Хмельницький національний університет
Факультет інформаційних технологій
Кафедра комп'ютерної інженерії та інформаційних систем

# КВАЛІФІКАЦІЙНА РОБОТА

Криптографічний прискорювач на базі ПЛІС (FPGA) для захищеного передавання даних
Назва теми

Рівень вищої освіти *перший (бакалаврський)*

Галузь знань *12 «Інформаційні технології»*
Шифр, назва

Спеціальність *123 «Комп'ютерна інженерія»*
Шифр, назва

Освітня програма *«Комп'ютерна інженерія та програмування»*
Назва

Шифр КвРКІ 22115.22.02.07 ПЗ

Виконав здобувач ІІІ курсу, група КІ2с-23-2 _____ Богдан ІЛЮШЕНКО
Підпис Ініціали, прізвище

Керівник канд. екон.наук, доцент _____ Світлана САЧЕНКО
Науковий ступінь, учене звання Підпис Ініціали, прізвище

Нормоконтролер канд. фіз.-мат. наук, доц. _____ Тетяна КИСІЛЬ
Науковий ступінь, учене звання Підпис Ініціали, прізвище

До захисту допускаю:
завідувач кафедри КІПС _____ Ольга ПАВЛОВА
«04» червня 2026 р. підпис Ініціали, прізвище
дата

Хмельницький 2026

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет Інформаційних технологій

Кафедра Комп'ютерної інженерії та інформаційних систем

Рівень вищої освіти перший (бакалаврський)

Галузь знань 12 Інформаційні технології

Спеціальність 123 Комп'ютерна інженерія

Освітня програма «Комп'ютерна інженерія та програмування»

ЗАТВЕРДЖУЮ
Завідувачка кафедри КІІС
Ольга ПАВЛОВА

" 10 " 01 2026 р.

**ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ**

Ілюшенко Богдану Олександровичу
Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Криптографічний прискорювач на базі ПЛІС (FPGA) для захищеного передавання даних
Керівник проекту (роботи) Саченко Світлана Іванівна, к.е.н., доцент
Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

Затверджена наказом ректора університету від 20.01.2026 р. № 7

2. Термін подання здобувачем роботи на кафедру 01.06.2026 р.

3. Вихідні дані до роботи Завдання на кваліфікаційну роботу

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити)
Аналіз предметної області та дослідження криптографічних алгоритмів
Проєктування криптографічного прискорювача на базі ПЛІС Cyclone V для захищеного передавання даних у середовищі Quartus
Реалізація та симуляція криптографічного прискорювача на базі пліс Cyclone V для захищеного передавання даних у середовищі Quartus

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень)
Структура криптографічного прискорювача на базі ПЛІС
Структура блоку шифрування AES-128 на ПЛІС Cyclone V
Результати симуляції в середовищі ModelSim

6. Консультанти розділів кваліфікаційної роботи

| Розділ | Прізвище, ініціали та посада консультанта | Підпис, дата | |
|---|---|---|---|
| | | завдання видав | завдання прийняв |
| | | | |
| | | | |

7. Дата видачі завдання «_10_»_____01_____2026 р.

## КАЛЕНДАРНИЙ ПЛАН

| №з/п | Назва етапів (розділів) дипломного проекту (роботи) | Термін виконання етапів проекту (роботи) | Примітка |
|---|---|---|---|
| 1 | Вибір напряму дослідження та узгодження тематики кваліфікаційної роботи з керівником | 10.01.2026 | виконано |
| 2 | Ознайомлення з предметною областю; формулювання мети та задач дослідження; визначення об'єкта та предмета дослідження | 01.02.2026 | виконано |
| 3 | Робота над розділом 1 – дослідження предметної області та постановка задачі | 01.03.2026 | виконано |
| 4 | Робота над розділом 2 – проєктування структури криптографічного прискорювача | 01.04.2026 | виконано |
| 5 | Робота над розділом 3 – реалізація та симуляція криптографічного прискорювача | 29.04.2026 | виконано |
| 6 | Оформлення пояснювальної записки згідно вимог | 25.05.2026 | виконано |
| 7 | Попередній захист ВКР | 25.05.2026 | виконано |
| 8 | Захист ВКР на засіданні ЕК | Червень 2026 року | |

Здобувач_____ Богдан ЛЮШЕНКО
Підпис Ім'я, ПРІЗВИЩЕ

Керівник кваліфікаційної роботи_____ Світлана САЧЕНКО
Підпис Ім'я, ПРІЗВИЩЕ

| № рядка | формат | Позначення | Найменування | Кіл. листів | № екз | Примітка |
|---|---|---|---|---|---|---|
| | | | Текстові документи | | | |
| 1 | | КвРКІ 22115.22.02.07 ПЗ | Пояснювальна записка | 60 | | |
| | | | | | | |
| | | | Графічні матеріали | | | |
| 2 | | КвРКІ 22115.22.02.07 Е8 | Структура криптографічного прискорювача на базі ПЛІС | 1 | | |
| 3 | | КвРКІ 22115.22.02.07 Е8 | Структура блоку шифрування AES-128 на ПЛІС Cyclone V | 1 | | |
| 4 | | КвРКІ 22115.22.02.07 Е8 | Результати симуляції в середовищі ModelSim | 1 | | |

| Зм | Арк | № докум | Підпис | Дата | КвРКІ 22115.22.02.07 ВП | | |
|---|---|---|---|---|---|---|---|
| Розробив | | Ілюшенко | | | Відомість проекту | Літера | Аркуш 1 / Аркушів 1 |
| Перевір. | | Саченко | | | | У | |
| Н. контр. | | Кисіль | | | | ХНУ, КІ2с-23-2 | |
| Затв. | | Павлова | | | | | |

# АНОТАЦІЯ

Тема кваліфікаційної роботи: «Криптографічний прискорювач на базі ПЛІС (FPGA) для захищеного передавання даних».

Автор роботи: Богдан ІЛЮШЕНКО.

Керівник роботи: Світлана САЧЕНКО.

Пояснювальна записка: 60 с., 18 рис., 1 табл., 3 дод., 46 джерел.

Графічна частина: 3 креслення.

КРИПТОГРАФІЧНИЙ ПРИСКОРЮВАЧ, ПЛІС, ШИФРУВАННЯ.

Сучасний розвиток інформаційно-комунікаційних технологій та стрімке збільшення обсягів цифрових даних зумовлюють необхідність забезпечення надійного захисту конфіденційної інформації. У високошвидкісних мережах програмна реалізація криптографічних алгоритмів дедалі частіше стає критичним вузьким місцем системи через значне навантаження на центральний процесор, що призводить до деградації загальної продуктивності обчислювального вузла. Традиційні обчислювальні системи на базі центральних процесорів не завжди здатні забезпечити необхідну пропускну здатність для магістральних каналів зв'язку в режимі реального часу.

Метою дипломної роботи є проєктування та реалізація криптографічного прискорювача на базі ПЛІС Cyclone V для захищеного передавання даних за допомогою алгоритму AES-128.

Підпис здобувача

01.06.2026

Дата

# ЗМІСТ

# ВСТУП

Сучасний розвиток інформаційно-комунікаційних технологій та стрімке збільшення обсягів цифрових даних зумовлюють необхідність забезпечення надійного захисту конфіденційної інформації. У високошвидкісних мережах програмна реалізація криптографічних алгоритмів дедалі частіше стає критичним вузьким місцем системи через значне навантаження на центральний процесор, що призводить до деградації загальної продуктивності обчислювального вузла. Традиційні обчислювальні системи на базі центральних процесорів не завжди здатні забезпечити необхідну пропускну здатність для магістральних каналів зв'язку в режимі реального часу.

Спеціалізовані апаратні прискорювачі на базі програмованих логічних інтегральних схем (FPGA) дозволяють вирішити цю проблему завдяки апаратному паралелізму, глибокій конвеєризації та можливості адаптації архітектури під потреби конкретного алгоритму. ПЛІС сімейства Cyclone V пропонують оптимальний баланс між вартістю ресурсів та продуктивністю, забезпечуючи високий рівень ізоляції критичних процесів та стійкість до атак за побічними каналами завдяки детермінованості часових характеристик. Це зумовлює актуальність розробки криптографічного прискорювача на базі ПЛІС Cyclone V для забезпечення захищеного передавання даних.

Метою дипломної роботи є проєктування та реалізація криптографічного прискорювача на базі ПЛІС Cyclone V для захищеного передавання даних за допомогою алгоритму AES-128.

Об’єктом дослідження є процес апаратного прискорення криптографічних перетворень для захисту інформації у високошвидкісних каналах передавання даних.

Предметом дослідження є методи та засоби проєктування ієрархічної модульної структури алгоритму AES-128 мовою проєктування апаратури

VHDL на базі архітектури Cyclone V з використанням засобів логічного синтезу та часового аналізу в середовищі Quartus II.

В межах кваліфікаційної роботи планується розробка та дослідження апаратного прискорювача для алгоритму симетричного шифрування AES-128. Реалізація пристрою передбачається на базі ПЛІС сімейства Cyclone V, що дозволить використовувати переваги апаратного паралелізму для досягнення значно вищої пропускної здатності порівняно з традиційними програмними методами. Основним архітектурним рішенням пропонується використання підходу повного розгортання, де всі десять раундів алгоритму реалізуються як послідовний комбінаційний ланцюжок, що забезпечує обробку 128-бітного блоку даних за один тактовий цикл

Практична цінність роботи полягає у створенні апаратного ядра шифрування, що дозволить збільшити пропускну здатність та швидкість обробки інформації у каналах зв'язку

# 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ДОСЛІДЖЕННЯ КРИПТОГРАФІЧНИХ АЛГОРИТМІВ

## 1.1 Огляд криптографічних алгоритмів

Сучасний розвиток інформаційно-комунікаційних технологій, стрімке збільшення обсягів цифрових даних та поширення мережевих сервісів зумовлюють необхідність забезпечення надійного захисту інформації. У глобальних мережах передаються конфіденційні персональні дані, фінансова інформація, державні ресурси та критично важливі дані промислових систем, тому питання інформаційної безпеки набуває особливої актуальності. Одним із базових механізмів забезпечення безпеки є криптографія, тобто наука про методи захисту інформації шляхом її математичного перетворення.

Криптографічні методи дозволяють забезпечити конфіденційність, цілісність, автентичність та невідмовність інформації. Конфіденційність гарантує недоступність даних для сторонніх осіб, цілісність забезпечує неможливість непомітної зміни інформації, автентичність підтверджує джерело повідомлення, а невідмовність унеможливлює заперечення факту передачі чи створення даних. Для реалізації цих властивостей використовуються різні класи криптографічних алгоритмів, серед яких основними є симетричні алгоритми шифрування, асиметричні алгоритми та криптографічні хеш-функції.

Симетричні алгоритми шифрування є найбільш поширеним засобом захисту великих обсягів даних. Принцип їх роботи полягає у використанні одного секретного ключа як для шифрування, так і для дешифрування інформації. Безпека системи повністю залежить від захищеності цього ключа, тому його передача між учасниками зв’язку повинна здійснюватися через безпечний канал або за допомогою додаткових криптографічних механізмів.

Основною перевагою симетричних алгоритмів є висока швидкість роботи та відносно невелика обчислювальна складність. Це дозволяє використовувати їх у високопродуктивних мережевих системах, серверах, VPN-з’єднаннях,

бездротових мережах, системах захисту баз даних та вбудованих IoT-пристроях. Саме завдяки високій продуктивності симетричні алгоритми найчастіше реалізуються апаратно на базі FPGA або ASIC-прискорювачів.

Симетричні алгоритми поділяються на потокові та блокові. Потокові шифри виконують шифрування послідовності бітів або байтів у реальному часі. Вони характеризуються низькою затримкою та високою швидкістю, що робить їх ефективними для потокового аудіо- та відеозв'язку. Прикладом потокового шифру є RC4, який раніше широко використовувався у протоколах бездротового зв'язку, однак згодом був визнаний недостатньо стійким через виявлені криптографічні вразливості.

Блокові шифри працюють із фіксованими блоками даних певного розміру. Найбільш відомим сучасним блоковим алгоритмом є, який став міжнародним стандартом симетричного шифрування. AES підтримує довжину ключа 128, 192 або 256 біт і виконує серію раундів математичних перетворень над блоками даних розміром 128 біт. До основних операцій алгоритму належать підстановка байтів, перестановка рядків, змішування стовпців та додавання раундового ключа.

Можна виділити такі переваги AES:

– висока криптостійкість;

– ефективність програмної та апаратної реалізації;

– підтримка сучасними процесорами;

– можливість глибокої конвеєризації при реалізації на FPGA.

Завдяки цим властивостям AES використовується у протоколах TLS, IPSec, WPA2/WPA3, системах шифрування дисків та хмарних сервісах.

Для блокових шифрів важливу роль відіграють режими роботи. Найпростішим режимом є ECB, однак він не забезпечує належного рівня безпеки через незалежне шифрування блоків. У сучасних системах частіше використовуються режими CBC (Cipher Block Chaining), CTR (Counter Mode) та

GCM (Galois/Counter Mode). Режим GCM особливо популярний, оскільки одночасно забезпечує як конфіденційність, так і перевірку цілісності даних.

Асиметричні криптографічні алгоритми використовують пару математично пов’язаних ключів – відкритий та закритий. Відкритий ключ може вільно розповсюджуватися серед користувачів, тоді як закритий ключ повинен залишатися конфіденційним. Такий підхід дозволяє вирішити проблему безпечного розподілу ключів, яка є одним із головних недоліків симетричної криптографії.

Найбільш відомим асиметричним алгоритмом є RSA, безпека якого базується на складності факторизації великих цілих чисел. Під час генерації ключів створюються два великі прості числа, на основі яких обчислюються модулі та експоненти для відкритого і закритого ключів. Для забезпечення високого рівня безпеки сучасні реалізації RSA використовують ключі довжиною 2048 або 4096 біт.

Іншим важливим напрямом є криптографія на основі еліптичних кривих Вона забезпечує аналогічний рівень безпеки при значно меншій довжині ключа. Наприклад, такі алгоритми із ключем 256 біт може забезпечувати рівень захисту, співставний із RSA-3072. Завдяки меншому обсягу обчислень алгоритми ECC широко використовуються у мобільних пристроях, смарт-картах та IoT-системах.

Асиметричні алгоритми застосовуються для:

– безпечного обміну ключами;

– створення електронного цифрового підпису;

– автентифікації користувачів;

– побудови інфраструктури відкритих ключів.

Недоліком асиметричних методів є значна обчислювальна складність. Операції модульного піднесення до степеня або множення точок еліптичної кривої потребують великої кількості ресурсів процесора та пам’яті. Через це асиметричне шифрування рідко використовується для безпосереднього

шифрування великих масивів даних. У практичних системах воно зазвичай застосовується лише на етапі встановлення захищеного з’єднання для передачі симетричного сеансового ключа.

Окремий клас криптографічних алгоритмів становлять хеш-функції. Їх основним призначенням є формування унікального цифрового відбитка повідомлення фіксованої довжини незалежно від розміру вхідних даних. Хеш-функції є незворотними, тобто відновити початкове повідомлення за результатом хешування практично неможливо.

Криптографічна хеш-функція повинна задовольняти такі вимоги:

– стійкість до пошуку прообразу;

– стійкість до пошуку другого прообразу;

– стійкість до колізій;

– висока швидкість обчислення.

Найбільш відомими сучасними хеш-алгоритмами є SHA-2 та SHA-3. Алгоритм SHA-256 широко використовується у цифрових сертифікатах, криптовалютах, блокчейн-технологіях та системах перевірки цілісності даних.

Хеш-функції застосовуються для:

– перевірки цілісності файлів;

– зберігання паролів;

– формування цифрових підписів;

– контролю автентичності повідомлень;

– технологій блокчейн.

У системах електронного цифрового підпису спочатку обчислюється хеш повідомлення, після чого він шифрується закритим ключем відправника. Це дозволяє значно зменшити обсяг обчислень та підвищити ефективність роботи системи.

До прикладу, на рис. 1.1 представлено шість ілюстрацій (у два ряди по три), які візуалізують принцип роботи криптографічних хеш-функцій, зокрема аваланшний ефект та дифузію даних.

Кожна панель показує тривимірну ґратчасту структуру (подібну до кубічної решітки або блоку даних), де: чорні точки це окремі елементи даних (біти або «частинки» інформації), сірі заштриховані області є активними блоками або ділянки, в яких відбувається обробка даних, а стрілки вказуються напрямки поширення впливу та перетворення даних. Кругові стрілки (вихори) символізують операції перемішування (mixing), які є ключовими для створення хаотичної залежності результату від вхідних даних.

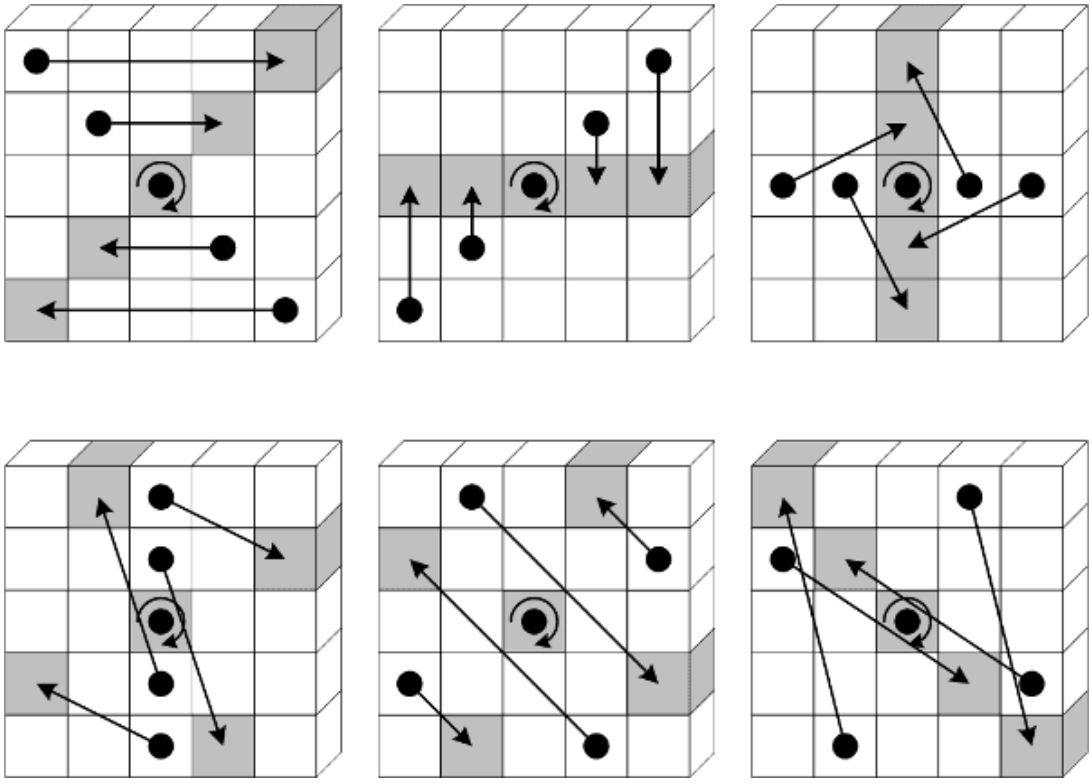


Рисунок 1.1 – Ілюстрація етапу криптографічного алгоритму Keccak, що який лежить в основі стандарту SHA-3 [1]

Всі панелі разом ілюструють, як у криптографічних хеш-функціях (наприклад, SHA-256) невелика зміна на вході (переміщення навіть однієї частинки) призводить до кардинально іншої картини на виході завдяки інтенсивному перемішуванню та лавиноподібному поширенню змін.

Таким чином, можна опису функціонування алгоритмів шифрування можна подати їх порівняння, що подано у таблиці 1.1.

Таблиця 1.1 – Порівняння криптографічних алгоритмів

| Критерій | Симетричні алгоритми | Асиметричні алгоритми | Хеш-функції |
|---|---|---|---|
| Принцип роботи | Один ключ для шифрування та дешифрування | Пара ключів: відкритий і закритий | Формування незворотного хешу |
| Основне призначення | Шифрування даних | Обмін ключами, цифровий підпис | Контроль цілісності даних |
| Швидкодія | Висока | Низька | Дуже висока |
| Обчислювальна складність | Низька | Висока | Середня |
| Розмір ключа | 128–256 біт | 2048–4096 біт (RSA) | Ключ відсутній або використовується в HMAC |
| Криптостійкість | Висока при правильному ключі | Висока завдяки складним математичним задачам | Залежить від стійкості до колізій |
| Переваги | Висока продуктивність, ефективність | Безпечний розподіл ключів | Незворотність, контроль цілісності |
| Недоліки | Проблема передачі ключа | Великі обчислювальні витрати | Неможливість відновлення даних |
| Приклади алгоритмів | AES, DES, ChaCha20 | RSA, ECC, ElGamal | SHA-256, SHA-3, MD5 |
| Основні сфери застосування | VPN, TLS, шифрування дисків | ЕЦП, PKI, SSL/TLS | Перевірка файлів, блокчейн, паролі |

## 1.2. Огляд відомих підходів до апаратної реалізації криптографічних алгоритмів

Розвиток сучасних систем захисту інформації демонструє чітку тенденцію до переходу від суто програмних рішень до спеціалізованих апаратних прискорювачів. Цей перехід зумовлений необхідністю забезпечення високої продуктивності та низьких затримок, що є недосяжним для універсальних процесорів. У науковій літературі виділяється кілька домінуючих підходів до проектування таких систем, кожен з яких пропонує свій компроміс між швидкістю обробки даних, використанням ресурсів кристала та рівнем захищеності.

Важливим напрямком досліджень є розробка процесорів із архітектурою команд, оптимізованою під конкретні алгоритми (ASIP). У роботі [2] автори детально досліджують проблему обчислювальних «вузьких місць» при реалізації стандарту SHA-3 на базі 32-бітних RISC-процесорів. Аналіз, проведений у цьому дослідженні, показує, що основною перешкодою для високої продуктивності є відсутність у стандартних наборах команд (ISA) засобів для ефективної обробки 64-бітних векторних ротацій та специфічних логічних операцій алгоритму Keccak. Для вирішення цієї проблеми автори пропонують концепцію розширення системи команд, яка реалізується двома шляхами: через модифікацію внутрішнього тракту даних та через впровадження зовнішнього співпроцесора. Порівняльний аналіз цих підходів у роботі [2] свідчить, що співпроцесорна модель забезпечує значно вищий приріст швидкості (до 61.4%), хоча і потребує більшої кількості логічних ресурсів ПЛІС. Це підтверджує тезу, відображену на схемі проектування, про критичну важливість етапу вибору архітектури між паралельною та конвеєрною обробкою на ранніх стадіях розробки.

Паралельно з розвитком ASIP-архітектур, значна увага приділяється повнофункціональним апаратним модулям для симетричного шифрування,

зокрема AES. У дослідженні [3] автори пропонують підходи до максимізації пропускної здатності за рахунок глибокої конвеєризації та розгортання ітераційних циклів (loop unrolling). Такий підхід дозволяє обробляти нові блоки даних на кожному такті синхросигналу, що є критичним для дата-центрів та магістральних каналів зв'язку. Проте, аналізуючи результати роботи, можна дійти висновку, що надмірне розгортання архітектури призводить до значного зростання енергоспоживання та площі пристрою, що ставить перед розробником завдання багатокритеріальної оптимізації використання ресурсів, як це передбачено в ключових особливостях реалізації на наведеній структурній схемі.

Окремий масив наукових праць, зокрема робота [4], присвячений реалізації алгоритмів асиметричної криптографії на базі еліптичних кривих та легковагової криптографії для пристроїв інтернету речей (IoT). Автори роботи підкреслюють, що для таких систем пріоритетом є не стільки абсолютна швидкість, скільки енергоефективність та мінімальний апаратний відбиток (hardware footprint). У цьому контексті досліджуються методи спільного використання ресурсів (resource sharing), де один і той самий арифметичний блок виконує різні етапи алгоритму в різні моменти часу. Це корелює з етапом аналізу вимог, оскільки обмеження по живленню в автономних пристроях вимагають відмови від максимально паралельних структур на користь компактних послідовних модулів.

Не менш важливим аспектом, що розглядається в роботах [2-4], є стійкість апаратних реалізацій до атак за побічними каналами (SCA). Оскільки фізичне втілення алгоритму на ПЛІС створює унікальні патерни енергоспоживання, автори пропонують інтегрувати механізми захисту безпосередньо в логічну структуру модуля. У роботі [3] детально аналізуються методи маскування (masking) та випадковізації (shuffling), які дозволяють де-корелювати оброблювані дані від фізичних сигналів пристрою. Аналіз цих підходів показує, що впровадження засобів захисту від апаратних атак зазвичай

призводить до зниження продуктивності на 20-30% та збільшення площі кристала, що підтверджує складність етапу проектування апаратного модуля, де необхідно балансувати між безпекою та швидкодією.

На основі аналізу розглянутих робіт можна визначити, що сучасний стан галузі характеризується переходом від універсальних прискорювачів до адаптивних систем, які поєднують гнучкість програмованої логіки ПЛІС із високою спеціалізацією обчислювальних блоків. Це дозволяє не лише досягти цільових показників швидкості передавання даних, але й забезпечити масштабованість архітектури для підтримки нових стандартів шифрування та методів протидії кіберзагрозам.

## 1.3. Архітектура ПЛІС Cyclone V

Сучасні програмовані логічні інтегральні схеми є ефективною платформою для реалізації високопродуктивних криптографічних систем, оскільки забезпечують можливість апаратного паралелізму, гнучкої реконфігурації та високої швидкодії. Однією з поширених FPGA платформ є сімейство Cyclone V компанії Intel Corporation, яке поєднує достатній рівень обчислювальних ресурсів із помірним енергоспоживанням та доступною вартістю (рис. 1.2). Архітектура Cyclone V орієнтована на застосування у вбудованих системах, цифровій обробці сигналів, телекомунікаційних пристроях та криптографічних прискорювачах.

Архітектурна концепція сімейства програмованих логічних інтегральних схем Cyclone V від компанії Intel ґрунтується на використанні технологічного процесу двадцятивосьми-нанометрової літографії, що дозволяє досягти безпрецедентного рівня інтеграції за умови збереження низького рівня тепловиділення. Основу внутрішнього устрою даної платформи складає масив адаптивних логічних модулів, які виступають базовими цеглинками для побудови обчислювальних структур будь-якої складності.

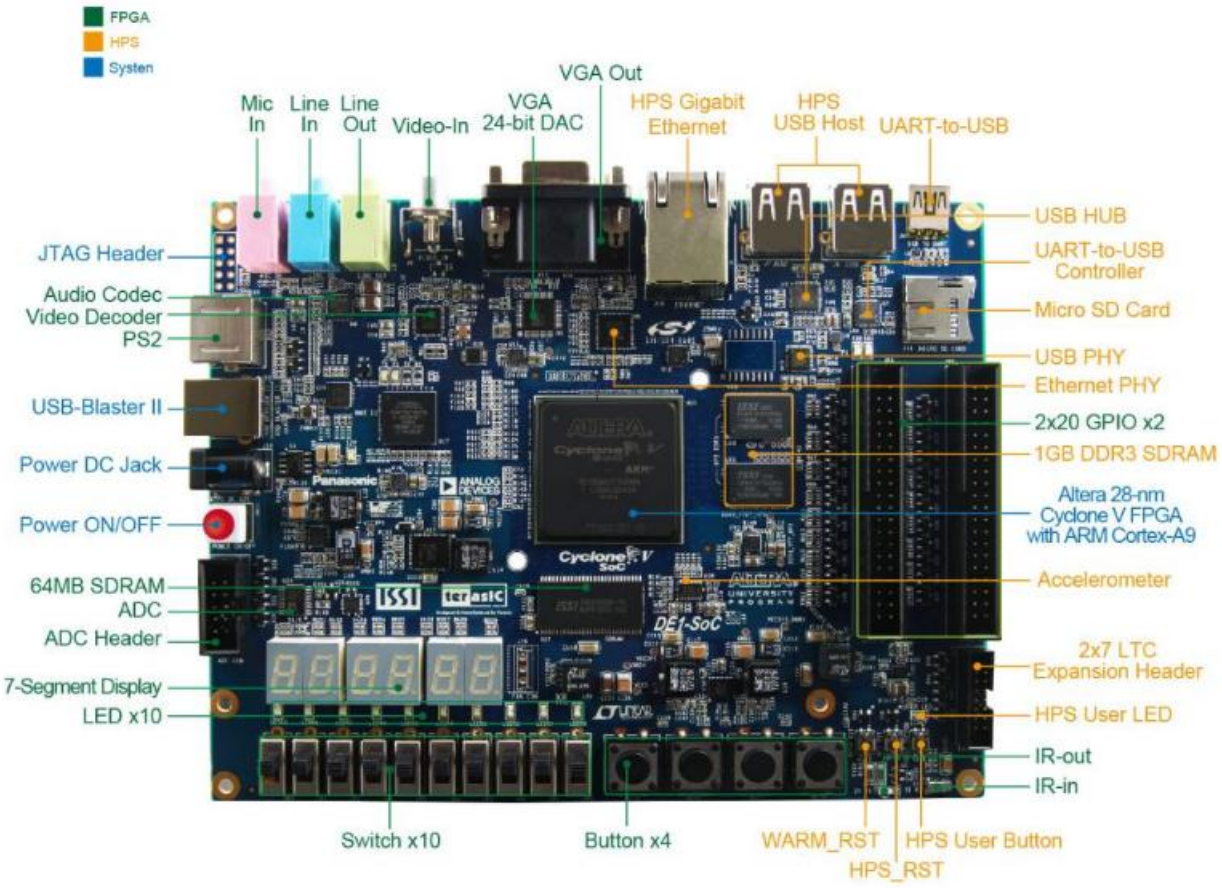



Рисунок 1.2 – ПЛІС Altera Cyclone V

Кожен адаптивний логічний модуль (рис. 1.3) у складі Cyclone V володіє унікальною здатністю до конфігурування завдяки наявності восьмивходових таблиць відповідності, які можуть бути розділені для виконання декількох логічних операцій одночасно. Така гнучкість архітектури дозволяє розробнику криптографічного прискорювача максимально ефективно реалізовувати логіку комбінаційних перетворень, що є критично важливим для розгортання раундів сучасних алгоритмів шифрування, де кожна наносекунда затримки впливає на загальну пропускну здатність системи.

Окремим високоефективним сегментом ресурсів ПЛІС виступають спеціалізовані блоки цифрової обробки сигналів, які в архітектурі Cyclone V реалізовані за принципом змінної точності. Дані блоки не просто виконують роль звичайних множників, а являють собою складні математичні вузли, здатні

здійснювати операції множення з накопиченням, що є фундаментом для реалізації криптографії на еліптичних кривих або складних систем асиметричного шифрування.

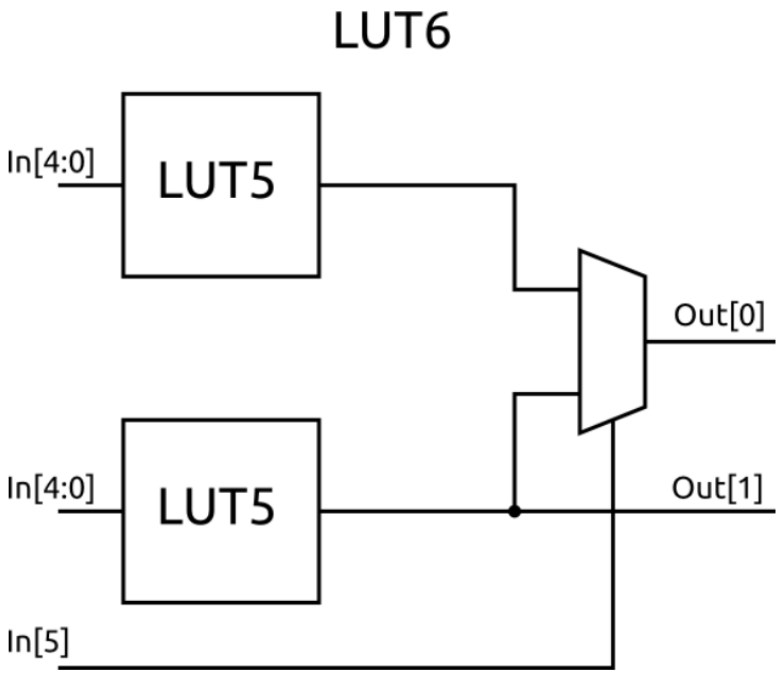



Рисунок 1.3 – Адаптивний логічний модуль

Використання блоків DSP дозволяє значно розвантажити загальні логічні ресурси кристала, оскільки вони оптимізовані на апаратному рівні для виконання високоінтенсивних арифметичних обчислень. Конструкція цих блоків передбачає наявність внутрішніх регістрів для конвеєризації обробки даних, що забезпечує стабільно високу тактову частоту пристрою навіть при виконанні багаторозрядних математичних операцій, необхідних для генерації та перевірки цифрових підписів у захищених каналах зв’язку.

Система збереження та швидкого доступу до даних у структурі Cyclone V представлена спеціалізованими блоками пам’яті типу M10K, кожен з яких забезпечує зберігання десяти кілобіт інформації. Ці блоки інтегровані безпосередньо в логічну матрицю, що гарантує мінімальні затримки при зверненні до пам’яті у порівнянні з використанням зовнішніх накопичувачів. В контексті розробки криптографічного прискорювача блоки M10K відіграють

вирішальну роль, оскільки вони добре підходять для формування таблиць заміни, відомих як S-блоки, що є невід'ємною частиною стандартів симетричного шифрування. Можливість функціонування пам'яті у двопортовому режимі дозволяє організовувати одночасні процеси зчитування та запису, що критично важливо для буферизації потокових даних при високошвидкісному передаванні інформації через мережеві інтерфейси. Окрім того, блоки M10K підтримують механізми корекції помилок, що підвищує загальну надійність криптографічного пристрою в умовах можливих електромагнітних завад.

Важливим аспектом архітектури є розвинена ієрархія внутрішніх з'єднань, яка забезпечує швидку маршрутизацію сигналів між логічними модулями, блоками DSP та пам'яттю. Висока щільність трасувальних каналів дозволяє мінімізувати явище джиттера та забезпечити точне дотримання часових параметрів схеми, що є обов'язковою умовою для стабільної роботи прискорювача на високих частотах. Структура Cyclone V також включає спеціалізовані модулі керування тактовими сигналами, які дозволяють створювати декілька незалежних частотних доменів для різних вузлів системи. Це дає змогу оптимізувати енергоспоживання пристрою, запускаючи інтерфейсні блоки на одній частоті, а обчислювальне ядро шифрування на максимально можливій швидкості.

Завдяки тому, що конфігурація ПЛІС базується на статичній пам'яті з довільним доступом, архітектура Cyclone V забезпечує повну свободу в модифікації апаратних засобів захисту інформації. Розробник має можливість переконфігурувати внутрішню логіку пристрою необмежену кількість разів, адаптуючи прискорювач до нових типів кіберзагроз або оновлених стандартів шифрування без необхідності заміни самого фізичного компонента. Такий підхід поєднує в собі продуктивність спеціалізованих інтегральних схем із гнучкістю програмних рішень, що робить дану платформу ідеальним вибором для побудови систем захищеного передавання даних. Використання апаратних

ресурсів Cyclone V дозволяє реалізувати паралельну обробку декількох потоків даних, що значно перевершує можливості традиційних процесорів загального призначення та забезпечує високий рівень безпеки завдяки фізичному розділенню процесів обробки ключів та передавання зашифрованої інформації.

При детальному розгляді архітектури Cyclone V важливо провести паралель із рішеннями основного конкурента на ринку програмованої логіки, а саме із сімействами Artix та Kintex від компанії Xilinx (рис. 1.4).

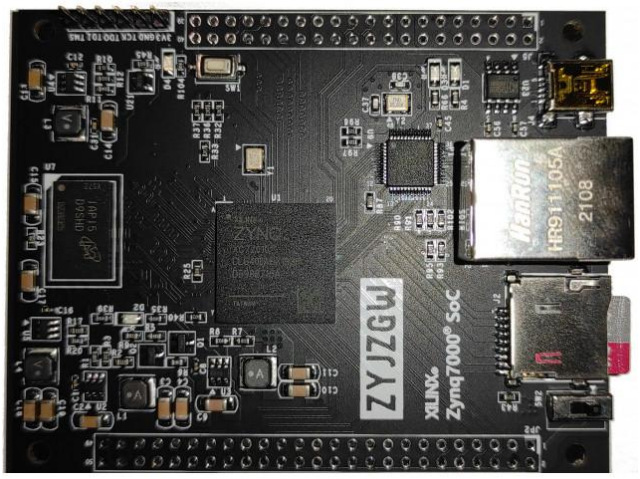



Рисунок 1.4 – ПЛІС Xilinx Zynq

Основна відмінність між цими двома архітектурними підходами полягає в структурі базового логічного елемента. У той час як Intel використовує в Cyclone V адаптивні логічні модулі, Xilinx базується на конфігурованих логічних блоках, кожен з яких містить два слайси з чотирма шестивходовими таблицями відповідності. Адаптивна структура Intel вважається більш гнучкою для реалізації функцій з великою кількістю входів, що може бути корисним для складних комбінаційних схем криптографічних перетворень. Проте архітектура Xilinx характеризується надзвичайно високою ефективністю при виконанні однотипних регулярних операцій, що також знаходить своє застосування в задачах обробки даних.

Суттєві відмінності спостерігаються і в організації вбудованої пам’яті, де замість модулів M10K компанія Xilinx пропонує блоки пам’яті об’ємом тридцять шість кілобіт, які можуть бути розділені на два незалежні блоки по вісімнадцять кілобіт. Більший об’єм одиничного блоку пам’яті в архітектурі Xilinx дозволяє ефективніше зберігати великі масиви даних, проте дрібніша структура M10K у Cyclone V забезпечує кращу гранулярність при побудові множини дрібних таблиць підстановок, що часто зустрічаються в алгоритмах симетричного шифрування. Крім того, підхід Xilinx передбачає можливість використання частини логічних ресурсів як розподіленої оперативної пам’яті, що дає розробнику додаткові інструменти для оптимізації затримок, хоча це і призводить до швидшого вичерпання загальних логічних ресурсів кристала.

Порівняння блоків цифрової обробки сигналів також демонструє різні філософії проектування, оскільки блоки DSP48 у рішеннях Xilinx мають фіксовану архітектуру з високим рівнем оптимізації під конкретні операції множення та додавання. Натомість блоки DSP змінної точності в Cyclone V пропонують вищу адаптивність, дозволяючи розробнику самостійно визначати розрядність обчислень залежно від потреб криптографічного алгоритму. Це дозволяє більш економно витрачати апаратний ресурс при роботі з нестандартними форматами даних, що часто виникають у сфері захисту інформації. Незважаючи на ці розбіжності, обидві архітектури забезпечують необхідний інструментарій для створення високопродуктивних обчислювальних систем, а вибір на користь Cyclone V у даній роботі обумовлений оптимальним поєднанням вартості логічних ресурсів та специфіки їхньої внутрішньої організації для реалізації

## 1.4 Порівняльний аналіз апаратних та програмних платформ для реалізації криптографічних алгоритмів

Вибір оптимальної платформи для розгортання криптографічних засобів захисту інформації є одним із головних завдань, яке потребує детального порівняння архітектурних особливостей процесорів загального призначення, спеціалізованих інтегральних схем та програмованої логіки. Традиційно найпоширенішим способом реалізації алгоритмів шифрування залишається використання центральних процесорів, оскільки програмний підхід забезпечує найвищий рівень доступності та простоту розробки. Однак архітектура класичних процесорів базується на послідовному виконанні команд згідно з парадигмою фон Неймана, що створює значне навантаження на обчислювальні ресурси при виконанні специфічних бітових операцій. Криптографічні перетворення зазвичай вимагають великої кількості маніпуляцій на рівні окремих бітів, циклічних зсувів та підстановок, які у програмному середовищі змушені проходити через довгий конвеєр вибірки інструкцій, декодування та звернення до пам’яті. Це призводить до виникнення суттєвих затримок та обмеження пропускної здатності, що робить центральні процесори малоефективними для обробки мережевого трафіку у реальному часі на гігабітних швидкостях.

На протилежному боці технологічного спектру знаходяться спеціалізовані інтегральні схеми, відомі як ASIC, які проектуються під конкретний алгоритм без можливості подальшої зміни логіки. Такі пристрої демонструють найвищу можливу продуктивність та мінімальне енергоспоживання на одиницю обчислень, оскільки кожен транзистор у їхньому складі оптимізований для виконання конкретної функції. Проте використання ASIC супроводжується надзвичайно високою вартістю розробки та тривалим циклом виробництва, що робить їх економічно виправданими лише при масовому тиражуванні. Головним недоліком жорстко заданої логіки є повна відсутність адаптивності, адже при виявленні вразливостей в алгоритмі або при появі нових криптографічних стандартів такий пристрій неможливо модернізувати, що потребує повної заміни апаратної бази. В умовах стрімкого

розвитку методів кібератак та переходу до постквантової криптографії така негнучкість стає критичним фактором ризику для довгострокових проектів.

Програмовані логічні інтегральні схеми типу FPGA займають проміжну нішу, поєднуючи в собі переваги апаратної швидкодії та програмної гнучкості. На відміну від процесорів, де алгоритм адаптується під фіксовану архітектуру, у ПЛІС сама архітектура адаптується під потреби конкретного алгоритму. Це дозволяє створювати глибоко конвеєризовані структури, де кожен раунд шифрування виконується на окремому апаратному рівні, що забезпечує обробку даних за кожен такт синхросигналу. Можливість паралельного виконання операцій є ключовою перевагою ПЛІС, оскільки розробник може розгорнути десятки ідентичних обчислювальних ядер на одному кристалі, досягаючи продуктивності, яка на кілька порядків перевищує можливості сучасних багатоядерних процесорів. При цьому ПЛІС зберігають можливість повної переконфігурації логічної матриці, що дозволяє дистанційно оновлювати криптографічні протоколи та впроваджувати додаткові механізми захисту без демонтажу обладнання.

Порівнюючи ці платформи з точки зору безпеки, важливо зазначити, що ПЛІС забезпечують вищий рівень ізоляції критичних процесів. У процесорних системах існує значна кількість вразливостей, пов’язаних із витоком даних через спільний кеш або помилки передбачення розгалужень, тоді як апаратна реалізація в ПЛІС дозволяє фізично розділити потоки даних та керування. Хоча ASIC залишаються лідерами в сегменті енергоефективності для стабільних протоколів, ПЛІС пропонують оптимальний компроміс між витратами на розробку та швидкістю виходу готового рішення на ринок. Можливість швидкого прототипування та тестування різних архітектурних рішень безпосередньо на залізі робить ПЛІС ідеальним вибором для створення криптографічних прискорювачів, які повинні поєднувати високу надійність із здатністю протистояти новітнім загрозам у динамічному середовищі захищеного передавання даних.

Загальний аналіз свідчить про те, що для завдань захисту інформації в сучасних мережах саме ПЛІС архітектура виглядає найбільш збалансованою завдяки своїй здатності виконувати масивні паралельні обчислення при збереженні детермінованості часових характеристик. Це особливо важливо для запобігання атакам по сторонніх каналах, де аналіз часу виконання операцій може дати зловмиснику інформацію про секретний ключ. У ПЛІС кожен цикл обробки даних є суворо визначеним, що мінімізує ризики таймінгових атак. Таким чином, вибір FPGA як платформи для криптографічного прискорювача дозволяє реалізувати високоефективну систему, яка за продуктивністю наближається до спеціалізованих мікросхем, але при цьому залишається такою ж гнучкою в експлуатації, як і традиційне програмне забезпечення.

## 1.6 Постановка задачі

За результатами проведеного аналізу предметної області, а також огляду існуючих підходів до апаратної реалізації криптографічних алгоритмів та особливостей архітектури ПЛІС можна сформувати конкретне технічне завдання. Відповідно до мети роботи, якою є проєктування, реалізація та верифікація криптографічного прискорювача на базі ПЛІС Cyclone V для забезпечення захищеного передавання даних за допомогою алгоритму AES-128 можна визначити наступний перелік завдань, які потірбно вирішити:

1. Провести аналіз предметної області, розглянути основні класи криптографічних алгоритмів (симетричні, асиметричні, хеш-функції) та визначити переваги апаратної реалізації шифрування порівняно з програмною.

2. Дослідити архітектуру цільової платформи ПЛІС Cyclone V (зокрема модель 5CSEMA5F31C6), проаналізувати її логічні ресурси (ALM, блоки пам'яті M10K, блоки DSP) та можливості для реалізації криптографічних функцій.

3. Сформулювати повний перелік вимог до криптографічного прискорювача, включаючи функціональні (відповідність стандарту FIPS 197), апаратні (тактова частота, ресурси) та архітектурні вимоги.

4. Розробити структуру системи захищеного передавання даних, визначивши ролі персональних комп'ютерів відправника/отримувача та місце FPGA-вузлів у каналі зв'язку.

5. Обґрунтувати вибір архітектурного підходу, зокрема архітектури повного розгортання, для мінімізації латентності та досягнення максимальної пропускної здатності прискорювача.

6. Спроєктувати ієрархічну модульну структуру прискорювача AES-128 мовою VHDL, реалізувавши окремі компоненти для розгортання ключів, виконання раундів, підстановки байтів та змішування стовпців.

7. Реалізувати розроблену архітектуру в середовищі Quartus II, налаштувати проєкт для цільової мікросхеми та виконати логічний синтез схеми.

8. Налаштувати середовище для симуляції ModelSim-Altera та розробити тестовий сценарій (тестбенч) для верифікації пристрою за допомогою офіційних еталонних векторів стандарту NIST FIPS 197.

9. Провести аналіз результатів (виконати RTL-симуляцію, перевірити коректність шифрування на різних наборах даних (включаючи граничні випадки) та проаналізувати структуру отриманої схеми за допомогою RTL Viewer).

10. Оцінити продуктивність розробленого прискорювача, визначивши максимальну тактову частоту та пропускну здатність, а також провести порівняльний аналіз із програмними реалізаціями на базі центральних процесорів.

# 2 ПРОЄКТУВАННЯ КРИПТОГРАФІЧНОГО ПРИСКОРЮВАЧА НА БАЗІ ПЛІС CYCLONE V ДЛЯ ЗАХИЩЕНОГО ПЕРЕДАВАННЯ ДАНИХ У СЕРЕДОВИЩІ QUARTUS

## 2.1 Перелік вимог до криптографічного прискорювача на базі ПЛІС (FPGA) для захищеного передавання даних

Задача проєктування криптографічного прискорювача є комплексним завданням, що потребує чіткого формулювання вимог ще до початку будь-якої технічної реалізації. Вимоги визначають межі системи, критерії її коректності та метрики оцінювання результату. У даному розділі систематизовано повний перелік функціональних, апаратних і технічних вимог до прискорювача, що реалізується на програмованій логічній інтегральній схемі Cyclone V у середовищі Quartus II.

Функціональні вимоги визначають що саме повинна виконувати система. Першою і головною вимогою є реалізація алгоритму симетричного блочного шифрування AES-128 відповідно до стандарту FIPS 197, опублікованого Національним інститутом стандартів і технологій США (NIST) у 2001 році [37]. Стандарт однозначно визначає всі операції алгоритму, набори тестових векторів та правила перевірки коректності реалізації. Відповідність стандарту є обов’язковою умовою для будь-якої реалізації AES незалежно від платформи.

Другою функціональною вимогою є підтримка режиму шифрування для 128-бітних блоків даних із 128-бітним ключем. Розмір блоку та ключа є фіксованими параметрами обраного варіанту AES-128 і не підлягають зміні.

Третя функціональна вимога стосується повноти реалізації алгоритму. Прискорювач повинен виконувати всі десять раундів шифрування включно з початковим AddRoundKey та фінальним раундом без операції MixColumns.

Четверта вимога полягає у коректному розгортанні ключів, тобто з одного 128-бітного початкового ключа система повинна генерувати одинадцять раундових ключів відповідно до алгоритму Key Schedule стандарту AES.

Апаратні вимоги визначають обмеження та можливості цільової платформи. Цільовим пристроєм є ПЛІС Cyclone V із маркуванням 5CSEMA5F31C6 виробництва Altera (Intel). Ця мікросхема належить до сімейства Cyclone V і містить 85 480 адаптивних логічних модулів (ALM), 172 600 регістрів, 4 450 Кбіт вбудованої пам’яті M10K та 336 блоків цифрової обробки сигналів DSP. Вимога до апаратного інтерфейсу передбачає використання простого паралельного інтерфейсу з шинами даних шириною 128 біт, що відповідає розміру блоку AES. Інтерфейс повинен включати сигнали керування start та done для синхронізації обміну даними із зовнішніми пристроями. Вимога до тактування передбачає роботу схеми від єдиного зовнішнього тактового сигналу clk з можливістю досягнення тактової частоти не менше 100 МГц на цільовому пристрої. Вимога до споживання ресурсів передбачає розміщення всієї логіки в межах доступних ресурсів мікросхеми 5CSEMA5F31C6 без перевищення кількості наявних ALM.

Архітектурні вимоги визначають структуру та принципи побудови схеми. Вимога до архітектурного підходу передбачає реалізацію за принципом повного розгортання, при якому всі десять раундів алгоритму реалізовані як окремі апаратні блоки, з’єднані послідовно у комбінаційний ланцюжок. Така архітектура забезпечує мінімальну латентність. Результат шифрування фіксується у вихідному регістрі вже через один тактовий цикл після подачі сигналу start. Вимога до модульності передбачає ієрархічну організацію коду: кожна операція AES реалізована як окремий VHDL-модуль з чітко визначеними портами. Вимога до мови реалізації передбачає використання VHDL відповідно до стандарту VHDL-93, що забезпечує сумісність із Quartus II версії 15.0. Вимога до детермінованості передбачає фіксовану та однакову латентність обробки для будь-яких вхідних даних.

Вимоги до верифікації визначають умови підтвердження коректності реалізації. Верифікація повинна проводитись методом функціональної RTL-симуляції у середовищі ModelSim-Altera. Обов’язковим є використання

офіційних тестових векторів стандарту NIST FIPS 197: вектор з Додатку B (відкритий текст 3243F6A8885A308D313198A2E0370734, ключ 2B7E151628AED2A6ABF7158809CF4F3C, очікуваний шифротекст 3925841D02DC09FBDC118597196A0B32) є мінімально необхідним для підтвердження коректності. Додатково повинні бути перевірені граничні значення: нульові вхідні дані та ключ, а також максимальні значення 0xFF для всіх байтів.

Вимоги до продуктивності визначають метрики оцінювання ефективності прискорювача. Основною метрикою є пропускна здатність, що розраховується як добуток максимальної тактової частоти Fmax на розрядність блоку 128 біт, поділений на кількість тактів на блок. Для архітектури на основі повного розгортання кількість тактів на блок дорівнює одиниці, тому пропускна здатність прямо пропорційна Fmax. Порівняння продуктивності повинно проводитись із програмними реалізаціями AES на процесорах загального призначення як без апаратного прискорення, так і з використанням інструкцій AES-NI.

## 2.2 Особливості апаратної реалізації криптографічних алгоритмів шифрування

У сучасних умовах стрімкого розвитку високошвидкісних мережевих технологій та зростання обсягів переданих даних програмна реалізація криптографічних методів захисту інформації дедалі частіше стає критичним вузьким місцем системи. Це зумовлено значним навантаженням на центральний процесор, що призводить до деградації загальної продуктивності обчислювального вузла. Як наслідок, актуальним стає перехід до використання спеціалізованих апаратних прискорювачів, розроблених на базі програмованих логічних інтегральних схем (FPGA) або спеціалізованих інтегральних мікросхем (ASIC). Структурно-логічна модель такої реалізації, що охоплює

переваги, архітектурні особливості та етапи проектування, представлена на рис. 2.1.

Згідно з наведеною схемою, ключові переваги використання програмованих логічних інтегральних схем та спеціалізованих інтегральних мікросхем полягають у здатності забезпечувати надвисоку продуктивність та мінімальні затримки при обробці трафіку. На відміну від універсальних процесорів, де операції виконуються послідовно, апаратна платформа дозволяє реалізувати глибоку паралельну обробку даних. Масштабованість архітектури дає змогу адаптувати обчислювальні ресурси під конкретні потреби системи, забезпечуючи при цьому високу енергоефективність, що є критичним параметром для вбудованих систем та центрів обробки даних. Важливою особливістю FPGA є можливість оновлення конфігурації алгоритму навіть після введення пристрою в експлуатацію, що поєднує гнучкість програмного забезпечення з потужністю апаратних засобів.

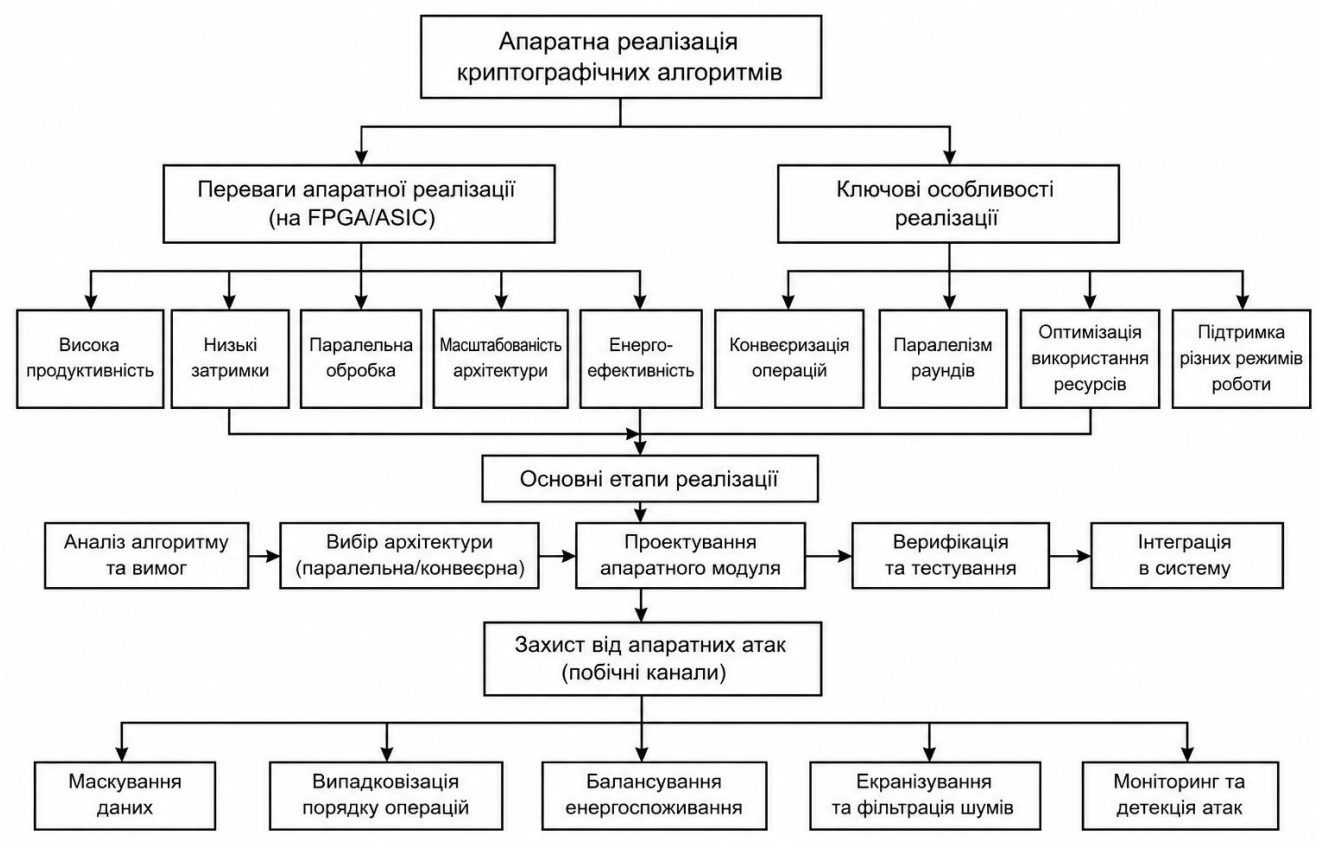


Рисунок 2.1 – Основні етапи та особливості апаратної реалізації криптографічних алгоритмів

Технічна складність та ефективність апаратної реалізації визначаються низкою ключових особливостей. Зокрема, застосування конвеєризації операцій дозволяє одночасно обробляти кілька блоків даних на різних стадіях алгоритму, що суттєво підвищує пропускну здатність. Паралелізм на рівні раундів шифрування, особливо характерний для алгоритмів сімейства AES, дозволяє досягати швидкостей у сотні Гбіт/с. При цьому особлива увага приділяється оптимізації використання ресурсів кристала, таких як логічні елементи та блоки пам'яті, а також забезпеченню підтримки різних режимів роботи алгоритму (наприклад, CBC, GCM, CTR) залежно від вимог до безпеки та цілісності даних.

Процес створення криптографічного прискорювача є ітераційним і включає основні етапи реалізації, чітко структуровані на схемі. Початковим кроком є глибокий аналіз алгоритму та технічних вимог, що передує вибору конкретної архітектури – паралельної або конвеєрної. Безпосереднє проектування апаратного модуля передбачає опис логіки мовами програмування апаратури (VHDL або Verilog), після чого слідує етап верифікації та тестування для підтвердження відповідності реалізованого алгоритму еталонним моделям. Завершується цикл інтеграцією розробленого модуля в загальну систему, де перевіряється його взаємодія з іншими компонентами та інтерфейсами передачі даних.

Окремим і важливим аспектом, є захист від апаратних атак, зокрема атак за побічними каналами. Оскільки фізичні параметри роботи мікросхеми, такі як коливання енергоспоживання або електромагнітне випромінювання, можуть непрямим чином розкрити секретний ключ, проектування прискорювача обов'язково має включати методи протидії. До них належать маскування даних шляхом додавання випадкового шуму до проміжних значень та випадковізація порядку виконання операцій, що ускладнює статистичний аналіз для

зловмисника. Крім того, застосовується балансування енергоспоживання для нівелювання кореляції між операціями та струмом, а також фізичне екранування і фільтрація шумів. Постійний моніторинг та вбудовані механізми детекції атак дозволяють системі оперативно реагувати на спроби несанкціонованого втручання, гарантуючи високий рівень безпеки в рамках дипломної роботи.

Таким чином, апаратна реалізація криптографічних алгоритмів на базі ПЛІС не лише вирішує проблему швидкодії у сучасних мережах, але й дозволяє створити цілісну, захищену та енергоефективну інфраструктуру для передачі конфіденційної інформації. Об’єднання паралельних архітектурних рішень із сучасними методами захисту від фізичних атак робить такі прискорювачі фундаментом для побудови надійних систем кібербезпеки.

## 2.3 Структура криптографічного прискорювача на базі ПЛІС (FPGA) для захищеного передавання даних

Пропонована структура криптографічного прискорювача на базі ПЛІС (FPGA) для захищеного передавання даних складається із чотирьох фізичних вузлів: двох персональних комп'ютерів та двох програмованих логічних інтегральних схем, що виконують функції апаратного шифрування та дешифрування відповідно. Структуру криптографічного прискорювача на базі ПЛІС (FPGA) для захищеного передавання даних наведено на рис. 2.2.

Першим вузлом є персональний комп'ютер відправника (ПК1), який є джерелом відкритих даних. ПК1 не виконує жодних криптографічних операцій самостійно, оскільки вся задача шифрування делегується апаратному прискорювачу. Комп'ютер передає відкритий текст блоками по 128 біт на ПЛІС1 через послідовний інтерфейс UART. Вибір UART обумовлений його універсальністю та наявністю відповідного апаратного блоку на навчальних платах з Cyclone V (зокрема DE1-SoC), де USB-UART перетворювач

вбудований безпосередньо на платі. ПК1 також відповідає за управління сеансом передавання визначає межі блоків даних, надсилає керуючі сигнали початку та завершення передавання.

Другим вузлом є ПЛІС1 із завантаженою конфігурацією шифратора AES-128, що є апаратним криптографічним прискорювачем на стороні відправника. ПЛІС1 приймає 128-бітний блок відкритого тексту та 128-бітний ключ від ПК1, виконує повний цикл шифрування AES-128 за один тактовий цикл і передає зашифрований блок у канал передавання. Апаратна реалізація забезпечує детерміновану затримку та пропускну здатність, що значно перевищує можливості програмної реалізації на ПК1. Секретний ключ може завантажуватись у ПЛІС1 окремим захищеним каналом або зберігатись у конфігураційній пам'яті ПЛІС.

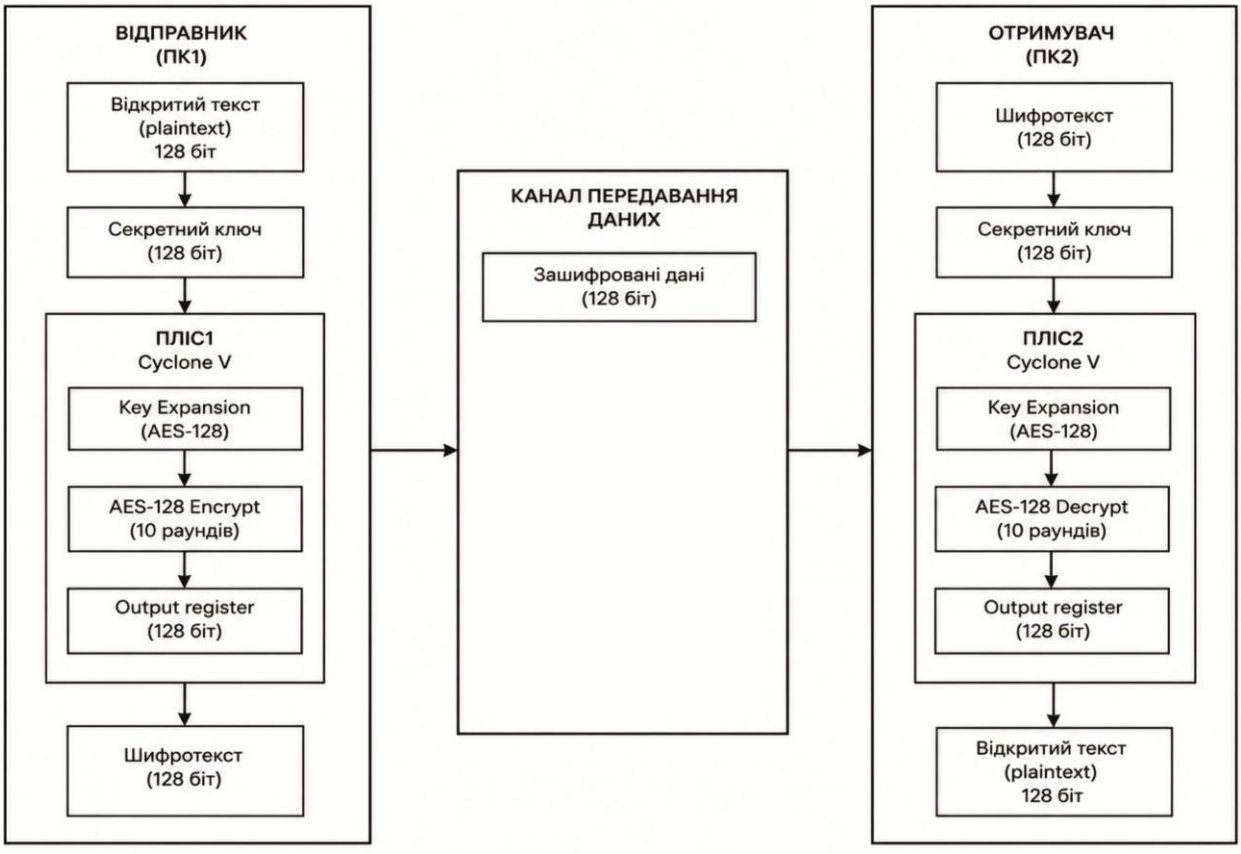



Рисунок 2.2 – Структура криптографічного прискорювача на базі ПЛІС (FPGA) для захищеного передавання даних

Третім вузлом є канал передавання даних. Він є середовищем транспортування зашифрованих блоків між ПЛІС1 та ПЛІС2. У лабораторному стенді канал може бути реалізований як пряме UART-з'єднання між двома платами за допомогою трьох провідників (TX, RX, GND). У реалістичній моделі канал реалізується через Ethernet: обидві плати з'єднуються через комутатор або безпосередньо, а зашифровані блоки інкапсулюються у UDP або TCP пакети. В обох випадках канал є незахищеним у тому сенсі що фізичний доступ до провідників або мережевого трафіку не дозволяє відновити вихідні дані без знання ключа. В такому разі зловмисник бачить виключно зашифровані блоки, стійкість яких забезпечується простором ключів AES розміром $2^{128}$.

Четвертим вузлом є ПЛІС2 із завантаженою конфігурацією дешифратора AES-128. Це є апаратним криптографічним прискорювачем на стороні отримувача. ПЛІС2 приймає зашифровані блоки з каналу, виконує зворотне перетворення AES-128 з тим самим ключем (симетричне шифрування) і передає відновлені блоки відкритого тексту на ПК2 через UART або SPI. Дешифратор AES будується за аналогічною ієрархічною структурою з використанням оберненого S-box (InvSubBytes), оберненого зсуву рядків (InvShiftRows), оберненого перетворення стовпців (InvMixColumns) та AddRoundKey з раундовими ключами у зворотному порядку.

П'ятий вузол – персональний комп'ютер отримувача (ПК2) приймає відновлений відкритий текст від ПЛІС2. З точки зору ПК2 процес є прозорим: він отримує ті самі дані що були відправлені з ПК1, не маючи жодної інформації про криптографічні перетворення що відбулись у проміжних вузлах.

У рамках даної бакалаврської роботи досліджується та симулюється центральний компонент описаної системи – блок шифрування AES-128 на ПЛІС Cyclone V (відповідає вузлу ПЛІС1). Верифікація проводитиметься методом RTL-симуляції у ModelSim без фізичного підключення до ПК або

каналу передавання, що є стандартною практикою на етапі проєктування цифрових схем. Реалізований та верифікований блок шифрування повністю придатний для подальшої інтеграції в описану систему шляхом додавання UART(або Ethernet) контролера як інтерфейсного рівня між ПК та криптографічним ядром AES.

2.4 Обгрунтування вибору архітектурного підходу повного розгортання для проєктування блоку шифрування

При проєктуванні апаратної реалізації алгоритму AES на FPGA можна виділити три основні архітектурні підходи, кожен з яких має різний баланс між продуктивністю та споживанням ресурсів. Вибір підходу є одним з ключових проєктних рішень, що визначає усі подальші характеристики системи.

Ітеративна архітектура передбачає реалізацію одного раундового блоку, через який дані проходять десять разів за десять тактових циклів. Такий підхід мінімізує споживання ресурсів FPGA, оскільки апаратна логіка одного раунду використовується повторно для кожного з десяти раундів. Проте латентність шифрування становить десять тактів, а пропускна здатність обмежена одним блоком на десять циклів. Для застосувань з обмеженими ресурсами або низькими вимогами до продуктивності цей підхід є оптимальним.

Конвеєрна архітектура  передбачає реалізацію десяти раундових блоків з регістрами між ними. Новий блок даних може подаватись на вхід кожного тактового циклу, оскільки різні блоки обробляються одночасно у різних раундових ступенях. Латентність складає десять тактів, але пропускна здатність досягає одного блоку за такт при повному завантаженні конвеєра. Споживання ресурсів є проміжним між ітеративною архітектурою та повного розгортання. Конвеєрна архітектура є оптимальною для потокової обробки великих обсягів даних.

Архітектура повного розгортання передбачає реалізацію всіх десяти раундових блоків як єдиної комбінаційної схеми без жодних проміжних

регістрів. Результат шифрування з’являється на виході комбінаційного ланцюжка через час, визначений затримкою критичного шляху, і фіксується у вихідний регістр протягом одного тактового циклу. Латентність становить рівно один такт, що є теоретичним мінімумом. Пропускна здатність досягає Fmax × 128 біт за такт. Недоліком є найвище споживання ресурсів серед усіх трьох підходів.

Для даної бакалаврської роботи обрано архітектуру повного розгортання з таких міркувань. По-перше, вона забезпечує максимальну пропускну здатність, що є ключовою метрикою для демонстрації переваг апаратного прискорення порівняно з програмною реалізацією. По-друге, вона найбільш повно демонструє паралелізм, притаманний FPGA. В такому випадку синтезатор розміщує всі десять раундів одночасно як незалежні апаратні блоки, з’єднані провідниками. По-третє, Cyclone V 5CSEMA5F31C6 має достатній обсяг ресурсів для розміщення повністю розгорнутої реалізації AES-128. Архітектура повного розгортання є також найпростішою для верифікації, оскільки не вимагає керування станом між тактами.

## 2.5 Реалізація алгоритму AES-128 та послідовність виконання операцій на ПЛІС

Алгоритм AES-128 реалізує блочне шифрування з фіксованим розміром блоку 128 біт та ключем довжиною 128 біт. Структурна схема алгоритму, наведена на рисунку 2.3. Ця схема відображає повний потік обробки даних від вхідного відкритого тексту до вихідного шифротексту і включає два паралельних тракти: тракт обробки даних та тракт розгортання ключів.

Тракт розгортання ключів є підготовчим і виконується одночасно з обробкою даних. На вхід блоку Key Expansion подається початковий ключ розміром 128 біт. Блок генерує одинадцять раундових ключів K0..K10, кожен розміром 128 біт, загальним обсягом 1408 біт. Ключ K0 використовується у

початковому перетворенні, ключі K1..K9 у раундах 1..9 відповідно, а ключ K10 – у фінальному раунді 10. В апаратній реалізації на ПЛІС весь блок Key Expansion є комбінаційною схемою, таким чином, що всі одинадцять ключів обчислюються та утримуються одночасно без тактування, що дозволяє подавати відповідний раундовий ключ безпосередньо на кожен із десяти паралельно реалізованих раундових блоків.

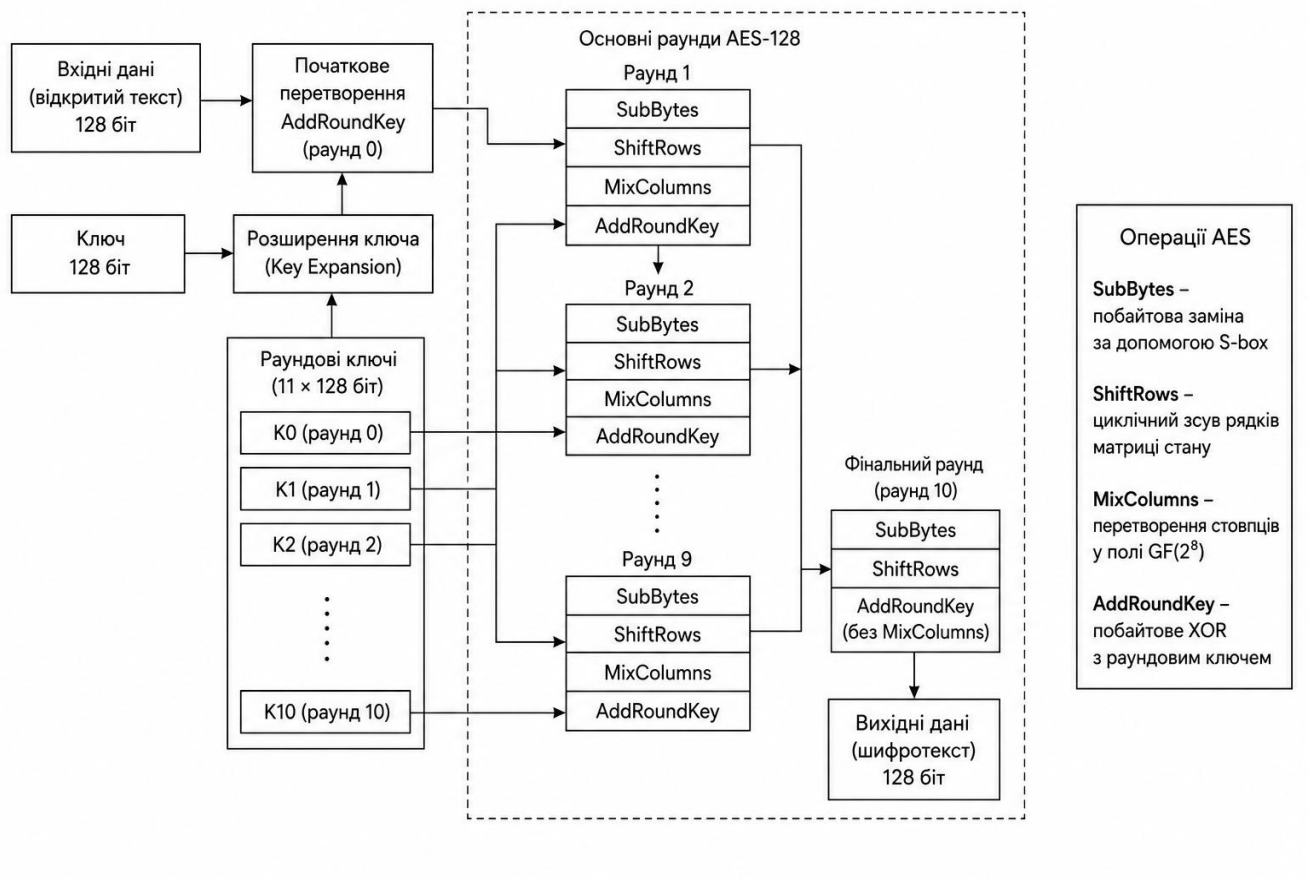



Рисунок 2.3 – Структурна схема реалізації алгоритму AES-128

Тракт обробки даних починається з початкового перетворення AddRoundKey (раунд 0). На цьому етапі 128-бітний вхідний відкритий текст піддається побітовій операції виключного АБО з нульовим раундовим ключем K0. Результат є матрицею стану розміром 4 на 4 байти, яка надходить на вхід першого основного раунду. Початковий AddRoundKey не є окремим раундом у повному розумінні, а є лише підготовчою операцією що вносить ключову

залежність ще до першого раунду і ускладнює атаки на початкові стани алгоритму.

Основні раунди з першого по дев'ятий мають ідентичну структуру і кожен складається з чотирьох послідовних операцій. Операція SubBytes виконується першою у кожному раунді: кожен з шістнадцяти байтів матриці стану незалежно замінюється відповідним байтом з таблиці нелінійної підстановки S-box. Ця операція забезпечує нелінійність алгоритму, що є необхідною умовою стійкості до лінійного та диференціального криптоаналізу. В апаратній реалізації кожен раунд містить шістнадцять паралельних інстанцій модуля sbox, що працюють одночасно над усіма байтами стану.

Операція ShiftRows виконується другою. Рядки матриці стану циклічно зсуваються вліво на різну кількість позицій. Нульовий рядок залишається без змін, перший зсувається на один байт, другий на два байти, третій на три байти. В результаті байти, що знаходились в одному стовпці, після зсуву опиняються у різних стовпцях. Це забезпечує міжстовпцеве перемішування і разом з MixColumns реалізує властивість повного лавинного ефекту. В апаратній реалізації на ПЛІС ShiftRows не споживає жодних логічних ресурсів, оскільки реалізується виключно переназначенням провідників між байтовими сигналами без будь-якої логічної схеми.

Операція MixColumns виконується третьою у раундах 1..9. Кожен із чотирьох стовпців матриці стану перетворюється незалежно шляхом множення на фіксовану матрицю у скінченному полі $GF(2^8)$. Операція забезпечує дифузію всередині стовпця: кожен вихідний байт стовпця залежить від усіх чотирьох вхідних байтів. В апаратній реалізації кожен з чотирьох стовпців обробляється окремою інстанцією модуля mix_single_col, де множення у полі $GF(2^8)$ зведено до операцій зсуву та виключного АБО без використання апаратних множників.

Операція AddRoundKey завершує кожен раунд. На цьому етапі матриця стану піддається побітовому XOR з відповідним раундовим ключем Ki.

Результат є вхідним станом наступного раунду. Після дев'ятого раунду стан передається на вхід фінального раунду.

Фінальний раунд 10 відрізняється від основних раундів відсутністю операції MixColumns. Виконуються лише три операції: SubBytes, ShiftRows та AddRoundKey з ключем K10. Відсутність MixColumns у фінальному раунді є вимогою стандарту FIPS 197 і не послаблює стійкості алгоритму, оскільки після останнього AddRoundKey MixColumns не вносить додаткової дифузії. Натомість відсутність MixColumns у фінальному раунді спрощує реалізацію дешифрування. Після виконання фінального раунду на виході з'являється шифротекст розміром 128 біт.

В апаратній реалізації за принципом повного розгортання всі десять раундових блоків фізично присутні одночасно у схемі ПЛІС і з'єднані послідовно, такимч чином, що вихідний стан кожного раунду є безпосереднім вхідним станом наступного без жодних проміжних регістрів. Блок Key Expansion постачає відповідний раундовий ключ безпосередньо до кожного з десяти раундів паралельно. Єдиний тактований елемент у всій схемі  це вихідний регістр після десятого раунду, який фіксує шифротекст за сигналом start. Таким чином, подана на рисунку 2.1 логічна послідовність операцій відображається в апаратній реалізації як просторово розгорнутий комбінаційний ланцюжок, де всі операції всіх десяти раундів виконуються фізично паралельно і безперервно.

## 2.6 Структура блоку шифрування AES-128 на ПЛІС Cyclone V

Ключовим елементом пропонованого криптографічного прискорювача на базі ПЛІС (FPGA) для захищеного передавання даних є блок шифрування AES-128 (див. структуру на рис. 2.1). Прискорювач пропонується реалізувати як ієрархічну систему з п’яти VHDL-модулів. Кожен модуль відповідає одній математичній операції або структурній одиниці алгоритму AES (рис. 2.2). Така

декомпозиція забезпечує незалежну верифікацію кожного рівня ієрархії, полегшує аналіз результатів синтезу та дозволяє повторно використовувати субблоки у різних конфігураціях.

Структуру блоку шифрування AES-128 на ПЛІС Cyclone V наведено на рис. 2.4.

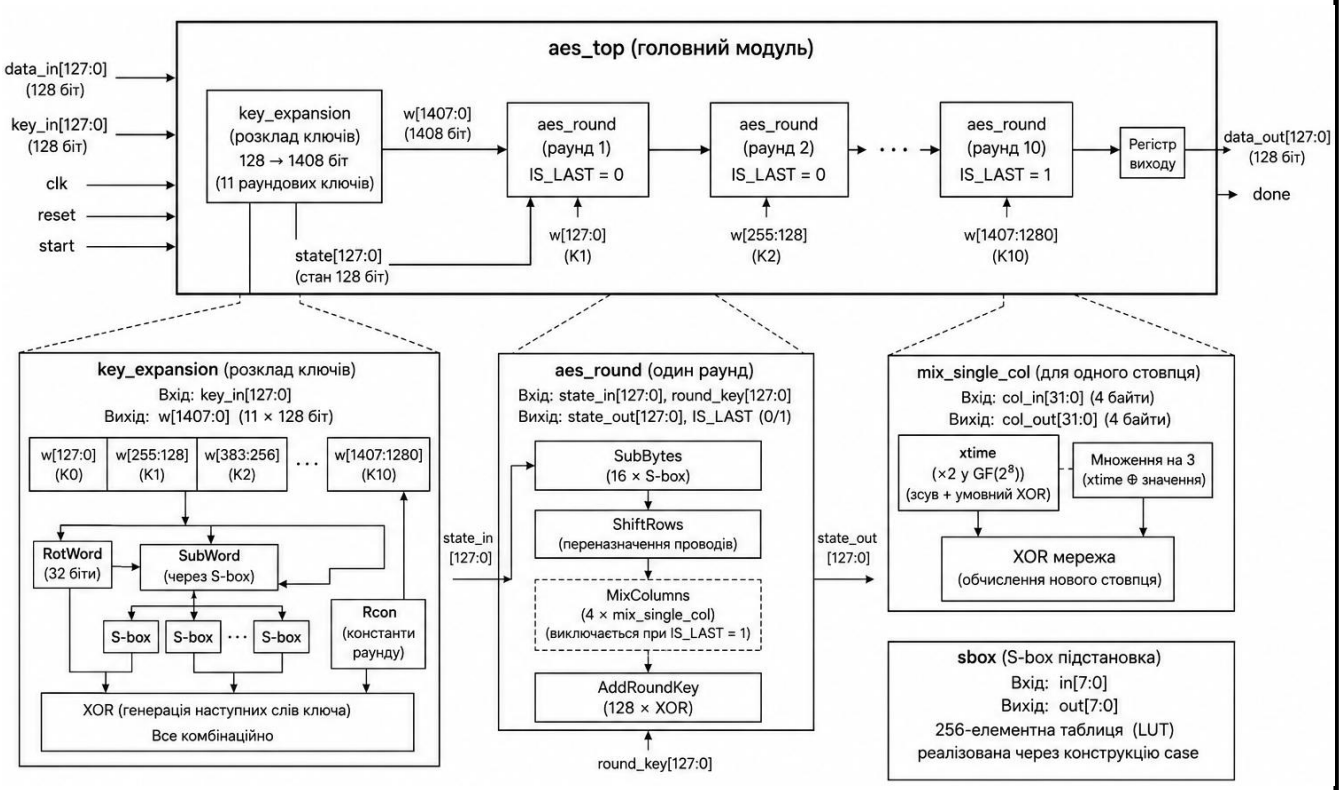



Рисунок 2.4 – Структура блоку шифрування AES-128 на ПЛІС Cyclone V

Верхньорівневий модуль aes_top є єдиною точкою входу в ієрархію і реалізує паралельний інтерфейс із зовнішнім середовищем. Він інкапсулює всі внутрішні з'єднання та приховує деталі реалізації від зовнішніх споживачів. Порти модуля: вхідний вектор даних data_in шириною 128 біт, вхідний ключ key_in шириною 128 біт, тактовий сигнал clk, сигнал скидання reset, сигнал запуску start, вихідний вектор data_out шириною 128 біт та сигнал готовності done. Всередині aes_top інстанціюється один блок key_expansion та десять блоків aes_round. Єдиний тактований елемент у всій ієрархії є вихідний регістр у aes_top, решта схеми є суто комбінаційною.

Модуль key_expansion реалізує алгоритм розгортання ключів Key Schedule стандарту AES-128. Вхідним є 128-бітний початковий ключ, а вихідним вектор шириною 1408 біт, що містить одинадцять 128-бітних раундових ключів. Внутрішня структура модуля базується на масиві з 44 слів по 32 біти кожне. Перші чотири слова є початковим ключем, кожні наступні чотири обчислюються через операції RotWord, SubWord та XOR з константами раунду Rcon. Для виконання операції SubWord інстанційовано 40 копій базового модуля sbox. Весь модуль є комбінаційним: раундові ключі обчислюються миттєво при зміні вхідного ключа.

Модуль aes_round реалізує один повний раунд алгоритму AES і є основним будівельним блоком прискорювача, що інстанціюється десять разів. Раунд складається з чотирьох послідовних перетворень матриці стану розміром 4 на 4 байти: SubBytes, ShiftRows, MixColumns та AddRoundKey. Модуль параметризований узагальненим параметром IS_LAST типу boolean: при значенні true операція MixColumns виключається з апаратної реалізації, що відповідає вимогам стандарту AES для десятого раунду. Вхідними є 128-бітний стан та 128-бітний раундовий ключ, вихідним – перетворений 128-бітний стан.

Операція SubBytes реалізується через 16 інстанцій базового модуля sbox по одній на кожен байт матриці стану. Операція ShiftRows не вимагає жодних апаратних ресурсів: вона реалізована як пряме переназначення сигналів, що виконується синтезатором без генерації будь-якої логіки. Операція MixColumns реалізована через 4 інстанції модуля mix_single_col – по одній на кожен стовпець матриці. Операція AddRoundKey реалізована як 128 паралельних операцій XOR між відповідними бітами стану та раундового ключа.

Модуль mix_single_col виконує операцію MixColumns для одного стовпця матриці стану, тобто чотирьох байтів. Математичною основою є матричне множення у скінченному полі $GF(2^8)$ з незвідним поліномом $x^8 + x^4 + x^3 + x + 1$. Матриця множення стандарту AES містить лише елементи 1, 2 та 3, що дозволяє замінити повноцінне множення у полі двома операціями: множенням

на 2 через функцію xtime (зсув та умовний XOR) та множенням на 3 як комбінацією xtime та XOR. Реалізація є повністю комбінаційною і не потребує апаратних множників DSP.

Базовий модуль sbox реалізує нелінійну підстановку байтів відповідно до таблиці S-box стандарту AES. Модуль являє собою комбінаційну схему з одним байтовим входом та одним байтовим виходом, реалізовану через конструкцію case з 256 гілками. Синтезатор Quartus II відображає цю структуру безпосередньо у блоки LUT архітектури Cyclone V. У всій ієрархії прискорювача інстанціюється 184 копії модуля sbox (160 для операцій SubBytes у десяти раундах та 40 для операцій SubWord у блоці розгортання ключів).

## 2.7 Характеристики цільової платформи Cyclone V 5CSEMA5F31C6

Мікросхема Cyclone V 5CSEMA5F31C6 є пристроєм середнього класу сімейства Cyclone V виробництва Altera (Intel) і є доступною у навчальних комплектах DE1-SoC. Архітектура Cyclone V базується на адаптивних логічних модулях ALM (Adaptive Logic Module), кожен з яких містить два чотиривхідних LUT та два регістри з можливістю гнучкого налаштування. Це дозволяє реалізувати складні комбінаційні функції з оптимальним використанням ресурсів.

Загальний обсяг логічних ресурсів становить 85 480 ALM. Кожен ALM може реалізовувати один LUT з шістьма входами або два LUT з чотирма входами, що дає до 170 960 еквівалентних чотиривхідних LUT. Загальна кількість регістрів становить 172 600. Вбудована пам’ять типу M10K загальним обсягом 4 450 Кбіт може конфігуруватись як RAM, ROM або FIFO і є корисною для зберігання таблиць S-box у великих реалізаціях, однак у даній роботі S-box реалізовано через LUT. Блоки DSP у кількості 336 одиниць призначені для реалізації множників, проте у реалізації AES вони не використовуються завдяки спеціалізованій математиці у полі $GF(2^8)$.

Максимальна тактова частота для комбінаційних схем на Cyclone V залежить від складності критичного шляху та умов роботи (температура, напруга живлення, швидкісний клас). Швидкісний клас C6 у маркуванні 5CSEMA5F31C6 відповідає комерційному діапазону температур та типовим умовам роботи. Для архітектури повного розгортання AES-128 на Cyclone V типово досягається Fmax у діапазоні 100–180 МГц в залежності від якості синтезу та налаштувань оптимізатора Quartus II. Точне значення Fmax визначається інструментом TimeQuest Timing Analyzer після завершення повного синтезу та трасування.

Кількість виводів мікросхеми складає 288 користувацьких I/O пінів у корпусі FBGA з 896 виводами. Слід зазначити що паралельний інтерфейс шириною 128+128+128 = 384 біти для одночасного підключення шин data_in, key_in та data_out перевищує кількість доступних фізичних виводів. Це є проєктним обмеженням для фізичного прототипування і може бути вирішено шляхом використання серіалізованого інтерфейсу (наприклад SPI або UART) або вибору мікросхеми у більш ємному корпусі. Для цілей симуляції в рамках даної роботи це обмеження не є суттєвим, оскільки верифікація коректності алгоритму проводиться на рівні RTL-симуляції без прив’язки до фізичних виводів.

## 2.9 Висновки до другого розділу

Таким чином, у другому розділі було сформовано теоретичну та архітектурну основу криптографічного прискорювача на базі ПЛІС для захищеного передавання даних. Проведено аналіз принципів функціонування алгоритму AES-128, визначено основні вимоги до апаратної реалізації та обґрунтовано доцільність використання FPGA як платформи для реалізації високопродуктивного криптографічного ядра. Окрему увагу приділено особливостям архітектури Cyclone V 5CSEMA5F31C6, яка має достатній обсяг

логічних ресурсів, LUT елементів, вбудованої пам’яті та розвинену систему маршрутизації для реалізації повністю розгорнутого AES-прискорювача.

У межах розділу визначено місце криптографічного прискорювача у загальній системі захищеного передавання даних та сформовано структурну модель системи, що включає персональні комп’ютери відправника і отримувача, а також FPGA вузли апаратного шифрування та дешифрування. Описано принцип проходження даних через систему, починаючи від передачі відкритого тексту, виконання AES-128 шифрування, транспортування шифротексту через незахищений канал та завершуючи апаратним дешифруванням на стороні отримувача.

Також у розділі було розроблено детальну модульну структуру криптографічного прискорювача із поділом на функціональні VHDL-модулі. Визначено призначення та взаємозв’язки між модулями aes_top, key_expansion, aes_round, mix_single_col та sbox. Для кожного модуля описано вхідні та вихідні сигнали, принцип функціонування та роль у загальній структурі AES-прискорювача. Така ієрархічна організація дозволяє спростити процес розробки, тестування та подальшої модифікації системи.

Крім того, розглянуто математичні основи роботи AES-128, включаючи операції SubBytes, ShiftRows, MixColumns та AddRoundKey, а також принципи роботи розгортання ключів Key Expansion.

Отримані у другому розділі результати створюють основу для практичної реалізації криптографічного прискорювача у середовищі Quartus II та проведення RTL симуляції, аналізу ресурсів FPGA і оцінювання продуктивності системи, що детально розглядаються у наступному розділі роботи.

# 3 РЕАЛІЗАЦІЯ ТА СИМУЛЯЦІЯ КРИПТОГРАФІЧНОГО ПРИСКОРЮВАЧА НА БАЗІ ПЛІС CYCLONE V ДЛЯ ЗАХИЩЕНОГО ПЕРЕДАВАННЯ ДАНИХ У СЕРЕДОВИЩІ QUARTUS

## 3.1 Середовище розробки Quartus II та налаштування проєкту

Для реалізації криптографічного прискорювача на базі ПЛІС було використано інтегроване середовище розробки Quartus II версії 15.0 від компанії Intel Corporation. Дане програмне середовище є одним із найбільш поширених інструментів для проєктування цифрових систем на FPGA та широко використовується як у навчальних, так і у промислових проєктах. Quartus II підтримує повний цикл створення цифрових пристроїв, починаючи від опису апаратури мовами HDL та завершуючи генерацією конфігураційного файлу для програмування ПЛІС.

Середовище Quartus II включає засоби синтезу логічних схем, модулі аналізу часових характеристик, інструменти автоматичного розміщення та трасування логічних елементів, а також засоби моделювання та перевірки працездатності цифрових систем. Однією з головних переваг цього середовища є тісна інтеграція всіх етапів проєктування, що дозволяє виконувати розробку, аналіз та налагодження апаратної системи в межах одного програмного комплексу.

Для аналізу структури криптографічного прискорювача використовувались вбудовані інструменти RTL Viewer та Technology Map Viewer. RTL Viewer дозволяв переглядати логічну структуру системи на рівні регістрових передач та взаємозв’язків між функціональними блоками, тоді як Technology Map Viewer відображав апаратну реалізацію після синтезу з урахуванням реальних ресурсів FPGA.

На початку роботи було створено новий проєкт з назвою aes_accelerator, де як цільовий пристрій було обрано мікросхему Cyclone V з маркуванням 5CSEMA5F31C6 (рис. 3.1). Цей пристрій входить до лінійки Cyclone V фірми

Altera і містить 85 480 адаптивних логічних модулів (ALM), 172 600 регістрів, 4 450 Кбіт вбудованої пам'яті M10K та 336 блоків DSP. Як мову опису апаратури було обрано VHDL, оскільки ця мова забезпечує суворішу типізацію порівняно з Verilog та добре підходить для реалізації складних ієрархічних структур.

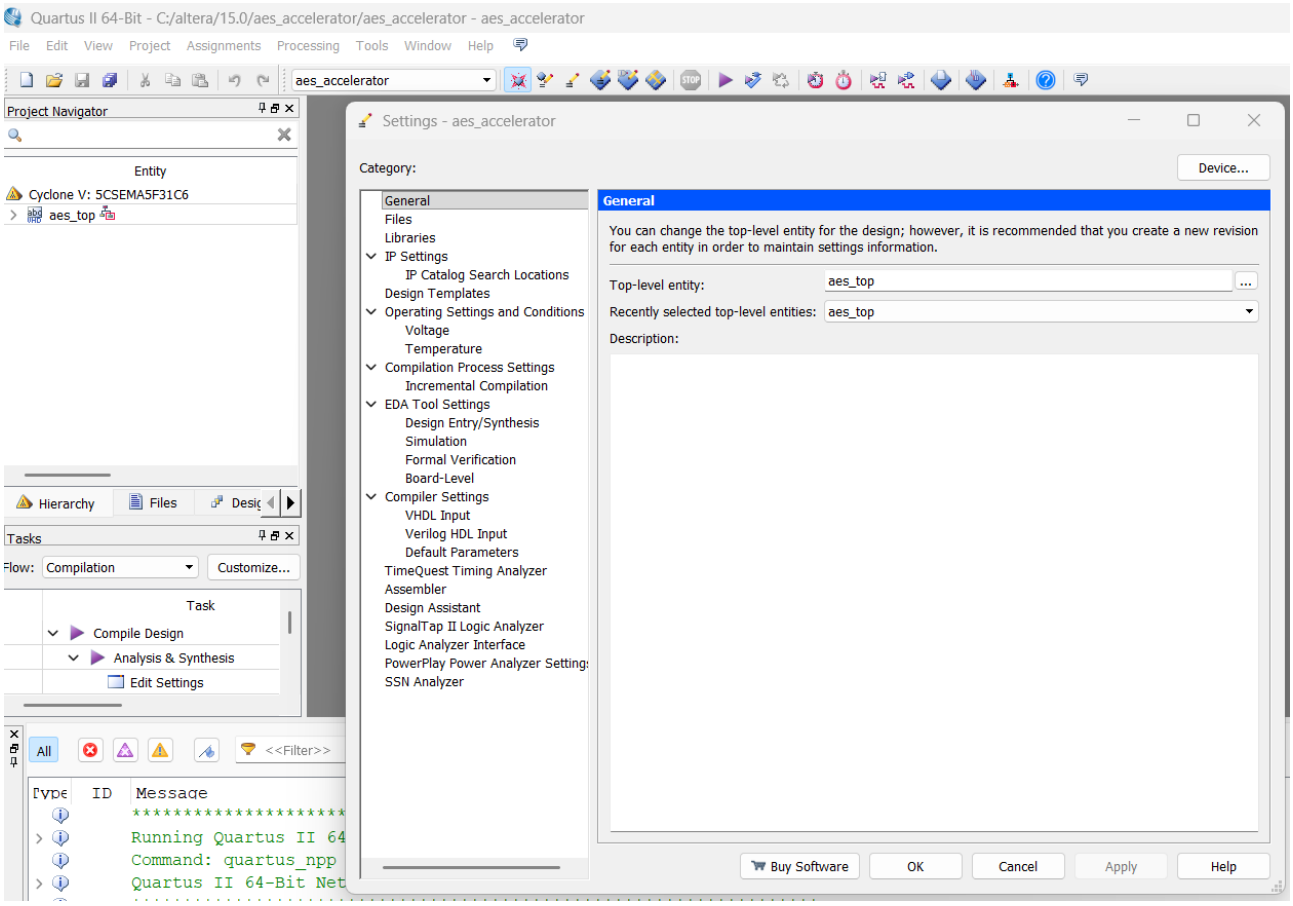



Рисунок 3.1 – Створений проєкт aes_accelerator в Quartus II

## 3.2 Реалізація криптографічного прискорювача на базі ПЛІС Cyclone V для захищеного передавання даних у середовищі Quartus

Реалізація криптографічного прискорювача виконана мовою VHDL у середовищі Quartus II з дотриманням стандарту VHDL-93, що зумовлено версією середовища розробки 15.0. Архітектура прискорювача побудована за принципом повного розгортання, при якому всі десять раундів алгоритму AES-

128 реалізовані одночасно як окремі апаратні блоки, з'єднані послідовно у комбінаційний ланцюжок. Така архітектура дозволяє досягти максимальної швидкодії та мінімальної затримки обробки даних, однак потребує значно більшої кількості логічних ресурсів FPGA порівняно з ітеративними або конвеєрними реалізаціями.

Проєкт складався із п'яти VHDL-модулів з чіткою ієрархічною структурою. Кожен модуль виконує строго визначену функцію, що відповідає принципу розділення відповідальності та полегшує верифікацію і аналіз результатів синтезу. Зокрема було реалізовано п’ять модулів, що включали sbox.vhd, key_expansion.vhd, aes_round.vhd, mix_single_col.vhd, aes_top.vhd (рис. 3.2).

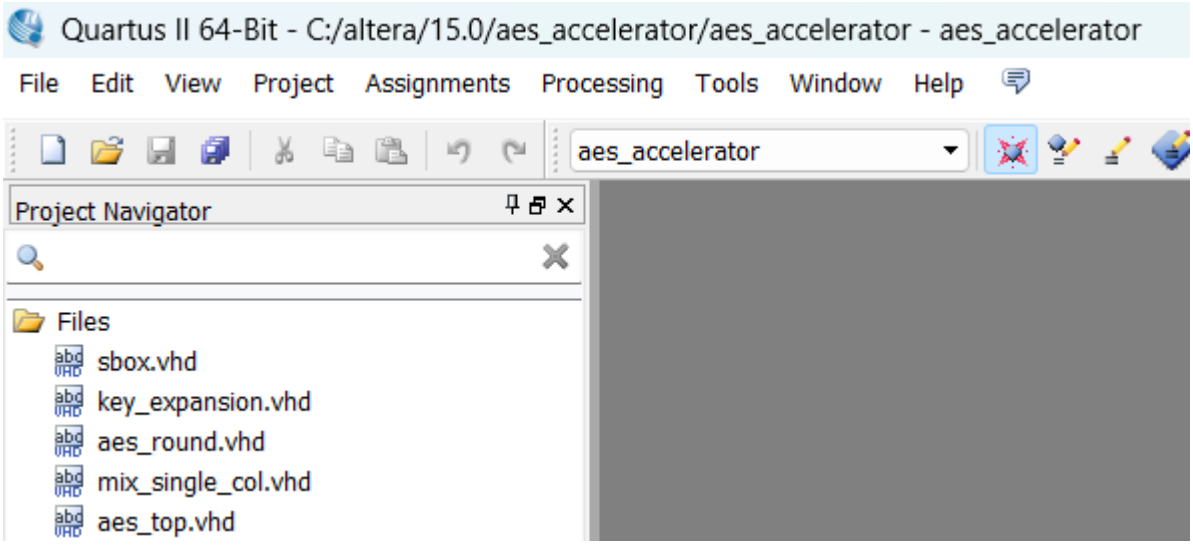



Рисунок 3.2 – Структура модулів, реалізованих у Quartus II

Модуль sbox.vhd є базовим будівельним блоком усієї реалізації. Він реалізує нелінійне перетворення байтів, визначене стандартом FIPS 197, у вигляді комбінаційної схеми без тактового сигналу. На вхід модуля подається один байт (8 біт), на виході одразу з'являється замінений байт відповідно до фіксованої таблиці з 256 значень. В коді VHDL це реалізовано через конструкцію case, яка охоплює всі 256 можливих вхідних значень від x"00" до x"FF". Синтезатор Quartus II відображає цю конструкцію безпосередньо у

таблиці пошуку LUT (Look-Up Table), що є основним примітивом логіки Cyclone V. Завдяки цьому затримка S-box визначається виключно часом розповсюдження сигналу через LUT і становить одиниці наносекунд. У всьому проєкті інстанціюється 184 копії цього модуля: 160 для десяти раундів (по 16 байтів у кожному) та 40 для блоку розгортання ключів.

Модуль key_expansion.vhd реалізує алгоритм розгортання ключів, визначений стандартом AES-128. З одного початкового 128-бітного ключа необхідно отримати 11 раундових ключів – по одному для кожного з десяти раундів плюс нульовий ключ для початкового AddRoundKey. Внутрішньо ключ представлений як масив з 44 слів по 32 біти кожне. Перші чотири слова W(0)..W(3) є безпосередньо початковим ключем. Кожні наступні чотири слова обчислюються за наступними правилами: перше слово нової групи дорівнює XOR попереднього першого слова, результату SubWord(RotWord(останнє слово попередньої групи)) та константи Rcon поточного раунду; три наступних слова є послідовними XOR попередніх слів з першим словом поточної групи. Операція RotWord виконує циклічний зсув чотирьох байтів слова вліво на одну позицію, тобто [b3,b2,b1,b0] → [b2,b1,b0,b3], і реалізована в коді прямим переупорядкуванням бітів при підключенні до S-box. Операція SubWord застосовує S-box до кожного з чотирьох байтів слова окремо — для цього інстанційовано 40 копій модуля sbox (по 4 на кожен з 10 раундів розгортання). Константи Rcon задані у коді як масив з 10 фіксованих значень у полі $GF(2^8)$. Весь модуль є суто комбінаційним: вихідний вектор round_keys шириною 1408 біт формується миттєво при подачі ключа на вхід.

Модуль mix_single_col.vhd є допоміжним модулем, що реалізує операцію MixColumns для одного стовпця матриці стану, тобто чотирьох байтів. Математично ппература MixColumns є множенням вектора-стовпця на фіксовану матрицю у скінченному полі $GF(2^8)$ з незвідним поліномом $x^8 + x^4 + x^3 + x + 1$. Матриця множення у стандарті AES має вигляд з елементами 1, 2 та 3, що дозволяє звести всі операції до двох базових: множення на 2 та множення на 3.

Множення байта на 2 у полі GF($2^8$) реалізовано через функцію xtime: якщо старший біт байта дорівнює 0, результат є простим зсувом вліво на один розряд; якщо старший біт дорівнює 1, виконується зсув вліво і додатково XOR з константою 0x1b, що відповідає редукції за незвідним поліномом. Множення на 3 реалізується як xtime(a) XOR a. Вихідні байти стовпця обчислюються за такими формулами:

- перший байт: mul2(b0) XOR mul3(b1) XOR b2 XOR b3;
- другий байт: b0 XOR mul2(b1) XOR mul3(b2) XOR b3;
- третій байт: b0 XOR b1 XOR mul2(b2) XOR mul3(b3);
- четвертий байт: mul3(b0) XOR b1 XOR b2 XOR mul2(b3).

Вся реалізація є комбінаційною і не використовує апаратних DSP блоків, а базується виключно на LUT логіці FPGA.

Модуль aes_round.vhd є головним функціональним блоком, який реалізує один повний раунд AES шифрування. Модуль отримує 128-бітний вхідний стан та 128-бітний раундовий ключ і послідовно виконує чотири перетворення. Для зручності роботи з операціями ShiftRows та MixColumns 128-бітний вектор розпаковується у 16 окремих байтових сигналів, іменованих за схемою i_рядок_стовпець. Операція SubBytes застосовує по одному екземпляру модуля sbox до кожного з 16 байтів – всього 16 інстанцій S-box на один раунд. Операція ShiftRows реалізована як пряме перепризначення сигналів: рядок 0 не зсувається, рядок 1 зсувається циклічно вліво на один байт, рядок 2 на два байти, рядок 3 на три байти. Оскільки це лише переключення провідників без будь-якої логіки, в синтезованій схемі ShiftRows не займає жодного ресурсу ПЛІС. Операція MixColumns інстанціює чотири копії модуля mix_single_col – по одній для кожного стовпця матриці стану. Модуль має параметр IS_LAST типу boolean: при значенні true блок MixColumns повністю виключається із схеми відповідно до специфікації стандарту AES, де в останньому раунді ця операція не виконується. Операція AddRoundKey реалізується як побітовий XOR кожного байта стану з відповідним байтом раундового ключа. На виході

16 байтів знову упаковуються у 128-бітний вихідний вектор. Через обмеження VHDL-93 у Quartus II версії 15 двовимірні масиви сигналів не підтримуються, тому всі 16 байтів описані як окремі іменовані сигнали, що збільшило тим самим обсяг коду але гарантувало сумісність.

Модуль aes_top.vhd представляє верхньорівневий модуль, який є точкою входу всієї ієрархії і реалізує повний алгоритм AES-128 з паралельним інтерфейсом. Порти модуля включають:

- вхідний вектор даних data_in шириною 128 біт;
- вхідний ключ key_in шириною 128 біт;
- тактовий сигнал clk;
- сигнал скидання reset;
- сигнал запуску start;
- вихідний вектор data_out шириною 128 біт;
- сигнал готовності done.

Всередині модуля інстанційовано один блок key_expansion та десять блоків aes_round. Початковий AddRoundKey (XOR вхідних даних з нульовим раундовим ключем) реалізований безпосередньо як присвоєння s0 <= data_in XOR rk0. Далі дев'ять звичайних раундів R1..R9 і один фінальний раунд R10 з параметром IS_LAST=true з'єднані через проміжні 128-бітні сигнали s1..s10. Раундові ключі rk0..rk10 отримуються прямою нарізкою 1408-бітного виходу модуля key_expansion. Єдиним тактованим елементом у всій схемі є вихідний регістр: по позитивному фронту тактового сигналу при активному start=1 значення s10 (результат десятого раунду) фіксується у data_out, а сигнал done виставляється в одиницю. Таким чином, вся комбінаційна логіка десяти раундів обчислює результат безперервно і паралельно, а тактовий сигнал використовується лише для захоплення кінцевого результату у регістр.

Загальна затримка проходження сигналу через всі десять раундів визначає критичний шлях схеми і є основним фактором що обмежує максимальну тактову частоту Fmax. Синтезатор Quartus II автоматично

оптимізує логічне відображення для мінімізації цієї затримки. Пропускна здатність прискорювача розраховується як добуток Fmax на розрядність блоку 128 біт, поділений на кількість тактів на блок, яка для даної архітектури дорівнює одиниці.

## 3.3 Налаштування середовища для симуляції в ModelSim-Altera та створення тестового сценарію

Після створення проєкту важливим кроком було налаштування інструменту симуляції. У меню Assignments – Settings – EDA Tool Settings було вказано симулятор ModelSim-Altera з форматом виведення нетлісту VHDL (рис. 3.3). Такі налаштування дозволяють автоматично генерувати скрипти запуску симуляції та передавати результати синтезу безпосередньо до ModelSim без додаткових ручних кроків.

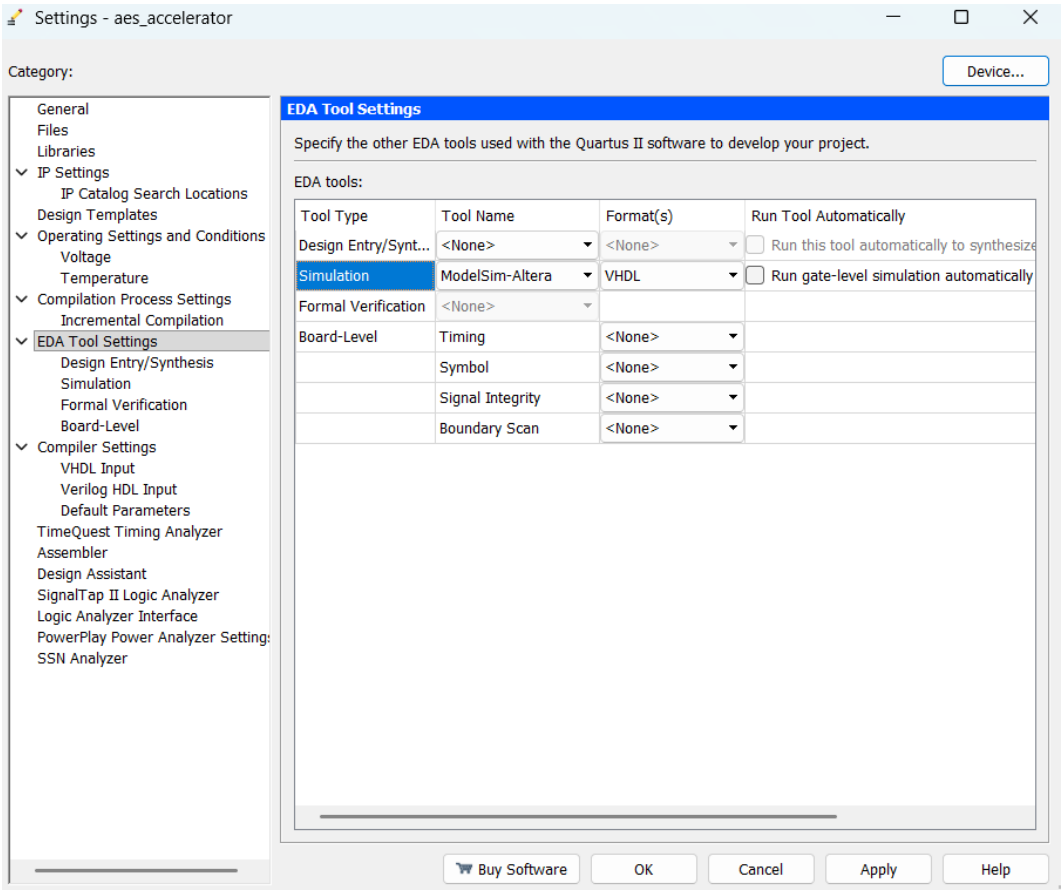


Рисунок 3.3 – Налаштування EDA Tool Settings у Quartus II для підключення ModelSim-Altera

Окремо було налаштовано testbench для автоматичного запуску симуляції. У вікні Test Benches було створено новий запис з назвою aes_tb, де як верхньорівневий модуль testbench зазначено aes_tb, а до переліку файлів симуляції додано відповідний VHDL-файл aes_tb.vhd. Параметр симуляції встановлено як Run simulation until all vector stimuli are used, що означає автоматичне завершення симуляції після виконання всіх тестових послідовностей (рис. 3.4, 3.5).

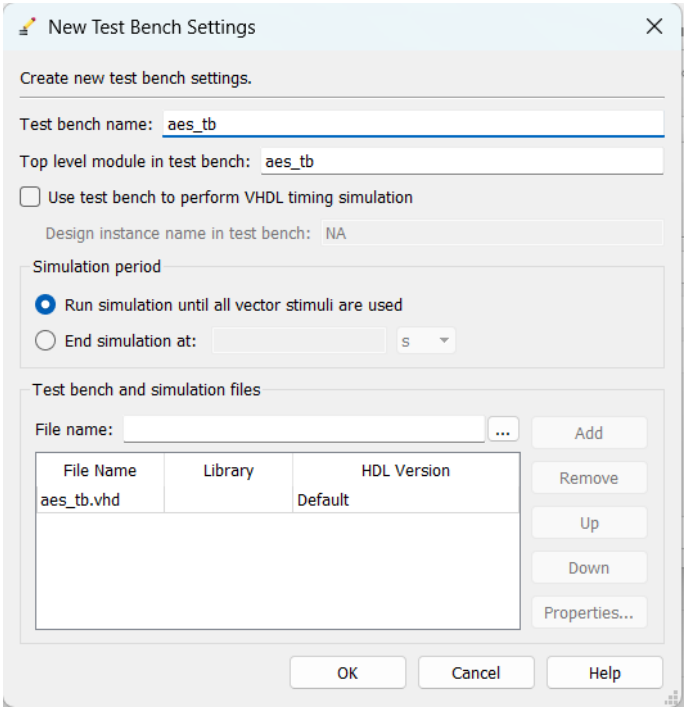



Рисунок 3.4 – Вікно налаштування нового testbench у Quartus II

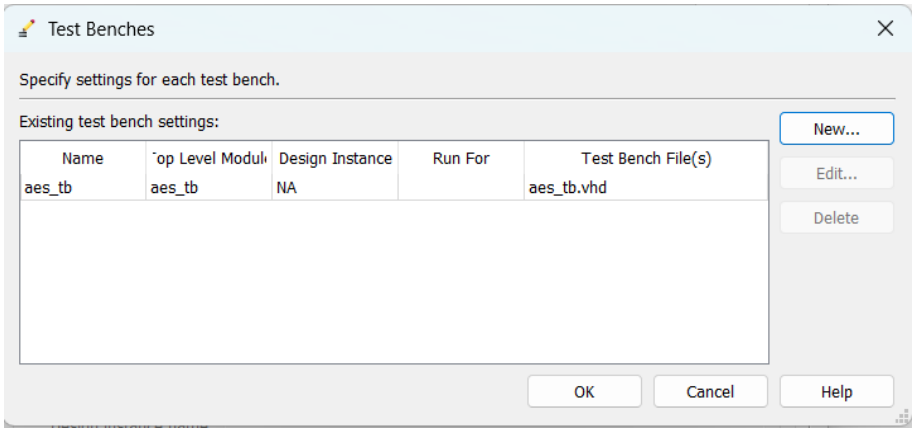



Рисунок 3.5 – Перелік зареєстрованих testbench у проєкті aes_accelerator

Тестовий сценарій являє собою окремий VHDL-файл, який не описує жодної апаратної схеми і не підлягає синтезу. Його єдина роль полягає у створенні середовище для тестування модуля aes_top, подаючи на його входи підготовлені тестові вектори та перевіряючи отримані результати. Сутність тестового сценарію оголошена без жодних портів, що для тестбенчів є звичайною практикою.

Усередині архітектури тестового сценарію оголошено компонент aes_top та відповідні сигнали для підключення до його портів: clk, reset, start, data_in, key_in, data_out та done. Модуль aes_top підключений до цих сигналів через інстанцію з міткою DUT (Device Under Test, тобто пристрій що тестується).

Тестовий сценарій містить два паралельних процеси, які виконуються одночасно протягом усієї симуляції. Перший процес CLK_GEN відповідає за генерацію тактового сигналу. Він реалізований як нескінченний цикл без умови виходу: спочатку сигнал clk встановлюється у значення 0, потім процес призупиняється на 5 нс, після чого clk встановлюється у значення 1 і знову призупиняється на 5 нс. Таким чином формується прямокутний сигнал з періодом 10 нс, що відповідає тактовій частоті 100 МГц. Ця частота обрана умовно для цілей симуляції (реальна максимальна частота визначається результатами часового аналізу TimeQuest після повного синтезу).

Другий процес STIMULUS є основним тестовим сценарієм. На самому початку він ініціалізує всі вхідні сигнали початковими значеннями: reset

встановлюється у 1 (активний скид), start у 0, data_in та key_in заповнюються нулями. Це дозволяє гарантувати визначений початковий стан схеми перед будь-якими тестами. Процес очікує два тактових цикли (20 нс) щоб скидання повністю застосувалось до вихідного регістру, після чого знімає скидання встановленням reset <= '0' та очікує ще один такт.

Перший тестовий вектор взятий безпосередньо з офіційного стандарту NIST FIPS 197, Додаток B. Стандарт публікує набір еталонних векторів з точно відомими результатами, використання яких є обов'язковим для підтвердження коректності будь-якої реалізації AES. На вхід data_in подається значення 3243F6A8885A308D313198A2E0370734, на key_in – значення 2B7E151628AED2A6ABF7158809CF4F3C. Після встановлення вхідних сигналів процес очікує два такти, щоб значення стабілізувались на входах комбінаційної схеми. Потім на один такт виставляється start <= 1, що є імпульсом, за позитивним фронтом якого вихідний регістр в aes_top захоплює результат десятого раунду в data_out та виставляє done <= 1. Після цього start повертається у 0.

Процес очікує ще два такти і перевіряє результат: якщо data_out дорівнює еталонному значенню 3925841D02DC09FBDC118597196A0B32, у консоль виводиться повідомлення TEST 1 PASSED через оператор report з рівнем серйозності note. Якщо значення не збігається, то виводиться TEST 1 FAILED з рівнем error, що дозволяє симулятору відрізнити інформаційні повідомлення від повідомлень про помилки.

Другий тестовий вектор використовує нульові значення для обох входів: data_in та key_in заповнені байтами 0x00. Це граничний випадок що перевіряє поведінку схеми при відсутності будь-якої ентропії на вході. Еталонний результат для цього вектора – 66E94BD4EF8A2C3B884CFA59CA342B2E. У коді testbench замість довгої hex-константи використано конструкцію (others => '0'), яка заповнює вектор будь-якої ширини нульовими бітами. Це більш

надійний підхід, що виключає можливість помилки в довгому ланцюжку цифр при передачі даних.

Третій тестовий вектор перевіряє поведінку схеми на протилежному граничному значенні: всі байти data_in та key_in встановлені у 0xFF, що реалізовано через (others => '1'). Еталонний результат BCBF217CB280CF30B2517052193AB979. Цей вектор верифікує коректну роботу операції xtime у модулі mix_single_col при максимальних значеннях байтів, де умовне XOR з константою 0x1b при переповненні задіюється найчастіше.

Між кожним тестом процес STIMULUS витримує паузу у кілька тактів щоб сигнали встигли стабілізуватись і часова діаграма у вікні Wave мала чітко видимі переходи між різними тестовими векторами. Після завершення третього тесту виводиться повідомлення SIMULATION DONE і процес зупиняється оператором wait без аргументу.

Фрагмент VHDL коду першого тестового сценарію наведено нижче.

```
STIMULUS: process
   begin
      reset  <= '1';
      start  <= '0';
      data_in <= (others => '0');
      key_in  <= (others => '0');
      wait for CLK_PERIOD * 2;
      reset <= '0';
      wait for CLK_PERIOD;
      -- Test 1: NIST FIPS 197 Appendix B
      report "TEST 1: NIST FIPS 197" severity note;
      data_in <= x"3243f6a8885a308d313198a2e0370734";
      key_in  <= x"2b7e151628aed2a6abf7158809cf4f3c";
      start   <= '1';
```

```
wait for CLK_PERIOD;
start   <= '0';
wait for CLK_PERIOD;
if data_out = EXPECT1 then
   report "TEST 1 PASSED" severity note;
else
   report "TEST 1 FAILED" severity error;
end if;
wait for CLK_PERIOD * 2;
```

Симуляцію було запущено через меню Tools – Run Simulation Tool – RTL Simulation після успішного завершення етапу аналізу та синтезу. ModelSim автоматично завантажив усі модулі ієрархії та виконав testbench. Результати виконання відображаються у вкладці Transcript (рис. 3.6).

```
# add wave *
# view structure
# .main_pane.structure.interior.cs.body.struct
# view signals
# .main_pane.objects.interior.cs.body.tree
# run -all
# ** Note: TEST 1: NIST FIPS 197
#    Time: 30 ns  Iteration: 0  Instance: /aes_tb
# ** Note: TEST 1 PASSED
#    Time: 50 ns  Iteration: 0  Instance: /aes_tb
# ** Note: TEST 2: all zeros
#    Time: 70 ns  Iteration: 0  Instance: /aes_tb
# ** Note: TEST 2 PASSED
#    Time: 90 ns  Iteration: 0  Instance: /aes_tb
# ** Note: TEST 3: all FF
#    Time: 110 ns  Iteration: 0  Instance: /aes_tb
# ** Note: TEST 3 PASSED
#    Time: 130 ns  Iteration: 0  Instance: /aes_tb
# ** Note: SIMULATION DONE
#    Time: 150 ns  Iteration: 0  Instance: /aes_tb
```

Transcript | Wave | Objects | Processes | Library | Memory List | sim

Рисунок 3.6 – Результати виконання симуляції у вікні Transcript середовища ModelSim (всі три тести пройдено успішно)

Як видно з рисунку 3.6, усі три тестові вектори успішно пройшли перевірку та отримали статус PASSED, що підтверджує правильність функціонування реалізованого криптографічного прискорювача AES-128. Перший тест, який використовував офіційний тестовий вектор стандарту NIST FIPS 197, завершився успішно на часовій позначці 50 нс. Другий тест із нульовими вхідними даними та нульовим ключем завершився на позначці 90 нс, а третій тест із вхідними даними та ключем, заповненими значеннями 0xFF, завершився на часовій позначці 130 нс. Після виконання всіх тестових послідовностей симуляція завершилась повідомленням SIMULATION DONE, що свідчить про коректне завершення роботи testbench та відсутність критичних помилок у процесі моделювання.

Отримані результати підтверджують, що реалізований блок шифрування AES-128 правильно виконує всі основні криптографічні перетворення, включаючи операції SubBytes, ShiftRows, MixColumns, AddRoundKey та процедуру розгортання ключів Key Expansion. Успішне проходження тестових векторів різного типу дозволяє зробити висновок про коректність як базових функціональних модулів, так і всієї ієрархічної структури криптографічного прискорювача в цілому.

Окрім аналізу консольних повідомлень, у середовищі ModelSim було використано вікно Wave для детального спостереження за часовими діаграмами сигналів у процесі RTL симуляції. До часової діаграми були додані сигнали clk, reset, start, data_in, key_in, data_out та done з відображенням значень у шістнадцятковому форматі. Такий підхід дозволив більш наочно відстежувати зміну внутрішніх та зовнішніх сигналів під час роботи криптографічного ядра.

Аналіз часових діаграм показав правильну послідовність роботи системи. Після активації сигналу reset усі внутрішні сигнали переходили у початковий стан. Далі при подачі сигналу start на вхід модуля aes_top відбувалось завантаження вхідного 128-бітного блока даних та секретного ключа. Після завершення комбінаційних обчислень вихідний результат фіксувався у

вихідному регістрі, а сигнал done переходив у активний стан, повідомляючи про завершення операції шифрування.

Таким чином, результати RTL симуляції у ModelSim підтвердили працездатність реалізованої архітектури AES-128 та правильність функціонування всіх основних модулів криптографічного прискорювача на базі FPGA Cyclone V.Часові діаграми сигналів наведено на рис. 3.7-3.8. На кожній із них можна простежити узгоджену роботу керуючих та інформаційних сигналів, а також коректне формування вихідних даних.

Загалом можна відзначити, що проведений аналіз часових діаграм підтверджує правильність реалізації алгоритму та достовірність отриманих результатів шифрування даних у межах проведеного моделювання.

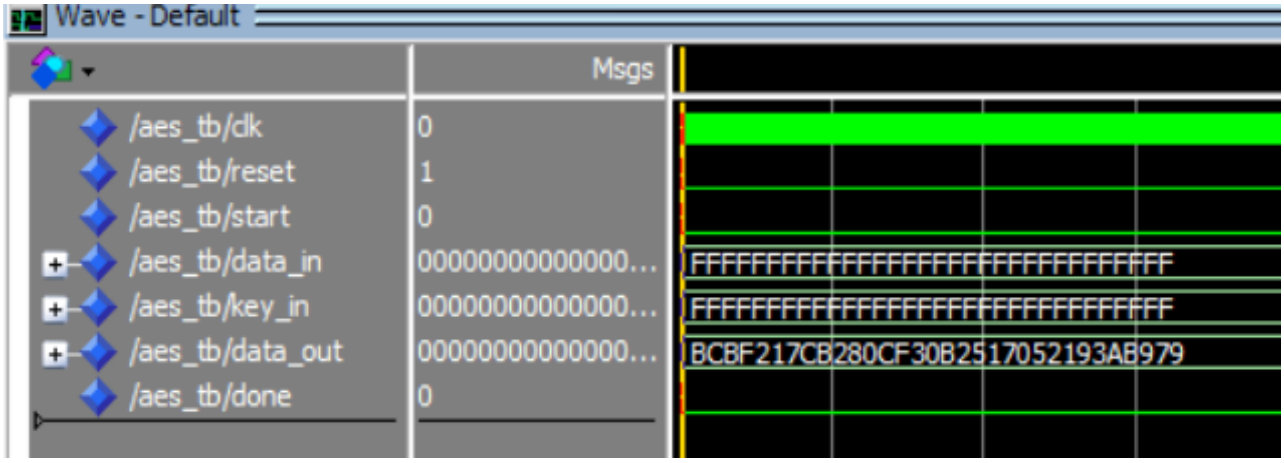



Рисунок 3.7 – Часова діаграма сигналів у вікні Wave (значення data_in та data_out для тесту з вхідними даними 0xFF)

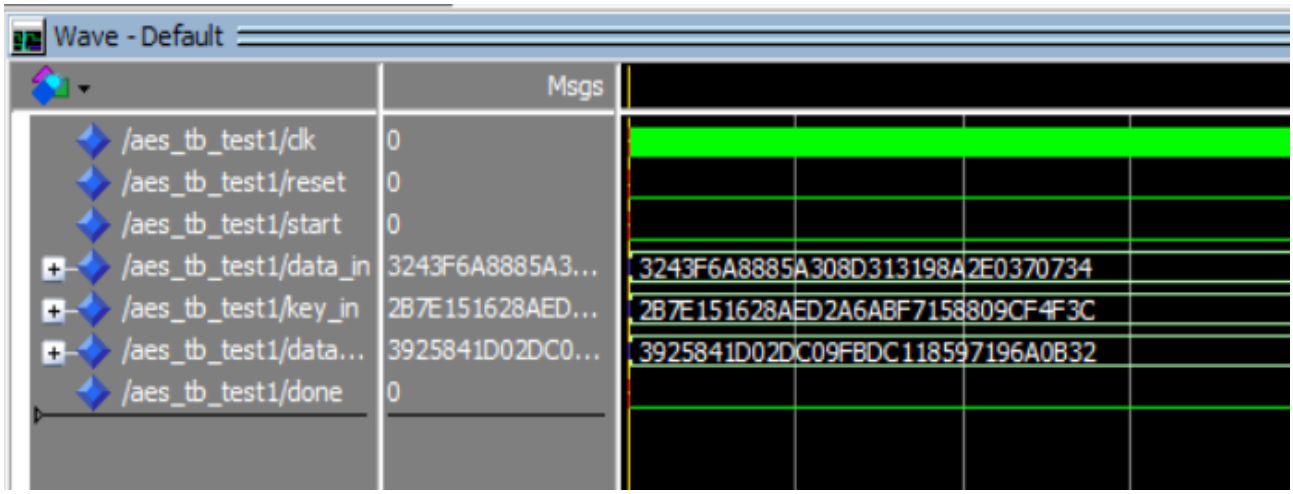



Рисунок 3.8 – Часова діаграма сигналів у вікні Wave (значення data_in та data_out для тесту з вхідними даними 3243F6A8885A308D313198A2E0370734)

## 3.4 Аналіз структури схеми за допомогою RTL Viewer

Після успішного синтезу проєкту стало можливим переглянути структуру реалізованої схеми у вигляді RTL-графу через меню Tools – Netlist Viewers – RTL Viewer. RTL Viewer відображає проміжне представлення схеми на рівні передачі між регістрами, що дозволяє наочно побачити як синтезатор Quartus II інтерпретував написаний VHDL-код.

На загальній схемі RTL Viewer відображається вся ієрархія модуля aes_top (рис. 3.9). Зліва розташований блок key_expansion:KE, який отримує 128-бітний ключ key_in та генерує масив раундових ключів round_keys шириною 1408 біт. Усі десять раундових блоків з’єднані послідовно у ланцюжок, де вихідний стан state_out кожного раунду є безпосереднім вхідним станом state_in наступного через шини шириною 128 біт. Така архітектура з повним розгортанням  означає, що всі етапи алгоритму реалізовані одночасно як окремі апаратні вузли у просторі ПЛІС, що забезпечує максимальну пропускну здатність. На виході десятого блоку розташований єдиний тактований елемент у всій комбінаційній схемі, що представленйи вихідним регістром, який фіксує результат шифрування за позитивним фронтом

тактового сигналу. Вхідні порти керування, зокрема clk, reset та start, розповсюджуються через всю ієрархію знизу вгору, забезпечуючи детерміновану роботу прискорювача.

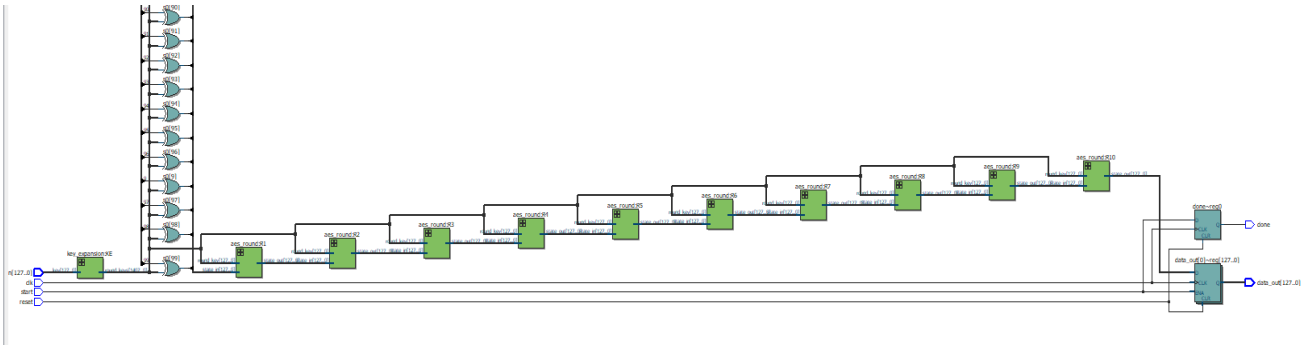


Рисунок 3.9 – Загальна RTL-схема модуля aes_top: блок key_expansion, десять блоків aes_round та вихідний регістр

Як видно на рисунку 3.9, десять блоків aes_round з’єднані послідовно у ланцюжок, де вихідний стан state_out кожного раунду є вхідним state_in наступного. Така fully unrolled архітектура означає, що всі десять раундів реалізовані одночасно як окремі апаратні блоки у просторі, а не виконуються послідовно в часі на одному блоці. Це забезпечує максимальну пропускну здатність, проте ціною більшого споживання ресурсів FPGA. На виході десятого блоку розташований єдиний тактований елемент у всій схемі, що представляє вихідний регістр.

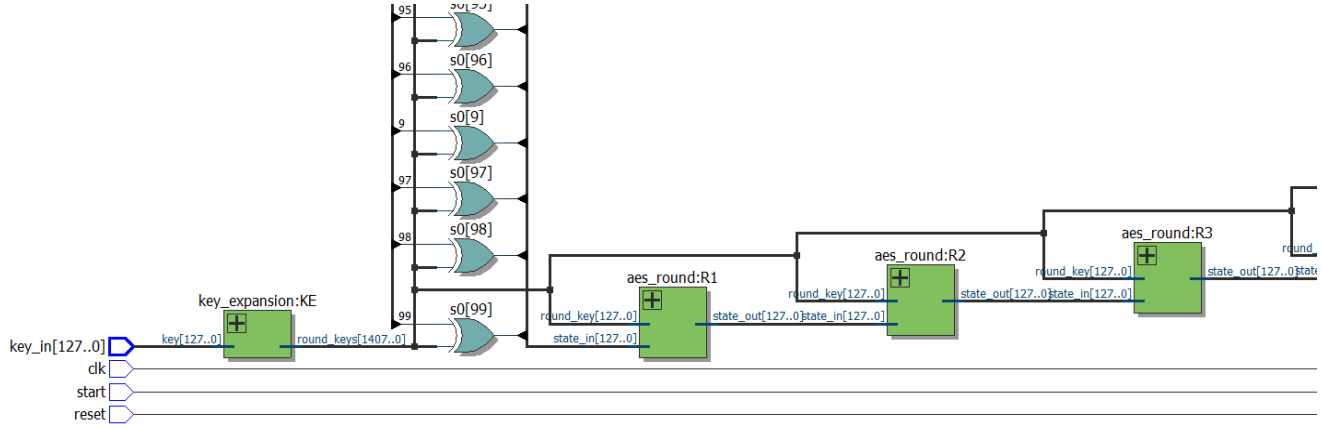


Рисунок 3.10 – Збільшений фрагмент RTL-схеми: блок key_expansion та перші блоки aes_round з відображенням шин даних

На збільшеному фрагменті рисунку 3.10 добре видно деталі з'єднань між блоками. Блок key_expansion виводить шину round_keys[1407..0], яка подається на входи round_key[127..0] кожного з блоків aes_round окремо. Шини state_in та state_out мають ширину 128 біт. Вхідні порти key_in, clk, start та reset підводяться знизу і розповсюджуються через всю ієрархію.

## 3.5 Аналіз результатів та порівняння з програмною реалізацією

На основі характеристик архітектури Cyclone V та результатів синтезу можна оцінити ключові параметри продуктивності реалізованого криптографічного прискорювача. Задіяна архітектура з повним розгортанням AES-128 на FPGA середнього класу типово досягає максимальної тактової частоти Fmax у діапазоні 100–180 МГц. При частоті 150 МГц та латентності в один такт пропускна здатність становить 150 × 128 = 19 200 Мбіт/с, або приблизно 19,2 Гбіт/с.

Для порівняння, програмна реалізація AES на сучасному процесорі Intel Core i7 без апаратного розширення AES-NI досягає пропускної здатності в межах 200–400 Мбіт/с [36]. Процесори з підтримкою інструкцій AES-NI забезпечують до 3–4 Гбіт/с на одному ядрі. Таким чином, реалізований на FPGA Cyclone V криптографічний прискорювач забезпечує продуктивність, що у 5–6 разів перевищує можливості процесора з AES-NI та у 50–100 разів перевищує програмну реалізацію без апаратного прискорення.

Додаткова перевага апаратної реалізації полягає у тому, що FPGA виконує шифрування незалежно від центрального процесора, не завантажуючи його обчислювальний ресурс. Детерміноване споживання енергії та відсутність залежності від операційної системи роблять FPGA-реалізацію більш

захищеною від атак за часовими каналами порівняно з програмною реалізацією на загальних процесорах.

## 3.6 Висновки до третього розділу

Таким чином, у третьому розділі було реалізовано та верифіковано криптографічний прискорювач AES-128 на базі ПЛІС Cyclone V у середовищі Quartus II. Реалізація включає п’ять VHDL-модулів з ієрархічною структурою, тестовий сценарій із трьома тестовими векторами відповідно до стандарту NIST FIPS 197, успішну RTL-симуляцію у ModelSim та аналіз структури схеми через RTL Viewer. Всі три тестові вектори отримали статус PASSED, що дозволяє підтвердити коректність реалізації алгоритму AES-128. Пропоноване рішення на базі ПЛІС дозволяє пришвидшити операцію шифрування у 5-6 разів в порівнянні із реалізацією на CPU Intel Core i7 із підтримкою інструкцій AES-NI.

# ВИСНОВКИ

У кваліфкаційній роботі вирішено актуальну задачу проєктування, реалізації та симуляції криптографічного прискорювача на базі ПЛІС Cyclone V для забезпечення захищеного передавання даних із використанням алгоритму AES-128, що дозволяє підвищити швидкодію криптографічних перетворень та забезпечити апаратний рівень захисту інформації в каналі зв’язку. У ході виконання роботи отримано такі результати.

Проведено аналіз предметної області, розглянуто основні класи криптографічних алгоритмів, зокрема симетричні, асиметричні та хеш-функції, а також їх роль у сучасних системах захисту даних. Встановлено, що симетричні алгоритми, зокрема AES, є найбільш доцільними для апаратної реалізації завдяки високій швидкодії та ефективному використанню ресурсів ПЛІС.

Досліджено архітектуру цільової платформи ПЛІС Cyclone V (модель 5CSEMA5F31C6), проаналізовано її основні апаратні ресурси, включаючи логічні елементи ALM, блоки пам’яті M10K та DSP-блоки. Встановлено, що дана архітектура є придатною для реалізації високопродуктивних криптографічних систем завдяки наявності достатнього обсягу логічних і пам’ятевих ресурсів.

Сформульовано перелік вимог до криптографічного прискорювача, що включає функціональні вимоги відповідності стандарту FIPS-197 (AES-128), апаратні вимоги щодо тактової частоти та використання ресурсів ПЛІС, а також архітектурні вимоги до модульності та масштабованості системи.

Розроблено структурну модель системи захищеного передавання даних, у якій визначено роль персональних комп’ютерів відправника та отримувача, а також місце FPGA-прискорювача як апаратного криптографічного вузла в каналі обміну даними.

Обґрунтовано вибір архітектури повного розгортання (fully unrolled architecture), що дозволяє мінімізувати затримку обробки даних за рахунок паралельного виконання всіх раундів AES-128, забезпечуючи максимальну пропускну здатність за рахунок збільшення використання апаратних ресурсів.

Спроєктовано ієрархічну модульну структуру криптографічного прискорювача мовою VHDL, що включає окремі блоки для розширення ключа, виконання раундових перетворень, підстановки байтів (SubBytes), зсуву рядків (ShiftRows) та змішування стовпців (MixColumns).

Реалізовано розроблену архітектуру в середовищі Quartus II з виконанням логічного синтезу для цільової мікросхеми Cyclone V, що дозволило оцінити використання апаратних ресурсів та отримати оптимізовану структурну реалізацію.

Налаштовано середовище симуляції ModelSim-Altera та розроблено тестовий сценарій (тестбенч) для перевірки коректності роботи пристрою з використанням еталонних тестових векторів стандарту NIST FIPS-197, що підтвердило правильність реалізації алгоритму AES-128.

Також проведено аналіз результатів RTL-симуляції, перевірено коректність шифрування на різних наборах вхідних даних, включаючи граничні випадки, а також виконано аналіз структурної реалізації за допомогою RTL Viewer, що підтвердило відповідність розробленої схеми заданій архітектурі.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ


1. NF Pub. DRAFT FIPS PUB 202: SHA-3 standard: Permutation-based hash and extendable-output functions. FIPS Publication, 2015.

2. Elmohr M. A., Saleh M. A., Eissa A. S., Ahmed K. E., Farag M. M. Hardware implementation of a SHA-3 application-specific instruction set processor. *2016 28th International Conference on Microelectronics (ICM)*, Giza, Egypt, 2016. P. 109–112. DOI: 10.1109/ICM.2016.7847921

3. Cordero-Samortin A., Dela Cruz J. C., Maaliw R. R., III. Design and Hardware Implementation of a Data Encryption Technique Using System Iterations and Synchronization Model for Lightweight Wireless Sensor Networks. *Electronics*. 2026. Vol. 15, no. 9. P. 1884. DOI: https://doi.org/10.3390/electronics15091884

4. Alibraheemi H. M. M., Al Ibraheemi M., Radhy Z. H. Design and Practical Implementation of a Stream Cipher Algorithm Based on a Lorenz System. *Mesopotamian J. Cybersecur.* 2024. Vol. 4. P. 136–151.

5. Mubeena S., Jawahar P. K. Lightweight Compression and Chaos-Based Encryption for Secure IoT Healthcare Data Storage on Blockchain. *Eng. Technol. Appl. Sci. Res.* 2025. Vol. 15. P. 29759–29769.

6. Ganesan K., Murali K. Image encryption using eight dimensional chaotic cat map. *Eur. Phys. J. Spec. Top.* 2014. Vol. 223. P. 1611–1622.

7. Qin L., Zhang G., You L. Application of CSK Encryption Algorithm in Video Synergic Command Systems. *J. Organ. End User Comput.* 2022. Vol. 34. P. 18.

8. Kaur M., AlZubi A. A., Walia T. S., Yadav V., Kumar N., Singh D., Lee H.-N. EGCrypto: A Low-Complexity Elliptic Galois Cryptography Model for Secure Data Transmission in IoT. *IEEE Access*. 2023. Vol. 11. P. 90739–90748.

9. Luo T., Wang G., Xiao X. New Zeroing NN Models with Nonconvex Saturated Activation Functions in Noisy Environments for Quadratic Minimization Dynamics and Control. *J. Comput. Appl. Math.* 2024. Vol. 448. P. 115884.

10. Bakhache B., Ghazal J. M., El Assad S. Improvement of the security of ZigBee by a new chaotic algorithm. *IEEE Syst. J.* 2014. Vol. 8. P. 1024–1033.

11. Boyraz O. F., Guleryuz E., Akgul A., Yildiz M. Z., Kiran H. E., Ahmad J. A Novel Security and Authentication Method for Infrared Medical Image with Discrete Time Chaotic System. *Optik*. 2022. Vol. 267. P. 1697171.

12. Lorenz E. N. Deterministic Nonperiodic Flow. *J. Atmos. Sci.* 1963. Vol. 20. P. 130–141.

13. Xia W., Liu B., Ren J., Mao Y., Wu X., Ullah R., Zhao L., Chen S., Wan Y., Ma Y. et al. High-security Transmission Scheme of Secure Key Generation and Distribution based on Polling-permutation Encryption. *J. Light. Technol.* 2023. Vol. 42. P. 149–157.

14. He D., Parthasarathy R., Li H., Geng Z. A Fast Image Encryption Algorithm based on Logistic Mapping and Hyperchaotic Lorenz System for Clear Text Correlation. *IEEE Access*. 2023. Vol. 11. P. 91441–91453.

15. Fu C., Huang J. B., Wang N. N., Hou Q. B., Lei W. M. A symmetric chaos-based image cipher with an improved bit-level permutation strategy. *J. Entropy*. 2014. Vol. 16. P. 770–788.

16. Wang X. Y., Zhang Y. Q., Bao X. M. A Colour Image Encryption Scheme using Permutation-Substitution Based on Chaos. *J. Entropy*. 2015. Vol. 17. P. 3877–3897.

17. Zhang Y. Q., Wang X. Y. A new image encryption algorithm based on non-adjacent coupled map lattices. *J. Appl. Soft Comput.* 2015. Vol. 26. P. 10–20.

18. Murillo-Escobar M. A., Cruz-Hernández C., Abundiz-Pérez F., López-Gutiérrez R. M., Del Campo O. A. A RGB image encryption algorithm based on total plain image characteristics and chaos. *J. Signal Process*. 2015. Vol. 10. P. 119–131.

19. Mollaeefar M., Sharif A., Nazari M. A novel encryption scheme for colored image based on high level chaotic maps. *J. Multimed. Tools Appl. Signal Process*. 2015. Vol. 76. P. 607–629.

20. Liu W., Sun K., Zhu C. A fast image encryption algorithm based on chaotic map. Opt. Lasers Eng. 2016. Vol. 84. P. 26–36.

21. Kiran H. E., Akgul A., Yildiz O. A New Chaos-Based Lightweight Encryption Mechanism for Microcomputers. *10th International Symposium on Digital Forensics and Security (ISDFS 2022)*. New York: Institute of Electrical and Electronics Engineers Inc., 2022.

22. Gafsi M., Amdouni R., Abbassi N., Hajjaji M. A., Mtibaa A. Implementation of a symmetric chaos-based cryptosystem for image security in real time. *2022 IEEE 9th International Conference on Sciences of Electronics, Technologies of Information and Telecommunications (SETIT 2022)*. New York: Institute of Electrical and Electronics Engineers Inc., 2022. P. 138–142.

23. Sisi T., Yaping W. A novel image encryption based on hyper-chaotic financial system. *2019 6th International Conference on Information Science and Control Engineering (ICISCE 2019)*. New York: Institute of Electrical and Electronics Engineers Inc., 2019. P. 95–99.

24. Pradhan B., Sengupta S. Chaotic-cipher based memory efficient symmetric key cryptosystem. *Proceedings of the 2018 Emerging Trends in Electronic Devices and Computational Techniques (EDCT)*, Kolkata, India, 8–9 March 2018. P. 1–3.

25. Lv W., Bai R., Sun X. Image Encryption Algorithm Based on Hyper-chaotic Lorenz Map and Compressed Sensing Theory. *Proceedings of the 2019 Chinese Control Conference (CCC)*, Guangzhou, China, 27–30 July 2019. P. 3405–3410.

26. Xing S., Wu C. Implementation of A Neuron Using Sigmoid Activation Function with CMOS. *Proceedings of the IEEE 5th International Conference on Integrated Circuits and Microsystems (ICICM)*, Nanjing, China, 23–25 October 2020. P. 201–204.

27. Zaki P. W., Hashem A. M., Fahim E. A., Mansour M. A., ElGenk S. M., Mashaly M., Ismail S. M. A Novel Sigmoid Function Approximation Suitable for

Neural Networks on FPGA. *Proceedings of the 15th International Computer Engineering Conference (ICENCO)*, Cairo, Egypt, 29–30 December 2019. P. 95–99.

28. Dela Cruz J. C., Centeno J. C. F., Faulve G. R., Pascasio G. J. R., Banlawe I. A. P. Classifying Adult Mango Pulp Weevil Activity using Support Vector Machine. *IEEE 12th International Conference*. Piscataway: IEEE, 2020. P. 116.

29. Що таке криптографія і як вона захищає транзакції. *PayPilot*. URL: https://www.paypilot.org/uk/sho-take-kriptografiya-i-yak-vona-zaxishae-tranzaktsii/ (дата звернення: 22.05.2026).

30. Криптографічні помилки, як уникнути компрометації даних. *HackYourMom*. URL: https://hackyourmom.com/osvita/kryptografichni-pomylky-yak-unyknuty-komprometacziyi-danyh/ (дата звернення: 22.05.2026).

31. Симетричне та асиметричне шифрування. *Binance Academy*. URL: https://www.binance.com/uk-UA/academy/articles/symmetric-vs-asymmetric-encryption (дата звернення: 22.05.2026).

32. Шкіль О., Рахліс Д., Філіпенко І., Корнієнко В. Проєктування та самодіагностика кіберфізичних пристроїв керування на платформі SoC. *Сучасний стан наукових досліджень та технологій в промисловості*. 2023. № 4 (26). С. 122–134. DOI: 10.30837/ITSSI.2023.26.122

33. FPGA.ua.URL: https://fpga.com.ua/index.php?route=product/product&product_id=16 (дата звернення: 23.05.2026).

34. Аврунін О.Г. Основи мови VDHL для проектування цифрових пристроїв на ПЛІС: навч. пос. / О. Г. Аврунін, Т. В. Носова, В. В. Семенець. – Харків : ХНУРЕ, 2018. – 196 с.

35. Аврунін О.Г. Основи мов SystemVerilog та VHDL для проєктування цифрових пристроїв на ПЛІС у прикладах і задачах: навч. посіб. / О. Г. Аврунін, Т. В. Носова, І. В. Прасол, В.В. Семенець, Є. А. Чугуй. – Електронне видання. – Харків: ХНУРЕ, 2025. – 383 с.

36. openssl-speed. *OpenSSL Documentation*. URL: https://docs.openssl.org/master/man1/openssl-speed/ (дата звернення: 24.05.2026).

*37.* Federal Information Processing Standards Publication 197: Announcing the Advanced Encryption Standard (AES). 2001. *26* November.

38. Zhang Y., Lu K., Gao Y., Wang M. NEQR: A novel enhanced quantum representation of digital images. *Quantum Inf. Process*. 2013. Vol. 12. P. 2833–2860.

39. Zhang J., Huang Z., Li X., Wu M., Wang X., Dong Y. Quantum image encryption based on quantum image decomposition. *Int. J. Theor. Phys*. 2021. Vol. 60. P. 2930–2942.

40. Li P., Zhao Y. A simple encryption algorithm for quantum color image. Int. *J. Theor. Phys.* 2017. Vol. 56. P. 1961–1982.

41. Gong L.-H., He X.-T., Cheng S., Hua T.-X., Zhou N.-R. Quantum image encryption algorithm based on quantum image XOR operations. *Int. J. Theor. Phys*. 2016. Vol. 55. P. 3234–3250.

42. Guo J., Li X., Lv Z., Yang Y., Li L. Design of real-time video transmission system for drone reliability. *Proc. IOP Conf. Ser. Mater. Sci. Eng.* 2020. Vol. 790. P. 012004.

43. Cheng Y. L., Liao Y. P., Chen C. Y., Huang T. W. Implementation of Quantum Image Encryption via Reversible Quantum Logic Gates Computing. *Spin*. 2023. Vol. 13. P. 2340020.

44. Zhang H.-F., Wang J., Cui K., Luo C.-L., Lin S.-Z., Zhou L., Liang H., Chen T.-Y., Chen K., Pan J.-W. A Real-Time QKD System Based on FPGA. *J. Light. Technol*. 2012. Vol. 30. P. 3226–3234.

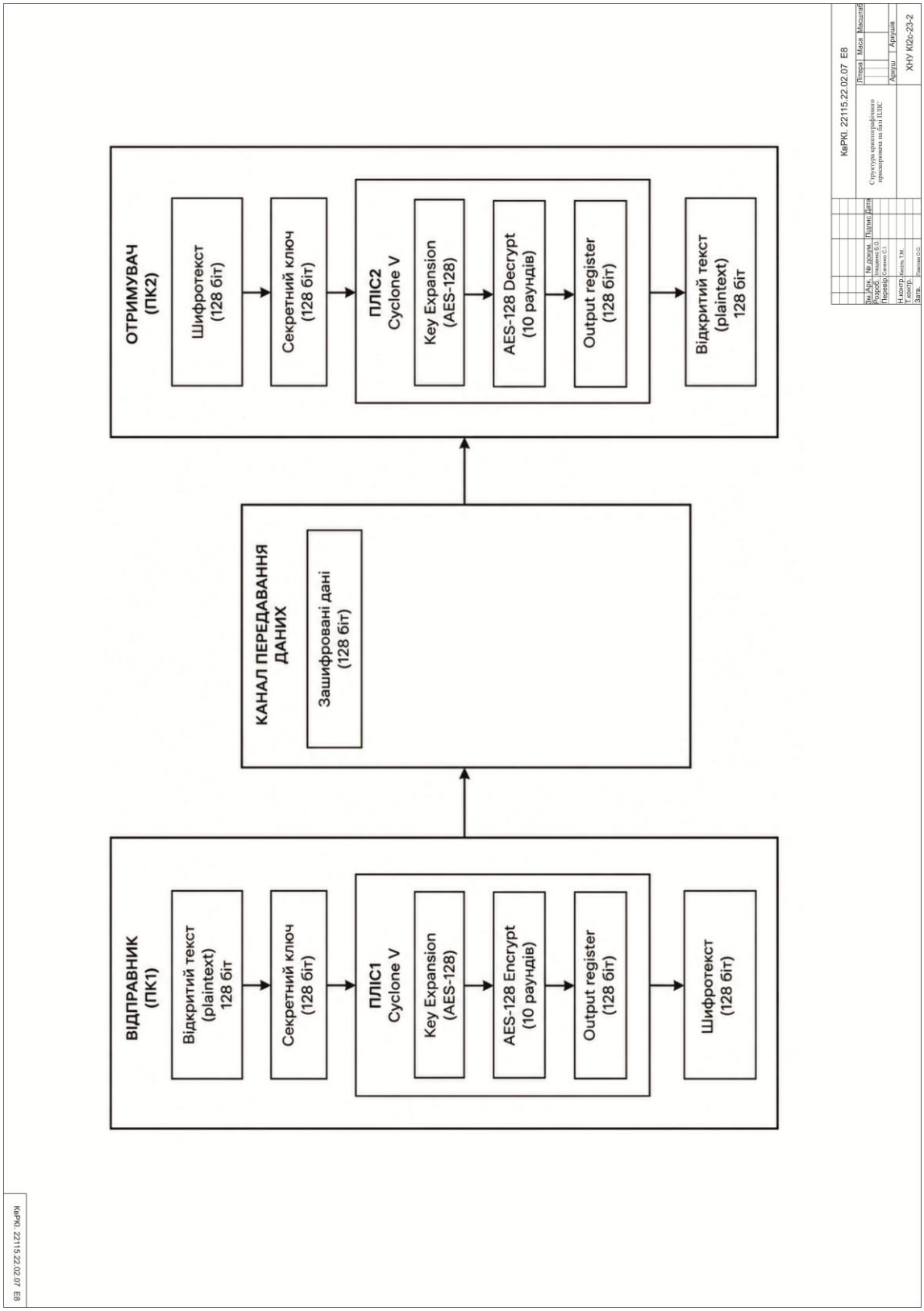
45. Gandelman S. P., Maslennikov A., Rozenman G. G. Hands-on quantum cryptography: Experimentation with the B92 protocol using pulsed lasers. *Photonics*. 2025. Vol. 12. P. 220.

46. Cong J., Fang Z., Lo M., Wang H., Xu J., Zhang S. Understanding performance differences of FPGAs and GPUs. *Proceedings of the 2018 IEEE 26th Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM)*, Boulder, CO, USA, 29 April – 1 May 2018. New York: IEEE, 2018. P. 93–96.

# ДОДАТОК А

(обов’язковий)

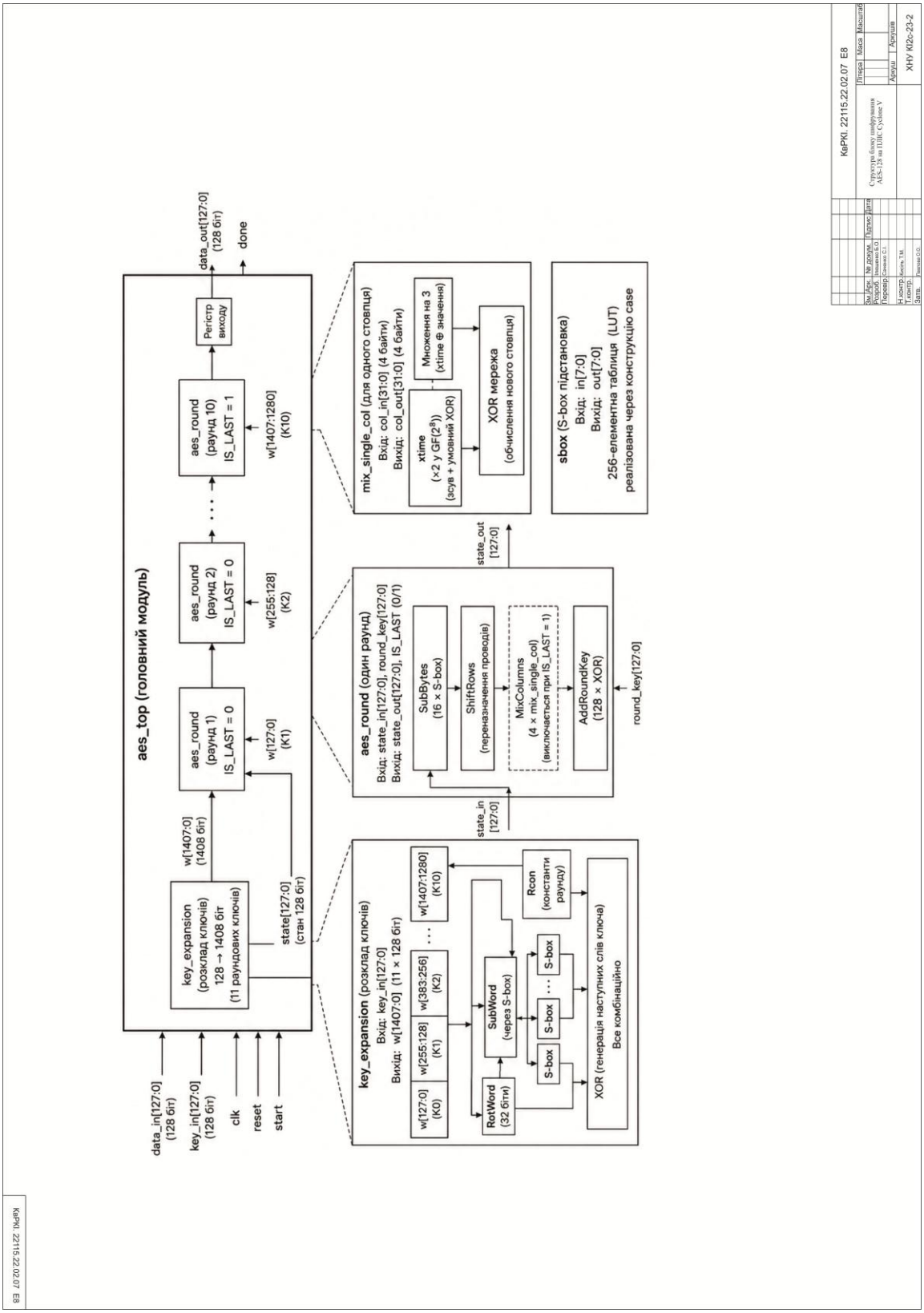
Копія креслення «Структура криптографічного прискорювача на базі ПЛІС»



# ДОДАТОК Б

(обов’язковий)

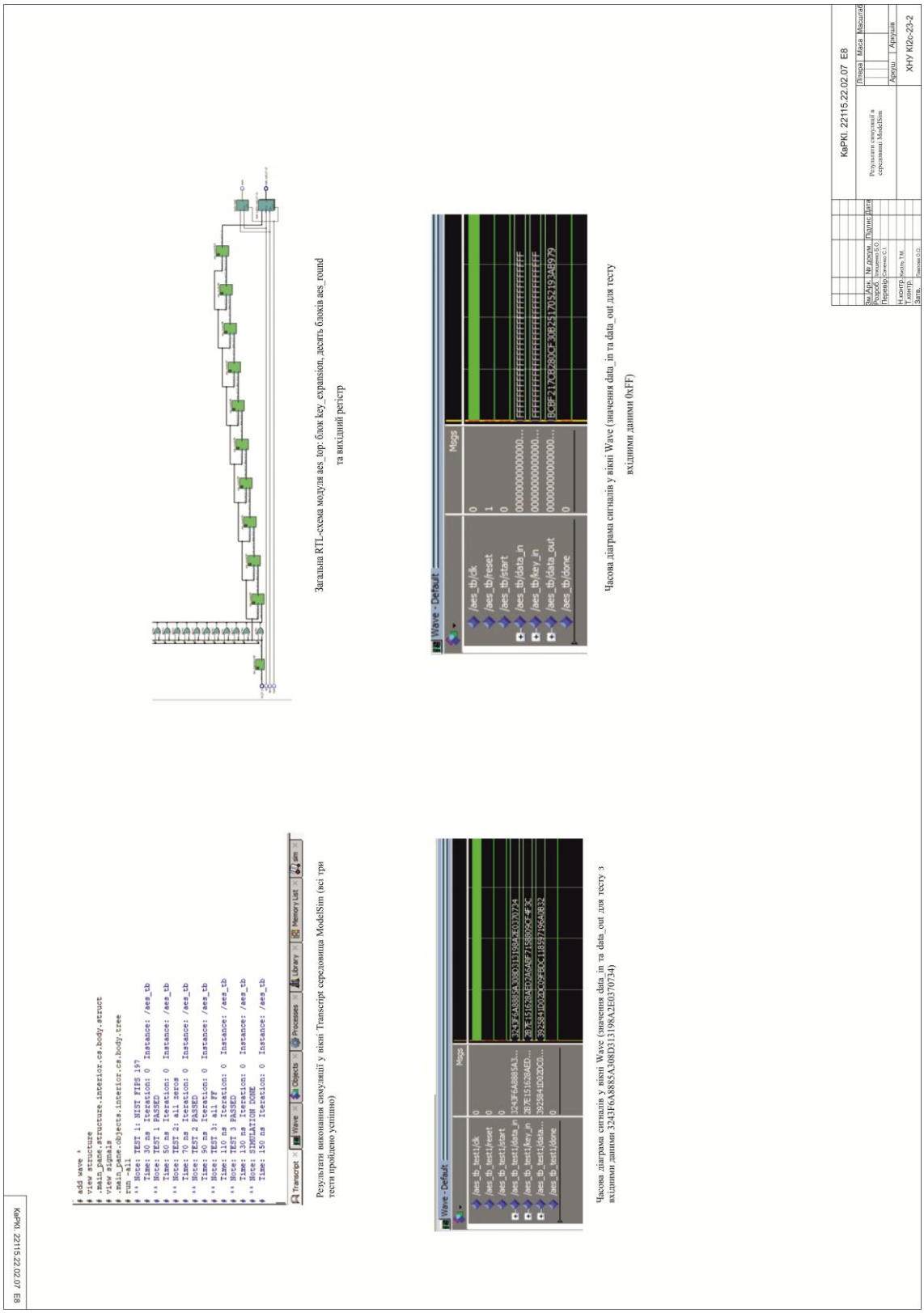
Копія креслення «Структура блоку шифрування AES-128 на ПЛІС Cyclone V»



# ДОДАТОК В

(обов’язковий)

Копія креслення «Результати симуляції в середовищі ModelSim»



Зав. кафедри КІІС
д-р. філософії Ользі ПАВЛОВІЙ

Богдан Ілюшенко
ПІБ здобувача вищої освіти

ФІТ, 3 курсу, групи КІ2с-23-2

ЗАЯВА

З правилами чинного Положення про систему забезпечення академічної доброчесності у Хмельницькому національному університеті, згідно з яким виявлення академічного плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту і застосування заходів академічної відповідальності, ознайомлений (а). Про використання спеціалізованих програмних засобів (СПЗ) StrikePlagiarism та Anti-Plagiarism для перевірки кваліфікаційних робіт здобувачів вищої освіти на наявність академічного плагіату оповіщений (а). Надаю університету право на передачу моєї роботи для обробки та збереження в базах даних СПЗ і використання роботи для виявлення академічного плагіату в інших роботах, які перевіряються СПЗ.

Також надаю свою згоду на обробку й збереження університетом моєї роботи в Інституційному репозитарії Хмельницького національного університету.

Робота надається для перевірки в електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

1 травня 2026 року

Wed May 27 10:10:03 EEST 2026, Медзатий Дмитро Миколайович, Хмельницький національний університет, ХНУ

# Anti-Plagiarism (http://ap.km.ua) v-15.701

**Максимальне співпадіння з одним документом 1.0%**

Словники перевірки: en_US, ru_RU, ua_UA. **Помилок в документах: 13%**

| | Документ | | Сумарний збіг по Базі Даних | |
|---|---|---|---|---|
| ID: 272409<br>Назва: БКР Криптографічний прискорювач на базі ПЛІС (FPGA) для захищеного передавання даних<br>Додано в БД: 2026-05-27<br>Автора: Богдан ІЛЮШЕНКО<br>Керівники: Світлана САЧЕНКО<br>Консультанти:<br>Опоненти: | Символи | Лексеми | Символи | Лексеми |
| | 92509 | 639 | 1127 (1%) | 13 (2%) |

Джерело плагіату

| ID | Опис | Наявність плагіату в документі | |
|---|---|---|---|
| | | Символи | Лексеми |

**Протокол аналізу звіту подібності експертом**

Заявляю, що я ознайомився (-лась) з Повним звітом подібності, який був згенерований Системою виявлення і запобігання плагіату щодо роботи:

**Автор:** Богдан ІЛЮШЕНКО

**Співавтор:**

**Назва:** Криптографічний прискорювач на базі ПЛІС (FPGA) для захищеного передавання даних

**Експерт:** Світлана САЧЕНКО

**Підрозділ:** Кафедра комп'ютерної інженерії та інформаційних систем

**Коефіцієнт подібності 1:**4.64%

**Коефіцієнт подібності 2:**1.79%

**Мікропробіли:** 14

**Заміна букв:** 0

**Інтервали:** 0

**Білі знаки:** 0

**Дата створення звіту:** 2026-05-27 02:53:58.0

**Після аналізу Звіту подібності констатую наступне:**

☒ **Запозичення, виявлені в роботі є законними і не є плагіатом. Рівень подібності не перевищує допустимої межі. Таким чином робота незалежна і приймається.**

☐ **Запозичення не є плагіатом, але перевищено граничне значення рівня подібностей. Таким чином робота повертається на доопрацювання.**

☐ **Виявлено запозичення і плагіат або навмисні текстові спотворення (маніпуляції), як передбачувані спроби укриття плагіату, які роблять роботу невідповідною вимогам законодавства (Ст. 32. ЗУ Про вищу освіту, пункт 3.1, Ст. 42. ЗУ Про освіту) та вимог НАЗЯВО (Критерій 5), а також кодексу етики і процедурам. Таким чином робота не приймається.**

Обгрунтування:

2026-05-27 Доцент Андрій Нічепорук

*Дата* експерт

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

# РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Дипломник: Богдан ЛЮШЕНКО

Тема: Криптографічний прискорювач на базі ПЛІС (FPGA) для захищеного передавання даних

Спеціальність: 123 «Комп'ютерна інженерія»

Обсяг кваліфікаційної роботи:

Кількість листів креслень ___3___ Кількість сторінок записки ___60___

1. Короткий зміст роботи та прийнятих рішень: Метою дипломної роботи є проєктування та реалізація криптографічного прискорювача на базі ПЛІС Cyclone V для захищеного передавання даних за допомогою алгоритму AES-128.

2. Висновок про відповідність роботи дипломному завданню: Робота повністю відповідає поставленому завданню.

3. Характеристика виконання кожного розділу, ступінь використання останніх досягнень науки і техніки і передових методів роботи: у першому розділі кваліфікаційної роботи проведено аналіз предметної області та огляд існуючих рішень у сфері апаратного криптографічного захисту даних, а саме: розглянуто роль та значення криптографічного захисту інформації у сучасних системах передавання даних; проаналізовано алгоритм симетричного блочного шифрування AES-128 відповідно до стандарту NIST FIPS 197; досліджено існуючі підходи до апаратної реалізації криптографічних алгоритмів на програмованих логічних інтегральних схемах; проведено порівняльний аналіз архітектурних підходів до реалізації AES на ПЛІС; сформовано висновки до першого розділу.

У другому розділі кваліфікаційної роботи виконано проєктування криптографічного прискорювача на базі ПЛІС Cyclone V для захищеного передавання даних, а саме: сформульовано перелік функціональних, апаратних та архітектурних вимог до прискорювача; розроблено загальну структурну схему системи захищеного передавання даних з визначенням місця прискорювача у ній; виконано модульну декомпозицію прискорювача на п'ять ієрархічних VHDL-модулів; обґрунтовано вибір

архітектури повного розгортання (fully unrolled); описано математичні основи операцій алгоритму AES-128 та їх апаратно-орієнтовані інтерпретації; визначено інтерфейс модуля та часову діаграму роботи; проаналізовано характеристики цільової платформи Cyclone V 5CSEMA5F31C6; сформовано висновки до другого розділу.

У третьому розділі кваліфікаційної роботи виконано реалізацію та симуляцію криптографічного прискорювача AES-128 на базі ПЛІС Cyclone V у середовищі Quartus II, а саме: налаштовано середовище розробки Quartus II та підключено симулятор ModelSim-Altera; реалізовано п'ять VHDL-модулів ієрархічної структури прискорювача мовою VHDL-93; розроблено тестовий сценарій з трьома тестовими векторами відповідно до стандарту NIST FIPS 197; проведено функціональну RTL-симуляцію у середовищі ModelSim з підтвердженням коректності всіх трьох тестових векторів зі статусом PASSED; виконано аналіз структури синтезованої схеми засобами RTL Viewer; проведено розрахунок та порівняльний аналіз пропускної здатності апаратної реалізації відносно програмних реалізацій AES на процесорах загального призначення; сформовано висновки до третього розділу.

5. Негативні сторони роботи: обмеженням роботи є відсутність фізичного тестування на реальному апаратному забезпеченні, оскільки верифікація виконувалась виключно методом RTL-симуляції без фізичного завантаження конфігурації у мікросхему Cyclone V та без підключення до реального каналу передавання даних.

6. Оцінка графічного оформлення та пояснювальної записки роботи: Пояснювальна записка оформлена коректно, згідно діючих стандартів оформлення документації.

7. Відгук про роботу в цілому: Робота виконана на задовільному технічному рівні.

8. Інші зауваження: _____

9. Оцінка дипломної роботи: задовільно D (70)

Рецензент (прізвище, ім'я, по батькові, посада, місце роботи) _____

Радюк Павло Михайлович, доцент кафедри КН ХНУ

"01" 06 2026 р. _____ (підпис)

РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ

КАФЕДРИ КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ
ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Назва кваліфікаційної роботи Криптографічний прискорювач на базі ПЛІС (FPGA) для захищеного передавання даних
Автор Богдан ЛЮШЕНКО
Освітня програма Комп'ютерна інженерія та програмування
Рівень вищої освіти перший (бакалаврський)
Спеціальність 123 Комп'ютерна інженерія
Науковий керівник: к.е.н., Світлана САЧЕНКО

На основі аналізу кваліфікаційної роботи на дотримання вимог академічної доброчесності (у т.ч. відсутності ознак академічного плагіату) з урахуванням результатів перевірки роботи спеціалізованим програмним засобом(ами) комісія зробила такий висновок:

| № | Висновок | Позначка про відповідність |
|---|---|---|
| 1 | Ознаки академічного плагіату | |
| 1.1 | Запозичення, виявлені в роботі, є законними і не є академічним плагіатом (далі – зазначаються підстави віднесення запозичень до правомірних, якщо потрібно). Робота приймається до захисту. | відповідає |
| 1.2 | Виявлені запозичення не є академічним плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи (далі – зазначаються детальні та аргументовані підстави віднесення запозичень до правомірних). Робота приймається до захисту, але має бути відкоригована. | |
| 1.3 | Виявлені запозичення не є академічним плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота може бути допущена до захисту після того як буде відкоригована та доопрацьована і успішно пройде повторну перевірку на академічний плагіат. | |
| 1.4 | Робота містить навмисні текстові спотворення, передбачувані спроби укриття текстових запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту. | |
| 2 | Інші види порушень академічної доброчесності | |

Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

1) усі запозичення фрагментарні, або мають належним чином оформленні посилання;

2) окремі виявлені збіги є загальновживаними фразами або виразами, про що свідчить посилання системи на збіг з джерелами на один фрагмент речення;

3) всі зафіксовані системою ознаки модифікації тексту відносяться до комбінування латинських символів зі україномовними скороченнями індексів в формулах, що не є модифікацією тексту.

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ ідентичності/схожості StrikePlagiarism, складає 4,64% і адресується до 28 першоджерела; та системою Anti-Plagiarism складає 1%, що, з урахуванням наведених обґрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

29.05.2026

Завідувач кафедри _____ Ольга ПАВЛОВА
Підпис Ім'я, ПРІЗВИЩЕ

Гарант освітньої програми _____ Андрій НІЧЕПОРУК
Підпис Ім'я, ПРІЗВИЩЕ

Керівник кваліфікаційної роботи _____ Світлана САЧЕНКО
Підпис Ім'я, ПРІЗВИЩЕ