

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет _____ комп'ютерної інженерії і управління
(повна назва)

Кафедра _____ електронних обчислювальних машин
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

Рівень вищої освіти _____ другий (магістерський)

**Метод створення керуючих програм
на базі TVP-технології**

(тема)

Виконав:

студент _____ II курсу, групи _____ СПЗм-20-1
Білик Ю. Ю.
(прізвище, ініціали)

Спеціальність _____
123 «Комп'ютерна інженерія»
(код і повна назва спеціальності)

Тип програми _____ освітньо-наукова
(освітньо-професійна або освітньо-наукова)

Освітня програма _____
Системне програмування
(повна назва освітньої програми)

Керівник: _____ доц. Бовчалоук С. Я.
(посада, прізвище, ініціали)

Допускається до захисту

В. о. зав. кафедри ЕОМ

(підпис)

Волк М.О.

(прізвище, ініціали)

2022 р.

Харківський національний університет радіоелектроніки

Факультет _____ комп'ютерної інженерії та управління _____

Кафедра _____ електронних обчислювальних машин _____

Рівень вищої освіти _____ другий (магістерський) _____

Спеціальність _____ 123 «Комп'ютерна інженерія» _____
(код і повна назва)

Тип програми _____ освітньо-наукова _____
(освітньо-професійна або освітньо-наукова)

Освітня програма _____ Системне програмування _____
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

“ _____ ” _____ 20__ р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

студенту _____ Білик Юлії Юріївні _____
(прізвище, ім'я, по батькові)

1. Тема роботи Метод створення керуючих програм на базі TVP-технології

затверджена наказом по університету від “ 25 ” березня 2022 р. № 33 Стз

2. Термін подання студентом роботи до екзаменаційної комісії 18 травня 2022 р.

3. Вхідні дані до роботи 1) технологія керування на паралельних структурах

2) технології програмування систем керування технологічним обладнанням

3) архітектура промислових контролерів

4) мови програмування ППЛК

4. Перелік питань, що потрібно опрацювати у роботі _____

1) Аналіз засобів керування промисловим технологічним обладнанням

2) Аналіз мов і технологій програмування промислових систем автоматизації

3) Перспективні підходи і технології програмування промислових систем автоматизації

4) Технологія побудови промислових систем керування з паралельною архітектурою

5) Технологія автоматичної трансляції технологічної циклограми у програмний код

6) Розробка TVP-технології трансляції циклограми у код ППЛС-контролера

7) Висновки

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів) Слайд-презентація – 14 слайдів

6. Консультанти розділів роботи (заповнюється за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Аналіз засобів керування промисловим обладнанням	25.03.22-01.04.22	
2	Аналіз мов і технологій програмування промислових систем автоматизації	02.04.22-10.04.22	
3	Дослідження структур паралельних ПЛК	11.04.22-14.04.22	
4	Дослідження технологій програмування ПЛК паралельної дії	15.04.22-17.04.22	
5	Розробка TVP-технології для трансляції циклограми у код ПЛІС-контролера	18.04.22-08.05.22	
6	Оформлення матеріалів атестаційної роботи	09.05.22-14.05.22	
7	Подання атестаційної роботи на рецензування	15.05.22-17.05.22	

Дата видачі завдання 28 березня 2022 р.

Студент _____
(підпис)

Керівник роботи _____
(підпис)

доц. Бовчалюк С. Я.
(посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 69 с., 16 рис., 3 дод., 21 джерела.

TVР-ТЕХНОЛОГІЯ ПРОГРАМОВАНІ ЛОГІЧНІ КОНТРОЛЕРИ, ППЛК, ПЛІС-КОНТРОЛЕРИ ПАРАЛЕЛЬНОЇ ДІЇ, HDL, ЯПЛК, ЯПЛК-М, ТЕХНОЛОГІЧНА ЦИКЛОГРАМА.

Метою кваліфікаційної роботи є вдосконалення методу проектування програмного забезпечення для промислових пристроїв логічного керування з паралельною архітектурою, шляхом створення середовища програмування на основі TVР-транслятора.

У ході виконання кваліфікаційної роботи було проведено аналіз засобів керування промисловим технологічним обладнанням, а також аналіз мов і технологій програмування промислових систем автоматизації. Виконано дослідження технологій побудови промислових систем керування з паралельною архітектурою, розглянуто архітектури класичних ППЛК і сучасних ПЛІС-контролерів паралельної дії. Досліджено технологію програмування ПЛК паралельної дії і визначено місце у ній TVР-технології. Розроблено елементи технології автоматичної трансляції циклограми у програмний код ПЛК і TVР-технологію трансляції технологічної циклограми мовою ЯПЛК-М у код ПЛІС-контролера.

ABSTRACT

Mater's thesis: 69 pages, 16 figures, 3 appendices, 21 sources.

TVP-TECHNOLOGY, PROGRAMMED LOGICAL CONTROLLERS, PLC OF PARALLEL ACTION, CPLD-CONTROLLERS, HDL, ЯПЛК, ЯПЛК-М, TECHNOLOGICAL CYCLOGRAM.

The purpose of the qualification work is to improve the method of designing software for industrial logic control devices with parallel architecture, by creating a programming environment based on TVP-translator.

During execution of the qualification work the analysis of means of control of industrial technological equipment, and also the analysis of languages and technologies of programming of industrial automation systems was carried out. Research of technologies of construction of industrial control systems with parallel architecture is executed, architectures of classical PPLK and modern FPGA controllers of parallel action are considered. The technology of parallel-action PLC programming has been studied and the place of TVP-technologies in it has been determined. Elements of the technology of automatic translation of the cyclogram into the program code of the PLC and TVP-technology of the translation of the technological cyclogram in the ЯПЛК-М language into the code of the FPGA controller have been developed.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ	8
ВСТУП	10
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	12
1.1 Аналіз засобів керування промисловим технологічним обладнанням	12
1.2 Аналіз мов і технологій програмування промислових систем автоматизації.....	19
1.3 Перспективні підходи і технології програмування промислових систем автоматизації.....	23
1.4 Визначення задач кваліфікаційної роботи	25
2 ТЕХНОЛОГІЯ ПОБУДОВИ ПРОМИСЛОВИХ СИСТЕМ КЕРУВАННЯ З ПАРАЛЕЛЬНОЮ АРХІТЕКТУРОЮ	27
2.1 Структура промислового програмованого логічного контролера паралельної дії.....	27
2.1.1 Класична структура ППЛК	27
2.1.2 Сучасна структура ПЛІС-контролерів паралельної дії.....	29
2.2 Технологія програмування ПЛК паралельної дії.....	31
3 TVP-ТЕХНОЛОГІЯ СТВОРЕННЯ КЕРУЮЧИХ ПРОГРАМ ДЛЯ СИСТЕМ КЕРУВАННЯ З ПАРАЛЕЛЬНОЮ АРХІТЕКТУРОЮ	36
3.1 Технологія автоматичної трансляції технологічної циклограми у програмний код ПЛК	36
3.2 Розробка TVP-технології для трансляції технологічної циклограми у код ПЛІС-контролера	40
ВИСНОВКИ	46
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	47
ДОДАТОК А Графічний матеріал кваліфікаційної роботи.....	50

ДОДАТОК Б Наукові публікації за темою кваліфікаційної роботи	58
ДОДАТОК В Приклади реалізації HDL-кодів блоків пам'яті ПЛІС- контролера	61
В.1 HDL-код БПК.....	61
В.2 HDL-код БПП.....	64
В.3 HDL-код БПС.....	65
В.4 HDL-код БПЗК.....	69

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

АРМ – автоматизоване робоче місце

БПЗК – блок пам'яті заборонених комбінацій

БПК – блок пам'яті команд

БПП – блок пам'яті переходів

БПС – блок пам'яті станів

ВІС – велика інтегральна схема

ІК – індустріальний комп'ютер

ЛКА – логічний керуючий автомат

ЛКА ПД – логічний керуючий автомат паралельної дії

МЕК – міжнародна електротехнічна комісія

МЕК 61131-3 – міжнародний стандарт мов програмування
програмованих логічних контролерів

МК – мікроконтролер

МПК – мікропроцесорний контролер

ПЛК – програмований логічний контролер

ПД – паралельна дія (або паралельної дії)

ПК – персональний комп'ютер

ПЗ – програмне забезпечення

ПЛІС – програмована логічна інтегральна схема

ППЛК – паралельний програмований логічний контролер

ЯППЛК – мова програмування ПЛК паралельної дії

ЯПЛК-М – вдосконалена мова програмування ПЛК паралельної дії

FBD – графічна мова функціональних блоків (англ., Functional Block
Diagram)

HDL – мова опису апаратури (англ., Hardware Description Language)

ІЛ – текстова мова низького рівня (англ., Instruction List)

LD – мова сходових діаграм (англ., Ladder Diagram)

PLC – програмований логічний контролер (англ., Programmable Logic Controller)

SFC – графічна мова описання алгоритму управління у вигляді набору пар, крок і перехід (англ, Sequential Function Chart)

ST – текстова мова програмування ПЛК високого рівня (англ, Structured Text)

TVP – технологічне візуальне програмування (англ., Technological Visual Programming)

ВСТУП

Відомо, що автоматизація і комп'ютеризація технологічних процесів є одним із найпотужніших факторів підвищення ефективності роботи виробничих підприємств. Разом з тим, темпи застосування сучасних систем управління на базі мікроелектронних і мікропроцесорних пристроїв у деяких галузях залишаються невисокими. Насамперед це характерно для систем управління відповідальними технологічними процесами, де відмова (як апаратна, так і програмна) може спричинити серйозні технологічні наслідки, загибель людей або значні економічні втрати [3, 11, 13]. Однією з проблем є наявність помилок у програмному кодї, оскільки на даний час технологія підготовки керуючих програм переважно залишається «традиційною», тобто незмінною і передбачає обов'язкову присутність у цьому процесі, щонайменше технолога і програміста. [2, 3, 4, 21]. У той же час відомі підходи, що дозволяють виконати формування програмного коду в автоматизованому режимі та уникнути значної частини помилок [1, 11, 21].

Існує думка, що основні причини, що знижують ефективність та гальмують темпи впровадження систем автоматизації, слід шукати у галузі технології підготовки програмного забезпечення. Справа в тому, що стиль роботи більшості сучасних програмістів не повністю відповідає рівню вимог, що пред'являються до програмних продуктів. Програмісти схильні сприймати свій вид діяльності як мистецтво, спираючись насамперед на власну інтуїцію, своє особисте, суб'єктивне сприйняття вирішуваних ними завдань. Такий підхід (його сьогодні цілком можна вважати загальноприйнятим) часто призводить до появи помилок у програмному забезпеченні, що, у свою чергу, стає причиною тяжких наслідків. У зв'язку з цим, набуває все більшої актуальності певною мірою парадоксальне питання: чи взагалі можна виключити програміста з процесу підготовки ПЗ або, принаймні, звести його участь і вплив на результат до мінімуму.

В даний час технологія підготовки керуючих програм залишається «традиційною»: результатом спільної роботи Виконавця та Замовника є алгоритм, представлений у неформалізованій формі (малюнки, креслення, мовний опис, схематичні пояснення, тощо), за яким програмісту необхідно створити керуючу програму для системи управління. Але відомо, що програмування, як було зазначено, є творчий процес – за одним тим же неформальним описом алгоритму, двома програмістами буде створено дві різні програми. Причому якість програмного забезпечення безпосередньо залежатиме від того, наскільки точно і однозначно описаний алгоритм управління, наскільки «правильно» програміст зрозумів сам алгоритм, від його професійних якостей тощо. Необхідно врахувати також і те, що чим складніший алгоритм управління, тим складнішим і об'ємнішим буде програмний код, отже, тим вищою стає ймовірність появи помилок (дефектів або помилок програмного забезпечення). Знаходження та виправлення цих дефектів і помилок вимагає багато ресурсів та часу, а розробка ПЗ перетворюється на монотонний багатоітераційний і, дуже часто, дорогий процес.

Необхідно також врахувати і те, що при створенні різних систем управління використовуються різні технології та мови програмування, вибір яких безпосередньо залежить від апаратних засобів (мови високого рівня для промислових комп'ютерів, спеціалізовані графічні мови та мови низького рівня для програмованих логічних контролерів, мови опису апаратури для ПЛІС-контролерів), що практично унеможлиблює розробку чи виконання аналізу ПЗ технологом – непрофесійним користувачем, які мають спеціальної підготовки у сфері програмування.

Враховуючи вищевказане, головною метою роботи є вдосконалення методу проектування програмного забезпечення для промислових пристроїв логічного керування з паралельною архітектурою, шляхом створення середовища програмування на основі TVP-транслятора, що дозволяє вирішити вищезазначене завдання.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Аналіз засобів керування промисловим технологічним обладнанням

Дано загальну характеристику промислового устаткування як об'єкта автоматизації для того щоб показати, чим і як необхідно керувати, які особливості керування об'єктами, що функціонують у режимі реального часу, а також сформулювати основні вимоги до сучасних засобів програмного керування промисловими об'єктами [7, 13].

У самому загальному вигляді процес автоматизованого керування промисловим устаткуванням може бути проілюстрований схемою, показаною на рисунку 1.1.

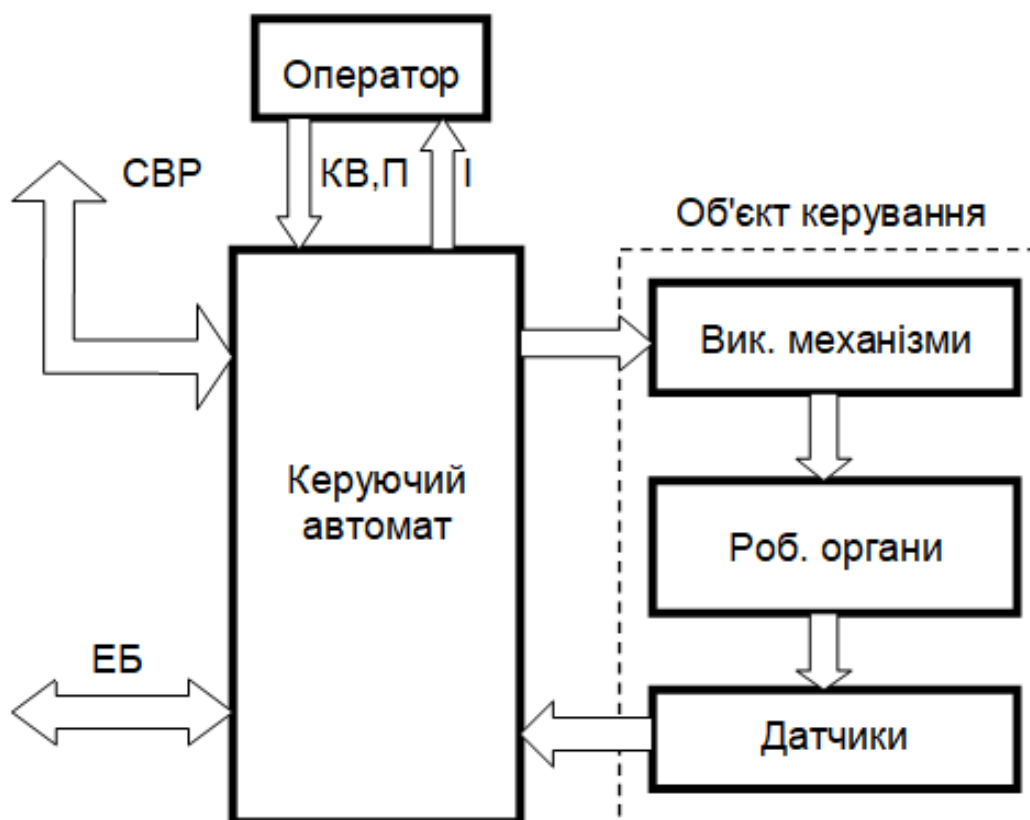


Рисунок 1.1 – Загальний вигляд процесу автоматизованого керування

У даній схемі два основні елементи: об'єкт керування (у якості якого можуть розглядатися стаціонарні і мобільні технологічні установки й агрегати, промислові роботи, транспортні засоби, допоміжне устаткування та інші об'єкти промислового призначення) і керуючий автомат (КА), що забезпечує автоматичне чи автоматизоване керування відповідно до заданого алгоритму функціонування.

Об'єкт керування містить у собі робочі органи, виконавчі механізми і датчики. Робочі органи забезпечують власне реалізацію технологічного процесу, тобто – перетворення вихідних матеріалів у напівфабрикати чи переміщення деталей (напівфабрикатів) з однієї позиції агрегату на іншу.

Виконавчі механізми служать для надання руху робочим органам. Датчики використовуються для одержання інформації про положення робочих органів і значення параметрів технологічного процесу.

У залежності від типу системи (автоматична чи автоматизована) керування промисловим об'єктом здійснюється автоматично чи за участю оператора, що може здійснювати керуючі впливи (КВ) і вводити в КА задані параметри (П) процесу, а також приймати з КА контрольну і діагностичну інформацію (І). Якщо технологічний агрегат функціонує у складі технологічного комплексу, то неодмінною умовою його нормальної роботи є наявність каналу зв'язку з пристроями керування суміжним устаткуванням (тобто, як мінімум, з агрегатами, встановленими в технологічному циклі перед і після розглянутого). За допомогою цього каналу здійснюється двосторонній обмін сигналами електричних блокувань (ЕБ).

Пристрої і системи керування сучасними технологічними комплексами, як правило, оснащуються також і каналами зв'язку із системами верхнього рівня (СВР) для одержання планової і передачі звітної і діагностичної інформації.

Якщо поверхнево поглянути на побудову сучасних систем автоматизації, то можна стверджувати, що вони переважно базуються на широкому використанні засобів мікропроцесорних контролерів (МПК).

Центральною частиною системи є мікропроцесорний пристрій, який виконує основні функції управління системою. На вхідні модулі МПК подаються сигнали від датчиків технологічних параметрів та інші вхідні сигнали. До вихідних модулів МПК підключаються виконавчі механізми та регулюючі органи. Вибір інших технічних засобів залежить від вирішення питання щодо принципів організації операторського інтерфейсу. У першому варіанті на операторському місці встановлюється пульт оператора-технолога, на якому розміщуються, як правило, малогабаритні цифрові прилади, що показують, і станції ручного управління, за допомогою яких оператор має можливість спостерігати і за станом технологічного процесу і, в разі необхідності, втручатися в процес його управління.

Найбільш поширеним сьогодні є варіант, при якому як операторську станцію використовують персональний комп'ютер (ПК). За допомогою спеціального програмного забезпечення на ПК створюється автоматизоване робоче місце (АРМ) оператора-технолога. Оператор спостерігає за технологічним процесом за допомогою кольорових мнемосхем і, користуючись клавіатурою та маніпулятором «миша», може здійснювати оперативне управління процесом: міняти завдання регуляторам окремих технологічних параметрів, переходити на ручний режим управління та безпосередньо керувати регулюючими органами, тощо. Крім того, за допомогою АРМ ведеться архівування даних, фіксація моментів виникнення аварійних та передаварійних ситуацій, ідентифікація дій оператора, підготовка та друк різних рапортів тощо.

Використання МПК як центрального керуючого пристрою має значні переваги перед традиційними локальними технічними засобами управління. Найважливішою перевагою використання мікропроцесорної техніки в системах управління є те, що при необхідності внесення змін до алгоритму управління об'єктом немає необхідності вносити зміни до структури технічних засобів, змінювати окремі регулятори та функціональні блоки, наново проводити монтажні роботи. У багатьох випадках це вирішується з

допомогою зміни у програмі управління об'єктом. Крім того, значно зростає надійність системи, насамперед за рахунок значного зменшення кількості фізичних з'єднань між окремими технічними засобами системи.

У світі існує багато фірм, що поставляють складові елементи МПК в дуже широкій номенклатурі. Нижче перераховані основні типи засобів, що поставляються фірмами, що займаються розробкою і виготовленням мікропроцесорної техніки для промислової автоматизації [11, 13]:

- однокристальні мікропроцесорні контролери (мікроконтролери) для вбудованих застосувань;
- процесорні плати;
- одноплатні мікроконтролери;
- уніфіковані шасі для компонування МП-пристроїв;
- модулі введення-виведення дискретних і аналогових сигналів;
- модулі перетворення і нормалізації вхідних сигналів;
- модулі пам'яті;
- вільно програмовані логічні контролери (ПЛК);
- індустріальні (промислові) комп'ютери;
- спеціалізовані МП-пристрої для розподілених систем збору даних і керування;
- модулі віддаленого збору даних і керування;
- персональні комп'ютери для панелей керування;
- промислові робочі станції;
- апаратура каналів зв'язку;
- допоміжне устаткування.

Модульний принцип побудови складних систем відомий ще з доісторичних часів, однак ніколи раніше не існувала можливість застосування його в таких масштабах, які спостерігаються в даний час.

Із зазначених вище сучасних засобів автоматизації найбільш масово в промисловості застосовуються ПЛК, МК і ІК. ПЛК і ІК пристосовані до роботи у важких виробничих умовах і відрізняються лише тим, що ПЛК в

основному орієнтовані на безпосереднє керування промисловим устаткуванням (тобто рішення задач у режимі «жорсткого» реального часу), а ПК більш ефективно вирішують задачі обробки і візуалізації великих обсягів інформації на верхніх рівнях складних інтегрованих систем автоматизації промислового виробництва.

В останні роки спостерігається тенденція істотного підвищення гнучкості технології промислового виробництва з метою забезпечення мінімальних термінів і вартості виготовлення продукції. Підвищення гнучкості технології виготовлення продукції викликає необхідність створення швидко переналагоджуемого устаткування, а, отже, і гнучких пристроїв, що програмно перенастроюються, і систем керування цим устаткуванням.

Для побудови пристроїв керування з програмованою логікою роботи в загальному випадку можливо і представляється природним використання універсальних комп'ютерів. Однак, застосування універсальних комп'ютерів для безпосереднього керування технологічним устаткуванням виявляється малоефективним з ряду причин, головними з яких є:

- непристосованість універсальних комп'ютерів до експлуатації в промислових умовах, тобто при впливі сильних електричних і магнітних полів, при значних коливаннях напруги живильної мережі та ін.;

- необхідність розробки і доукомплектування універсальних комп'ютерів спеціалізованими пристроями зв'язку з керуємим об'єктом;

- відсутність можливості безпосереднього програмування комп'ютерів на спрощених мовах, що не вимагають спеціальної підготовки обслуговуючого персоналу з програмування.

Обмежені можливості і низька ефективність використання універсальних комп'ютерів для керування технологічним устаткуванням сприяли появі нового класу проблемно-орієнтованих керуючих пристроїв, що одержали найменування «програмовані логічні контролери». У всіх сучасних ПЛК внутрішні електричні кола розв'язані від зовнішніх за допомогою

оптоелектронних приладів.

На даний час розроблені й освоєні промисловістю сотні моделей ПЛК, що відрізняються між собою номенклатурою і кількістю використовуваних модулів введення-виведення, швидкодією, об'ємом пам'яті для збереження керуючих програм, конструктивним виконанням та ін. Разом з тим, представляється доцільним відзначити загальні для всіх типів і моделей ПЛК і специфічні з погляду користувачів систем промислової автоматизації особливості, що істотно і вигідно відрізняють програмовані логічні контролери від універсальних комп'ютерів:

- простота спілкування з користувачем, що полягає в можливості програмування ПЛК за принциповою електричною схемою, або за логічними (булевими) рівняннями, або за допомогою простої алгоритмічної мови;

- пристосованість до роботи у важких виробничих умовах за рахунок застосування оптоелектронної гальванічної розв'язки входів і виходів від зовнішніх електричних кіл, за рахунок пристосованості ПЛК до розширеного діапазону умов експлуатації;

- модульність конструкції, що дозволяє компонувати з обмеженого числа уніфікованих модулів контролери різного функціонального призначення і необхідної конфігурації, що відкриває широкі перспективи в частині підвищення гнучкості створюваних систем керування складними технологічними модулями і роботизованими технологічними комплексами;

- різке скорочення витрат на проектування за рахунок істотного зниження вартості програмування, а також за рахунок спрощення прив'язки модульної конструкції ПЛК до конкретного об'єкта керування;

- значне скорочення термінів розробки і тиражування систем керування устаткуванням за рахунок можливості паралельного проведення робіт з їхнього проектування (у тому числі програмування) і виготовлення систем керування на основі ПЛК;

- можливість коректування алгоритмів керування безпосередньо в виробничих умовах (при монтажу, пуску, випробуваннях та модернізації

устаткування), що істотно поліпшує адаптивні якості систем керування технологічним устаткуванням;

- широка номенклатура і можливість сполучення модулів введення-виведення контролерів з датчиками, керуємими механізмами, а також керуючими пристроями, виконаними на інших принципах і іншій елементній базі, що позитивно позначається на ефективності побудови комплексних систем керування з використанням ПЛК;

- наявність вбудованої автоматичної функціональної діагностики дозволяє істотно спростити процес експлуатації і підвищити ремонтпридатність як власне контролера, так і керованого технологічного устаткування.

Завдяки перерахованим особливостям програмовані логічні контролери в короткий термін завоювали популярність у користувачів устаткування і широко застосовуються в різних галузях промисловості.

В основу структурної організації ПЛК покладена типова структура МП-пристрою, дооснащеного модулями зв'язку з керуємих об'єктом, а також пультом (пристроєм) користувача, за допомогою якого реалізуються функції програмування, налагодження, діагностування керуючої програми і відображення станів керуємого об'єкта.

Узагальнена структура ПЛК приведена на рисунку 1.2. ПЛК, як правило, містить у собі: керуючий автомат, пульт користувача і модулі зв'язку з керуємих об'єктом. Основним пристроєм ПЛК є керуючий автомат (КА), виконаний на базі МП і який здійснює програмне керування процесом запису, збереження і відпрацьовування керуючих програм (КП) у ПЛК.

Пульт користувача складається з панелі (пульта, пристрою) програмування і налагодження і пристрою відображення інформації, на який можна виводити інформацію про стан керуємого об'єкта, фрагменти КП (при запису, налагодженні, виконанні і коректуванні) і інша інформація.

Модулі зв'язку з керуємих об'єктом являють собою набори модулів введення-виведення дискретних і аналогових сигналів (типи і кількість

модулів, що підключаються, залежать від архітектури і конструкції конкретної моделі ПЛК). Для стикування з іншими мікропроцесорними системами ПЛК може оснащуватися стандартними промисловими інтерфейсами.

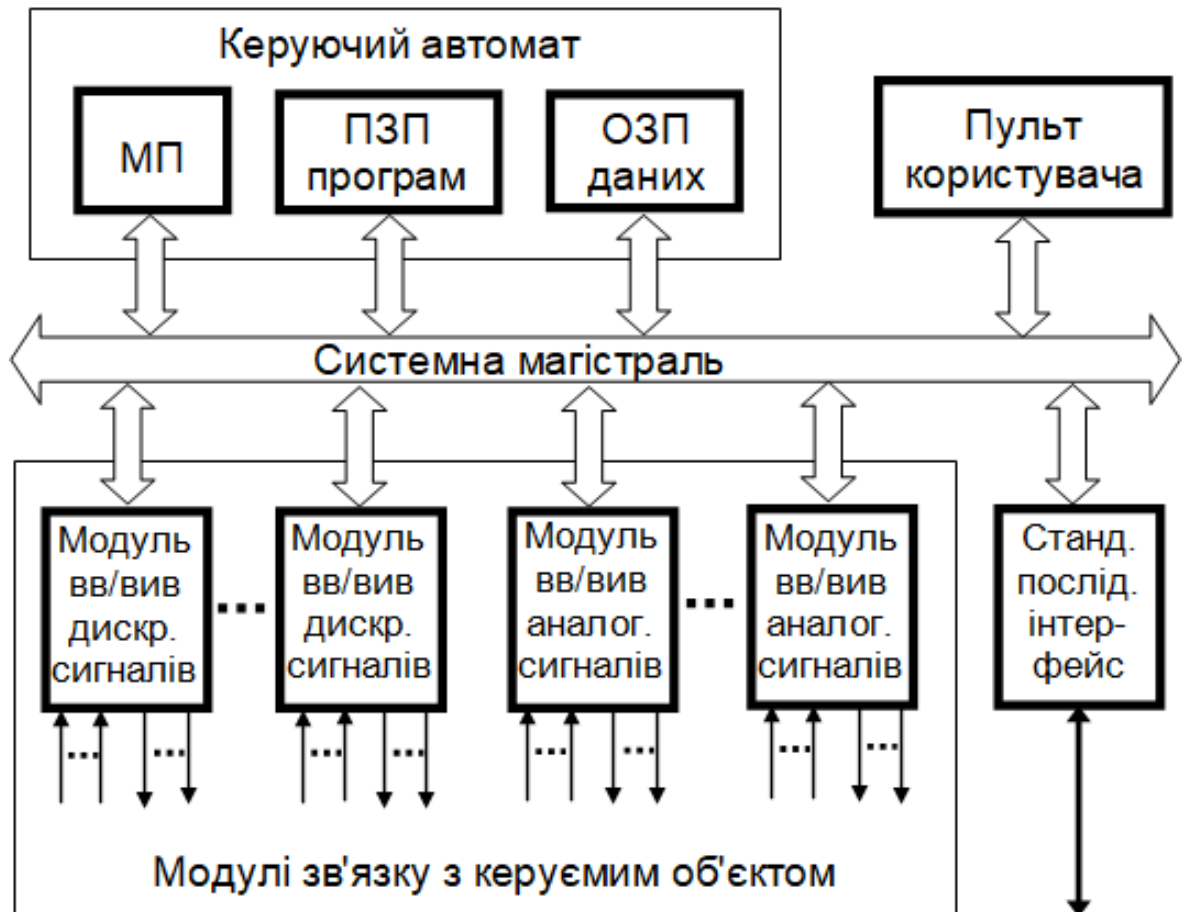


Рисунок 1.2 – Узагальнена структура ПЛК

1.2 Аналіз мов і технологій програмування промислових систем автоматизації

Однією з найважливіших вимог до ПЛК є можливість експлуатації існуючим технічним персоналом та можливість швидкої модернізації обладнання. Тому стандартні процедурні мови програмування обчислювальної техніки погано підходять для програмування ПЛК. В цьому

випадку для повного задоволення потреб поряд з текстовими мовами загального призначення необхідні більш прості, наочні, проблемно-орієнтовані мови. Необхідні мови досить досконалі, але спеціалізовані під задачі логічного керування та тому доступні фахівцям з області автоматики, а не інформатики. Тому з моменту появи ПЛК для завдання алгоритму керування широко застосовували безліч варіацій мов релейно-контактних символів, булевих формул, мнемонічних мов.

Уперше, після впровадження в системі керування конвеєрного виробництва (США, 1969 р.), ПЛК прийшов на зміну релейно-контактній базі систем керування. Для того щоб фахівець-технолог міг вільно перенести алгоритм керування на нову елементну базу, «перерисувавши» існуючу схему керування на дисплеї програмуючої станції ПЛК, в США була розроблена мова релейно-контактних схем (LD). В Європу ж мода на ПЛК прийшла трохи пізніше, коли релейні шафи були вже успішно замінені на шафи з логічними мікросхемами. Тому виникла необхідність винаходу інших мов програмування, зрозумілих новому поколінню інженерів. Так у Німеччині з'явилися мови простих текстових інструкцій, що нагадують асемблер (IL). У Франції виникли графічні мови функціональних блокових діаграм (FBD) та діаграми опису етапів та умов переходів («Графсет», сучасний SFC). Застосовувалися також мови, що використовуються для програмування комп'ютерів (Pascal, Basic). Наприкінці сімдесятих років зложилася вкрай складна ситуація. Кожен виробник ПЛК (в тому числі й у Радянському союзі) розробляв власну мову програмування, тому ПЛК різних виробників були програмно несумісні, крім того, існувала проблема апаратної несумісності. Заміна ПЛК на продукт іншого виробника перетворювалася у величезну проблему. Покупець ПЛК був змушений використовувати вироби тільки однієї фірми, або витратити сили на вивчення різних мов та коштів на придбання відповідних інструментів.

У підсумку, в 1979 році в рамках Міжнародної Електротехнічної Комісії (МЕК) була створена спеціальна група технічних експертів з проблем

ПЛК. Перед нею було поставлено задачу виробити стандартні вимоги до апаратних засобів, програмного забезпечення, правил монтажу, тестування, документування та засобів зв'язку ПЛК. В 1982 році був опублікований перший чорновий варіант стандарту, що отримав найменування МЕК 1131. Через складність документу, що вийшов, було вирішено розбити його на кілька частин, питанням програмування присвячена третя частина стандарту «Мови програмування ПЛК». Оскільки з 1997 року МЕК перейшла на п'ятизначні позначення, сьогодні правильне найменування міжнародної версії частини стандарту, присвяченої мовам програмування ПЛК – МЕК 61131-3.

Робочою групою МЕК з усього різноманіття існуючих на момент розробки стандарту мов програмування ПЛК були виділені 5 мов, що отримали найбільше поширення. Специфікації мов були дороблені, так що стало можливим використовувати в програмах, що написані на кожній з цих мов, стандартизований набір елементів та типів даних. Реалізація подібного підходу дозволила залучити до програмування того самого ПЛК фахівців різних галузей знань та різної кваліфікації: фахівців з релейної автоматики, електриків (мова релейно-контактних схем LD), фахівців з галузі напівпровідникової схемотехніки та автоматичного регулювання (мова функціональних блоків FBD), програмістів, що мають досвід написання програм для комп'ютерів мовою асемблера (мова інструкцій IL), на мовах високого рівня (мова ST), та навіть далеких від програмування фахівців-технологів (мова SFC). Хоча впровадження МЕК систем програмування й не дозволило повністю відмовитися від послуг професійних програмістів, але дозволило знизити вимоги до кваліфікації й, відповідно, витрати на оплату праці програмістів ПЛК.

Стандартизація мов дозволила частково вирішити проблему залежності користувача ПЛК від конкретного виробника. Всі сучасні ПЛК оснащуються засобами МЕК 61131-3 програмування, що спрощує роботу користувачам контролерів (можна використовувати ПЛК різних фірм без витрат на

перенавчання) та одночасно знімає ряд проблем для виробників ПЛК (можна використовувати компоненти ПЛК інших виробників). Хоча деякі найстарші виготовлювачі ПЛК дотепер змушені підтримувати свої власні мови (системи програмування), однак всі вони, в тій чи іншій формі, прагнуть забезпечити підтримку МЕК 61131-3.

Стандарт істотно розширив на ринку праці можливості фахівця, що займається програмуванням ПЛК. Фахівець, що вивчив мови МЕК 61131-3, зможе розібратися з програмою будь-якого сучасного ПЛК. Це дозволило зменшити як залежність фірми від фахівця з програмування ПЛК, так і фахівця від фірми.

Таким чином, стандарт МЕК 61131-3 описує синтаксис та семантику п'яти мов програмування [6, 16]:

- SFC (Sequential Function Chart) – графічна мова, що використовується для опису у вигляді набору зв'язних пар: крок (step) та перехід (transition);
- LD (Ladder Diagram) – графічна мова програмування, що є стандартизованим варіантом класу мов релейно-контактних схем;
- FBD (Functional Block Diagram) – графічна мова, по суті схожа на LD, але замість реле в цій мові використовуються функціональні блоки;
- ST (Structured Text) – текстова мова високого рівня загального призначення, по синтаксису орієнтована на Паскаль;
- IL (Instruction List) – текстова мова низького рівня, стандартизований асемблер.

Контролери, програмування яких дотепер здійснюється з вбудованого або зовнішнього пульта зустрічаються сьогодні практично не зустрічаються (в основному для спеціалізованих ПЛК – керування освітленням, температурою й т.д.). В якості програматора універсальних ПЛК застосовуються персональні комп'ютери зі спеціалізованими середовищами розробки керуючих програм. Сьогодні практично всі виробники систем програмування ПЛК реалізують можливість програмування на мовах стандарту МЕК 61131-3. Великі виробники контролерів пропонують потужні

програмні комплекси з підтримкою мов стандарту, але орієнтовані суцього на контролери власного виробництва. Розробка й підтримка повноцінної системи програмування є складним та дорогим процесом. Але такий підхід робить залежним розроблювача від конкретного виробника, тому що при переході на елементну базу іншого виробника доведеться купувати та освоювати й нову систему програмування. Складність розробки повноцінного середовища програмування й відкритість стандарту МЕК 61131-3 привели до появи фірм, що спеціалізуються винятково на розробці засобів програмування ПЛК, що підтримують контролери з відкритою архітектурою різних виробників.

1.3 Перспективні підходи і технології програмування промислових систем автоматизації

На сьогоднішній день багато фахівців усвідомлюють необхідність удосконалення технології проектування ПЗ. Серед безлічі підходів, спрямованих на вирішення цього завдання, можна виділити наступні:

- прагнення використовувати спеціалізовані візуальні мови алгоритмізації, такі як мова функціональних блокових діаграм FBD, сходових діаграм LD (релейноконтактних символів), мова «Графсет» тощо;

- так зване автоматне програмування, тобто. опис алгоритму управління як кінцевого автомату [20];

- схемна емуляція – новий нетрадиційний підхід, що охоплює як сам процес програмування, так і методи реалізації як функціонально закінчених апаратних засобів [20].

Кожен із перерахованих підходів має певні переваги і має область ефективного використання. Разом із цим, слід зазначити, що жоден з них не дозволяє автоматизувати процес синтезу програмного коду з урахуванням первинного формального представлення алгоритму логічного управління (наприклад, циклограмми).

Одним з підходів, що має суттєві переваги і є таким, що можна вважати перспективним – це мова технологічних циклограм.

Мова технологічних циклограм дозволяє у табличній формі описати алгоритм управління безпосередньо у термінах технологічного процесу, а не в абстрактних термінах інформатики. Таблична форма представлення алгоритму управління (у вигляді циклограми) є простим, наочним та водночас універсальним засобом спілкування між фахівцями різного профілю: технологами, конструкторами, програмістами, налагодчиками. Тому даний підхід підготовки керуючих програм отримав назву TVP-технологія (Technological Visual Programming, або технологічне візуальне програмування).

Програма є виконаною за певними правилами формалізованою технологічною циклограмою у вигляді двох (або більше) таблиць: таблиці станів та таблиці переходів. Вхідними станами є стани входів контролера (стани зовнішньої середовища), вихідними – стани виходів контролера (команди управління виконавчими механізмами). Приклад таблиці станів наведено на рисунку 1.3

У процесі аналізу технологічного процесу формується простір можливих станів об'єкта. Якщо управляюча програма повинна обробляти позаштатні ситуації, то простір станів розширюється за рахунок додавання аварійних станів. Кожний рядок таблиці станів містить інформацію про номер стану і виконувану в цей момент операцію (номер операції, умова виконання та команди управління виконавчими механізмами, реалізовані на поточному кроці циклограми).

Для опису переходів між станами (динаміки процесу управління) використовується таблиця переходів технологічної циклограми. Кожен рядок таблиці містить номер стану керуючого автомата, умову переходу до нового стану, ознаку аварійного переривання та номер стану, до якого повинен перейти керуючий автомат при виконанні заданої умови.

Дані таблиці є взаємозалежними: умовні переходи між станами

керуючого автомата описуються таблицею станів. Такий підхід дозволяє самому технологу формалізувати алгоритм управління шляхом заповнення таблиць циклограми, використовуючи природні терміни технологічного процесу. При цьому на етапі формалізації алгоритму управління не важливо, на базі якого мікроелектронного засобу буде побудовано систему автоматизації (ПЛК, ІК, ПЛС-контролер) та на якій мові програмування буде закодовано алгоритм (мов стандарту МЕК 61131-3, Сі, VHDL і т.д.).

Технологічні операції й групи операцій (мікроцикли)	Адреса	Логічна операція	Стан входів				Команди керування					
			Датчик 1	Датчик 2	Датчик 3	...	Виконавець 1	Виконавець 2	...	Ознака кінця відпрацювання		
			1	2	3	...	1	2	...	L		
Початкові установки	0	I									1	
Стан 1	Операція 1.1	1	АБО		1	1			1			
	Операція 1.2	2	I	1	0				0	1		
									
	Операція 1.X	n	I									1
Стан 2	Операція 2.1	1	АБО	1		0				1		
	Операція 2.2	2	I									
									
	Операція 2.X	k	АБО		1							1

Рисунок 1.3 – Таблиця станів технологічної циклограми

1.4 Визначення задач кваліфікаційної роботи

Отже з матеріалу попередніх розділів можна зробити висновок, що більшість сучасних систем промислової автоматизації будується на основі високонадійних і легко компонуємих програмованих логічних контролерів та

індустріальних комп'ютерів. Ефективність практичного застосування ПЛК значною мірою залежить від використовуваних методів і засобів автоматизації їхнього програмування, для чого створені і широко застосовуються на практиці різні високопродуктивні інструментальні засоби. TVP-технологію, що пропонується, можна розглядати як додатковий засіб автоматизації процесу програмування ПЛК, що може бути застосований разом з відомими інструментальними засобами.

У зв'язку із зазначеним актуальною є розв'язання науково-прикладної задачі дослідження методу створення керуючих програм на базі TVP-технології. Відповідно до зазначеної мети необхідно розв'язати наступні часткові завдання дослідження:

- виконати аналіз засобів керування промисловим технологічним обладнанням, а також мов і технологій їх програмування;
- визначити перспективні підходи і технології програмування промислових систем автоматизації;
- дослідити структури класичного ППЛК і ПЛІС-контролера паралельної дії;
- виконати дослідження технології програмування ПЛК паралельної дії;
- розробити TVP-технологію для трансляції технологічної циклограми у код ПЛІС-контролера.

2 ТЕХНОЛОГІЯ ПОБУДОВИ ПРОМИСЛОВИХ СИСТЕМ КЕРУВАННЯ З ПАРАЛЕЛЬНОЮ АРХІТЕКТУРОЮ

2.1 Структура промислового програмованого логічного контролера паралельної дії

2.1.1 Класична структура ППЛК

Істотне підвищення якості функціонування обчислювальних і керуючих пристроїв і систем може бути досягнуто за рахунок використання паралельних архітектур на основі використання як елементної бази регулярних структур програмованих логічних інтегральних схем (ПЛІС), так і спеціальних інструментальних засобів для їх програмування.

В основу методології побудови паралельних керуючих автоматів покладено раціональне поєднання і практичне використання властивостей регулярності, властиве об'єктам управління дискретної циклічної дії, технологічним мовам опису алгоритмів управління такими об'єктами, і матричним мікроелектронним структурам. На рисунку 2.1 показано структуру класичного ППЛК, яка була запропонована академіком І. О. Фурманом у 1979 році [2, 13]. Цей ППЛК забезпечує програмно-логічне керування технологічним обладнанням та іншими об'єктами дискретної циклічної дії як з детермінованою так і з випадковою послідовністю виконуваних операцій. Саме на базі цієї структури було реалізовано зразки промислових ПЛК, що використовували принципи паралельної архітектури: ПЛ-1, БЛ1-8, БЛ2-1. У рамках даної кваліфікаційної роботи не розглядаються принципи описання об'єктів дискретної циклічної дії математична модель логічних керуючих автоматів паралельної дії, але основні позначення, поняття і визначення буде використано. З детальним описом функціонування ППЛК паралельної дії можна ознайомитись у роботі [13].

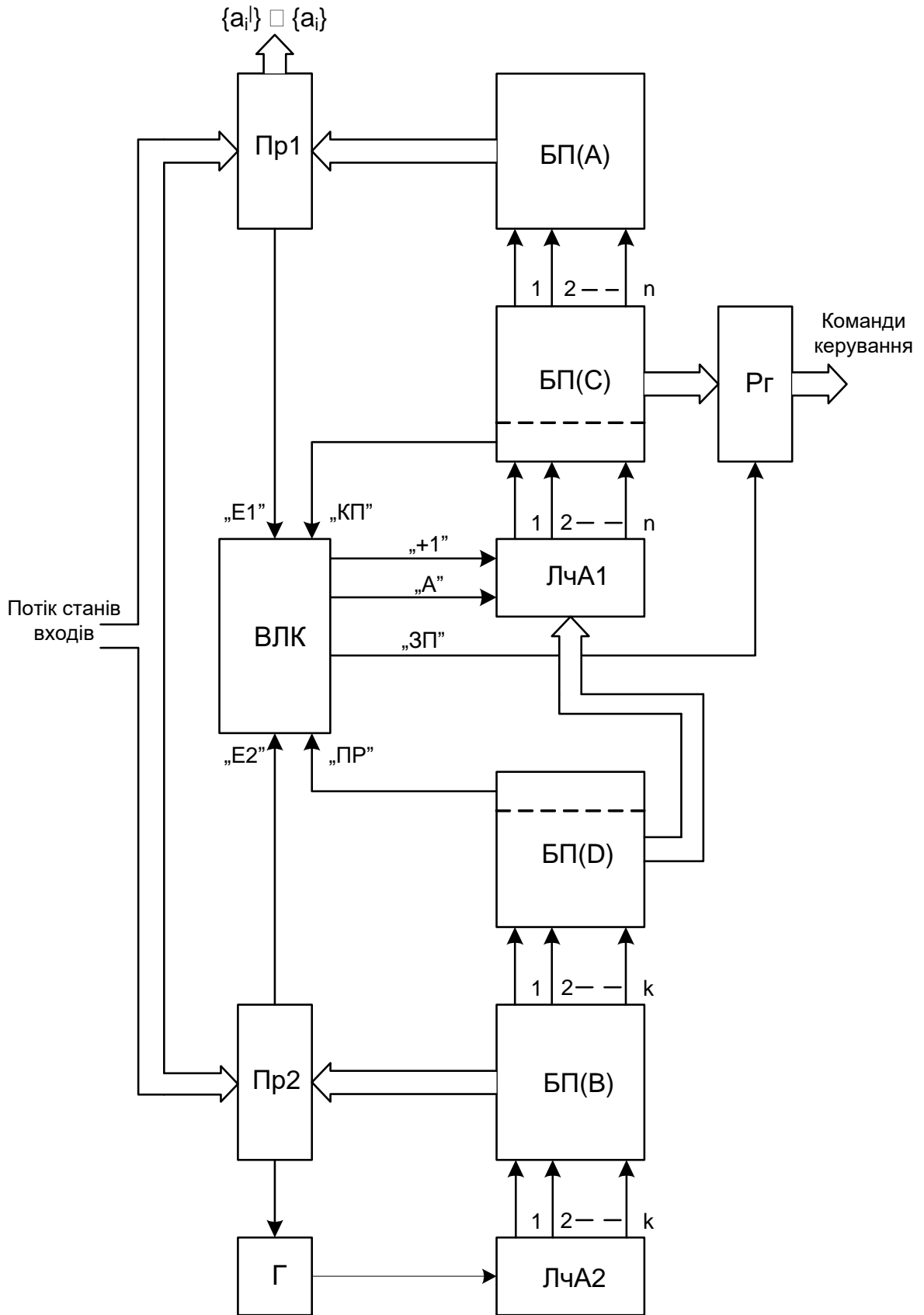


Рисунок 2.1 – Структура класичного ППЛК

В основі ППЛК (рисунок 2.1) містяться чотири блоки пам'яті (БП). Призначення блоків пам'яті наступне. В БП(А) записується матриця А очікуваних станів керованого об'єкту, до блоку БП(В) – матриця Б очікуваних станів зовнішнього середовища, у блок пам'яті записується БП(С) – матриця С команд керування, а до блоку БП(Д) – матриця D адрес переходів. Також у структурі ППЛК присутній також генератор (Г) та лічильники адрес (ЛчА1 – адреси підпрограми і ЛчА2 – переходів), схеми порівняння (Пр1, Пр2), вихідний регістр (Рг) і вузол логічного керування (ВЛК).

2.1.2 Сучасна структура ПЛІС-контролерів паралельної дії

Досвід практичного використання промислових зразків класичних ППЛК ПЛ-1, БЛ1-8, БЛ2-1, а також аналіз класичної структури показав, що вона має суттєві недоліки: в ній не врахована можливість автоматичного виявлення і заборони видачі аварійних комбінацій вихідних сигналів; крім того, обмежено можливості логічного аналізу потоку станів входів. Зазначені недоліки враховані при розробці сучасної архітектури ПЛІС-контролерів паралельної дії [6], структуру якого показано на рисунку 2.2.

ПЛІС-контролер складається з наступних блоків: блоку індикації – БІ; схеми порівняння – СП; блоку вибору операції – БВО; блоку логічного керування – БЛК; лічильника адреси – ЛА; вихідного регістру – ВР; а також блоків пам'яті станів, команд, переходів та заборонених станів – БПС, БПК, БПП, БПЗК. Основною важливою відмінністю від попередньої архітектури є наявність блоків вибору операції та пам'яті заборонених комбінацій. Основним призначенням вищевказаних блоків є наступне. блок вибору операцій призначений для вибору виконуваної операції над значеннями від датчиків: якщо на певному кроці керуючої програми необхідно порівнювати фактичний стан всіх датчиків циклу з їх очікуваними значеннями, то в останній стовпець і-го рядка, записаної в блок пам'яті станів, записується «0»

і блок вибору операції формує сигнал «=1»; якщо на i -му кроці виконання керуючої програми для переходу до наступного кроку циклу достатньо наявності сигналу хоча б від одного датчика серед безлічі датчиків, спрацьовування яких можна очікувати на i -му рядку, то в останній стовпець i -го рядка, записаної в БПС записується «1» і блок вибору операції формує сигнал «АБО=1», що перемикає схему порівняння на реалізацію логічної операції «АБО».

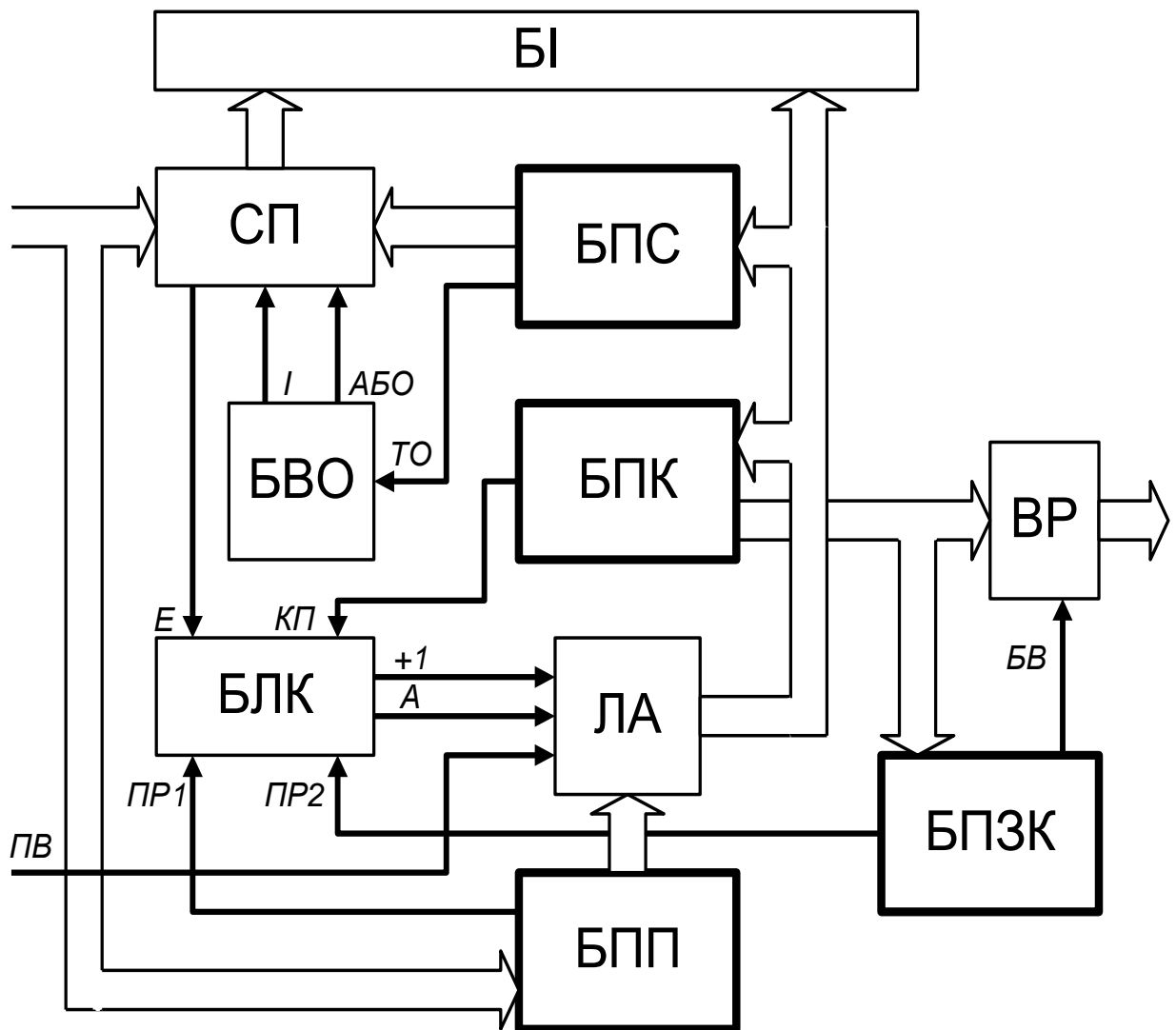


Рисунок 2.2 – Структура ПЛІС-контролера паралельної дії

БПЗК призначений для зберігання інформації про заборонені комбінації команд керування, що можуть бути формовані у результаті

відмови у роботі контролера і поява яких на його виході може призвести до аварійних ситуацій на керованому об'єкті. Саме на базі розглянутої структури створено промисловий зразок ПЛІС-контролера паралельної дії (рисунок 2.3), що має розширені функціональні можливості і може розглядатись, як технічна база для реалізації TVP-технологій програмування.

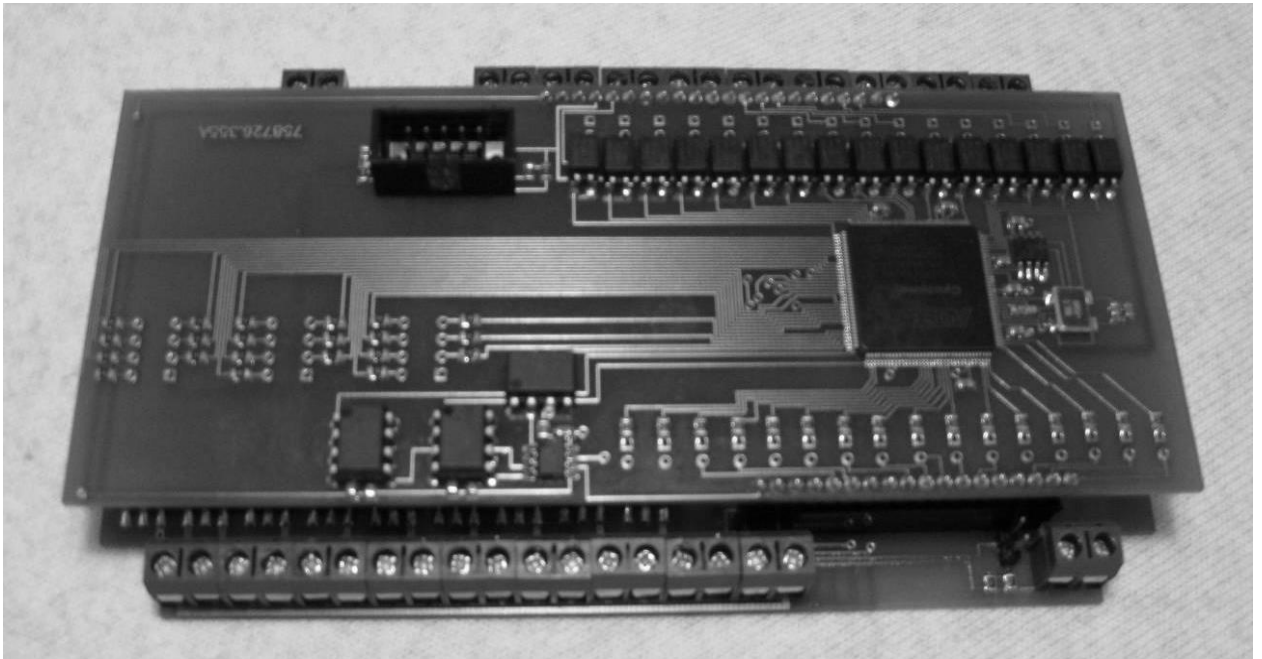


Рисунок 2.3 – Промисловий зразок ПЛІС-контролера паралельної дії

2.2 Технологія програмування ПЛК паралельної дії

У [13] описана мова програмування ПЛК паралельної дії – ЯПЛК, що призначена для програмування ПЛК паралельної дії першого покоління, структуру якого показано на рисунку 2.1. Але ця мова не враховує архітектурних особливостей та нових функціональних можливостей ПЛІС-контролерів, як паралельних ПЛК нового покоління. У той же час мова програмування ЯПЛК-М [2] (що фактично є циклограмою з розширеними можливостями) враховує особливості архітектури та функціональні можливості нового покоління ПЛІС-контролерів. Вона дозволяє виключити

проміжні етапи між складанням технологічної циклограми та записом програми на згадку про контролера, тобто. мови, в якій відсутні відмінності між вхідною мовою користувача та машинною мовою контролера. Саме мову ЯПЛК-М можна вважати базовою для формування ПЗ ПЛІС-контролерів паралельної дії на базі TVP-технології.

Розглянемо у скороченому вигляді основні положення, що становлять концепцію мови програмування ПЛІС-контролерів паралельної дії – ЯПЛК-М. Керуюча програма, записана мовою ЯПЛК-М, являє собою два, виконаних за певними правилами формалізованих (табличних) записи:

- таблиця 1, що складається з двох частин: ліва описує можливі стани керованого об'єкта та зовнішнього середовища, а також заборонені комбінації управляючих сигналів контролера, поява яких на виході може призвести до аварійних ситуацій на об'єкті управління; та права, яка однозначно визначає початкові адреси керуючих та аварійних підпрограм, а також ознаки переривань, що визначають перехід до виконання цих аварійних підпрограм або до аварійного зупинення роботи контролера у разі появи забороненої комбінації вихідних сигналів;

- таблиця 2, що являє собою сукупність технологічних та аварійних підпрограм, що підлягають реалізації залежно від поточних значень змінних таблиці 1 (тобто станів керованого об'єкта та зовнішнього середовища). Таблиця 2 також складається з двох частин: у лівій частині записуються очікувані стани сигналів від датчиків детермінованих входів та тип логічної операції, що виконується над зазначеними сигналами для виконання переходу до наступного рядка підпрограми; у правій частині таблиці вказуються команди управління технологічним обладнанням, які видаються контролером на поточному етапі виконання підпрограми. У кожному рядку таблиці 2 вказується адреса, що відповідає певному етапу виконання технологічної операції або відпрацювання аварійної підпрограми.

Програми, що записуються в таблиці 1 і 2, складаються з двох частин – описової та операторної:

- в описовій частині таблиці 1 здійснюється прив'язка елементів програми до об'єкта управління, а також наводиться опис комбінацій станів датчиків та команд управління, що визначають адреси технологічних підпрограм, аварійних підпрограм та заборонених комбінацій управляючих сигналів; в операторній частині здійснюється запис станів датчиків стохастичних входів (умов) та команд управління, що призводять до формування внутрішніх змінних – адрес переходів, ознаки переходу ПР1 та ознаки забороненої комбінації ПР2;

- в описовій частині таблиці 2 також здійснюється прив'язка елементів програми до об'єкта управління та наводяться найменування технологічних операцій або груп операцій (мікроциклів) та аварійних підпрограм; в операторній частині здійснюється формалізований запис циклограми роботи технологічного обладнання.

Операторна частина програми таблиці 1 являє собою послідовність з $P+Q$ рядків. При цьому число рядків P відповідає кількості технологічних і аварійних підпрограм, а число рядків Q – кількості заборонених комбінацій керуючих сигналів. Операторна частина програми таблиці 2 являє собою послідовність N рядків, кожна з яких відповідає певному етапу виконання технологічної операції або аварійної підпрограми і являє собою найпростіший оператор.

Складання програми мовою ЯПЛК-М полягає наступному – для кожної вхідної, вихідної та внутрішньої змінних на кожному рядку програми ставиться у відповідність їх стан та дія над ними. При цьому для кожної змінної вказується стан датчиків (включений, відключений і несуттєвий для даного рядка), команд управління (включити, відключити, не впливати) та значення внутрішніх сигналів, а саме: а) для таблиці 1 – станам датчиків стохастичних входів та команд управління (вхідні змінні), ставляться у відповідність значення адрес переходів, а також сигнали ознаки переходу ПР1 та ознаки забороненої комбінації ПР2 (внутрішні змінні); б) для таблиці 2 – значення адреси (внутрішня змінна) ставляться у відповідність стану

датчиків детермінованих входів (вхідні змінні), типу логічної операції ТО (внутрішня змінна) та стану команд управління (вихідні змінні).

Приклад готової програми мовою ЯПЛК-М показано на рисунках 2.4 і 2.5 (табличні записи 1 і 2 відповідно).

Таблиця 1

Найменування комбінацій станів датчиків і команд керування, що визначають початкові адреси технологічних і аварійних підпрограм, а також заборонені комбінації керуючих сигналів	Стани датчиків (стохастичних входів)																		Команди керування										
	Результат "Алгоритма"	Кнопка "Аварійна зупинка"	SQ11 Завантаження позиції P1-1	SQ12 Завантаження позиції P1-2	SQ14 Вильна позиція Зав2-1	SQ15 Вильна позиція Зав2-2	SQ16 Вильна позиція Зав2-3	SQ1 PR над P1-1	SQ2 PR над P1-2	SQ8 PR над Зав2-1	SQ9 PR над Зав2-2	SQ10 PR над Зав2-3	SQ17 Вильна позиція PR	SQ5 Схват вправо	PR рухається вправо	PR рухається вліво	PR рухається вправо	PR рухається вліво	Виск. списання вправо	Виск. списання вліво	Виск. промоду PR вправо	Виск. промоду PR вліво	Виск. промоду підняття схвату	Виск. промоду опускання схвату	Виск. ротикування схвату	Виск. списання схвату	Ознака переходу PR1	Адреси переходів	Ознака забороненої комбінації PR2
Підпрограма 1: перевантаження з P1-1 (SQ11) на Зав2-1(SQ14)	10	10	10																							0	7	0	
Підпрограма 2: перевантаження з P1-1 (SQ11) на Зав2-2(SQ15)	10	10		10																							0	18	0
Підпрограма 3: перевантаження з P1-1 (SQ11) на Зав2-3(SQ16)	10	10			10															11							0	29	0
Підпрограма 4: перевантаження з P1-2 (SQ12) на Зав2-1(SQ14)	10		10	10																							0	40	0
Підпрограма 5: перевантаження з P1-2 (SQ12) на Зав2-2(SQ15)	10		10	10	10																						0	51	0
Підпрограма 6: перевантаження з P1-2 (SQ12) на Зав2-3(SQ16)	10		10		10															11							0	62	0
Підпрограма 7: повернення PR у вихідне положення з P1-1	10					10																					0	1	0
Підпрограма 8: повернення PR у вихідне положення з P1-2	10						10																				0	1	0
Підпрограма 9: повернення PR у вихідне положення із Зав2-1	10							10																			0	3	0
Підпрограма 10: повернення PR у вихідне положення із Зав2-2	10								10																		0	3	0
Підпрограма 11: повернення PR у вихідне положення із Зав2-3	10									10																	0	3	0
Аварійна ситуація 1: схват утисну, PR рухається вправо	10												10	10													1	73	0
Аварійна ситуація 2: схват утисну, PR рухається вліво	10												10		10												1	73	0
Аварійна ситуація 3: зустрічний рух PR і PRn+1	10													10	10												1	73	0
Аварійна ситуація 4: зустрічний рух PR і PRn-1	10														10	10											1	73	0
Аварійна ситуація 5: аварійна зупинка обладнання	01	10																									1	73	0
Заборонена комбінація 1: одночасна видача команд на вмигання приводу PR вправо і вліво																					10	10					0		1
Заборонена комбінація 2: одночасна видача команд на вмигання приводу підняття і опускання схвату																						10	10				0		1
Заборонена комбінація 3: одночасна видача команд на вмигання списання і ротикування схвату																							10	10			0		1

Рисунок 2.4 – Табличний запис 1 мовою ЯПЛК-М (таблиця адрес переходів)

Представлені фрагменти є частиною програмного забезпечення промислової лінії управління промисловим роботом, аналіз яких показує, що по суті програма має форму таблиць, заповнення заголовків яких використовується описова частина мови, а записи вмісту рядків таблиць – операторна частина.

Таким чином, мова програмування ЯПЛК-М, яка по суті є слабо формалізованою, близькою до природної версією технологічної мови опису поведінки об'єктів дискретної дії є простим і зручним інструментом для програмування ПЛІС-контролерів нового покоління.

3 TVP-ТЕХНОЛОГІЯ СТВОРЕННЯ КЕРУЮЧИХ ПРОГРАМ ДЛЯ СИСТЕМ КЕРУВАННЯ З ПАРАЛЕЛЬНОЮ АРХІТЕКТУРОЮ

3.1 Технологія автоматичної трансляції технологічної циклограми у програмний код ПЛК

Ефективність практичного застосування ПЛК значною мірою залежить від використовуваних методів і засобів автоматизації їхнього програмування, для чого створені і широко застосовуються на практиці різні високопродуктивні інструментальні засоби. Технологію програмування, що розглянуто у цьому розділі, можна розглядати як додатковий засіб автоматизації процесу програмування класичних ПЛК послідовної дії, що може бути застосованим разом з відомими інструментальними засобами.

Ідея підходу, виникла в результаті накопиченого досвіду практичного використання для ПЛК паралельної дії – ППЛК, як мови програмування технологічних такто- і циклограм у її явному виді.

Проста, наочна, легка для сприйняття користувачами промислового устаткування мова технологічних циклограм у її явному виді не може бути використана для безпосереднього програмування ПЛК із традиційною архітектурою (послідовних ПЛК), тобто тих, котрі складають абсолютну більшість застосовуваних у промисловості, у зв'язку з чим і виникла ідея розробки програми-транслятора, що дозволяє автоматично перетворювати технологічну циклограму, побудовану на екрані монітора персонального комп'ютера, у готову керуючу програму на мові конкретного ПЛК і таким чином реалізувати пряме технологічне візуальне програмування ПЛК, або TVP (Tehnolodgy Vigiel Programming).

Програма TVP надає користувачу віконний інтерфейс у виді 4-х вікон: основне вікно, вікно прив'язки входів-виходів контролера до датчиків органів керування і виконавчих механізмів керуваного об'єкта (вікно вх/вих), вікно

введення і редагування технологічної циклограми роботи керуемого об'єкта (вікно циклограми), вікно відтрансльованої керуючої програми.

В основному вікні (рисунок 3.1) розміщені елементи керування програмою TVP: кнопки створення, відкриття і збереження циклограми, кнопки редагування таблиці вх/вих і циклограми, кнопка перегляду відтрансльованої програми.

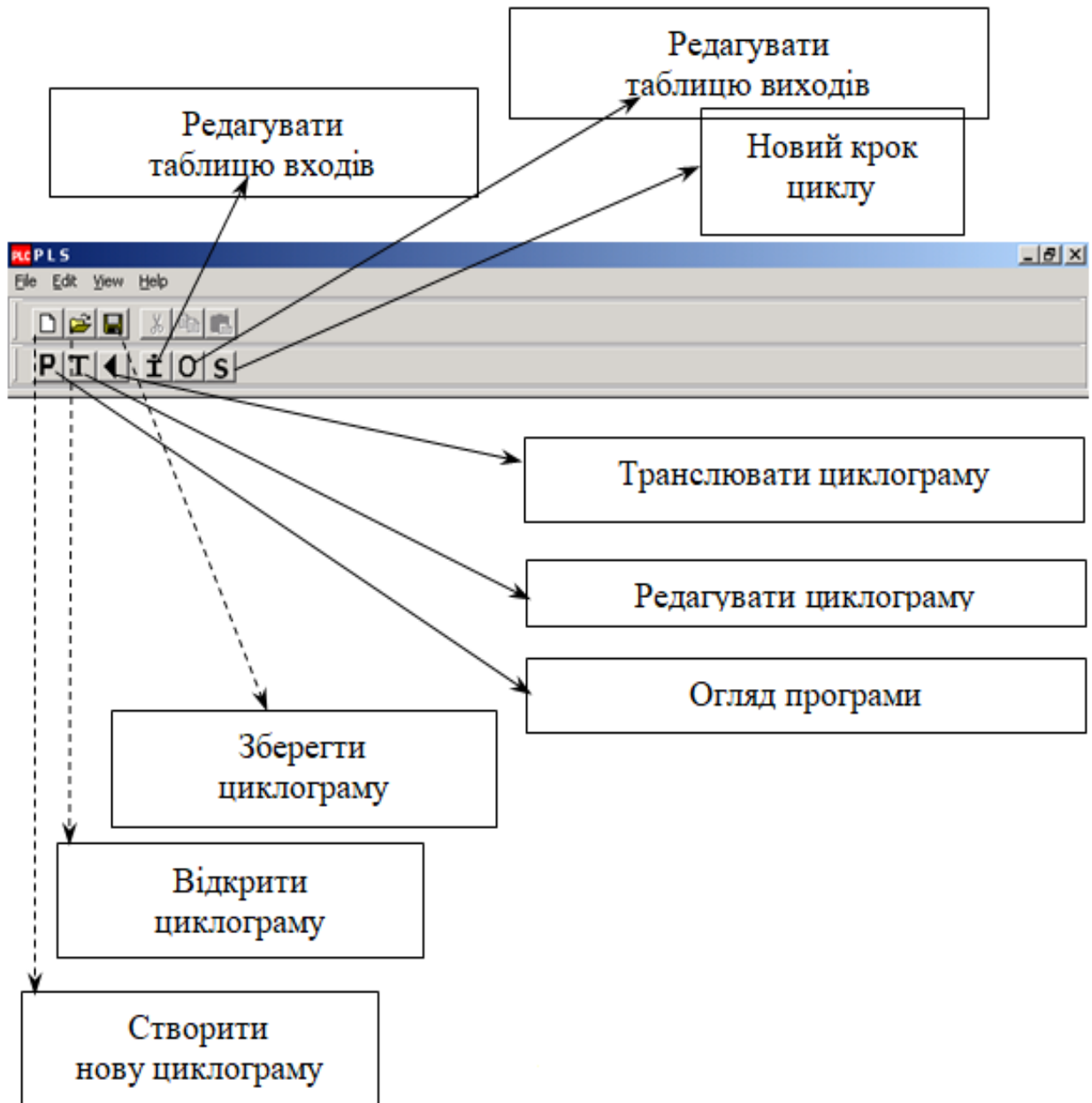


Рисунок 3.1 – Основне вікно транслятора TVP

Технологія програмування ПЛК як правило містить у собі 3 етапи: прив'язка вх/вих, складання логічних рівнянь, написання тексту програми мовою обраного контролера. Технологія застосування TVP передбачає аналогічну послідовність дій, тільки замість складання логічних рівнянь заповнюється таблиця-циклограма, а текст керуючої програми формується автоматично засобами TVP.

Вікно вх/вих (рисунок 3.2) служить для введення позначень і найменувань контролюємих датчиків, органів керування, станів виходів таймерів і лічильників, а також керуємих виконавчих механізмів. Елементи, що вводяться та й ті, що видаляються, відображаються у верхньому рядку таблиці.

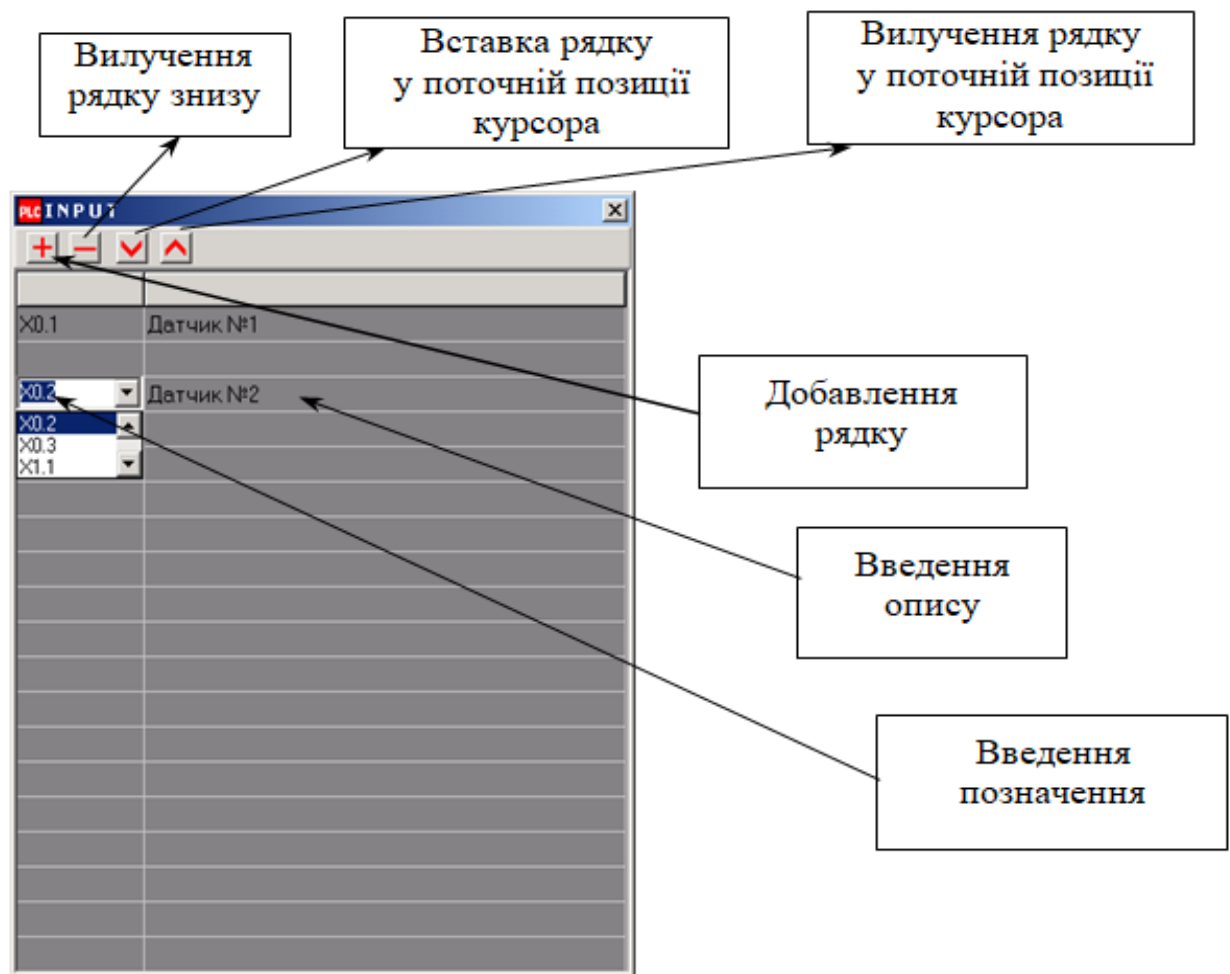


Рисунок 3.2 – Вікно прив'язки входів/виходів контролера

Вікно циклограми (рисунок 3.3) являє собою таблицю, у верхньому рядку якої записуються позначення датчиків, органів керування, виконавчих механізмів, таймерів і лічильників, а в першому стовпці вказуються такти (кроки) технологічного циклу.

Позначення пристроїв (вх/вих ПЛК)

step	X0.1	X0.2	X0.3	X0.4	X0.5	X0.6	X1.1	X1.2	X1.3	X1.4	T22	C16	Y0.1	Y0.2
1	0	0	1	0	0									
2	1	0		0	0		1				1			1
3	1	0		0	0		0	1					1	
4	1	0		0	0	1		0		1		1	0	
5	1			1	0	1				1			0	0

Номери кроків

Виконавчий код

Рисунок 3.3 – Вікно циклограми

Процес заповнення (кодування) таблиці як у частині контролюємих входів ПЛК, так і керуємих його виходів здійснюється шляхом переміщення покажчика поточної комірки і уведенням відповідно «1» чи «0» за допомогою миші або клавіатури. Причому, «1» для входів означає включений (збуджений) стан датчика, одиничний стан виходу таймера або лічильника, а «0» – відповідно – відключений стан датчика. «1» для виходів означає команду на включення механізму, а «0» – на відключення. Завдання нового кроку циклу здійснюється за допомогою кнопки «S» в основному вікні.

Зміна складу датчиків і виконавчих механізмів, використовуваних для реалізації технологічного процесу, здійснюється за допомогою редактора

вх/вих. Програма TVP забезпечує також автоматичне редагування відтрансльованої програми при внесенні змін у раніше складену циклограму.

Якщо на n наступних один за одним кроках циклу сукупність керуючих команд залишається незмінною, то TVP при трансляції циклограми в керуючу програму автоматично реалізує функцію «АБО» для відповідних n векторів станів входів ПЛК.

Запуск трансляції циклограми в керуючу програму здійснюється натисканням в основному вікні кнопки із символом «◀». Для виведення на екран монітора відтрансльованої програми (рисунок 3.4) необхідно натиснути кнопку в основному вікні із символом «P».

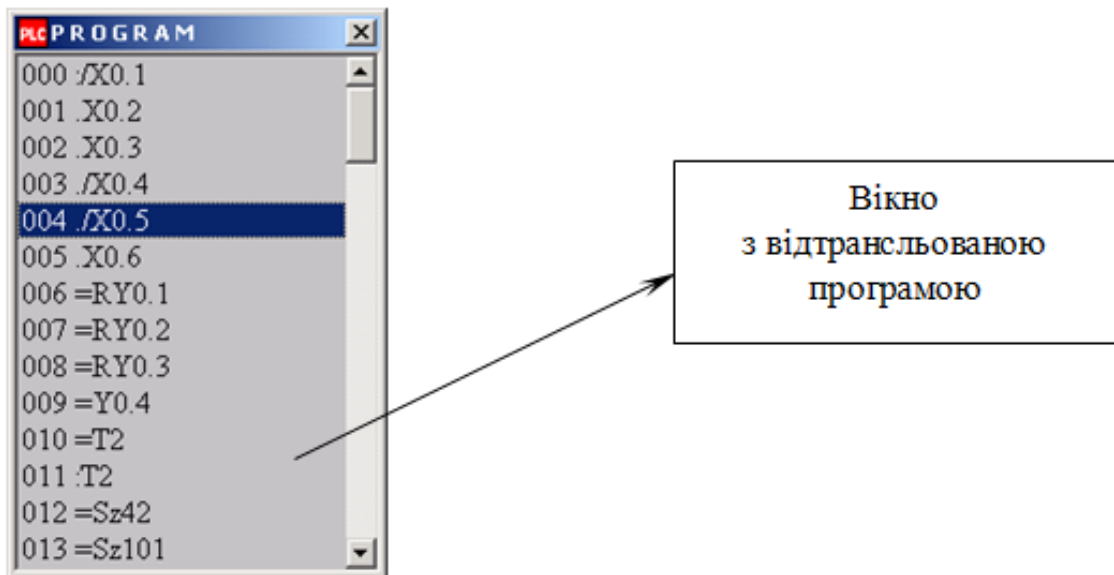


Рисунок 3.4 – Вікно з результатом роботи транслятора TVP

3.2 Розробка TVP-технології для трансляції технологічної циклограми у код ПЛІС-контролера

Для реалізації можливості використання циклограми (або тактограми) безпосередньо у якості мови програмування, як засобів автоматизації на базі ПЛІС-контролерів паралельної дії, так і універсальних засобів автоматизації

на базі ПЛК класичної послідовної дії, пропонується застосувати наступний загальний підхід, етапи або структуру якого показано на рисунку 3.5. З цієї структури явно видно, що жоден із етапів не передбачає участь професійного програміста, всі функції якого автоматично виконуються середовищем розробки TVP.

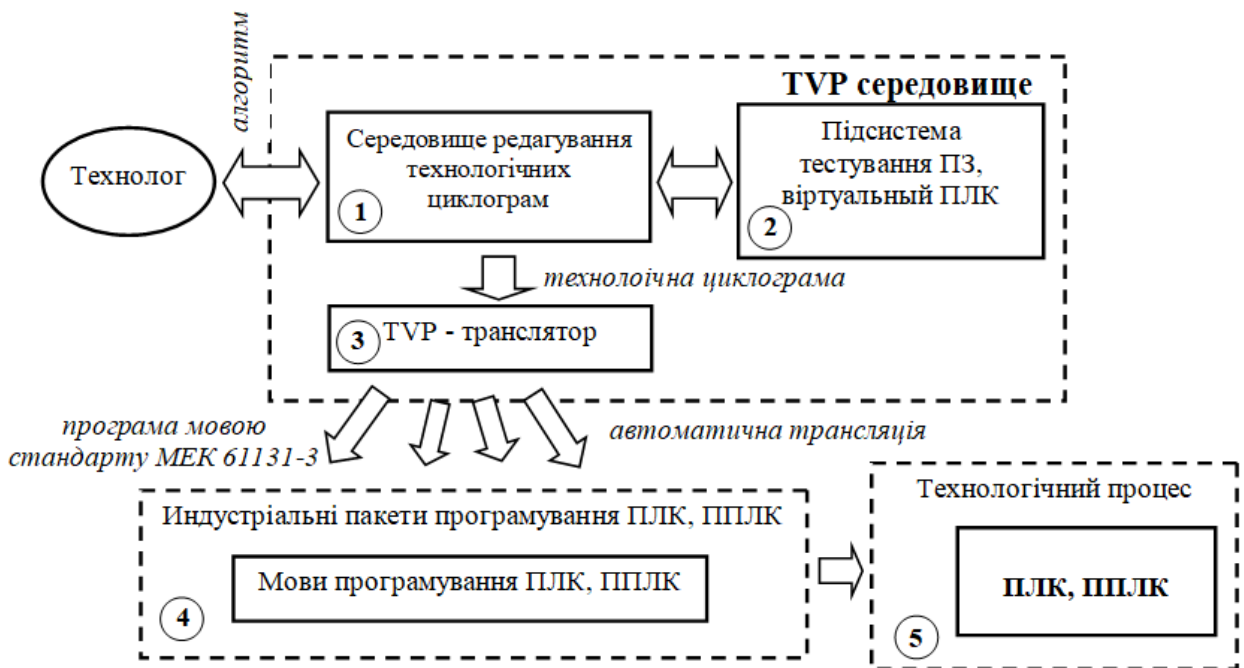


Рисунок 3.5 – Етапи розробки ПЗ у рамках TVP-технології

Процес підготовки керуючих програм у межах даної технології складається з наступних етапів, що показані на рисунку 3.5 відповідними цифрами:

- етап 1. Технолог в інтерактивному режимі відповідно до технологічного процесу формалізує алгоритм управління у вигляді технологічної циклограми (прив'язка входів-виходів, заповнення таблиць циклограми);
- етап 2. використовуючи свої знання та досвід, технолог виробляє налагодження, верифікацію формалізованого алгоритму управління;
- етап 3. За створеним та налагодженим формалізованим алгоритмом

управління у вигляді технологічної циклограми TVP-транслятор автоматично генерує керуючу програму обраною мовою програмування ПЛК або ППЛК;

- етап 4. Згенерована програма завантажується в індустріальне середовище програмування ПЛК конкретного розробника, яке адаптоване до роботи з вибраним засобом автоматизації;

- етап 5. З індустріального середовища результуюча керуюча програма завантажується у використовуваний ПЛК (ППЛК).

Нижче описується алгоритм автоматичної трансляції програми, що складена мовою ЯПЛК-М, у HDL-код, а потім у текст програми для безпосереднього завантаження до кристалу ПЛІС-контролера паралельної дії (рисунок 3.6).

Процес трансляції складається з двох етапів: аналізу та синтезу. На першому етапі виконується аналіз програми мовою ЯПЛК-М, створення та заповнення таблиць ідентифікаторів. У результаті виконання першого етапу транслятором формується код, що відповідає внутрішньому уявленню (інтерпретації) програми у вигляді набору логічних рівнянь, що описують залежність між вхідними, внутрішніми та вихідними змінними. На етапі синтезу на підставі внутрішнього представлення програми та інформації, що міститься в таблиці ідентифікаторів, формується текст результуючої програми мовою програмування апаратури.

На TVP-транслятор, що функціонує відповідно до розглянутого алгоритму, і являє собою систему програмування високого рівня, також покладаються типові для таких систем функції: пошук та аналіз синтаксичних та логічних помилок, видача попереджень та рекомендацій, можливість створення типів даних користувача, тощо.

Реалізує вказаний алгоритм інформаційна система програмування, що включає транслятор табличної циклограми стандартні текстові мови опису апаратури. Компоненти, такої системи показано на рисунку 3.7 і вона включає до себе:

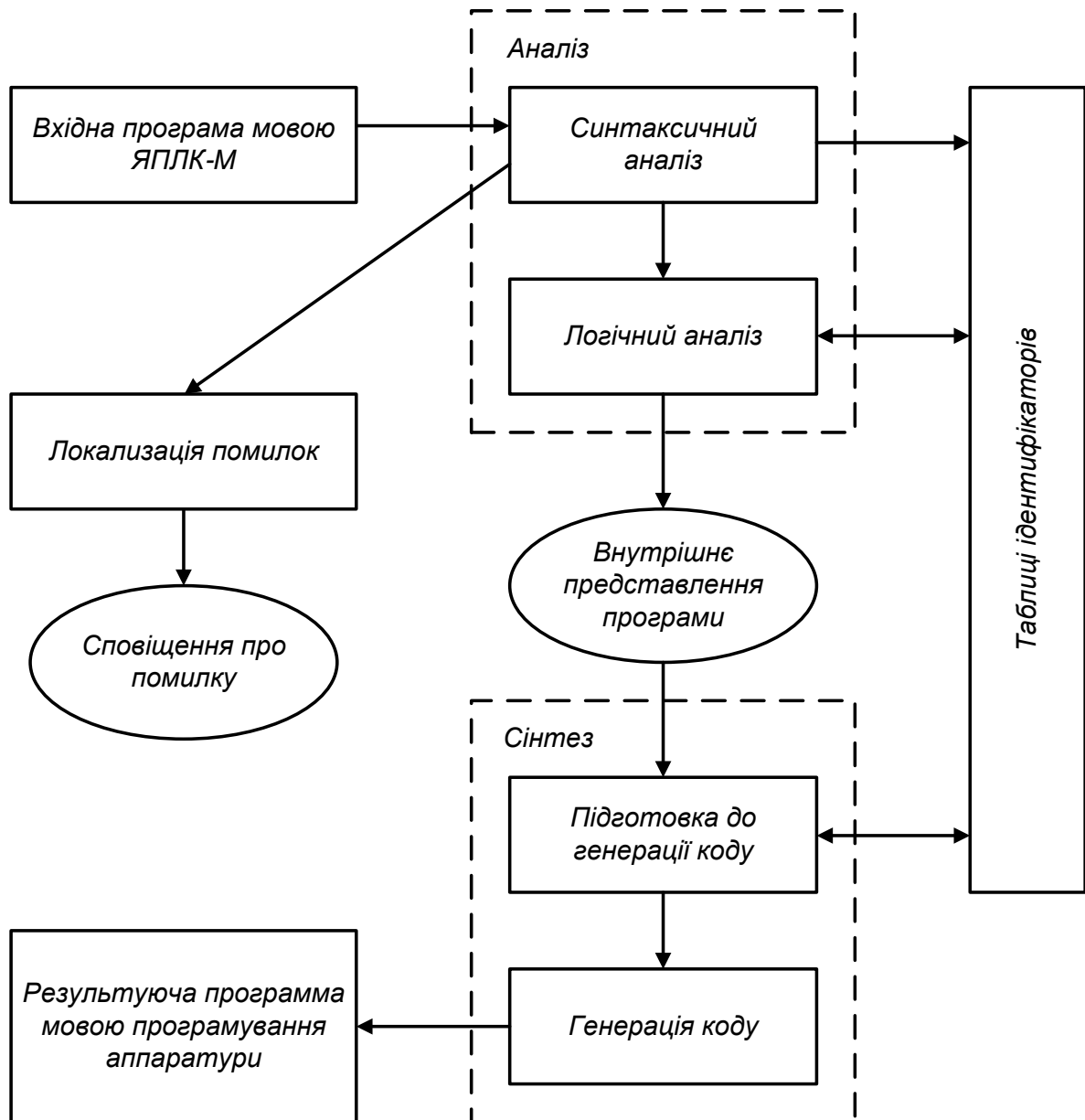


Рисунок 3.6 – Алгоритм трансляції програми мовою ЯПЛК-М у HDL-код

-підсистему редагування змінних проекту (вхідні, вихідні змінні, таймери);

- табличний процесор, що дозволяє в інтерактивному режимі редагувати технологічні та аварійні підпрограми;

- табличний процесор, що дозволяє в інтерактивному режимі редагувати переходи між підпрограмами;

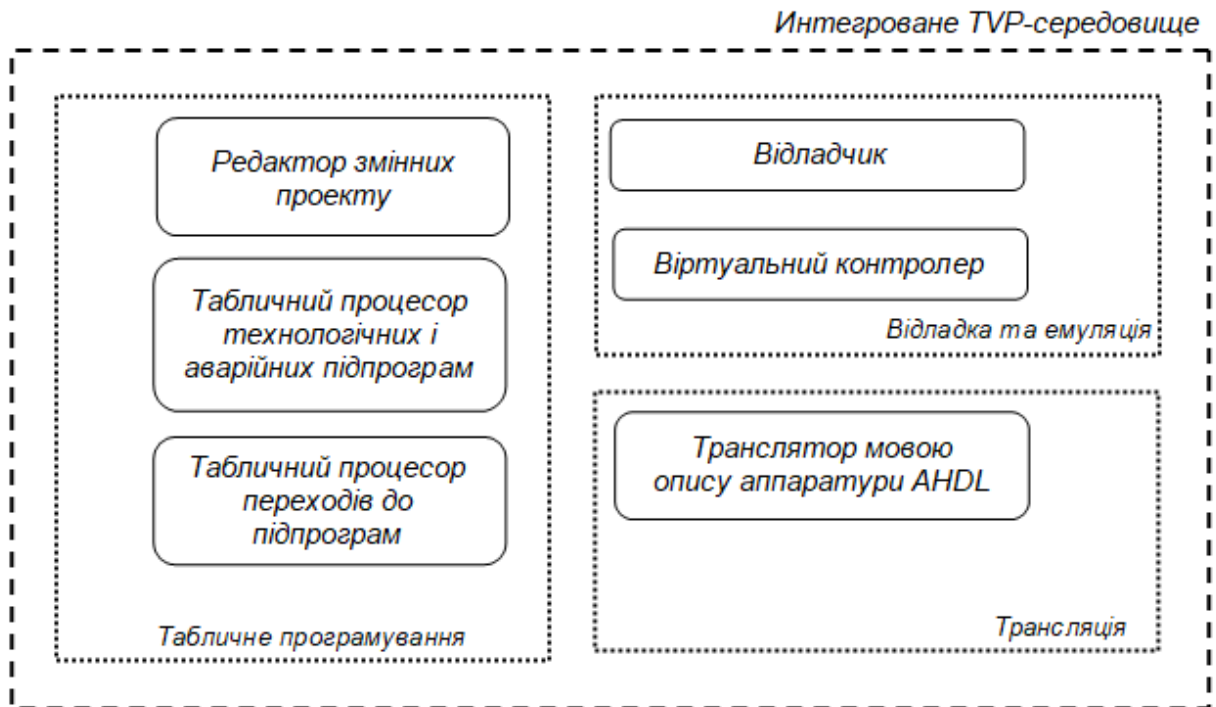


Рисунок 3.7 – Архітектура системи табличного програмування

- відладчик, що дозволяє фіксувати помилки у процесі створення програми;
- віртуальний контролер, що дозволяє емулювати за допомогою персонального комп'ютера роботу керуючого автомата відповідно до створеної програми;
- спеціалізований транслятор, що дозволяє автоматично за створеною та налагодженою програмою ЯПЛК-М генерувати програмний код мовою опису апаратури AHDL.

На рисунку 3.8 показано віконний інтерфейс середовища TVP. На ньому видно процес заповнення фрагменту технологічної циклограми у програму мовою ЯПЛК-М. Із застосуванням даного середовища програмування формується HDL-код програми для безпосереднього завантаження до кристалу ПЛІС-контролера паралельної дії.

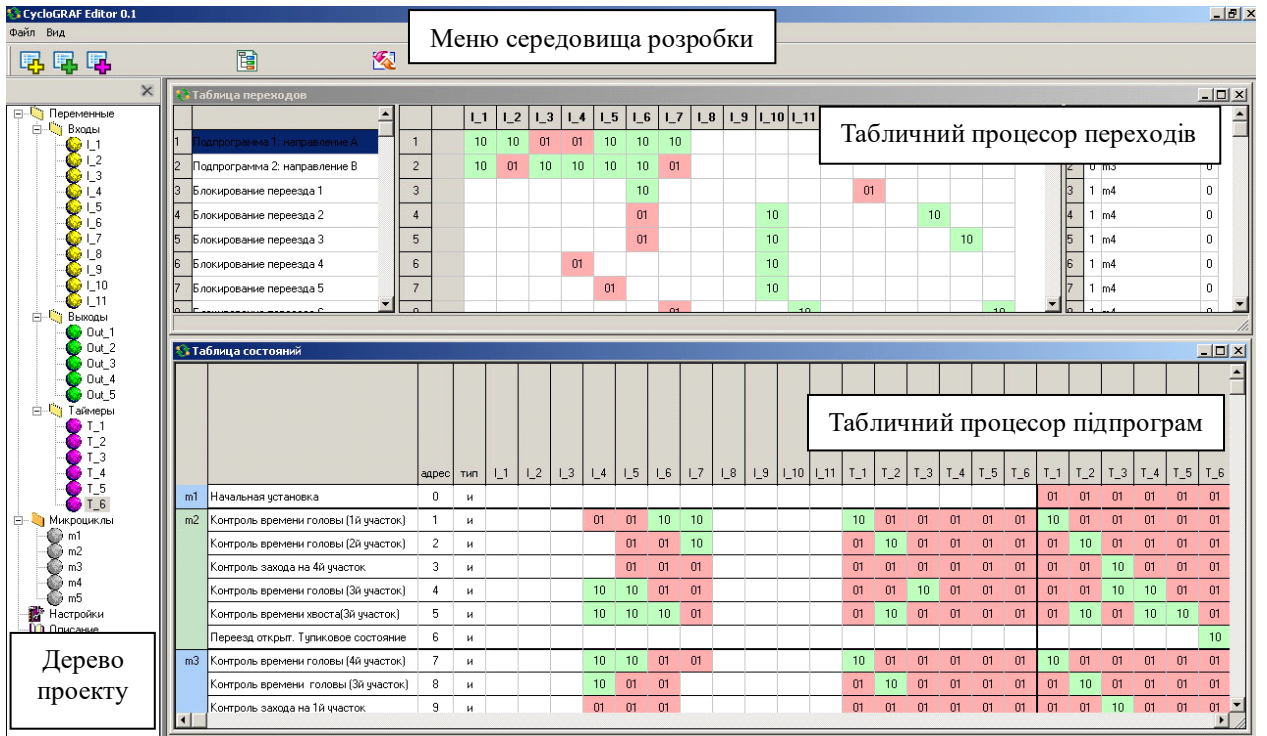


Рисунок 3.8 – Віконний інтерфейс середовища програмування TVP-технології

ВИСНОВКИ

У процесі виконання кваліфікаційної роботи було проведено аналіз засобів керування промисловим технологічним обладнанням, а також аналіз мов і технологій програмування промислових систем автоматизації. За результатами цього аналізу визначено найбільш перспективне обладнання і технології для реалізації елементів TVP-технології програмування.

Виконано дослідження технологій побудови промислових систем керування з паралельною архітектурою, а саме розглянуто архітектури класичних ППЛК і сучасних ПЛІС-контролерів паралельної дії. Розглянуто технологію програмування ПЛК паралельної дії і визначено місце у ній TVP-технології.

Розроблено елементи технології автоматичної трансляції циклограми у програмний код ПЛК і TVP-технологію трансляції технологічної циклограми мовою ЯПЛК-М у код ПЛІС-контролера.

Таким чином завдання, що поставлені у кваліфікаційній роботі успішно виконані, що дозволяє у значній мірі автоматизувати технологію програмування контролерів паралельної дії і значно зменшити ймовірність появи у програмному коді помилок.

За результатами досліджень кваліфікаційної роботи опубліковано тези доповіді на дванадцятій міжнародній науково-технічній конференції «Сучасні напрями розвитку інформаційно-комунікаційних технологій та засобів управління» 27-28 квітня 2022 року [1].

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Білик Ю. Ю. Метод створення керуючих програм на базі TVP-технології [Текст] / Ю. Ю. Білик, С. Я. Бовчалоук // СУЧАСНІ НАПРЯМИ РОЗВИТКУ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ ТА ЗАСОБІВ УПРАВЛІННЯ. Тези доповідей одинадцятої міжнародної науково-технічної конференції (27-28 квітня 2022 року), Том 1: секції 1-4. – 2022. – С. 54.
2. Бовчалоук, С. Я. Вдосконалення архітектури, мови програмування та технічної реалізації ПЛК паралельної дії [Текст] / С. Я. Бовчалоук // Збірник міжнародної науково-технічної конференції «ІКТМ-2005». – Харків: ХАІ, 2005. – С. 290.
3. Бовчалоук, С. Я. Новая информационная технология логического управления в энергетике и на транспорте [Текст] / С. Я. Бовчалоук // Системи управління, навігації та зв'язку. – К.: Центральний науково-дослідний інститут навігації і управління, 2007. – Вип. 3 – С. 47-51.
4. Бовчалоук, С. Я. Развитие архитектуры ПЛК параллельного действия: от абстрактной модели параллельного автомата, до инженерной реализации безопасного ПЛИС-контроллера [Текст] / С. Я. Бовчалоук, И. А. Фурман, М. Л. Малиновский // Энергетика та комп'ютерно-інтегровані технології в АПК. 2016. – №. 2 (5). – С. 62-66.
5. Выжигин, А. Ю. Гибкие производственные системы учеб. Пособие для вузов [Текст] / А. Ю. Выжигин. // 1. – 2009. – С. 286.
6. Зюбин, В. Е. Программирование ПЛК: языки МЭК 61131-3 и возможные альтернативы [Текст] / В. Е. Зюбин // Промышленные АСУ и контроллеры. – 2005. – No 11. – С. 31–35.
7. Мишель, Ж. Програмуємі контроллери: Архітектура і застосування: пер. с фр. [Текст] / Ж. Мишель ; Переводчик И. В. Федотов ; Ред. Б. И. Лыткин . – М. : Машиностроение, 1992 . – 320 .

8. Поспелов, Д. А. Ситуационное управление: теория и практика. [Текст] / Д. А. Поспелов – М.: Наука. – Гл. ред. физ.-мат. лит., 1986. – 288 с.
9. Стешенко, В. Б. ПЛИС фирмы ALTERA: проектирование устройств обработки сигналов. [Текст] / В. Б. Стешенко. – М.: Додэка, 2000. – 128с.
10. Соловьев, В. В. Проектирование цифровых систем на основе программируемых логических интегральных схем. [Текст] / В. В. Соловьев. – М.: Горячая линия-Телеком, 2001. – 636 с.
11. Фурман, И. А. Контроллеры и процессоры с параллельной архитектурой [Текст] / И. А. Фурман, В. А. Краснобаев, М. Л. Малиновский // Учебник для ВУЗов. – Харьков: УкрГАЖТ, 2006. – 416 с.
12. Фурман, И. А. Концепция, методы и средства моделирования на ПЛИС контроллеров и процессоров с параллельной архитектурой [Текст] / И. А. Фурман, В. А. Краснобаев, М. Л. Малиновский [и др.] // Автомобильный транспорт: Сб. научных трудов, вып. 16. – Харьков, 2005. – С. 338–341.
13. Фурман, И. А. Программируемые контроллеры для систем управления. Часть 1. Архитектура и технология применения. [Текст] / И. А. Фурман, Г. И. Загарий, Н. О. Ковзель, В. И. Поддубняк, А. И. Стасюк. – Харьков: ХФИ «Транспорт Украины»; – Харьков: Издательство «Регион-информ», 2001. – 316 с.
14. Фурман, И. А., Совершенствование математической модели и архитектуры логических управляющих автоматов параллельного действия [Текст] / И. А. Фурман, С. Я. Бовчалюк // Інформаційно-керуючі системи на залізничному транспорті. – 2006. – №3 (59). – С. 72–76.
15. Фурман, И. А. Технологическое визуальное программирование – новое средство автоматизации разработки программного обеспечения ПЛК [Текст] / И. А. Фурман, С. А. Колесников // Інформаційно- керуючі системи на залізничному транспорті. – 2003. – N 4. – С. 46–48.
16. IEC 61131-3. Programmable controllers. Part 3: Programming languages [Text]. – 2nd ed. – Int. Electrotechnic Commission, 1998.
17. Dubinin, V. Engineering of Validatable Automation Systems Based on

an Extension of UML Combined With Function Blocks of IEC 61499 [Text] / V. Dubinin, V. Vyatkin, T. Pfeiffer // Proceedings of the 2005 IEEE International Conference on Robotics and Automation. – 2005. doi: 10.1109/robot.2005.1570732.

18. Parr, T. Language Implementation Patterns. Create Your Own Domain-Specific and General Programming Languages [Text] / T. Parr. – Pragmatic Bookshelf, 2010. – 380 p.

19. Evans, E. Domain-Driven Design: Tackling the Complexity in the Heart of Software [Text] / E. Evans. – Boston, MA: Addison-Wesley, 2004. – 560 p.

20. Furman, Ilya. Development and study of technological visual programming of logic control problems [Текст] / Ilya Furman, Stanislav Bovchaliuk, Alexander Allashev, Aleksey Piskarev // Eastern-European Journal of Enterprise technologies, – 2017. – № 6/2 (90). –P. 23–31.

21. Stanislav Bovchaliuk. The Architecture of Fuzzy Logic Automat of Parallel Action for the Intelligent Smart Grid Networks / S. Bovchaliuk, S.Tymchuk, S. Shendryk, V. Shendryk // New Technologies, Development and Application III. NT 2020. Lecture Notes in Networks and Systems, vol. 128. Springer, – 2020. – P. 462–468.