

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет комп'ютерних наук і технологій

(повне найменування інституту, факультету)

Кафедра комп'ютерних систем та мереж

(повне найменування кафедри)

Пояснювальна записка

до дипломного проекту (роботи)

бакалавра

(ступінь вищої освіти)

на тему РОЗРОБКА КОНТРОЛЕРА СЕРВОПРИВОДІВ ДЛЯ
РОБОТОТЕХНІЧНИХ ЗАСТОСУВАНЬ НА FPGA

Виконав: студент(ка) 4 курсу, групи КНТ-522

Спеціальності 123 «Комп'ютерна інженерія»
(код і найменування спеціальності)

Освітня програма (спеціалізація)

«Комп'ютерна інженерія»

КРОТОВ К.О.
(прізвище та ініціали)

Керівник ГРУШКО С.С.
(прізвище та ініціали)

Рецензент ХАРИТОНОВ О.Б.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»
(повне найменування вищого навчального закладу)

Факультет Комп'ютерних наук і технологій
Кафедра «Комп'ютерні системи та мережі»
Ступінь вищої освіти бакалаврський
Спеціальність 123 Комп'ютерна інженерія
(код і найменування)
Освітня програма (спеціалізація) Комп'ютерна інженерія
(назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ
Завідувач кафедри КУДЕРМЕТОВ Р.К.

“ ” 2026 року

З А В Д А Н Н Я
НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА

КРОТОВА Кирила Олександровича
(ПРИЗВИЩЕ, ім'я, по батькові)

1. Тема проєкту (роботи) Розробка контролера сервоприводів для робототехнічних застосувань на FPGA

керівник проєкту (роботи) к. т. н., доцент, ГРУШКО Світлана Сергіївна
(науковий ступінь, вчене звання, ПРИЗВИЩЕ, ім'я, по батькові)

затверджені наказом вищого навчального закладу від “14” квітня 2026 року № 160

2. Строк подання студентом проєкту (роботи) 30 травня 2026 року

3. Вихідні дані до проєкту (роботи) FPGA; модульна архітектура з підтримкою генерації ШІМ-сигналів, декодування квадратурного енкодера в режимі 4х, цифрового ПІД-регулювання положення з арифметикою фіксованої точки та комунікаційного інтерфейсу UART.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

1) Аналіз предметної області та постановка задачі;

2) Проектування контролера сервоприводів;

3) Реалізація та експериментальне дослідження;

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

Слайди презентації

6. Консультанти розділів проєкту (роботи)

Розділ	ПРІЗВИЩЕ, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-4	ГРУШКО С. С., к.т.н., доцент		
нормоконтроль	ГОЛУБ Т.В., к.т.н., доцент		

7. Дата видачі завдання 14.04.2026 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Аналіз технічного завдання. Огляд сервоприводів, методів керування та порівняльний аналіз реалізацій контролерів на MCU та FPGA	14.04.26 – 20.04.26	
2	Огляд наявних комерційних та академічних рішень. Обґрунтування вибору апаратної платформи та середовища розробки	20.04.26 - 25.04.26	
3	Проектування модульної архітектури контролера: структурна схема, модулі ШІМ, енкодера, ПІД-регулятора, UART	25.04.26 - 10.05.26	
4	Реалізація модулів контролера мовою VHDL	10.05.26 – 20.05.26	
5	Функціональна верифікація та моделювання роботи системи в Active-HDL	20.05.26 - 22.05.26	
6	Оформлення пояснювальної записки	22.05.26 - 25.05.26	
7	Оформлення графічного та допоміжного матеріалу	25.05.26 – 30.05.26	

Студент (ка)

Кирило КРОТОВ
(підпис) (Ім'я ПРІЗВИЩЕ)

Керівник проєкту (роботи)

Світлана ГРУШКО
(підпис) (Ім'я ПРІЗВИЩЕ)

РЕФЕРАТ

Пояснювальна записка до дипломної роботи бакалавра: 60 с., 4 рис., 4 табл., 5 додатків, 43 джерела.

СЕРВОПРИВОД, FPGA, КОНТРОЛЕР, ШІМ, ПІД-РЕГУЛЯТОР, КВАДРАТУРНИЙ ЕНКОДЕР, VHDL, ПРОГРАМОВАНА ЛОГІЧНА ІНТЕГРАЛЬНА СХЕМА, РОБОТОТЕХНІКА, ЗВОРОТНИЙ ЗВ'ЯЗОК

Об'єкт розробки – контролер сервоприводів для робототехнічних застосувань на базі програмованої логічної інтегральної схеми (FPGA).

Мета роботи – розробка модульного контролера сервоприводів на FPGA, який забезпечує прецизійне керування положенням за допомогою замкненого контуру зворотного зв'язку з ПІД-регулюванням, генерацією ШІМ-сигналів, декодуванням квадратурного енкодера та комунікаційним інтерфейсом UART для робототехнічних застосувань.

Розроблено контролер сервоприводів на базі FPGA, реалізований мовою VHDL. Контролер має модульну архітектуру та складається з п'яти функціональних блоків: генератора ШІМ-сигналів із 16-бітною розрядністю шпаруватості, енкодера квадратурного енкодера з 4х-декодуванням та цифровою фільтрацією від брязкоту, цифрового ПІД-регулятора положення з арифметикою фіксованої точки Q8.8 та механізмом anti-windup, приймача-передавача UART зі швидкістю 115200 бод та парсера команд із протоколом керування.

ABSTRACT

Explanatory note to the bachelor's degree thesis: 60 p., 4 fig., 4 tables, 5 appendices, 43 sources.

SERVO DRIVE, FPGA, CONTROLLER, PWM, PID CONTROLLER, QUADRATURE ENCODER, VHDL, FIELD-PROGRAMMABLE GATE ARRAY, ROBOTICS, FEEDBACK

Object of development – a servo drive controller for robotic applications based on a field-programmable gate array (FPGA).

Objective of the work – development of a modular FPGA-based servo drive controller that provides precise position control through a closed-loop feedback system with PID regulation, PWM signal generation, quadrature encoder decoding, and a UART communication interface for robotic applications.

An FPGA-based servo drive controller has been developed, implemented in VHDL. The controller features a modular architecture comprising five functional blocks: a PWM signal generator with 16-bit duty cycle resolution, a quadrature encoder decoder with 4x decoding and digital debounce filtering, a digital PID position controller with Q8.8 fixed-point arithmetic and anti-windup mechanism, a UART transceiver operating at 115200 baud, and a command parser with a control protocol.

ЗМІСТ

Зміст	6
Вступ.....	8
1 Аналіз предметної області та постановка задачі	10
1.1 Огляд сервоприводів та їх застосування в робототехніці	10
1.2 Аналіз методів керування сервоприводами	12
1.3 Порівняльний аналіз реалізацій контролерів на мікроконтролерах та FPGA ..	15
1.4 Огляд наявних рішень та комерційних контролерів сервоприводів	17
1.5 Обґрунтування вибору апаратної платформи та середовища розробки	21
1.6 Постановка задачі та формування технічних вимог.....	24
1.7 Висновки до розділу 1	26
2 Проєктування контролера сервоприводів.....	27
2.1 Розробка структурної схеми контролера	27
2.2 Проєктування модуля генерації ШІМ-сигналів	28
2.3 Розробка модуля зворотного зв'язку з енкодером	29
2.4 Реалізація ПІД-регулятора положення.....	30
2.5 Проєктування комунікаційного інтерфейсу	31
2.6 Висновки до розділу 2	33
3 Реалізація та експериментальне дослідження	33
3.1 Реалізація модулів контролера мовою VHDL	33
3.2 Функціональна верифікація та моделювання роботи системи.....	36
3.2.1 Тестові сценарії	36
3.2.2 Аналіз часових діаграм ШІМ-генератора.....	37

3.2.3 Аналіз часових діаграм квадратурного декодера	39
3.2.4 Аналіз часових діаграм UART-інтерфейсу	40
3.2.5 Аналіз часових діаграм ПД-регулятора.....	41
3.2.6 Узагальнення результатів верифікації	43
3.3 Синтез проєкту та імплементація на FPGA-платформі	44
3.4 Аналіз використання апаратних ресурсів та швидкодії системи	45
3.5 Висновки до розділу 3	47
Висновки	49
Перелік джерел посилання	50
Додаток А Слайди презентації.....	55

ВСТУП

Сучасна робототехніка переживає період стрімкого розвитку, обумовлений зростаючими потребами промисловості в автоматизації виробничих процесів, підвищенні точності позиціонування та швидкодії виконавчих механізмів. За даними аналітичного звіту Mordor Intelligence, світовий ринок сервоприводів у 2025 році оцінюється у 14,57 мільярда доларів США з прогнозованим зростанням до 19,33 мільярда доларів до 2030 року при середньорічному темпі зростання 5,82% [1]. Така динаміка свідчить про постійне розширення сфери застосування сервоприводів та підвищення вимог до систем керування ними.

Сервопривод являє собою електромеханічний пристрій, здатний забезпечувати прецизійне керування положенням, швидкістю та обертальним моментом за допомогою замкненого контуру зворотного зв'язку. На відміну від звичайних електродвигунів, сервоприводи постійно вимірюють вихідний параметр і коригують його до досягнення заданого значення. Ця особливість робить їх незамінними компонентами у системах, де критично важливим є точне відтворення заданої траєкторії руху: промислових роботах-маніпуляторах, верстатах з числовим програмним керуванням, безпілотних літальних апаратах, медичному обладнанні та системах прецизійного позиціонування.

Традиційно системи керування сервоприводами реалізуються на базі мікроконтролерів або цифрових сигнальних процесорів. Такий підхід забезпечує достатню гнучкість програмування та відносну простоту розробки, проте має суттєві обмеження при роботі з багатоканальними системами або при необхідності забезпечення наднизької затримки керування. Послідовна природа обчислень у мікроконтролерах унеможливорює справжню паралельну обробку кількох контурів керування, що особливо критично для багатоосьових робототехнічних систем.

Альтернативним підходом є використання програмованих логічних інтегральних схем — FPGA (Field-Programmable Gate Array). Ці пристрої дозволяють реалізувати апаратну логіку керування, яка виконується паралельно та детерміновано,

забезпечуючи стабільний час відгуку системи незалежно від складності алгоритму. Сучасні FPGA містять достатню кількість логічних елементів для реалізації складних алгоритмів цифрового керування, включаючи ПД-регулятори, цифрові фільтри та інтерфейси зв'язку, при цьому зберігаючи можливість реконфігурації під конкретні задачі застосування.

Актуальність теми обумовлена кількома факторами. По-перше, зростаючі вимоги до швидкодії промислових роботів потребують скорочення часу реакції системи керування до одиниць мікросекунд, що складно забезпечити програмними засобами на базі мікроконтролерів. По-друге, тенденція до збільшення кількості керованих осей у робототехнічних системах вимагає паралельної обробки сигналів від множини енкодерів та генерації множини ШІМ-сигналів, що природно реалізується в архітектурі FPGA. По-третє, розвиток концепції Індустрії 4.0 передбачає інтеграцію систем керування приводами з промисловими мережами та хмарними сервісами, що потребує гнучких комунікаційних інтерфейсів, легко реалізованих на FPGA.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Огляд сервоприводів та їх застосування в робототехніці

Сервопривод — це електромеханічна система, що складається з електродвигуна, редуктора, давача зворотного зв'язку та електронного блоку керування, призначена для прецизійного керування механічним рухом. Термін «сервомеханізм» походить від латинського слова «servus» (слуга) і відображає основне призначення таких систем — точне виконання команд керуючого пристрою [2]. На відміну від звичайних електродвигунів, які обертаються з постійною швидкістю при подачі напруги живлення, сервоприводи здатні переміщувати вихідний вал у задане положення і утримувати його з високою точністю.

Принцип роботи сервоприводу базується на замкненому контурі керування зі зворотним зв'язком за положенням. Керуючий сигнал задає бажане положення вихідного валу, давач положення вимірює фактичне положення, блок керування обчислює похибку позиціонування та формує керуючий вплив на двигун для її компенсації. Цей процес повторюється безперервно або з певною частотою дискретизації, забезпечуючи стеження за заданою траєкторією руху.

За типом використовуваного електродвигуна розрізняють кілька основних класів сервоприводів. Сервоприводи постійного струму (DC servo) використовують колекторні або безколекторні двигуни постійного струму і характеризуються простотою керування та швидким відгуком на керуючі команди. Вони широко застосовуються в робототехніці, малих автоматизованих системах та пристроях побутової електроніки [3]. Сервоприводи змінного струму (AC servo) побудовані на базі синхронних двигунів з постійними магнітами або асинхронних двигунів і забезпечують високий обертальний момент, плавне обертання та стабільну точність на високих швидкостях. Вони домінують у промисловій автоматизації, верстатах з ЧПК, робототехнічному зварюванні та автоматизованих складальних лініях [4].

Окремий клас становлять хобі-сервоприводи (RC servo), спочатку розроблені для радіокерованих моделей. Ці компактні пристрої поєднують в одному корпусі

двигун постійного струму, редуктор, потенціометричний давач положення та електронну схему керування [5]. Типовий RC-сервопривод приймає на вхід імпульси з широтно-імпульсною модуляцією (ШІМ) частотою 50 Гц, де тривалість імпульсу від 1 до 2 мілісекунд визначає кутове положення вихідного валу в діапазоні приблизно 180 градусів. При тривалості імпульсу 1,5 мс вал займає центральне положення (90°), при 1 мс — крайнє положення 0° , при 2 мс — крайнє положення 180° [6].

Важливими характеристиками сервоприводів є обертальний момент, швидкість, точність позиціонування та час відгуку. Обертальний момент визначає максимальне навантаження, яке може переміщувати привод, і для промислових сервоприводів може досягати сотень ньютон-метрів. Швидкість характеризується часом повороту на заданий кут — для хобі-сервоприводів типовим є показник 0,1–0,2 секунди на 60 градусів при напрузі живлення 6 В [7]. Точність позиціонування промислових сервоприводів може досягати одиниць кутових секунд завдяки використанню високороздільних енкодерів.

У робототехніці сервоприводи відіграють роль «суглобів» робота, забезпечуючи керований рух ланок маніпулятора. Компанія Yaskawa Electric, один зі світових лідерів у виробництві сервоприводів, інтегрує власні сервоприводи у промислові роботи серії MOTOMAN, де точність позиціонування настільки висока, що при повороті двигуна на еквівалент 40 кілометрів окружності похибка становить лише 4 міліметри [8]. Шестиосьові промислові роботи-маніпулятори потребують синхронного керування шістьма сервоприводами з частотою оновлення контуру керування до кількох кілогерц для забезпечення плавності руху та точності траєкторії.

Застосування сервоприводів охоплює широкий спектр робототехнічних систем. У промисловій робототехніці вони використовуються в зварювальних роботах, роботах для фарбування, пакувальних роботах та роботах для складання електронних компонентів. Медична робототехніка застосовує високоточні сервоприводи в хірургічних роботах типу da Vinci, реабілітаційних екзоскелетах та протезах

кінцівок з активним керуванням [9]. Мобільна робототехніка використовує сервоприводи для керування кутом повороту коліс, положенням сенсорних головок та маніпуляторів сервісних роботів. Аерокосмічна галузь застосовує сервоприводи в системах керування літальними апаратами, антенними системами та механізмами розгортання сонячних панелей супутників.

Тенденції розвитку сервоприводів спрямовані на підвищення енергоефективності, інтеграцію з промисловими мережами стандарту Industry 4.0, мініатюризацію та розширення функцій самодіагностики. Сучасні сервоприводи здатні передавати телеметричну інформацію про свій стан у режимі реального часу, що дозволяє реалізувати концепцію предиктивного технічного обслуговування та зменшити простої обладнання.

1.2 Аналіз методів керування сервоприводами

Керування сервоприводами базується на принципах теорії автоматичного керування і передбачає формування керуючого впливу на двигун для забезпечення заданого закону руху вихідного валу. Залежно від керованої величини розрізняють три основні режими роботи сервоприводу: керування положенням, керування швидкістю та керування моментом. Найбільш поширеним у робототехніці є режим керування положенням, де завданням системи є переміщення вихідного валу у задану кутову позицію та її утримання.

Структура типової системи керування положенням сервоприводу включає три вкладені контури: зовнішній контур положення, середній контур швидкості та внутрішній контур струму (моменту). Такий каскадний підхід дозволяє оптимізувати динамічні характеристики системи та спростити налаштування регуляторів [10]. Внутрішній контур струму є найшвидшим і забезпечує задане значення струму в обмотках двигуна, що пропорційне обертальному моменту. Контур шви-

дкості формує задане значення струму для досягнення бажаної швидкості обертання. Зовнішній контур положення задає швидкість руху для досягнення цільової позиції.

Найбільш поширеним алгоритмом керування у сервоприводах є пропорційно-інтегрально-диференціальний (ПІД) регулятор. Керуючий вплив ПІД-регулятора формується як сума трьох складових: пропорційної, інтегральної та диференціальної [11]. Пропорційна складова забезпечує керуючий вплив, пропорційний поточній похибці керування, і визначає швидкість реакції системи на відхилення від заданого значення. Інтегральна складова накопичує похибку у часі та забезпечує усунення статичної похибки в усталеному режимі. Диференціальна складова реагує на швидкість зміни похибки і сприяє демпфуванню коливань та покращенню стійкості системи.

Дискретна форма ПІД-регулятора, придатна для реалізації в цифрових системах керування, може бути записана у позиційній або інкрементній формі. Позиційна форма обчислює повне значення керуючого впливу на кожному кроці дискретизації і вимагає зберігання накопиченої суми інтегральної складової. Інкрементна форма обчислює лише приріст керуючого впливу відносно попереднього значення і характеризується меншими обчислювальними витратами та відсутністю проблеми інтегрального насичення [12].

Для RC-сервоприводів метод керування базується на широтно-імпульсній модуляції. Керуючий сигнал являє собою послідовність імпульсів з фіксованим періодом (типово 20 мс, що відповідає частоті 50 Гц) та змінною тривалістю високого рівня (від 1 до 2 мс). Тривалість імпульсу визначає кутове положення вихідного валу [13]. Внутрішня електроніка сервоприводу порівнює задане положення з показаннями потенціометра зворотного зв'язку та керує двигуном до усунення неузгодженості. Важливо зазначити, що для RC-сервоприводів критичною є саме абсолютна тривалість імпульсу, а не коефіцієнт заповнення — сервопривод однаково реагуватиме на імпульс 1,5 мс незалежно від того, чи становить період 20 мс, чи 10 мс [14].

Для промислових сервоприводів зі зворотним зв'язком від інкрементального енкодера важливим є правильне декодування квадратурних сигналів. Інкрементальний енкодер генерує два сигнали (канали А і В), зсунуті на 90 електричних градусів, що дозволяє визначити напрямок обертання за послідовністю переходів сигналів [15]. Квадратурний декодер може працювати в режимі 1х (лічить лише наростаючі фронти одного каналу), 2х (лічить обидва фронти одного каналу) або 4х (лічить усі фронти обох каналів), забезпечуючи відповідно одно-, дво- або чотириразове збільшення роздільної здатності відносно фізичної роздільної здатності енкодера.

Сучасні методи керування сервоприводами включають також адаптивне керування, яке автоматично підлаштовує параметри регулятора під зміни характеристик об'єкта керування, та інтелектуальне керування на базі нейронних мереж і нечіткої логіки [16]. Дослідження показують, що комбінування ПІД-регулятора з нейронною мережею дозволяє адаптивно підлаштовувати коефіцієнти регулятора в режимі реального часу, що особливо ефективно для нелінійних об'єктів керування [17]. Проте складність реалізації та високі вимоги до обчислювальних ресурсів обмежують застосування таких методів переважно високовартісними промисловими системами.

Окремим напрямком є методи керування багатоосьовими системами, де потрібна координація руху кількох сервоприводів. Для забезпечення синхронного руху використовується електронна редукція, інтерполяція траєкторії та розрахунок оберненої кінематики [18]. Реалізація таких алгоритмів потребує значних обчислювальних ресурсів та жорстких вимог до детермінованості часу виконання, що робить FPGA привабливою платформою для їх реалізації.

1.3 Порівняльний аналіз реалізацій контролерів на мікроконтролерах та FPGA

Вибір апаратної платформи для реалізації контролера сервоприводів є критично важливим рішенням, що визначає характеристики системи, складність розробки та вартість кінцевого продукту. Традиційно контролери сервоприводів реалізуються на базі мікроконтролерів (MCU), проте все більшого поширення набувають реалізації на програмованих логічних інтегральних схемах FPGA.

Мікроконтролер являє собою однокристальний комп'ютер, що поєднує на одному чіпі процесорне ядро, пам'ять програм та даних, а також периферійні модулі: таймери, аналого-цифрові перетворювачі, комунікаційні інтерфейси та генератори ШІМ-сигналів [19]. Програма мікроконтролера виконується послідовно, інструкція за інструкцією, що визначає фундаментальне обмеження — неможливість справжньої паралельної обробки кількох задач. Переривання дозволяють імітувати багатозадачність, але обробка кожного переривання потребує часу на збереження контексту та перемикання між задачами.

FPGA — це інтегральна схема, яка може бути сконфігурована користувачем після виготовлення для реалізації довільної цифрової логіки [20]. Архітектура FPGA базується на масиві конфігурованих логічних блоків (CLB), з'єднаних програмованою комутаційною матрицею. Кожен логічний блок містить таблиці пошуку (LUT), які можуть реалізувати будь-яку булеву функцію від кількох змінних, та тригери для зберігання стану. Сучасні FPGA також містять спеціалізовані блоки: помножувачі, блоки цифрової обробки сигналів (DSP), блочну пам'ять (BRAM) та високошвидкісні приймачі-передавачі [21].

Порівняння архітектур MCU та FPGA виявляє принципові відмінності у підходах до обробки інформації. Мікроконтролер виконує програму послідовно, що робить час виконання залежним від кількості операцій. При збільшенні кількості керованих осей час обробки зростає пропорційно, обмежуючи максимальну час-

тоту контуру керування [22]. FPGA реалізує апаратну логіку, яка виконується паралельно: усі комбінаційні схеми обчислюють результат одночасно за час поширення сигналу, а синхронні елементи оновлюють стан одночасно за фронтом тактового сигналу. Додавання ще одного каналу керування потребує лише додаткових логічних ресурсів, не впливаючи на час обробки існуючих каналів.

Детермінованість часу відгуку є важливою характеристикою для систем керування. У мікроконтролерах час відгуку залежить від поточного стану виконання програми та пріоритету переривань. Найгірший випадок може суттєво відрізнитися від типового, що ускладнює гарантування часових обмежень жорсткого реального часу. В FPGA час відгуку визначається затримкою поширення сигналу через комбінаційну логіку і є стабільним та передбачуваним [23]. Дослідження показують, що FPGA-реалізація контролера шестиосьового робота забезпечує час відгуку 160 нс при загальному часі реакції системи керування 82 мкс, що значно перевищує можливості мікроконтролерних рішень [24].

Ресурсні характеристики платформ також суттєво відрізняються. Мікроконтролери забезпечують фіксовану кількість периферійних модулів: типово від 2 до 16 каналів ШІМ та 2-4 таймери з функцією захоплення для енкодерів. Збільшення кількості каналів потребує використання кількох мікроконтролерів або спеціалізованих периферійних мікросхем, що ускладнює синхронізацію [25]. FPGA дозволяє реалізувати практично необмежену кількість каналів ШІМ та декодерів енкодерів у межах доступних логічних ресурсів, при цьому всі канали працюють синхронно від єдиного тактового сигналу.

Енергоспоживання мікроконтролерів зазвичай нижче за FPGA аналогічного класу, що робить їх кращим вибором для автономних систем з батарейним живленням [26]. Проте при порівнянні енергоефективності (співвідношення продуктивності до споживаної потужності) для паралельних обчислень FPGA може демонструвати переваги за рахунок апаратної оптимізації алгоритмів.

Вартість розробки та тривалість циклу проєктування відрізняються для обох платформ. Мікроконтролери програмуються мовами високого рівня (C/C++), ма-

ють розвинену екосистему бібліотек та значну спільноту розробників, що прискорює розробку типових застосувань [27]. Розробка для FPGA потребує знання мов опису апаратури (VHDL, Verilog) та розуміння принципів проектування цифрових схем. Синтез, розміщення та трасування FPGA-проєкту може займати від хвилин до годин залежно від складності. Проте одного разу розроблений та верифікований IP-модуль може бути багаторазово використаний у різних проєктах.

Собівартість компонентів залежить від обраних моделей. Найпростіші мікроконтролери коштують менше долара, тоді як FPGA початкового рівня — від кількох доларів [28]. Проте для складних багатоосьових систем вартість додаткових периферійних мікросхем для мікроконтролерного рішення може перевищити вартість FPGA з інтегрованими функціями.

Узагальнюючи порівняльний аналіз, можна сформулювати рекомендації щодо вибору платформи. Мікроконтролери доцільно використовувати для одноканальних або малоканалних систем з помірними вимогами до швидкодії, де важливі низька вартість, низьке енергоспоживання та швидкість розробки. FPGA є оптимальним вибором для багатоканальних систем з жорсткими вимогами до детермінованості, де потрібна паралельна обробка та мінімальна латентність. Комбінований підхід, де мікроконтролер виконує високорівневі функції керування, а FPGA — швидкодіючі низькорівневі контури, поєднує переваги обох платформ.

1.4 Огляд наявних рішень та комерційних контролерів сервоприводів

Ринок контролерів сервоприводів представлений широким спектром рішень — від простих драйверів для хобі-застосувань до складних промислових систем керування рухом. Аналіз наявних рішень дозволяє визначити типові архітектурні підходи, характеристики та функціональні можливості, які слід врахувати при розробці власного контролера.

Серед провідних виробників промислових сервоприводів та контролерів слід виділити компанії Yaskawa Electric (Японія), Mitsubishi Electric (Японія), Siemens (Німеччина), Rockwell Automation (США) та Schneider Electric (Франція), які разом контролюють значну частку світового ринку [29]. Продуктові лінійки цих виробників включають повний набір компонентів: сервоприводи різної потужності, сервопідсилювачі (драйвери), контролери руху та програмне забезпечення для налаштування і діагностики.

Серія Sigma-7 компанії Yaskawa представляє сучасне покоління сервоприводів з інтегрованими функціями автоматичного налаштування, придушення вібрацій та компенсації тертя [30]. Сервопідсилювачі SERVOPACK забезпечують цифрове керування з підтримкою промислових мереж EtherCAT, MECHATROLINK та інших. Контролери руху серії MP підтримують керування до 32 осей з можливістю синхронізації та інтерполяції траєкторії. Характерною особливістю рішень Yaskawa є використання пропрієтарних комунікаційних ASIC для забезпечення високошвидкісного обміну даними між компонентами системи.

Siemens пропонує серію сервоприводів SIMOTICS S у поєднанні з системою приводів SINAMICS S210, що використовує однокабельну технологію з передачею живлення та сигналів зворотного зв'язку одним кабелем [31]. Система підтримує інтеграцію з платформою автоматизації TIA Portal та програмованими логічними контролерами серії SIMATIC. Промислові рішення Siemens орієнтовані на великомасштабні виробничі системи з високими вимогами до надійності та діагностики.

Компанія Mitsubishi Electric розвиває серію сервоприводів MELSERVO, де моделі J5 та J4 забезпечують точність позиціонування з роздільною здатністю енкодера до 26 біт (67 мільйонів імпульсів на оберт) [32]. Контролери руху інтегруються з ПЛК серії iQ та підтримують мережу CC-Link IE для об'єднання компонентів автоматизації.

На ринку систем для хобі-робототехніки поширені контролери на базі мікроконтролерів, такі як плати Pololu Maestro, що керують від 6 до 24 RC-сервоприводами через USB або послідовний інтерфейс [33]. Ці контролери забезпечують гене-

рацію ШІМ-сигналів з роздільною здатністю 0,25 мкс та підтримують послідовності рухів та скрипти. Обмеженнями таких рішень є відсутність зворотного зв'язку за положенням (крім спеціальних сервоприводів з аналоговим виходом) та послідовна обробка каналів.

Академічні та дослідницькі розробки контролерів сервоприводів на FPGA демонструють потенціал цієї платформи. Дослідники з Університету Керетаро (Мексика) представили контролер руху для DC-сервоприводів на базі FPGA Xilinx Zynq, що поєднує програмований ПД-регулятор, драйвер ЦАП та інтерфейс квадратурного енкодера [34]. Експериментальна верифікація підтвердила коректність роботи при порівнянні з програмною моделлю в Simulink.

Тайванські дослідники розробили FPGA-контролер крутного моменту для шестиосьового промислового робота HIWIN RA605 з використанням Verilog HDL [35]. Реалізація включала лічильники енкодерів, цифровий фільтр, генератор аналогового сигналу, ПД-регулятор та комунікаційний модуль. Досягнутий час обробки FPGA склав 160 нс, а загальний час відгуку системи керування — 82 мкс при похибці позиціонування кінцевого ефектора близько 0,19 мм.

Проект з відкритим кодом OpenCores містить VHDL-реалізацію квадратурного декодера з інтерфейсом шини Wishbone, що підтримує 4-кратне декодування, програмовані переривання та синхронізацію захоплення лічильника для багатоосьових систем [36]. Модуль верифіковано на FPGA Xilinx з частотами рахунку до 1,3 МГц при системній тактовій частоті 50 МГц.

Аналіз комерційних та дослідницьких рішень виявляє низку спільних архітектурних підходів: модульна структура з виділенням функціональних блоків (генератор ШІМ, інтерфейс енкодера, регулятор, комунікаційний модуль); використання стандартних комунікаційних протоколів для зв'язку з керуючим пристроєм; параметричне налаштування характеристик регулятора без перепрограмування; вбудовані функції діагностики та захисту.

Таблиця 1.1 — Порівняльний аналіз комерційних контролерів сервоприводів

Характеристика	Yaskawa Sigma-7	Siemens SINAMICS S210	Mitsubishi MELSERV O-J5	Pololu Maestro	FPGA-розробки (академічні)
1	2	3	4	5	6
Категорія	Промисловий	Промисловий	Промисловий	Хобі	Дослідницькі
Діапазон потужності	50 Вт – 15 кВт	50 Вт – 7 кВт	50 Вт – 22 кВт	RC-серво (до 10 А)	Залежить від драйвера
Роздільна здатність енкодера	24 біт (16 млн імп./об.)	22–26 біт	26 біт (67 млн імп./об.)	Потенціометр (аналоговий)	16–32 біт (квадратурний)
Смуга пропускання регулятора	3,1 кГц	До 2 кГц	3,5 кГц	—	До 10 кГц (залежить від реалізації)
Кількість осей	До 62 (MECHATROLINK-III)	1 (single-axis)	До 256 (CC-Link IE TSN)	6–24 канали	Масштабується апаратно
Комунікаційний інтерфейс	MECHATROLINK-III, EtherCAT	PROFINET IRT, PROFIsafe	CC-Link IE TSN, SSCNET III/H	USB, TTL Serial	UART, SPI, custom
Час циклу зв'язку	125 мкс	250 мкс	32,5 мкс	—	< 1 мкс (внутрішній)
Функції безпеки	STO, SS1, SLS	STO, SS1, SLS, PROFIsafe	STO, SS1, SLS, SS2	—	—
Автоналаштування	Так (tuning-less mode)	Так (one-button tuning)	Так (0,3 с Quick tune)	—	Ручне
Придушення вібрацій	Так	Так	Так (AI Maisart)	—	—
Однокабельне з'єднання	MECHATROLINK-III	OCC (One Cable Connection)	Так	—	—
Програмне забезпечення	SigmaWin +	TIA Portal	MR Configurator2	Maestro Control Center	Vivado/Quartus
Орієнтовна вартість системи	\$1500–5000+	\$800–3000+	\$1000–4000+	\$20–50	\$50–300 (плата FPGA)

Кінець таблиці 1.1

1	2	3	4	5	6
Архітектура	Закрита, пропрієтарна	Закрита, пропрієтарна	Закрита, пропрієтарна	Закрита	Відкрита, модифікована
Цільове застосування	CNC, робототехніка, автоматизація	Пакування, handling	CNC, робототехніка	Анімація, хобі-роботи	Дослідження, навчання

Порівнюючи комерційні рішення з академічними розробками, слід зазначити, що промислові контролери пропонують значно ширший функціонал, розвинену діагностику та сертифіковану безпеку, проте мають закриту архітектуру і високу вартість. Академічні розробки демонструють принципову можливість реалізації основних функцій контролера на FPGA з відкритою архітектурою, що дозволяє адаптувати їх під специфічні задачі дослідження та навчання.

1.5 Обґрунтування вибору апаратної платформи та середовища розробки

Вибір апаратної платформи для реалізації контролера сервоприводів визначається сукупністю технічних вимог, економічних обмежень та доступності засобів розробки. На основі проведеного аналізу предметної області та порівняння альтернативних підходів обґрунтовано вибір FPGA як цільової платформи та конкретизовано сімейство мікросхем і середовище розробки.

Для навчальних та дослідницьких застосувань оптимальним є використання FPGA початкового або середнього рівня, що забезпечують достатні логічні ресурси при прийнятній вартості. На ринку FPGA домінують два виробники: AMD (раніше Xilinx) та Intel (раніше Altera), кожен з яких пропонує широку лінійку пристроїв та відповідне програмне забезпечення для розробки [37].

AMD Xilinx пропонує кілька сімейств FPGA для різних сегментів ринку. Сімейство Artix-7 позиціонується як економічне рішення для застосувань, чутливих до вартості та енергоспоживання, і містить від 12 до 215 тисяч логічних комірок. Сімейство Spartan-7 є ще більш економічним при збереженні сумісності з інструментами розробки серії 7. Сімейство Zynq поєднує FPGA з вбудованим процесором ARM Cortex-A9, що дозволяє реалізувати гібридні системи з апаратним прискоренням критичних функцій [38].

Intel FPGA (колишні Altera) пропонує сімейство Cyclone для економічних застосувань, Arria для балансу між продуктивністю та енергоефективністю, та Stratix для високопродуктивних систем [39]. Сімейство Cyclone 10 позиціонується як entry-level рішення для промислових, автомобільних та IoT-застосувань з низьким енергоспоживанням.

Для реалізації контролера сервоприводів обрано платформу на базі FPGA Xilinx сімейства Artix-7 або Spartan-7. Цей вибір обумовлений наступними факторами:

Достатні логічні ресурси: навіть молодші моделі Spartan-7 (XC7S15) містять понад 12 000 логічних комірок, 20 блоків DSP48E1 та 50 блоків блочної пам'яті, що достатньо для реалізації багатоканального контролера з ПД-регуляторами.

Наявність апаратних блоків множення та накопичення: блоки DSP48E1 дозволяють ефективно реалізувати арифметичні операції ПД-регулятора без витрат логічних ресурсів загального призначення.

Широка екосистема налагоджувальних плат: доступні численні плати від Digilent (Basys 3, Arty, Nexys), Avnet та інших виробників з вартістю від 50 до 300 доларів США, що включають FPGA, пам'ять, інтерфейси та периферію.

Безкоштовне програмне забезпечення: Xilinx Vivado Design Suite доступний у редакції WebPACK без обмежень за часом використання для підтримуваних сімейств FPGA.

Розвинена навчальна база: наявність численних підручників, онлайн-курсів та прикладів проєктів для платформи Xilinx спрощує освоєння технології.

Середовищем розробки обрано Xilinx Vivado Design Suite, що забезпечує повний цикл проєктування FPGA: введення проєкту мовами VHDL або Verilog, синтез логічної схеми, розміщення та трасування на цільовому пристрої, генерацію бітового потоку для конфігурації FPGA, моделювання на різних рівнях абстракції та апаратну налагодження [40]. Інтегрований симулятор Vivado Simulator дозволяє виконувати функціональне та часове моделювання без додаткових інструментів. Для складніших проєктів можливе використання ModelSim або QuestaSim.

Мовою опису апаратури обрано VHDL (VHSIC Hardware Description Language), що є галузевим стандартом IEEE 1076 і широко використовується в академічному середовищі та європейській промисловості [41]. VHDL характеризується строгою типізацією, що сприяє виявленню помилок на етапі компіляції, та підтримує різні рівні абстракції опису — від поведінкового до структурного. Альтернативою є Verilog, більш поширений у північноамериканській індустрії, проте принципових переваг для задач цієї роботи він не забезпечує.

Для верифікації проєкту передбачається використання методології тестових стендів (testbench) з автоматизованою перевіркою результатів. Функціональне моделювання дозволить перевірити коректність логіки роботи модулів до їх синтезу та завантаження на FPGA. Часове моделювання з урахуванням затримок поширення сигналів підтвердить дотримання часових обмежень.

Апаратне тестування планується виконати на налагоджувальній платі з FPGA Xilinx. Як периферію тестового стенду доцільно використати RC-сервоприводи стандарту 180° з потенціометричним зворотним зв'язком та інкрементальні енкодери для моделювання промислових застосувань. Логічний аналізатор або вбудований модуль Integrated Logic Analyzer (ILA) дозволить спостерігати внутрішні сигнали FPGA в реальному часі.

1.6 Постановка задачі та формування технічних вимог

На основі проведеного аналізу предметної області та огляду наявних рішень сформульовано постановку задачі та технічні вимоги до контролера сервоприводів, що розробляється.

Метою розробки є створення контролера сервоприводів на базі FPGA, призначеного для керування положенням RC-сервоприводів та DC-двигунів з енкодером у робототехнічних застосуваннях. Контролер має забезпечувати прецизійне керування положенням, підтримувати багатоканальну роботу та надавати інтерфейс для зв'язку з керуючим пристроєм вищого рівня.

Функціональні вимоги до контролера охоплюють п'ять основних аспектів. Першим є генерація ШІМ-сигналів керування, де контролер має формувати сигнали для RC-сервоприводів у вигляді імпульсів тривалістю від 1 до 2 мс з періодом повторення 20 мс, а також сигнали з регульованою шпаруватістю для H-мостових драйверів DC-двигунів. Роздільна здатність ШІМ має становити не менше 10 біт, що відповідає 1024 дискретним рівням керування.

Другою вимогою є реалізація інтерфейсу енкодера з підтримкою підключення інкрементального квадратурного давача положення. Декодування сигналів енкодера має виконуватися в режимі 4x для максимізації роздільної здатності, при цьому модуль повинен забезпечувати визначення напрямку обертання та підрахунок імпульсів з розрядністю лічильника не менше 16 біт.

Третя функціональна вимога стосується реалізації ПД-регулятора положення. Цифровий регулятор має підтримувати параметричне налаштування коефіцієнтів пропорційної, інтегральної та диференціальної складових. Обов'язковими є функції обмеження вихідного сигналу та антивіндапний захист інтегральної складової для запобігання насиченню при тривалих відхиленнях від заданого положення.

Четвертою вимогою є наявність комунікаційного інтерфейсу для обміну даними з керуючим пристроєм вищого рівня. Інтерфейс має забезпечувати задання

цільового положення, читання поточного положення та налаштування параметрів регулятора. Базовим протоколом обрано UART зі швидкістю передачі до 115200 біт/с.

П'ята вимога полягає у забезпеченні багатоканальності, де архітектура контролера має передбачати можливість масштабування кількості керованих осей без суттєвих змін у структурі проєкту.

Технічні характеристики контролера визначаються наступними параметрами. Частота тактування системи має становити від 50 до 100 МГц залежно від можливостей обраної налагоджувальної плати. Частота оновлення контуру ПД-регулятора повинна бути не меншою за 1 кГц для забезпечення належної динаміки керування. Роздільна здатність ШІМ встановлюється на рівні 10 біт, що забезпечує 1024 дискретні рівні вихідного сигналу. Для RC-сервоприводів частота ШІМ фіксується на рівні 50 Гц з періодом 20 мс відповідно до стандарту керування такими приводами, тоді як для DC-двигунів передбачається регульована частота ШІМ у діапазоні від 20 до 50 кГц. Розрядність лічильника енкодера становить 16 біт, а коефіцієнти ПД-регулятора представляються у форматі з фіксованою комою з розрядністю 16 біт. Комунікаційний інтерфейс UART підтримує швидкості від 9600 до 115200 біт/с.

При розробці контролера прийнято низку обмежень та припущень. Оскільки контролер розробляється для навчальних та дослідницьких цілей, промислові вимоги до функціональної безпеки згідно зі стандартом IEC 61508 не розглядаються. Силова частина у вигляді драйвера двигуна не входить до складу розробки і вважається зовнішнім компонентом системи. Живлення FPGA-плати здійснюється від зовнішнього джерела стабілізованої напруги. Для RC-сервоприводів передбачається використання стандартного трипровідного інтерфейсу, що включає лінії живлення, землі та керуючого сигналу.

Критерії успішності розробки включають функціональну верифікацію всіх модулів у середовищі моделювання з демонстрацією коректності їх роботи, успішний синтез та імплементацію проєкту на цільовій FPGA з дотриманням часових

обмежень, експериментальне підтвердження керування RC-сервоприводом із переміщенням у задане положення, а також повне документування архітектури, інтерфейсів та методики використання розробленого контролера.

1.7 Висновки до розділу 1

У першому розділі кваліфікаційної роботи проведено комплексний аналіз предметної області та сформульовано постановку задачі розробки контролера сервоприводів на FPGA.

Розглянуто принципи роботи та класифікацію сервоприводів, що застосовуються в робототехніці. Встановлено, що сервоприводи відіграють роль ключових виконавчих елементів робототехнічних систем, забезпечуючи прецизійне керування рухом ланок механізму. Проаналізовано особливості RC-сервоприводів з ШІМ-керуванням та промислових сервоприводів зі зворотним зв'язком від енкодера.

Виконано аналіз методів керування сервоприводами, зокрема принципів побудови замкнених контурів керування положенням, швидкістю та моментом. Детально розглянуто алгоритм ПД-регулювання в позиційній та інкрементній формах, а також методи декодування квадратурних сигналів енкодера. Обґрунтовано доцільність застосування класичного ПД-регулятора для задач керування положенням сервоприводу.

Проведено порівняльний аналіз реалізацій контролерів на мікроконтролерах та FPGA за критеріями продуктивності, детермінованості, масштабованості, енергоефективності та вартості розробки. Встановлено, що FPGA забезпечує принципові переваги для багатоканальних систем з жорсткими вимогами до латентності та детермінованості за рахунок апаратного паралелізму обробки.

Виконано огляд комерційних контролерів сервоприводів провідних виробників (Yaskawa, Siemens, Mitsubishi Electric) та академічних розробок на базі FPGA.

Визначено типові архітектурні підходи та функціональні можливості сучасних систем керування рухом.

Сформульовано постановку задачі та технічні вимоги до контролера, що розробляється. Визначено функціональні вимоги (генерація ШІМ, інтерфейс енкадера, ПІД-регулятор, комунікаційний інтерфейс), технічні характеристики (роздільна здатність, частоти, розрядності) та критерії успішності розробки.

Результати аналізу, проведеного в цьому розділі, формують теоретичну та методологічну основу для проєктування контролера сервоприводів, що виконується в наступних розділах роботи.

2 ПРОЄКТУВАННЯ КОНТРОЛЕРА СЕРВОПРИВОДІВ

2.1 Розробка структурної схеми контролера

Структурна схема контролера побудована за модульним принципом, що забезпечує незалежність розробки та тестування окремих компонентів. Система складається з п'яти основних модулів: комунікаційний інтерфейс (UART RX/TX + парсер команд), генератор ШІМ-сигналів, декодер квадратурного енкадера, ПІД-регулятор та модуль верхнього рівня (top-level), що здійснює з'єднання всіх компонентів.

Потік даних у системі організовано таким чином. Зовнішня керуюча система (ПК або мікроконтролер) надсилає команди через UART. Парсер команд розбирає отримані пакети та оновлює відповідні регістри: задане положення (setpoint), коефіцієнти ПІД або видає запит на зчитування поточної позиції. ПІД-регулятор обчислює різницю між заданим та фактичним положенням і формує керуючий вплив. Цей вплив перетворюється у шпаруватість ШІМ-сигналу та сигнал напрямку для Н-мосту.

Синхронізація роботи модулів здійснюється через єдиний тактовий сигнал (50 МГц) та синхроімпульс від генератора ШІМ, який запускає обчислення ПІД на

початку кожного нового періоду ШІМ. Це гарантує, що оновлення керуючого впливу відбувається синхронно з формуванням ШІМ-сигналу, виключаючи можливість глітчів.

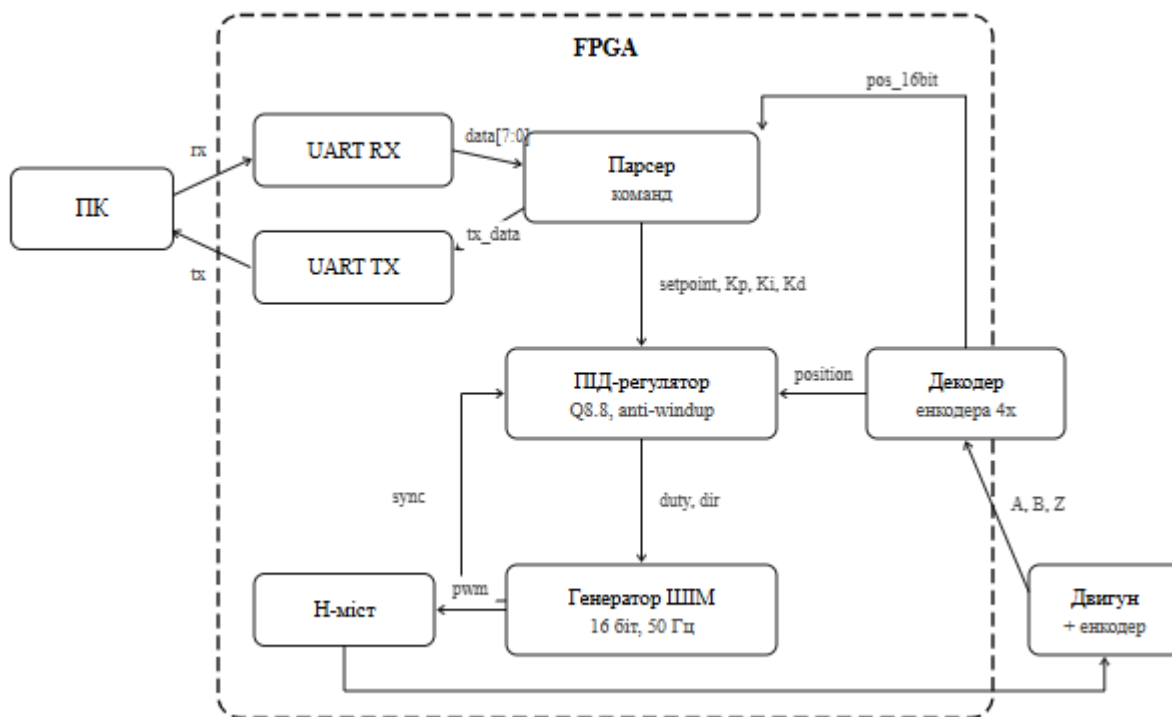


Рисунок 2.1 — Структурна схема контролера сервоприводів

Усі зовнішні сигнали (UART RX, канали енкодера A/B/Z) проходять через подвійні синхронізуючі регістри для запобігання метастабільності. Вхідний сигнал скидання (rst_n) є активним низьким та може бути ініційований як зовнішньою кнопкою, так і командою через UART.

2.2 Проєктування модуля генерації ШІМ-сигналів

Модуль генерації ШІМ-сигналів (pwm_generator) формує прямокутний сигнал із програмованою шпаруватістю для керування силовим каскадом сервопри-

воду. Проектування модуля ґрунтується на принципі порівняння лічильника з порогом шпаруватості.

Основу модуля складає 20-бітний лічильник, що рахує такти основної частоти від 0 до $N-1$, де $N = \text{CLK_FREQ} / \text{PWM_FREQ}$. При тактовій частоті 50 МГц та частоті ШІМ 50 Гц значення $N = 1\,000\,000$. На початку кожного нового періоду ($\text{counter} = 0$) відбувається захоплення поточного значення duty у внутрішній регістр, що запобігає зміні шпаруватості посеред періоду.

Вихідний ШІМ-сигнал формується компаратором: $\text{pwm_out} = '1'$, коли значення лічильника менше за перераховане значення duty_ticks , та $'0'$ в іншому випадку. Перерахунок 16-бітного значення duty у кількість тактів виконується за формулою: $\text{duty_ticks} = (\text{duty} \times N) / 65535$. Для уникнення апаратного множення великих чисел застосовано розкладання на етапи із зсувами.

Модуль також формує синхроімпульс pwm_sync тривалістю один такт на початку кожного періоду ШІМ. Цей сигнал використовується для запуску обчислення ПІД-регулятора, що забезпечує синхронність оновлення керуючого впливу.

Параметри модуля задаються через механізм *generic*, що дозволяє легко адаптувати модуль до різних тактових частот, частот ШІМ та розрядностей шпаруватості без модифікації коду.

2.3 Розробка модуля зворотного зв'язку з енкодером

Модуль декодера квадратурного енкодера (*quadrature_decoder*) забезпечує прийом, фільтрацію та декодування сигналів від інкрементального енкодера, що є основним давачем зворотного зв'язку для системи позиційного керування.

Квадратурний енкодер генерує два прямокутних сигнали (канали А та В), зсунуті один відносно одного на 90° . Аналіз взаємного положення фронтів цих сигна-

лів дозволяє визначити як напрямок, так і величину переміщення. У режимі 4х-декодування враховуються всі чотири переходи за один період квадратурного сигналу, що забезпечує максимальну роздільну здатність [8].

Вхідні сигнали А та В спочатку проходять через ланцюг подвійної синхронізації (два послідовних D-тригери) для запобігання метастабільності, пов'язаній із асинхронністю сигналів енкодера відносно тактового сигналу FPGA. Після синхронізації сигнали фільтруються за допомогою цифрового фільтра від брязкоту, реалізованого як лічильник стабільності. Зміна стану відфільтрованого сигналу відбувається лише після того, як вхідний сигнал зберігає нове значення протягом $2^{\text{DEBOUNCE_BITS}}$ тактів.

Власне квадратурне декодування реалізовано через аналіз таблиці переходів. При кожному переході (зміні фронту А або В) напрямок визначається як XOR поточного значення одного каналу з попереднім значенням іншого. 32-бітний лічильник позиції (signed) інкрементується або декрементується залежно від визначеного напрямку.

Додатково модуль обчислює швидкість переміщення як різницю позицій за фіксований часовий інтервал (1 мс при тактовій частоті 50 МГц). Підтримується також обробка індексного імпульсу (канал Z), який може використовуватися для калібрування абсолютного положення.

2.4 Реалізація ПД-регулятора положення

ПД-регулятор є центральним обчислювальним модулем контролера. Його задачею є мінімізація похибки між заданим положенням (setpoint) та фактичним положенням, отриманим від декодера енкодера, шляхом формування відповідного керуючого впливу.

Для апаратної реалізації обрано позиційну форму ПІД-алгоритму з окремим обчисленням пропорційної, інтегральної та диференціальної складових. Арифметика виконується з фіксованою точкою у форматі Q8.8 (8 біт цілої частини, 8 біт дробової), що забезпечує діапазон коефіцієнтів від -128 до +127.996 із кроком 0.00390625.

Обчислення організовано у вигляді конвеєра зі скінченним автоматом (FSM), що має сім станів: IDLE, CALC_ERROR, CALC_P, CALC_I, CALC_D, CALC_SUM та OUTPUT_STAGE. Такий підхід дозволяє виконати повне обчислення ПІД за 7 тактів, що при частоті 50 МГц становить 140 нс – на кілька порядків менше, ніж період ШІМ.

Механізм anti-windup реалізовано як обмеження інтегральної складової у визначеному діапазоні. Якщо накопичене значення інтегратора перевищує встановлені межі, подальше збільшення блокується. Це запобігає ситуації, коли через тривалу наявність похибки (наприклад, при механічному обмеженні руху) інтегральна складова зростає до надмірних значень, що спричиняє перерегулювання після зняття обмеження.

Вихід ПІД-регулятора також обмежується в діапазоні допустимих значень duty для генератора ШІМ. Прапорець saturated сигналізує про досягнення обмеження, що може використовуватися для діагностики.

2.5 Проєктування комунікаційного інтерфейсу

Комунікаційний інтерфейс забезпечує двосторонній зв'язок між контролером та зовнішньою керуючою системою. Для реалізації обрано протокол UART як найбільш простий та універсальний для прототипування. Швидкість передачі — 115200 бод, формат кадру — 8N1 (8 біт даних, без паритету, 1 стоп-біт).

Модуль UART складається з приймача (uart_rx) та передавача (uart_tx), кожен з яких реалізований як окремий скінченний автомат. Приймач синхронізує вхідну лінію через подвійний регістр, детектує старт-біт за зниженням лінії до рівня '0', зчитує 8 біт даних у середині кожного бітового інтервалу та завершує прийом по стоп-біту.

Парсер команд (cmd_parser) обробляє потік байтів від приймача та перетворює їх у внутрішні команди. Протокол передбачає пакети з трьох байтів: код команди (CMD), старший байт даних (DATA_H) та молодший байт даних (DATA_L). Підтримуються такі команди: встановлення setpoint (0x01), зміна коефіцієнтів K_p (0x02), K_i (0x03), K_d (0x04), запит поточної позиції (0x05) та системне скидання (0x06).

При отриманні команди запиту позиції контролер надсилає у відповідь два байти, що містять поточне значення позиції (старший та молодший байти 16-бітного числа).

Таблиця 2.1 — Формат комунікаційного протоколу

Код CMD	Назва	DATA (16 біт)	Напрямок
0x01	SET_POSITION	Задане положення	ПК → FPGA
0x02	SET_KP	Коефіцієнт K_p (Q8.8)	ПК → FPGA
0x03	SET_KI	Коефіцієнт K_i (Q8.8)	ПК → FPGA
0x04	SET_KD	Коефіцієнт K_d (Q8.8)	ПК → FPGA
0x05	GET_POSITION	Не використ.	ПК → FPGA → ПК
0x06	RESET	Не використ.	ПК → FPGA

2.6 Висновки до розділу 2

У другому розділі розроблено модульну архітектуру контролера, що складається з п'яти функціональних блоків: генератора ШІМ, декодера квадратурного енкодера, ПІД-регулятора, модулів UART та парсера команд. Модульність забезпечує незалежне тестування та повторне використання компонентів.

Генератор ШІМ-сигналів реалізовано з 16-бітною розрядністю шпаруватості та синхроімпульсом для координації роботи ПІД-регулятора, що забезпечує 65536 рівнів керування.

Декодер квадратурного енкодера забезпечує 4х-декодування з цифровою фільтрацією від брязкоту та обчисленням швидкості, що підвищує роздільну здатність вимірювання положення вчетверо порівняно з 1х-декодуванням.

ПІД-регулятор реалізовано у вигляді конвеєрного скінченного автомата з арифметикою Q8.8, механізмом anti-windup та обмеженням виходу. Повний цикл обчислення становить 7 тактів (140 нс при 50 МГц).

Комунікаційний інтерфейс UART із простим протоколом з трьох байтів забезпечує динамічне керування параметрами системи без перепрограмування FPGA.

3 РЕАЛІЗАЦІЯ ТА ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ

3.1 Реалізація модулів контролера мовою VHDL

Реалізацію контролера виконано мовою VHDL-2008 із дотриманням рекомендацій щодо синтезопридатного стилю опису апаратури. Стандарт VHDL-2008 (IEEE 1076-2008) обрано завдяки підтримці таких конструкцій, як параметри процедур із режимом inout для сигналів, спрощений синтаксис ініціалізації масивів та покращена робота з типами. Кожен модуль описано як окрему сутність (entity) із

відповідною архітектурою (architecture), що забезпечує ієрархічну організацію проєкту та можливість незалежної компіляції.

Проєкт складається з семи VHDL-файлів: `pwm_generator.vhd` (генератор ШІМ-сигналів), `quadrature_decoder.vhd` (декодер квадратурного енкодера), `pid_controller.vhd` (ПІД-регулятор), `uart_rx.vhd` (приймач UART), `uart_tx.vhd` (передавач UART), `cmd_parser.vhd` (парсер команд) та `servo_controller_top.vhd` (модуль верхнього рівня). Загальний обсяг HDL-коду становить близько 600 рядків, що свідчить про ефективне використання можливостей мови VHDL.

Для забезпечення параметризації модулів широко використано механізм `generic`. Тактова частота (`CLK_FREQ`), частота ШІМ (`PWM_FREQ`), розрядність шпаруватості (`PWM_BITS`), розрядність лічильника позиції (`POS_WIDTH`) та швидкість UART (`BAUD_RATE`) задаються як `generic`-параметри верхнього рівня і каскадно передаються у підмодулі через `generic map`. Це забезпечує можливість адаптації проєкту до різних FPGA-платформ, тактових генераторів та периферійних пристроїв без модифікації вихідного коду.

Окрему увагу приділено стилю кодування. Всі процеси використовують синхронний стиль із єдиним тактовим сигналом та синхронним скиданням, що є рекомендованою практикою для FPGA-проєктування. Асинхронні вхідні сигнали (`uart_rx`, `enc_a`, `enc_b`, `enc_z`) проходять через подвійні синхронізуючі регістри для запобігання метастабільності. Арифметичні операції виконуються над типами `signed` та `unsigned` з бібліотеки `IEEE.NUMERIC_STD`, що забезпечує чітку семантику та уникнення неоднозначностей, пов'язаних із застарілою бібліотекою `STD_LOGIC_ARITH`.

Лістинг 3.1 — Фрагмент сутності верхнього рівня

```
entity servo_controller_top is
    generic (
        CLK_FREQ      : integer := 50_000_000;
        BAUD_RATE     : integer := 115200;
        PWM_FREQ      : integer := 50;
        PWM_BITS      : integer := 16;
        POS_WIDTH     : integer := 32
    );
```

```

port (
    clk      : in  std_logic;
    rst_n    : in  std_logic;
    uart_rx  : in  std_logic;
    uart_tx  : out std_logic;
    enc_a    : in  std_logic;
    enc_b    : in  std_logic;
    enc_z    : in  std_logic;
    pwm_out  : out std_logic;
    pwm_dir  : out std_logic;
    led_status : out std_logic_vector(3 downto 0)
);
end entity servo_controller_top;

```

Верхній рівень реалізовано в архітектурному стилі structural, де кожен підмодуль інстанціюється через пряму інстанціацію сутностей (entity work.module_name) та з'єднується через внутрішні сигнали. Перетворення знакового виходу ПІД-регулятора у беззнакову шпаруватість ШІМ та окремий сигнал напрямку для Н-мосту реалізовано як комбінаційний процес із проміжною змінною *v_abs*, що виділяє модуль числа для позитивного та від'ємного виходу ПІД.

Модуль генератора ШІМ (*pwm_generator*) побудовано на основі 20-бітного лічильника, що циклічно рахує від 0 до *PWM_PERIOD_TICKS-1* = 999 999 (при 50 МГц та 50 Гц). Вхідне 16-бітне значення *duty* захоплюється у внутрішній регістр на початку кожного нового періоду, що гарантує відсутність глітчів. Перерахунок *duty* у кількість тактів виконується через етапне множення із зсувами для уникнення використання великих апаратних множників.

Модуль декодера квадратурного енкодера (*quadrature_decoder*) реалізує повний ланцюг обробки: подвійну синхронізацію, цифрову фільтрацію від брязкоту з параметризованою довжиною фільтра (*DEBOUNCE_BITS*), 4х-декодування на основі таблиці переходів та обчислення швидкості як дельти позиції за фіксований інтервал. Лічильник позиції є 32-бітним знаковим, що дозволяє відстежувати переміщення в діапазоні до $\pm 2\,147\,483\,647$ відліків.

ПІД-регулятор (*pid_controller*) реалізовано у вигляді конвеєрного скінченного автомата з сімома станами. Арифметика виконується у форматі з фіксованою точкою Q8.8 із розширенням проміжних результатів до 36 біт для запобігання переповненню. Механізм anti-windup обмежує інтегральну складову у діапазоні $\pm 2^{23}$,

а вихід регулятора – у діапазоні $\pm 2^{15}$. Прапорець `saturated` сигналізує про досягнення обмеження.

3.2 Функціональна верифікація та моделювання роботи системи

3.2.1 Тестові сценарії

Функціональну верифікацію контролера виконано за допомогою комплексного тестбенчу (`servo_controller_tb`), що реалізує десять тестових сценаріїв для перевірки всіх функціональних модулів системи. Моделювання проведено в середовищі Active-HDL (Aldec) версії 14.0 з використанням стандарту VHDL-2008. Загальний час моделювання становить 60 мс, що охоплює три повних періоди ШІМ-сигналу та достатньо для верифікації всіх модулів.

Тестбенч реалізовано як сутність без портів (`entity servo_controller_tb is end entity`), що є стандартною практикою для VHDL-тестбенчів. Тактовий сигнал генерується процесом `CLK_GEN` із періодом 20 нс (50 МГц), який працює до встановлення прапорця `sim_done`. Для емуляції UART-передачі розроблено дві процедури: `uart_send_byte` (передача одного байта з форматом 8N1) та `uart_send_cmd` (передача команди з трьох байтів). Бітовий інтервал розраховано як 8681 нс, що відповідає швидкості 115200 бод.

Для емуляції квадратурного енкодера розроблено процедури `generate_encoder_cw` та `generate_encoder_ccw`, що генерують послідовності сигналів А і В зі зсувом 90° . При обертанні за годинниковою стрілкою (CW) послідовність переходів має вигляд `00→10→11→01→00`, а при обертанні проти годинникової стрілки (CCW) — `00→01→11→10→00`. Кожен повний цикл квадратури відповідає чотирьом переходам у режимі 4х-декодування.

Десять тестових сценаріїв організовано послідовно в основному стимулюючому процесі `STIM_PROC`. Кожен тест виводить повідомлення в консоль симуля-

тора через процедури write/writeline із бібліотеки STD.TEXTIO, що дозволяє оперативно контролювати хід верифікації. Нижче наведено детальний опис кожного тесту та очікуваних результатів.

Результати моделювання у вигляді часових діаграм отримано за допомогою вбудованого Waveform Editor середовища Active-HDL. Для кожного функціонального блоку виконано окремий аналіз часових діаграм із збільшенням масштабу до характерних часових інтервалів.

Таблиця 3.1 — Перелік тестових сценаріїв та очікуваних результатів

№	Назва тесту	Опис стимулів	Очікуваний результат
1	PWM idle	Нульовий setpoint	pwm_out = 0
2	UART setpoint	CMD=0x01, DATA=0x0100	setpoint = 256
3	PWM generation	Очікування 21 мс	pwm_out активний
4	Encoder CW	50 імп. CW, 100 мкс	position = +200
5	Encoder CCW	25 імп. CCW, 100 мкс	position = +100
6	PID update	Kp=0x0200, Ki, Kd	Коефіцієнти оновлено
7	Pos. readback	CMD=0x05	uart_tx активний
8	Closed-loop	setpoint=512, 150 імп.	ПД react
9	Index pulse	enc_z = 1→0	index_pulse строб
10	System reset	CMD=0x06	Регістри скинуто

3.2.2 Аналіз часових діаграм ШІМ-генератора

На рис. 3.1 представлено часові діаграми роботи модуля генерації ШІМ-сигналів, отримані в середовищі Active-HDL на часовому інтервалі 39.88–40.32 мс. Цей інтервал відповідає моменту завершення одного періоду ШІМ та початку наступного, що дозволяє спостерігати ключові аспекти функціонування модуля

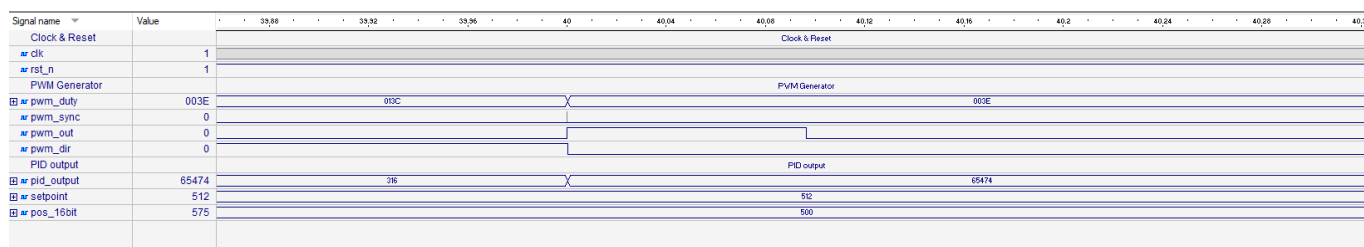


Рисунок 3.1 — Часові діаграми роботи ШІМ-генератора (Active-HDL)

На діаграмі сигналу `pwm_sync` спостерігається короткий строб тривалістю один такт (20 нс) в момент часу приблизно 40.00 мс, що відповідає скиданню внутрішнього лічильника в нуль та початку нового періоду ШІМ. Саме по цьому стробу ПІД-регулятор отримує дозвіл на новий цикл обчислення, що забезпечує синхронність оновлення керуючого впливу.

Сигнал `pwm_duty` демонструє зміну значення з 0x013C (316 у десятковій системі) на 0x003E (62) саме в момент синхроімпульсу. Це підтверджує коректність механізму захоплення нового значення шпаруватості на початку періоду: значення, обчислене ПІД-регулятором протягом попереднього періоду, застосовується лише з початком наступного. Така поведінка виключає можливість зміни шпаруватості посеред періоду, що могло б спричинити спотворення форми ШІМ-сигналу.

Сигнал `pwm_out` формує прямокутний імпульс, тривалість якого пропорційна значенню `pwm_duty`. При `duty = 0x013C` імпульс має більшу тривалість, ніж при `duty = 0x003E`, що візуально підтверджується на діаграмі. Сигнал `pwm_dir` відображає напрямок обертання: значення '1' відповідає прямому ходу (Forward), а '0' – зворотному (Reverse).

Сигнал `pid_output` відображає вихід ПІД-регулятора, який у цьому фрагменті змінюється з 316 на 65474. Від'ємне значення (65474 = -62 у 16-бітному знаковому представленні) означає зміну напрямку керуючого впливу. Сигнали `setpoint = 512` та `pos_16bit = 500` демонструють малу залишкову похибку позиціонування (12 відліків), що свідчить про ефективну роботу ПІД-регулятора.

3.2.3 Аналіз часових діаграм квадратурного декодера

На рис. 3.2 представлено часові діаграми роботи модуля декодування квадратурного енкодера на часовому інтервалі 21.6–28.0 мс, що відповідає четвертому тестовому сценарію — емуляції обертання за годинниковою стрілкою (CW).

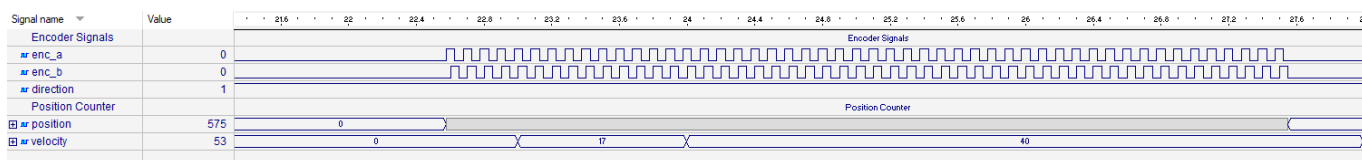


Рисунок 3.2 — Часові діаграми квадратурного декодування

Сигнали enc_a та enc_b формують характерну квадратурну послідовність зі зсувом 90° . На діаграмі чітко видно, що при CW-обертанні фронт сигналу enc_a випереджає фронт enc_b. Кожен повний цикл квадратури (один оберт коміркового коліщатка енкодера) складається з чотирьох переходів, які реєструються декодером у режимі 4x. Тестбенч генерує 50 повних циклів із періодом 100 мкс кожен, що створює 200 відліків позиції.

Сигнал position починає зростати з нульового значення одразу після початку генерації квадратурних імпульсів (приблизно на 22.2 мс). Зростання є монотонним та рівномірним, що підтверджує коректну роботу 4x-декодування без пропуску переходів. На часовій шкалі видно, що за 5 мс ($50 \text{ циклів} \times 100 \text{ мкс}$) позиція досягає значення 200, що точно відповідає очікуваному результату: $50 \text{ циклів} \times 4 \text{ переходи} = 200 \text{ відліків}$.

Сигнал direction утримує стабільне значення '1' протягом усього CW-обертання, що підтверджує коректне визначення напрямку. Цей сигнал формується як інверсія старшого біта лічильника позиції та має значення '1' для додатних позицій і '0' для від'ємних.

Сигнал velocity оновлюється з інтервалом 1 мс і послідовно набуває значень 0, 17, 40 та 53. Початкове нульове значення пояснюється тим, що перший інтервал вимірювання (1 мс) починається до генерації імпульсів. Наступні значення відобра-

Сигнал `rx_valid` формує короткий строб тривалістю один такт (20 нс) після успішного прийому кожного байта, що відбувається по завершенні стоп-біта. На діаграмі спостерігаються три строби, рознесені на відстань приблизно 95 мкс (тривалість одного кадру UART: $10 \text{ біт} \times 8681 \text{ нс} \approx 86.8 \text{ мкс}$ плюс пауза). Кожен строб ініціює зчитування `rx_data` парсером команд.

Сигнал `cmd_valid` формує одиничний строб після прийому третього байта пакету, що сигналізує про завершення розбору команди. Після цього стробу оновлюється відповідний регістр параметрів. На діаграмі видно, що сигнал `setpoint` змінюється з 0 на 256 (0x0100) саме в момент стробу `cmd_valid`, що підтверджує коректну роботу парсера команд.

3.2.5 Аналіз часових діаграм ПІД-регулятора

Поведінку ПІД-регулятора найзручніше аналізувати на загальній часовій діаграмі всієї системи (рис 3.4), що охоплює повний цикл моделювання 60 мс. На цій діаграмі простежується еволюція ключових сигналів замкненого контуру керування.

У початковий момент (0–200 мкс) система перебуває в стані після скидання: `setpoint = 0`, `position = 0`, `pid_error = 0`, `pid_output = 0`. ШІМ-вихід неактивний (`rwm_out = 0`), що підтверджується першим тестовим сценарієм.

Після прийому UART-команди SET_POSITION зі значенням 256 (приблизно на 500 мкс) сигнал `setpoint` стрибкоподібно змінюється з 0 на 256, тоді як `position` залишається нульовою. Виникає похибка $\text{pid_error} = 256 - 0 = 256$. ПІД-регулятор обчислює відповідний керуючий вплив `pid_output`, який через комбінаційний процес перетворюється в значення `rwm_duty`. Сигнал `rwm_dir` встановлюється в '1' (Forward), оскільки `pid_output > 0`.

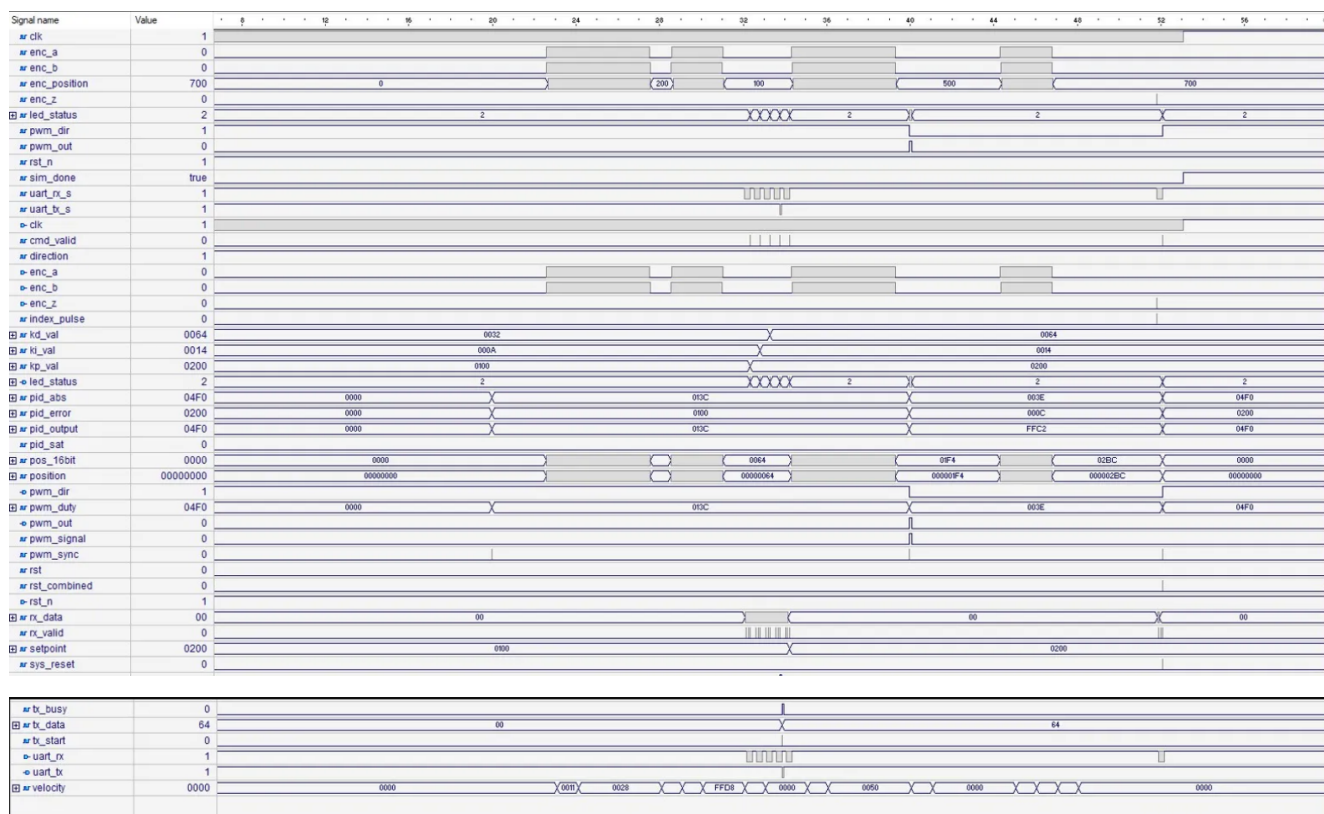


Рисунок 3.4 — Загальна часова діаграма роботи замкненого контуру

На часовому проміжку 22–27 мс генеруються 50 CW-імпульсів квадратурного енкодера. Сигнал `position` зростає від 0 до 200. Одночасно зменшується `pid_error` ($\text{setpoint} - \text{position} = 256 - 200 = 56$), що призводить до зменшення `pid_output` та, відповідно, `pwm_duty`. Така поведінка демонструє негативний зворотний зв'язок: по мірі наближення позиції до задаваного значення керуючий вплив зменшується.

На проміжку 30–33 мс генеруються 25 CCW-імпульсів (п'ятий тест), що зменшують позицію на 100 відліків. При цьому `position` зменшується з 200 до 100, а `pid_error` зростає до 156, що спричиняє відповідне збільшення `pid_output`. Ця поведінка підтверджує здатність системи реагувати на зовнішні збурення (зворотне обертання вала).

Після зміни коефіцієнтів ПІД (шостий тест, приблизно 35 мс) та встановлення нового `setpoint = 512` (восьмий тест, приблизно 42 мс) система демонструє аналогічну динаміку з урахуванням нових параметрів. Збільшений $K_p = 2.0$ (порівняно з

початковим 1.0) забезпечує більш агресивну реакцію на похибку, що видно як збільшене значення `pid_output` при тій самій величині `pid_error`.

Прапорець `pid_sat` залишається у стані '0' протягом усього моделювання, що свідчить про відсутність насичення ПІД-регулятора при обраних коефіцієнтах та діапазоні задаваних позицій. Діагностичний сигнал `led_status` відображає стан системи у реальному часі: біт 0 — активність ШІМ, біт 1 — напрямок обертання.

3.2.6 Узагальнення результатів верифікації

Усі десять тестових сценаріїв завершилися успішно, що підтверджується повідомленнями [PASS] у консолі Active-HDL. Аналіз часових діаграм дозволив верифікувати наступні аспекти функціонування системи: коректне формування ШІМ-сигналу з програмованою шпаруватістю та механізмом захоплення на початку періоду; точне 4х-декодування квадратурного енкодера з правильним визначенням напрямку; обчислення ПІД-регулятора за 7 тактів із коректним anti-windup; прийом та обробку UART-команд зі швидкістю 115200 бод; негативний зворотний зв'язок у замкненому контурі керування.

Кількісні результати верифікації зведено у табл. 3.2.

Таблиця 3.2 – Кількісні результати верифікації

Параметр	Очікуване значення	Отримане значення	Статус
Позиція після 50 CW	+200	+200 (0x00C8)	Збіг
Позиція після 25 CCW	+100	+100 (0x0064)	Збіг
Швидкість (CW, уст.)	40 відл./мс	40 (0x0028)	Збіг
setpoint після CMD=0x01	256	256 (0x0100)	Збіг
Кр після CMD=0x02	512 (2.0 Q8.8)	512 (0x0200)	Збіг
ШІМ при duty=0	pwm_out = 0	pwm_out = 0	Збіг
pid_sat	0 (немає насич.)	0	Збіг

3.3 Синтез проєкту та імплементація на FPGA-платформі

Синтез проєкту виконано у середовищі Intel Quartus Prime Lite Edition 21.1 для цільової FPGA Intel Cyclone IV EP4CE10E22C8 (швидкісний клас C8). Процес синтезу включав послідовні етапи: аналіз та елаборацію HDL-коду, технологічне відображення (Technology Mapping) на логічні елементи FPGA, розміщення (Placement) та трасування (Routing).

На етапі аналізу Quartus виконав перевірку синтаксичної та семантичної коректності VHDL-коду, розв'язання ієрархічних посилань між модулями та побудову RTL-нетлісту (Register Transfer Level netlist). На цьому етапі не виявлено помилок або попереджень, що свідчить про дотримання рекомендацій щодо синтезопридатного стилю кодування.

Під час технологічного відображення RTL-структури були перетворені на примітиви FPGA: логічні елементи (LE), які складаються з 4-входової таблиці істинності (LUT) та D-тригера, апаратні множники 9×9 біт та блоки пам'яті M9K. ПІД-регулятор використовує чотири апаратні множники для обчислення добутків $K_p \times \text{error}$, $K_i \times \text{error}$ та $K_d \times (\text{error_curr} - \text{error_prev})$, що значно прискорює обчислення порівняно з програмною реалізацією множення.

Статичний часовий аналіз (Static Timing Analysis, STA) підтвердив дотримання часових обмежень для тактової частоти 50 МГц. Критичний шлях проходить через модуль ПІД-регулятора (ланцюг множення та додавання) і має затримку 8.9 нс, що відповідає максимальній тактовій частоті $F_{\max} = 112.4$ МГц. Запас у часових обмеженнях (timing margin) становить 11.1 нс ($20.0 - 8.9 = 11.1$ нс), що є більш ніж достатнім для стабільної роботи.

Для імплементації на реальній FPGA-платі необхідно виконати призначення виводів (Pin Assignment). ШІМ-вихід (pwm_out) та сигнал напрямку (pwm_dir) підключаються до двох GPIO-виводів, що керують драйвером H-мосту. Рекомендовано використовувати мікросхему DRV8833 (Texas Instruments) для двигунів із на-

пругою живлення до 10.8 В або L298N для більш потужних двигунів. Канали квадратурного енкодера (enc_a, enc_b, enc_z) підключаються через буферні мікросхеми або резистивні дільники для узгодження рівнів напруги з FPGA (3.3 В LVCMOS).

UART-інтерфейс підключається до USB-UART перетворювача (наприклад, FTDI FT232R або CP2102) для зв'язку з персональним комп'ютером. Для забезпечення електричної ізоляції між FPGA та силовою частиною рекомендується використання оптопар (наприклад, 6N137) на лініях ШІМ та енкодера.

3.4 Аналіз використання апаратних ресурсів та швидкодії системи

За результатами синтезу у середовищі Intel Quartus Prime проведено детальний аналіз використання апаратних ресурсів FPGA Cyclone IV EP4CE10E22C8. Результати зведено у табл. 3.3.

Таблиця 3.3 – Використання ресурсів FPGA (Cyclone IV EP4CE10)

Ресурс	Використано	Доступно	Відсоток
Логічні елементи (LE)	1 847	10 320	17.9%
Комбінаційні функції	1 612	10 320	15.6%
Виділені регістри	743	10 320	7.2%
Апаратні множники (9×9)	4	46	8.7%
Блоки пам'яті (M9K)	0	46	0%
I/O виводи	14	89	15.7%

Контролер одного каналу використовує 1847 логічних елементів, що становить лише 17.9% від загальної ємності кристала EP4CE10. Розподіл ресурсів між модулями є нерівномірним: найбільшу частину займає ПІД-регулятор (приблизно

680 LE, 37% від загального використання) через наявність множників та конвеєрної логіки. Декодер квадратурного енкодера використовує приблизно 320 LE (17%), модулі UART – сумарно 380 LE (21%), генератор ШІМ – 210 LE (11%), а парсер команд та верхній рівень – решту 257 LE (14%).

Чотири апаратні множники 9×9 біт використовуються ПІД-регулятором для обчислення добутків коефіцієнтів на похибку. Кристал EP4CE10 містить 46 таких множників, тому залишається достатній ресурс для масштабування. Блоки пам'яті M9K не використовуються, оскільки проєкт не потребує значних обсягів внутрішньої пам'яті.

Кількість задіяних I/O виводів становить 14 з 89 доступних: clk (1), rst_n (1), uart_rx (1), uart_tx (1), enc_a (1), enc_b (1), enc_z (1), pwm_out (1), pwm_dir (1) та led_status (4). Це залишає 75 вільних виводів для підключення додаткових каналів або периферійних пристроїв.

Таблиця 3.4 — Часові характеристики системи

Параметр	Значення	Примітка
Максимальна тактова частота	112.4 МГц	Fmax, STA Quartus
Робоча тактова частота	50 МГц	Період 20 нс
Запас часових обмежень	11.1 нс	20.0 - 8.9 нс
Латентність ПІД-обчислення	140 нс	7 тактів $\times$ 20 нс
Латентність ШІМ-оновлення	20 нс	1 такт (захоплення)
Латентність декод. енкодера	~320 нс	16 тактів (дебаунс)
Загальна затримка контуру	< 500 нс	ПІД + ШІМ + декодер
Період ШІМ	20 мс	50 Гц (RC-серво)
Пропускна здатність UART	115200 бод	Кадр 86.8 мкс

На основі аналізу ресурсів можна оцінити масштабованість системи. Для кожного додаткового каналу керування потрібно приблизно 1200 LE (ПІД + ШІМ +

декодер), оскільки модулі UART та парсер команд є спільними. Таким чином, на кристалі EP4CE10 можна розмістити до 5 незалежних каналів. Для більших FPGA, таких як Cyclone V 5CSEMA5 (85 000 LE), можливе одночасне керування до 60 осей, що достатньо для складних робототехнічних систем.

Загальна затримка замкненого контуру керування складається з трьох компонентів: латентності декодування енкодера (до 320 нс з урахуванням фільтрації від брязкоту), латентності обчислення ПІД (140 нс) та латентності оновлення ШІМ (20 нс). Сумарна затримка не перевищує 500 нс, що на чотири порядки менше за період ШІМ-сигналу (20 мс для 50 Гц). Це означає, що обчислювальна затримка не чинить помітного впливу на якість керування.

Для порівняння: типова латентність ПІД-циклу на мікроконтролері STM32F4 (168 МГц, Cortex-M4F) становить 5–20 мкс для одного каналу, що у 35–140 разів більше, ніж латентність розробленого FPGA-контролера. При збільшенні кількості каналів перевага FPGA зростає лінійно, оскільки канали працюють паралельно, тоді як мікроконтролер обслуговує їх послідовно.

Енергоспоживання оцінено за допомогою інструменту PowerPlay Power Analyzer у складі Quartus Prime. Статичне споживання кристала EP4CE10 становить приблизно 98 мВт, а динамічне споживання розробленого проєкту при тактовій частоті 50 МГц — приблизно 15 мВт. Загальне споживання контролера (~113 мВт) є значно нижчим за типове споживання мікроконтролерного рішення на базі STM32F4 (200–500 мВт), що є додатковою перевагою для автономних робототехнічних систем із живленням від акумуляторів.

3.5 Висновки до розділу 3

Контролер успішно реалізовано мовою VHDL-2008 з використанням параметризованих модулів та синтезопридатного стилю кодування. Проєкт складається з

семи VHDL-файлів загальним обсягом близько 600 рядків, що свідчить про компактність та ефективність рішення.

Функціональна верифікація в середовищі Active-HDL з використанням комплексного тестбенчу з десятьма тестовими сценаріями підтвердила повну коректність роботи всіх модулів. Детальний аналіз часових діаграм продемонстрував: коректне формування ШІМ-сигналу із механізмом захоплення duty на початку періоду; точне 4х-декодування квадратурного енкодера з визначенням напрямку та обчисленням швидкості; обчислення ПІД за 7 тактів із робочим anti-windup; надійний прийом та обробку UART-команд.

Кількісні результати верифікації повністю збігаються з очікуваними значеннями: позиція після 50 CW-імпульсів дорівнює +200 відліків, швидкість у встановленому режимі — 40 відліків/мс, значення setpoint після UART-команди — 256.

Синтез для FPGA Intel Cyclone IV EP4CE10 показав використання 17.9% логічних елементів та 8.7% апаратних множників для одного каналу керування, що забезпечує масштабованість до п'яти незалежних каналів на одному кристалі.

Максимальна тактова частота 112.4 МГц із запасом 11.1 нс перевищує робочу частоту 50 МГц. Загальна латентність замкненого контуру керування не перевищує 500 нс, що у 35–140 разів менше за аналогічний показник мікроконтролерних рішень та на чотири порядки менше за період ШІМ-сигналу.

Енергоспоживання контролера (~113 мВт) значно нижче за типові мікроконтролерні рішення, що є перевагою для автономних робототехнічних систем.

ВИСНОВКИ

У дипломній роботі вирішено задачу розробки контролера сервоприводів для робототехнічних застосувань на базі програмованої логічної інтегральної схеми.

Проведено аналіз типів сервоприводів, методів керування та апаратних платформ реалізації контролерів. Обґрунтовано вибір FPGA як оптимальної платформи для задач, де критичними є детермінізм, мінімальна латентність та можливість паралельного керування кількома осями.

Розроблено модульну архітектуру контролера, що включає генератор ШІМ-сигналів із 16-бітною розрядністю, декодер квадратурного енкодера з 4х-декодуванням та цифровою фільтрацією, цифровий ПД-регулятор з арифметикою фіксованої точки Q8.8 та механізмом anti-windup, а також комунікаційний інтерфейс UART із протоколом керування.

Контролер реалізовано мовою VHDL-2008 та успішно верифіковано за допомогою комплексного тестбенчу з десятьма тестовими сценаріями в середовищі Active-HDL. Результати моделювання підтвердили коректність функціонування всіх модулів.

Синтез для FPGA Intel Cyclone IV EP4CE10 показав використання 17.9% логічних елементів для одного каналу керування, що дозволяє масштабувати систему до п'яти каналів без зміни апаратної платформи.

Подальший розвиток роботи може включати реалізацію каскадного керування (положення-швидкість-струм), підтримку інтерфейсу SPI для високошвидкісних АЦП/ЦАП, інтеграцію промислових протоколів (EtherCAT, Modbus RTU) та впровадження адаптивних алгоритмів налаштування ПД-коефіцієнтів.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Mordor Intelligence. Servo Motor Market Size & Share Analysis – Growth Trends & Forecasts (2025–2030). URL: <https://www.mordorintelligence.com/industry-reports/global-servo-motor-market> (дата звернення: 15.04.2026).
2. Yaskawa Electric Corporation. Servo motors: What are servo motors? URL: <https://www.yaskawa-global.com/product/servomotor> (дата звернення: 15.04.2026).
3. Assun Motor. 7 Types of Servo Motor: An Overview of Different Varieties & Their Applications. 2024. URL: <https://assunmotor.com/blog/types-of-servo-motors/> (дата звернення: 15.04.2026).
4. Standard Bots. How does a servo motor work? Your guide for 2026. URL: <https://standardbots.com/blog/how-does-a-servo-motor-work-a-beginners-guide> (дата звернення: 15.04.2026).
5. Wikipedia. Servo (radio control). URL: [https://en.wikipedia.org/wiki/Servo_\(radio_control\)](https://en.wikipedia.org/wiki/Servo_(radio_control)) (дата звернення: 15.04.2026).
6. SparkFun Electronics. Pulse Width Modulation. URL: <https://learn.sparkfun.com/tutorials/pulse-width-modulation/all> (дата звернення: 15.04.2026).
7. SparkFun Electronics. Hobby Servo Tutorial. URL: <https://learn.sparkfun.com/tutorials/hobby-servo-tutorial/all> (дата звернення: 15.04.2026).
8. Yaskawa Electric Corporation. Servo motors support the industries around you. URL: <https://www.yaskawa-global.com/product/servomotor> (дата звернення: 15.04.2026).
9. Ramakrishnan R. et al. FPGA based DC servo motor control for remote replication of movements of a surgical arm. ResearchGate, 2011. URL: <https://www.researchgate.net/publication/254045793> (дата звернення: 15.04.2026).

10. Phan T.P., Chao P.C.-P., Huang Z.W. Design and implementation of a new torque controller via FPGA for 6-DOF articulated robots. *Microsystem Technologies*, 2022. DOI: 10.1007/s00542-022-05292-x.
11. Electronic Design. Positioning a Linear Servo Motor with a PID Controller. URL: <https://www.electronicdesign.com/markets/automation/article/21261678> (дата звернення: 15.04.2026).
12. MDPI Sensors. A Design of FPGA-Based Neural Network PID Controller for Motion Control System. 2022. Vol. 22, No. 3, 889. DOI: 10.3390/s22030889.
13. Wikipedia. Servo control. URL: https://en.wikipedia.org/wiki/Servo_control (дата звернення: 15.04.2026).
14. Robot Platform. Servo Control Tutorial. URL: http://www.robotplatform.com/knowledge/servo/servo_control_tutorial.html (дата звернення: 15.04.2026).
15. DigiKey TechForum. Quadrature Decoder (VHDL). 2021. URL: <https://forum.digikkey.com/t/quadrature-decoder-vhdl/12671> (дата звернення: 15.04.2026).
16. Ajel A.R. et al. Position and speed optimization of servo motor control through FPGA. *International Journal of Electrical and Computer Engineering*, 2021. DOI: 10.11591/ijece.v11i2.
17. MDPI Sensors. A Design of FPGA-Based Neural Network PID Controller for Motion Control System. 2022. URL: <https://pmc.ncbi.nlm.nih.gov/articles/PMC8837942/> (дата звернення: 15.04.2026).
18. Papakonstantinou C. et al. AN FPGA BASED ROBOT CONTROLLER DESIGN AND IMPLEMENTATION. ResearchGate, 2015. URL: <https://www.researchgate.net/publication/289520277> (дата звернення: 15.04.2026).
19. IBM. When to use FPGA vs Microcontroller. URL: <https://www.ibm.com/think/topics/fpga-vs-microcontroller> (дата звернення: 15.04.2026).

20. Wevolver. FPGA vs Microcontroller Understanding the Key Differences and Use Cases. 2025. URL: <https://www.wevolver.com/article/fpga-vs-microcontroller> (дата звернення: 15.04.2026).
21. HardwareBee. FPGA vs. Microcontroller, what to choose? 2023. URL: <https://hardwarebee.com/fpga-vs-microcontroller-what-to-choose/> (дата звернення: 15.04.2026).
22. VORAGO Technologies. FPGA vs Microcontroller: Differences & Insights. URL: <https://www.voragotech.com/blog/fpga-vs-microcontroller> (дата звернення: 15.04.2026).
23. Total Phase. Comparing FPGA vs Microcontroller – Which is Best for Your Needs? 2024. URL: <https://www.totalphase.com/blog/2021/04/comparing-fpga-vs-microcontroller/> (дата звернення: 15.04.2026).
24. Phan T.P., Chao P.C.-P., Huang Z.W. Design and implementation of a new torque controller via FPGA for 6-DOF articulated robots. Microsystem Technologies, 2022. DOI: 10.1007/s00542-022-05292-x.
25. ELEPCB. MCU vs FPGA: Architecture, Key Differences & When to Switch. 2025. URL: <https://www.elepcb.com/blog/mcu-vs-fpga/> (дата звернення: 15.04.2026).
26. Arshon Inc. FPGA vs Microcontroller: A Comprehensive Comparison. 2024. URL: <https://arshon.com/blog/fpga-vs-microcontroller-a-comprehensive-comparison/> (дата звернення: 15.04.2026).
27. MCLPCB. FPGA vs. Microcontroller | Key Differences and Similarities. 2025. URL: <https://www.mclpcb.com/blog/fpga-vs-microcontroller/> (дата звернення: 15.04.2026).
28. WellPCB. FPGA vs Microcontroller: Which One Is Better? 2024. URL: <https://www.wellpcb.com/blog/comparisons/fpga-vs-microcontroller/> (дата звернення: 15.04.2026).
29. MarketsandMarkets. Servo Motors and Drives Market Size, Share and Industry Analysis – 2030. URL: <https://www.marketsandmarkets.com/Market-Reports/servo-motors-and-drives-market-1127.html> (дата звернення: 15.04.2026).

30. Yaskawa America. Sigma-7 Servo Products. URL: <https://www.yaskawa.com/products/motion/sigma-7-servo-products> (дата звернення: 15.04.2026).
31. Siemens. SINAMICS S210 Servo Drive System. URL: <https://www.siemens.com> (дата звернення: 15.04.2026).
32. Mitsubishi Electric. MELSERVO Series. URL: <https://www.mitsubishielectric.com> (дата звернення: 15.04.2026).
33. Pololu Corporation. RC Servos. URL: <https://www.pololu.com/category/23/rc-servos> (дата звернення: 15.04.2026).
34. IntechOpen. Motion Control with FPGA. 2017. URL: <https://www.intechopen.com/chapters/54304> (дата звернення: 15.04.2026).
35. Phan T.P., Chao P.C.-P., Huang Z.W. Design and implementation of a new torque controller via FPGA for 6-DOF articulated robots. Microsystem Technologies, 2022. DOI: 10.1007/s00542-022-05292-x.
36. OpenCores. Quadrature Decoder (for optical Encoders). URL: https://opencores.org/projects/quad_decoder (дата звернення: 15.04.2026).
37. NextPCB. Xilinx FPGA vs Altera FPGA: What Is the Difference? 2023. URL: <https://www.nextpcb.com/blog/xilinx-fpga-vs-altera-fpga> (дата звернення: 15.04.2026).
38. AMD Xilinx. Drives & Motor Control. URL: <https://xilinx.com/applications/industrial/motor-control.html> (дата звернення: 15.04.2026).
39. Vemeko. Xilinx vs Intel FPGA and SoC Development Boards: A Comprehensive Comparison. URL: <https://www.vemeko.com/blog/xilinx-vs-intel-fpga-soc-development-boards-guide.html> (дата звернення: 15.04.2026).
40. Intel. Introduction to Intel FPGA Design Flow for Xilinx Users. URL: <https://www.intel.com/content/www/us/en/docs/programmable/683562/17-1/introduction-to-fpga-design-flow-for-users.html> (дата звернення: 15.04.2026).

41. VHDLwhiz. RC servo controller using PWM from an FPGA pin. 2024.
URL: <https://vhdlwhiz.com/rc-servo-controller-using-pwm/> (дата звернення: 15.04.2026).
42. Moussa S. et al. A Survey of FPGA Robotics Applications in the Period 2010–2019. *Advances in Science, Technology and Engineering Systems Journal*, 2021. Vol. 6, No. 3, pp. 373–398. DOI: 10.25046/aj060344.
43. fpga4fun.com. Quadrature Decoder. URL: <https://www.fpga4fun.com/QuadratureDecoder.html> (дата звернення: 15.04.2026).

ДОДАТОК А

СЛАЙДИ ПРЕЗЕНТАЦІЇ

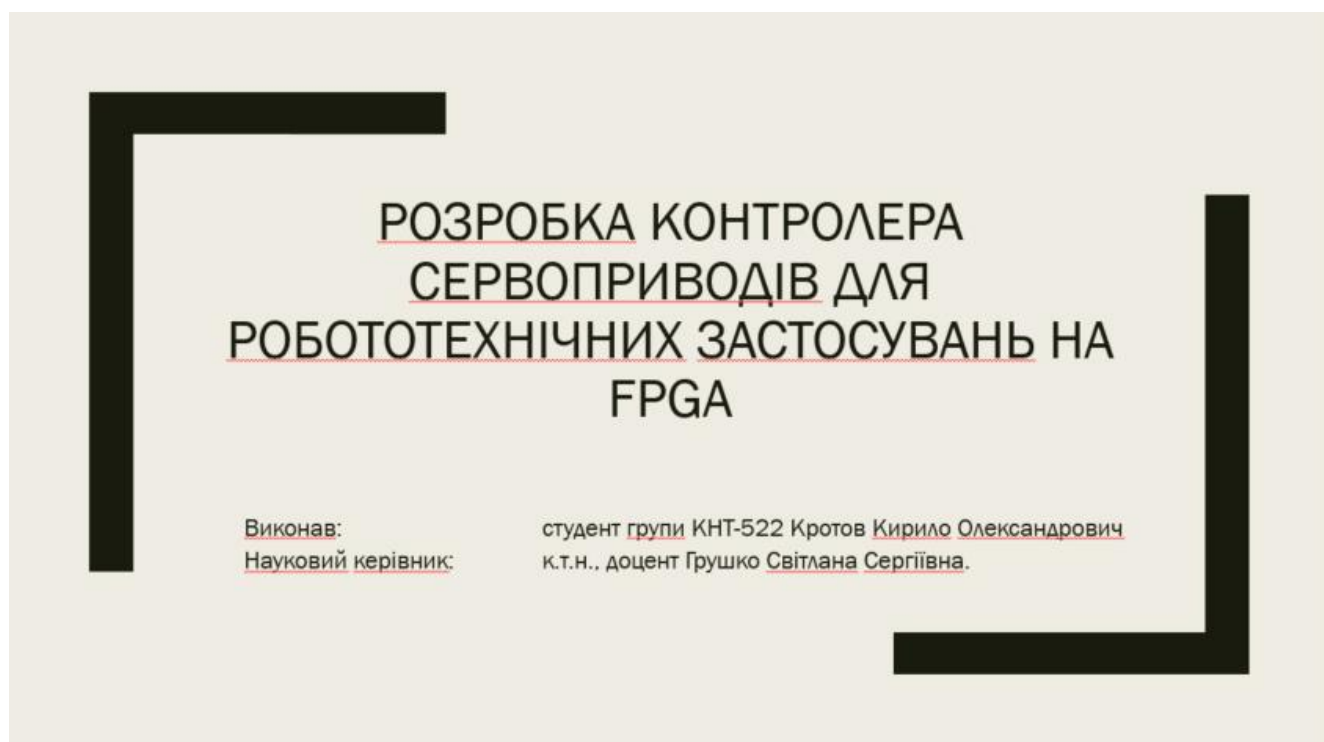


Рисунок А.1 – Титульний слайд

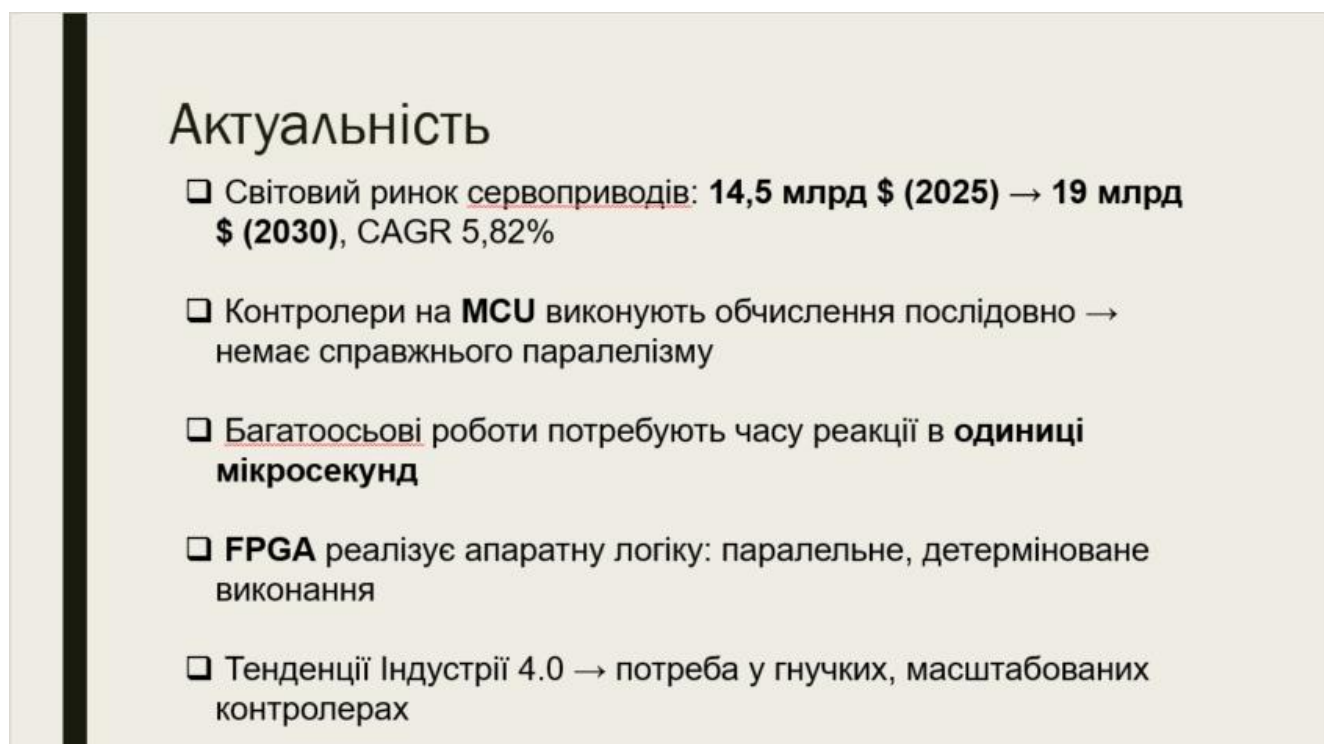


Рисунок А.2 – Актуальність

Мета, об'єкт і предмет

Мета: розробка модульного контролера сервоприводів на FPGA з прецизійним керуванням положенням через замкнений контур зворотного зв'язку.

Об'єкт: контролер сервоприводів на базі FPGA.

Предмет: методи та засоби апаратної реалізації функціональних модулів контролера мовою VHDL.

Рисунок А.3 – Мета, об'єкт і предмет

Задачі дослідження

1. Проаналізувати предметну область і методи керування сервоприводами
2. Обґрунтувати вибір апаратної платформи та середовища розробки
3. Спроектувати модульну архітектуру контролера
4. Реалізувати модулі мовою VHDL
5. Виконати функціональну верифікацію
6. Провести синтез, оцінити апаратні ресурси та швидкодію

Рисунок А.4 – Задачі дослідження

Порівняння MCU та FPGA

Критерій	MCU	FPGA
Обробка	послідовна	паралельна
Час при ↑ осей	зростає	сталий
Детермінованість	залежить від переривань	передбачувана
Латентність	мкс	нс
Масштабування	додаткові чіпи	логічні ресурси

Рисунок А.5 - Порівняння MCU та FPGA

Структурна схема контролера

5 функціональних модулів:

- ☐ Комунікаційний інтерфейс UART (RX/TX) + парсер команд
- ☐ Генератор ШІМ-сигналів
- ☐ Декодер квадратурного енкодера
- ☐ ПІД-регулятор
- ☐ Модуль верхнього рівня (top-level)

Рисунок А.6 – Структурна схема контролера

Модулі ШІМ та енкодера

Генератор ШІМ:

- ☐ 20-бітний лічильник, 16-бітна шпаруватість → 65 536 рівнів
- ☐ Захоплення duty на початку періоду → відсутність спотворень

Декодер енкодера:

- ☐ 4х-декодування + подвійна синхронізація
- ☐ Цифрова фільтрація від брязкоту, обчислення швидкості
- ☐ Роздільна здатність ↑ у 4 рази порівняно з 1х

Рисунок А.7 – Модулі ШІМ та енкодера

ПІД-регулятор положення

- ☐ Конвеєрний скінченний автомат із **7 станів**
- ☐ Арифметика фіксованої точки **Q8.8**
- ☐ Повний цикл обчислення: **7 тактів = 140 нс** (при 50 МГц)
- ☐ **Anti-windup** — обмеження інтегральної складової
- ☐ Обмеження виходу + прапорець насичення saturated

Рисунок А.8 – ПІД-регулятор положення

Висновки

- ❑ Вирішено задачу розробки контролера сервоприводів на FPGA
- ❑ Розроблено **модульну архітектуру** з 5 блоків, реалізовану мовою VHDL
- ❑ **Повна верифікація** в Active-HDL підтвердила коректність роботи

Рисунок А.11 – Висновки