляється графічною мовою, в якій, як і в мові релейно-контактних схем, функції, що управляють, і умови їх виконання зображуються спеціальними графічними логічними елементами. Ці елементи складають логічний базис і реалізують елементарні булеві функції (рис. 3.15).

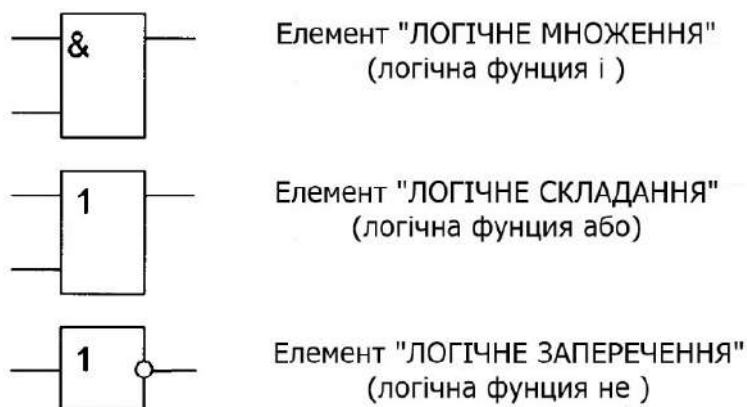


Рисунок 3.15 – Основні графічні елементи мови логічних схем

Представлені логічні елементи мають один або декілька входів і зазвичай один вихід. Їх входи відповідають булевим змінним, а вихід – булевій функції, що реалізовується. У обчислювальних ланцюжках входи одних логічних елементів з'єднуються з виходами інших відповідно до вимог реалізації деякої логічної функції.

На рис. 3.16 показаний приклад обчислювального ланцюжка, що забезпечує програмну реалізацію розглянутої вище релейно-контактної схеми на мові логічних схем для контролера фірми «Omron» (Японія). Ланцюжок побудований безпосередньо за рівнянням початкової релейної схеми.

Мова логічних схем дозволяє легко програмувати обчислювальні ланцюжки, що реалізують цільові функції, для яких відомі логічні рівняння.

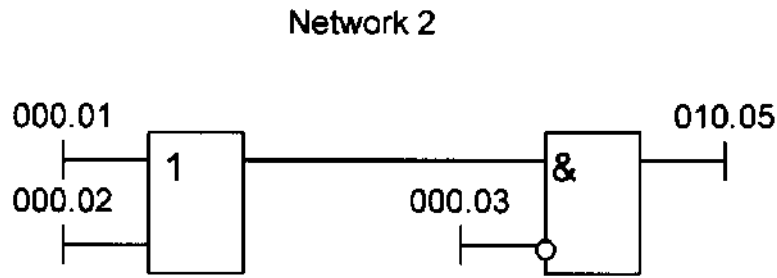


Рисунок 3.16 – Приклад обчислювального ланцюжка у вигляді логічної системи

3.4.4 Технічні засоби систем управління з ПЛК

Система управління з ПЛК є комплексом технічних засобів, вирішальних задач збору і обробки інформації з метою управління процесом в режимі реального часу. Окрім власне програмованого контролера в цей комплекс входять засоби збору інформації і виконавчі пристрої, які з'єднуються з процесором через підсистеми введення і виведення сигналів.

3.4.5 Датчики технологічних систем

Інформація про процес представляється у вигляді даних, які передаються за допомогою вимірювальних сигналів, тобто сигналів, що містять в собі кількісну інформацію про вимірювану величину. Носіями сигналів можуть виступати такі фізичні величини, як електричний струм і напруга, тиск газу і рідини, а також фізичні характеристики електромагнітних коливань тощо.

Розрізняють три види вимірювальних сигналів (рис. 3.17):

- аналогові (інформаційні параметри можуть набувати будь-якого значення в заданому діапазоні або часовому інтервалі);
- дискретні (інформаційні параметри можуть набувати дискретних значень за часом або за рівнем);
- цифрові (інформаційні параметри можуть набувати дискретних значень як за часом, так і за рівнем).

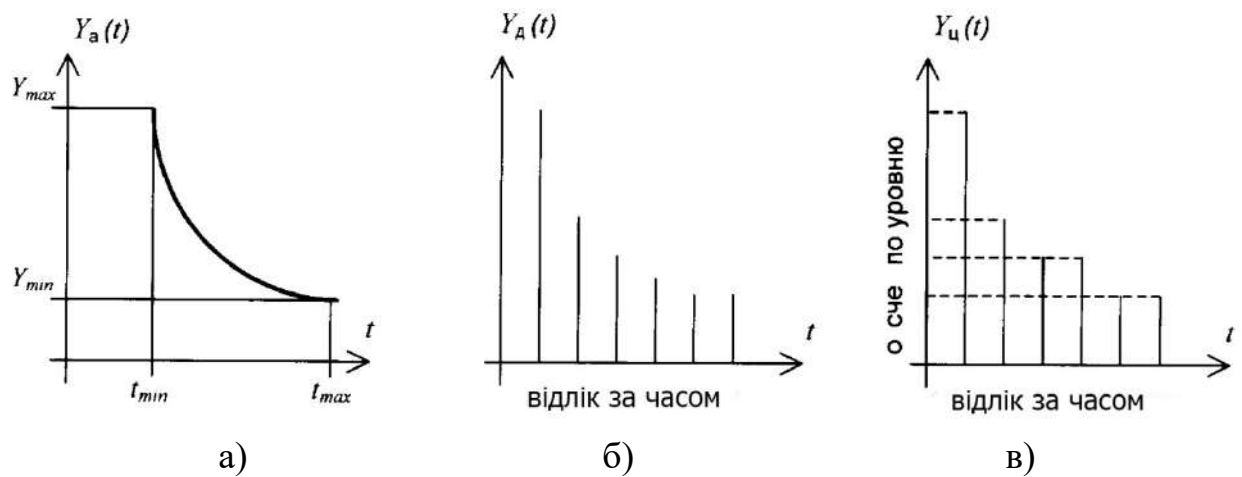


Рисунок 3.17 – Вимірювальні сигнали: а) аналоговий;
 б) дискретний (за часом); в) цифровий

Формування вимірювального сигналу у формі, зручній для передачі, подальшого перетворення, обробки або зберігання, але недоступної для безпосереднього сприйняття людиною, здійснюється вимірювальним перетворювачем. При вимірюванні параметрів технологічного процесу відбувається ряд перетворень одних фізичних величин в інші, які функціонально з ними пов'язані. Перший вимірювальний перетворювач у вимірювальному ланцюзі називають первинним перетворювачем, а усю конструктивну сукупність подібних технічних засобів (розміщуваних безпосередньо в об'єкті вимірювань) – датчиком.

В АСУТП застосовуються датчики неелектричних фізичних величин (переміщення, швидкість, тиск, температура, вологість, витрата, рівень фізичної величини тощо). Для таких датчиків входною є вимірювана величина, а вихідним – електричний сигнал. Відповідно до форми сигналів розрізняють аналогові вимірювальні перетворювачі, значення входних і вихідних величин яких можуть безперервно мінятися в усьому діапазоні вимірювання, і перетворювачі з дискретним (цифровим) входом або виходом.

Первинні перетворювачі датчиків неелектричних величин підрозділяють на параметричні (пасивні) і генераторні (активні).

Первинні параметричні перетворювачі (ППП) отримали найбільше поширення в електричних вимірюваннях неелектричних величин. Вони

перетворюють неелектричні вхідні величини в параметри електричних ланцюгів (опір R , індуктивність L , ємність C , взаємну індуктивність M), а потім в напругу (U), струм (I) або частоту (f_i). Відповідно до виду інформаційного параметра розрізняють резистори, індуктивні, ємнісні і трансформаторні ППП. У подібних перетворювачах енергія вихідного сигналу забезпечується тільки за рахунок енергії вхідного сигналу. Наприклад, електричний опір датчика резистора переміщень змінюється тільки при переміщенні його рухливого контакту. Для ППП потрібні електричні джерела живлення. В той же час ППП можуть забезпечити збільшення інтенсивності сигналів, що передаються.

Первинні генераторні перетворювачі (ПГП) перетворюють неелектричні вхідні величини безпосередньо в електричну величину (напругу або струм). Наприклад, термоелектричні вимірювальні перетворювачі (термопари) виробляють вихідний сигнал, функціонально пов'язаний з вимірюваною температурою, при цьому джерело живлення для них не потрібно. В той же час в ПГП може бути присутнім додаткове джерело енергії. Зокрема, перетворювачі, ґрунтовані на використанні ефекту Холу, потребують джерела живлення. Також до ПГП відносяться індукційні і п'єзоелектричні перетворювачі.

Розглянемо приклади датчиків, використовуваних в системах управління з ПЛК. Передусім це датчики кутових і лінійних переміщень, вимикачі і командо-апарати (пристрої з перемикальними електричними контактами).

Простим вимірювальним перетворювачем переміщень є перетворювач потенціометра резистора, принцип дії якого базується на зміні електричного опору провідника зі змінною його довжини. На рис. 3.18 показані пристрій і принципова схема перетворювача потенціометра лінійних переміщень.

У циліндричному корпусі 2 перетворювачі за віссю переміщається шток 1, на якому закріплений струмознімальний вузол, що складається з діелектричної основи 5 і ковзаючого контакту 4. Ковзаючий контакт, у свою чергу, переміщується за закріпленою в корпусі перетворювача пластині з діелектрика 3, на яку нанесені дві доріжки: одна, з резистивного матеріалу,

утворює розподілений резистор, а інша, з матеріалу, що проводить, спільно з ковзаючим контактом утворює відведення від проміжних точок резистора.

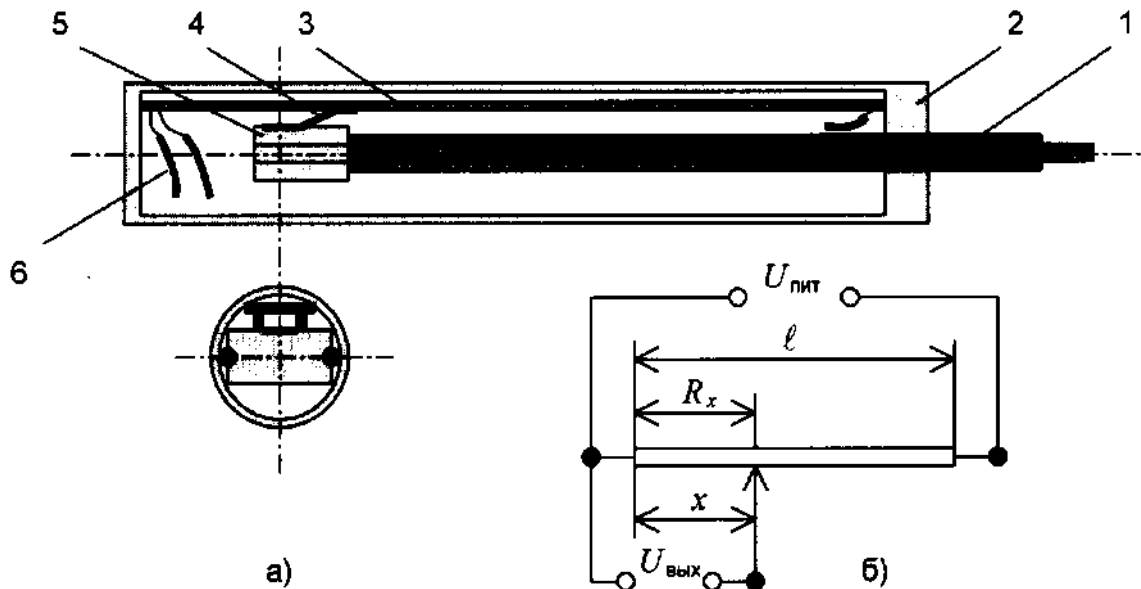


Рисунок 3.18 – Пристрій (а) і схема ввімкнення (б) перетворювача потенціометра лінійних переміщень

Таким чином, перетворювач потенціометра є резистором зі змінним електричним опором R (змінний резистор з рухливим контактом). Величина опору змінного резистора залежить від положення рухливого струмознімального контакту (рис. 3.18, б):

$$R_x = \frac{R \cdot x}{l}. \quad (3.18)$$

Для отримання вихідного електричного сигналу до клем виводів б потенціометра прикладається постійна або змінна напруга $U_{живл}$, а вихідна напруга R_x знімається з частини потенціометра, яка визначається переміщенням x ковзаючого контакту 4. Таким чином, механічне переміщення ковзаючого контакту потенціометра перетвориться в електричну напругу на виході перетворювача:

$$U_{\text{вых}} = \frac{U_{\text{num}} \cdot R_x}{R} = \frac{U_{\text{num}} \cdot x}{l}. \quad (3.19)$$

Зрозуміло, що датчик потенціометра переміщень є аналоговим датчиком, тому для роботи з ним у складі системи управління з ПЛК потрібен спеціальний модуль розширення – аналого-цифровий перетворювач (АЦП).

За допомогою датчиків потенціометрів забезпечується безперервний контроль положення робочих органів механізмів, наприклад пневмо- і гідроциліндрів. В той же час при управлінні технологічним устаткуванням дискретної дії вимагається контролювати тільки кінцеві положення робочих органів. Такий контроль зазвичай здійснюється за допомогою дискретних датчиків переміщень – путніх вимикачів, в яких можуть застосовуватися як аналогові первинні перетворювачі, так і дискретні.

Аналогові первинні перетворювачі переміщень забезпечують безконтактний контроль взаємного положення об'єктів. Вихідний електричний сигнал в дискретних датчиках з аналоговими перетворювачами формується електронною схемою з попаданням контрольованого об'єкту в зону дії датчика. Залежно від принципу дії розрізняють дискретні датчики генераторного і хвилевого типу.

Значень різних типів змінних, при цьому в пам'яті формується певний образ входів/виходів. На рис. 3.19 показаний образ точок входу для моделей процесора CPU 22х.

Слід зазначити, що моделі ЦПП S7-200 CPU 222 і старше зберігають в пам'яті також образ аналогових входів/виходів. Об'єм логічного простору, що надається для аналогових входів/виходів, складає:

- в моделі CPU 222 16 входів і 16 виходів;
- в моделях CPU 224 і CPU 226 32 входи і 32 виходи.

Адресація аналогових входів/виходів відрізняється від адресації дискретних входів/виходів. Входи/виходи додають шляхом приєднання модулів розширення, з підключенням до ЦПП, що утворюють ланцюг.

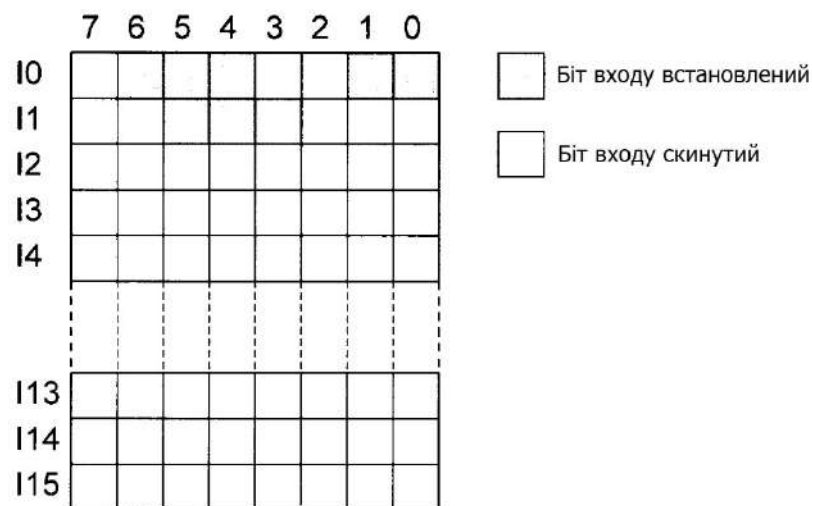


Рисунок 3.19 – Образ входів CPU 22x

Адреси точок модуля визначаються його типом і положенням в ланцюгу по відношенню до попереднього модуля того ж типу. Тому можливі адреси входів/виходів різних модулів не перекриваються в пам'яті. У дискретних модулях розширення регістровий простір образів завжди формується побайтно. Отже, якщо для окремих бітів кожного зарезервованого байту модулем не забезпечується фізична наявність входу або виходу, ці невикористані біти втрачаються і не можуть бути назначені для подальших модулів входу/виходу в ланцюгу. В той же час для вихідних модулів невикористані біти в зарезервованих байтах можуть задіюватися як внутрішні біти пам'яті (М-біти). Для вхідних модулів ці невикористані біти обнуляються в кожному циклі сканування і не можуть задіюватися як М-біти.

Наприклад, при експлуатації базового блоку CPU 224 з модулем розширення 41n/4Qut під область пам'яті образів резервується по три байти для входів (I0 - I2) і виходів (Q1 - Q2). У цих байтах для ідентифікації входів і виходів використовуються наступні біти:

- I0.0 - I0.7 і I1.0 - I1.5 – входи CPU 224;
- I2.0 - I2.3 – входи модуля розширення;
- Q0.0 - Q0.7 і Q1.0, Q1.1 – входи CPU 224;
- Q2.0 - Q2. – входи модуля розширення.

Біти Q1.2 - Q1.7 і Q2.4 - Q2.7 можуть бути використані в якості внутрішніх бітів пам'яті (прапорів). Біти I1.6, I1.7 і I2.4 - I2.7 не можуть бути

використані взагалі. Те ж саме відноситься і до аналогових модулів розширення.

Внутрішня пам'ять бітів (М-пам'ять) CPU 224 містить 256 бітів. Пам'ять спеціальних бітів (SM-пам'ять) містить 1440 бітів. Біти пам'яті ідентифікуються аналогічно бітам входів/виходів, наприклад: M1.5, M17.7, SM43.1 тощо.

Пам'ять змінних (V-пам'ять) розміщується в ОЗУ і припускає як читання, так і запис інформації. V-пам'ять ділиться на два розділи:

- блок даних 1 (DB1), інформація з якого зберігається у внутрішній енергонезалежній пам'яті при кожному завантаженні блоку;
- розділ, аналогічний блоку DB1, інформація з якого не може зберігатися в енергонезалежній пам'яті.

3.4.6 Таймери і лічильники

ПЛК серії S7-200 підтримують функції затримки часу і рахунку, що реалізується таймерами T і лічильниками C , які є програмованими користувачем апаратно-програмними засобами, що використовують базове програмне забезпечення контролера. Таймери і лічильники широко використовуються в програмах управління технічними об'єктами для вироблення сигналів про виконання певних умов в процесі логічних обчислень.

Таймери і лічильники працюють за одним й тим самим принципом: вони запускаються за сигналом дозволу рахунку, а після досягнення заданого користувачем значення рахунку ними встановлюються в одиничний стан так звані біти закінчення рахунку в пам'яті контролера (рис. 3.20).

У початковому стані біт таймера T скинутий (має значення 0). Під час запуску таймера у момент часу t_i починається відлік заданого часового інтервалу Δt_z , після закінчення якого біт T встановлюється в 1. Аналогічно працює і лічильник C , для якого задається чисельне значення кінця рахунку n_c . Скидання бітів T і C здійснюється програмним шляхом за командою скидання.

Максимальна кількість таймерів і лічильників, підтримувана моделями CPU 22х, дорівнює 256 для кожного типу пристроїв.

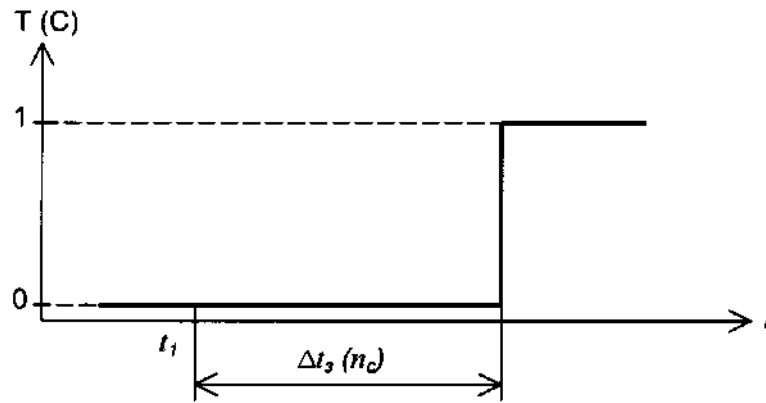


Рисунок 3.20 – Принцип роботи таймерів і лічильників

При використанні таймерів слід мати на увазі, що затримка Δt_3 формується цілим числом фіксованих малих інтервалів часу – дискрет. У процесорах S7-200 використовуються значення дискрет 1 мс, 10 мс і 100 мс.

Необхідне значення затримки забезпечується за допомогою завдання необхідного числа дискрет в програмі. Це число називається уставкою таймера (позначається PT) і визначається співвідношенням:

$$PT = \Delta t_3 / D, \quad (3.20)$$

де D – вибрана дискрета таймера.

Наприклад, для реалізації затримки $\Delta t_3 = 5$ с можна вибрати таймер T_{49} зі значеннями дискрета $D=100$ мс і уставкою $PT = 5/0,1 = 50$.

3.4.7 Програмне забезпечення STEP 7-Micro/WIN 32

У системі автоматизації SIMATIC S7-200 використовується програмне забезпечення (ПО) STEP 7-Micro/WIN 32, спеціально розроблене для максимального полегшення переходу до технології ПЛК.

STEP 7-Micro/WIN 32 підтримує досить потужний набір команд, що дозволяють виконувати наступні операції:

- двійкові операції (базова логіка релейно-контактних схем);
- логічні операції над словами;
- зсув і зміщення слів і бітів;

- переміщення, запис і пошук даних;
- математичні операції складання, віднімання, множення і ділення цілих чисел, операції над числами з плаваючою точкою (включаючи розрахунок квадратного кореня), а також операції порівняння;
- перетворення форматів представлення чисел (наприклад, двійкового в двійково-десятковий);
- реалізація програмних циклів;
- виклик підпрограм;
- реалізація високошвидкісних лічильників, робота з імпульсними виходами, обробка переривань;
- обробка таблиць;
- реалізація таймерів і лічильників;
- визначення фронтів імпульсів вхідних сигналів;
- вбудовування протоколу передачі даних, який визначається користувачем (режим вільного порту);
- виконання мережевих інструкцій читання/запису даних на віддалені станції;
- реалізація функцій відладки програм, серед яких:
 - а) одноразове або безперервне оновлення значень змінних;
 - б) одноразовий або безперервний контроль значень змінних;
 - в) примусове завдання значень змінних, які зберігаються в пам'яті навіть з обривом з'єднання (на відміну від змінних, які визначаються користувачем і ЦПУ, що призначаються);
 - г) зупинка ЦПП після виконання заданого числа програмних циклів (при цьому значення найважливіших змінних можуть бути виведені на дисплей).

Моделі ЦПП S7-200 надають користувачеві велике число інструкцій, що дозволяють вирішувати різноманітні завдання автоматизації. Ці інструкції входять до складу двох базових наборів команд: SIMATIC і IEC 1131-3. Пакет програмування STEP 7-Micro/WIN 32 надає користувачеві можливість використати будь-який з вказаних наборів. Проте слід зазначити, що набір команд SIMATIC розроблений спеціально для ПЛК фірми «Siemens» і, отже, забезпечує велику спадкоємність при переході від одного ряду контролерів до


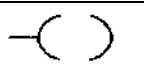
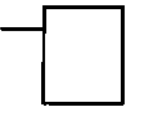
іншого в порівнянні з набором команд IEC 1131-3, можливість застосування якого залежить від виробника ПЛК. Крім того, команди з набору SIMATIC виконуються процесорами ПЛК серії S7 з більш високою швидкістю. Тому далі розглядатиметься саме набір команд SIMATIC.

Пакет STEP 7-Micro/WIN 32 дає можливість використати три мови програмування при складанні програм на базі набору команд SIMATIC: мова релейно-контактних схем (LAD).

3.4.8 Мова релейно-контактних схем (LAD)

Мова релейно-контактних схем (LAD) відноситься до графічних мов програмування ПЛК. У мові LAD функції, що управляють, формуються за допомогою обчислювальних ланцюжків. Кожен обчислювальний ланцюжок (Network) є сукупністю елементів, сполучених між собою відповідно до логіки управління. Основні елементи мови LAD представлені в таблиці 3.2.

Таблиця 3.2 – Основні елементи мови релейно-контактних схем (LAD)

Елемент	Опис
	Контакт. Є перемикачем, через який може проходити струм. Через замикаючий (нормально відкритий) контакт струм проходить тільки тоді, коли контакт замкнений (логічне значення дорівнює 1). Через розмикаючий (нормально замкнутий) контакт струм проходить тільки тоді, коли контакт розімкнений (логічне значення дорівнює 0)
	Обмотка. Є виходом релейного типу, що активізується при проходженні струму
	Бокс. Є функцією (наприклад, функцію таймера, лічильника або математичної операції), що виконується з проходженням струму через бокс

Обчислювальний ланцюжок складається з елементів, перерахованих в таблиці 3.2, і є закінченим електричним ланцюгом, струм якій йде від шини живлення (у програмі на мові LAD зображується вертикальною лінією, розташованою зліва від логічних ланцюжків, див. рис. 3.21) через замкнуті контакти, активізуючи відповідні обмотки і бокси.

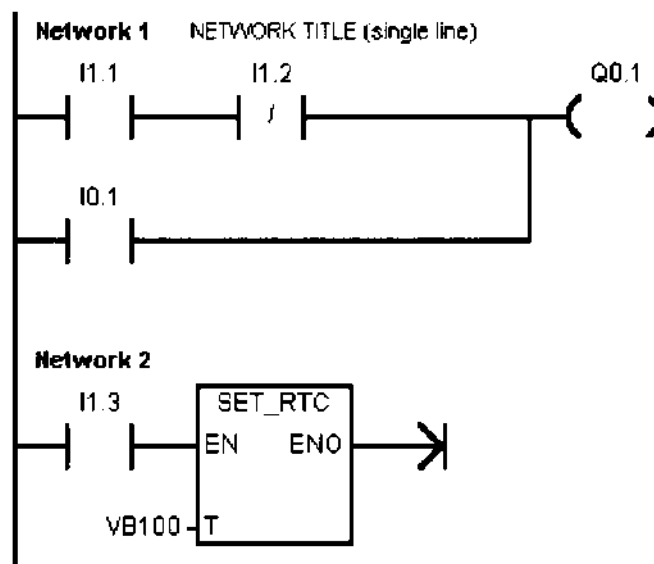


Рисунок 3.21 – Фрагмент програми на мові контактно-релейних схем (LAD)

Кожному контакту обчислювального ланцюжка ставиться у відповідність деяка логічна змінна, одиничне значення якої ототожнюється із замкнутим станом контакту, а нульове – з розімкненим. Логічні змінні контактів можуть бути вхідними (наприклад, сигнал від дискретного датчика), вихідними (наприклад, біт пам'яті – прапор) і проміжними змінними. Обмотці реле також відповідає певна логічна змінна, що набуває значення 1, якщо обмоткою «проходить струм», або 0, якщо «струм не проходить».

Логічна змінна обмотки реле може бути вихідною змінною (наприклад, бітом пам'яті), за допомогою яким контролер управляє зовнішнім пристроєм.

Кожен елемент обчислювального ланцюжка позначається ідентифікатором з метою забезпечити можливість управління їм за допомогою базового програмного забезпечення. Ідентифікатор повинен відповідати типу елемента. Зокрема, дискретна вхідна змінна позначається як IXX.X, а дискретна вихідна – як QXX.X, де XX.X – адреса змінної.

Елементи обчислювальних ланцюжків з'єднуються, як і в релейно-контактних схемах, послідовно (з утворенням гілки, або ряду, Row), і паралельно (при цьому формуються паралельні гілки, або галуження). З точки зору алгебри логіки (булевої алгебри) послідовне включення контактів реалізує логічну функцію І, паралельне – логічну функцію АБО. В цілому

обчислювальний ланцюжок можна розглядати як задану логічну функцію послідовності команд, що виконуються процесором ПЛК.

На рис. 3.21 як приклад наводиться фрагмент програми на мові LAD.

У цьому прикладі обчислювальний ланцюжок Network 1 включає дві паралельні гілки: верхню з послідовно сполученими контактами I1.1 і I1.2 і нижню, включаючи один контакт I0.1. Ланцюжок має один дискретний вихід Q0.1.

Обчислювальний ланцюжок Network 2 також має один вихід, представлений боксом установки таймера реального часу. Логіка управління таймером включає один контакт I1.3. Гілка Network 2 закінчується індикатором струму у вигляді стрілки. Стрілка індикатора струму, що виходить з боксу, показує, що після елемента боксу в ланцюжок можуть бути додані і інші логічні елементи. (Якщо ж індикатор струму розташований відразу після контакту, необхідно повністю сформувати обчислювальний ланцюжок, інакше програма не буде скомпільована.)

У LAD-програмі усі логічні ланцюжки розташовуються відповідно до нумерації, при цьому формується чітка послідовність обчислювальних дій. При цьому кожен ланцюжок обробляється зліва направо. Усі обчислювальні ланцюжки об'єднані з лівого боку загальною вертикальною лінією, відповідній потенційній шині живлення релейно-контактної схеми (нульова шина в ПЗ STEP 7-Micro не відображається).

При написанні LAD-програми тип контакту, що поміщається в ланцюжок, обирається виходячи з наступного простого правила: якщо при одиничному значенні певної змінної для виконання необхідної дії (включення приводу, подання сигналу, обчислення значення функції та ін.) вихід обчислювального ланцюжка повинен встановлюватися в 1 (збуджуватися), то ця змінна має бути представлена в ланцюжку замикаючим контактом; якщо вихід ланцюжка повинен збуджуватися за нульовим значенням цієї змінної, то вона має бути представлена розмикаючим контактом.

Це правило виключає чисто формальний підхід до побудови обчислювальних ланцюжків при переході від релейно-контактної реалізації системи управління до програмної, із застосуванням контролера. Неприпустимо ототожнювати контакти релейно-контактної схеми і контакти

LAD-програми. Розглянемо, наприклад, добре відому схему ввімкнення електромагнітного реле або контактора з самозатримкою (рис. 3.22).

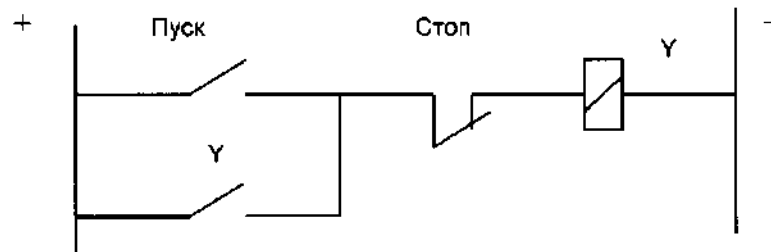


Рисунок 3.22 – Електрична схема включення електромагнітного реле з самозатримкою

Працює схема таким чином. З натисканням на кнопку «Пуск», що містить замикаючий контакт, струм проходить через розмикаючий контакт кнопки «Стоп» на обмотку реле Y , що призводить до спрацьовування реле з виконанням необхідних перемикачів. Замикаючий контакт реле Y , включений паралельно контакту кнопки «Пуск», забезпечує живлення обмотки реле і у тому випадку, коли кнопка «Пуск» не натиснута. Реле відключається натисненням кнопки «Стоп», з спрацьовуванням розмикаючого контакту, яким ланцюг живлення обмотки реле розривається.

На рис. 3.23 наведено два обчислювальні ланцюжки програмної реалізації розглянутої релейно-контактної схеми.

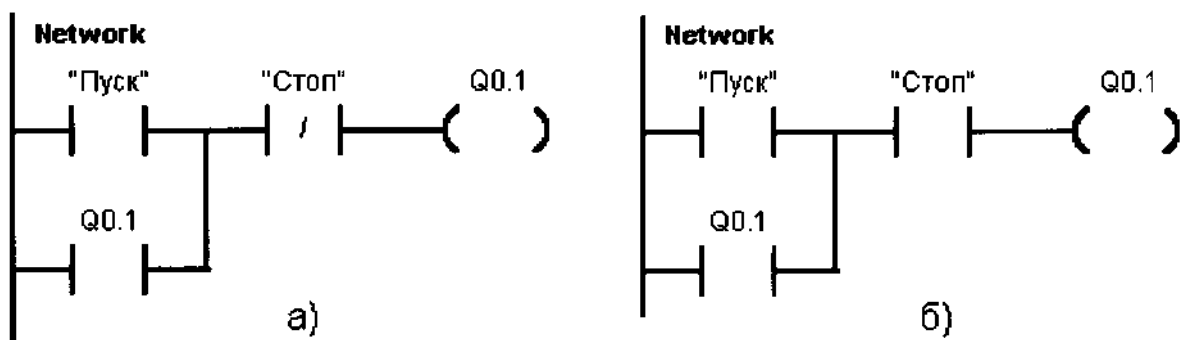


Рисунок 3.23 – Програмна реалізація кнопкової станції:
а) неправильна; б) правильна

У обчислювальний ланцюжок на рис. 3.23, а включений розмикаючий контакт, що відповідає кнопці «Стоп», тобто аналогічний розмикаючому контакту релейно-контактної схеми. З виконанням цієї програми вихід цього ланцюжка встановлюватиметься в 1 з натисненням кнопки «Пуск» і розмиканні контакту кнопки «Стоп» (тобто з її натисненням); реле спрацюватиме за наявності умов, відмінних від початкових.

Приклад ланцюжка на рис. 3.23, б показує правильне включення в програму кнопки «Стоп». Тут вихід ланцюжка збуджується при замкнутому стані контактів кнопки «Стоп» і реле. При натисненні кнопки «Стоп» відбувається розмикання її контакту і реле відключається, що і повинно відбуватися у процесі правильної роботи початкової релейно-контактної схеми.

3.5 Контрольні запитання та завдання

1. Що можна віднести до основних завдань механіки маніпуляторів?
2. Наведіть структурну схему маніпулятора.
3. Як виконується силовий розрахунок маніпулятора?
4. Як виконується розрахунок швидкодії промислового робота?
5. Наведіть структурну схему автоматизованої системи управління автономним об'єктом.
6. Наведіть типові функції управління технологічним устаткуванням.
7. Поясніть пристрій і роботу програмованих логічних контролерів
8. Які ви знаєте основні мови програмування ПЛК?
9. Наведіть приклади технічних засобів систем управління з ПЛК.
10. Поясніть принцип роботи таймерів та лічильників.
11. Наведіть основні елементи мови релейно-контактних схем.

4 ТЕХНОЛОГІЯ ПРОГРАМУВАННЯ ПЛК

4.1 Загальні елементи програмування мовами стандарту МЕК

Перш ніж переходити до вивчення програмування мовами стандарту МЕК, необхідно ознайомитися із загальними елементами цих мов. Загальні елементи є єдиним фундаментом, що дозволяє об'єднати багатомовні компоненти в одному проєкті.

4.1.1 Типи даних

Тип даних змінної визначає рід інформації, діапазон представлення і безліч допустимих операцій. Мови МЕК використовують ідеологію строгої перевірки типів даних. Це означає, що будь-яку змінну можна використовувати тільки після її оголошення. Привласнювати значення однієї змінної іншій можна, тільки якщо вони обидві одного типу. Допускається також присвоювання значення змінної сумісного типу, що має більш широку множину допустимих значень. В цьому випадку відбувається неявне перетворення типу без втрат. Неявні перетворення типів даних з втратами заборонені. Так, наприклад, логічну змінну, здатну приймати тільки два значення (логічні 0 і 1), можна привласнити змінній типу SINT (-128...+127), але не навпаки.

Під час трансляції програми всі подібні спроби відслідковуються і вважаються грубими помилками. Якщо ж це дійсно необхідно, то виконати присвоювання з втратами можливо, але тільки за допомогою спеціальних операторів. Оператори перетворення в МЕК виконують також і більш складні операції, наприклад перетворення числа або календарної дати в текстовий рядок, і навпаки.

Найбільша різноманітність типів даних у стандарті передбачена для представлення цілих чисел. Сенс застосування широкого спектра цілочисельних змінних полягає насамперед в оптимізації коду програми. Швидкість обчислень залежить від того, як мікропроцесор оперує зі змінними даного типу. Так, цілком очевидно, що 16-розрядний процесор виконує

складання двох 16-розрядних значень однією командою. Додавання ж двох значень 32-розрядних змінних – це підпрограма з декількох команд.

Додаткові затримки можуть утворюватися за рахунок мультиплексування шини даних, що зв'язують процесор і пам'ять, особливостей мікросхем пам'яті тощо. У загальному випадку менші за діапазоном представлених значень типи змінних вимагають менше пам'яті, менше коду, і обчислення з їх участю виконуються значно швидше.

Типи даних МЕК поділяються на дві категорії – елементарні і складені. Елементарні або базові типи є основою для побудови складених типів. До складених типів належать перерахування, масиви, структури, масиви структур тощо.

4.1.2 Елементарні типи даних

Цілочисельні типи

Цілочисельні змінні відрізняються різним діапазоном даних, що зберігається і, відповідно, різними вимогами до пам'яті. Детально дані характеристики представлені в таблиці 4.1.

Таблиця 4.1 – Основні типи даних та їх характеристики

Тип	Нижня границя	Верхня границя	Розмір, у байтах
BYTE	8 біт		1
WORD	16 біт		2
DWORD	32 біта		4
LWORD	64 біта		8
SINT	-128	127	1
INT	-32768	32767	2
DINT	-2^{31}	$2^{31}-1$	4
LINT	-2^{63}	$2^{63}-1$	8
USINT	0	255	1
UINT	0	65535	2
UDINT	0	232-1	4
ULINT	0	264-1	8

Нижня границя діапазону цілих без знаку 0, верхня границя визначається як $(2^n) - 1$, де n – число розрядів числа. Для чисел зі знаком нижня границя $-(2^{n-1})$, верхня границя $(2^{n-1}) - 1$.

Найменування цілих типів даних утворюються із застосуванням префіксів, що виражають відношення розміру до 16-розрядних слів: S (short *1/2) коротке, D (double *2) подвійне, L (long *4) довге. Префікс U (unsigned) вказує на представлення цілих без знаку.

Змінні типів BYTE, WORD, DWORD і LWORD визначаються стандартом як бітові рядки ANY_BIT. Говорити про діапазон значень чисел для цих змінних взагалі некоректно. Вони представляють рядки з 8, 16 і 32 біт, відповідно. Крім звернення до таких змінних як до єдиних цілих, їх можна використовувати побітно.

Цілі числа можуть бути представлені в двійковій, вісімковій, десятковій або шістнадцятковій системі числення. Числові константи, відмінні від десяткових, вимагають зазначення основи системи числення перед знаком «#». У таблиці 4.2 наведено приклад запису чисел в різних системах числення.

Таблиця 4.2 – Приклад запису чисел в різних системах числення

Система числення	Приклад запису чисел
двійкова	2#0100_1110
вісімкова	8#116
шістнадцяткова	16#4E
десятькова	78

Для позначення шістнадцятирічних цифр від 10 до 15 використовуються латинські літери від А до F.

Символ підкреслення «_» не впливає на значення і використовується виключно для поліпшення зорового сприйняття числа. Наприклад: 10_000, 16#01_88. Підкреслення можна застосовувати тільки між цифрами або в кінці числа. Два або більш підкреслення поспіль застосовувати не можна.

За початкової ініціалізації цілочисельні змінні отримують нульові значення. Якщо необхідно задати інші початкові значення, це можна зробити безпосередньо з оголошенням змінної.

Приклади:

VAR

wVar0, wVar1: WORD; (*2 змінних типу WORD*)

byVar3 : BYTE ; (*тип BYTE початкове значення 0*)

byVar2: BYTE := 16#55; (*тип BYTE початкове
значення 55h*)

END_VAR

byVar2 := 2#1_0_0_0_1_0_0_0; (*рівнозначно 2#1000_1000*)

byVar3 := 2#1_0_0_0_1_0_0_0; (*помилка*)

Логічний тип

Логічні змінні оголошуються ключовим словом BOOL. Це означає їх приналежність до алгебри Буля. Вони можуть приймати тільки значення логічного нуля FALSE (ХИБНІСТЬ) або логічної одиниці TRUE (ІСТИНА). При початковій ініціалізації логічне значення за замовчуванням – ХИБНІСТЬ.

Приклад:

VAR

bVar1: BOOL := TRUE;

wVar2: WORD;

END VAR

З перетворенням значення логічної змінної в цілу FALSE повертає 0, а TRUE 1:

wVar2 := BOOL_TO_WORD(bVar1); (*результат 1 *)

Зі зворотним перетворенням будь-якого цілого в логічну змінну істину утворює будь-яке ненульове значення:

wVar2 := 0;

bVar1 := WORD_TO_BOOL(wVar2); (*результат FALSE*)

Результати операцій, що дають логічне значення, можна присвоювати змінним типу BOOL:

```
bVar1 := wVar2 > 5000;
```

За визначенням BOOL – це рядок з одного біта, але з міркувань ефективності коду під час автоматичного розподілу пам'яті транслятором під бітову змінну виділяється, як правило, 1 байт пам'яті повністю. Змінні типу BOOL, пов'язані з дискретними входами-виходами або визначені з прямою бітовою адресою, дійсно фізично представлені одним бітом.

Дійсні типи

Змінні дійсного типу REAL є дійсні числа в діапазоні $\pm 10^{\pm 38}$. З 32 біт, що займається числом, мантиса займає 23 біта. В результаті точність подання приблизно становить 6-7 десяткових цифр. Довгий дійсний формат LREAL займає 64 біти. Число містить 52-бітову мантису. Точність подання приблизно становить 15-16 десяткових цифр. Діапазон чисел довгого дійсного $\pm 10^{\pm 307}$.

Числа з плаваючою комою записуються в форматі з точкою: 14.0, 0.33_, -120.2, або в експоненційній формі: $-1.2E10$, $3.1e7$.

Інтервал часу

Змінні типу TIME використовуються для вираження інтервалів часу. На відміну від часу доби (TIME OF DAY) часовий інтервал не обмежений максимальним значенням в 24 години.

Числа, які виражають часовий інтервал, мають починатися з ключового слова TIME# або в скороченій формі T#. У загальному випадку подання часу складається з полів днів (d), годин (h), хвилин (t), секунд (s) і мілісекунд (ms). Порядок подання має бути саме такий, хоча непотрібні елементи можна опускати. Для кращого зорового сприйняття поля допускається розділяти символом підкреслення.

Наприклад:

VAR

TIME1: TIME := t#10h_14m_5s;

END_VAR

Старший елемент може перевищувати верхню межу діапазону подання. Отже, якщо у поданні присутні дні або години, то секунди не можуть перевищувати значення 59. Якщо секунди стоять першими, то їх значення може бути й більшим.

TIME1 := t#1m65s; (*помилка*)

TIME1 := T#125s; (*правильно*)

Молодший елемент можна подати у вигляді десяткового дробу:

TIME1 := T#1.2S; (*рівносильне T#1s200ms*)

Час доби і дата

Типи змінних, що виражають час дня чи дату, подаються згідно з ISO 8601.

У таблиці 4.3 подані типи змінних, що виражають час дня чи дату.

Таблиця 4.3 – Типи змінних, що виражають час дня чи дату

Тип	Коротке позначення	Початкове значення
DATE	D	1 січня 1970 р.
TIME_OF_DAY	TOD	00:00
DATE_AND_TIME	DT	00:00 1 січня 1970 р.

Дата записується у форматі «рік»-«місяць»-«число». Час записується у форматі «години»:«хвилини»:«секунди».«соті». Дата визначається ключовим словом DATE# (скорочено D#), час дня TIME_OF_DAY# (скорочено TOD#), дата і час DATE_AND_TIME# (скорочено DT#).

Приклад запису дати і часу:

DATE#2002-01-31 або D#2002-01-31

TIME_OF_DAY# 16:03:15.47 або TOD#16:03:15.47

DATE_AND_TIME#2002-01-31-16:03:15.47 або

DT#2002-01-31-16:03:15.47

Всі три типи даних фізично займають 4 байти (DWORD). Тип TOD містить час доби в мілісекундах, починаючи з 0 годин. Типи DATE і DT містять час у секундах, починаючи з 0 годин 1 січня 1970 року.

Рядки

Тип рядкових змінних STRING визначає змінні, що містять текстову інформацію. Розмір рядка задається при оголошенні. Наприклад, оголошення рядка str1, що вміщає до 20 символів, і str2 – до 60 символів:

```
VAR
```

```
    str1: STRING(20);
```

```
    str2: STRING(60) := 'Протягування';
```

```
END_VAR
```

Якщо початкове значення не задано, то в ході ініціалізації буде створений порожній рядок.

Кількість необхідної пам'яті визначається заданим з оголошенням розміром рядка. Для типу STRING кожен символ займає 1 байт (WSTRING 2 байти). Рядкові константи задаються між одинарних лапок:

```
    str1 := 'Включення насосу';
```

За необхідності розміщення в рядок коду, що не має друкованого відображення, використовується знак (\$) і наступний за ним код з двох цифр у шістнадцятковій системі числення. У таблиці 4.4 наведені позначення для поширених керуючих термінальних кодів.

Таблиця 4.4 – Позначення для термінальних кодів

Позначення	Код
\$\$	Знак долара
\$'	Одиночна лапка
\$L або \$l	Перевід рядка
\$N або \$n	Новий рядок
\$P або \$p	Перевід сторінки
\$R або \$r	Розрив рядка
\$T або \$t	Табуляція

Ієрархія елементарних типів

Наведена в таблиці 4.5 ієрархія елементарних типів застосовується виключно для зручності опису програм. Кожне найменування ANY_... об'єднує деяку множину типів. Так, у ході опису будь-якої бітової операції зручніше вказати, що вона може бути застосовна для ANY_BIT, ніж перераховувати щоразу допустимі елементарні типи. Застосовувати ANY_ при оголошенні змінних не можна.

Таблиця 4.5 – Ієрархія елементарних типів

ANY	ANY_NUM	ANY_INT	SINT, INT, DINT, LINT, USINT, UINT, UDINT, ULINT
		ANY_REAL	REAL, LREAL
	ANY_BIT		BOOL, BYTE, WORD DWORD, LWORD
	STRING		
	TIME		
	ANY_DATE		DATE, TIME_OF_DAY, DATE_AND_TIME

4.1.3 Призначені для користувача типи даних

Опис призначених для користувача типів даних (крім масивів) має виконуватися на рівні проекту (в CODESYS на вкладці «Типи даних» => «Організатор об'єктів»). Оголошення типу завжди починається з ключового слова TYPE і закінчується рядком END_TYPE.

Масиви

Масиви – це множина однотипних елементів з довільним доступом. Масиви можуть бути багатовимірними. Розмірність масиву і діапазони індексів задаються при оголошенні. Приклад оголошення тривимірного масиву:

```
<ім'я масива>:ARRAY  
[<li1>...<hi1>,<li2>...<hi2>,<li3>...<hi3>] OF <тип елемента>;
```

де li1, li2, li3 вказують нижні границі індексів; hi1, hi2 і hi3 – верхні границі. Індеси мають бути цілого типу і тільки позитивні. Негативні індекси використовувати не можна.

Приклади оголошення масивів:

```
XYbass:   ARRAY [1..10,1..20] OF INT;  
TxtMsg:   ARRAY [0..10] OF STRING(32);  
Mass1:    ARRAY [1..6] OF SINT := 1,1,2,2,2,2;  
Mass2:    ARRAY [1..6] OF SINT := 1,1,4(2);
```

Два нижніх приклади показують, як можна виконати ініціалізацію елементів масиву при оголошенні. Обидва приклади створюють однакові масиви. У першому прикладі всі початкові значення наведені через кому. У другому прикладі присутнє скорочення N(a,b,c..) яке означає – повторити послідовність a, b, c.. N раз. Багатовимірні масиви ініціалізуються порядково:

```
Mass2d:   ARRAY [1..2,1..4] OF SINT := 1,2,3,4,5,6,7,8;
```

Результат ініціалізації Mass2d показаний на рис. 4.1.

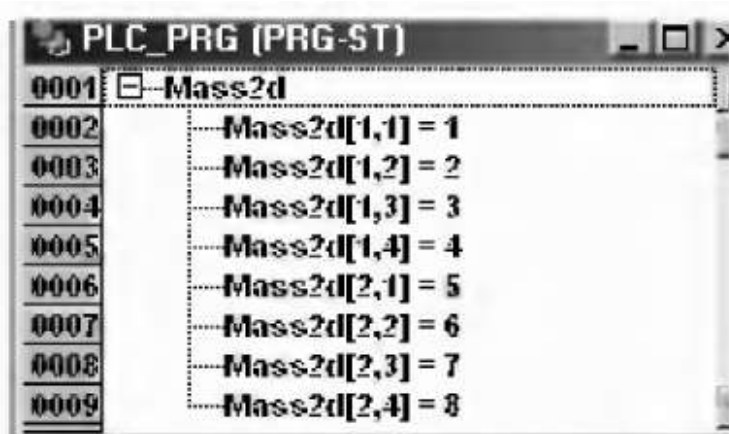


Рисунок 4.1 – Результат ініціалізації Mass2d

Для доступу до елементів масиву застосовується такий синтаксис:

<Ім'я_масива>[Індекс1, Індекс2, Індекс3]

Для двовимірного масиву використовуються два індекси. Для одновимірного, очевидно, достатньо одного. Наприклад:

```
XUbase[2,12] := 1;
i := STR_TO_INT(TxtMsg[4]);
```

Рекомендується використовувати в масивах нумерацію з нуля. У цьому випадку обчислення фізичної адреси елемента при виконанні простіше. Внаслідок цього код буде коротшим.

Структури

Структури призначені для створення нових типів даних на основі елементів різних базових типів. Зі змінною типу структура можна поводитися як з єдиним елементом, передавати як параметр, створювати покажчики, копіювати тощо.

На відміну від масивів структура дійсно вводить новий тип даних. Це означає, що до застосування конкретної змінної потрібно виконати як мінімум два оголошення. Спочатку потрібно описати структуру.

Опис структури відбувається глобально, на рівні проекту. Описана структура отримує ідентифікатор (ім'я структури). Але це ще не змінна, це новий тип даних. Тепер, використовуючи новий ідентифікатор, потрібно оголосити одну або скільки завгодно змінних, так само, як і для базових типів. Тільки тепер змінна нового типу отримує «тілесну оболонку» або, іншими словами, конкретне місце в пам'яті даних.

Оголошення структури має починатися з ключового слова STRUCT і закінчуватися END_STRUCT. Синтаксис оголошення виглядає так:

```
TYPE <Ім'я _структури>:  
    STRUCT  
        < Оголошення змінної 1 >  
        < Оголошення змінної n >  
    END_STRUCT  
END_TYPE
```

Приклад оголошення структури на ім'я Trolley:

```
TYPE Trolley:  
    STRUCT  
        Start:    TIME;  
        Distance: INT;  
        Load, On: BOOL$  
        Articl:   STRING(16);  
    END_STRUCT  
END_TYPE
```

Оголошення в програмі змінної Telegal типу Trolley і початкова ініціалізація структури виглядає так:

```
Telegal: Trolley := (Articl:='Порожній');
```

Стан елементів після початкової ініціалізації Telegal показано на рис. 4.2.

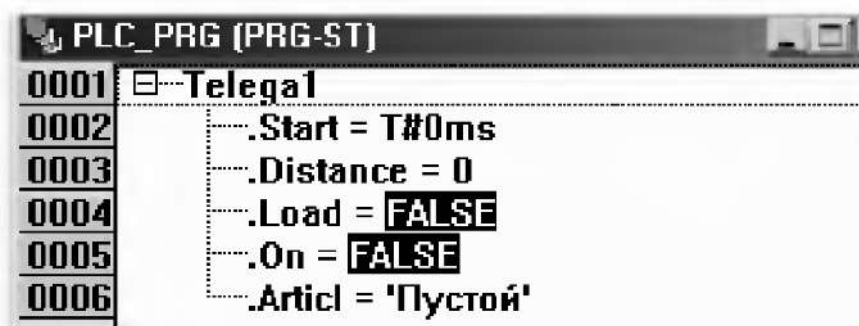


Рисунок 4.2 – Стан елементів після початкової ініціалізації Telegal

За початкової ініціалізації не обов'язково ставити значення для всіх елементів. Елементи, які не мають чітко зазначених початкових значень, за замовчуванням отримують нульові значення.

Для доступу до елементів структури використовується такий синтаксис:

<Ім'я_змінної>.<Ім'я_елемента>

Наприклад:

Telegal.On := True;

Структури можуть включати інші структури, масиви й самі утворювати масиви. Приклад оголошення та ініціалізації масиву структур:

```
TrolleySet: ARRAY[0..2] OF Trolley := (Articl := 'T 1'),  
(Articl := 'T 2'), (Articl := 'T 3');  
TrolleySet[i].On := TRUE;
```

Якщо структура містить вкладену структуру, то доступ до елементів вкладеної структури здійснюється із застосуванням складного імені, що містить дві точки:

```
train.wagon[5].weight; (*wagon[] вкладений масив структур*)
```

Оскільки фізичний розмір елементів структури відомий транслятору заздалегідь, звернення до елемента структури не дає жодних накладних витрат порівняно з простою змінною. Транслятор має можливість розрахувати абсолютні адреси елементів у ході компіляції. Це не стосується масивів структур. Щоб не мати проблем у використанні декількох різних змінних однієї структури, застосовувати прямі адреси в структурі не можна.

Перерахування

Перерахування дозволяє визначити кілька послідовних значень змінної і присвоїти їм найменування. Перерахування – це зручний інструмент, що дозволяє обмежити множину значень змінної і посилити контроль під час трансляції. Як і структура, перерахування створює новий тип даних, визначення якого виконується на рівні проекту:

```
TYPE <Ім'я перерахувань>:  
    (<Елемент 0>, <Елемент 1>, ... <Елемент n>);  
ENDTYPE
```

Оголошена пізніше змінна типу <Ім'я перерахування> може приймати тільки перераховані значення. В ході ініціалізації змінна отримує перше зі списку значення. Якщо числові значення елементів перерахування не вказані явно, їм присвоюються послідовно зростаючі числа, починаючи з 0. Фактично елемент перерахування – це число типу INT з обмеженим набором значень. Якщо необхідно, значення елементам можна присвоїти явно під час оголошення типу перерахування.

Наприклад:

```
TYPE TEMPO: (Adagio := 1, Andante := 2, Allegro := 4);  
END_TYPE
```

Ідентифікатори елементів перерахування використовуються в програмі як значення змінної:

```
VAR
    LiftTemp : TEMPO := Allegro;
END VAR
```

Якщо в різні перерахування включені елементи з однаковими іменами, виникає неоднозначність. Для вирішення цієї проблеми застосовується префікс, що містить перерахування: TEMPO#Adagio. У CODESYS всі найменування елементів перерахування мають бути унікальними.

Обмеження діапазону

Тип змінних з обмеженим діапазоном значень дозволяє визначити допустиму множину значень змінної. Оголошення типу змінної з обмеженим діапазоном має відбуватися безпосередньо між ключовими словами TYPE і END_TYPE:

```
TYPE <Ім'я> :
    <Цілий тип> (<від>..<до>)
END_TYPE
```

Наприклад:

```
TYPE DAC10:
    INT (0..16#3FF);
ENDTYPE
```

Застосування змінної з обмеженням діапазону покажемо на прикладі:

```
VAR
    dac: DAC10;
END_VAR
dac := 2000;
```

Зі спробою трансляції даного прикладу виникає помилка:
Error: Type mismatch: Cannot convert '2000' to 'INT(0..1023)'.

Псевдоніми типів

Проблема вибору відповідного типу даних не завжди вирішується легко. Припустимо, ви працюєте з температурою, заміряною 16-розрядним АЦП. Чи може бути температура тільки вище нуля або коли-небудь буде потрібно працювати в негативній області, ще не зовсім очевидно. В одному випадку потрібно використовувати тип змінних `UINT`, а в іншому – `INT`. Тут зручно визначити новий тип даних:

```
TYPE TEMPERATURA : UINT;  
END TYPE
```

Далі скрізь у програмі використовується тип `TEMPERATURA` під час оголошення змінних. Якщо раптом знадобиться змінити тип температури на `INT`, то це легко і швидко можна буде зробити в одному місці.

Аналогічні псевдоніми типів зручно створювати для будь-яких часто використовуваних в програмі типів. Наприклад, для масивів або інших типів, що мають довге і невиразне визначення.

Специфіка реалізації типів даних CODESYS

Стандарт тільки визначає сумісні типи даних, але не вимагає обов'язкової підтримки усіх типів для всіх реалізацій систем МЕК-програмування. `CODESYS` має найбільш повну підтримку стандартних типів. Але навіть він у версії 2.x не підтримував 64-розрядні цілі і текстові рядки `Unicode`.

Крім того, обмеження підтримуваних типів даних можливо навіть у рамках одного комплексу програмування для різних контролерів. Так, восьмирядний генератор коду `CODESYS` не підтримує дійсні змінні, перерахування та змінні, які виражають час доби і календарну дату. Обмеження підтримки типів обґрунтовується досягненням мінімальної вартості за максимальної ефективності ПЛК різних категорій. Так, повна реалізація ядра системи виконання `CODESYS` (включаючи налагодження функції і трасування значень змінних) для `Intel 8051` сумісного мікроконтролера вимагає загалом 6 Кб пам'яті коду. Тому і код прикладної

програми має бути максимально компактним, для чого доводиться йти на певні компроміси. Але за необхідності будь-які спеціалізовані типи даних можна визначити на базі елементарних типів і підтримати за допомогою бібліотек.

У CODESYS немає обмеження на спосіб застосування бітових рядків. Внаслідок цього типи BYTE, WORD і DWORD можна застосовувати в операціях, що вимагають цілих без знака (USINT, UINT і UDINT), але не навпаки.

Внутрішній формат змінних типу TIME стандартом не обмежений. У CODESYS інтервали часу зберігаються у змінній типу DWORD в мілісекундах, що забезпечує надання інтервалів майже 50 діб. Якщо довжина рядка STRING з оголошенням не зазначена, то приймається значення за замовчуванням – 80 символів. У CODESYS використовуються нультерміновані (як у C-компіляторах) рядки. Тобто під рядок завжди заздалегідь виділяється область пам'яті заданого максимального розміру.

Будь-який рядок закінчується нульовим байтом, який не входить до складу рядка, а слугує виключно для визначення кінця рядка функціями, які оперують із рядками. Порожній рядок складається з єдиного нульового байта. З оголошенням рядка необхідно ставити розмір на одиницю більше необхідного для символу «кінець рядка». Така форма подання найбільш компактна (лише 1 допоміжний байт), але, очевидно, не оптимальна в плані швидкодії. Якщо, наприклад, потрібно об'єднати два рядки, то функція конкатенації рядків зобов'язана спочатку знайти, де закінчується перший рядок. В інших системах програмування можна зустріти реалізацію рядків у вигляді структури, що містить максимальний розмір, поточну довжину рядка і сам рядок (масив байт). Взагалі ж робота із рядками в ПЛК потребується не часто.

Для підтримки перевірки значень змінних з обмеженим діапазоном під час роботи система виконання повинна надавати засоби контролю. У CODESYS це завдання вирішується так – дії, які виконуються, зі спробою виходу за діапазон визначаються програмістом. Для цього призначені спеціальні функції (CheckRangeSigned, CheckRangeUnsigned), які необхідно включити до проекту. На вході функції отримують три параметра: дві границі

діапазону і значення. Будь-яка необхідна реакція на порушення границь (обмеження змінної, індикація помилки тощо) описується в тілі функцій контролю.

4.1.4 Змінні

Кожна змінна обов'язково має найменування й тип. Сутність змінної може бути різною. Змінна може представляти вхід або вихід ПЛК, дані в оперативній або незалежній пам'яті. Далі ми розглянемо правила оголошення та деякі практичні складнощі й тонкощі, які виникають під час роботи зі змінними.

Ідентифікатори

Ім'я змінної (її ідентифікатор) має бути складено з друкованих символів і цифр. Цифру не можна ставити на перше місце. Пробіли в найменуванні використовувати не можна. Замість них зазвичай застосовується символ підкреслення. Символ підкреслення є значущим. Так імена 'Var1', 'Var_1' і '_Var1' є різними. Два підкреслення поспіль використовувати не можна. Регістр літер не враховується. Так 'VAR1' і 'Var1' одне й те саме. Як мінімум, шість перших знаків ідентифікатора є значущими для всіх систем програмування.

У CODESYS такого обмеження немає – всі символи найменування є значущими. Символи кирилиці в ідентифікаторах застосовувати не можна. Це обмеження характерне для всіх програмних систем.

Аналогічні вимоги стосуються і будь-яких ідентифікаторів МЕК-програм (компоненти, мітки, типи тощо).

Розподіл пам'яті змінних

Контролер з точки зору МЕК програми має кілька областей пам'яті, що мають різне призначення:

- область входів ПЛК;
- область виходів ПЛК;
- область пам'яті з прямою адресацією;
- оперативна пам'ять користувача (ОЗП).

Апаратні ресурси ПЛК присутні в МЕК-проектах у неявній формі. Розміщення змінної в одній з трьох перших областей призводить до її зв'язку з певною апаратурою – входами, виходами або змінними системи виконання (діагностика модулів, настроювання параметрів ядра тощо). Розподіл змінних у цих областях визначається виробником ПЛК. Прив'язка до конкретних адрес задається за допомогою прямої адресації.

Для забезпечення переносимості програмного забезпечення прямі адреси потрібно використовувати тільки в розділі оголошень. У мовах програмування стандарту не передбачено операцій прямого читання входів-виходів. Цю роботу виконує система виконання. За необхідності для низькорівневого звернення виробником ПЛК поставляються спеціальні бібліотеки.

Оголошення змінної без префікса АТ фізично означає виділення їй певної пам'яті в області ОЗП. Розподіл доступної пам'яті ОЗП транслятор здійснює автоматично.

Змінні прийнято розділяти на глобальні та локальні за областю видимості. Глобальні змінні визначаються на рівні ресурсів проекту (VAR_GLOBAL) і доступні для всіх програмних компонентів проекту. Локальні змінні описуються з оголошенням компонента і доступні тільки всередині нього.

Опис будь-якого програмного компонента містить, як мінімум, один розділ оголошення локальних змінних VAR, змінних інтерфейсу VAR_INPUT, VAR_OUTPUT, VAR_IN_OUT і зовнішніх глобальних змінних VAR_EXTERNAL.

Найменування розділів оголошення змінних можуть містити додаткові ключові слова, уточнюючі спосіб застосування, як показано в таблиці 4.6.

Пряма адресація

Для створення змінної, що прямо адресується, використовується таке оголошення:

ім'я змінної АТ %пряма адреса тип;

Таблиця 4.6 – Ключові слова для уточнення способу застосування змінної

Ключове слово	Застосування змінної
RETAIN	Змінні потрібно розмістити в незалежній пам'яті, що зберігає значення з вимкненим живленням. Така пам'ять не є обов'язковою і є не у всіх ПЛК
CONSTANT	Константи, доступні тільки для читання

Пряма адреса починається з літери, яка визначає область пам'яті, як показано в таблиці 4.7.

Таблиця 4.7 – Визначення області пам'яті

Символ	Область пам'яті
I	Область входів
Q	Область виходів
M	Пам'ять, що прямо адресується

Далі йде символ, який визначає тип прямої адреси. У таблиці 4.8 наведено повний список символів, що визначає тип прямої адреси.

Таблиця 4.8 – Повний список символів, що визначає тип прямої адреси

Символ	Область пам'яті
ni	Біт
X	Біт
B	Байт
W	Слово
D	Подвійне слово
L	Довге слово

Завершує пряму адресу число – складена ієрархічна адреса, поля якої розділені крапкою. У найпростішому випадку використовується два поля

адреси: номер елемента і номер біта. На рис. 4.3 наведена схема побудови імені змінної, що прямо адресується.

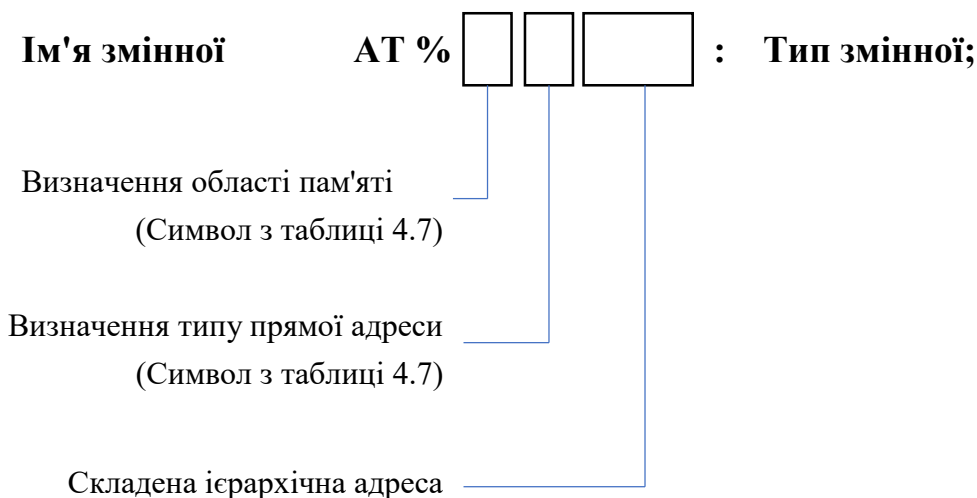


Рисунок 4.3 – Схема побудови імені змінної, що прямо адресується

В кінці оголошення, як і для змінних, що автоматично розміщуються, необхідно вказати тип змінної. Із зазначенням адреси одного біта тип змінної може бути тільки `BOOL`.

У прямій адресі вказується саме номер елемента. Це докорінно відрізняється від фізичних адрес мікропроцесора. Якщо пряму адресу визначає байт, то номер елемента – це номер байта. Якщо пряму адресу визначає слово, то номер елемента – це номер слова, і, відповідно, один елемент займає два байти. Так, наступні три записи адресують один і той самий байт:

```
dwHeat    AT %MD1:    BYTE;
wbHeat    AT %MW2:    BYTE;
byHeat    AT %MB4:    BYTE;
```

Нумерацію елементів пам'яті для даного прикладу ілюструє таблиця 4.9.

На випадок, якщо така схема виявиться чомусь неприйнятною, компілятор `CODESYS` має спеціальний прапорець, який примусово включає байтову адресацію для всіх типів.

Таблиця 4.9 – Приклад розподілу змінних у пам'яті

D	0				1			
W	0		1		2		3	
B	0	1	2	3	4	5	6	7

У кожній області пам'яті адресація елементів починається з нуля. Фізичне розміщення областей, які прямо адресуються в ОЗП, визначається конфігурацією контролера.

Очевидно, що зіставлення ідентифікаторів змінних прямим адресам є справою, що вимагає великої акуратності. Тому для складних модульних контролерів застосовуються спеціальні фірмові конфігуратори, що підключаються до оболонки комплексу програмування і дозволяють графічно «зібрати» ПЛК і визначити всі необхідні інтерфейсні змінні.

Входи ПЛК – це змінні з прямими адресами в області I. Вони доступні в прикладних програмах тільки за читанням. Виходи Q – тільки за записом. Змінні в області M доступні за записом і читанням.

В області пам'яті M розміщують зазвичай змінні, які не можна однозначно віднести до входів або виходів. Це можуть бути діагностичні ресурси модулів, параметри системи виконання тощо.

Прямі адреси можна використовувати в програмах безпосередньо, наприклад:

```
IF %IW4 > 1 THEN ... (*Значення входу IW4*)
```

Проте все ж бажано компактно зосередити в проекті всі апаратно-залежні моменти.

Зверніть увагу, що пряма адресація дозволяє розмістити кілька різнотипних змінних в одній і тій самій пам'яті. Наприклад, спеціально для швидкого обнулення 16-дискретних виходів (BOOL) можна використовувати змінну типу WORD. Або, наприклад, поєднати змінну STRING і декілька змінних типу BYTE, що дасть можливість організувати форматування виводу

без застосування рядкових функцій. Оскільки фізичний розподіл адрес відомий на етапі трансляції, компілятор формує максимально компактний код для таких об'єднань, чого не вдається досягти у роботі з елементами масиву, де потрібна динамічна адресація.

Порозрядна адресація

У стандарті передбачена зручна форма роботи з окремими бітами змінних типу бітових рядків – порозрядна адресація. Необхідний біт вказується через точку після ідентифікатора. Аналогічно можна використовувати окремі біти пам'яті, що прямо адресується. Молодшому біту відповідає нульовий номер. Порозрядна нумерація не має перевищувати межі відповідного типу числа.

```
VAR
```

```
    a: WORD;
```

```
    bStop AT %IX64.3:    BOOL;
```

```
ENDVAR
```

```
a := 0;
```

```
a.3 := 1;          (*або a.3 := TRUE; – результат 2#0000_1000*)
```

```
a.18 := TRUE;     (*помилка, в WORD не може бути біт a.18*)
```

```
IF a.15 THEN ...  (*а менше нуля?*)
```

Перетворення типів

Перетворення типів відбувається з присвоєнням значення змінної одного типу змінній іншого типу. Перетворення змінює фізичне подання значення у пам'яті даних, але не має змінювати саме значення. Якщо це неможливо, то перетворення призводить до часткової втрати даних. Але в такому випадку транслятор вимагає явної вказівки необхідності виконання такої операції.

Розглянемо спочатку роботу з цілими числами. Нехай, наприклад, оголошена змінна *siVar* типу коротке ціле (SINT 8 біт) і змінна *iVar* типу ціле (INT 16 біт). Припустимо *siVar* = 100 а *iVar* = 1000. Вираз *iVar* := *siVar* є цілком допустимим, оскільки числа типу SINT є підмножиною INT (*iVar* прийме значення 100). Тут перетворення типу буде виконано транслятором

автоматично, без будь-яких додаткових вказівок. Зворотне присвоєння `siVar := iVar` призведе до переповнення і втрати даних. Змусити транслятор виконати перетворення з імовірною втратою даних можна тільки в явній формі за допомогою спеціального оператора `siVar := INT_TO_SINT(iVar)`. Результат дорівнює 24 (в шістнадцятковій формі 1000 це 16#03E8 і тільки молодший його байт перейде в SINT, значення 16#E8 відповідає десятковому числу 24).

Аналогічна ситуація виникає під час роботи з дійсними числами довгого LREAL і короткого REAL типів.

Оператори явного перетворення базових МЕК-типів утворюють свої найменування з двох частин. Спочатку вказується «вихідний тип», потім «_TO_» і «тип результату». Наприклад:

```
si := INT_TO_SINT(16#55AA);      (* Результат 16#AA*)
si := TIME_TO_SINT(T# 120ms);    (*120*)
i := REAL_TO_INT(2.7);           (*Результат 3*)
i := TRUNC(2.7);                 (*Результат 2*)
t := STRING_TO_TIME('T#216ms'); (*Результат T#116ms*)
```

Операція TRUNC виконує відкидання дробової частини на відміну від перетворення REAL_TO_INT, що виконує округлення.

Зверніть увагу, що операції перетворення допустимі для будь-яких комбінацій базових типів, а не тільки для сумісних типів (наприклад, дату в рядок). Так, перетворення `<...> _TO_STRING` фактично замінюють оператор PRINT, поширений у мовах загального застосування.

У конкретній реалізації окремі перетворення можуть не підтримуватися або мати певні особливості, насамперед це стосується перетворень рядків в інші типи і назад.

Формат BCD

Двійково-кодований десятковий формат подання BCD (binary coded decimal) являє собою числа в позиційній десятковій системі, де кожна цифра числа займає 4 біти. Наприклад, десяткове число 81 буде подано у вигляді `2#1000_0001`. Арифметичні операції з BCD-числами вимагають застосування

спеціального математичного апарату, малоефективні порівняно зі звичайним двійковим поданням. Але, з іншого боку, VCD виявляється дуже зручним у процесі організації клавіатурного вводу та індикації. Наприклад, функції виведення числа на принтер або навіть на сегментний індикатор виходять тривіальними (одна одновимірної таблиця на 10 констант).

Для зберігання чисел у форматі VCD стандарт МЕК пропонує використовувати змінні типів ANY BIT (крім BOOL, звичайно). Арифметика VCD-обчислень зазвичай не підтримується у стандартному комплекті бібліотек систем програмування ПЛК. У бібліотеці утиліт CODESYS реалізовані дві прості функції VCD перетворення: VCD TO INT і INT TO VCD.

4.1.5 Визначення компонента

Компоненти організації програм є базовими елементами, з яких будується код проекту. Аналогічно електронні пристрої складаються зазвичай з модулів. Кожен компонент програми має власне найменування, певний інтерфейс і опис однією з МЕК-мов.

Один компонент може викликати інші компоненти. Виклик самого себе (рекурсія) у стандарті МЕК не дозволена. Комбінувати різні мови в одному проекті можна з описом різних компонентів, але окремий компонент цілком реалізується однією мовою МЕК. Під час виклику компонента мова його реалізації значення не має.

До компонентів організації програм у МЕК-стандарті належать функції, функціональні блоки і програми. Всі вони багато в чому схожі, але мають певні особливості й різне призначення.

Компонент має властивість інкапсуляції – працює як «чорний ящик», приховуючи деталі реалізації. Для роботи з компонентом достатньо знати його інтерфейс, що включає опис входів і виходів. Внутрішню його будову знати необов'язково. У графічній формі подання компонент виглядає як прямокутник з входами зліва і виходами справа. Локальні (внутрішні) змінні компонента недоступні ззовні і в графічному поданні не відображаються.

Завдяки інкапсуляції компоненти успішно вирішують завдання структурної декомпозиції проекту. На верхньому рівні подання ми працюємо з більшими компонентами. Кожен з них виконує значну для даного проекту

задачу. Зайві подробиці на цьому рівні тільки заважають розумінню проблеми. Розкриваючи вкладені компоненти один за одним, ми можемо дістатися до самого детального подання.

Готовий компонент завжди можна розкрити, вивчити й поправити. Це стосується тільки призначених для користувача компонентів і відкритих бібліотек. Деякі стандартні компоненти включені в транслятор і не доступні для перегляду і зміни. Це стосується і зовнішніх бібліотек. Зовнішні бібліотеки реалізуються у вигляді об'єктного коду за допомогою зовнішніх засобів, наприклад компілятор С або асемблера. Можливо навіть, що компонент реалізований не тільки програмно, а використовує допоміжні апаратні засоби, наприклад годинник реального часу або математичний співпроцесор.

Ще одним завданням, що вирішується компонентами, є локалізація імен змінних. Це означає, що в різних компонентах можна використовувати повторювані імена. Так, наприклад, улюблену змінну з оригінальним ідентифікатором «X» можна використовувати в кожному компоненті, і щоразу це буде нова змінна. Область видимості локальних змінних визначається рамками одного компонента. Обмеження області видимості є обов'язковим у всіх сучасних системах програмування.

Екземпляри функціональних блоків, оголошені всередині інших компонентів, також мають локальну область видимості. Програми та функції завжди визначені глобально.

Оголошення POU

Реалізації будь-якого POU завжди має передувати розділ оголошень. Оголошення функції, функціонального блоку і програми починаються відповідно з ключових слів FUNCTION, FUNCTION_BLOCK і PROGRAM. За ним йде ідентифікатор (ім'я компонента). Далі визначається інтерфейс POU. До інтерфейсу компонента належать входи VAR_INPUT, виходи VAR_OUTPUT і змінні типу вхід-вихід VAR_IN_OUT. Завершують розділ оголошень локальні змінні VAR.

У функціях розділи VAR_OUTPUT і VAR_IN_OUT відсутні. Виходом функції є єдина змінна, що збігається з ім'ям функції. Тип значення, що

повертається, вказується під час визначення ідентифікатора через двокрапку.
Наприклад: FUNCTION iNearby : INT.

Структура розділу оголошень POU показана в таблиці 4.10.

Таблиця 4.10 – Структура розділу оголошень POU

Тип POU	Функція	Функціональний блок	Програма
	FUNCTION ім'я: ТИП	FUNCTION_BLOCK ім'я	PROGRAM ім'я
Інтерфейс	VARINPUT	VAR_INPUT	VAR_INPUT
	—	VAR_OUTPUT	VAR_OUTPUT
	—	VAR_IN_OUT	VAR_IN_OUT
Локальні змінні	VAR	VAR	VAR

Всі розділи змінних є не обов'язковими.

Формальні й актуальні параметри

Інтерфейс компонента утворюється вхідними та вихідними змінними. Інтерфейсні вхідні змінні називають формальними параметрами. З використанням компонента його формальні параметри зв'язуються з актуальними параметрами. І нарешті, під час виклику параметри компонента набувають актуальних або поточних значень. Ці поняття необхідні для усунення двозначності під час опису техніки роботи з компонентами.

Пояснимо їх відмінності на прикладі. Візьмемо стандартний блок R_TRIG. Він має вхід з назвою CLK. Ми використовуємо його в програмі, в якій визначена певна підходяща змінна, наприклад bPulse. Під час виклику блоку з нашої програми ми подаємо bPulse на вхід CLK. Далі програма компілюється і завантажується в контролер. Змінна bPulse набуває деякого значення, наприклад TRUE. Вхід CLK, відповідно, теж матиме значення TRUE. Тут відмінності вже практично очевидні. CLK – це формальний параметр, bPulse – актуальний параметр, а TRUE – фактичне значення. З формальними параметрами доводиться мати справу в ході

проектування ROU й опису його інтерфейсу. Актуальні параметри працюють під час використання компонента. Поточні значення народжуються тільки в «залізі» у процесі виконання.

Параметри і змінні компоненти

Під час оголошення ROU можна зустріти такі заголовки:

Формальні вхідні параметри VAR_INPUT

Передаються ROU за значенням шляхом копіювання. Під час виклику блоку такої змінної можна привласнити значення іншої змінної (сумісного типу), константи або виразу. Будь-які зміни такої змінної всередині ROU жодним чином не відображаються на дані компонента, який викликає. Застосовується в будь-яких ROU. Можуть мати значення за замовчуванням. Відображаються в графічному поданні з лівого боку компонента.

Формальні вихідні параметри VARJDUTPUT

Відображають результати роботи компонента. Передаються з ROU за значенням шляхом копіювання. Читання значення виходів зазвичай має сенс після виконання блоку. Поза компонента параметри VAR_OUTPUT доступні тільки за читанням. Не використовуються у функціях, оскільки функція має тільки одне значення, що повертається. Можуть мати початкові значення. Відображаються у графічному поданні справа.

Параметр типу VAR_IN_OUT

Цей параметр водночас є входом і виходом. Передача змінної екземпляру блоку виконується за посиланням. Це означає, що зовнішня змінна немовби сама працює всередині блоку на правах внутрішньої змінної. До компонента передається тільки адреса її розташування в пам'яті даних. Для змінної VAR_IN_OUT не можна:

- використовувати її у функціях;
- привласнювати початкове значення;
- звертатися як до елемента структури даних, через точку;
- привласнювати константу як актуальний параметр.

Присвоєння зовнішньої змінної для VAR_IN_OUT можна робити тільки з викликом блоку.

Найважливішою властивістю VAR_IN_OUT є відсутність копіювання зовнішніх даних. Параметри VAR_INPUT і VAR_OUTPUT можуть оперувати з масивами і структурами, але щоразу зі зверненням до компонента відбуватиметься повне копіювання даних. Це може забирати багато часу. Присвоєння одного масиву іншому для VAR_IN_OUT означає фактично переключення компонента з одного масиву на інший. Локальна копія даних у цьому випадку не створюється.

Як і глобальні змінні, параметри VAR_IN_OUT порушують ідеологію незалежності компонентів. Правильний компонент не повинен мати можливості зіпсувати чужу пам'ять. Тому застосовувати їх потрібно дуже акуратно й тільки у випадках, коли це дійсно необхідно.

Локальні змінні VAR

Доступні тільки всередині компонента, поза компонента доступу немає. Можуть мати початкові значення. Для функцій локальні змінні розміщуються в динамічній пам'яті (зазвичай у стеку). Після закінчення роботи функції пам'ять звільняється і може використовуватися в інших функціях. У програмах і екземплярах функціональних блоків змінні VAR зберігають свої значення між викликами програм і екземплярів. У графічному поданні компонента локальні змінні не відображаються.

4.1.6 Функції

Функція – це програмний компонент, що відображає множину значень вхідних параметрів на вихід. Функція завжди повертає тільки одне значення. З оголошенням функції вказується тип значення, ім'я функції і список вхідних параметрів. Виклик функції проводиться за ім'ям із зазначенням вхідних параметрів. Функція може використовуватися в математичних виразах поряд з операторами і змінними. Функція не має внутрішньої пам'яті. Це означає, що функція з одними й тими самими значеннями вхідних параметрів завжди повертає одне й те саме значення. Функція – це чистий код. Багаторазове використання функції не призводить до повторного включення коду функції при компонуванні. Реалізація функції присутня в коді проекту тільки один раз. Щоразу з виконанням функції процесор виконує той самий поіменований код.

Функція може мати локальні (тимчасові) змінні. Але по закінченні своєї роботи функція звільняє локальну пам'ять.

Тип функції (тип значення) може бути будь-яким з числа стандартних типів даних або типів, створених користувачем. Тіло функції може бути описано мовами IL, ST, LD або FBD. Використовувати в ГС не можна. З функції можна викликати бібліотечні функції та інші функції поточного проекту. Викликати функціональні блоки і програми з функцій не можна.

Виклик функції з перерахуванням значень параметрів

У працьових мовах ST – мовах Паскаль і С виклик функції проводиться за ім'ям з перерахуванням в дужках списку актуальних вхідних параметрів, через кому, зліва направо. Аналогічний спосіб прийнятний і в мові ST. Наприклад:

$Y := \text{MUX}(0, x1, x2);$ (*Повертає нульовий вхід – X_1 *)

Тут необхідно звернути увагу на те, що найменування параметрів не важливі. З перерахуванням параметрів важливо тільки дотримуватися правильної послідовності відповідно до визначення в оголошенні функції. У графічних мовах порядок вхідних параметрів заданий напрямком зверху донизу (рис. 4.4).

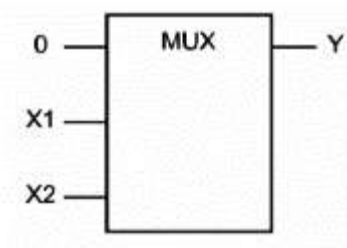


Рисунок 4.4 – Графічне відображення виклику функції

Деякі функції можуть мати рівноцінні параметри, тоді їх порядок очевидно неважливий. Наприклад:

$y := \text{MAX}(x1, x2);$ (*Повертає найбільше зі значень вхідних параметрів *)

$y := \text{MAX}(x2, x1);$ (*Результат такий самий*)

Присвоєння значень параметрам функції

Другий спосіб виклику функції передбачає безпосереднє присвоювання значень параметрам функції за іменами:

```
stResult := CONCAT(STR1 := 'Висока', STR2 := 'температура');
```

Це те саме:

```
stResult:= CONCAT('Висока', 'температура');
```

або:

```
stResult := CONCAT('Висока', STR2 := 'температура');
```

Якщо у програмі вже визначена змінна з ім'ям, що збігається з найменуванням вхідного параметра (STR1 := 'Висока;'), то такий запис може викликати нерозуміння:

```
stResult:= CONCAT(STR1 := STR1, STR2 := 'температура');
```

Насправді тут все правильно: зліва від знака присвоювання – параметр функції, праворуч – змінна.

Описаний спосіб виклику функції передбачає можливість задавати параметри в довільному порядку і опускаючи деякі з них. Версія CODESYS 2.x не забезпечує функцій таку можливість. Єдиний сенс такої нотації – в універсальності, прийнятній для функціональних блоків і програм.

Передача параметрів функції завжди відбувається шляхом копіювання. За будь-якого способу виклику функція отримує локальні копії значень змінних.

Функції зі змінною кількістю параметрів

Для багатьох функцій важко передбачити, скільки значень потрібно буде обробити в конкретному випадку. Наприклад, для функції AND можна

обмежитися двома входами і використовувати «драбинку» викликів функцій для обробки більшої кількості змінних (рис. 4.5).

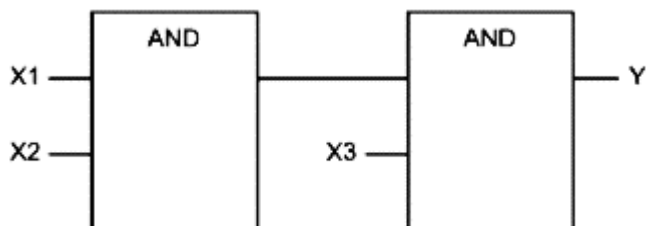


Рисунок 4.5 – З'єднання двовходових AND

На рис. 4.5 подана не дуже практична конструкція. Було б значно зручніше мати функцію, яка могла б «розширюватися» і адаптуватися під змінну кількість параметрів. Така реалізація показана на рис. 4.6.

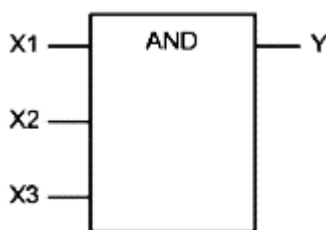


Рисунок 4.6 – Розширювана реалізація AND

Стандарт МЕК дійсно передбачає таку можливість. У текстових мовах розширення проводиться додаванням змінних у кінець списку параметрів:

```
y := MUX(x_n, x1, x2, x3, x4, x5);
```

Через складність реалізації транслятора змінна кількість параметрів у призначених для користувача функціях не використовується.

Оператори і функції

Оператори – це символи певних операцій. Але їх можна визначити і як функції, наділені певними привілеями. По-перше, код для операторів транслятор створює сам і не вимагає підключення будь-яких бібліотек. По-друге, багато операторів мають особливі форми запису у виразах ST.

Наприклад, математичні оператори (додавання, віднімання, множення і ділення) мають традиційне символічне представлення у текстових мовах (+, -, *, /). У графічних мовах оператори виглядають як звичайні функції.

Можна обходитися без символічного подання операторів. Наприклад:

```
Y := SUB(MUL(4,x),3);
```

Але символічне подання у ST виглядає значно краще:

```
Y := 4 * X - 3;
```

Можна записати ще коротше:

```
Y = 4x - 3.
```

Всі три записи рівноцінні за змістом. Символьні вирази зрозуміліші і дають змогу більше сконцентруватися на суті виразу, а не на формі його подання.

Під час роботи з операторами необхідно звертати увагу на наявність символічної форми подання. Так, для математичних і логічних операторів у мові ST, як правило, допускається тільки символічне подання. Вираз

```
Y := AND(x1, x2)
```

викличе помилку компіляції. Необхідно писати так:

```
Y := x1 AND x2;
```

Якщо оператор не має символічного подання, то на нього поширюються звичайні правила виклику функцій. Наприклад:

```
y := SQRT(x);
```

Зверніть увагу, що імена вхідних параметрів для операторів в описі не задані. Це означає, що викликати такі функції в ST можна тільки перерахуванням параметрів.

4.1.7 Функціональний блок

Функціональний блок – програмний компонент, що відображає множину значень вхідних параметрів на множину вихідних. Після виконання екземпляру функціонального блоку всі його змінні зберігаються до наступного виконання. Отже, функціональний блок, що викликається з одними й тими самими вхідними параметрами, може виробляти різні вихідні значення. Зберігаються всі змінні, включаючи вхідні та вихідні. Так, якщо ми викличемо екземпляр функціонального блоку, не визначаючи значення деяких вхідних параметрів, він використовуватиме раніше встановлені значення. Можливість завдання змінної кількості вхідних значень закладена за визначенням і не вимагає яких-небудь додаткових зусиль. Ззовні доступні тільки входи і виходи функціонального блоку, отримати доступ до внутрішніх змінних блоку не можна.

З позицій об'єктно-орієнтованого програмування (ООП) функціональні блоки – це об'єкти, які добре реалізують інкапсуляцію, тобто приховування деталей реалізації. Об'єднання коду і даних в одному модулі споріднює функціональні блоки з класами ООП. Можливість успадкування і поліморфізм, на жаль, поки відсутні.

Створення екземпляра функціонального блоку

Перш ніж використовувати функціональний блок, необхідно створити його екземпляр. Ця операція аналогічна за змістом оголошенню змінної. Описавши новий блок, ми фактично створили новий тип даних, подібний структурі. Кожен функціональний блок може мати будь-яку кількість екземплярів. Так, різні екземпляри блоку «таймер» абсолютно незалежні один від одного. Кожен з них має власні настройки і живе власним життям.

Кожен екземпляр функціонального блоку має свій власний ідентифікатор і свою область у статичній пам'яті даних. Оголошення ще одного екземпляру блоку призводить до виділення ще однієї області в пам'яті даних. Код, очевидно, як і для функції, залишається спільним (рис. 4.7).

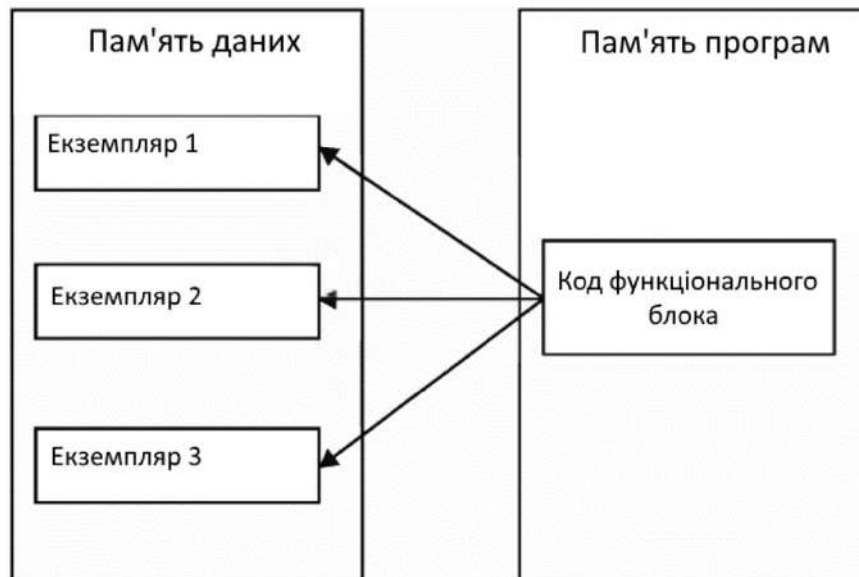


Рисунок 4.7 – Розподіл пам'яті для екземплярів функціональних блоків

Екземпляр функціонального блоку створюється в розділі оголошень змінних функціонального блоку, програми або в розділі глобальних змінних проекту. Як і змінні, він має отримати унікальний ідентифікатор. Наприклад, створення екземпляра стандартного функціонального блоку «інкрементний лічильник» з ідентифікатором `ctuTimeMeter` має такий вигляд:

```
ctuTimeMeter: STU;
```

Очевидно, що створювати екземпляри можна тільки для відомих системі блоків. Це бібліотечні блоки або блоки, раніше реалізовані користувачем. З точки зору транслятора, створення екземпляра означає виділення необхідної пам'яті для розміщення змінних блоку.

Екземпляр функціонального блоку можна не тільки викликати, але й використовувати як вхідні змінні інших функціональних блоків.

Функціональним блоком іноді називають екземпляр функціонального блоку. У даній книзі такі неоднозначні скорочення не застосовуватимуться. Дозволимо собі лише називати іноді функціональний блок просто блоком, а екземпляр функціонального блоку – екземпляром.

Доступ до змінних екземпляра

Після створення екземпляра функціонального блоку можна відразу почати працювати з його даними. При цьому зовсім не обов'язково викликати його. Звертатися до змінних екземпляра можна так само, як до елементів структури даних, через точку:

```
ctuTimeMeter.RESET := FALSE;  
ctuTimeMeter.PV := 100;  
x := ctuTimeMeter.CV;
```

Входи екземпляра блоку доступні для запису і читання ззовні. Виходи – тільки для читання. Змінювати значення виходів можна тільки з тіла блоку, ззовні не можна. Транслятор відстежує такі спроби і видає повідомлення про помилку.

Виклик екземпляра блоку

Викликати екземпляр функціонального блоку з перерахуванням параметрів як функцію не можна. Значення вхідних змінних мають присвоюватися безпосередньо. У текстових мовах вхідні змінні перераховуються в дужках, після імені екземпляра. Присвоєння вхідних значень виконується операцією «:=».

Мовою ST:

```
ctuTimeMeter (RESET := FALSE);
```

Мовою IL:

```
CAL ctuTimeMeter (RESET := FALSE)
```

Спеціальний символ «=>» дозволяє отримати значення виходів після виконання блоку:

```
ctuTimeMeter (RESET := FALSE, CU := Inp1, CV => x);
```

Під час виклику екземпляра можна визначити тільки необхідні параметри, причому в довільному порядку. У графічних мовах невикористовувані входи й виходи екземпляра блоку просто залишаються не підключеними. Як показано на рис. 4.8, вхід PV залишився не підключеним.

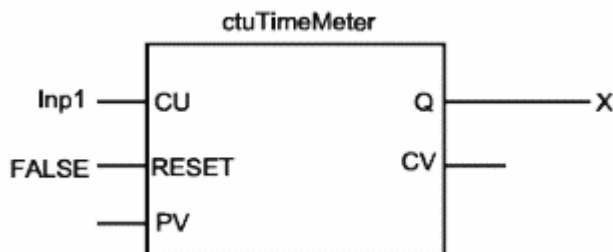


Рисунок 4.8 – Виклик екземпляра функціонального блоку (FBD)

У мові ST за відсутності параметрів порожні дужки після імені екземпляра ставити не потрібно:

```
ctuTimeMeter(); (*зайві дужки*)
```

Використовувати екземпляри функціональних блоків у виразах не можна, але можна використовувати їх входи і виходи:

```
X := ctuTimeMeter.PV - ctuTimeMeter.CV + 1;
```

Можна визначити значення входів заздалегідь і викликати екземпляр функціонального блоку взагалі без параметрів.

Мовою ST:

```
ctuTimeMeter.RESET := FALSE;
ctuTimeMeter;
```

Мовою IL:

```
LD FALSE
ST ctuTimeMeter.RESET
CAL ctuTimeMeter
```

Потрібно звернути увагу на те, що МЕК не заохочує використання елементів даних окремо від виклику екземпляра блоку, оскільки це може привести до проблем з використанням екземпляра блоку в багатозадачних проектах. З іншого боку, багаторазове повторне присвоювання вхідних значень збільшує розмір коду і знижує ефективність програми.

Ініціалізація даних екземпляра

Під час опису блоку в розділі оголошень можна явно привласнити початкові значення змінним. Наприклад:

```
FUNCTION_BLOCK SyncSwitch
VAR_INPUT
...
Sync:      BOOL := TRUE;
```

Зі створенням екземпляра функціонального блоку SyncSwitch вхідна змінна Sync отримає значення TRUE. Якщо початкові значення не задані, використовуються нульові значення.

Екземпляр функціонального блоку може потребувати індивідуальної ініціалізації, відмінної від тієї, яка визначена в процесі реалізації. Установку початкових значень змінних найпростіше виконати під час створення екземпляра. Значення, задані у ході створення екземпляра, сильніше значень, заданих під час реалізації блоку.

```
SyncSw1: SyncSwitch := (Sync := FALSE);
```

Тепер змінна SyncSw1.Sync отримає початкове значення FALSE, незважаючи на значення, вказане в оголошенні блоку.

Фізично початкові значення змінних отримують ще до першого використання екземпляра. Операція початкової ініціалізації змінних проводиться зі скидання, яке виконується безпосередньо після завантаження

проекту в пам'ять ПЛК, за командою відладчика або з перезапуском контролера.

Деякі транслятори мають опцію відключення ініціалізації за замовчуванням з метою прискорення запуску ПЛК. У цьому випадку покладатися на те, що змінні, які не мають явно зазначених початкових значень, отримуватимуть однакові значення з перезапуском системи, не можна.

Можливі випадки, коли екземпляру функціонального блоку потрібна розумна ініціалізація. Наприклад, для настроювання блоку необхідно провести деякі обчислення. Спеціальної процедури ініціалізації в функціональних блоках не передбачено. Тут доведеться витратити на ініціалізацію один або кілька перших циклів виконання екземпляра. Закінчення складної процедури ініціалізації повідомляють зазвичай індикацією виходом готовності (ENO). Часто буває, що зручно застосувати для ініціалізації дію і зосередити контроль над ініціалізацією в одному місці (зазвичай за крок Init SFC діаграми). Такий метод дозволяє проводити ініціалізацію даних, екземплярів блоків і програм в необхідній послідовності і взаємозв'язку. У більшості ж практичних випадків для блоків, які потребують певної настройки, виявляється достатнім ввести кілька спеціальних входів (уставок). Так зроблено в усіх стандартних блоках.

Тиражування екземплярів

За необхідності отримати копію екземпляра функціонального блоку можна, використовуючи оператор присвоювання:

```
SyncSw, SyncSw2: SyncSwitch; (*працюємо з екземпляром SyncSw1*)
```

```
...
```

```
Мовою ST:
```

```
SyncSw2 := SyncSw1;
```

Мовою LD реалізація цього прикладу показана на рис. 4.9.

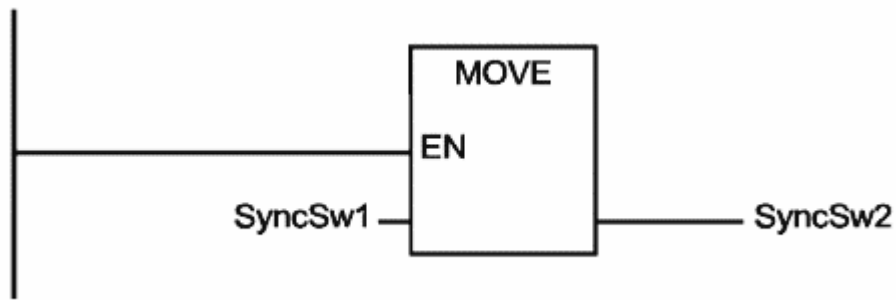


Рисунок 4.9 – Присвоювання екземплярів функціонального блоку

Оператор присвоювання виконує побайтне копіювання даних екземпляра блоку за аналогією зі структурами даних. Очевидно, що така техніка копіювання підходить тільки для екземплярів одного й того самого функціонального блоку.

Особливості реалізації та застосування функціональних блоків

Вхідні змінні всередині блоку доступні для запису. Це викликає певний інтерес для програміста. Так, наприклад, вхідну змінну зручно застосувати як лічильник ітерацій, якщо вона саме відображає кількість потрібних повторень. Це дозволить уникнути створення додаткової локальної змінної. Оскільки під час виклику екземпляра блоку ввідна змінна має отримати нове значення, нічого страшного на перший погляд немає. Виклик екземпляра не зобов'язаний супроводжуватися привласненням значень усім формальним параметрам.

Можливо, у певний момент ви вирішите, що вхідний параметр вже визначено, і можна не ставити його повторно. В результаті значення параметра дорівнюватиме тим значенням, які він мав під час роботи в ролі локальної змінної, при попередньому виклику екземпляра. Звичайно, можна придумати багато прикладів, коли зміна значення вхідної змінної безпечно. Проте використовувати такий прийом потрібно виключно обдумано і обережно. У загальному випадку в ході реалізації блоку вхідні змінні потрібно розглядати як константи.

Застосування глобальних змінних у функціональних блоках викликає ті самі проблеми, що і в функціях. Екземпляри блоку перестають бути незалежними. Зміни змінної, виконані одним екземпляром, проявлять себе

зовсім в іншому місці. Іноді це дійсно необхідно, але у звичайній практиці бажано обмежувати таке застосування глобальних змінних.

Якщо екземпляри функціонального блоку використовують глобальну змінну тільки для читання, то жодних побічних явищ виникнути не може. Аналогічна ситуація виникає в ході застосування змінних, що мають пряму адресацію.

Зі входами проблем немає. Єдине обмеження – це погіршення можливостей блоку в плані його повторного застосування. З використанням блоку в іншій програмі або проекті прямі адреси доведеться поправити. Добре рішення цієї проблеми дають шаблонні змінні.

Шаблонні змінні

Шаблонні змінні або, як їх іноді називають, конфігураційні змінні (в CODESYS variable configuration) є частково визначеними змінними з прямою адресацією. Прямий доступ замінюється зірочкою. Визначення повної адреси шаблонної змінної дається у спеціальному розділі ресурсів.

Розглянемо, як це робиться на прикладі. Створимо блок SHABLON, який матиме стандартну змінну bMarvel. Для нашого прикладу достатньо розділу оголошень:

```
FUNCTION_BLOCK SHABLON
VAR
bMarvel AT %I* : BOOL;
END_VAR
```

Далі створимо два екземпляри блоку SHABLON у головній програмі:

```
PROGRAM PLC_PRG
VAR
Shablon1: SHABLON;
Shablon2: SHABLON;
END_VAR
```


Їх потрібно прописати в ресурсах проекту:

```
VAR_CONFIG
PLC_PRG.Shablon1.bMarvel AT %IX1.0 : BOOL;
PLC_PRG.Shablon1.bMarvel AT %IX1.1 : BOOL;
END_VAR
```

Типи даних шаблонної змінної, які зазначені в оголошенні блоку і в ході настроювання адреси в ресурсах, мають співпадати.

Приклад функціонального блоку

Як простий приклад реалізуємо блок синхронного перемикача SyncSwitch. Алгоритм його роботи такий: вихід перемикача Q приймає значення, які рівні входу Start, але перемикання виходу дозволено тільки при Sync: = TRUE. Графічно це відображено на рис. 4.10.

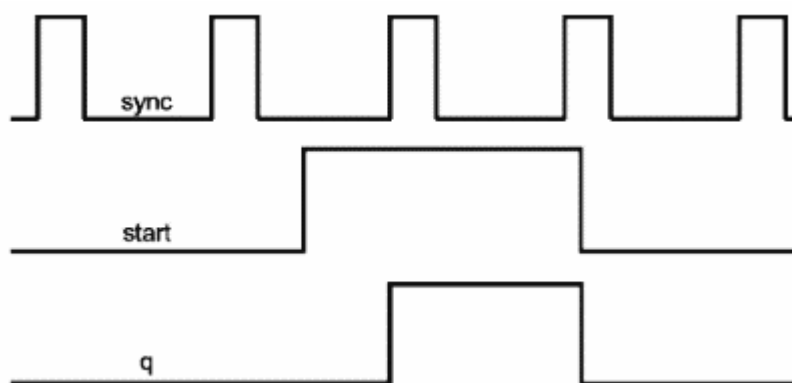


Рисунок 4.10 – Діаграма роботи синхронного перемикача

Умова включення виходу виражається рівнянням: $Q = \text{start AND sync}$, а умова виключення $Q = \text{NOT start AND sync}$. Значення виходу має зберігатися між синхроімпульсами, тому використовувати тут функцію не можна. Мовою ІЛ блок SyncSwitch можна реалізувати так:

```
FUNCTION_BLOCK SyncSwitch VAR INPUT
    Sync:      BOOL
```

Start:	BOOL
END_VAR	
VAR_OUTPUT	
Q:	BOOL
END_VAR	
LD	Sync
AND	Start
S	Q
LD	Sync
ANDN	Start
R	Q

Такий блок корисний під час реалізації «безударного» перемикаччя в ланцюзі змінного струму. Імпульси синхронізації мають відображати інтервали, коли миттєве значення напруги близьке до нуля. Тоді перемикаччя силового ланцюга з виходу SyncSwitch відбуватиметься без кидка струму.

Звичайно, практичне значення нашого нового блоку дещо програє, якщо згадати про стандартний доміантний перемикач – SR. З ним завдання вирішується в один рядок (ST):

```
SR_1(SET1:= Start AND Sync, RESET:= Sync, Q1=> Q);
```

Дії

У функціональних блоках МЕК дуже не вистачає можливості виконувати декілька різних операцій. Особливо, якщо блок містить об'ємні дані. Можна, звичайно, зробити додатковий вхід і по ньому аналізувати, що ми хочемо від блоку. У CODESYS ця проблема вирішена так. Функціональні блоки і програми можна доповнювати діями. Дія працює всередині блоку з повним правом доступу до всіх даних. Її можна викликати як з тіла блоку, так і ззовні. Дія вказується через точку після назви екземпляра блоку і може мати перелік значень входів і виходів. Під час виклику дії з тіла блоку найменування екземпляра не потребується. Дія не має власних даних і

використовує входи, виходи і локальні змінні блоку. Мова реалізації дії може бути довільною.

Для прикладу доповнимо вищеописаний блок SyncSwitch дією EmergencyBreak. Нехай його виклик призводить до миттєвого безумовного виключення виходу. Для визначення дії потрібно обрати блок в організаторі об'єктів CoDeSys і дати команду «Add Action». Опишемо дію мовою ST:

```
Q := FALSE;
```

Викликати дану дію з ST-програми можна так:

```
SyncSwitch1.EmergencyBreak(Q => q);
```

У графічних мовах прямокутник, який представляє дану дію, матиме заголовок SyncSwitch1.EmergencyBreak.

Зверніть увагу, що вікно редактора для дій не має розділу оголошень. Компоненти програм з діями мають списки дій, що розкриваються в організаторі об'єктів. Список дій у розділі оголошень ROU не відображається.

Дії аналогічні методам класу в C ++. Термін «дія» навіть більш зрозумілий, ніж «метод». Своїм походженням дії зобов'язані SFC. У CODESYS дії можна використовувати як підпрограми.

4.2 ПЛК, як кінцевий автомат

Як було описано вище, ПЛК функціонує циклічно – читання входів, виконання прикладної програми і запис виходів. В результаті прикладне програмування для МЕК ПЛК істотно відрізняється від традиційної моделі, яка застосовується під час роботи мовами високого рівня ПК. Розглянемо як ілюстрації найпростішу задачу: необхідно запрограмувати мигаючий світловий індикатор. Очевидно, що алгоритм має бути приблизно такий:

- увімкнути вихід;
- витримати паузу;
- вимкнути вихід;

- витримати паузу;
- перехід до кроку 1 (початок програми);
- кінець програми.

Реалізована за цим алгоритмом програма для ПЛК не працюватиме. По-перше, вона містить безкінечний цикл. Весь код прикладної програми виконується від початку і до кінця в кожному робочому циклі. Будь-яка прикладна програма ПЛК є частиною робочого циклу і має повертати управління системі виконання. Тому крок 5 «перехід на початок програми» зайвий.

Якщо в нашому алгоритмі видалити «перехід на початок», програма працюватиме. Хоча і не так, як задумано. Вихід завжди залишатиметься у вимкненому стані, оскільки фізично встановлення значень виходів здійснюється після закінчення прикладної програми один раз. Проміжні зміни значень виходів не відображаються на апаратні засоби. Звичайно, значення змінної змінюватиметься багаторазово, але визначальним для виходу стане тільки останнє значення.

Що ще поганого для ПЛК у даному алгоритмі, так це затримка часу. Цілком імовірно, що, крім мигання одним виходом, ПЛК має буде виконувати ще й іншу роботу. Тобто програму необхідно буде доповнювати. Але якщо контролер зайнятий очікуванням, то в даному алгоритмі це означає, що нічого іншого він робити не зможе. Отже, затримку часу необхідно організувати інакше. Достатньо засікти час і зайнятися іншими справами, контролюючи періодично годинник. Тут немає нічого особливого. Так роблять зазвичай і більшість людей в очікуванні призначеного часу.

З урахуванням наведених міркувань алгоритм мигаючого індикатора для ПЛК має бути таким:

1. Перевірити таймер, якщо час паузи вийшов, то:
 - інвертувати вихід (включити, якщо вимкнений, і навпаки);
 - почати відлік нової паузи;
2. Кінець програми.

Незважаючи на описані складності, алгоритм вийшов в результаті простіше. Так і має бути. Технологія ПЛК спеціально орієнтована на подібні завдання.

Щоб писати хороші програми для ПЛК, потрібно навчитися думати певним чином. Секрет полягає в тому, щоб уявляти собі контролер не як машину, що послідовно виконує команди програми, а як кінцевий автомат.

У будь-якому автоматі існує множина входів (X), множина виходів (Y) і множина можливих станів (S). У нашому випадку це кінцеві множини, оскільки число входів-виходів ПЛК обмежена, так само, як і обсяг пам'яті змінних (що визначають можливі стани). Початковий стан ($s_0 \in S$) однозначно визначено. Автомат працює за тактами, для ПЛК це робочий цикл. У кожному такті значення входів відомі. Значення виходів визначаються (функція виходів λ) значеннями входів і поточним станом. Реакція автомата залежить тільки від поточного стану без передісторії, тобто не має значення, як він прийшов у даний стан. Разом з тим поточний стан також змінюється за тактами, автомат переходить у новий стан (функція переходів δ). У теорії автоматів описані шість об'єктів $A = \{X, Y, S, s_0, \lambda, \delta\}$ прийнято називати кінцевим автоматом Мілі.

Класична сфера застосування ПЛК – це програмна реалізація автоматів. Саме це і зумовило підхід до програмування ПЛК. Контролер обчислює програмно задану функцію виходів і функцію переходів. У кожному робочому циклі ПЛК виконує розрахунок нових значень для виходів, які необхідно змінити. В результаті класична прикладна програма ПЛК виявляється більш схожою на обчислення за формулою.

Дещо розширивши поняття автомата, ми можемо розглядати переходи як функції подій. Події не обов'язково мають бути пов'язані зі входами, це досить абстрактне поняття. Тоді закінчення часу очікування можна буде просто розуміти як подію, причому цілком не важливо, як саме реалізований сам таймер. Модель такої системи зручно подати у вигляді спрямованого графа станів (state charts). Стани відображаються овалами, які містять значення набору змінних, а переходи – спрямованими дугами (рис. 4.11). Діаграми станів дуже ефективний інструмент проектування й аналізу автоматів.

Технічна база для побудови автоматів досить широка. Це механічні вузли, пневматичні елементи, реле або логічні мікросхеми тощо. Але на відміну від будь-яких інших реалізацій автоматів технологія ПЛК забезпечує швидке й виключно гнучке рішення. Безумовно, в ході побудови автоматів на

базі програмованих логічних матриць і мікропроцесорів перепрограмування також можливо, але значно більш трудомістке. Це можна зробити тільки за наявності відповідного обладнання і спеціальної підготовки.

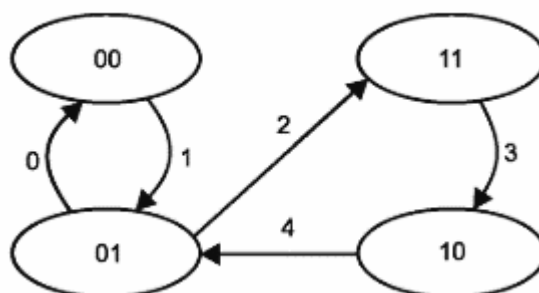


Рисунок 4.11 – Граф станів для двох змінних

Реально можливості ПЛК суттєво перевищують кінцеві автомати. Далеко не все, що можна зробити на ПЛК, вписується в рамки кінцевих автоматів. Це функції управління за часом, математична обробка даних, регулювання тощо. Проте застосування формалізму кінцевих автоматів дозволяє значно спростити процес проектування.

Причому це стосується не тільки ПЛК. Подібний підхід лежить в основі універсальної мови моделювання Unified Modeling Language (UML). Пакет розширення Stateflow матричної системи комп'ютерної математики MATLAB забезпечує побудова анімаційних діаграм стану моделей різних пристроїв і систем. Він дозволяє виконувати ситуаційне моделювання додатково до імітаційного моделювання, що виконується потужним пакетом розширення системи MATLAB-Simulink.

4.3 Структурований текст (ST)

4.3.1 Особливості мови ST

Мова ST (Structured Text) – це мова високого рівня. Синтаксично ST – це дещо адаптована мова Паскаль. Замість процедур Паскаля в ST використовуються компоненти програм стандарту МЕК.

Для фахівців, знайомих з мовою С, освоєння ST також не буде занадто складним.

У більшості комплексів програмування ПЛК мова ST за замовчуванням пропонується для опису дій і умов переходів SFC. Це дійсно максимально потужний тандем, що дозволяє ефективно вирішувати будь-які завдання. Розглянемо основні елементи мови ST.

4.3.2 Вирази

Основою ST-програми є вирази. Результат обчислення виразу присвоюється змінній за допомогою оператора «: =», як і в Паскалі. Кожен вираз обов'язково закінчується крапкою з комою «;». Вираз складається із змінних констант і функцій, розділених операторами:

```
iVar1 := 1 + iVar 2 / ABS(iVar2);
```

Стандартні оператори у виразах ST мають символічне подання, наприклад математичні дії: +, − *, /, операції порівняння тощо.

Крім операторів, елементи виразу можна відокремлювати пробілами і табуляціями для кращого сприйняття. У текст можуть бути введені коментарі. Скрізь, де допустимі пасивні роздільники, можна вставляти і коментарі:

```
iVar1 := 1 + (*отримати знак*) iVar2 / ABS(iVar2); (*перевірка на 0 була вище*)
```

Декілька виразів можна записати поспіль в один рядок. Проте гарним стилем вважається запис одного виразу в рядку. Довгі вирази можна перенести на наступний рядок. Перенесення рядка рівноцінно пасивному роздільнику.

Вираз може включати інший вираз, який записують у дужках. Вираз, який записаний у дужках, обчислюється насамперед. Тип виразу визначається типом результату обчислень:

```
bAlarm := byInp1 > byInp2 AND byInp1 + byInp2 <> 0 OR bAlarm2;
```

4.3.3 Порядок обчислення виразів

Обчислення виразу відбувається відповідно до правил пріоритету операцій. Першими виконуються операції з найвищим пріоритетом.

У порядку зменшення пріоритету операції розташовуються так: вираз в дужках; виклик функції; ступінь EXP; заміна знаку (-); заперечення NOT; множення, ділення й ділення за модулем MOD; додавання і віднімання (+, -); операції порівняння (<, >, <=, >=); рівність (=); нерівність (<>); логічні операції логічні операції AND, XOR і OR. Пріоритет операцій у виразах дуже важливий. Насамперед з математичної точки зору:

$X := 2 + 2 * 2; \quad (* = 6*)$

$X := (2 + 2) * 2; \quad (* = 8*)$

Тут результати очевидні. Множення має більш високий пріоритет, ніж складання, і виконується раніше. Дужки змінюють порядок обчислень, і результат виявляється іншим.

У ході складання виразів обов'язково необхідно враховувати можливий діапазон зміни значень і типи змінних. Помилки, пов'язані з переповненням, виникають у процесі виконання і не можуть бути виявлені транслятором.

iVar1: SINT;

siVar2: SINT := 120;

siVar1 := 120 - siVar2 + 20; (*120 - 120 = 0, 0 + 20 = 20*)

siVar1 := 120 - (siVar2 + 20); (*120 + 20 = -116, 120 +116 = -20*)

4.3.4 Порожній вираз

Порожній вираз складається з крапки з комою «;». Для крапки з комою транслятор не генерує жодного коду. Якщо випадково поставити зайву «;», це не викличе помилки. Єдине осмислене застосування порожнього виразу – це забезпечення правильності мовних конструкцій. Наприклад, може знадобитися відтранслювати проект, що містить ще не реалізований ROU. Для коректної трансляції достатньо написати в тілі ROU один порожній оператор. Ще один приклад, де порожній оператор виявляється до речі, – це умова IF, що не містить розділ THEN:


```

IF x = Threshold THEN
    ;                                (*так правильно*)
ELSIF x > Threshold THEN
    bMarker := bMarker - 1;        (*крок униз*)
ELSE
    bMarker := bMarker + 1;        (*крок вгору*)
END_IF;                             (*зайва ; *)

```

4.3.5 Оператор вибору IF

Оператор вибору дозволяє виконати різні групи виразів залежно від умов, виражених логічними виразами. Повний синтаксис оператора IF (ЯКЩО) виглядає так:

```

IF <логічний вираз IF>
THEN
<вираз IF> ;
[
ELSIF <логічний вираз ELSEIF 1>
THEN
    <вираз ELSEIF 1> ;
    ...
ELSIF <логічний вираз ELSEIF n>
THEN
    <вираз ELSEIF n> ;
ELSE
    <вираз ELSE> ;
]
END_IF

```

Якщо <логічний вираз IF> ІСТИНА, то виконуються вирази першої групи – <вираз IF>. Інші вирази пропускаються, альтернативні умови не перевіряються.

Частина конструкції в квадратних дужках є необов'язковою і може бути відсутня.

Якщо <логічний вираз IF> ХИБНІСТЬ, то одне за одним перевіряються умови ELSIF. Перша істинна умова призведе до виконання відповідної групи виразів. Інші умови ELSIF не аналізуватимуться. Група ELSIF може бути декілька або не бути зовсім.

Якщо всі логічні вирази дали помилковий результат, то виконуються вирази групи ELSE, якщо вона є. Якщо групи ELSE немає, то не виконується нічого.

У найпростішому випадку оператор IF містить тільки одну умову. Наприклад:

```
IF bReset THEN
    iVar1 := 1;
    iVar2 := 0;
END_IF
```

На перший погляд конструкція IF з декількома групами ELSIF виглядає складною, але насправді виявляється досить виразною, наприклад:

```
IF bReset THEN
    iVar1 := 1;
    ELSIF byLeft < 16 THEN
        iVar1 := 2;
    ELSIF byLeft < 32 THEN
        iVar1 := 3;
    ELSIF byLeft < 64 THEN
        iVar1 := 4;
ELSE
    bReset := TRUE;
END IF
```

4.3.6 Оператор множинного вибору CASE

Оператор множинного вибору CASE дозволяє виконати різні групи виразів залежно від значення однієї цілочисельної змінної або виразу. Синтаксис:

```
CASE < цілочисельний вираз> OF
    <значення 1>:
        <вираз 1> ;
    <значення 2> , <значення 3> :
        <вираз 3> ;
    <значення 4>..<значення 5> :
        <вираз 4> ;
    ...
[
ELSE
    <вираз ELSE> ;
]
END CASE
```

Якщо значення виразу збігається із заданою константою, то виконується відповідна група виразів. Інші умови не аналізуються (<значення 1>: <вираз 1>;).

Якщо кілька значень констант мають відповідати одній групі виразів, їх можна перерахувати через кому (<значення 2>, <значення 3>: <вираз 3> ;).

Діапазон значень можна визначити через двокрапку (<значення 4> .. <значення 5>: <вираз 4> ;).

Група виразів ELSE є необов'язковою. Вона виконується під час незбігу жодної з умов (<вираз ELSE> ;). Наприклад:

```
CASE byLeft/2 OF
0,127:
    bReset := TRUE;
    Var1 := 0;
```

```
16..24:
    Var1 := 1;
ELSE
    Var1 := 2;
END_CASE
```

Значеннями вибору CASE можуть бути тільки цілі константи, змінні використовувати не можна.

Однакові значення в альтернативах вибору задавати не можна, навіть у діапазонах.

Так, наступний приклад викликає помилку в ході трансляції:

```
CASE byLeft OF
20:      Var1 := 0;
16..24:  Var1 := 1;
END_CASE
```

Безумовно, оператор CASE «слабкіше» оператора IF, який не має подібних обмежень. Але формат CASE не тільки виразніший для програміста, але й більш ефективний. Використання цілочисельних констант дозволяє транслятору виконати оптимізацію коду, часто досить суттєву.

4.3.7 Цикли WHILE і REPEAT

Цикли WHILE і REPEAT забезпечують повторення групи виразів, поки вірний умовний логічний вираз. Якщо умовний вираз завжди істинний, то цикл стає безкінечним. Синтаксис WHILE:

```
WHILE <Умовний логічний вираз> DO
    <Вираз – тіло циклу>
END WHILE
```

Умова в циклі WHILE перевіряється до початку циклу. Якщо логічний вираз спочатку має значення ХИБНІСТЬ, тіло циклу не буде виконано жодного разу.

Синтаксис REPEAT:

```
REPEAT
    <Вираз – тіло циклу>
UNTIL <Умовний логічний вираз>
END_REPEAT
```

Умова в циклі REPEAT перевіряється після виконання тіла циклу. Якщо логічний вираз спочатку має значення ХИБНІСТЬ, тіло циклу буде виконано один раз, наприклад:

```
ci := 64;
WHILE ci > 1 DO
    Var1 := Var1 + 1;
    ci := ci/2;
END_WHILE
```

Правильно побудований цикл WHILE або REPEAT обов'язково має змінювати змінні, які складають умову закінчення в тілі циклу, поступово наближаючись до умови завершення. Якщо цього не зробити, цикл не закінчиться ніколи.

Потрібно намагатися не використовувати точні рівність або нерівність для припинення циклу. Інакше є ймовірність помилково проскочити граничну умову. Краще використовувати умови більше й менше. У наступному прикладі помилка добре видна виключно завдяки його простоті:

```
ci := 1;
WHILE ci <> 100 DO
    Var1 := Var1 + 1;
    ci := ci + 10;
```

END_WHILE

Очевидно, лічильник ci за початкового значення 1 і прирощення 10 ніколи не стане рівним 100.

Для реалізації мінімального часу виконання циклу необхідно уникати в тілі циклу і в умовному виразі обчислень, які можна було зробити заздалегідь. Такі обчислення повторюються в циклі, щоразу забираючи час. Наприклад:

```
WHILE ci < 5 + x DO
    Var := Var1 + 2*x*x + 1;
    ci := ci + 1;
END_WHILE
```

Даний цикл можна оптимізувати за швидкістю:

```
iMax := 5+x;
iPoly := 2*x*x + 1;
WHILE ci < iMax DO
    Var := Var1 + iPoly;
    ci := ci + 1;
END_WHILE
```

4.3.8 Цикл FOR

Цикл FOR забезпечує задану кількість повторень групи виразів.
Синтаксис:

```
FOR <Цілий лічильник> := <Початкове значення>
TO <Кінцеве значення>
[BY <Крок>] DO
    <Вираз – тіло циклу>
END_FOR
```

Перед виконанням циклу лічильник отримує початкове значення. Далі тіло циклу повторюється, поки значення лічильника не перевищить кінцевого значення. Лічильник збільшується в кожному циклі. Початкове й кінцеве значення і крок можуть бути як константами, так і виразами.

Лічильник змінюється після виконання тіла циклу. Тому якщо задати кінцеве значення менше початкового, то за позитивного приросту цикл не буде виконаний жодного разу. За однакових початковому й кінцевому значеннях тіло циклу буде виконано один раз.

Частина конструкції ВУ у дужках не обов'язкова, вона визначає крок збільшення лічильника. За замовчуванням лічильник збільшується на одиницю в кожній ітерації.

Як лічильник можна використовувати змінну будь-якого цілого типу.
Приклад:

```
Var1 := 0;  
FOR cw := 1 TO 10 DO  
    Var1 := Var1 + 1;  
END_FOR
```

Даний цикл буде виконано 10 разів і відповідно Var1 матиме значення 10.

Крок зміни лічильника ітерацій може бути і негативним. Початкова умова в цьому випадку має бути більше кінцевої. Цикл буде закінчений, коли значення лічильника стане менше кінцевого значення. Наприклад:

```
Var1 := 0;  
FOR ci := 10 TO 1 BY -1 DO  
    Var1 := Var1 + 1;  
END_FOR
```

Цикл FOR зручний для ітерацій із заздалегідь відомою кількістю повторів. Причому, щоб створити безкінечний цикл FOR, потрібно добре

постаратися. Можна, наприклад, спробувати поставити нульовий крок збільшення (в CODESYS це не допомагає) або скинути лічильник в тілі циклу.

Для побудови правильного циклу достатньо дотримуватися двох простих формальних вимог:

- не змінюйте лічильник циклу і умову закінчення в тілі циклу. Лічильник та змінні, які утворюють кінцеву умову в циклі, можна використовувати тільки для читання;

- не ставте як кінцеву умову максимальне для типу змінної лічильника значення. Так, якщо для однобайтного цілого без знака задати константу 255, то умова закінчення ніколи не буде виконана. Цикл стане безкінечним.

У CODESYS лічильник змінюється в тілі циклу завжди, включаючи завершальну ітерацію, коли умова закінчення вже досягнута. Але в стандарті такі тонкощі не обумовлені. З метою оптимізації за швидкістю транслятор може обійти змінення лічильника зайвий раз. Тому не рекомендується використовувати значення лічильника поза тіла циклу. Переносність такої програми гарантувати не можна. Уникайте використовувати цикл FOR зі складними умовами закінчення і у випадках, коли після закінчення циклу необхідно визначити причину закінчення. Наприклад, коли цикл може бути перерваний оператором EXIT і є необхідність дізнатися, скільки ітерацій насправді було виконано (див. нижче приклад з EXIT).

Інший момент, який необхідно враховувати під час створення переносної програми, – це негативне прирощення лічильника. На жаль, далеко не всі системи програмування підтримують таку можливість.

4.3.9 Переривання ітерацій операторами EXIT і RETURN

Оператор EXIT, поміщений у тілі циклів WHILE, REPEAT і FOR, призводить до негайного закінчення циклу. Гарний стиль програмування закликає уникати такого прийому, але іноді він дуже зручний. Розглянемо, наприклад, пошук елемента масиву з певним значенням (x). Найпростіше організувати лінійний перебір за допомогою циклу FOR:

```
bObtained:= FALSE;  
FOR cN := 1 TO MaxIndex DO
```



```

    IF x = aX[cN] THEN
        Index := cN;
        bObtained := TRUE;
        EXIT;
    END_IF
END_FOR
IF bObtained THEN      (*елемент знайдено, його індекс – Index*)

```

Для вкладеного циклу оператор EXIT завершує тільки «свій» цикл, зовнішній цикл продовжуватиме роботу. Наприклад:

```

FOR y := 0 TO 9 DO
    FOR x := 0 TO 99 DO (*обробляємо рядок масиву Arr[y][x]*)
        ;
        IF ... THEN EXIT;
    END_FOR
ENDFOR

```

За необхідності завершення зовнішнього циклу за умови, яка з'явилася у вкладеному циклі, можна використовувати пару синхронізованих операторів EXIT:

```

bBreakY := FALSE;
FOR y := 0 TO 9 DO
    FOR x := 0 TO 99 DO
        ;
        IF ... THEN bBreakY := TRUE; EXIT;
        (*перервати обробку*)
    END_FOR
    IF bBreakY THEN EXIT;
END_FOR

```

Оператор RETURN здійснює негайне повернення з ROU. Це єдиний спосіб перервати вкладені ітерації без введення додаткових перевірок умов. Оператор RETURN виконується дуже швидко, фактично це одна машинна команда процесора. Але не варто ним зловживати. Оскільки в тексті компонента, що має, наприклад, 50 виходів, розібратися досить непросто.

Іноді буває зручно створити безумовний цикл, а умови виходу формувати в тілі циклу з використанням EXIT. Наприклад, можуть знадобитися кілька рівноймовірних, але невзаємопов'язаних умов виходу з циклу. Створити безумовний (безкінечний) цикл у ST найпростіше так: WHILE TRUE DO...

4.3.10 Ітерації на базі робочого циклу ПЛК

Використовувати для умови виходу з циклів WHILE і REPEAT входи, виходи або інші апаратно залежні змінні ПЛК не можна. Дані змінні не змінюють своїх значень у межах одного робочого циклу користувальницької програми, тому цикл завжди буде безкінечним. Якщо подібна необхідність все ж існує, використовуйте для ітерацій робочий цикл ПЛК. Виконання ітерації задається простою умовою IF.

Аналогічно можна робити при необхідності побудови тривалих циклів. Наприклад, ініціалізація або копіювання великих масивів даних. Розподілення об'ємних операцій на декілька робочих циклів є стандартним прийомом, що дозволяє уникнути небажаного уповільнення інших, паралельно виконуваних завдань.

Так, ініціалізація даних функціонального блоку з виставленням сигналу готовності може виглядати так:

```
ENO:      BOOL := FALSE;
niCounter: INT := 1000;
aiVar:    ARRAY[0..999] OF INT;

IF niCounter = 0 THEN
    ENO := TRUE;
    ... (*основна робота блоку*)
```

```

ELSE
    niCounter := niCounter - 1;
    aiVar[niCounter] := GetlnitVal(niCounter);
END IF

```

4.3.11 Оформлення тексту

Оформлення текстів ST-програм може бути абсолютно довільним. Розташування операторів і виразів в рядку не впливає на правильність програм. Але дуже важливо виробити свій власний стиль і чітко дотримуватися його.

Найважливішу роль в оформленні відіграють відступи на початку рядків. Відступи візуально об'єднують рядки, що містять вирази одного рівня вкладення. Текст, вирівняний у вигляді драбинки, кожна сходинка якої належить до одного циклу або умови, читається легко. Незважаючи на можливість горизонтальної прокрутки в редакторі, бажано, щоб за шириною текст містився на одній сторінці. Не варто розташовувати кілька виразів в один рядок. Нічого страшного немає в тому, що текст виявиться розтягнутим по вертикалі: лаконічні вирази і навіть порожні рядки тільки допомагають зоровому аналізу. Наприклад:

```

FOR icY := 0 TO 8 DO
    FOR icX := 0 TO 16 DO
        IF iaPos[icY,icX] > iLevel THEN
            iBalance := iBalance + 1;
        ELSE
            IF iaPos[icY,icX] < iLevel THEN
                iBalance := iBalance - 1;
            END_IF
        END_IF
    END_FOR
    iLevel := iLevel *2;
END_FOR

```

Погано оформлений ST-текст читати вкрай важко, навіть редактор з колірним виділенням інструкцій тут не допоможе. Мало того, помилки у схемі відступів здатні заплутати розуміння сенсу:

```
FOR icY := 0 TO 8 DO
  FOR icX := 0 TO 16 DO
    IF iaPos[icY,icX] > iLevel THEN
      iBalance := iBalance + 1;
    ELSE
      IF iaPos[icY,icX] < iLevel THEN
        iBalance := iBalance - 1;
      END_IF
    END_IF
  END_FOR
iLevel := iLevel *2;
END_FOR
```

Для оформлення ST текстів цілком можна застосувати рекомендації, які зустрічаються в літературі з програмування на Паскалі і С. Зверніть увагу, що в ST відсутні програмні дужки (в Паскалі: begin, end; в С: {}). Замість них кожен вираз мови має власну кінцівку (WHILE .. END_WHILE, IF .. END_IF). Тобто, програмна дужка, що закриває, є інформативною.

Візуально такий текст сприймається явно краще. Зі створенням складних вкладень у мові С дужки, що закривають, часто розташовані суцільною драбинкою. У таких випадках досвідчені програмісти застосовують стислі коментарі після кожної дужки, що закриває. Коментарі підказують, з чого розпочато даний рівень відступу.

Наприклад: (*FOR x*). Це гарний прийом, але при грамотному застосуванні відступів у рядках ST така необхідність виникає значно рідше, ніж у С і Паскалі.

4.4 Релейні діаграми (LD)

4.4.1 Загальні відомості про мову LD, поняття ланцюгів

Мова релейних діаграм LD (Ladder Diagram) або релейно-контактних схем (РКС) – графічна мова, реалізує структури електричних ланцюгів. На початку 70-х рр. ХХ ст. релейні автомати складальних конвеєрів почали поступово витіснятися програмованими контролерами. Деякий час ті й інші працювали водночас й обслуговувалися одними й тими самими людьми. Так з'явилася задача прозорого перенесення релейних схем у ПЛК. Різні варіанти програмної реалізації релейних схем створювалися практично всіма провідними виробниками ПЛК. Завдяки простоті подання РКС знайшов заслужену популярність, що й стало основною причиною включення його в стандарт МЕК.

Слова «релейна логіка» звучать сьогодні досить архаїчно, майже як «ламповий комп'ютер». Тим більше у зв'язку зі створенням численних швидкодіючих і надійних безконтактних (зокрема, оптоелектронних) реле і потужних перемикальних приладів, таких як потужні польові транзистори, керовані тиристори й прилади IGBT [–36]. Але, незважаючи на це, релейна техніка все ще дуже широко застосовується.

Релейна схема являє собою дві вертикальні шини живлення, між ними розташовані горизонтальні ланцюги, утворені контактами та обмотками реле. Кількість контактів у ланцюзі довільна, реле одне. Кожне реле має контакти, які можна використовувати в інших ланцюгах. Якщо послідовно з'єднані контакти замкнуті, струм йде ланцюгом і реле вмикається (у прикладі на рис. 4.12 Lamp1). За необхідності можна підключити паралельно декілька реле, послідовне підключення не допускається.

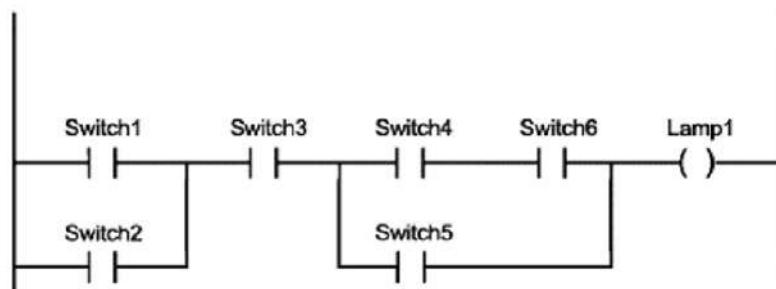


Рисунок 4.12 – Схема LD з одним ланцюгом

Логічно послідовне (І), паралельне (АБО) з'єднання контактів та інверсія (НІ) утворюють базис Буля. В результаті LD ідеально підходить не тільки для побудови релейних автоматів, а й для програмної реалізації комбінаційних логічних схем. Завдяки можливості підключення в LD функцій і функціональних блоків, виконаних іншими мовами, сфера застосування мови практично не обмежена.

У LD кожному контакту ставиться у відповідність логічна змінна, яка визначає його стан. Якщо контакт замкнутий, то змінна має значення ІСТИНА. Якщо розімкнутий – ХИБНІСТЬ. Ім'я змінної пишеться над контактом і фактично є його назвою.

Послідовне з'єднання контактів або ланцюгів рівноцінно логічній операції І. Паралельне з'єднання утворює монтажне АБО.

Ланцюг може бути або замкнутим (ON), або розімкнутим (OFF). Це саме й відображається на обмотці реле і відповідно на значенні логічної змінної обмотки (ІСТИНА / ХИБНІСТЬ).

Наведена на рис. 4.12 схема еквівалентна виразу:

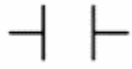
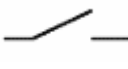


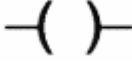
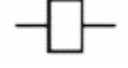
$$\text{Lamp1} := (\text{Switch1 OR Switch2}) \text{ AND Switch3 AND } ((\text{Switch4 AND Switch6}) \text{ OR Switch5});$$

Зорове сприйняття LD-діаграм має бути інтуїтивно зрозумілим. Вітчизняним розробникам цього певною мірою перешкоджає прийнята система умовних графічних позначень, що базується на американському стандарті NEMA. Перевага таких позначень полягає в можливості застосування символів псевдографіки для побудови LD-діаграм.

Зіставлення позначень базових елементів LD і позначень ЄСКД приведено в таблиці 4.11.

Контакт може бути інверсним – нормально замкнутим. Такий контакт позначається за допомогою символу $|/|$ і замикається, якщо значення змінної ХИБНІСТЬ. Інверсний контакт рівнозначний логічній операції НІ.

Таблиця 4.11 – Зіставлення позначень базових елементів LD і позначень ЄСКД

LD	ЄСКД	Позначення
		Нормально розімкнутий контакт
		Нормально замкнутий контакт
		Обмотка реле

Перемикаючий контакт утворюється комбінацією прямого й інверсного контактів (див. приклад на рис. 4.13).

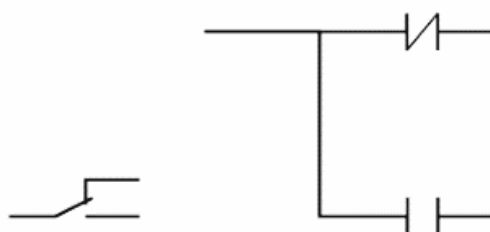


Рисунок 4.13 – Перемикаючий контакт

Обмотки реле також можуть бути інверсними, що позначається символом (/). Якщо обмотка інверсна, то у відповідну логічну змінну копіюється інверсне значення стану ланцюга.

4.4.2 Реле з самофіксацією

Крім звичайних реле, в релейних схемах часто застосовуються поляризовані реле. Таке реле має дві обмотки, які перемикають його з одного положення в інше. Перемикання проводиться імпульсами струму. З відключенням струму живлення поляризоване реле залишається в заданому положенні, що реалізує елементарну комірку пам'яті.

У LD таке реле реалізується за допомогою двох спеціальних обмоток SET і RESET. Обмотки типу SET позначаються літерою S всередині круглих дужок (S). Обмотки типу RESET позначаються літерою R. Якщо відповідна обмотці (S) змінна приймає значення ІСТИНА, то зберігає його безкінечно. Повернути дану змінну в ХИБНІСТЬ можна тільки обмоткою (R).

Очевидно, що повної аналогії з поляризованим реле програмно досягти неможливо. Навіть якщо значення логічного виходу зберігається в енергонезалежній пам'яті, стан самого електричного ланцюга при вимкненому живленні ПЛК визначається його схематикою. Фіксація безпечного положення апаратури під час аварії живлення системи управління може бути досягнута тільки апаратно.

Умова вимкнення реле не завжди рівнозначна відсутності умови увімкнення. Завдяки (R) і (S) обмоткам умови увімкнення і вимкнення реле можна формувати абсолютно незалежно, причому в будь-якому ланцюзі і скільки завгодно разів. Обмотки (R) і (S) забезпечують фіксацію умов управління, що необхідно в ході реалізації автоматів з пам'яттю.

Самофіксацію нескладно організувати і на простому реле, використовуючи додатковий контакт у ланцюзі живлення. Приклад цього наведено на рис. 4.15.

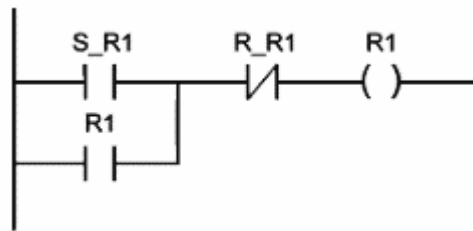


Рисунок 4.15 – Реле з самофіксацією

Контакт S_R1 вмикає, а R_R1 вимикає реле R1. Завдяки контакту R1 реле отримує живлення після розмикання S_R1. Застосування SET / RESET-обмоток не дає нічого принципово нового, але робить LD-діаграму простіше й красивіше.

4.4.3 Порядок виконання і зворотні зв'язки

Ідеологія релейних схем передбачає паралельну роботу всіх ланцюгів. Струм в усі ланцюги подається водночас.

У LD рішення діаграми виконується послідовно зліва направо і зверху вниз. У кожному робочому циклі одноразово виконуються всі ланцюги діаграми, що і створює ефект паралельності роботи ланцюгів. Будь-яка змінна

в рамках одного ланцюга завжди має одне й те саме значення. Якщо навіть реле в ланцюзі змінить змінну, то нове значення надійде на контакти тільки в наступному циклі. Ланцюги розташовані нижче, отримують нове значення змінної відразу. Ланцюги розташовані вище – тільки в наступному циклі. Суворий порядок виконання схеми дуже важливий. Випадковий або навіть істинно паралельний порядок виконання ланцюгів міг би приводити до ефекту «гонок», який зустрічається в електронних схемах з тригерами. Завдяки жорсткому порядку виконання LD-діаграми зберігають стійкість за наявності зворотних зв'язків.

У наведеній на рис. 4.16 схемі вмикання Key викличе миттєве (в тому ж циклі) вмикання P2 і вимикання P3. Реле P1 буде увімкнене тільки в наступному циклі, причому навіть якщо Key вже в обриві (ХИБНІСТЬ).

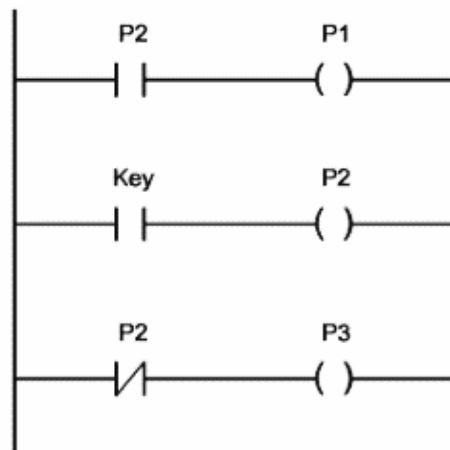


Рисунок 4.16 – LD-діаграма зі зворотним зв'язком

Використовуючи вищеописаний принцип циклічності виконання LD-діаграм, дуже легко побудувати генератор одиничних імпульсів. Приклад такої побудови показаний на рис. 4.17.



Рисунок 4.17 – Генератор одиничних імпульсів

Період імпульсів на реле P1 дорівнюватиме подвоєній тривалості робочого циклу ПЛК.

4.4.4 Управління порядком виконання

Порядок виконання ланцюгів діаграми можна примусово змінювати, використовуючи мітки (labels) і переходи (jumps).

Мітку можна ставити тільки в початок ланцюга. Імена міток підпорядковані правилам найменування змінних. Для наочності можна закінчити мітку двокрапкою. Двокрапка не утворює нової мітки. Таким чином, M1: і M1 – це одне й те саме.

Ланцюг може мати тільки одну мітку і один перехід. Перехід рівнозначний вихідному реле і виконується, якщо вихідна змінна має значення ІСТИНА. Перехід може бути інверсним, у цьому випадку він виконується зі значенням ланцюга ХИБНІСТЬ. Використовуючи перехід, можна пропустити виконання частини діаграми. Пропущені ланцюги не скидаються, а саме не виконуються – залишаються в тому положенні, в якому були раніше. Перехід вгору допускається і дозволяє створювати цикли (рис. 4.18).

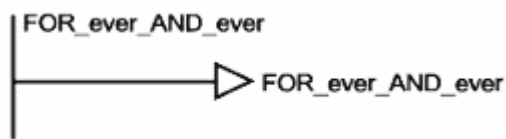


Рисунок 4.18 – Простий безкінечний цикл

Ідеологічно переходи суперечать аналогії LD з релейними схемами, порушуючи закони електричних ланцюгів. У схемі LD з переходами розібратися буває складно. Бажано не займатися управлінням порядком виконання LD-діаграми в ній самій, а використовувати для цього більш виразні засоби. Наприклад, розділити LD-діаграму на модулі (дії), а порядок виконання описати в SFC.

4.4.5 Розширення можливостей LD

У LD-діаграму можна вставити функції та функціональні блоки. Функціональні блоки повинні мати логічні вхід і вихід. На рис. 4.19 показаний приклад організації циклу на 10 повторів на базі функціонального блоку декрементний лічильник. Перший ланцюг завантажує лічильник кількістю повторів. Другий ланцюг – генератор одиничних імпульсів. Третій –

декрементний лічильник з перевіркою умови закінчення циклу. Тіло циклу на рисунку не наведено.

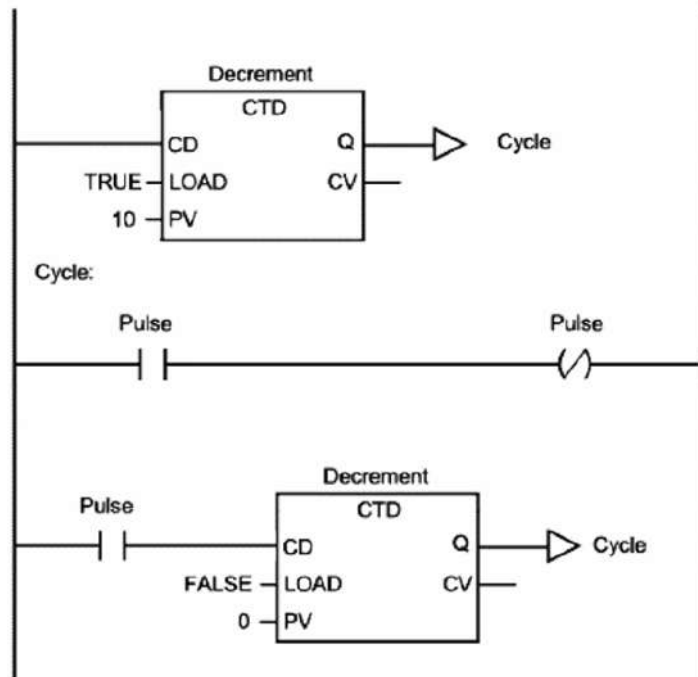


Рисунок 4.19 – Цикл на 10 повторів на базі функціонального блоку CTD

Для включення в діаграму функцій у них штучно вводиться додатковий логічний вхід, що позначається EN (Enable) (рис. 4.20). Логічне значення на вхід EN дозволяє або забороняє виконання функції. Сама функція не змінюється з додаванням входу EN.

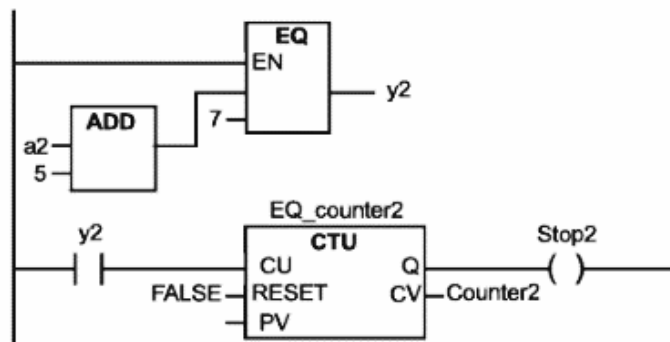


Рисунок 4.20 – Використання оператора EQ, який управляється входом EN

У першій редакції стандарт МЕК визначав контакти й обмотки, які управляються фронтами імпульсів: контакт |P| і обмотка (P) переднього фронту, контакт |N| і обмотка (N) заднього фронту. Зараз підтримка таких

контактів і обмоток не є обов'язковою, тому що аналогічний ланцюг легко можна побудувати за допомогою функціональних блоків R_TRIG і F_TRIG.

4.4.6 Особливості реалізації LD в CODESYS

Неважко помітити, що довжина LD-ланцюгів різна залежно від складності. Наприклад, нехай перший ланцюг складається з 50 контактів, а другий з двох. У цьому випадку шини живлення на схемі мають бути широко рознесені, щоб вмістити всі з'єднання першого ланцюга. Другий ланцюг виглядатиме невиправдано розтягнутим. Ця особливість РКС завжди викликала труднощі у реалізації станцій ПЛК, що програмують. Якщо ланцюг не вміщувався в один екран, її розривали і переносили залишок нижче. Перенесення легко читається для одного проводу, але ланцюг може бути розгалуженим.

Стандарт МЕК допускає не зображувати загальну праву шину взагалі і вирівнювати ланцюг вліво для кращого зорового сприйняття. Графічний редактор CODESYS не обмежує можливу ширину LD-ланцюга і не вимагає застосування переносів. Складні ланцюги зображуються зливо і для роботи з ними необхідно користуватися горизонтальною прокруткою екрана (рис. 4.21). Права шина зображується вертикальними відрізками в межах кожного ланцюга і вирівнюється вліво, ланцюги пронумеровані й розділені горизонтальними лініями.

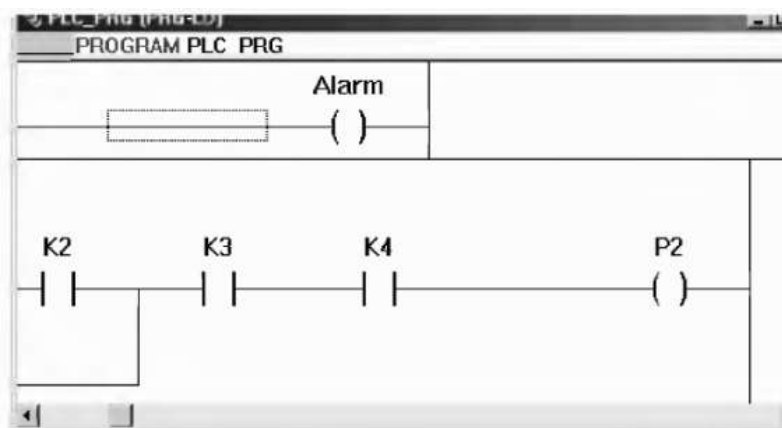


Рисунок 4.21 – Правий край двох ланцюгів різної довжини

Час виконання одного LD-ланцюга не постійний. Якщо процесор виявив, що ланцюг розімкнутий, то вже немає сенсу аналізувати його далі, можна відразу присвоїти значення FALSE вихідній змінній. Компілятор коду CODESYS так і робить.

Для функціональних блоків CODESYS дозволяє графічно підключати тільки один вхід у логічний ланцюг. Стандарт не накладає таких обмежень. Наприклад, у системі MULTIPROG перший ланцюг, показаний на рис. 4.19, може виглядати, як на рис. 4.22.

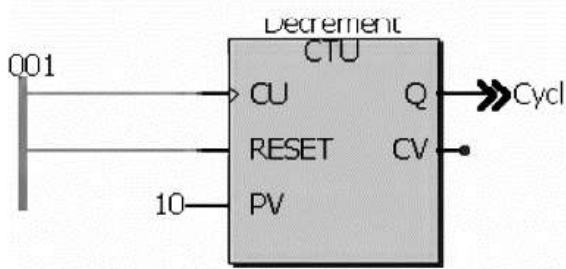


Рисунок 4.22 – Зображення декрементного лічильника в MULTIPROG

Вхід RESET тут отримує значення ІСТИНА безпосередньо від шини живлення. Така схема більш наочна, але прийнятна тільки для логічних змінних. Аналогові змінні все одно не вписуються в ідеологію LD і вимагають безпосереднього присвоювання.

Функція або функціональний блок у LD розглядаються як виконавчий пристрій – аналог реле. Застосування функції фактично означає кінець LD-діаграми і початок FBD. Правила побудови LD-діаграм тут вже не працюють.

Крім входу EN, стандарт пропонує визначати для функцій і додатковий вихід ENO (Enable Out), що показує подальше проходження струму в ланцюзі. Вихід ENO не є обов'язковим. У CODESYS така можливість не підтримується. Вихід ENO має слугувати для індикації помилок у функції. В CODESYS контроль помилок виконання реалізований інакше.

За визначенням функція має тільки один вихід. Завдяки цьому функції можна використовувати у виразах ST. Функції з додатковим виходом ENO вирішують одну проблему, але створюють іншу.

4.4.7 LD- діаграми в режимі виконання

У режимі Online обмотки реле, контакти і провідники, що знаходяться в стані On (під струмом) блакитного (кольори за замовчуванням) кольору. CODESYS дозволяє змінювати значення логічних змінних (ІСТИНА / ХИБНІСТЬ) безпосередньо в графічній діаграмі подвійним клацанням мишки на імені змінної. Значення входів-виходів функціональних блоків відображаються числовими значеннями.

Точка зупинника може встановлюватися тільки в цілому на ланцюг. Для установки або скидання точки зупинника необхідно клацнути кнопкою мишки по номеру ланцюга. У режимі зупинника номер ланцюга підсвічений червоним. Покрокове – по одному ланцюгу виконання досягається командами «Step over» і «Step in».

4.5 Функціональні діаграми FBD

4.5.1 Відображення ROU

FBD (Function Block Diagram) – це графічна мова програмування. Діаграма FBD дуже нагадує принципову схему електронного пристрою на мікросхемах. На відміну від LD, «провідники» у FBD можуть проводити сигнали (передавати змінні) будь-якого типу (логічний, аналоговий, час тощо). Іноді говорять, що в релейних схемах з'єднувальні провідники передають енергію. Провідники FBD теж передають енергію, але в більш широкому сенсі. Очевидно, що шини живлення і контакти тут вже не ефективні. Шини живлення на FBD-діаграмі не відображаються. Виходи блоків можуть бути подані на входи інших блоків або безпосередньо на виходи ПЛК. Самі блоки, подані на схемі як «чорні ящики», можуть виконувати будь-які функції.

FBD-схеми дуже чітко відображають взаємозв'язок входів і виходів діаграми. Якщо алгоритм від початку добре описується з позиції сигналів, то його FBD-подання завжди виходить наочніше, ніж текстовими мовами.

Діаграма FBD будується з компонентів, що відображаються на схемі прямокутниками. Входи ROU зображуються зліва від прямокутника, виходи справа. У середині прямокутника вказується тип ROU і найменування входів і виходів. Для екземпляра функціонального блоку його найменування

вказується у верхній частині, над прямокутником. У графічних системах програмування прямокутник компонента може містити зображення, що відображає його тип. Розмір прямокутника залежить від кількості входів і виходів і встановлюється графічним редактором автоматично. Приклад графічного подання екземпляра Blinker функціонального блоку BLINK дано на рис. 4.23.

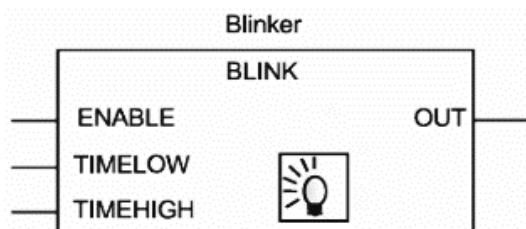


Рисунок 4.23 – Графічне подання екземпляра функціонального блоку

Програма у FBD не обов'язково має представляти велику єдину схему. Як і в LD, діаграма утворюється з ланцюгів, які виконуються одна за одною. У CODESYS всі ланцюги одного POU відображаються в єдиному графічному вікні, пронумеровані і розділені горизонтальними лініями (рис. 4.24). Значення змінних, які обчислені в одному ланцюзі, доступні в наступних ланцюгах відразу в тому самому робочому циклі.

4.5.2 З'єднувальні лінії

Прямокутники POU у FBD з'єднані лініями зв'язку. З'єднання мають спрямованість зліва направо. Вхід блоку може бути з'єднаний з виходом блоку, розташованого зліва від нього. Крім цього, вхід може бути з'єднаний зі змінною або константою. З'єднання має пов'язувати змінні або входи й виходи одного типу. На відміну від компонента, змінна зображується на діаграмі без прямокутної рамки. Ширина сполучної лінії в FBD не важлива. Стандарт допускає використання з'єднувальних ліній різної ширини і стилю для з'єднань різного типу.

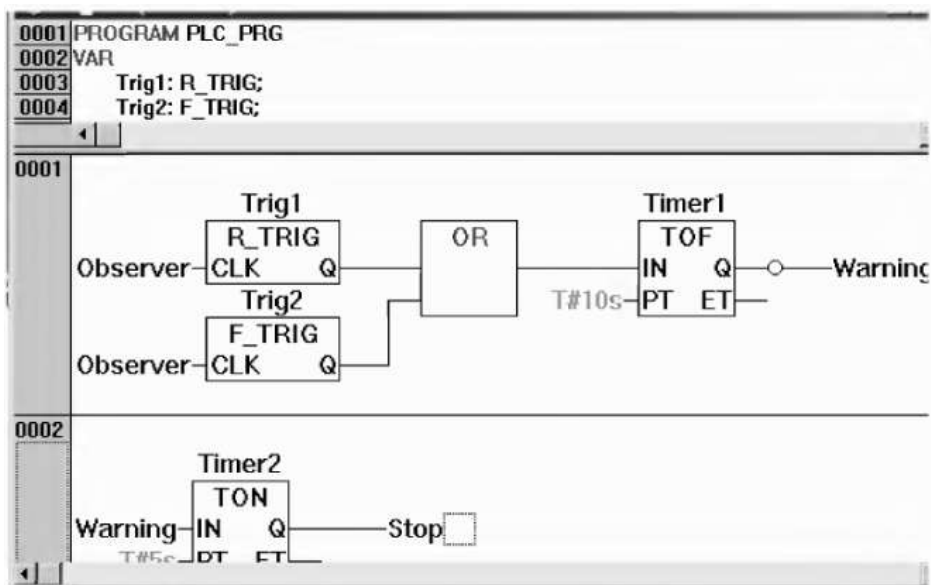


Рисунок 4.24 – Діаграма FBD з двох ланцюгів

4.5.3 Порядок виконання FBD

Виконання FBD-ланцюгів йде зліва направо, зверху вниз. Блоки, розташовані лівіше, виконуються раніше. Блок починає обчислюватися тільки після обчислення значень усіх його входів. Подальших обчислень не буде, доки не будуть обчислені значення на всіх виходах. Іншими словами, значення на всіх виходах графічного блоку з'являються водночас. Обчислення ланцюга вважається закінченим тільки після обчислення значень на виходах усіх елементів, які входять до нього.

У деяких системах програмування користувач має можливість вільно пересувати блоки зі збереженням зв'язків. У цьому випадку орієнтуватися потрібно, виходячи з порядку з'єднань. Редактор FBD COSESYS автоматично розставляє блоки в порядку виконання.

4.5.4 Інверсія логічних сигналів

Інверсія логічного сигналу в FBD зображується у вигляді кола на з'єднанні, перед входом або змінною. Інверсія не є властивістю самого блоку і може бути легко додана або скасована безпосередньо в діаграмі. У CODESYS це робиться командою «Negate». На рис. 4.24 вихід Q екземпляра функціонального блоку TOF інвертується перед присвоєнням його значення змінній Warning.

4.5.5 З'єднувачі і зворотні зв'язки

З'єднувачі (connectors) являють собою поймаєне з'єднання, яке можна розірвати і перенести в наступний ланцюг. Такий прийом може знадобитися за обмеженої ширини вікна редактора FBD. У CODESYS ширина вікна не обмежена, тому з'єднувачі тут не потрібні.

Стандарт не забороняє з'єднання, що надходять з виходу блоку на свій вхід або вхід раніше виконуваних блоків. Зворотний зв'язок не утворює цикл, подібний FOR, просто деяке обчислене значення надійде на вхід з наступним викликом діаграми. Фактично це означає неявне створення змінної, яка зберігає своє значення між викликами діаграми.

В редакторі FBD CODESYS зворотні з'єднання заборонені. Для створення зворотного зв'язку використовуйте явно оголошену внутрішню змінну.

За необхідності перенесення або розгалуження з'єднання в інші ланцюги також необхідно використовувати проміжні локальні змінні.

4.5.6 Мітки, переходи і повернення

Порядок виконання FBD-ланцюгів діаграми можна примусово змінювати, використовуючи мітки й переходи, так само, як і в релейних схемах.

Мітка ставиться на початку будь-якого ланцюга, і є назвою даному ланцюгу. Ланцюг може містити тільки одну мітку. Імена міток підпорядковані загальним правилам найменування ідентифікаторів МЕК. Графічний редактор автоматично нумерує ланцюги діаграми. Ця нумерація застосовується виключно для документування і не може замінювати мітки.

Перехід обов'язково пов'язаний з логічною змінною і виконується, якщо змінна має значення ІСТИНА. Для створення безумовного переходу використовується константа ІСТИНА, яка пов'язана з переходом. Мітки та переходи в FBD наведені у прикладі на рис. 4.25.

Зверніть увагу на останній ланцюг на рис. 4.25 – він є порожнім. Порожній ланцюг позначається єдиною константою TRUE.

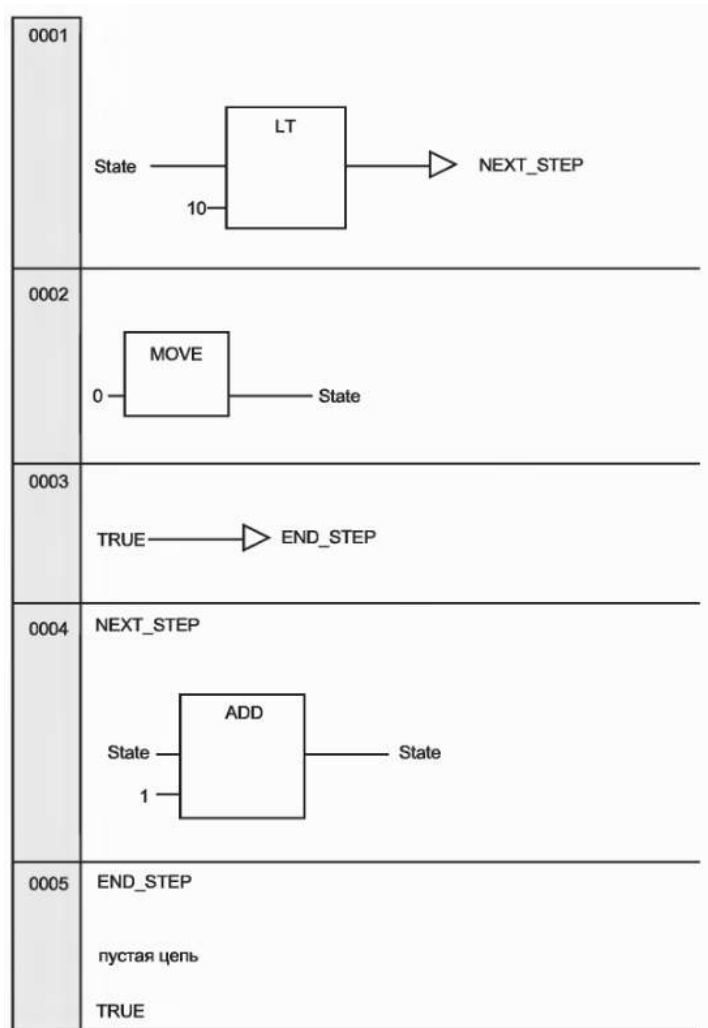


Рисунок 4.25 – Мітки та переходи в FBD

Оператор повернення RETURN можна використовувати в FBD так само, як і перехід на мітку, тобто у зв'язці з логічною змінною. Повернення призводить до негайного закінчення роботи програмного компонента і повернення на верхній рівень вкладень. Для основної програми – це початок робочого циклу ПЛК.

4.5.7 Вирази ST в FBD

COSEDYS дозволяє записувати вирази ST на вході графічних блоків. Такий прийом розширює стандартний FBD і часто виявляється досить зручним. Компактна форма подання слів полегшує запис і читання функціональних діаграм (рис. 4.26).

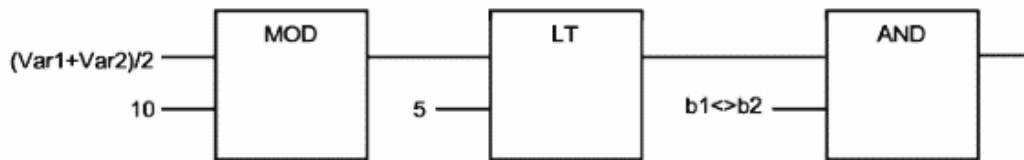


Рисунок 4.26 – Фрагмент FBD-діаграми з виразами

4.6 Контрольні запитання та завдання

1. Які типи даних МЕК ви можете навести?
2. Для чого призначені структури?
3. Що таке змінні? Як розподіляється пам'ять змінних?
4. Чим відрізняється пряма від порозрядної адресації?
5. Що таке функція?
6. Як створити екземпляр функціонального блоку?
7. Як здійснюється виклик екземпляра блоку?
8. Які особливості реалізації та застосування функціональних блоків?
9. Поясніть роботу ПЛК, як кінцевого автомату.
10. Що таке структурований текст?
11. Наведіть загальні відомості про мову LD.
12. Дайте визначення поняттю ланцюгів.
13. Поясніть особливості реалізації LD в CODESYS.
14. Як використовуються функціональні діаграми FBD?

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Kryvoplyas-Volodina L. Research of dynamic process in the pneumatic cylinder system of double action at the stable movement / Journal of food and packaging science technique and technologies. 2014. Vol. 2. № 4. С. 138–142.
2. Пашков Е.В. Электропневмоавтоматика в производственных процессах: учебн. / Е.В. Пашков, Ю.А. Осинський, А.А. Четверкин. - Сев.: СевНТУ, 2003. – 496 с.
3. Пашков Е.В. Промышленные мехатронные системы на основе пневмопривода: учебн. / Е.В. Пашков, Ю.А. Осинський. – Сев.: СевНТУ, 2007. – 401 с.
4. Невлюдов І.Ш. Технологія програмування промислових контролерів в інтегрованому середовищі CODESYS: Навчальний посібник / І.Ш. Невлюдов, С.П. Новоселов, О.В. Сичова. – Харків: ХНУРЕ, 2019. – 286 с. ISBN 978-966-659-265-4.
5. Лукинов А.П. Проектирование мехатронных и робототехнических устройств: учебн. / А.П. Лукинов – М.: Лань, 2012. – 308 с.
6. Попов Д.Н. Динамика и регулирование гидро- и пневмосистем: учебн. / Д.Н. Попов. – М.: 2001. – 210 с.
7. Якимчук М.В.: конспект лекцій «Гідро- та пневмоприводи» для студ. спец. 7.05050206 та 7.05050207 денної та заочної форми навчання./ уклад. М.В. Якимчук, А.І. Волчко. – К.: РВЦ НУХТ, 2007. – 84с.
8. Мехатронне обладнання для пакувальної галузі компанії Фесто. – Режим доступу: <http://www.festo-didactic.com/ru-ru/> – 07.11.2019 р.
9. Мехатронне обладнання для пакувальної галузі компанії Камоці. – Режим доступу: http://www.camozzi.ua/index.php?option=com_virtuemart&Itemid=293 – 07.11.2019 р.
10. Методичне та літературне забезпечення з напрямку мехатроніки. – Режим доступу: <http://www.labyrinth.ru/books/356405/> – 07.11.2019 р.
11. Черевко О.І., Кіптела Л.В., Михайлов В.М. та ін. Автоматизація виробничих процесів: підручник. – Харків: ХДУХТ, 2014. – 186 с.
12. Шишов О.В. Интеллектуальные датчики в системах промышленной автоматизации. Электроника и информационные

технологии. – 2011. Вып. 2. URL: <http://fetmag.mrsu.ru/> (дата звернения: 25.03.2019).

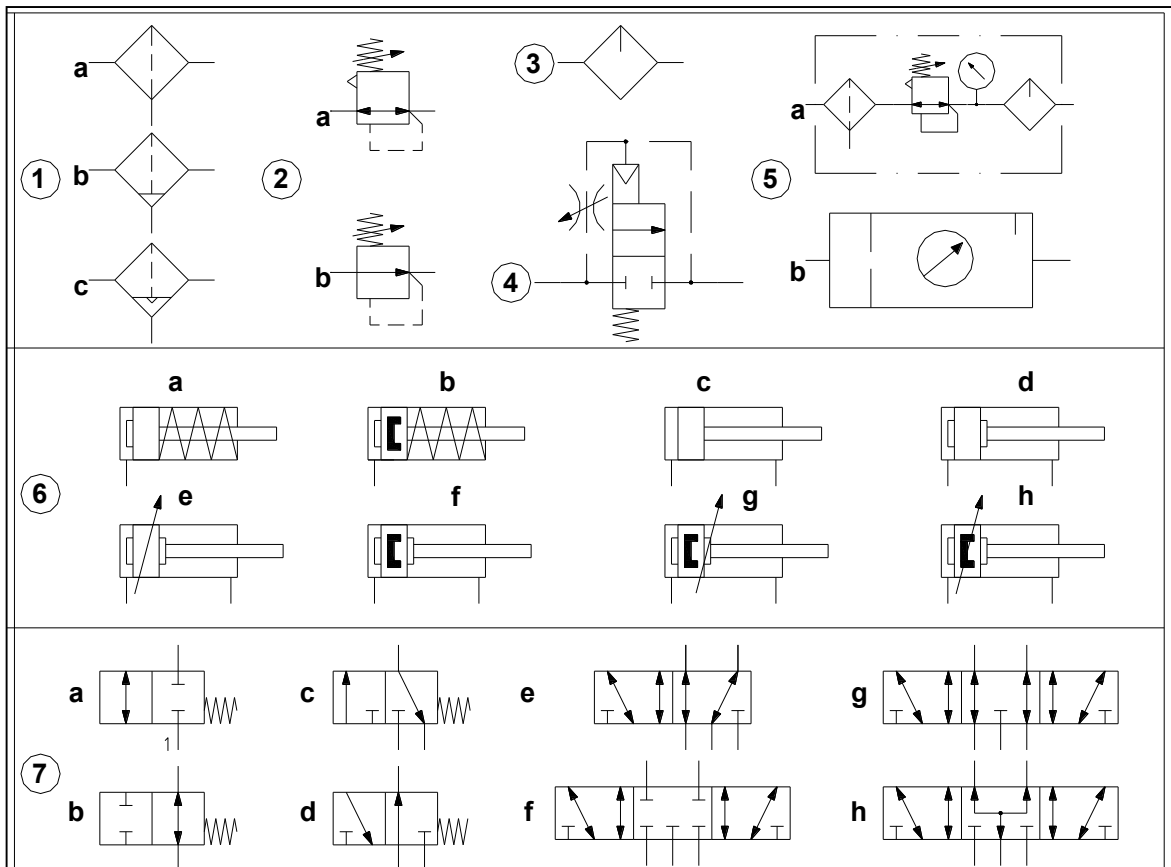
13. Петров И.В. Программируемые контроллеры. Стандартные языки и приемы прикладного программирования / Под ред. проф. В.П. Дьяконова. – М.: СОЛОН-Пресс, 2004. – 256 с.

14. Минаев И.Г., Самойленко В.В. Программируемые логические контроллеры: практическое руководство для начинающего инженера. – Ставрополь: АРГУС, 2009. – 100 с.

15. Макаров И.М. Интеллектуальные системы автоматического управления / Под ред. И. М. Макарова, В. М. Лохина. – М.: ФИЗМАТЛИТ, 2001. – 576 с.

ДОДАТОК А

СИСТЕМА ПОЗНАЧЕНЬ У ПНЕВМАТИЦІ



Підготовка повітря

1а Фільтр узагальнена індикація

1b "з ручним розвантаженням конденсату

1c "з автоматичним розвантаженням конденсату

2а Регулятор тиску із розвантаженням надлишкового тиску (відрізок з двома стрілками позначає напрямок потоку до пристрою подачі повітря і напрямок розвантаження надлишкового тиску).

2b "без розвантаження надлишкового тиску (відсутній трикутник на відводі)

3 Масилорозпилювач, узагальнена індикація

4 Прогресивний пусковий пристрій, відрегульований потік на виході поступово нагнітає тиск на установку. Після досягнення близько 50% тиску на вході, пусковий пристрій повністю відкриває прохід. Стрілка показує напрямок потоку.

5a Група вузла підготовки повітря БПП (фільтр, регулятор тиску, мастилорозпилювач)

5b "спрощена індикація БПП

Циліндри

6a пневматичний циліндр односторонньої дії

6b пневматичний циліндр односторонньої дії магнітний(магніт вбудовано у тілі поршня)

6c пневматичний циліндр двосторонньої дії

6d пневматичний циліндр двосторонньої дії з амортизацією

6e пневматичний циліндр двосторонньої дії з демпфуванням

6g пневматичний циліндр двосторонньої дії з амортизацією по прямому ходу, із демпфуванням, магнітний

6h пневматичний циліндр двосторонньої дії з амортизацією, із демпфуванням, магнітний

Розподільники

7a 2/2 моностабільний, нормально закритий (2/2 NC) – двох лінійний двох позиційний;

7b 2/2 моностабільний, нормально відкритий (2/2 NO) – двох лінійний двох позиційний (2/2 NO);

7c 3/2 моностабільний, нормально закритий (3/2 NC) – двох лінійний двох позиційний (3/2 NC);

7d 3/2 моностабільний, нормально відкритий (3/2 NO) – двох лінійний двох позиційний (3/2 NO);

7e 5/2 – п'яти лінійний двох позиційний

7f 5/3 – п'яти лінійний трьох позиційний , закриті центральні позиції (5/3 CC)

7g 5/3 – п'яти лінійний трьох позиційний , відкриті центральні позиції (5/3 CO)

7h 5/3 – п'яти лінійний трьох позиційний , "центральні позиції під тиском (5/3 CP)

Напрямок стрілки відповідає напрямку потоку.

Деякі позначення доповнюються відповідними позначками пристроїв приводу.

Приклад формування пневматичних схем.

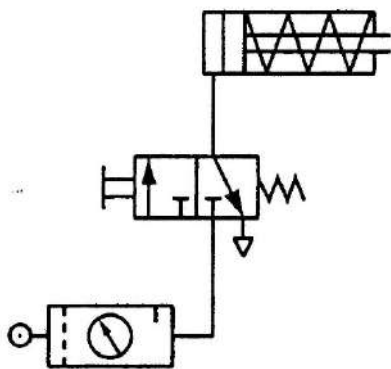


Схема 1

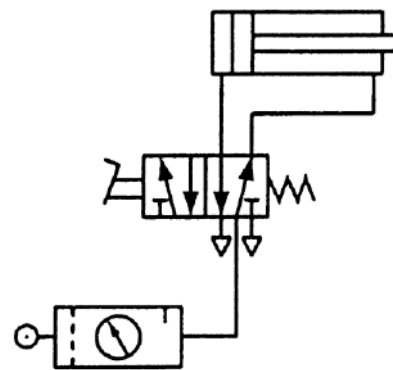


Схема 2

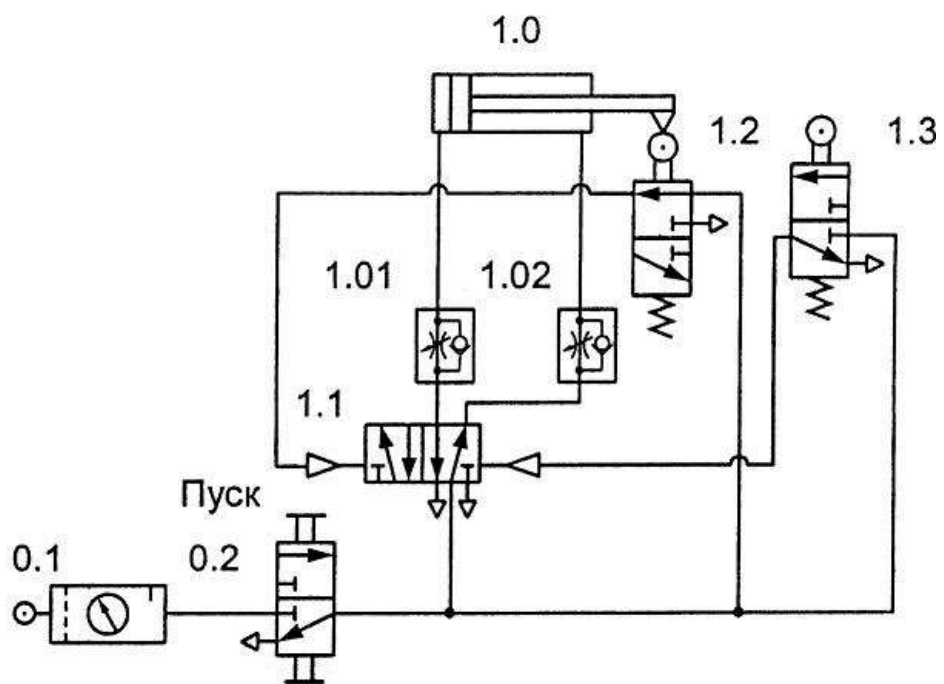


Схема 3

ДОДАТОК Б

СИСТЕМА ПОЗНАЧЕНЬ В ЕЛЕКТРИЧНИХ СХЕМАХ

<p>①</p> <p>a) </p> <p>b) </p> <p>c) </p>	<p>②</p> <p>a) </p> <p>b) </p>	<p>③</p> <p>a) </p> <p>b) </p>	<p>④</p> <p>a) </p> <p>b) </p>	<p>⑤</p> <p></p>
<p>⑥</p> <p>a) </p> <p>b) </p>	<p>⑦</p> <p>a) </p> <p>b) </p>	<p>⑧</p> <p>a) </p> <p>b) </p>	<p>⑨</p> <p>a) </p> <p>b) </p> <p>c) </p>	

У пневматиці і електриці використовуються такі визначення як: нормально відкритий і нормально закритий.

Ці визначення, в двох різних контекстах мають різні значення:

- в пневматиці:

- a) клапан відкритий, коли він дозволяє проходження через нього повітря;
- б) закритий, коли проходу для повітря немає.

- в електриці:

- a) контакти, кожен з яких складається з двох металевих частин, одна з яких рухома;
- б) визначення відкритий контакт має на увазі, що струм не проходить, оскільки дві деталі не з'єднані, немає механічної безперервності;

в) визначення закритий контакт має на увазі, що дві деталі з'єднані, є механічна безперервність, струм проходить через контакт.

Контакти можуть приводити в дію різного типу приводи із системами керування: ручними, механічними або електричними. Наприклад - перемикачі, девіатори, кнопки, реле, дистанційні вимикачі, та інші.

Електричне з'єднання в цих елементах зазвичай виконано з використанням затискачів або фастонів. Схематично, на пневматичних схемах мережу розподілу стисненого повітря відображають безперервним відрізком, а знаки управління за допомогою штрихової лінії.

На електричній схемі керування, силова лінія відображається двома безперервними відрізками, виділеними жирною лінією, лінії логіки відходять від силових ліній, вони більш тонкі і вказують функції, такі як котушки дистанційних вимикачів або реле, відповідні контакти, заряд тощо.

Читання електричної схеми здійснюється зверху вниз і зліва направо. Рекомендується нумерувати кожен окрему лінію логіки і вказувати на контакт ідентифікуючу букву, що позначає реле, частиною якої вона є.

1а силова лінія;

1b підключення сполучних ліній;

1с перетин сполучних ліній без підключення;

2а управління кнопкою з нормально відкритим контактом;

2b управління кнопкою з нормально закритим контактом;

3а управління селектором з нормально відкритим контактом;

3b управління селектором з нормально закритим контактом;

4а управління кнопкою безпеки з нормально відкритим контактом;

4b управління кнопкою безпеки з нормально закритим контактом;

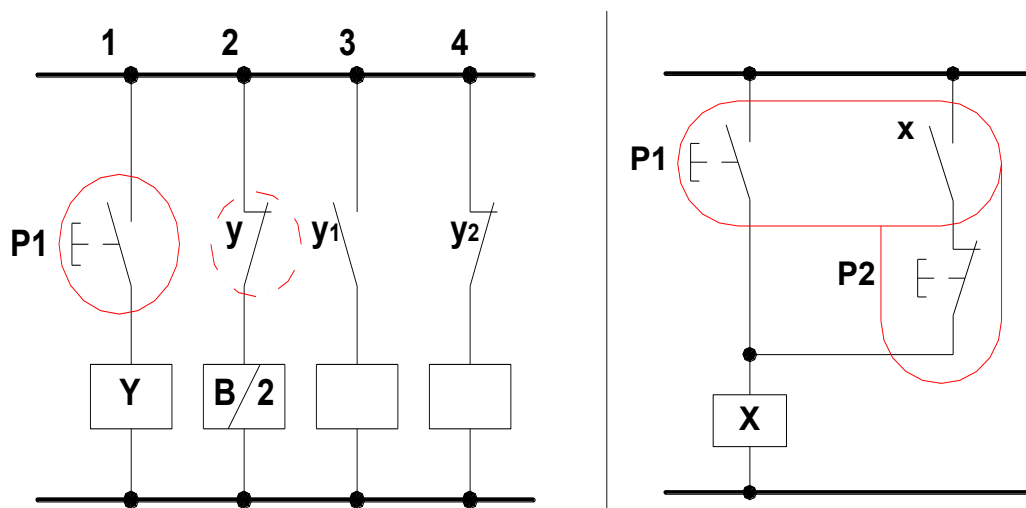
5 перемикаючий (перекідний) контакт;

6а мікрОВИМИКАЧ кінцевого вимикача з нормально відкритим контактом;

6b мікрОВИМИКАЧ кінцевого вимикача з нормально закритим контактом;

- 7a реле з нормально відкритим контактом;
- 7b реле з нормально закритим контактом;
- 8a пристрій з нормально відкритим контактом;
- 8b пристрій з нормально закритим контактом;
- 9a котушка електроклапана;
- 9b котушка реле;
- 9c котушка зовнішнього пристрою.

Приклад формування РКС схем:



Навчальне видання

НЕВЛЮДОВ Ігор Шакирович

КРИВОПЛЯС-ВОЛОДИНА Людмила Олександрівна

НОВОСЕЛОВ Сергій Павлович

СИЧОВА Оксана Володимирівна

ПНЕВМАТИЧНІ ПРИСТРОЇ ТА ЗАСОБИ АВТОМАТИЗАЦІЇ
МЕХАТРОННИХ СИСТЕМ

Навчальний посібник