

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет комп'ютерної інженерії та управління
(повна назва)

Кафедра електронних обчислювальних машин
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

Рівень вищої освіти другий (магістерський)

Модель перспективного керуючого автомату
паралельної дії

(тема)

Виконав:

здобувач 2 року навчання,
групи СПМ-23-5

Олег ДРОЗД
(власне ім'я, прізвище)

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. Станіслав БОВЧАЛЮК
(посада, власне ім'я, прізвище)

Допускається до захисту

Завідувач кафедри ЕОМ

(підпис)

Андрій КОВАЛЕНКО

(власне ім'я, прізвище)

2025 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

здобувачеві _____ Дрозду Олегу Юрійовичу _____
(прізвище, ім'я, по батькові)

1. Тема роботи _____ Модель перспективного керуючого автомату паралельної дії _____

затверджена наказом по університету від “ 21 ” квітня 2025 р. № 296 Ст

2. Термін подання здобувачем роботи до екзаменаційної комісії _____ 16 червня 2025 р.

3. Вхідні дані до роботи _____ 1) інформаційна технологія паралельного логічного керування;

_____ 2) Математична модель контролерів паралельної дії;

_____ 3) Структури ППЛК та ПЛК ПД;

_____ 4) Технічна документація щодо практичної реалізації керуючих структур з паралельною архітектурою.

4. Перелік питань, що потрібно опрацювати у роботі _____

_____ 1) Аналіз математичної моделі ЛКА ПД;

_____ 2) Аналіз відомих структур керуючих пристроїв з паралельною архітектурою:
_____ універсального ПЛК паралельної дії, класичного ЛКА ПД, АПДН, ПЛІС-контролера
_____ ПД з програмованими таймерами;

_____ 3) Розробка структури перспективного керуючого автомата паралельної дії;

_____ 4) Дослідження HDL-моделі відомих ЛКА ПД;

_____ 5) Розробка елементів HDL-моделі МЛКА ПД;

_____ 6) Висновки.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій Слайд-презентація – 13 слайдів

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

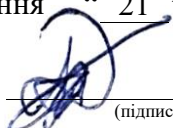
Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

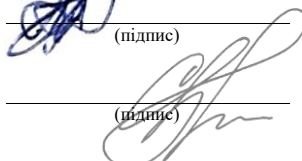
№	Назва етапів роботи	Строк / терміни виконання етапів роботи	Примітка
1	Аналіз математичної моделі ЛКА ПД	22.04.25-25.04.25	
2	Аналіз універсального ПЛК паралельної дії	26.04.25-28.04.25	
3	Дослідження класичного ЛКА ПД	29.04.25-01.05.25	
4	Аналіз особливостей АПДН	02.05.25-03.05.25	
5	Аналіз архітектури ПЛІС-контролера ПД з програмованими таймерами	04.05.25-05.05.25	
6	Розробка структури перспективного керуючого автомата паралельної дії	06.05.25-23.05.25	
7	Дослідження HDL-моделі відомих ЛКА ПД	24.05.25-27.05.25	
8	Розробка елементів HDL-моделі МЛКА ПД	28.05.25-02.06.25	
9	Оформлення матеріалів кваліфікаційної роботи	03.06.25-05.06.25	
10	Подання кваліфікаційної роботи керівникові та її попередній захист	06.06.25-09.06.25	
11	Подання кваліфікаційної роботи на рецензування	10.06.25-12.06.25	

Дата видачі завдання “ 21 ” квітня 2025 р.

Здобувач


(підпис)

Керівник роботи


(підпис)

доц. Станіслав БОВЧАЛЮК
(посада, власне ім'я, прізвище)

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 84 с., 10 рис., 2 дод., 22 джерел.

ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ПАРАЛЕЛЬНОГО ЛОГІЧНОГО КЕРУВАННЯ, КЕРУЮЧІ ПРИСТРОЇ З ПАРАЛЕЛЬНОЮ АРХІТЕКТУРОЮ, ПЛІС-КОНТРОЛЕР ПАРАЛЕЛЬНОЇ ДІЇ, СТРУКТУРА ЛКА ПД.

Метою кваліфікаційної роботи є розробка «гнучкої» або «модульної» архітектури керуючих автоматів паралельної дії та створення підґрунтя для побудови математичної та HDL-моделей таких автоматів.

У ході виконання кваліфікаційної роботи було проведено аналіз відомих архітектур та математичної моделі керуючих автоматів паралельної дії: розглянуто елементи математичної моделі класичного ЛКА ПД, наведено та досліджено відомі на даний час структури керуючих пристроїв з паралельною архітектурою. Виконано розробку структури перспективного ЛКА ПД, що отримав назву модульний логічний керуючий автомат паралельної дії – МЛКА ПД. Досліджено HDL-моделі відомих ЛКА ПД та запропоновано варіант реалізації такої моделі для МЛКА ПД.

У результаті виконаних досліджень усунуто такий недолік керуючих пристроїв з паралельною архітектурою, як їх фактична побудова за принципом «жорсткої логіки», або недовикористання можливостей створення «гнучких» систем керування на базі сучасних кристалів ПЛІС.

Запропонована концепція «модульної архітектури» керуючих пристроїв паралельної дії дозволяє створювати «гнучку» структуру логічних керуючих автоматів. Саме на базі цієї концепції з'являється можливість синтезувати структури МЛКА із заданою функціональністю у залежності від конкретних задач системи керування, що будується на його основі.

ABSTRACT

Master's thesis: 84 pages, 10 figures, 2 appendices, 22 sources.

INFORMATION TECHNOLOGY OF PARALLEL LOGIC CONTROL,
CONTROL DEVICES WITH PARALLEL ARCHITECTURE, CPLD
CONTROLLER OF PARALLEL ACTION, STRUCTURE OF PLC PD.

The purpose of the qualification work is to develop a “flexible” or “modular” architecture of parallel-action control automata and to create the basis for building mathematical and HDL-models of such automata.

In the course of the qualification work, an analysis of the known architectures and mathematical model of parallel-action control automata was carried out: elements of the mathematical model of the classical PLC of PA were considered, currently known structures of control devices with parallel architecture were presented and studied. The development of the structure of a promising PLC PA, called a modular logic control automaton of parallel action – MPLC PA, is carried out. HDL-models of known PLC PA are investigated and a variant of the implementation of such a model for MPLC PA is proposed.

As a result of the performed research, such a drawback of control devices with parallel architecture as their actual construction on the principle of “rigid logic” or underutilization of the possibilities of creating “flexible” control systems based on modern FPGA crystals has been eliminated.

The proposed concept of a “modular architecture” of parallel-acting control devices allows creating a “flexible” structure of logic control automata. It is on the basis of this concept that it becomes possible to synthesize the structure of MPLC with a given functionality, depending on the specific tasks of the control system built on its basis.

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ	7
ВСТУП	8
1 АНАЛІЗ ВІДОМИХ АРХІТЕКТУР ТА МАТЕМАТИЧНОЇ МОДЕЛІ КЕРУЮЧИХ АВТОМАТІВ ПАРАЛЕЛЬНОЇ ДІЇ.....	11
1.1 Елементи математичної моделі ЛКА ПД	11
1.2 Відомі структури керуючих пристроїв з паралельною архітектурою.....	14
1.2.1 Універсальний ПЛК паралельної дії	16
1.2.2 Сучасний ЛКА ПД	19
1.2.3 Структура АПДН	24
1.2.4 Структура ПЛІС-контролера ПД з програмованими таймерами.....	28
1.3 Постановка завдання.....	33
2 РОЗРОБКА СТРУКТУРИ ПЕРСПЕКТИВНОГО КЕРУЮЧОГО АВТОМАТА ПД.....	35
3 HDL-МОДЕЛЬ ПЕРСПЕКТИВНОГО КЕРУЮЧОГО АВТОМАТА	46
3.1 Обґрунтування реалізації ЛКА ПД на ПЛІС.....	46
3.2 Розробка HDL-моделі перспективного ЛКА ПД.....	49
3.2.1 HDL-моделі відомих ЛКА ПД	49
3.2.2 HDL-модель МЛКА ПД.....	61
ВИСНОВКИ.....	65
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	66
ДОДАТОК А Графічний матеріал кваліфікаційної роботи.....	70
ДОДАТОК Б Наукові публікації за темою кваліфікаційної роботи	78

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

- АПДН – автомат паралельної дії з реалізацією нечіткої логіки
- ВІС – велика інтегральна схема
- ІМС – інтегральна мікросхема
- ІТПЛК – інформаційна технологія паралельного логічного керування
- ЛКА – логічний керуючий автомат
- МЛКА – модульний логічний керуючий автомат
- ПД – паралельної дії
- ПЛІС – програмована логічна інтегральна схема
- ПЛК – програмований логічний контролер
- ПЛК ПД – програмований логічний контролер паралельної дії
- ППЛК – паралельний програмований логічний контролер
- СЗК – системи залишкових класів
- ТА – технологічний агрегат
- ТА_{дц} – технологічний агрегат дискретної циклічної дії
- ЧСЗ – числова система залишків
- HDL – мова програмування апаратури (англ., Hardware Description Language)
- Smart Grid – «розумні мережі», «розумна енергетика»
- RNS – числова система залишків (англ., Residue Number System)
- TVP – технологічне візуальне програмування (англ., Technological Visual Programming)

ВСТУП

Від моменту публікації структури сучасного керуючого пристрою паралельної дії, яка сьогодні вже вважається класичною [1-3], минуло майже два десятиліття. За цей час було проведено значну кількість досліджень, результатом яких стало формування інформаційної технології паралельного логічного керування (ІТПЛК) [4]. Ця технологія об'єднує в собі комплекс апаратних і програмних засобів, серед яких:

- технічні пристрої – логічні керуючі автомати паралельної дії (ЛКА ПД), програмовані логічні контролери паралельної дії (ПЛК ПД), а також ПЛІС-контролери паралельної дії;

- мови програмування – ЯППЛК та ЯПЛК-М (його розширена та модифікована версія);

- програмна технологія – TVP (Technological Visual Programming), технологія візуального програмування для автоматизованого створення програм керування.

У ряді нещодавніх публікацій, присвячених розвитку ІТПЛК, були проаналізовані основні тенденції та напрямки вдосконалення керуючих пристроїв із паралельною архітектурою – зокрема, ПЛК ПД і ПЛІС-контролерів. У роботі [5] запропоновано виокремити три основні напрямки розвитку ІТПЛК, які можна коротко сформулювати наступним чином:

- перехід до сучасної елементної бази для реалізації ПЛК ПД [6-8];
- вдосконалення мов та технологій програмування, орієнтованих на ПЛК ПД [9-11];

- розвиток внутрішньої структури логічних автоматів паралельної дії, зокрема на архітектурному рівні [1, 3, 12–14].

Для того, щоб визначитись з напрямом подальшого руху, коротко оцінимо кожен з напрямків.

Перший напрям розвитку наразі вичерпав свій потенціал. Відбувся

остаточний перехід від використання дискретних ІМС з низьким рівнем інтеграції до сучасних кристалів ПЛІС, які забезпечують високу щільність логіки, гнучкість та масштабованість. Сьогодні, на нашу думку, не існує альтернативних мікроелектронних рішень, які б перевершували ПЛІС за сукупністю характеристик для побудови систем з паралельною логікою. Єдине актуальне питання – це вибір платформи (наприклад, AMD-Xilinx чи Intel-Altera) та мови опису апаратури (VHDL, Verilog HDL, AHDL тощо).

Другий напрям є актуальним і перспективним, проте він значною мірою залежить від змін у структурі ПЛК ПД. Основна мета – адаптація технології TVP до нових функціональних можливостей контролерів. Прикладами таких змін є підтримка програмованих користувачем таймерів [14], а в майбутньому – можлива інтеграція лічильників, елементів виконання арифметичних операцій та інших функціональних блоків. Тобто розвиток мови та інтерфейсу програмування тісно пов'язаний з архітектурним удосконаленням керуючих пристроїв.

Третій напрям – розвиток внутрішньої логіки ЛКА ПД. Він є основним рушієм усієї технології ІТПЛК. Саме він визначає необхідність змін у мовах програмування, структурі програм, а також способах формування керуючих алгоритмів. Будь-яке розширення функціональних можливостей контролера (наприклад, підтримка нечіткої логіки, таймерів) потребує змін на рівні внутрішньої організації ЛКА ПД, що, у свою чергу, впливає на всі інші складові системи.

Таким чином, саме третя із зазначених траєкторій розвитку – тобто вдосконалення внутрішньої організації ЛКА ПД є найперспективнішим напрямом у контексті подальшого розвитку інформаційної технології паралельного логічного керування. Вона має потенціал забезпечити гнучкість, масштабованість та адаптивність сучасних керуючих систем.

Однак, попри свою значущість, цей напрямок демонструє ознаки стагнації. Як зазначено у [14]: «аналіз моделей і структур програмованих керуючих автоматів паралельної дії свідчить про їх «негнучкість»,

«закритість» або, фактично, реалізацію за допомогою концепції «жорсткої логіки». Іншими словами, підходи, які використовуються для побудови ПЛК ПД, у значній мірі залишаються консервативними та орієнтованими на фіксовані, незмінні архітектури, що обмежує їхню адаптивність до нових вимог та умов експлуатації. Впровадження додаткового функціоналу, такого як програмовані таймери, лічильники, або навіть арифметичні блоки, безумовно, розширює можливості ЛКА ПД. Проте це не усуває головну проблему, яка полягає у жорсткій архітектурній фіксованості, що була успадкована ще з епохи побудови ПЛК на основі дискретних логічних елементів. У сучасних умовах, коли контролери реалізуються на програмованих логічних інтегральних схемах, зберігання фіксованої структури контролера не дає змоги повною мірою використати переваги ПЛІС-технологій. Зокрема, не реалізується їх головна перевага – архітектурна гнучкість, яка дозволяє створювати індивідуалізовані, спеціалізовані структури контролера, орієнтовані на конкретні прикладні задачі. Натомість, у більшості реалізацій спостерігається лише адаптація жорстких, уніфікованих структур та розробка програмного забезпечення під них, що знижує ефективність, ускладнює масштабування та не сприяє розвитку інженерних інновацій.

Отже, подальший прогрес у розвитку ІТПЛК напряму залежить від відходу від концепції «жорсткої логіки» і переходу до побудови відкритих, конфігурованих архітектур контролерів, що повністю використовують потенціал сучасних ПЛІС для створення адаптивних, ефективних і цілеспрямованих керуючих систем.

1 АНАЛІЗ ВІДОМИХ АРХІТЕКТУР ТА МАТЕМАТИЧНОЇ МОДЕЛІ КЕРУЮЧИХ АВТОМАТІВ ПАРАЛЕЛЬНОЇ ДІЇ

1.1 Елементи математичної моделі ЛКА ПД

Щоб чітко усвідомити відмінності між керуючими автоматами послідовної та паралельної дії, слід звернути увагу на основний принцип, який дозволяє однозначно розмежовувати ці типи пристроїв. Для цього розглянемо математичну модель керуючих автоматів паралельної дії у скороченому вигляді.

У загальному випадку як послідовні, так і паралельні ПЛК можна охарактеризувати через тривалість одного циклу обслуговування всіх контрольованих входів і керованих виходів. Цей цикл визначається як:

$$T_u = t_I + t_{II}, \quad (1.1)$$

де T_u – тривалість циклу однократного обслуговування всіх входів-виходів контролера в одиницях дискретного автоматного часу;

t_I – дискретний автоматний час, що витрачається на аналіз станів входів;

t_{II} – дискретний автоматний час, формування команд керування.

Під послідовними ПЛК розуміються контролери, що працюють за принципом послідовного аналізу входів і послідовного формування вихідних сигналів. У цьому випадку загальний час циклу обслуговування пропорційно залежить від кількості оброблюваних сигналів і визначається так:

$$t^I = \sum_{i=1}^k t_i^I, \quad t^{II} = \sum_{j=1}^m t_j^{II}, \quad (1.2)$$

де k – кількість контрольованих входів контролера;
 m – кількість команд керування виконавчими механізмами.

Паралельними ПЛК будемо називати контролери, в яких використовується паралельний принцип обслуговування входів-виходів. Для таких контролерів час визначається наступними виразами:

$$t^I = t_i^I (i=1, 2, \dots, k), \quad t^{II} = t_j^{II} (j=1, 2, \dots, m). \quad (1.3)$$

З урахуванням наведеної вище диференціації, розглянемо математичну модель класичного ЛКА ПД, який на практиці реалізує як паралельну обробку вхідної інформації, так і паралельне формування керувальних команд для виконавчих механізмів об'єкта керування [1-4].

Основою для побудови цієї моделі є узагальнене формалізоване представлення поведінки технологічних агрегатів дискретної циклічної дії (ТА_{дц}), що є типовими об'єктами для застосування ІТПЛК.

Під ТА_{дц} розуміють такі об'єкти, чия поведінка у часі й просторі є виключно детермінованою і може бути описана сукупністю певних характеристик. Зокрема, для опису логіки їх роботи розглянемо набір відповідних множин. Множина, що включає кінцеву кількість станів виконавчих механізмів, у яких вони перебувають протягом одного повного циклу функціонування агрегату:

$$C = \{c_1, c_2, \dots, c_m\}. \quad (1.4)$$

Множина станів датчиків, що контролюють відповідні механізми і відображають поточні умови зовнішнього середовища:

$$A = \{a_1, a_2, \dots, a_k\}. \quad (1.5)$$

Множина дискретних інтервалів автоматного часу, які описують часову структуру циклу роботи ТА_{дц}.

$$T = \{t_1, t_2, \dots, t_{sf}\}. \quad (1.6)$$

Таким чином, кожен цикл функціонування об'єкта дискретної дії моделюється як послідовність логічно пов'язаних станів виконавчих механізмів, сигналів з датчиків та інтервалів часу, що й лежить в основі формалізованого опису поведінки у межах паралельної логіки керування.

Слід зазначити, що для кожного i -го інтервалу дискретного автоматного часу визначається скінченна підмножина (комбінація) станів датчиків, яка позначається як:

$$A_i = \{a_{i1}, a_{i2}, \dots, a_{ik}\}, A_i \subset A. \quad (1.7)$$

Ця підмножина є єдиною допустимою комбінацією вхідних сигналів, за якої дозволяється активація відповідної єдиної підмножини виконавчих механізмів, що описується множиною:

$$C_i = \{c_{i1}, c_{i2}, \dots, c_{im}\} C_i \subset C. \quad (1.8)$$

Інакше кажучи, кожному набору вхідних станів у певний дискретний момент часу відповідає строго визначена комбінація вихідних дій, що й визначає принцип функціонування керованого об'єкта.

Таким чином, циклограму роботи ТА_{ДЦ} можна формально представити за допомогою двох прямокутних матриць фіксованого розміру – матриці станів датчиків A та матриці станів механізмів C .

При цьому рядки обох матриць мають жорстке детерміноване розташування, і кожному i -му рядку матриці A однозначно відповідає i -й рядок матриці C :

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1k} \\ a_{21} & a_{22} & \dots & a_{2k} \\ \dots & \dots & \dots & \dots \\ a_{s1} & a_{s2} & \dots & a_{sk} \end{pmatrix}, \quad C = \begin{pmatrix} c_{11} & c_{12} & \dots & c_{1m} \\ c_{21} & c_{22} & \dots & c_{2m} \\ \dots & \dots & \dots & \dots \\ c_{s1} & c_{s2} & \dots & c_{sm} \end{pmatrix}. \quad (1.9)$$

Це відповідність є ключовою для реалізації алгоритму керування у логічному автоматі паралельної дії, де логіка керування визначається взаємозв'язком між очікуваними станами датчиків і необхідними діями виконавчих механізмів.

У матриці A кількість стовпців k відповідає загальній кількості контрольованих датчиків, які встановлено на механізмах технологічного агрегату. Відповідно, у матриці C кількість стовпців m дорівнює числу виконавчих механізмів, задіяних у керуванні агрегатом. Кількість рядків s в обох матрицях однакова, оскільки вона визначає кількість етапів (кроків) у циклограмі, тобто кількість послідовних станів у повному циклі роботи агрегату.

З огляду на те, що в системах керування об'єктами дискретної дії основними є саме кінцеві стани механізмів і датчиків, такі як «увімкнено» або «вимкнено», можна стверджувати, що процеси керування здебільшого реалізуються на основі бінарних сигналів. Тому циклограму функціонування подібного технологічного обладнання доцільно подати у вигляді булевих матриць, що відображають відповідність між наборами вхідних і вихідних сигналів на кожному етапі.

Таким чином, для математичного опису поведінки ТА_{дц} можуть бути використані булеві матриці типу C_{im} і A_{ik} , у яких вектори-рядки, що містять стани механізмів і датчиків, розташовуються у чітко визначеній, детермінованій послідовності. Це дозволяє формалізувати процес керування як послідовність логічно пов'язаних станів, що легко реалізується у структурі логічного автомата паралельної дії.

1.2 Відомі структури керуючих пристроїв з паралельною архітектурою

У даному розділі буде розглянуто відомі структури логічних керуючих автоматів паралельної дії, що були створені науковою школою під керівництвом професора Фурмана Іллі Олександровича у період з кінця 70-х

років минулого сторіччя і по теперішній час.

Перш ніж перейти безпосередньо до розгляду відомих архітектур ЛКА ПД, необхідно окремо підкреслити низку принципових особливостей, які визначають унікальні переваги пристроїв паралельної дії порівняно з традиційними контролерами послідовного типу. Саме ці особливості формують теоретичну та практичну основу для подальшого вдосконалення архітектури ЛКА ПД. Ці ключові відмінності були чітко сформульовані професором І. О. Фурманом, який зазначав: принципова відмінність описаної структури від відомих структур мікропрограмних автоматів полягає в тому, що управління процесом відпрацювання програми відбувається не від імпульсів тактового генератора, як у мікропрограмних автоматах, побудованих за класичною схемою Вілкса і її удосконаленим варіантам, і не потоком команд, як в ПЛК послідовної дії, а потоком станів входів, тобто потоком, що формують в асинхронній послідовності інтервалів дискретного автоматного часу «дозволені» (такі що відповідають запрограмованим) комбінації фактичних станів керованого об'єкта та зовнішнього середовища. Крім того, паралельна реалізація операцій над булевими векторами в автоматі дозволяє оперувати машинними словами довільної довжини» [4, 15].

Це означає, що логіка керування в ЛКА ПД побудована не на лінійній обробці послідовних інструкцій, а на подієвій реакції на зміну входних сигналів, що формуються незалежно один від одного в реальному часі. Система таким чином працює асинхронно, реагуючи виключно на появу запрограмованих станів входів, які вважаються допустимими або цільовими з точки зору керування. Крім того, важливою перевагою є паралельна обробка булевих векторів, що дозволяє оперувати машинними словами довільної довжини. Це дає змогу реалізувати складну логіку прийняття рішень без втрати швидкодії навіть при значному розширенні кількості входів та виходів.

Таким чином, головною особливістю ЛКА ПД порівняно з автоматами

послідовної дії є здатність:

- одночасно, в один такт дискретного часу, аналізувати велику кількість вхідних сигналів;
- паралельно формувати керуючі команди для практично необмеженої кількості виконавчих механізмів;
- працювати у реальному часі без необхідності циклічного опитування вхідів або жорстко визначених тактових імпульсів.

Ці переваги забезпечують високу швидкодію, гнучкість та масштабованість ЛКА ПД і є основою для подальшого розвитку ефективних архітектур керування з використанням ПЛІС-технологій у різних сферах – від промислової автоматики до інтелектуальних розподілених систем.

1.2.1 Універсальний ПЛК паралельної дії

На рисунку 1.1 показано структуру універсального ПЛК паралельної дії, запропоновану І. О. Фурманом і на базі якої було реалізовано промислові зразки контролерів паралельної дії різних типів: БЛ2-1, БЛ1-8, ПЛ-1 [4, 21].

Основними блоками універсального паралельного ПЛК, що визначають його роботу є блоки пам'яті (БП), до яких записуються стани керуемого об'єкта, команди керування, а також адреси переходів – тобто матриці A , B , C і D , призначення і детальний розгляд яких буде наведено нижче. Таким чином до БП(A) записується матриця A очікуваних станів об'єкта керування, до блоку БП(B) поміщається матриця B очікуваних станів зовнішнього середовища, блок пам'яті БП(C) містить матрицю C команд керування, а блоці БП(D) розміщується матриця D адрес переходів.

Робота ПЛК ПД може бути описана наступним чином. Лічильник адреси $LчA2$, що тактується генератором Γ , забезпечує послідовне зчитування рядків з блоку пам'яті БП(B). При цьому за допомогою схеми порівняння $Ср2$ відбувається паралельне логічне порівняння фактичного стану зовнішнього середовища з векторами що записані (запрограмовані) у

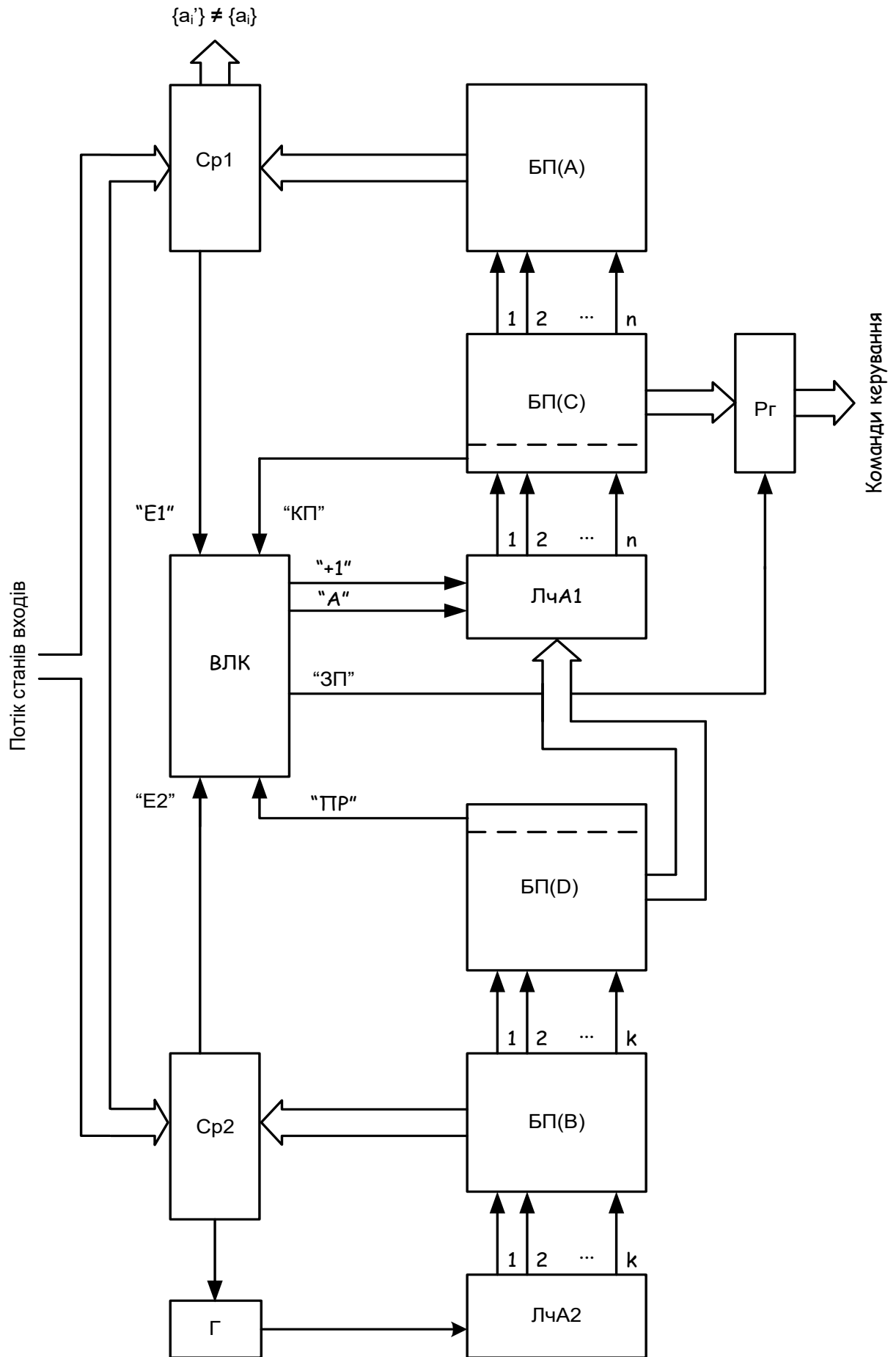


Рисунок 1.1 – Структура універсального ПЛК паралельної дії

БП(В). У випадку, якщо відбувається збіг комбінацій обох векторів вузол логічного керування (ВЛК) отримує від Ср2 команду еквівалентності «Е2». Одночасно Ср2 блокує роботу генератора Г, через що лічильник адреси ЛчА2 зупиняється, а з БП(Д) на ЛчА1 видається початкова адреса відповідної підпрограми з БП(А) і БП(С).

При цьому перевіряється умова закінчення відпрацювання попередньої підпрограми: якщо вона ще знаходиться на стадії виконання, сигнал КП – кінець підпрограми, відсутній; якщо попередня підпрограма вже відпрацьована, сигнал «КП» приймає значення логічної одиниці і відбувається перехід до першого рядка нової підпрограми за адресою, що зчитано з БП(Д). При цьому запис початкової адреси обраної підпрограми до ЛчА1 забезпечує ВЛК за допомогою сигналу «А» – адреса. У результаті формування на ЛчА1 адреси, відбувається адресація відповідних рядків блоків пам'яті БП(С) і БП(А) і, як наслідок видача на вихідний регістр Рг комбінації команд керування і подача на схему порівняння Ср1 очікуваної комбінації станів детермінованих входів. Також у цей момент ВЛК формує сигнал «ЗП» – запис, за яким регістр Рг фіксує вектор команд керування і їх видачу на виконавчі механізми керованого об'єкта.

Наслідком виконання команд стає спрацювання відповідних механізмів технологічного агрегату, що, у свою чергу, має призводити до формування відповідної комбінації станів на входах схеми порівняння Ср1. Ця схема, по аналогії з Ср2, виконує порівняння комбінації фактичного стану керованого об'єкта з комбінацією, що має бути сформована у результаті виконання даного рядка поточної підпрограми, тобто є наслідком виконання запрограмованих на поточному рядку БП(С) команд керування. Якщо схема Ср1 спрацьовує, це призводить до формування сигналу еквівалентності Е1 і, як наслідок, видачу блоком ВЛК на ЛчА1 команди «+І». Сигнал «+І» призводить до інкрементації адреси на блоках БП(А) і БП(С) і їх переходу на наступний рядок поточної підпрограми. У подальшому процес відпрацювання поточної підпрограми відбувається аналогічно.

У випадку, якщо спрацювання відповідного виконавчого механізму не відбулося, сигнал «ЕІ» на виході Ср1 не буде сформовано і перехід до наступного рядка не відбудеться, тобто фактично перехід буде заблоковано. При цьому на пристрій індикації (на структурі не показано) видається інформація про неспівпадіння фактичного стану i -го механізму (тобто i -го датчика) запрограмованому $\{a_i'\} \neq \{a_{ij}\}$. Ця інформація може бути використана обслуговуючим персоналом для пошуку несправностей або збоїв у роботі технологічного агрегату. Як тільки виявлена несправність буде усунута, ПЛК ПД автоматично продовжить відпрацювання наступних рядків поточної підпрограми.

У результаті відпрацювання останнього рядка поточної підпрограми до ВЛК поступає відповідний сигнал «КП», що призводить до ігнорування ВЛК сигналу «ЕІ» від схеми порівняння Ср1 і очікування переходу до наступної підпрограми, тобто очікуванню появи сигналу Ср2.

Описана вище процедура вибору підпрограми і її відпрацювання характерна для «нормального» ходу технологічного процесу. У той же час у процесі функціонування реальних промислових об'єктів існують ситуації, що вимагають негайного втручання через наявність аварійних комбінацій станів виконавчих механізмів. Характерним прикладом може бути наявність кнопки аварійної зупинки технологічного процесу «STOP», що примусово зупиняє рух усіх механізмів. У випадку наявності такої ситуації необхідно виконати перехід до початкової адреси перериваючої підпрограми, незалежно від того чи була виконана до кінця попередня. Для забезпечення такої процедури в усіх рядках блоку пам'яті БП(D) виділений один розряд для запису ознаки переривання «ПР».

1.2.2 Сучасний ЛКА ПД

На рисунку 1.2 показано структуру найбільш актуального сучасного ЛКА ПД. Наведену структуру покладено в основу декількох варіантів

- блок пам'яті переходів БПП (відповідно БП(D) і БП(B), та Ср2);
- блок індикації – БІ (аналогічний не показаному на попередній структурі елемент індикації невідповідності векторів $\{ai'\} \neq \{ai\}$);
- схема порівняння – СП (Ср1);
- блок логічного керування – БЛК (еквівалент ВЛК);
- лічильник адреси – ЛА (виконую функції ЛчА1 і ЛчА2);
- вихідного регістра – ВР (аналогічний Рг).

Розширення безпекових можливостей і можливостей формування логічного порівняння забезпечується додатковими блоками:

- блоком пам'яті заборонених комбінацій – БПЗК;
- блоком вибору операції – БВО.

Виконання основних функцій логічного керування здійснюється шляхом перевірки істинності рівнянь (1.10) і (1.11), які закладені у блок логічного керування:

$$A = КП \vee ПП1 \vee ПП2 \quad (1.10)$$

$$+1 = E \wedge \overline{КП} \wedge \overline{ПП1} \wedge \overline{ПП2} \quad (1.11)$$

де $КП$ – ознака кінця підпрограми;

$ПП1$ – ознака переривання від БПП;

$ПП2$ – ознака переривання від БПЗК;

A – початкова адреса підпрограми;

E – сигнал еквівалентності;

$+1$ – сигнал переходу до наступного рядка.

Процес виконання керуючої програми поділяється на два ключові етапи, які функціонують у режимі незалежної паралельної обробки.

Перший етап полягає в аналізі комбінацій вхідних сигналів, що надходять від датчиків, які відображають поточний стан зовнішнього середовища або об'єкта керування. На цьому етапі система порівнює фактичні стани з тими, які були попередньо запрограмовані як умови

переходу, і на основі результатів цього аналізу визначає початкову адресу підпрограми, яка має бути виконана.

Другий етап охоплює безпосереднє виконання вибраної підпрограми відповідно до її алгоритмічної логіки, закладеної у відповідні блоки пам'яті. Важливою особливістю є те, що аналіз вхідних станів здійснюється асинхронно та паралельно з виконанням основної підпрограми, що забезпечує високу швидкодію та адаптивність системи.

У типових архітектурах ЛКА ПД, визначення початкової адреси підпрограми покладено на блок пам'яті переходів (БПП). У випадку, якщо на його вході з'являється одна із запрограмованих комбінацій вхідних станів, лічильник адреси (ЛА) встановлюється у позицію, що відповідає цій комбінації. Така операція реалізує виконання логічного рівняння (1.10), що описує умову істинності для переходу, де при досягненні рівності $KП = A$ (ознака кінця підпрограми дорівнює початковій адресі), блок логічного керування (БЛК) формує сигнал «А» (Адреса). Цей сигнал ініціює переадресацію блоків пам'яті станів (БПС) і команд (БПК) на перший рядок відповідної підпрограми.

Крім того, у структурі програмного забезпечення ЛКА ПД передбачено, що кожна підпрограма завершується спеціальною міткою – ознакою кінця підпрограми (КП). Вона також обов'язково міститься в нульовому рядку, який використовується як початковий стан системи або як логічний завершувач циклів керування. Саме наявність цієї мітки дає дозвіл автомату перейти до обробки наступної підпрограми відповідно до нових умов, що формуються з боку зовнішнього середовища.

Таким чином, завдяки паралельному та умовно-незалежному функціонуванню компонентів, структура ІТЛК дозволяє забезпечити швидке реагування системи на зміну вхідних сигналів та ефективну зміну активної логіки керування без необхідності переривання виконання поточної програми.

Виконання обраної підпрограми здійснюється завдяки послідовному

відпрацюванню її кроків, кероване умовами переходу, визначеними логічним рівнянням (1.11). Саме це рівняння формалізує механізм переходу на наступний етап (рядок) підпрограми.

Якщо умови рівняння (1.11) виконуються, блок логічного керування (БЛК) генерує керуючий сигнал «+I». Цей сигнал слугує імпульсом для лічильника адреси (ЛА), який, у відповідь, перемикає активну позицію в блоках пам'яті станів (БПС) та команд (БПК) на наступний – $(i+1)$ -й рядок поточної підпрограми. Таким чином, система поступово проходить усі етапи заданого алгоритму.

Ключовою умовою для формування сигналу «+I» є поява сигналу еквівалентності («E»). Його виникнення залежить від логічного аналізу фактичних сигналів від датчиків і визначається двома можливими сценаріями.

Перший сценарій – повне співпадіння усіх очікуваних станів із фактичними (режим логічної операції «I»). У цьому випадку до відповідного рядка БПС записується ознака, що вказує на необхідність перевірки всіх входів. Тоді блок вибору операції (БВО) активує сигнал «I=I», який перемикає схему порівняння (СП) в режим кон'юнкції. Сигнал «E» формується лише тоді, коли усі датчики перебувають у станах, які збігаються з очікуваними значеннями.

Другий сценарій – часткове співпадіння. Спрацьовування хоча б одного датчика (режим логічної операції «АБО»). Якщо перехід на наступний крок залежить лише від одного з можливих входів, БВО формує сигнал «АБО=I». У цьому режимі СП переходить до диз'юнктивної логіки, і сигнал «E» з'являється при першій збіжності хоча б одного входу із заданим значенням.

Ця гнучкість у налаштуванні умов переходу дозволяє адаптувати підпрограму до широкого спектра технологічних сценаріїв, включаючи як суворо визначені, так і більш гнучкі ситуації керування.

Окрему групу сигналів у системі становлять ознаки переривань – «PP1» та «PP2», а також сигнали «PB» (поточний вхід) і «BB» (блокування

виходу). Хоча вони не входять до основної логіки формування сигналу переходу, їх присутність є необхідною з точки зору технічної реалізації. Вони забезпечують обслуговування асинхронних подій, організацію взаємодії з периферією та інші допоміжні функції, важливі для надійної і коректної роботи автомата в умовах реального часу.

1.2.3 Структура АПДН

У межах розвитку технології паралельного логічного керування, класичні ЛКА ПД, а також ПЛІС-контролери, що реалізують ці автомати на програмованих логічних інтегральних схемах, демонструють найвищу ефективність переважно при керуванні дискретними об'єктами циклічної дії. Зокрема, це стосується технологічних процесів із повторюваними операціями, де ключовими є швидкодія, відмовостійкість та точне відтворення команд у фіксованій послідовності.

Однак, із розширенням сфер застосування ІТПЛК, зокрема у складних, слабкоструктурованих системах (наприклад, в інтелектуальних енергетичних мережах), виникла необхідність у вдосконаленні та розширенні функціональних можливостей ЛКА ПД та відповідних ПЛІС-контролерів. У зв'язку з цим було проведено ряд досліджень, спрямованих на пошук шляхів адаптації цієї технології до умов невизначеності та нечітких вхідних даних.

У результаті запропоновано низку рішень, серед яких особливо слід відзначити два ключові напрямки:

- уведення програмованих таймерів, які дозволяють реалізовувати затримки, очікування та інші часові залежності без втручання у основну логіку автомата;

- впровадження апарату нечіткого логічного висновку, на основі якого створено модифіковану версію керуючих автоматів – АПДН (автомат паралельної дії з нечіткою логікою).

З погляду організації входів та виходів, АПДН розширює класичну структуру

ЛКА ПД, дозволяючи обробляти як чіткі (цифрові та аналогові), так і фазифіковані (нечіткі) сигнали. На рисунку 1.3 показано структуру вхідних/вихідних даних АПДН, що включає:

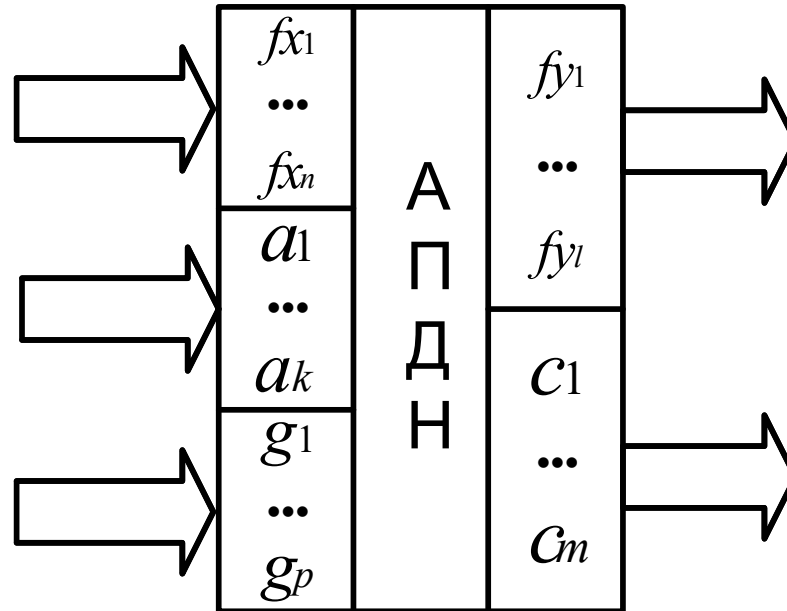


Рисунок 1.3 – Узагальнене представлення АПДН

- $fx_1 \dots fx_n$ – фазифіковані вхідні дані, що представляють розмиті (нечіткі) величини;
- $a_1 \dots a_k$ – цифрові двійкові входи, традиційно використовувані у логічних схемах;
- $g_1 \dots g_p$ – аналогові входи, що підлягають попередній обробці (оцифруванню або фазифікації);
- $fy_1 \dots fy_l$ – фазифіковані вихідні дані, які дозволяють формувати нечіткі керуючі сигнали;
- $c_1 \dots c_m$ – цифрові двійкові виходи для керування класичними виконавчими пристроями.

Реалізована структура АПДН дозволяє застосовувати технологію паралельного логічного керування:

- у системах з неповними, розмитими або нечіткими вхідними даними;
- у середовищах, де висока швидкодія має поєднуватися з

адаптивністю до нестандартних ситуацій;

- для розв'язання задач в енергетиці, промисловій автоматизації, транспорті та «розумних мережах» – Smart Grid.

Таким чином, АПДН є еволюційним продовженням класичних автоматів паралельної дії, яке відкриває нові горизонти застосування ІТПЛК в умовах сучасних технічних викликів.

Розглянемо більш детально архітектуру автомата паралельної дії з нечіткою логікою. За основу його побудови береться модифікована структура класичного логічного керуючого автомата паралельної дії, що зображена на рисунку 1.2. Вдосконалена архітектура представлена на рисунку 1.4 [12, 13].

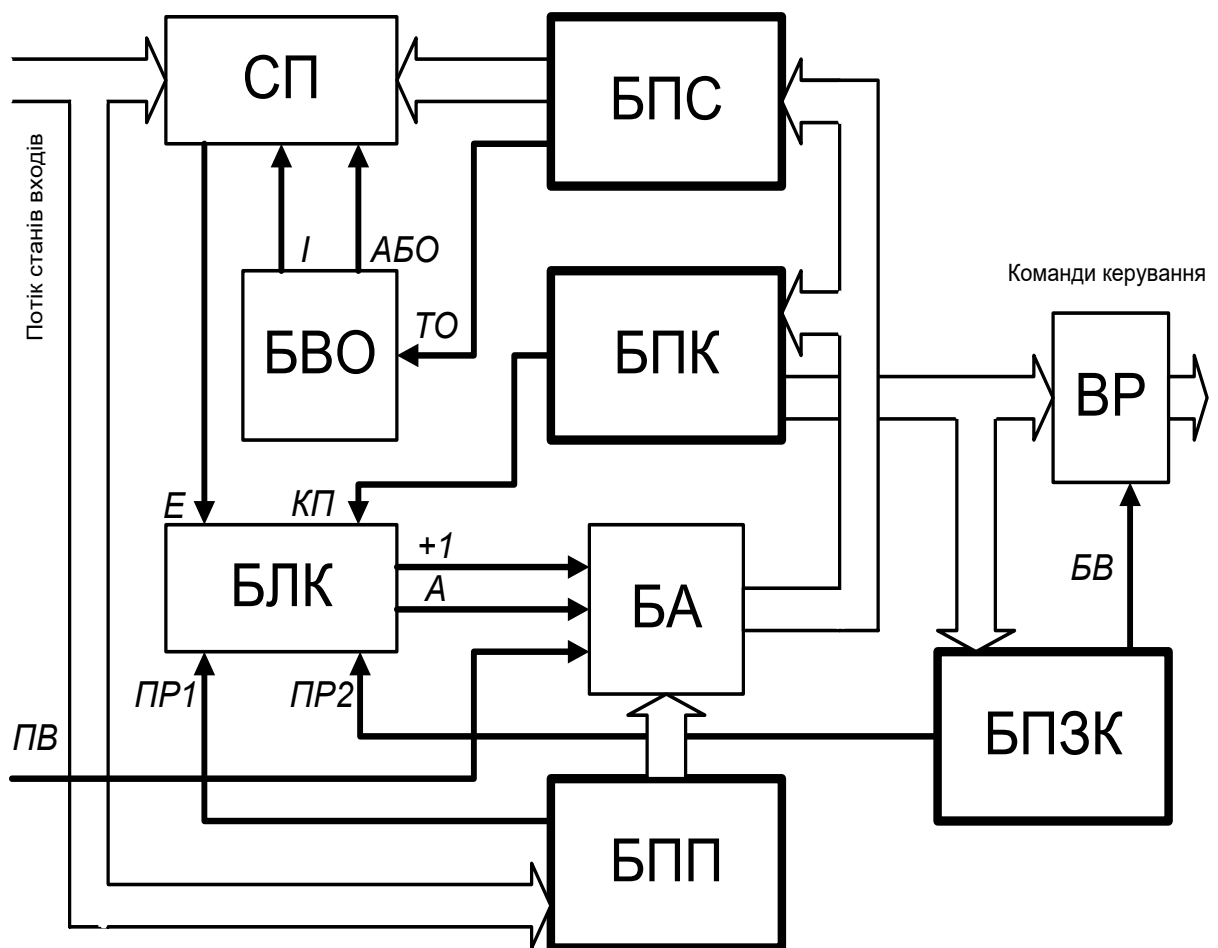


Рисунок 1.4 – Структура автомата паралельної дії з функціями нечіткого логічного висновку

Вкажемо основні структурні та функціональні відмінності АПДН від ЛКА ПД.

Удосконалення структури АПДН пов'язане з необхідністю реалізації нечіткого логічного висновку, що значно змінює функціональне призначення окремих блоків та загальну організацію обчислювального процесу в автоматі.

Вилучено блок індикації (БІ). У класичному ЛКА ПД блок індикації забезпечував візуальний контроль за виконанням технологічного процесу, що було актуально в умовах роботи з дискретними, детермінованими даними. Проте в АПДН цей блок втрачає свою актуальність, оскільки:

- система оперує фазифікованими (нечіткими) даними, що представлені у вигляді еквівалентного двійкового коду;
- візуалізація внутрішніх станів і переходів не несе практичного сенсу для користувача;
- автомат функціонує в автономному режимі без необхідності постійного моніторингу.

Зміна функцій блоку пам'яті переходів (БПП) і блоку пам'яті команд (БПК). У традиційній структурі ЛКА ПД БПП формував адреси переходів у підпрограмах залежно від умов на входах, реалізуючи алгоритмічне розгалуження. У випадку АПДН, де переважає логіка нечіткого висновку на основі поточного вхідного контексту, така функція втрачає актуальність.

У зв'язку з цим, БПП та БПК інтегруються у логічну пару, що реалізує:

- зберігання бази правил нечіткого висновку у вигляді таблиць;
- логічну обробку фазифікованих вхідних даних за заздалегідь визначеними умовами;
- пошук відповідних керуючих дій шляхом порівняння з прецедентами.

Таким чином, ці два блоки стають центральними елементами АПДН, на які припадає основне обчислювальне навантаження.

Функція блоку пам'яті станів (БПС). Блок БПС зберігає послідовність виконуваних команд і використовується тоді, коли логіка автомата потребує

жорсткої алгоритмічної структури з фіксованими переходами. Проте, якщо автомат працює в адаптивному режимі з реакцією на змінний потік вхідних даних (що типово для АПДН), цей блок може бути вилючено, і автомат переходить до умовно-ситуативного керування, що базується на прямих правилах.

До структури АПДН додано новий блок – блок адресації (БА), що виконує функції подібні до лічильника адреси (ЛА) у структурі ЛКА ПД. Така зміна пов'язана з тим, що базовою функцією АПДН є формування нечіткого логічного висновку (зв'язка блоків БПП, БПК та БА), а не послідовне відпрацювання рядків програми (де базовою була зв'язка блоків БПС та ЛА).

Отже структура АПДН – це функціонально трансформована версія ЛКА ПД, орієнтована на обробку нечітких даних у реальному часі, гнучку адаптацію до поточного стану середовищ та зниження апаратної складності, якщо відсутня потреба у жорстких сценаріях керування. Це робить АПДН ефективним інструментом для побудови інтелектуальних систем керування, зокрема в енергетиці, транспорті, Smart Grid, розподілених автоматизованих системах.

1.2.4 Структура ПЛІС-контролера ПД з програмованими таймерами

У межах розвитку технології актуальним завданням стала інтеграція функцій програмованих таймерів у структуру керуючих автоматів паралельної дії. Аналіз структури та математичної моделі таких автоматів, представлений раніше, засвідчує, що ефективна реалізація таймерів потребує чіткого визначення їх ролі у логіці керування, зокрема у контексті двох ключових логічних рівнянь: умов переходу між підпрограмами (1.10) та умов переходу між рядками всередині підпрограми (1.11).

При розгляді можливих варіантів застосування таймерів виникає питання: чи повинні їх внутрішні змінні (наприклад, лічильники часу або

сигнали спрацювання) впливати на обидва рівняння, чи лише на одне з них? Поглиблений аналіз вказує на те, що повноцінне використання потенціалу таймерів вимагає їх участі в обох логічних рівняннях, а саме:

- у рівнянні (1.11) – для ініціації переходу на наступний крок (рядок) поточної підпрограми;

- у рівнянні (1.10) – як частина умов формування сигналу для переходу до нової підпрограми, наприклад сигналу «КП».

Розглянемо деякі особливості участі таймерів у процесі керування.

Пряме формування сигналу «+I». Програмований таймер може самостійно формувати сигнал «+I», що активує лічильник адреси (ЛА) і переводить виконання на наступний рядок поточної підпрограми. Такий сценарій особливо актуальний у випадках, коли необхідна затримка виконання команди на заданий інтервал часу.

Обмеження участі у переходах між підпрограмами. Формування адреси переходу до іншої підпрограми (відповідно до рівняння (1.10)) залишається винятковою прерогативою блоку аналізу станів стохастичних входів – тобто датчиків зовнішнього середовища. Це зумовлено необхідністю точного реагування на зміну реального стану об'єкта керування.

Непряма участь у формуванні сигналу «КП». Внутрішні сигнали таймера можуть брати участь у формуванні сигналу «КП» або аналогічної ознаки завершення підпрограми. У цьому випадку таймер не визначає конкретну адресу переходу, але ініціює дозвіл на зміну логіки керування за результатами аналізу зовнішніх умов.

Щоб реалізувати описану функціональність, окремі елементи класичної структури автомата (рисунок 1.2) мають бути модифіковані, зокрема необхідно було:

- розширити функціональність блоку порівняння – для врахування внутрішніх сигналів таймера поряд із сигналами з БПС та зовнішніх входів;

- доповнити логіку формування сигналу «КП» – щоб передбачити ситуації, в яких завершення підпрограми залежить не лише від зовнішнього

стану, але й від внутрішнього таймера;

- впровадити спеціалізований блок обробки таймерів (БВТ) – для керування режимами запуску, зупинки, скидання та порівняння з заданими інтервалами часу.

Отже реалізація програмованих таймерів у структурі ЛКА ПД вимагає системного перегляду логіки обох основних рівнянь керування. Таймери можуть виступати як тригерні механізми для переходів усередині підпрограм і як допоміжні умови для ініціації переходів між підпрограмами. Такий підхід значно розширює функціональні можливості автомата, підвищує гнучкість і адаптивність системи керування, особливо у випадках роботи в умовах реального часу, перерв і затримок.

Таким чином було запропоновано модифіковану структуру ЛКА ПД або його інженерного втілення – ПЛІС-контролера паралельної дії. Така структура представлена на рисунку 1.5. У структурі впроваджено повноцінну підтримку програмованих внутрішніх таймерів відповідно до положень [5, 14].

Ключовою відмінністю цієї нової архітектури є введення двох нових функціональних елементів, які забезпечують інтеграцію часової логіки в загальну структуру керування:

- блок пам'яті внутрішніх змінних (БПВЗ) – призначений для збереження числових параметрів та проміжних значень внутрішніх таймерів, наприклад: тривалостей відліку, поточних лічильників, ознак завершення;

- блок внутрішніх таймерів (БВТ) – є самостійним функціональним блоком, відповідальним за організацію, запуск і завершення процедур відліку часу. БВТ можна умовно поєднати з блоком логічного керування (БЛК), проте це значно ускладнило б структурний опис і зменшило б прозорість роботи автомата, особливо для цілей діагностики та верифікації.

Розглянемо базовий принцип дії інтегрованих таймерів.

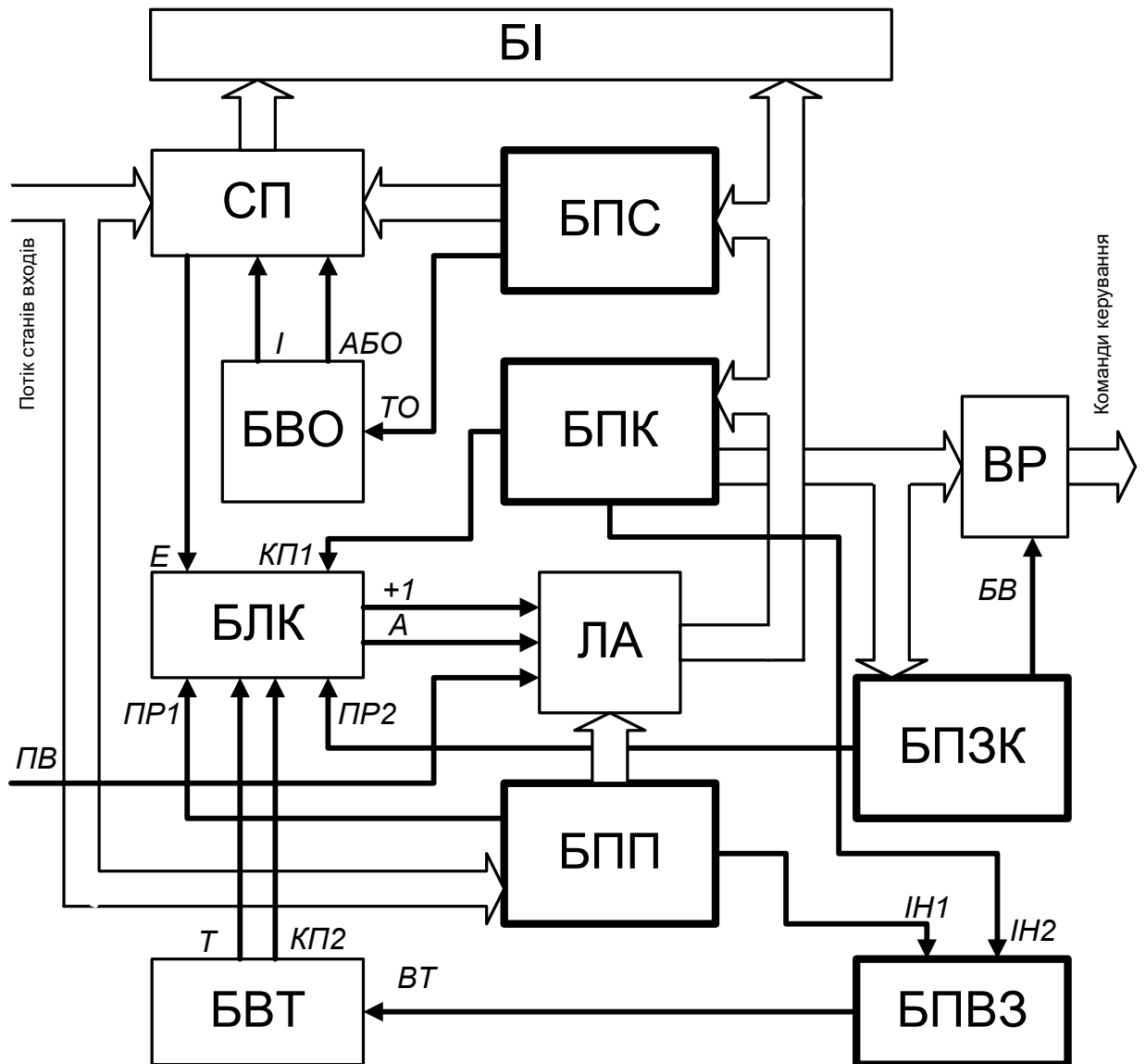


Рисунок 1.5 – Структура ЛКА ПД з елементами реалізації програмованих таймерів

Ініціація таймерів. Сигнали «ІН1» та «ІН2» – це тригери запуску таймерів, які можуть формуватись як з блоку пам'яті команд (БПК), так і з блоку пам'яті переходів (БПП). Ці сигнали визначають, який саме блок ініціює запуск таймера, та яку функціональну логіку він обслуговує (перехід між кроками чи між підпрограмами). Важливо, що відлік часу виконується незалежно від основної логіки ЛКА ПД. Він реалізується апаратними засобами ПЛІС-кристалу, що гарантує паралельність: поки таймер працює, автомат може одночасно виконувати інші функції керування.

Завершення відліку таймерами. Сигнал «ВТ» формується блоком БПВЗ

у момент завершення відліку. Це є означенням готовності таймера: заданий проміжок часу повністю відпрацьовано. За сигналом «BT» БВТ формує один із двох логічних сигналів:

- «T» – що може, наприклад, замінити сигнал «+1» та ініціювати перехід до наступного рядка програми;
- «КП2» – додаткова ознака завершення, яка виконує роль умовного дозволу на перехід до нової підпрограми.

Вибір сигналу залежить від контексту запуску таймера: якщо його ініціював БПК – виконується внутрішній перехід, якщо БПП – зовнішній (між підпрограмами).

За наявності сигналу «T», блок логічного керування формує сигнал «+1», що означає перехід до наступного, $(i+1)$ -го рядка поточної підпрограми. Це дозволяє зробити висновок про те, що принцип роботи з внутрішнім таймером загалом подібний до обробки сигналів від детермінованих входів. Водночас існує відмінність: якщо у випадку з детермінованими входами схема порівняння очікує появи конкретної наперед визначеної комбінації сигналів, то при використанні таймера така комбінація виникає автоматично – після завершення проміжку часу, заданого у блоці пам'яті внутрішніх змінних (БПВЗ).

Отже, формування сигналу «+1» відбувається згідно з наступним формалізованим виразом:

$$+1 = (E \vee T) \wedge \overline{КП1} \wedge \overline{КП2} \wedge \overline{ПП1} \wedge \overline{ПП2} \quad (1.12)$$

На наступному етапі розглянемо логіку функціонування автомата під час переходу до іншої підпрограми. У випадку, коли ініціатором запуску таймера виступає блок пам'яті переходів (БПП), блок внутрішніх таймерів (БВТ) формує сигнал «КП2». Його функціональне призначення повністю відповідає логіці сигналу «КП», який у даній структурі позначено як «КП1». Відповідно, перехід до нової підпрограми можливий лише після завершення

відліку часу, встановленого внутрішнім таймером. Такий принцип роботи з підпрограмами є аналогічним до випадку, коли ознака завершення підпрограми формується блоком пам'яті команд (БПК) у стандартних умовах, без залучення таймерів.

У такій структурі формування сигналу «А» відбувається згідно з наступним рівнянням:

$$A = КП1 \vee КП2 \vee ПП1 \vee ПП2 \quad (1.13)$$

Отже, механізм переходу за таймером у структурі ЛКА ПД чи ПЛІС-контролера паралельної дії добре інтегрується в основну логіку керування без порушення її структури, імітує логіку роботи з зовнішніми входами, проте базується на внутрішньому обчислювальному процесі та забезпечує високу точність та прогнозованість переходів, особливо у системах, де важливе часове моделювання дій (затримки, паузи, очікування тощо).

Таким чином, внутрішні таймери стають органічною складовою ІТПЛК, що суттєво розширює її функціональність і можливості адаптації до складних сценаріїв керування.

1.3 Постановка завдання

Проведений аналіз засвідчив, що керуючі пристрої з паралельною архітектурою, як показано в роботах [14, 11, 17], мають низку суттєвих переваг. Серед них варто виокремити дві ключові.

Першою перевагою є практично повна відсутність залежності між кількістю контрольованих входів та керованих виходів і швидкодією контролера. Це означає, що розширення системи введення/виведення не призводить до зниження продуктивності, що є важливою властивістю для масштабованих або складних систем автоматизації.

Другою важливою перевагою є можливість застосування технології

TVR, що дозволяє створювати керуючі програми без глибоких знань у сфері програмування. Завдяки використанню візуально-зрозумілих, спрощених мов опису технологічних процесів, така технологія:

- значно знижує ймовірність помилок у кодї програми;
- пришвидшує процес розробки логіки автоматики;
- спрощує комунікацію між технологами та програмістами;
- дозволяє делегувати програмування інженерам-технологам, що розуміють суть процесу, але не мають глибоких знань у програмуванні.

Усе це свідчить про актуальність подальшого вдосконалення керуючих пристроїв з паралельною архітектурою з метою розширення їх застосування у практичних задачах керування, як у промисловості, так і у сфері побутової автоматизації.

Виходячи з цього, головним завданням дослідження є усунення базового недоліку розглянутих традиційних рішень побудови ЛКА ПД, який полягає у «жорсткій», «негнучкій» логічній структурі та «закритій» архітектурі. Такий підхід обмежує адаптацію та масштабування системи. Тому завдання полягає у формуванні структури перспективного керуючого автомата паралельної дії, реалізованого на принципах «відкритої архітектури», яка б дозволяла легко модифікувати, розширювати та адаптувати систему до нових умов і вимог керування.

Таким чином головною метою кваліфікаційної роботи є розробка «гнучкої» або «модульної» архітектури керуючих автоматів паралельної дії та створення підґрунтя для побудови математичної та HDL-моделей таких автоматів.

2 РОЗРОБКА СТРУКТУРИ ПЕРСПЕКТИВНОГО КЕРУЮЧОГО АВТОМАТА ПД

Базовою, для побудови сучасних керуючих пристроїв паралельної дії, можна уважати структуру, що показана на рисунку 1.2 кваліфікаційної роботи [3, 4]. Саме її покладено в основу промислових зразків ПЛІС-контролерів паралельної дії на кристалах компанії Intel-Altera [8]. Саме результатом її розвитку є логічні керуючі автомати з розширеними функціональними можливостями, розглянуті у першому розділі [12-14]. Саме цю структуру потрібно узяти за основу для розробки перспективної архітектури керуючого автомата.

Поглиблений аналіз існуючих структур ЛКА ПД, а також окреслених у [5] перспективних напрямків їхнього розвитку, зокрема, у контексті впровадження лічильників та блоків виконання арифметичних операцій, свідчить, що основна логіка функціонування ЛКА ПД зводиться до формування двох ключових керуючих сигналів: сигналу «А» (адреса переходу до нової підпрограми) та сигналу «+1» (перехід до наступного рядка поточної підпрограми). Незалежно від того, який саме функціональний блок ініціює появу цих сигналів – схема порівняння, блок пам'яті переходів (БПП), блок внутрішніх таймерів (БВТ), або у перспективі блок лічильників чи арифметичних операцій – результатом є лише розширення логічного опису. Тобто базові рівняння (1.10) і (1.11), що визначають формування сигналів «А» і «+1», будуть трансформуватись у більш складні вирази (наприклад, рівняння (1.12) і (1.13)), які включатимуть дедалі більше змінних, умов, логічних елементів та ознак переривань. Це у свою чергу вказує на необхідність архітектурної гнучкості в структурі ЛКА ПД.

З огляду на це можна сформулювати основну ідею концепції модульної архітектури керуючих пристроїв паралельної дії наступним чином [22]: «перспективна архітектура ЛКА ПД має передбачати можливості «гнучкої»

зміни внутрішньої його структури за рахунок додавання або виключення блоків, що забезпечуватимуть необхідний рівень функціональності, без внесення змін до базових блоків, тобто його «ядра». Іншими словами, автомат повинен мати модульну архітектуру, в якій розширення функціоналу (наприклад, підтримка таймерів, лічильників, арифметики або нечіткої логіки) досягається додаванням незалежних блоків, що взаємодіють із ядром через стандартизовані інтерфейси. Це дозволяє адаптувати ЛКА ПД до специфіки конкретного об'єкта керування без перепроєктування всієї структури, підвищуючи таким чином гнучкість, масштабованість і універсальність системи керування.

Отже розглянемо основні положення, що формують пропоновану концепцію «модульної архітектури» ЛКА ПД [22]:

- ЛКА може складатись з необмеженої кількості блоків, що приймають участь у формуванні сигналів « A » і « $+I$ » – назвемо їх блоками визначення наступного етапу (БВНЕ);

- усі блоки БВНЕ мають відповідати умові паралельного і незалежного один від одного функціонування;

- логіка роботи ЛКА ПД має забезпечувати реалізацію пріоритезації блоків БВНЕ;

- базова архітектура ЛКА має забезпечувати додавання або виключення блоків БВНЕ без внесення змін до усіх інших елементів структури.

Запропонована структура керуючого автомата, розроблена відповідно до вищезазначених принципів та технічних рішень, отримала назву «модульний логічний керуючий автомат паралельної дії» (МЛКА ПД). Її загальний вигляд представлено на рисунку 2.1.

Зосередимось на ключових відмінностях цієї структури від класичної моделі ЛКА ПД, а також на призначенні нових функціональних блоків і зв'язків, що з'явилися у модернізованій архітектурі.

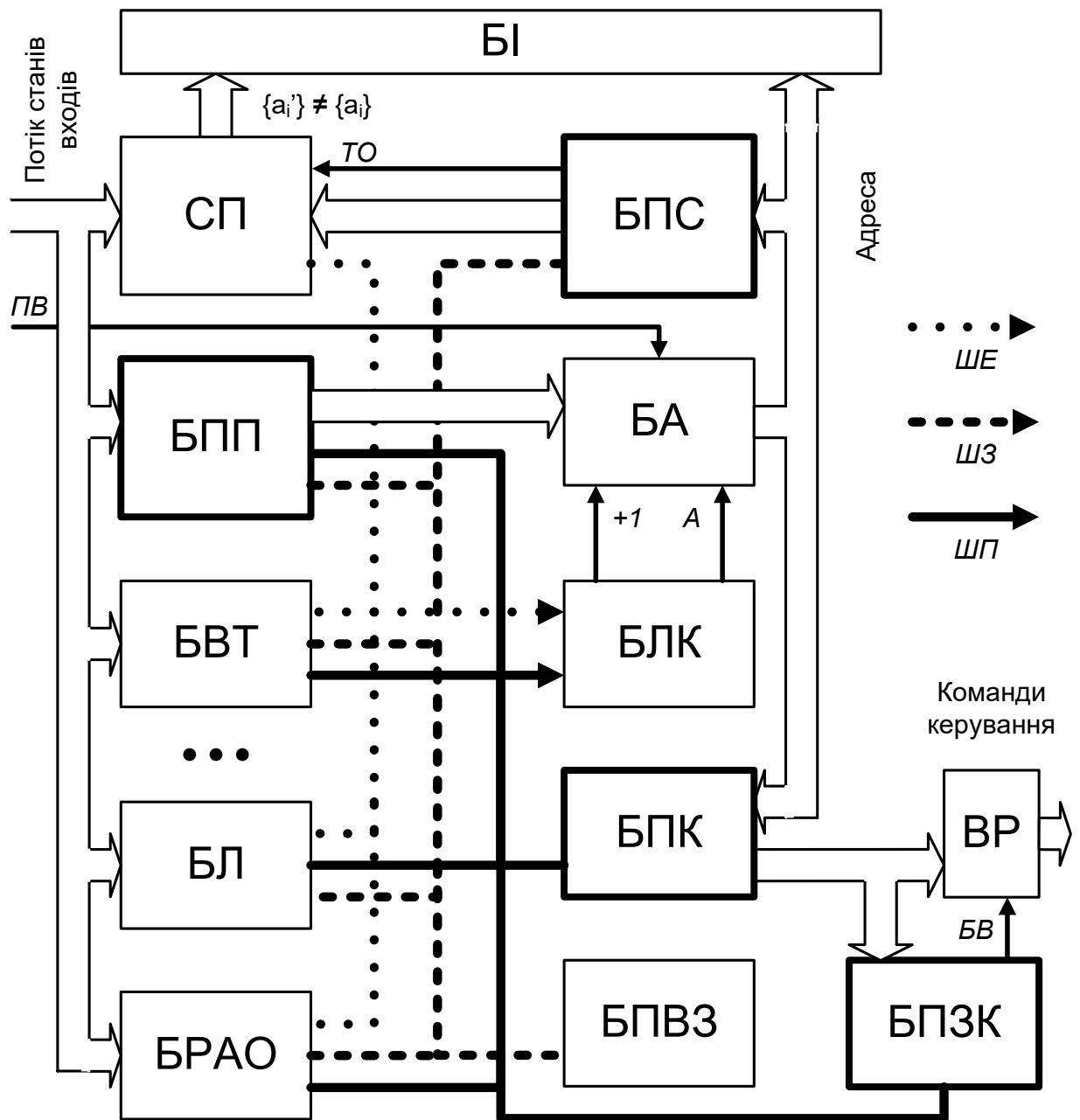


Рисунок 2.1 – Структура модульного логічного керуючого автомата паралельної дії

У частині, що відповідає за визначення наступного етапу виконання програми, структура МЛКА ПД зберігає чотири основні блоки:

- БПС – блок пам'яті станів;
- БПП – блок пам'яті переходів;
- БПК – блок пам'яті команд;
- БПЗК – блок пам'яті зовнішніх комбінацій.

Функціональне призначення та логіка роботи цих блоків залишаються

незмінними, і вони повністю відповідають тій ролі, яку вони виконували в класичному варіанті ЛКА ПД, що було розглянуто у першому розділі дослідження. У новій структурі ці компоненти виступають як стандартні модулі ядра, навколо яких можуть будуватись додаткові функціональні розширення, зберігаючи при цьому сумісність із раніше реалізованими архітектурними рішеннями.

До складу модульного логічного керуючого автомата введено нові функціональні блоки, які відповідають за визначення наступного етапу виконання програми, доповнюючи базову архітектуру класичного ЛКА ПД. Ці блоки розширюють можливості автомата, забезпечуючи підтримку додаткових функцій, важливих для сучасних задач керування:

- БВТ – блок внутрішніх таймерів. Це компонент, що реалізує функції програмованих користувачем таймерів, які можуть застосовуватись для відліку певних часових інтервалів у процесі керування. Хоча блок БВТ присутній також у попередньо розглянутій структурі (рисунок 1.4), у межах МЛКА ПД змінено логіку його взаємодії з іншими компонентами, адаптовано інтерфейси та взаємозв'язки, що дозволяє йому ефективно працювати у складі модульної системи. При цьому функціональне призначення блоку залишилось незмінним, як описано у [14];

- БЛ – блок лічильників. Це перспективний модуль, який дозволить реалізувати функції підрахунку імпульсів або подій, як внутрішніх (формованих логікою автомата), так і зовнішніх (отриманих із середовища або об'єкта керування). Блок БЛ відкриває можливість реалізації циклічних процесів, контролю кількості повторень та побудови логіки, що залежить від кількісних параметрів;

- БРАО – блок реалізації арифметичних операцій. Ще один перспективний компонент, призначений для виконання базових арифметичних дій (додавання, віднімання, множення, тощо). Його впровадження значно розширить спектр задач, що може вирішувати МЛКА ПД, зокрема у випадках, коли керування потребує обчислень над числовими

параметрами.

Ці нові блоки є прикладом модульного підходу до архітектури, що дозволяє гнучко формувати конфігурацію контролера відповідно до конкретних задач автоматизації, без потреби змінювати базове «ядро» ЛКА ПД. Такий підхід сприяє масштабованості, розширюваності та універсальності систем керування, створених на базі ІТПЛК.

Як уже зазначалося, до структури МЛКА ПД може бути додано необмежену кількість додаткових функціональних блоків типу БВНЕ. Ці блоки, так само як і описані раніше, інтегруються у загальну архітектуру шляхом підключення до внутрішніх системних шин, які забезпечують стандартизовану взаємодію між усіма складовими автомату.

Нижче наведено опис основних типів таких шин.

- ШЕ – шина еквівалентностей. Ця шина використовується для передачі до блоку логічного керування (БЛК) логічних сигналів, що свідчать про виконання певної умови в одному з блоків автомату. Ця умова є необхідною для формування сигналу «+I», який ініціює перехід до наступного рядка поточної підпрограми. Прикладами таких сигналів можуть бути сигнал еквівалентності «E», що з'являється при повному або частковому збігу очікуваних і фактичних входних станів «T» – сигнал завершення відліку таймера;

- ШП – шина переривань. Ця шина призначена для передачі до БЛК сигналів, що вказують на необхідність переривання виконання поточної підпрограми та переходу до нової підпрограми, тобто формування сигналу «A» – адреси першого рядка нової підпрограми. Сигнали, що передаються цією шиною, ініціюються при настанні певних перериваючих подій, прикладами яких можуть бути «PP1», «PP2» – стандартні сигнали переривань, які вже використовувались у попередніх структурах;

- ШЗ – шина внутрішніх змінних і констант. Ця шина введена до структури МЛКА ПД з метою організації обміну даними між окремими блоками, які у своїй роботі використовують внутрішні параметри,

лічильники, часові інтервали або константні значення. Ця шина забезпечує зв'язок з блоком пам'яті внутрішніх змінних (БПВЗ), який слугує для постійного зберігання внутрішніх налаштувань і змінних (наприклад, констант користувача), тимчасового зберігання проміжних результатів, значень лічильників або таймерів, тощо.

Упровадження шин ШЕ, ШП і ШЗ дозволяє організувати уніфіковану систему взаємодії між ядром автомата та додатковими блоками розширення. Це є ключовим елементом модульної архітектури МЛКА ПД, що забезпечує:

- масштабованість структури;
- гнучкість налаштування;
- можливість додавання нових функцій без зміни базової логіки керування.

Слід зазначити, що БВНЕ можуть бути приєднані як до усіх трьох шин, так і до лише однієї з них. Прикладом може слугувати БПС, що приєднаний лише до ШЗ. Саме за рахунок появи такого нового зв'язку значно розширюється функціональність формування умов переходу на наступний крок поточної підпрограми за рахунок появи можливості оперативної (у процесі роботи) зміни умов, що записані до БПС.

Слід зазначити, що у порівнянні зі структурою, наведеною на рисунку 1.2, функціональність блоку логічного керування у МЛКА ПД зазнала суттєвого розширення. Серед основних змін варто виділити наступне: у зв'язку з тим, що в оновленій архітектурі можливе одночасне виникнення кількох активних сигналів на шині переривань, виникає потреба у чіткому визначенні пріоритетів обробки цих сигналів. Це необхідно для коректного вибору тієї підпрограми, до якої має здійснитися перехід на наступному етапі роботи системи.

Як показано на рисунку 2.1, сигнал «БВ» (блокування виходу), так само як і в попередніх варіантах архітектури, формується не в БЛК, а у блоці пам'яті заборонених комбінацій (БПЗК). Попри те, що за логікою функціонування БПЗК є типовим представником блоків визначення

наступного етапу та підключений до шини переривань, саме він, а не БЛК, ініціює заборону видачі керуючих сигналів на виконавчі механізми. Таке архітектурне рішення є свідомо обраним і спрямоване на забезпечення підвищеного рівня безпеки. Реалізація функції блокування поза межами БЛК дозволяє:

- відокремити критично важливі елементи системи контролю заборонених станів (БПЗК) та відхідного регістру (ВР) від основної логіки керування;

- розміщувати ці компоненти окремо на кристалі ПЛІС, що знижує ризик внутрішніх збоїв;

- у найвідповідальніших випадках – реалізувати ці елементи на окремому кристалі, ізольованому від основного керуючого пристрою.

Додатково варто зауважити, що у структурі МЛКА ПД, якщо блок БПЗК присутній, то сигнал переривання, який він формує, завжди має найвищий пріоритет на шині ШП. Це гарантує, що умови блокування завжди будуть оброблені в першу чергу, незалежно від наявності інших активних запитів. Такий підхід дозволяє реалізувати повноцінний механізм апаратної безпеки у складі паралельного ПЛІС-контролера, що відповідає вимогам до сучасних систем критичного застосування.

Що стосується функціональності блоку індикації (БІ), то на рисунку 2.1, як і в попередніх реалізаціях, показано його стандартне підключення до лінії адреси поточного рядка програми, а також до ліній індикації нееквівалентності виходів схеми порівняння, тобто до сигналів, що відповідають умові $\{a_i'\} \neq \{a_i\}$. Завдяки цьому забезпечується функціональна діагностика роботи об'єкта керування, зокрема виявлення ситуацій, коли фактичні входи не збігаються з очікуваними значеннями, що особливо важливо для аналізу помилок та оцінки стабільності процесу [15]. Однак, враховуючи модульну й гнучку архітектуру МЛКА ПД, блок індикації не обмежується лише зазначеними каналами підключення. У рамках процесу конфігурації ПЛІС, БІ може бути підключений до будь-якої з внутрішніх шин

або окремих сигналів автомату. Це розширить його можливості та дозволить виконувати ще й наступні функції:

- візуалізація роботи будь-якого функціонального блоку, наприклад таймера, лічильника, модуля арифметичних операцій;
- моніторинг логічних станів у процесі налагодження та тестування системи;
- контроль роботи переривань, умов переходів, сигналів безпеки тощо;
- адаптація інтерфейсу оператора до вимог конкретного технологічного процесу.

Особливо важливою є роль БІ на етапі прототипування або діагностики, коли розробник потребує гнучкого доступу до внутрішніх сигналів системи для верифікації логіки, виявлення некоректних станів та оптимізації алгоритмів керування. Таким чином, у контексті МЛКА ПД блок індикації набуває статусу універсального засобу контролю й відображення, функціональність якого може бути гнучко адаптована до завдань конкретної конфігурації та умов експлуатації.

Особливу увагу необхідно приділити окремим аспектам реалізації блоку реалізації арифметичних операцій (БРАО) у перспективних проєктах МЛКА ПД. У роботі [22] підкреслюється, що необхідною умовою ефективної інтеграції БРАО до систем керування паралельної дії є дотримання паралельного принципу виконання обчислень – тобто реалізація арифметичних операцій «за один такт дискретного автоматного часу». Ця вимога не є випадковою. Як показано в [4, 11, 15], фундаментальною відмінністю ПЛК ПД від класичних програмованих логічних контролерів є саме виконання кожної операції за один автоматний такт (хоча з урахуванням фізичних обмежень ПЛІС, фактична кількість апаратних тактів може бути більшою). Саме ця особливість забезпечує високу швидкодію та паралельну обробку сигналів, яка є визначальною перевагою таких систем.

Однак реалізація арифметичних операцій, зокрема додавання, віднімання або множення, за своєю природою має послідовний характер.

Навіть у найпростішому випадку (наприклад, при додаванні двох чисел у двійковому представленні) виникає необхідність перенесення значення з молодших розрядів у старші, що створює принципове протиріччя з паралельною природою функціонування ЛКА ПД. Подолати це протиріччя шляхом простої модифікації або вдосконалення класичної структури автомату неможливо.

Одним із можливих і, на думку авторів, найперспективніших рішень у цьому контексті є застосування числової системи залишкових класів (СЗК, ЧСЗ або RNS) [18-20]. Цей підхід передбачає представлення чисел у вигляді набору залишків за кількома взаємно простими модулями, що дозволяє виконувати арифметичні операції паралельно над кожним модулем незалежно. У такій системі відсутні переноси між розрядами, а отже арифметичні операції можуть виконуватись безпосередньо за один такт, що ідеально відповідає вимогам ПЛК ПД.

Попри те, що у літературі вже зустрічаються окремі спроби аналізу можливостей застосування СЗК у контексті логіки паралельного керування [6], наразі відсутні реальні інженерні реалізації або навіть теоретично завершені архітектури ЛКА ПД, у яких арифметичний блок був би побудований на базі числової системи залишків. Це відкриває новий перспективний напрям досліджень, що поєднує апаратну ефективність ПЛІС з математичною перевагою СЗК, і може стати основою для розробки високопродуктивних керуючих пристроїв нового покоління.

Нескладно продемонструвати, що запропонована структура МЛКА ПД дає змогу реалізувати керуючий пристрій паралельної дії з будь-яким рівнем функціональності, що відповідає декларованим можливостям архітектури. Завдяки своїй гнучкій будові, система може бути адаптована до різних задач шляхом конфігурації її складових без необхідності змін у базовому «ядрі» автомата.

Зокрема, якщо використати лише чотири стандартні блоки визначення наступного етапу – блок пам'яті станів, блок пам'яті переходів, блок пам'яті

команд і блок пам'яті заборонених комбінацій, отримаємо структуру, яка повністю функціонально відповідає класичному ЛКА ПД (рисунок 1.2). У разі додавання до цієї конфігурації блоку внутрішніх таймерів та блоку пам'яті внутрішніх змінних, створюється варіант, аналогічний структурі, зображеній на рисунку 1.5, з розширеними можливостями і збереженням внутрішніх параметрів.

Очевидно, що закладений у проєкт принцип модульності відкриває широкі перспективи для подальшого розвитку. Така структура дозволяє реалізовувати контролери паралельної дії з функціональністю, яка ще не описана в межах цієї публікації, але цілком може бути впроваджена в майбутньому. Це надає архітектурі МЛКА ПД характер універсального, масштабованого рішення, придатного для реалізації як стандартних, так і новітніх концепцій автоматизації.

На завершення слід підкреслити, що запропонована концепція модульної архітектури логічного керуючого автомата паралельної дії становить лише початковий етап у процесі створення новітніх ПЛІС-контролерів паралельної дії. Реалізація принципу модульності на рівні апаратної архітектури відкриває широкі можливості для побудови гнучких, адаптивних і масштабованих керуючих пристроїв, які можуть легко підлаштовуватись під специфіку різних об'єктів і задач автоматизації. Проте для повноцінного впровадження цієї концепції на практиці необхідно розширити її на рівень програмного забезпечення.

Наступним логічним кроком розвитку є модернізація або повне переосмислення мови програмування ЯПЛК-М. Оскільки існуюча мова була розроблена під класичні, жорстко структуровані контролери, вона не забезпечує належної підтримки динамічної та конфігурованої архітектури, яка властива МЛКА ПД. Це ставить завдання створення нової мови програмування, яка так само, як і апаратна частина, повинна бути побудована за принципом модульності. Така мова має передбачати можливість опису структури контролера на логічному рівні, підтримку окремих типів блоків,

параметризацію їхньої взаємодії, а також автоматичну генерацію конфігурацій у залежності від обраної структури.

Окрім того, важливо забезпечити сумісність нової архітектури з технологією програмування TVP. Це дозволить реалізувати візуальний, інтуїтивно зрозумілий підхід до розробки керуючих програм, що суттєво спрощує процес для користувачів без глибоких знань у галузі апаратного або системного програмування. TVP повинна бути адаптована до нової архітектурної гнучкості, підтримувати створення модульних керуючих алгоритмів та взаємодію між різними блоками структури.

Таким чином, повноцінна реалізація концепції модульної архітектури ЛКА ПД передбачає не лише апаратні зміни, але й глибоку трансформацію мовних та програмно-логічних засобів. Це дозволить створити нове покоління ПЛІС-контролерів, які будуть не просто швидкими та надійними, а ще й відкритими, конфігурованими, інтуїтивно керованими й адаптованими до вимог сучасних та майбутніх автоматизованих систем.

3 HDL-МОДЕЛЬ ПЕРСПЕКТИВНОГО КЕРУЮЧОГО АВТОМАТА

3.1 Обґрунтування реалізації ЛКА ПД на ПЛІС

Програмовані логічні інтегральні схеми являють собою високорівневі мікроелектронні компоненти, призначені для створення спеціалізованих цифрових пристроїв з індивідуально заданою логікою. Вони поєднують у собі переваги серійного фабричного виробництва інтегральних мікросхем з гнучкістю налаштування, притаманною індивідуальній розробці апаратного забезпечення. Завдяки цьому ПЛІС широко використовуються для побудови як простих логічних вузлів, так і повноцінних систем керування, включаючи перетворювачі кодів, периферійні інтерфейси, мікропрограмні автомати, кінцеві автомати, цифрові помножувачі, сигнальні процесори й навіть компактні спеціалізовані обчислювальні пристрої (зокрема, для реалізації алгоритмів швидкого перетворення Фур'є).

Процес розробки електронного пристрою на основі ПЛІС полягає у проектуванні структури цифрової логіки на рівні з'єднань між елементарними логічними комірками, що входять до складу мікросхеми. Після цього створене з'єднання програмується в саму ПЛІС за допомогою спеціалізованого програматора. Важливо, що програмування може бути виконане безпосередньо розробником, без участі виробника мікросхеми. Такий підхід усуває необхідність дорогих і тривалих етапів виготовлення фізичної мікросхеми, наприклад, створення металевих масок або виготовлення корпусів з індивідуальними кристалами.

Основною перевагою ПЛІС у порівнянні з іншими видами спеціалізованих інтегральних схем є надзвичайно короткий час переходу від проекту до готового пристрою. Конфігурація схеми в ПЛІС може бути здійснена за лічені секунди або хвилини, що значно пришвидшує цикл розробки, тестування і впровадження нових електронних систем. Це зробило

ПЛІС особливо популярними у сфері прототипування, дослідницьких розробок, а також у проєктах з малосерійним або індивідуальним виробництвом.

Сучасні ПЛІС, окрім стандартних логічних комірок, містять у своїй структурі розширені модулі, такі як:

- блоки оперативної пам'яті;
- DSP-ядра (ядра цифрових сигнальних процесорів);
- помножувачі;
- PLL-блоки (схеми фазового автопідстроювання частоти);
- інші елементи, що розширюють функціональні можливості пристрою.

Як правило, розробнику проєкту не зобов'язаний детально знати структуру елементної бази конкретної ПЛІС. Замість цього йому достатньо описати необхідну логіку в одній із мов опису апаратури, таких як Verilog або VHDL. Потім система автоматизованого проєктування (CAD/EDA) компілює цей опис, інтерпретує його відповідно до специфікації ПЛІС, автоматично розміщує логічні функції в доступних логічних блоках та здійснює необхідне з'єднання між ними за допомогою внутрішніх програмованих комунікацій.

Таким чином, ПЛІС забезпечують універсальну платформу для створення цифрових пристроїв будь-якої складності, що відзначається гнучкістю, високою продуктивністю, швидкістю реалізації та зручністю у використанні завдяки широкому спектру інструментальних засобів розробки.

Мови опису апаратури набувають дедалі більшого поширення серед інженерів та розробників цифрових систем, особливо у сфері проєктування логіки для ПЛІС. Їх популярність пояснюється тим, що вони дозволяють чітко, формально та зручно описати функціональність цифрових пристроїв на рівні логічних операцій, структур і взаємозв'язків, з подальшою автоматичною трансляцією у конкретну конфігурацію ПЛІС.

Великий внесок у популяризацію таких мов зробила система

автоматизованого проектування MAX+PLUS II BASELINE, створена компанією Altera. Ця САПР підтримувала три основні мови опису апаратури: AHDL (Altera HDL), VHDL і Verilog, що надало користувачеві гнучкість у виборі підходу до проектування. Після появи MAX+PLUS II, саме AHDL швидко завоювала лідируючі позиції серед розробників, особливо на пострадянському просторі. За деякими джерелами, вона й досі зберігає найвищий рівень популярності в цьому регіоні. Такий успіх пояснюється двома основними факторами. По-перше, мова AHDL відзначається простотою синтаксису та чудово структурованою документацією, що особливо приваблює інженерів із традиційною інженерною освітою, зокрема тих, хто розпочинав роботу ще в добу дискретних логічних елементів. Інтерфейс довідкової системи давав змогу легко орієнтуватися в конструкціях мови, що значно полегшувало процес навчання та практичного використання. По-друге, на AHDL існує значна кількість навчальної та прикладної літератури, у тому числі орієнтованої на вітчизняного читача. На відміну від цього, мова VHDL, незважаючи на її розповсюдженість у міжнародній спільноті, впродовж довгого часу мала обмежене представлення в спеціалізованих джерелах на ринку пострадянських країн, що ускладнювало її освоєння для широкого кола розробників.

Слід також відзначити, що AHDL є внутрішньою (рідною) мовою системи. Це означає, що незалежно від того, в якому форматі створено проєкт – графічному, мовному (на VHDL чи Verilog) чи навіть у вигляді часових діаграм (Waveform Editor), компілятор спочатку транслює проєкт у проміжне представлення на AHDL. Лише після цього створюється фінальний файл прошивки з розширенням .prof, який може бути завантажений до кристалу ПЛІС. Таким чином, навіть при використанні альтернативних мов або візуальних редакторів, AHDL фактично виступає універсальним внутрішнім ядром трансляції, що підтверджує її значення у внутрішній логіці середовища. Це робить її знання корисним і в деяких випадках навіть необхідним для повноцінного освоєння екосистеми розробки від Altera.

3.2 Розробка HDL-моделі перспективного ЛКА ПД

3.2.1 HDL-моделі відомих ЛКА ПД

Для практичної реалізації пристроїв керування паралельної дії, за структурою що показана на рисунку 1.2 реалізовано HDL-модель (рисунок 3.1). Ця модель дозволяє виконати перехід від абстрактних моделей (наприклад математичної) і структур до практичного втілення таких моделей спеціалізованими інструментальними засобами розробки і моделювання кристалів ПЛІС, що розглянуті вище [7].

Для повного розуміння принципів побудови та функціонування HDL-моделі класичного ЛКА ПД необхідно детально розглянути основні функціональні блоки, що формують структуру цієї моделі, а також внутрішню організацію кожного блоку, їхнє функціональне призначення, виконувані завдання та результати, які вони формують у процесі моделювання. Такий аналіз є ключовим етапом, який дозволяє здійснити перехід до розробки HDL-моделі, що розробляється.

Одним з основних функціональних компонентів є блок пам'яті станів, який реалізується у HDL-проекті як модуль *bps:_bps*. Цей блок відповідає за зберігання комбінацій станів детермінованих входів, які є дозволеними (тобто логічно допустимими) для активації певних виконавчих механізмів у заданому циклі роботи агрегату. Таким чином, він є функціональним аналогом блоку БПС, описаного у структурі класичного ЛКА ПД.

HDL-блок *bps:_bps* містить:

- вхід *adr[q..1]*, що приймає адресу звернення до пам'яті і має ширину *q* біт;

- вихід *vec_Ai[1..k][1..2]*, який формує двобітну інформацію для кожного з *k* датчиків, що відображає дозволена комбінацію станів;

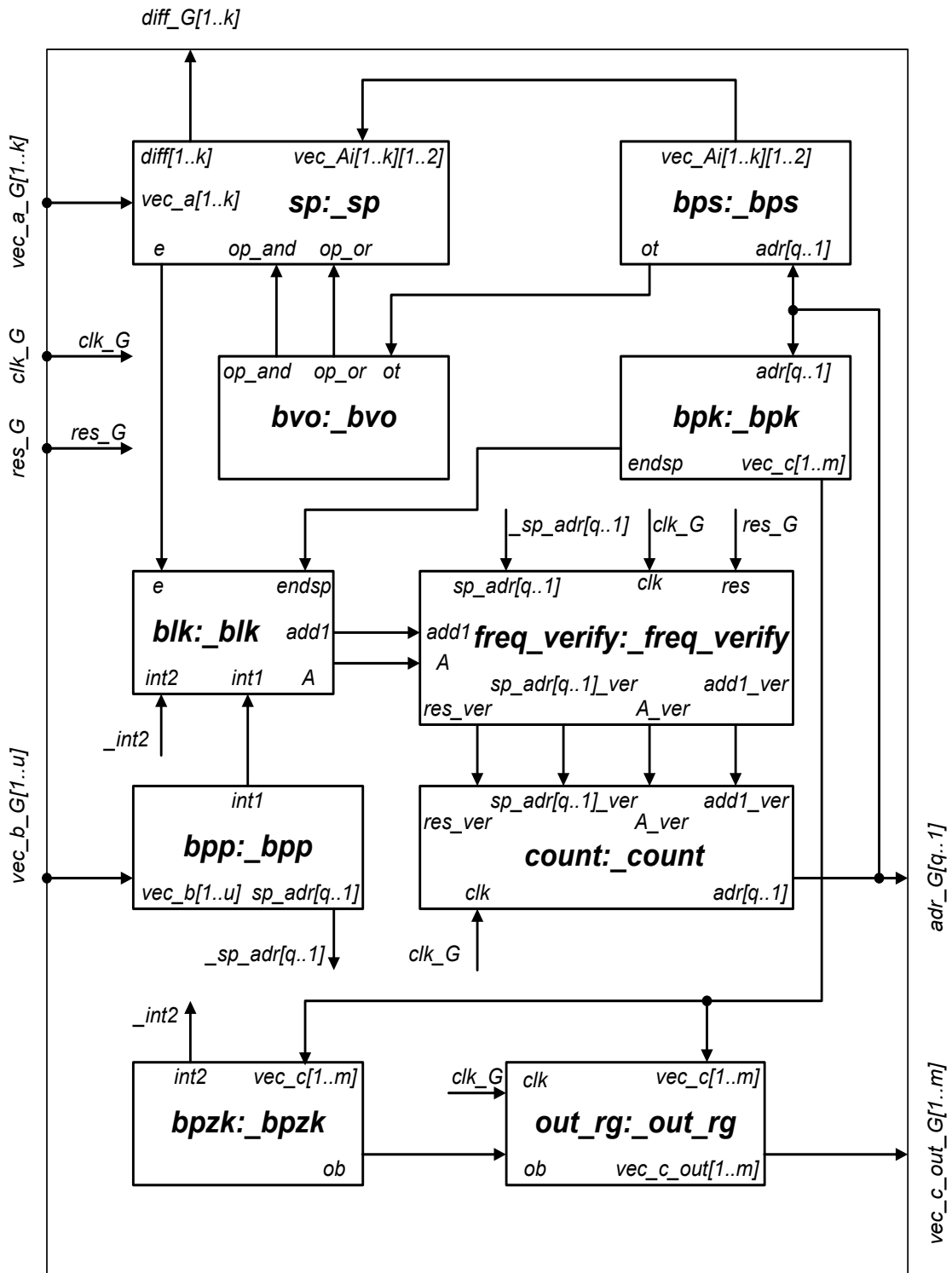


Рисунок 3.1 – HDL-модель сучасного ЛКА ПД

- додатковий вихід *ot*, який генерує логічний сигнал («0» або «1»), що визначає тип логічної операції, яка має виконуватись схемою порівняння (СП). Зокрема, це дозволяє перемикає СП на виконання логічної операції «І» чи «АБО», залежно від вимог поточного рядка програми.

Наступним модулем є блок пам'яті команд, реалізований у HDL-проекті під назвою *bpk:_bpk*. Цей компонент відповідає за зберігання вихідних команд, які формуються на основі аналізу вхідних сигналів. По суті, *bpk:_bpk* виконує функції класичного блоку БПК, який визначає, які саме механізми мають бути активовані на кожному кроці керуючої програми.

Склад HDL-модуль *bpk:_bpk*:

- вхід *adr[q..1]* для подачі адреси відповідного рядка в пам'яті;
- вихід *vec_c[1..m]*, який формує керуючі команди для *m* виконавчих механізмів;
- додатковий вихід *endsp*, що сигналізує ознаку завершення поточної підпрограми. Коли у векторі команд присутній маркер кінця підпрограми («КП»), *endsp* набуває значення логічної одиниці, що дозволяє контролеру перейти до іншої підпрограми або зупинити виконання.

У класичному ЛКА ПД вихід «КП» є єдиним і забезпечує базовий механізм переходів у програмах із розгалуженням. У HDL-моделі *bpk:_bpk* реалізує цю логіку у повній відповідності до архітектури ЛКА ПД, дозволяючи забезпечити коректний алгоритмічний перебіг у керуючому автоматі паралельної дії.

Функціональний блок *bpp:_bpp*, що реалізує блок пам'яті переходів (БПП) у структурі ЛКА ПД, призначений для зберігання та видачі початкової адреси підпрограми, яка має бути активована у відповідь на певну комбінацію сигналів. Цей блок реалізується у кристалі ПЛІС і має вхідний вектор даних *vec_b[1..u]* шириною *u* біт, який містить інформацію про поточний стан системи, а також вихід *sp_adr[q..1]*, що формує адресу початку відповідної підпрограми у форматі *q*-розрядного коду. У разі, якщо серед вхідних сигналів *vec_b[1..u]* присутня ознака аварійного переривання 1, блок формує сигнал логічної «1» на виході *int1*. Це ініціює перехід до підпрограми аварійного реагування, минаючи завершення поточної підпрограми, що є критично важливим для реалізації функцій безпеки.

Функціональний блок *bpzk:_bpzk* виконує роль блоку пам'яті

заборонених комбінацій (БПЗК). Аналогічно іншим елементам, він реалізується у ПЛІС і забезпечує запам'ятовування неприпустимих комбінацій вихідних команд, які не можуть бути видані на технологічний агрегат через можливість виникнення аварійної ситуації. Блок має вхід $vec_c[1..m]$, який приймає поточний вектор керувальних сигналів для m виконавчих механізмів, а також два виходи: $int2$ – сигнал аварійного переривання 2, який активується при виявленні забороненої комбінації та ob – логічний вихід, що використовується для блокування передачі забороненого вектору на фізичні виходи системи.

Функціональний блок sp_sp відповідає за виконання логічного порівняння між фактичними сигналами на вході та еталонними (очікуваними) значеннями. Для цього він приймає два набори вхідних даних: $vec_a[1..k]$ – поточні значення від k контрольованих входів, $vec_Ai[1..k][1..2]$ – очікувані двобітні вектори станів від блоку БПС. Для вибору типу логічної операції, що має виконуватись при порівнянні, передбачені входи op_and і op_or , які дозволяють перемикатись між логікою «І» (усі умови мають бути виконані) та логікою «АБО» (достатньо виконання хоча б однієї умови). Вихід e формує логічну одиницю у випадку, якщо обрана умова еквівалентності виконується згідно з вибраною логікою. Крім цього, вихід $diff[1..k]$ формує масив сигналів, що відображає результат порозрядного порівняння вхідних векторів, тобто інформує про невідповідність по кожному окремому біту. Ці сигнали можуть бути передані до блоку індикації для візуального контролю оператором або для використання в алгоритмах діагностики.

Функціональний блок bvo_bvo служить для реалізації механізму вибору типу логічної операції, що має виконуватись у блоці порівняння sp_sp . Як уже зазначалося раніше, порівняння вхідних сигналів із запрограмованими до блоку пам'яті станів значеннями може здійснюватися за двома логіками: «І» чи «АБО», саме для реалізації цього функціоналу в архітектурі ЛКА ПД і використовується блок bvo_bvo . Цей модуль має вхід

ot, що визначає тип логічної операції та два виходи: *op_and* і *op_or*, які, відповідно, приймають значення логічної «1» у разі вибору операції «І» чи «АБО». Ці сигнали передаються до схеми порівняння і визначають її режим роботи в конкретному такті автоматного часу.

Функціональний блок *count:_count* виконує роль лічильника адреси, який необхідний для послідовного адресування рядків у блоках БПС та БПК. Це ключовий елемент у забезпеченні правильного перебігу виконання підпрограм. Блок має: вхід *sp_adr_ver[q..1]*, який використовується для завантаження адреси переходу у разі перемикання між підпрограмами; вихід *adr[q..1]*, що формує поточну адресу, яка передається до інших функціональних блоків ЛКА ПД для вибірки відповідного рядка даних. Для забезпечення точного й стабільного функціонування блоку у складі системи передбачено кілька службових сигналів:

- *clk* – тактовий вхід для синхронізації роботи лічильника;
- *res_ver* – вхід команди скидання, що повертає лічильник у початковий стан;
- *A_ver* – вхід команди паралельного завантаження нової адреси;
- *add1_ver* – вхід команди інкременту, який ініціює додавання одиниці до поточної адреси.

Функціональний блок *blk:_blk* відповідає за логічне керування основним процесом функціонування ЛКА ПД. Його основна роль полягає у формуванні сигналів керування для лічильника адреси, зокрема визначенні, чи слід перейти до наступного кроку поточної підпрограми або здійснити перехід до іншої підпрограми. Це рішення приймається на основі аналізу сигналів, що надходять із суміжних функціональних блоків системи. Блок *blk:_blk* формує централізовану логіку переходів у програмі ЛКА ПД, реагуючи як на штатні, так і на аварійні ситуації, забезпечуючи правильний перебіг обробки підпрограм. Блок працює з наступними вхідними та вихідними сигналами:

- вхід *e* – сигнал еквівалентності (інформує про збіг поточних вхідних

даних з очікуваними);

- вхід *endsp* – індикація завершення підпрограми;
- входи *int1* та *int2* – сигнали аварійних переривань першого та другого типів відповідно;
- вихід *A* – сигнал на паралельне завантаження адреси до лічильника;
- вихід *add1* – сигнал, що ініціює інкрементацію адреси (перехід до наступного рядка).

Функціональний блок *out_rg:out_rg* реалізує вихідний регістр, що відповідає за буферизацію та передачу команд керування до виконавчих механізмів технологічного агрегату. Його основне завдання – тимчасово зберігати сформовані команди та подавати їх на вихід, якщо не виявлено заборонених комбінацій. У ситуаціях, коли БПК з тієї або іншої причини генерує заборонену комбінацію (наприклад, через логічну помилку, пошкодження структури або критичний збій), *out_rg:out_rg* блокує подачу цих команд на вихід, тим самим запобігаючи аварійному впливу на ТА. Структура блоку включає:

- вхід *vec_c[1..m]* – команди керування від БПК;
- вихід *vec_c_out[1..m]* – команди, що передаються на виконавчі механізми;
- вхід *clk* – тактовий сигнал синхронізації;
- вхід *ob* – сигнал блокування виходу при виявленні недопустимої комбінації.

Функціональний блок *freq_verify:freq_verify* є модулем багатократного контролю достовірності сигналів, який відіграє важливу роль при практичній реалізації промислових зразків ПЛС-контролерів паралельної дії. Цей блок здійснює багатократну (наприклад, 8-кратну, або загалом *n*-кратну) перевірку коректності ключових сигналів керування, що формуються в процесі роботи ЛКА ПД. Цей контроль дозволяє виявити можливі збої, імпульсні спотворення або некоректні комбінації, які можуть виникати як через внутрішні порушення, так і через зовнішні впливи. Блок

freq_verify: freq_verify значно підвищує надійність та безпеку ПЛІС-контролера, особливо в умовах експлуатації на об'єктах з підвищеними вимогами до функціональної стійкості.

Перевірки підлягають:

- сигнали від блоку логічного керування (*blk: blk*);
- сигнали скидання ЛКА;
- сигнали переходу між підпрограмами.

Структурна схема цього блоку наведена на рисунку 3.2.

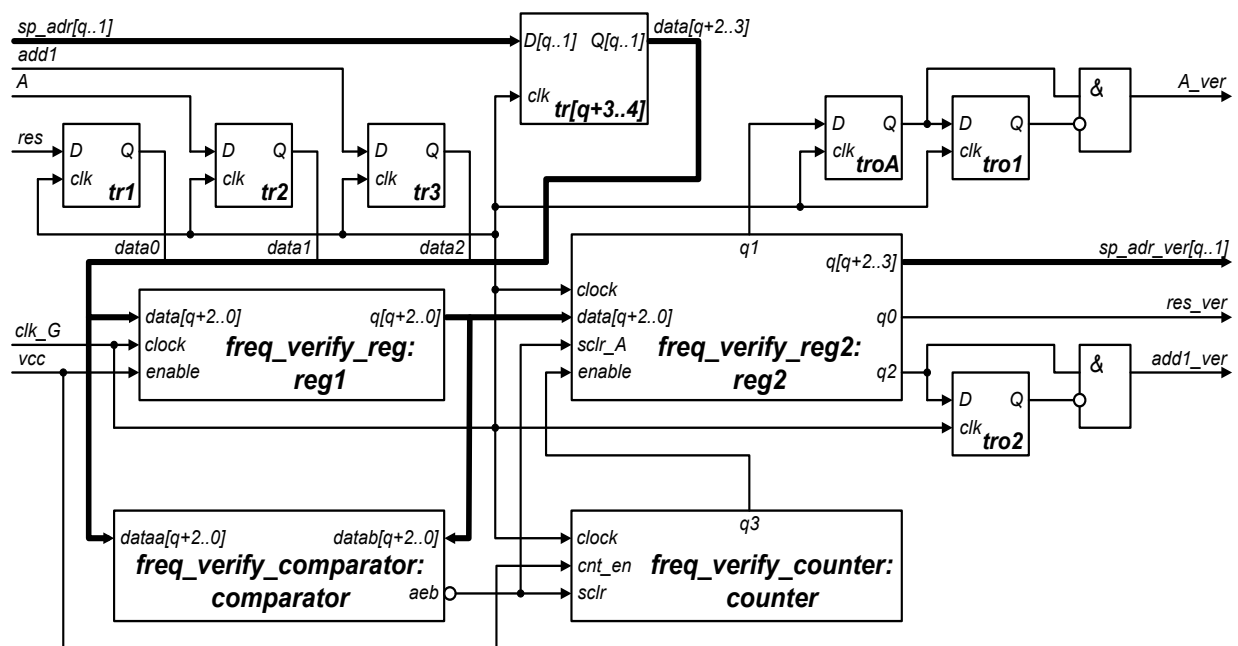


Рисунок 3.2 – Структурна схема блоку *freq_verify: freq_verify*

Для кращого розуміння призначення та логіки роботи блоку *freq_verify: freq_verify*, необхідно описати його внутрішню структуру, а також сигнали, що надходять на входи та формуються на виходах.

Основним елементом блоку є компонент *freq_verify_reg: reg1*, який являє собою паралельний регістр розрядністю $(q+3)$. До його складу входить вхід синхронізації *clock*, вхід дозволу запису *enable*, а також інформаційні входи та виходи відповідної ширини – *data[q+2..0]* та *q[q+2..0]*. Три молодші розряди цього регістра використовуються для зберігання логічних сигналів, пов'язаних із керуванням перебігом підпрограм – це сигнали *res*

(скидання), A (завантаження адреси) та $add1$ (інкремент адреси). Інші, старші q розрядів, призначені для фіксації адреси переходу $sp_adr[q..1]$, яка визначає точку входу до нової підпрограми у структурі ЛКА ПД.

Інший компонент – $freq_verify_reg2:reg2$ виконує функцію збереження достовірного стану сигналів керування. За структурою цей регістр аналогічний попередньому, має ту ж розрядність ($q+3$), однак відрізняється наявністю додаткового входу $sclr_A$, який виконує функцію асинхронного скидання лише одного конкретного біта – розряду, що відповідає за команду паралельного завантаження A_ver . У цьому регістрі старші q розрядів зберігають перевірену адресу переходу $sp_adr_ver[q..1]$, тоді як молодший розряд, $q0$, відповідає за сигнал достовірної команди скидання res_ver .

Загалом, обидва регістри працюють у тандемі для багаторазової перевірки та підтвердження коректності важливих сигналів логіки керування, забезпечуючи підвищену надійність роботи ЛКА ПД і зменшуючи ймовірність помилок, викликаних імпульсними перешкодами або помилковими комбінаціями сигналів.ф

Компонент $freq_verify_comparator:comparator$ виконує розширені функції, аналогічні схемі порівняння, і слугує для виявлення змін у станах вхідних сигналів. Його основне завдання полягає в тому, щоб фіксувати факт відмінності між двома наборами даних $dataa[q+2..0]$ і $datab[q+2..0]$. Якщо між цими наборами виявлено розбіжність, на інверсному виході aeb з'являється логічна одиниця. Таким чином, цей компонент сигналізує про будь-яке відхилення вхідних даних від очікуваних значень, що може свідчити про збої або нестабільність сигналів.

Компонент $freq_verify_counter:counter$ являє собою чотирирозрядний лічильник, який використовується для контролю сталості сигналів у часі. Його робота синхронізується через вхід $clock$, активується через вхід cnt_en , а також може бути скинута за допомогою сигналу $sclr$. Вихід $q3$ переходить у логічну одиницю лише в тому випадку, якщо вхідні дані залишаються незмінними протягом певного наперед визначеного періоду. Це дозволяє

переконалися в тому, що сигнали є стабільними і не містять короткочасних збурень або перешкод.

Компонент *tr* (від *tr1* до *tr[q+3]*) реалізований на базі синхронних D-тригерів і призначений для узгодження сигналів ЛКА ПД – *res*, *A*, *add1* та *sp_adr[q..1]* із глобальним тактовим сигналом *clk_G*. Його присутність пояснюється особливостями архітектури кристалу ПЛІС та специфікою синхронної логіки. При цьому сам компонент не змінює логіку функціонування ЛКА ПД, а виконує виключно технічну функцію синхронізації. Він містить типові входи *D* і *clk* та вихід *Q*.

Компонент *tro* (зокрема *tro1* і *tro2*) також побудований на базі синхронних D-тригерів. У поєднанні з логічними елементами типу «І», він забезпечує формування достовірних сигналів *ver_A* і *ver_add1*, тривалістю один такт глобального сигналу синхронізації *clk_G*. Як і попередній компонент, *tro* виконує допоміжну функцію, не впливаючи безпосередньо на логіку роботи ЛКА ПД.

Компонент *troA*, аналогічно до *tr*, затримує формування команди *A_ver* на один такт глобального синхросигналу *clk_G*. Його основна мета – забезпечення точного позиціонування у таймінговій структурі системи, з огляду на архітектурні особливості ПЛІС. Цей компонент, як і інші тригери, не змінює логічної поведінки керуючого автомата, а лише відповідає за синхронізацію на рівні сигналів у цифровій системі.

На рисунку 3.3 показано HDL-модель ЛКА ПД, що реалізують функціонал програмованих користувачем таймерів, тобто реалізують структуру, що показана на рисунку 1.5 кваліфікаційної роботи. Саме ця модель є останньою з відомих на даний час.

Однією з ключових змін, що були внесені до HDL-моделі ПЛІС-контролера паралельної дії, стало розширення її структури шляхом інтеграції нових функціональних елементів, призначених для реалізації додаткових блоків, передбачених у вдосконаленій архітектурі, показаній на рисунку 1.5. Зокрема, у структурі контролера було передбачено два додаткових

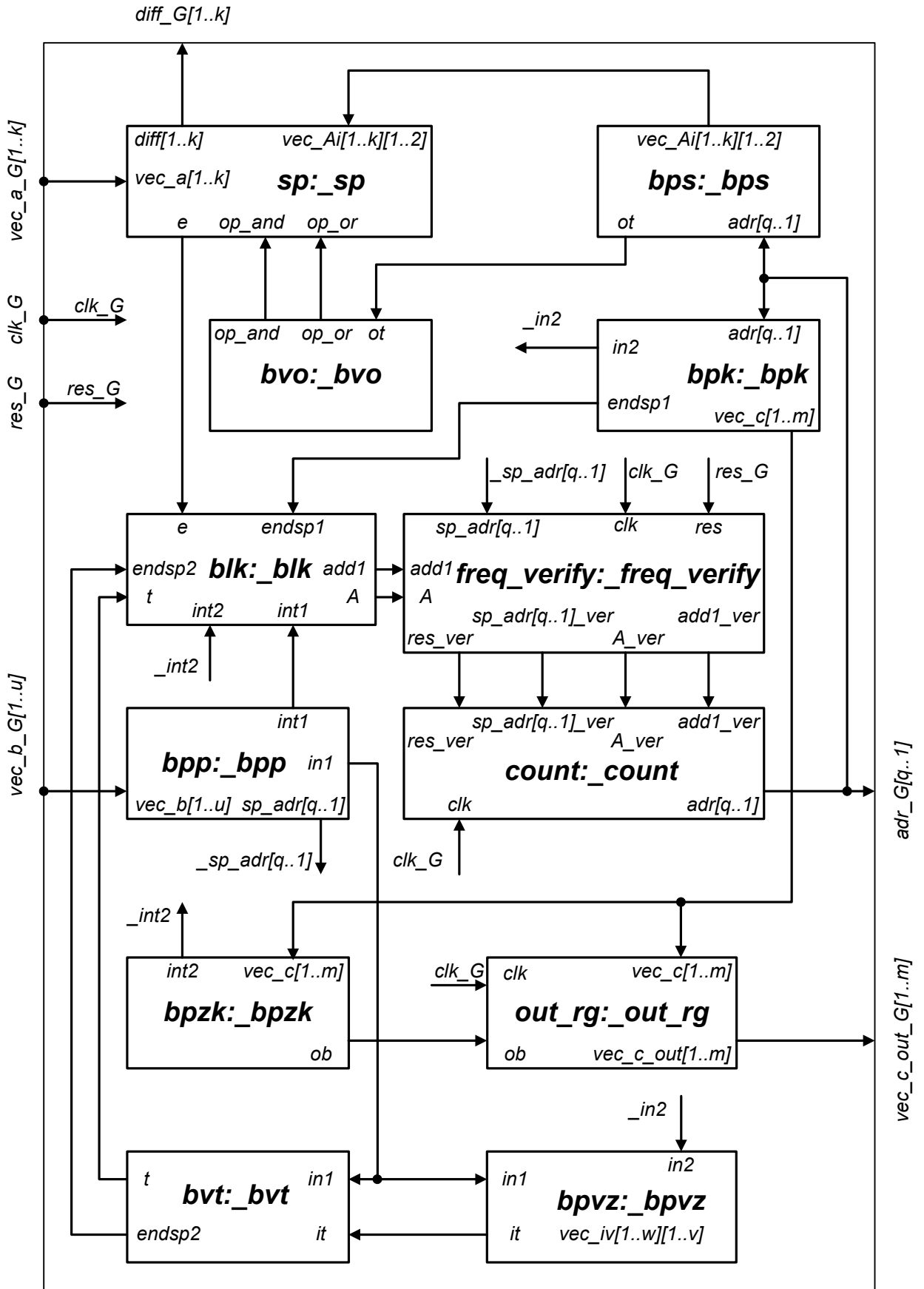


Рисунок 3.3 – HDL-модель ЛКА ПД з програмованими користувачем таймерами

компоненти – блок пам'яті внутрішніх змінних (БПВЗ) і блок внутрішніх таймерів (БВТ). Для їх реалізації у складі HDL-моделі додано відповідні функціональні блоки з ідентифікаторами *bpvz: _bpvz* і *bvt: _bvt*.

Крім появи нових блоків, зміни також торкнулися вже наявних функціональних компонентів HDL-моделі. Вони проявилися у формуванні нових входів, виходів, а також у перерозподілі або уточненні призначення окремих внутрішніх сигналів. Зокрема, це стосується блоку логічного керування *blk: _blk*, блоку пам'яті переходів *bpp: _bpp* і блоку пам'яті команд *bpk: _bpk*. Розширення інтерфейсів цих елементів забезпечує підтримку логіки взаємодії з новими функціональними модулями, дозволяючи гнучко керувати процесами переходів, обробки внутрішніх змінних і керування часовими інтервалами.

Для повного уявлення про оновлену HDL-модель доцільно детально розглянути, яким чином кожен функціональний блок взаємодіє з іншими складовими системи, а також яке функціональне навантаження він виконує у процесі реалізації алгоритмів логічного керування в структурі паралельного ПЛІС-контролера.

У функціональному блоці *bpk: _bpk*, який виконує роль блоку пам'яті команд, було впроваджено низку змін, що розширюють його функціональність у порівнянні з базовою моделлю, розглянутої раніше. Одним із важливих нововведень є оновлення сигналу, що вказує на завершення виконання поточної підпрограми. Якщо у класичній моделі цей сигнал мав позначення *endsp*, то в новій HDL-реалізації він отримав позначення *endsp1*. Це пов'язано з тим, що у вдосконаленій структурі перехід до нової підпрограми може здійснюватись не лише після завершення попередньої, а й у результаті впливу додаткових умов, наприклад, після відпрацювання внутрішнього програмованого таймера. Отже, механізм переходу набуває більшої гнучкості. Крім того, до кожного рядка блоку пам'яті команд може бути записано спеціальну команду, яка ініціює запуск внутрішнього таймера. Це нововведення реалізовано у HDL-моделі через

додатковий вихід *in2*, що з'явився у структурі блоку *bpk:_bpk*. Його призначення – передача команди на запуск таймера, який далі бере участь у логіці керування переходами та часовими затримками в рамках паралельної архітектури. Отже у новій HDL-моделі блок *bpk:_bpk* набуває розширених можливостей для підтримки більш складної логіки керування, зокрема за рахунок взаємодії з новими функціональними модулями, такими як внутрішні таймери та блоки змінних.

Функціональний блок *blk:_blk*, який відповідає за логічне керування в архітектурі ЛКА ПД, зазнав суттєвих змін у порівнянні з класичною HDL-моделлю. Однак з метою спрощення побудови та опису оновленої структури контролера частину логіки, пов'язаної з обробкою програмованих таймерів, було винесено з цього блоку в окрему функціональну одиницю – блок внутрішніх таймерів (БВТ). Тому у даному описі зосередимо увагу лише на тих змінах, яких зазнав *blk:_blk* у межах загальної логіки роботи контролера.

Основні логічні входи блоку, такі як *e*, що сигналізує про еквівалентність станів, а також *int1* та *int2*, які відповідають за виявлення аварійних ситуацій першого та другого типів, зберегли свою функціональність без змін. Аналогічно, виходи *A* та *add1*, які керують адресним лічильником, здійснюють паралельне завантаження нової адреси та інкремент поточної – продовжують виконувати ті самі функції, що й у класичному ЛКА ПД. Натомість вхід *endsp*, який раніше позначав завершення підпрограми, отримав нову назву – *endsp1*. Ця зміна обумовлена тим, що тепер ініціація переходу до іншої підпрограми може бути викликана не лише сигналом з блоку пам'яті переходів, як це було раніше, а й сигналом від програмованого таймера, який працює паралельно до основної логіки контролера.

Блок *bvt:_bvt* реалізує функціональність внутрішніх таймерів і є логічним розширенням архітектури, що дозволяє інтегрувати часову логіку до загальної структури ЛКА ПД. Цей блок приймає сигнал *it* – команду на запуск таймера, після чого, по завершенню встановленого інтервалу часу,

формує один із двох вихідних сигналів: t або $ends2$. Вибір сигналу залежить від того, який саме блок – пам'яті команд (bpk_bpk) чи пам'яті переходів (bpp_bpp) був ініціатором запуску таймера. Ця інформація подається на вхід $in1$: логічна одиниця означає, що ініціатором є bpp_bpp , отже, необхідно згенерувати сигнал $ends2$, що трактується як завершення підпрограми. Якщо ж на вході $in1$ присутній логічний нуль, це означає, що запуск здійснено блоком bpk_bpk , і формування сигналу t забезпечує перехід до наступного рядка в межах тієї ж підпрограми.

Блок $bpvz_bpvz$ виконує роль пам'яті внутрішніх змінних і фактично є місцем зберігання числових значень для проміжків часу, які обробляються таймерами. Крім основної функції збереження, блок має два входи – $in1$ і $in2$, які дозволяють ідентифікувати, хто саме ініціював запуск відліку часу: блок пам'яті переходів або пам'яті команд. Завершення таймерного інтервалу фіксується появою логічної одиниці на виході it , що слугує підтвердженням готовності до переходу або подальшого виконання підпрограми. Варто також підкреслити, що архітектура $bpvz_bpvz$ є достатньо універсальною – у вона може бути адаптована не лише для реалізації таймерів, а й для побудови лічильників, або зберігання фіксованих констант і допоміжних даних, що розширює потенціал застосування блоку в інших функціональних модулях контролера.

У сукупності ці зміни демонструють поступовий перехід від жорстко структурованої логіки класичного ЛКА ПД до більш гнучкої, модульної та адаптивної архітектури, що здатна підтримувати як стандартні алгоритми керування, так і складні ситуації з часовими та логічними залежностями.

3.2.2 HDL-модель МЛКА ПД

Проведений аналіз HDL-моделей класичного ЛКА ПД та ЛКА з програмованими таймерами, показує, що їх можна узяти за основу при реалізації HDL-моделі перспективного МЛКА ПД. Проте, якщо розглядати у

якості бази для такої реалізації, структуру, що показана на рисунку 2.1, то стає зрозуміло, що значна частина елементів є перспективним, тобто такими для яких ще не надано обґрунтування у вигляді математичної моделі або повного опису їх можливого функціонування. До таких елементів слід віднести блок БРАО – реалізації арифметичних операцій, та БЛ – блок лічильників. Це не є недоліком показаної структури, а вказує на те, що ми знаходимось лише на самому початку шляху у розробці перспективного напрямку реалізації нових структур керуючих автоматів у складі інформаційної технології паралельного керування.

Таким чином для побудови HDL-моделі МЛКА ПД за основу буде взято структуру, що наведено на рисунку 2.1, але з урахуванням того, що на даний час існують розробки і обґрунтування лише для базових елементів ВНЕ – БПП, БПС, БПК та БПЗК, а також елементи, що реалізують функціонал внутрішніх програмованих таймерів – БВТ. Отже, з урахуванням вищевказаного HDL-модель модульного керуючого автомата паралельної дії, показано на рисунку 3.4. Розглянемо більш детально тільки принципові зміни та доповнення, що з'явилися у наведеній моделі.

Як видно з рисунку, у порівнянні з базовою моделлю, на ній відсутній функціональний блок вибору типу операції *bvo:_bvo*. Це пов'язано з тим, що як показано у [3, 4] цей блок фактично призначений для обґрунтування і детального опису розширених функціональних можливостей сучасного ЛКА ПД, у порівнянні з ППЛК попередніх поколінь. Насправді тип операції порівняння «І» чи «АБО» може бути просто заданий бінарним сигналом, наприклад «І» – нулем, а «АБО» – одиницею. На сучасному етапі розвитку ІТПЛК вказаної деталізації опису вже не потрібно, отже відпала і необхідність к блоці ТО, що зреалізовувався блоком *bvo:_bvo* HDL-моделі. Отже тепер вибір типу операції порівняння вектору вхідних станів з їх очікуваними значеннями задається внутрішнім бінарним сигналом *op_type* від функціонального блоку *bps:_bps* до блоку *sp:_sp*, що і показано на рисунку 3.4.

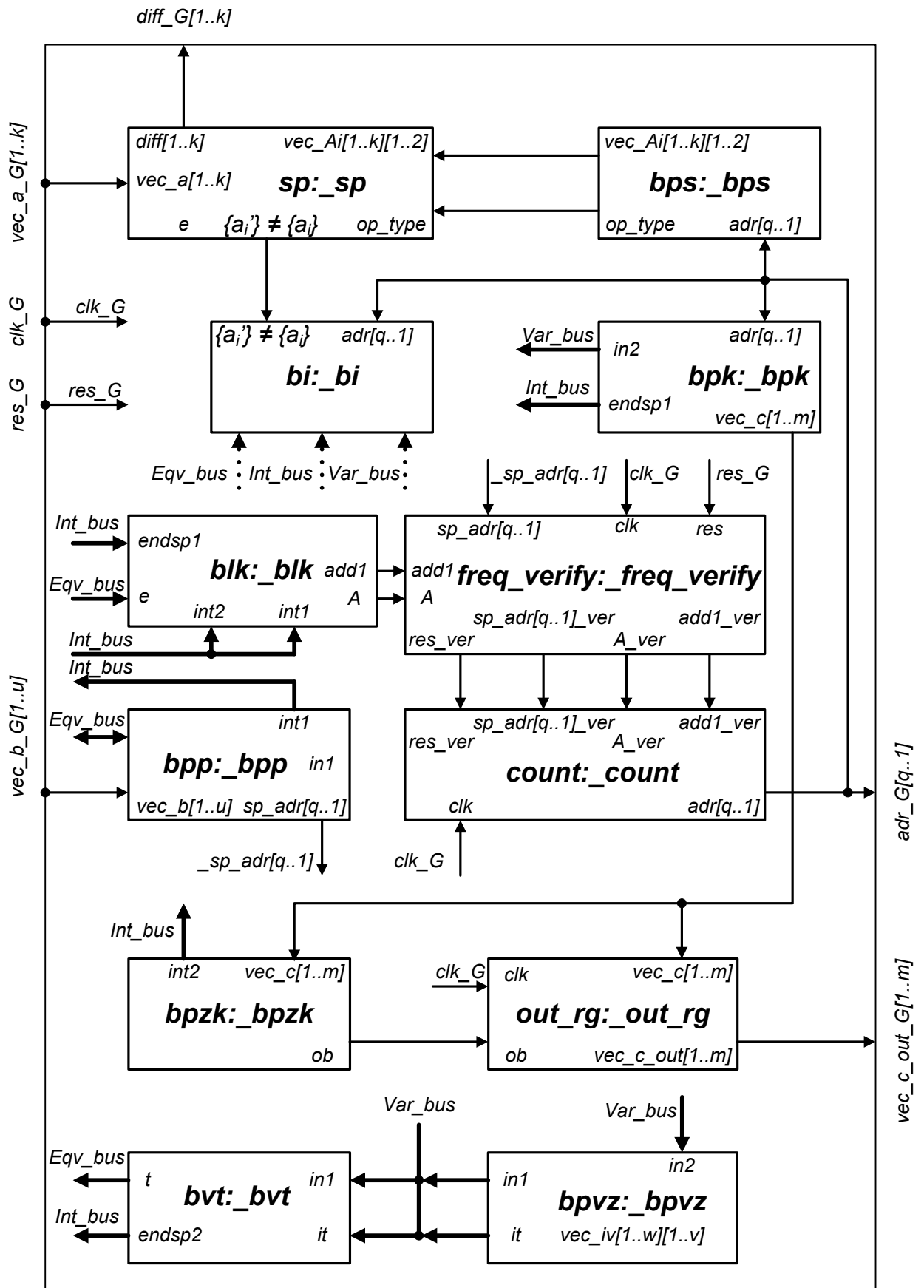


Рисунок 3.4 – HDL-модель модульного логічного керуючого автомату паралельної дії – МЛКА ПД

Ще однією відмінністю від попередньої моделі є поява функціонального блоку індикації bi_bi , що призначений для реалізації однойменного блоку структури МЛКА ПД. Як видно з моделі він реалізує відображення стану нееквівалентності вхідного вектору $\{ai'\} \neq \{ai\}$ та адреси $adr[q..1]$, що фактично є номером рядка програми. Саме ці функції покладені на відповідний блок БІ у структурах, що були розглянуті у першому розділі кваліфікаційної роботи. Проте, як видно з рисунка 3.4, функціональний блок bi_bi може відображати також додаткову інформацію: стан на шинах Int_bus , Eqv_bus , Var_bus , що показано пунктирною лінією і обґрунтування чому було наведено у другому розділі роботи.

Тепер розглянемо найбільш важливу відмінність HDL-моделі МЛКА ПД від її більш ранніх версій – це поява трьох вже згаданих шин Int_bus – шини переривань, Eqv_bus – шини еквівалентностей, та Var_bus – шини внутрішніх змінних. Повторювати їх місце і призначення у структурі немає сенсу, це було детально розглянуто у другому розділі роботи при описі структури МЛКА ПД. Проте зазначимо, що уведення до HDL-моделі цих шин дозволяє «розвантажити» її, та надає можливість зробити її більш структурованою, що і буде зроблено у подальших дослідженнях. Якщо говорити про поточний варіант моделі у порівнянні з попередньою (рисунок 3.3), то можна помітити, що внутрішні сигнали переривань $in2$, $int1$, $endsp1$ та $endsp2$ тепер передаються шиною Int_bus , сигнали e і t шиною еквівалентностей – Eqv_bus , а внутрішня змінна $in2$ тепер є складовою сигналів шини Var_bus . Сам загальний вигляд моделі було вирішено таким, що відповідає рисунку 3.3, для надання можливості простої оцінки прийнятих рішень.

Таким чином реалізована HDL-модель МЛКА ПД повністю відповідає його структурі (рисунок 2.1) і може бути взята за основу для створення інших перспективних керуючих автоматів у рамках інформаційної технології паралельного логічного керування.

ВИСНОВКИ

У процесі виконання кваліфікаційної роботи було проведено аналіз відомих архітектур та математичної моделі керуючих автоматів паралельної дії: розглянуто елементи математичної моделі класичного ЛКА ПД, наведено та досліджено відомі на даний час структури керуючих пристроїв з паралельною архітектурою. Виконано розробку структури перспективного ЛКА ПД, що отримав назву модульний логічний керуючий автомат паралельної дії – МЛКА ПД. Досліджено HDL-моделі відомих ЛКА ПД та запропоновано варіант реалізації такої моделі для МЛКА ПД.

У результаті виконаних досліджень усунуто такий недолік керуючих пристроїв з паралельною архітектурою, як їх фактична побудова за принципом «жорсткої логіки», або недовикористання можливостей створення «гнучких» систем керування на базі сучасних кристалів ПЛІС.

Запропонована концепція «модульної архітектури» керуючих пристроїв паралельної дії дозволяє створювати «гнучку» структуру логічних керуючих автоматів. Саме на базі цієї концепції з'являється можливість синтезувати структури МЛКА із заданою функціональністю у залежності від конкретних задач системи керування, що будується на його основі.

Результати наукових досліджень кваліфікаційної роботи опубліковано у збірнику наукових праць «Системи управління навігації та зв'язку» випуск 2 (80) 2025 року [22].

Таким чином усі поставлені завдання дослідження було виконано у повному обсязі.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Бовчалоук С. Я. Вдосконалення архітектури програмованих логічних контролерів паралельної дії / С. Я. Бовчалоук, І. О. Фурман, М. Л. Малиновський // Проблеми енергозабезпечення та енергозбереження в АПК України: Вісник ХНТУСГ імені Петра Василенка, вип. 37, том 2 – Харків, 2005. – С. 164–168.
2. Бовчалоук С. Я. Вдосконалення архітектури, мови програмування та технічної реалізації ПЛК паралельної дії // Міжнародна науково-технічна конференція «Інтегровані комп'ютерні технології в машинобудуванні ІКТМ-2005»: Тези доповідей. – Харків: Національний аерокосмічний університет «Харківський авіаційний інститут», 2005. – С. 290.
3. Бовчалоук С. Я. Вдосконалення математичної моделі та архітектури логічних керуючих автоматів паралельної дії / С. Я. Бовчалоук, І. О. Фурман // Інформаційно-керуючі системи на залізничному транспорті. – 2006. – №3(59). – С. 72–76.
4. Бовчалоук С. Я. Моделі, методи і засоби інформаційної технології паралельного логічного керування об'єктами залізничної автоматики: дис. ... канд. техн. наук: 05.13.06 / Бовчалоук Станіслав Ярославович. – Харків, 2008. – 203 с.
5. Бовчалоук С. Я. Визначення напрямків розвитку керуючих пристроїв з паралельною архітектурою на базі ПЛІС / С. Я. Бовчалоук, О. М. Піскар'юв, С. С. Радченко, [та ін.] // Системи управління, навігації та зв'язку. Збірник наукових праць. – Полтава: ПНТУ, 2023. – Випуск 1 (71). – С. 69-72 DOI: 10.26906/SUNZ.2023.1.069.
6. Фурман І. О. Концепція, методи і засоби моделювання на ПЛІС контролерів і процесорів з паралельною архітектурою / І. О. Фурман, В. А. Краснобаєв, С. Я. Бовчалоук [та ін.] // Автомобільний транспорт: Зб. наукових праць, вип. 16. – Харків, 2005. С. 338–341.

7. Бовчалоук С. Я. HDL-модель програмованого логічного керуючого автомата паралельної дії / С. Я. Бовчалоук, І. О.Фурман // *Радіоелектронні і комп'ютерні системи*. – 2007. – №6 (25). – С. 202–205.

8. Бовчалоук С. Я. Технічна реалізація промислового зразка ПЛІС-контролера паралельної дії / С. Я. Бовчалоук, І. О.Фурман, М. С. Деренько [та ін.] // *Проблеми енергозабезпечення та енергозбереження в АПК України: Вісник ХНТУСГ імені Петра Василенка*, вип. 87. – Харків, 2009. – С. 126–127.

9. Бовчалоук С. Я. Методологія побудови, синтаксис і семантика мови програмування ПЛІС-контролерів паралельної дії / С. Я. Бовчалоук, І. О. Фурман // *Інформаційно-керуючі системи на залізничному транспорті*. – 2007. – № 4 (66). – С. 38–44.

10. Бовчалоук С. Я. Концепція розробки технологічної мови для програмування ПЛІС-контролерів паралельної дії / С. Я. Бовчалоук, І. О. Фурман // *Проблеми енергозабезпечення та енергозбереження в АПК України: Вісник ХНТУСГ імені Петра Василенка*, вип. 57, том 2 – Харків, 2007. – С. 133–138.

11. Ilya Furman. Development and study of technological visual programming of logic control problems / Ilya Furman, Stanislav Bovchaliuk, Alexander Allashev, Aleksey Piskarev // *Eastern-European Journal of Enterprise technologies*, – 2017. – № 6/2 (90). –P. 23–31. DOI: 10.15587/1729-4061.2017.118833/

12. Тимчук С. О. Реалізація функцій нечіткого логічного керування на базі автомата паралельної дії для інтелектуальних електричних мереж / С. О. Тимчук, С. Я. Бовчалоук, І. О. Фурман [та ін.] // *Проблеми енергозабезпечення та енергоз-береження в АПК України: Вісник ХНТУСГ імені Петра Василенка*, вип. 196. – Харків, 2018. – С. 86–88.

13. Serhii Tymchuk. The Architecture of Fuzzy Logic Automat of Parallel Action for the Intelligent Smart Grid Networks / Serhii Tymchuk, Stanislav Bovchaliuk, Vira Shendryk, Sergii Shendryk *New Technologies, Development and*

Application III. NT 2020. Lecture Notes in Networks and Systems, vol. 128. Springer, – 2020. – P. 462–468. DOI:10.1007/978-3-030-46817-0_53.

14. Бовчалоук С. Я. Розвиток моделі та структури керуючих пристроїв з паралельною архітектурою / С. Я. Бовчалоук, Я. В. Гаращенко, Б. М. Коломоєць [та ін.] // Системи управління, навігації та зв'язку. Збірник наукових праць. – Полтава: ПНТУ, 2024. – Випуск 2 (76). – С. 64-66. DOI: 10.26906/SUNZ.2024.2.064.

15. Фурман І. О. Науково-технічні основи створення і промислового застосування програмованих логічних контролерів на програмованих ВІС з матричною структурою: дис. ... докт. техн. наук: 05.13.05 / Фурман Ілля Олександрович. – К., 1989. – 197 с.

16. Бовчалоук С. Я. Вдосконалення алгоритму функціонування програмованого логічного контролера паралельної дії / С. Я. Бовчалоук, І. О. Фурман // Інформаційно-керуючі системи на залізничному транспорті. – 2007. – №2 (64). – С. 38–42.

17. Бовчалоук С. Я. Розвиток архітектури ПЛК паралельної дії: від абстрактної моделі паралельного автомата до інженерної реалізації безпечного ПЛІС-контролера / С. Я. Бовчалоук, І. О. Фурман, М. Л. Малиновський // Енергетика та комп'ютерно-інтегровані технології в АПК. 2016. – №. 2 (5). – С. 62-66.

18. Akushskiy I, Jiuditskiy D. Mashinnaia arifmetika v ostatochnikh klassax: M., Radio i Sviaz, 1968. 444 с.

19. Краснобаєв В. А. Основні властивості непозиційної системи числення у класі лишків і їх вплив на структуру та принципи реалізації арифметичних операцій комп'ютерної системи / В. А. Краснобаєв, С. О. Кошман, В. М. Курчанов [та ін.] // Системи управління, навігації та зв'язку, 2019. – Вип. 2(54) – С. 114-118.

20. Krasnobayev, V., Koshman, S., & Kovalchuk, D. (2022). THE CONCEPT OF PERFORMING THE ADDITION OPERATION IN THE SYSTEM OF RESIDUAL CLASSES. *Advanced Information Systems*, 6(1), 43–

47. DOI: 10.20998/2522-9052.2022.1.07.

21. Фурман І. А. Перспективи розвитку структури і технології застосування паралельних логічних контролерів/ І. А. Фурман // Електротехніка. – 1990. – №4. – С. 98-100.

22. Дрозд О. Ю. Розвиток моделі та структури керуючих пристроїв з паралельною архітектурою / О. Ю. Дрозд, С. Я. Бовчалюк, Н. І. Бовчалюк // Системи управління, навігації та зв'язку. Збірник наукових праць. – Полтава: ПНТУ, 2025. – Випуск 2 (80). – С. 47-53. DOI: 10.26906/SUNZ.2025.2.064.