Вінницький національний технічний університет
(повне найменування вищого навчального закладу)

Факультет інфокомунікацій,радіоелектроніки та наносистем
(повне найменування інституту, назва факультету (відділення))

Кафедра радіотехніки
(повна назва кафедри (предметної, циклової комісії))

Пояснювальна записка
до магістерської кваліфікаційної роботи

«Магістр»
(освітньо-кваліфікаційний рівень)

на тему: **«Розробка апаратної платформи для реалізації SHA-алгоритмів на FPGA»**

Виконав: студент 2-го курсу, групи РТ-19м
спеціальності 172 – Телекомунікації та
радіотехніка_Освітня програма - Радіотехніка_
(шифр і назва напряму підготовки, спеціальності)
Прокопчук С.С.
(прізвище та ініціали)

Керівник: к.т.н., доцент каф. РТ
Воловик А.Ю.
(прізвище та ініціали)
«____» _____ 2020 р.

Рецензент: к.т.н., доцент каф. ТКСТБ
Васильківський М.В.
(прізвище та ініціали)
«____» _____ 2020 р.

Вінниця ВНТУ - 2020 рік

Вінницький національний технічний університет
Факультет Інфокомунікацій , радіоелектроніки та наносистем
Кафедра Радіотехніки
Освітньо-кваліфікаційний рівень Магістр
Спеціальність 172 – Телекомунікації та радіотехніка
(шифр і назва)

ЗАТВЕРДЖУЮ
Завідувач кафедри РТ
д.т.н., професор О.В. Осадчук
“___” _____20__ року

З А В Д А Н Н Я
НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Прокопчуку Сергію Сергійовичу
(прізвище, ім’я, по батькові)

1. Тема роботи «Розробка апаратної платформи для реалізації SHA-алгоритмів на FPGA»
керівник роботи Воловик Андрій Юрійович, к.т.н., доцент
(прізвище, ім’я, по батькові, науковий ступінь, вчене звання)
затверджені наказом вищого навчального закладу від “25” 09 2020 року №214
2. Строк подання студентом роботи 10.12.2020 року.
3. Вихідні дані: Кількість системних вентилів: до 50 тисяч; Кількість логічних комірок: до 1047; Сімейство логіки: КМОН; Об'єм вбудованого EEPROM (ЕСППЗП): до 4 Мбіт; Вбудований інтерфейс для відладки/завантаження ПЛІС на основі шини USB; Лінії введення/виведення загального призначення: 32 цифрових лінії введення/виведення, максимальна напруга 3.3 В, максимальний струм 8 мА; Джерело живлення постійного струму : 15 В постійного струму, 650 мА; Загальна споживана потужність: 6 Вт макс.
4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) вступ; техніко-економічне обґрунтування. Огляд засобів проектування. Методи захисту ПЗ ПЛІС. Розробка модуля криптозахисту ПЗ ПЛІС. Опис перевірки працездатності і результати симуляції. Цивільна оборона. Безпека життєдіяльності. Організаційно-економічний розділ. Висновки.
5.Перелік графічного матеріалу (з точним зазначенням обов’язкових креслень) Структурна схема налагоджувальної плати. Етапи проектування пристрою в середовищі розробки WebPack ISE., Реалізація концепції IFF. Структура алгоритму шифрування. Опис роботи модуля аутентифікації. Опис перевірки працездатності і результати симуляції. Лістинги програми.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Основна частина	к.т.н., доцент Воловик А.Ю.		
Охорона праці та безпека в надзвичайних ситуаціях	к.т.н., доцент Березюк О. В.		
Економічна частина	к.е.н., ст. викл. Кавецький В.В.		

7. Дата видачі завдання 29.09.2020 року

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів магістерської кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	Огляд літературних джерел. Вибір та узгодження теми МКР	03.09.2020-14.09.2020	
2.	Аналіз літературних джерел. Попередня розробка основних розділів	15.09.2020-21.09.2020	
3.	Затвердження теми. Розробка технічного завдання	21.09.2020-25.09.2020	
4.	Аналіз вирішення поставленої задачі. Розробка структурної схеми	26.09.2020-09.10.2020	
5.	Електричні розрахунки. Експериментальне дослідження	10.10.2020-25.10.2020	
6.	Розділ моделювання	26.10.2020-04.11.2020	
7.	Розробка графічної частини МКР	05.11.2020-15.11.2020	
8.	Аналіз економічної ефективності розробки	16.11.2020-19.11.2020	
9.	Охорона праці (ОП)	19.11.2020-22.11.2020	
10	Оформлення пояснювальної записки та графічної частини	23.11.2020-29.11.2020	
11.	Нормоконтроль	30.11.2020-01.12.2020	
12.	Попередній захист МКР, доопрацювання, рецензування МКР	02.12.2020-04.12.2020	

Студент _____ Прокопчук С.С.
(підпис)

Керівник роботи _____ Воловик А.Ю.
(підпис)

РЕФЕРАТ

УДК 621.396

Прокопчук С.С. Розробка апаратної платформи для реалізації SHA-алгоритмів на FPGA / Магістерська кваліфікаційна робота / Вінниця: ВНТУ, 2020, – 136 с. Укр. мовою.

Бібліограф. найменувань 24, ілюстрацій 41, таблиць 37.

Метою **магістерської кваліфікаційної роботи** є конструкторсько-технологічне проектування апаратної платформи для реалізації SHA-алгоритмів на FPGA, із метою подальшого виготовлення.

Методи дослідження - застосування сучасних САПР, порівняльний аналіз, використання існуючих баз даних для створення конструкції приладу.

Результатом розробки є частина технічної документації, необхідна для виготовлення та експлуатації вимірювальних приладів.

В **магістерській кваліфікаційній роботі** наводиться необхідний матеріал для підтвердження актуальності розробки, наводиться принцип дії із висвітленням наукової новизни, обраховані основні електричні, конструктивні та технологічні характеристики виробу, а також електромагнітна сумісність та захист. Наведене обґрунтування загального конструкторського рішення, обґрунтування вибору комплектуючих та матеріалів.

Описано технологію виготовлення друкованої плати та порядок складання пристрою. Проведено техніко-економічне обґрунтування доцільності розробки. Розраховано економічний ефект від розробки та впровадження пристрою. Розглянуті питання безпеки життєдіяльності під час виготовлення пристрою та стійкості його роботи при дії електромагнітного імпульсу та іонізуючого випромінювання.

Загалом запланована науково-дослідна робота з проведення дослідження з розробки апаратної платформи для реалізації SHA-алгоритмів на FPGA вимагає грошового вкладення для виконання в межах 334495,59 грн.

Отримані результати досліджень мають високий рівень наукової значимості (в межах 0,70), що свідчить про доцільність проведення розробок та значимість науково-дослідної роботи з дослідження з розробки апаратної платформи для реалізації SHA-алгоритмів на FPGA

Ключові слова: алгоритми шифрування, FPGA, Xilinx, Intel, Altera.

ABSTRACT

Prokopchuk S.S. Development of a hardware platform for the implementation of SHA-algorithms on FPGA / Master's thesis / Vinnytsia: VNTU, 2020, - 136 p. Ukr. language.

Bibliographer. names 24, illustrations 41, tables 37.

The purpose of the master's qualification work is the design and technological design of the hardware platform for the implementation of SHA-algorithms on the FPGA, with the aim of further manufacturing.

Research methods - the use of modern CAD, comparative analysis, the use of existing databases to create the design of the device.

The result of the development is part of the technical documentation required for the manufacture and operation of measuring instruments.

The master's qualification work provides the necessary material to confirm the relevance of the development, provides the principle of operation with coverage of scientific novelty, calculated the basic electrical, structural and technological characteristics of the product, as well as electromagnetic compatibility and protection. The substantiation of the general design decision, the substantiation of a choice of accessories and materials is resulted.

The technology of PCB manufacturing and the order of assembly of the device are described. Feasibility study of development expediency is carried out. The economic effect of the development and implementation of the device is calculated. The issues of life safety during the manufacture of the device and the stability of its operation under the action of electromagnetic pulse and ionizing radiation are considered.

The technology of PCB manufacturing and the order of assembly of the device are described. Feasibility study of development expediency is carried out. The economic effect of the development and implementation of the device is calculated. The issues of life safety during the manufacture of the device and the stability of its operation under the action of electromagnetic pulse and ionizing radiation are considered.

In general, the planned research work on conducting research on the development of a hardware platform for the implementation of SHA-algorithms on the FPGA requires a monetary investment to perform within 334495.59 UAH.

The obtained research results have a high level of scientific significance (within 0.70), which indicates the feasibility of development and the importance of

research work on research to develop a hardware platform for the implementation of SHA-algorithms on FPGA.

Keywords: encryption algorithms, FPGA, Xilinx, Intel, Altera.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

PLD – Programmable logic device (програмовані логічні пристрої)

ПЛІС – Програмовані логічні інтегральні схеми

CPLD – Complex Programmable Logic Device (комплексні програмовані логічні пристрої)

FPGA – Field Programmable Gate Array (програмовані вентильні матриці)

SHA – Secure Hash Algorithm

DSP – Digital signal processor (Цифровий сигнальний процесор)

ЦОС – Цифрова обробка сигналів

ASIC – application–specific integrated circuit (інтегральна схема спеціального призначення)

САПР – Система автоматизованого проектування

GAL – Gate Array Logic(базова матрична логіка)

ПМЛ – Програмована матрична логіка

VHDL – VHSIC (Very high speed integrated circuits) Hardware Description Language (мова опису апаратури)

Verilog – Verilog HDL (Verilog Hardware Description Language) - мова опису апаратури

ISE – Integrated Synthesis Environment (інтегроване середовище розробки)

JTAG – Joint Test Action Group (робоча група по розробці стандарту IEEE 1149)

SHA – Advanced Encryption Standard (вдосконалений стандарт шифрування) - симетричний алгоритм блокового шифрування

DES – Data Encryption Standard (стандарт шифрування даних) симетричний алгоритм шифрування

ВСТУП

Актуальність теми. Нині одним з напрямів, що інтенсивно розвиваються, в мікроелектроніці є PLD (programmable logic device - програмовані логічні пристрої). PLD є порожнім чіпом. На відміну від звичайних цифрових мікросхем, логіка роботи ПЛІС визначається не на фабриці при виготовленні, а задається за допомогою додаткового програмування (проектування) (field - programmable, від англійської абревіатури FPGA – "Field Programmable Gate Array") за допомогою спеціальних засобів: програматорів і програмного забезпечення. Ця технологія дозволяє розробити свою мікросхему з власною архітектурою.

ПЛІС — високоінтегровані гнучкі універсальні логічні пристрої з внутрісистемним перепрограмуванням. Збільшення попиту на пристрої з можливістю зміни їх внутрішньої структури в реальному часі зі швидкою перебудовою виконуваних функцій розширило сфери застосування ПЛІС. Доцільність її застосування обумовлюється необхідністю або розробити оригінальну апаратуру, або замінити звичайну ІС, що дозволяє зменшити розміри пристрою, понизити споживану потужність і збільшити надійність. Найбільш ефективне використання ПЛІС у виробах, що вимагають нестандартних рішень схемотехніки. У цих випадках ПЛІС навіть середній мірі інтеграції (24 виведення) замінює, як правило, до 10-15 звичайних ІС [1].

ПЛІС широко використовуються в мікропроцесорній техніці. На її основі можна розробляти цифрові фільтри, дешифратори, логіку обрамлення мікропроцесорів, формувачі сигналів, що управляють.

Переваги і сфери застосування ПЛІС:

1. Реалізація цифрових (кінцевих) автоматів (state machine). ПЛІС природним чином вписується в довільний синхронний автомат. Тригерна матриця і дискретна комбінаційна логіка поступається в ціні, вартості проектування і відладки і у розмірі кінцевого продукту;

2. Гнучкість. Оскільки ПЛІС - програмована схема, вона має важливу

перевагу: на етапі проектування підсумкової схеми є можливість перепрограмувати свій продукт без втрати часу і засобів, у разі, якщо припустилася помилки або необхідно щось модифікувати. Це корисно, коли розробник ще не до кінця продумав, як працюватиме його схема у результаті, що дасть можливість застосувати різні ідеї [2].

У зв'язку з широким використанням ПЛІС на світовому ринку електроніки важливе місце займає захист інтелектуальної власності.

Продукти інтелектуальної діяльності мають вартісні оцінки, оскільки вони можуть бути включені в товарообіг на комерційних умовах. Загрозу представляє крадіжка інтелектуальної власності, наслідками якої є економічні втрати. Запобігти їй можна захистивши цифрові електронні пристрої.

Поширення нелегального копіювання програмного продукту, висока вартість розробки ПЗ (сучасні складні програми пишуться командою програмістів), можливість витягання інформації з постійних пристроїв, що запам'ятовують, ПЛІС безперешкодно, наявність недобросовісної конкуренції на ринку електроніки примушує замислитися про захист інтелектуальної власності.

Ця робота присвячена захисту програмного забезпечення ПЛІС, грунтованою на криптографічних методах.

Метою роботи є створення та дослідження апаратної платформи для реалізації SHA-алгоритмів на FPGA.

Об’єктом дослідження є розробка теоретичних засад, методів та засобів для шифрування програмного забезпечення для ПЛІС.

Предметом дослідження – є платформи для реалізації SHA-алгоритмів на FPGA.

В магістерській кваліфікаційній роботі для досягнення поставленої мети **розв’язуються такі завдання**:

1. Проведення аналізу сучасного стану розробки апаратної платформи для реалізації SHA-алгоритмів на FPGA та виявлено базові переваги та недоліки приладів, що вже існують.

2. Попередній розрахунок структурної схеми апаратної платформи для реалізації SHA-алгоритмів на FPGA з функцією передачі даних, що включає в себе розробку таких блоків та вузлів: блок циклічного запису даних з модуля дешифрування , модуль пошуку серійного номера, генератор випадкових чисел.

3. Розрахунок каскадів апаратної платформи для реалізації SHA-алгоритмів на FPGA, а саме: модель порушника, вбудовані методи захисту шифруванням бітового потоку, зберігання ключів, ОЗП з живленням від батареї, eFUSE- регістри, атака на вбудовані методи захисту, метод захисту за допомогою вбудовування дешифратора, атака на метод захисту, що використовує вбудовуваний дешифратор, використання мікросхем спеціальної пам'яті, атака на метод захисту, що використовує мікросхему спеціальної пам'яті, проблема декомпіляції конфігураційного файлу, концепція розробленого модуля криптозахисту, опис вибраного криптоалгоритму шифрування.

4. Проведення експериментальних досліджень блоку циклічного запису даних з модуля дешифрування, модуля пошуку серійного номера, генератора випадкових чисел за допомогою системи автоматизованого проектування показало, що результати моделювання відповідають умовам технічного завдання та принципу роботи пристрою в цілому.

5. Розробка блоків апаратної платформи для реалізації SHA-алгоритмів на FPGA з застосуванням нової елементної бази, дозволить підвищити точність та діапазон роботи пристрою, при покращенні масо-габаритних властивостей.

Методи дослідження ґрунтуються на використанні:

а) основних положень теорії функції комплексної змінної (створення математичних моделей);

б) диференціального та інтегрального числення (створення математичних моделей);

в) методів розрахунку лінійних електричних кіл з використанням матриць (електричні розрахунки);

г) ЕОМ для обрахунків та проведення моделювання.

Наукова новизна одержаних результатів

Наукова новизна роботи полягає в отриманні таких результатів:

1. Запропоновано новий підхід при побудові апаратної платформи для реалізації SHA-алгоритмів на FPGA з застосуванням спеціалізованих мікросхем.

2. За рахунок використання сучасної елементної бази вдалося вдосконалити апаратну платформу для реалізації SHA-алгоритмів на FPGA.

3. У порівнянні з іншими серійними апаратними платформами для реалізації SHA-алгоритмів на FPGA, які побудовані на іншій елементній базі, пристрій побудований з застосуванням швидкодіючих цифрових пристроїв, що дає змогу добитися високої надійності та більшої стабільності процесі роботи.

4. Удосконалено математичні моделі елементів схеми, що застосовуються при моделювання у САПР, які, на відміну від існуючих, враховують зміни температури та стабільності напруги живлення.

5. Проведені експериментальні дослідження підтвердили вірність електричних розрахунків схем пристрою з похибкою, що відповідає вимогам технічного завдання.

Практичне значення отриманих результатів полягає у тому, що розробка електронно-радіовимірювальної апаратури є угрупування певної частини приладів в комплекси з максимально можливим числом загальних блоків, наприклад блоку живлення, відлікового пристрою, блоку обробки інформації і т.п. У вимірювальних комплексах можна застосовувати як аналогову, так і цифрову обробку і виведення інформації. Цифрові вимірювальні прилади до недавнього часу не мали широкого розповсюдження через складність виготовлення, вартості, габаритних розмірів і маси. Застосування сучасної елементної бази, що включає

мікросхеми середнього і великого ступеня інтеграції, а особливо FPGA дозволяє розробленим приладам за вартістю, габаритними розмірами і масою набагато перевищити аналогові аналоги.

Особистий внесок. Усі розрахунки, вимірювання, моделювання та конструювання були проведені самостійно. При розробці програмного інтерфейсу було використано нові підходи для реалізації продукту. Також застосовувались нові ідеї та рішення для покращення параметрів апаратної платформи для реалізації SHA алгоритмів на FPGA в цілому.

1 ОГЛЯД ЗАСОБІВ ПРОЕКТУВАННЯ

1.1 Загальні відомості про архітектуру FPGA

FPGA (Field Programmable Gate Array) - програмовані по простору вентильні матриці, що сполучаються в необхідну електричну схему за допомогою кіл міжз'єднань, розміщених між логічними блоками. Програмування мікросхеми з такою архітектурою зводиться до внесення змін до логіки роботи принципової схеми, ці зміни можуть бути застосовані у будь-який час. Мікросхеми складаються з логічних блоків, що конфігуруються, з множиною входів і одним виходом (логічні вентилі). У цифрових схемах такі схеми реалізують базові двійкові операції AND, NAND, OR, NOR і XOR [3].

ПЛІС з архітектурою FPGA включає три типи програмованих елементів : логічні блоки(логічні елементи LE), блоки введення/виведення і програмовані ключі, що виконують функції внутрішніх зв'язків між блоками.

Сучасні FPGA енергозалежні. Конфігурація FPGA зберігається в енергозалежній пам'яті, внаслідок чого при відключенні живлення їх конфігурація зникає, тому виникає необхідність в спеціальній flesh – пам'яті, яка знаходиться безпосередньо на платі з ПЛІС, вона і містить той конфігураційних файл, який завантажується у вентильну матрицю при включенні живлення.

На ринку цифрової електроніки великою популярністю користуються ASIC (Application – Specific Integrated Circuit). ASIC – замовлені мікросхеми, їх можна назвати аналогами ПЛІС, але з деякими обмовками.

Порівняльна характеристика ПЛІС і ASIC [4]:

1. Вартість. При дрібносерійному або середньосерійному виробництві вигідніше купувати ПЛІС, чим ASIC, також те, що припускається помилки в описі проектованого пристрою приведе до збитків порівнянним з вартістю виробництва цілої партії. У разі, коли планується великосерійне виробництво,

попри те, що використання ASIC може виявитися дешевше, не можна унеможливлювати великі витрати на проектування і відладку мікросхем(будь-яка модифікація ASIC на виробництві спричиняє за собою великі витрати на роботи по створенню шаблонів);

2. Зміна прошивки. Великою гідністю ПЛІС є можливість модифікації пристрою на етапі його проектування, виробництві, а також подальшого обслуговування, наприклад, оновлення прошивки. ASIC, у свою чергу, після запуску у виробництво не має можливості « безболісної» модифікації, також відсутня можливість оновити пристрій у клієнта, окрім як поміняти саму мікросхему, що рідко представляється здійснимим;

Лідерами в області створення ПЛІС-мікросхем є дві компанії: Xilinx і Altera. Ці компанії є основними конкурентами, працюючими в цьому напрямі. Кожна з них пропонує ряд різних серій FPGA- чіпів. У Xilinx – це Virtex(високопродуктивна, призначена для вирішення ресурсоємних завдань), Spartan(менш продуктивна і дешевша, призначена для використання в облаштуваннях великосерійного виробництва з недорогими комплектуючими), у Altera – Stratix (високопродуктивна мікросхема), Cyclone (для менш ресурсоємних завдань) і Arria (як компроміс між виробництвом і вартістю)[5].

В якості програмного і апаратного забезпечення, необхідного для отримання макетного варіанту, фірма Xilinx випускає макетну плату на базі ПЛІС, САПР і програматор. Кожна компанія — виробник ПЛІС розробляє і випускає свою САПР, реалізацію усіх етапів проектування, що забезпечує, для кожного типу програмованої логіки. САПР включає синтез і моделювання на HDL, розміщення і трасування проектів, а також програмування кристалів по JTAG — інтерфейсу. ISE WebPACK забезпечує просту в освоєнні і повнофункціональне середовище проектування і знаходиться у безкоштовному доступі [6].

Найширше в цьому середовищі проектування використовуються такі мови опису апаратури, як VHDL і Verilog.

1.2 Використовувані технології

Для опису апаратури існує багато мов, але явними лідерами у використанні є— VHDL і Verilog. Синтаксис мови Verilog схожий з синтаксисом мови З, що робить його простіше освоюваним. Мова проектування VHDL в сучасних обчислювальних системах є базовою мовою, він має ширший набір базових типів даних, а також дозволяє створювати власні складні типи.

VHDL (англ. VHSIC (Very high speed integrated circuits) Hardware Description Language) — мова опису апаратури інтегральних схем, розроблений в 1983 р. за замовленням Пентагону з метою формального опису логічних схем для усіх етапів розробки електронних систем. Мова призначена для моделювання переважно на вентильному рівні, рівні регістрових передач і корпусів мікросхем, а також успішно використовується і при синтезі пристроїв.

VHDL підтримує три різні стилі для опису апаратної архітектури :

– Структурний опис (structural description) - стиль, в якому архітектура описується, як ієрархія пов'язаних компонентів;

– Потоковий опис (data - flow description) - стиль, в якому архітектура описується, як безліч паралельних регістрових операцій, які, у свою чергу, управляються вентильними сигналами;

– Поведінковий опис(behavioral description) - алгоритмічний стиль, в якому перетворення описується послідовними програмними інструкціями, подібними тим, що використовуються в будь-яких сучасних високорівневих мовах;

Мова VHDL використовується у багатьох системах для моделювання цифрових схем, проектування програмованих логічних інтегральних мікросхем, базових матричних кристалів, замовлених інтегральних мікросхем[7].

1.3 Технічні характеристики налагоджувальної плати

Налагоджувальна плата NI Digital Electronics FPGA Board - це інструмент розробки електронних схем, побудований на основі мікросхеми ПЛІС XC3S500E Xilinx Spartan - 3E. Окрім ПЛІС, налагоджувальна плата містить движкові перемикачі, кнопки, світлодіоди, семисегментний індикатор з двома знакомісцями, а також нажимну поворотну кнопку для вибору зовнішнього синхросигналу зі світлодіодами для відображення вибраного режиму роботи, роз'єми DiglentPmod для підключення зовнішніх пристроїв, інтерфейс USB, призначений для програмування мікросхеми ПЛІС.

Налагоджувальна плата NI Digital Electronics FPGA Board може бути використана в автономному режимі в якості незалежного пристрою або як додаткова макетна плата для NI ELVIS. Для роботи в автономному режимі налагоджувальній платі потрібно персональний комп'ютер для завантаження нової програми, що управляє, і контролю за виконанням програми. З'єднання здійснюється за допомогою кабелю, що підключається до роз'єму USB.

Технічні характеристики ПЛІС: Xilinx XC3S500E - 4FTG256C[8]:

– Кількість системних вентилів : 500 тисяч;

– Кількість логічних комірок : 10476;

– Сімейство логіки : КМОН;

– Об'єм вбудованого EEPROM: 4 Мбіт;

– Вбудований інтерфейс для відладки/завантаження ПЛІС на основі шини USB;

– Лінії введення/виведення загального призначення: 32 цифрових лінії введення/виведення, максимальна напруга 3.3 В, максимальний струм 8 мА;

– Джерело живлення постійного струму : 15 В постійного струму, 650 мА;

– Загальна споживана потужність: 6 Вт макс;

На рис.1.1 показана верхня сторона налагоджувальної плати.

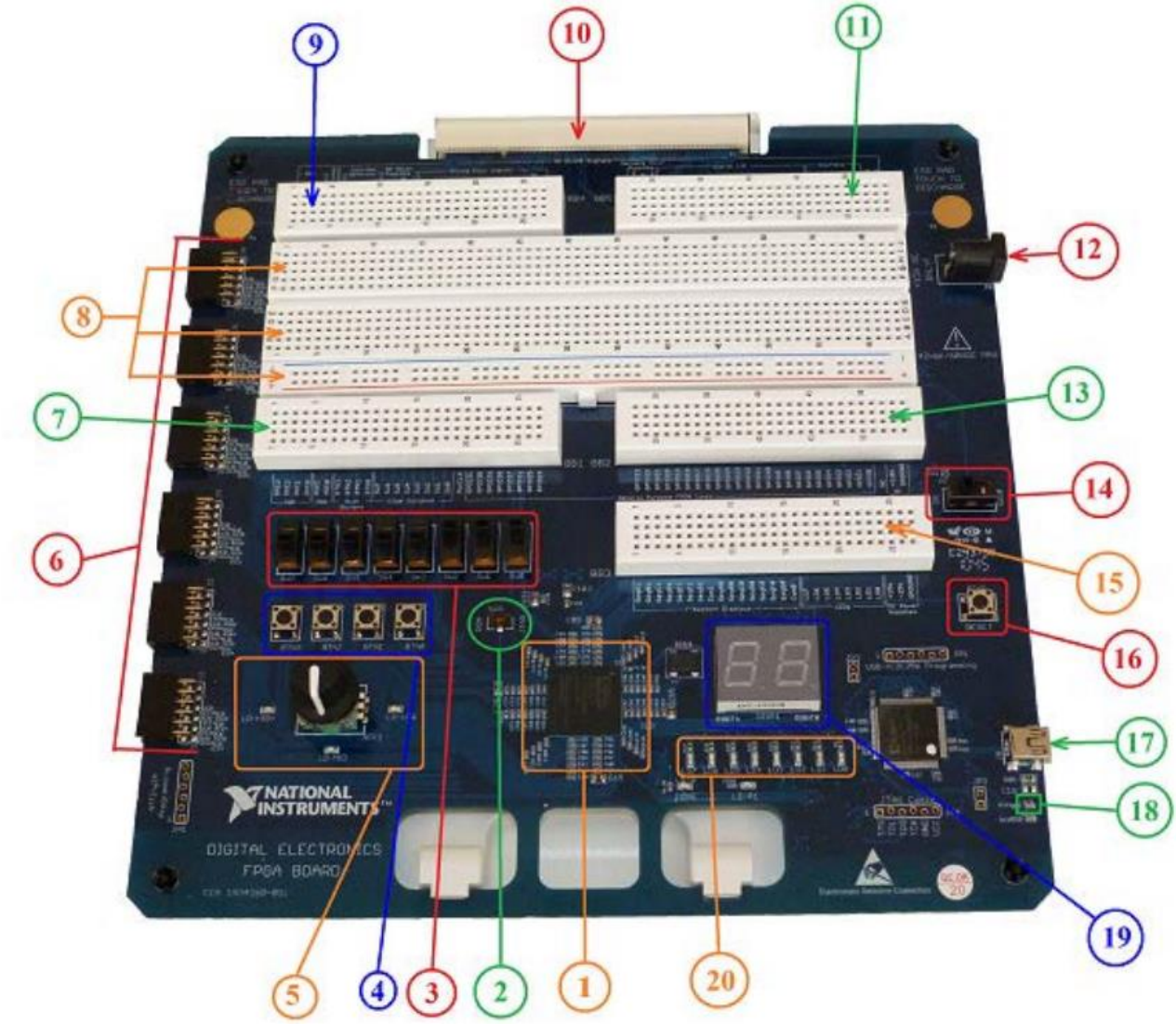


Рисунок 1.1 – Налагоджувальна плата NI Digital Electronics FPGA Board

1. ПЛІС XC3S500E Xilinx Spartan - 3E
2. Перемикач SW9
3. Рухомі перемикачі (SW0 - SW7)
4. Кнопки (BTN0 - BTN3)
5. Натискний перемикач
6. Роз'єми Digilent Pmod
7. Зона макетування: сигнальний роз'єм BB1
8. Роз'єми загального призначення
9. Зона макетування: сигнальний роз'єм BB4

10. Роз'єм для підключення до N1 ELVIS II+.
11. Зона макетування: сигнальний роз'єм ВВ5
12. Роз'єм підключення джерела живлення
13. Зона макетування: сигнальний роз'єм ВВ2
14. Вимикач живлення
15. Зона макетування: сигнальний роз'єм ВВ3
16. Кнопка скидання (Reset)
17. Роз'єм USB
18. Світлодіод LD - G
19. Семисегментні індикатори
20. Світлодіоди (LD0 - LD7)

Програмування пристроїв, що входять до складу налагоджувальної плати, можливо тільки з використанням інструментів інтегрованого середовища розробки.

1.4 Основні характеристики пакету WebPack ISE

Інтегрований пакет засобів розробки WebPACK ISE є системою, що реалізовує усі етапи створення цифрового пристрою на базі ПЛІС, включаючи розробку проекту, синтез, моделювання, трасування і завантаження в кристал.

Керуюча оболонка пакету WebPACK ISE Project Navigator™ (навігатор проекту - зручний інтерфейс для роботи з проектом і управління усіма етапами проектування і програмування ПЛІС. Усі програмні модулі пакету, що вимагаються, запускаються безпосередньо в середовищі Project Navigator [9].

Початкові дані про пристрій, що розробляється, можуть бути представлені у вигляді RTL- схем, описів на мові HDL, діаграм станів і бібліотек користувача. При синтезі на підставі початкових модулів проекту визначається список ланцюгів, який надалі використовується як початкові дані засобами трасування.

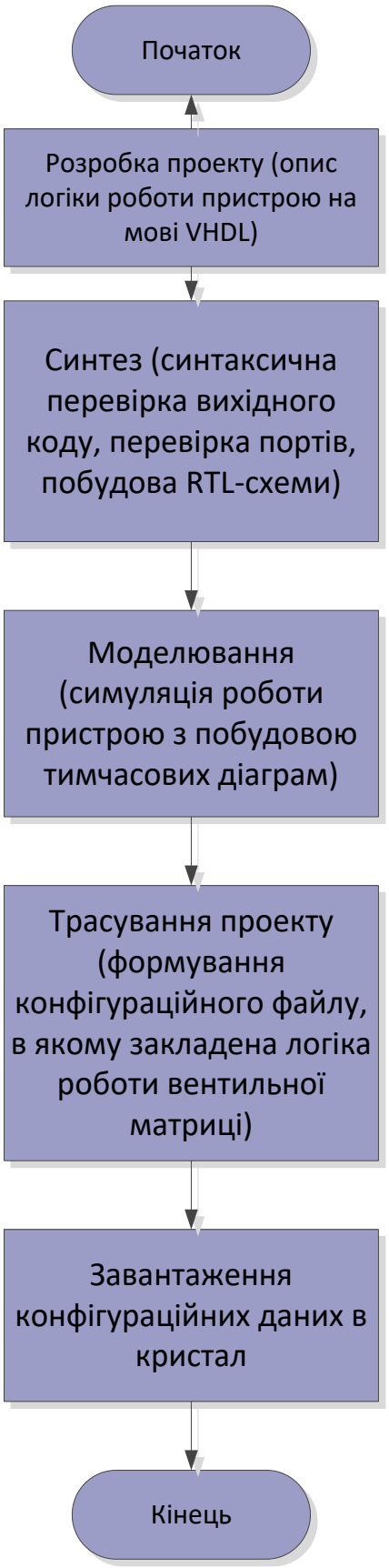


Рисунок 1.2 – Етапи проектування пристрою в середовищі розробки WebPack ISE

Кінцевий етап трасування проекту - конфігураційний файл, в якому закладена логіка роботи вентильної матриці. Останнім етапом проектування цифрового пристрою є завантаження конфігураційних даних в кристал за допомогою відповідних програмних модулів пакету WebPACK ISE і завантажувального кабелю.

Усі етапи проектування пристрою відбиті на рис. 1.2

Моделювання дозволяє уникнути безлічі можливих помилок на ранніх етапах проектування, істотно зменшивши загальний час розробки пристрою.

1.5 Оцінка наукового, технічного та економічного рівня НДДКР

Проаналізуємо рівень науково-дослідної роботи яка пов’язана з дослідженням з розробки апаратної платформи для реалізації SHA-алгоритмів на FPGA. Виходячи з відповідних вимог НТП, доцільно орієнтуватися на час проведення НДДКР 3 роки (+1), при чому технічні показники результатів плануються на вище рівня кращих світових зразків (+2); наявність можливості отримання авторських свідоцтв на винахід - впевненість в отриманні авторських свідоцтв (+2); а строк окупності витрат 2 роки і менше (+3).

В таблиці 1.1 наведено критерії та бальна оцінка для визначення наукового та технічного рівня науково-дослідної роботи

Таблиця 1.1 – Критерії та бальна оцінка для визначення наукового, технічного та економічного рівня науково-дослідної роботи.

Критерії оцінки	Шкала критеріїв	Індекс оцінки
Час, необхідний для проведення НДР	2 роки і менше 3 роки 4 роки 5-6 років 7 років і більше	+2 +1 0 -1 -2
Технічні показники результатів розробки	Вище рівня кращих світових зразків На рівні кращих світових зразків Нижче рівня кращих світових зразків	+2 0 -2
Можливості отримання авторських свідоцтв на винахід	Впевненість в отриманні авторських свідоцтв Часткові можливості Можливості немає	+2 0 -1
Строк окупності витрат	2 роки і менше 3-4 роки 5 років 6-7 років 8 років і більше	+3 +2 0 -1 -2

В таблиці 1.2 наведено можливі результати оцінки теми НДДКР.

Таблиця 1.2 – Можливі результати оцінки теми НДДКР

Сума індексів	Оцінка теми
Позитивна(+)	Розробка є досить перспективною
Задовільна(0)	Розробка перспективна
Негативна(-)	Розробка не перспективна

Проаналізувавши дані таблиць 1.1 та 1.2, та підрахувавши загальну суму балів (+1+2+2+3=+8), робимо висновок, що дана науково-дослідна робота з дослідження з розробки апаратної платформи для реалізації SHA-алгоритмів на FPGA є досить перспективною.

1.6 Оцінювання комерційного потенціалу розробки

Метою проведення технологічного аудиту є оцінювання комерційного потенціалу результатів НДДКР. В результаті оцінювання можна зробити

висновок щодо напрямів (особливостей) організації подальшого впровадження результатів з врахуванням встановленого рейтингу.

Рекомендується здійснювати оцінювання комерційного потенціалу розробки за 12-ма критеріями, наведеними в таблиці 1.3. [10]

Таблиця 1.3 - Рекомендовані критерії оцінювання комерційного потенціалу розробки та їх можлива бальна оцінка

Бали (за 5-ти бальною шкалою)					
Крите-рій	0	1	2	3	4
Технічна здійсненність концепції:					
1	Достовірність концепції не підтверджена	Концепція підтверджена експертними висновками	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено роботоздат-ність продукту в реальних умовах
Ринкові переваги (недоліки):					
2	Багато аналогів на малому ринку	Мало аналогів на малому ринку	Кілька аналогів на великому ринку	Один аналог на великому ринку	Продукт не має аналогів на великому ринку
3	Ціна продукту значно вища за ціни аналогів	Ціна продукту дещо вища за ціни аналогів	Ціна продукту приблизно до-рівнює цінам аналогів	Ціна продукту дещо нижче за ціни аналогів	Ціна продукту значно нижче за ціни аналогів
4	Технічні та споживчі влас-тивості продук-ту значно гірші, ніж в аналогів	Технічні та споживчі влас-тивості продукту трохи гірші, ніж в аналогів	Технічні та споживчі влас-тивості продук-ту на рівні аналогів	Технічні та споживчі влас-тивості продук-ту трохи кра-щі, ніж в ана-логів	Технічні та споживчі вла-стивості про-дукту значно кращі, ніж в аналогів
5	Експлуатаційні витрати значно вищі, ніж в	Експлуатаційні витрати дещо вищі, ніж в	Експлуатаційні витрати на рівні експлуа-	Експлуатаційні витрати трохи нижчі, ніж в	Експлуатаційні витрати значно нижчі, ніж в
Ринкові перспективи					
6	Ринок малий і не має позити-вної динаміки	Ринок малий, але має позитивну динаміку	Середній ринок з позитивною динамікою	Великий стабільний ринок	Великий ринок з позитивною динамікою
7	Активна конкуренція великих ком-паній на ринку	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкурентів немає
Практична здійсненність					

Бали (за 5-ти бальною шкалою)					
Крите-рій	0	1	2	3	4
8	Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї	Необхідно наймати фахівців або витрачати значні кошти та час на навчання наявних фахівців	Необхідне незначне навчання фахівців та збільшення їх штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так і з комерційної реалізації ідеї
9	Потрібні значні фінансові ресурси, які відсутні. Джерела фінансування ідеї відсутні	Потрібні незначні фінансові ресурси. Джерела фінансування відсутні	Потрібні значні фінансові ресурси. Джерела фінансування є	Потрібні незначні фінансові ресурси. Джерела фінансування є	Не потребує додаткового фінансування
10	Необхідна розробка нових матеріалів	Потрібні матеріали, що використовують- ся у військово-промисловому комплексі	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно використовуються у виробництві
11	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій більше 10-ти років	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності інвестицій більше 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій від 3-х до 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше 3-х років
12	Необхідна розробка регламентних документів та отримання великої кількості дозвільних документів на виробництво та реалізацію продукту	Необхідно отримання великої кількості дозвільних документів на виробництво та реалізацію продукту, що вимагає значних коштів та часу	Процедура отримання дозвільних документів для виробництва та реалізації продукту вимагає незначних коштів та часу	Необхідно тільки повідомлення відповідним органам про виробництво та реалізацію продукту	Відсутні будь-які регламентні обмеження на виробництво та реалізацію продукту

Результати оцінювання комерційного потенціалу розробки зведемо до таблиці 1.4.

Таблиця 1.4 - Результати оцінювання комерційного потенціалу розробки

Критерії	експерт		
	1	2	3
	Бали, виставлені експертами:		
1. Технічна здійсненність концепції	4	3	4
2. Ринкові переваги (наявність аналогів)	4	3	4
3. Ринкові переваги (ціна продукту)	3	4	3
4. Ринкові переваги (технічні властивості)	5	4	5
5. Ринкові переваги (експлуатаційні витрати)	3	3	2
6. Ринкові перспективи (розмір ринку)	3	4	2
7. Ринкові перспективи (конкуренція)	3	2	4
8. Практична здійсненність (наявність фахівців)	4	4	3
9. Практична здійсненність (наявність фінансів)	4	5	4
10. Практична здійсненність (необхідність нових матеріалів)	1	2	1
11. Практична здійсненність (термін реалізації)	2	3	4
12. Практична здійсненність (розробка документів)	3	3	3
Сума балів	39	40	39
Середньоарифметична сума балів СБ	**39,3**		

За даними таблиці 1.4 зробимо висновок щодо рівня комерційного потенціалу дослідження. При цьому доцільно користуватися рекомендаціями, наведеними в таблиці 1.5. [10]

Таблиця 1.5 - Рівні комерційного потенціалу розробки

Середньоарифметична сума балів СБ , розрахована на основі висновків експертів	Рівень комерційного потенціалу розробки
0 - 10	Низький
11 - 20	Нижче середнього
21 - 30	Середній
31 - 40	Вище середнього
41 - 48	Високий

Згідно проведених досліджень рівень комерційного потенціалу розробки становить 39,3 бала, що, згідно таблиці 1.5, свідчить про комерційну важливість проведення даних досліджень (рівень комерційного потенціалу розробки вище середнього).

1.7 Розрахунок узагальненого коефіцієнта якості

В процесі дослідження необхідно розглянути основні технічні показники, пристрою, що може бути спроектований в результаті проведення дослідження з розробки апаратної платформи для реалізації SHA-алгоритмів на FPGA. Ці показники по-різному впливають на загальну якість проектної розробки.

Узагальнений коефіцієнт якості ($B_н$) для нового технічного рішення розрахуємо за формулою [**11**]

$$B_н = \sum_{i=1}^{k} \alpha_i \cdot \beta_i , \qquad (1.1)$$

де k – кількість найбільш важливих технічних показників, які впливають на якість нового технічного рішення;

α_i – коефіцієнт, який враховує питому вагу *i-го* технічного показника в загальній якості розробки. Коефіцієнт α_i визначається експертним шляхом і при цьому має виконуватись умова $\sum_{i=1}^{k} \alpha_i = 1$,

β_i – відносне значення *i-го* технічного показника якості нової розробки.

Відносні значення β_i для різних випадків розраховують за такими формулами:

- для показників, зростання яких вказує на підвищення в лінійній залежності якості нової розробки

$$\beta_i = \frac{I_{ні}}{I_{аі}}, \qquad (1.2)$$

де $I_{ні}$ та $I_{на}$ – чисельні значення конкретного i-го технічного показника якості відповідно для нової розробки та аналога;

- для показників, зростання яких вказує на погіршення в лінійній залежності якості нової розробки

$$\beta_i = \frac{I_{аі}}{I_{ні}}. \qquad (1.3)$$

Використовуючи наведені залежності можемо проаналізувати та порівняти техніко-економічні характеристики аналогу та майбутньої розробки на основі отриманих наявних та проектних показників, а результати порівняння зведемо до таблиці 1.6.

Таблиця 1.6 – Порівняння основних параметрів пристрою що проектується та аналога.

Показники (параметри)	Одиниця вимірю-вання	Аналог	Проектований пристрій	Відношення параметрів нової розробки до аналога	Питома вага показника
Напруга живлення	В	10..30	9..18	1,67	0,2
Верхня границя виміру температури	С?	1300	1500	1,15	0,35
Вага	Кг	0,1	0,17	0,58	0,1
Споживана потужність	Вт	10	8	1,25	0,2
Похибка вимірювання	%	2	1,5	1,33	0,15

Узагальнений коефіцієнт якості ($B_н$) для нового технічного рішення складе

$$B_н = \sum_{i=1}^{k} \alpha_i \cdot \beta_i = 1,67*0,2+1,15*0,35+0,58*0,1+1,25*0,2+1,33*0,15 = 1,24.$$

Отже за технічними параметрами, згідно узагальненого коефіцієнту якості розробки, проектований компонент переважає існуючі аналоги приблизно в 1,24 рази.

1.8 Прогнозування витрат на виконання НДДКР

Для детального обґрунтування доцільності проведення дослідження з розробки апаратної платформи для реалізації SHA-алгоритмів на FPGA необхідно здійснити попередній розрахунок витрат на проведення науково-дослідної роботи. Для визначення суми витрат на проведення НДДКР передбачено складання приблизного кошторису цих витрат **[10].**

Таблиця 1.7 – Основна заробітна плата дослідників та розробників

Найменування посади	Місячний посадовий оклад, грн.	Оплата за робочий день, грн.	Число днів роботи	Витрати на заробітну плату, грн.
1. Керівник проекту	12400,00	563,64	52	29309,09
2. Консультатнт-розробник архітектури ЕОМ	10100,00	459,09	15	6886,36
3. Інженер-схемотехнік	9200,00	418,18	35	14636,36
4. Інженер-конструктор радіоелектронних систем	9000,00	409,09	40	16363,64
5. Лаборант	5200,00	236,36	45	10636,36
Разом				77831,82

Витрати на основну заробітну плату працівників (*Зр*), що здійснюють підготовку робочих місць необхідних для досліджень, підготовку та формування інформаційних пакетів, підготовку та монтаж обладнання та макетів для НДДКР складе в межах 5400,00 грн.

Додаткова заробітна плата дослідників складе приблизно 9200,00 грн.

Нарахування на заробітну плату дослідників складуть приблизно 20500,00 грн.

Таблиця 1.8 – Витрати на основні матеріали

Найменування матеріалу, марка, тип, сорт	Одиниця виміру	Ціна за одиницю, грн.	Витрачено	Вартість витраченого матеріалу, грн.
Папір офісний	уп.	81,00	6	486,00
Диск оптичний	шт.	10,50	6	63,00
Канцтовари	компл.	200,00	6	1200,00
Тека пластикова	шт	110,00	6	660,00
Картридж Canon 23-12X	шт	345,00	2	690,00
Блок памяті 32 Gb	шт	299,00	2	598,00
Бензосуміш	кг	40,00	0,25	10,00
Лак	кг	135,00	0,1	13,50
Склотекстоліт СФ-2Н-35	кг	152,00	0,2	30,40
Смола поліамідна 68С	кг	68,70	0,15	10,31
Пресматеріал К- 124-38	кг	74,30	0,1	7,43
Стержні текстолітові	кг	110,00	0,2	22,00
Стрічка поліхлорвінілова ізоляційна	кг	105,00	0,08	8,40
Всього				3799,04

Таблиця1.9 – Витрати на комплектуючі для формування компонентів для НДДКР

Найменування комплектуючих	Кількість, шт.	Ціна за штуку, грн.	Сума, грн.
1. Варикапи			
КВ130А	2	2,00	4,00
2. Мікросхеми			
LM26	1	60,00	60,00
HIH-4000	1	160,00	160,00
MPX4115	1	148,00	148,00
AD1B60BJ	1	220,00	220,00
ATtiny25	1	26,00	26,00
3. Конденсатори			
К10-36-50В±20%	2	1,20	2,40
К10-19-32В±20%	9	2,30	20,70
К10-23-16В±20%	4	2,30	9,20
КПЕ-30В±20%	1	8,40	8,40
К10-40-50В±20%	1	2,40	2,40
4. Кварц			
ГК-163	1	30,00	30,00

Найменування комплектуючих	Кількість, шт.	Ціна за штуку, грн.	Сума, грн.
РВ-59	1	15,00	15,00
5. Резистори			
С2-10± 5%	9	1,00	9,00
С2-23± 5%	8	2,10	16,80
С2-11± 5%	7	1,30	9,10
6. Роз’єми			
ОНП-КГ-4-5/16-Р	2	2,20	4,40
7. Транзистори			
MRF962	1	6,00	6,00
2N3600	4	5,40	21,60
8. Індуктивність	9	7,30	65,70
Всього			838,70

Амортизація обладнання

Таблиця 1.10 - Величина амортизаційних відрахувань

Найменування обладнання	Балансова вартість, грн	Строк корисного використання, років	Термін використання обладнання, міс.	Величина амортиза-ційних відрахувань, грн
Аналітичний комплекс обробки даних	32000,00	5	2	1066,67
Система метрологічного аналізу	27000,00	5	2	900,00
Апаратна платформа для реалізації SHA-алгоритмів на FPGA	21000,00	4	2	875,00
Повноциклова автоматизована система проектування	32800,00	5	2	1093,33
Оргтехніка	15000,00	4	2	625,00
Приміщення лабораторії	330000,00	25	2	2200,00
Всього				6760,00

Таблиця 1.11 – Витрати на електроенергію при проведенні досліджень

Найменування обладнання	Кількість годин роботи обладнання, год.	Встановлена потужність, кВт	Коефіцієнт використання потужності	Величина оплати
Аналітичний комплекс обробки даних	400	0,65	0,93	703,64
Система метрологічного аналізу	300	0,35	0,93	284,16
Апаратна платформа для реалізації SHA-алгоритмів на FPGA	300	0,32	0,93	259,80
Повноциклова автоматизована система проектування	350	0,65	0,93	615,68
Оргтехніка	200	0,86	0,93	465,48
Всього				2328,77

Інші витрати складуть в межах 210000,00 грн.

Загальні витрати на проведення досліджень стосовно дослідження з розробки апаратної платформи для реалізації SHA-алгоритмів на FPGA

$$В = 77831,82+5400,00+9200,00+20500,00+3799,04+838,70+6760,00+2328,77+ +210000,00=336658,32 \text{ (грн.)}.$$

1.9 Доцільність науково-дослідної роботи

Для обґрунтування доцільності виконання науково-дослідної роботи використовується спеціальний комплексний показник, що враховує важливість, результативність роботи, можливість впровадження її результатів у практичну діяльність, величину витрат на роботу.

Комплексний показник K_P рівня НДДКР розраховується за формулою **[11]**

$$K_P = \frac{I^n \cdot T_C \cdot R}{B \cdot t}, \qquad (1.4)$$

де I - коефіцієнт важливості роботи, $I = 2...5$;

n - коефіцієнт використання результатів роботи; $n = 0$, коли результати роботи не будуть використовуватись; $n = 1$, коли результати роботи будуть використовуватись частково; $n = 2$, коли результати роботи будуть використовуватись в дослідно-конструкторських розробках; $n = 3$, коли результати можуть використовуватись навіть без проведення дослідно-конструкторських розробок;

T_C - коефіцієнт складності роботи, $T_C = 1...3$.

R - коефіцієнт результативності роботи; якщо результати роботи плануються вище відомих, то $R = 4$; якщо результати роботи відповідають відомому рівню, то $R = 3$; якщо нижче відомих результатів, то $R = 1$;

B – вартість НДР, (тис. грн.) B=205,3 тис. грн;

t - час дослідження роботи, років.

Підставляючи числові дані і отримаємо

$$K_p = 4^2*2*4 / (336,7*2/12) = 2,28.$$

Оскільки $K_P > 1$, тому науково-дослідну роботу з дослідження з розробки апаратної платформи для реалізації SHA-алгоритмів на FPGA можна вважати економічно доцільною з достатньо високим науковим, технічним та економічним рівнем.

1.10 Висновки по розділу

У цьому розділі були розглянуті архітектура ПЛІС, на основі якої необхідно розробити пристрій, технології, використовувані для опису логіки роботи пристрою, а також етапи його проектування.

Згідно проведеного аналізу рівень комерційного потенціалу розробки становить 39,3 балів, що свідчить про комерційну важливість проведення даних досліджень (рівень комерційного потенціалу розробки вище середнього). До того ж комплексний показник рівня НДДКР (Кр >1) свідчить про те, що науково-дослідну роботу з дослідження з розробки апаратної платформи для реалізації SHA-алгоритмів на FPGA можна вважати економічно доцільною з достатньо високим науковим, технічним та економічним рівнем. Також рівень якості проектованого компоненту в порівнянні з аналогами вищий в 1,24 рази, що додатково свідчить про позитивні перспективи, як науково-дослідної роботи так і відповідного проектного компоненту.

Таблиця 1.12 – Основні техніко-економічні показники аналога і нової розробки

Показники	Одиниця виміру	Аналог	Засіб, що проектується	Відношення параметрів
Напруга живлення	В	10..30	9..18	1,67
Верхня границя виміру температури	С	1300	1500	1,15
Вага	Кг	0,1	0,17	0,58
Споживана потужність	Вт	10	8	1,25
Похибка вимірювання	%	2	1,5	1,33
Вартість	грн	1670	1450	0,87

Структурна схема налагоджувальної плати наведена в додатку Б 08-36.МКР.012.00.000 ПЛ1.

2 МЕТОДИ ЗАХИСТУ ПЗ ПЛІС

2.1 Потенційні уразливості ПЗ ПЛІС

Оскільки ПЛІС з архітектурою FPGA не має можливості зберігати свою програму при відключенні живлення, виникає потреба в зовнішньому модулі енергонезалежної пам'яті (EEPROM англ. Electrically Erasable Programmable Read - Only Memory). Програмне забезпечення(бітовий потік) може бути прочитане зловмисником як безпосередньо з ПЗП, так і під час конфігурації чіпа ПЛІС. За відсутності криптозахисту програмного забезпечення зловмисник отримає готову програму, якою він зможе скористатися як для декомпіляції, тобто отримання початкового коду, так і для її нелегального копіювання в інші пристрої ПЛІС, а також для несанкціонованої модифікації. Існують різні методи захисту ПЗ : шифрування бітового потоку, використання мікросхем спеціальної пам'яті, перевірка серійного номера ПЛІС.

2.2 Модель порушника

Основною зацікавленою особою в нелегальному доступі до інтелектуальної власності являється конкурент або особа, працююча на замовлення, існую також компанії, які спеціалізуються на розкритті «засобів інтелектуальної власності», вирішуючи завдання зворотного проектування. Оскільки головною метою є клонування облаштування незалежно від розуміння принципів його роботи, для витягання програми зломщикові досить мати базові знання електроніки, також навичками роботами з ПЛІС і мікроконтролерами [10].

2.3 Вбудовані методи захисту шифруванням бітового потоку

Середовище розробки, що випускається фірмою Xilinx, надає можливість

захисту програми «З коробки» для деяких серій сімейств чіпів. Шифрування бітового потоку, уперше представлене Xilinx на чіпі Virtex 2, служить як для того, щоб запобігти клонуванню пристрою, так і для того, щоб захистити конфіденційність інтелектуальної власності.

Облаштування Virtex4, 5,6 і Spartan 6 мають на чіпі блок дешифрування SHA для підтримки зашифрованих бітових потоків. Система шифрування бітового потоку Xilinx складається з двох частин: шифрування бітового потоку на комп'ютері середовищем розробки і розшифровка бітового потоку на чіпі з виділеною пам'яттю для зберігання 256-бітового ключа шифрування. Використовуючи Xilinx ISE – програмне забезпечення користувач генерує відразу і ключ шифрування, і зашифрований бітовий потік [13].

Перший алгоритм шифрування застосовувався для серій Virtex 2- алгоритм симетричного шифрування 3 – DES (Data Encryption Standard), він застарів, для новіших серій, таких як Spartan - 7, Virtex - 7 зараз використовують вдосконалений алгоритм під назвою «Рейндол». Цей алгоритм був офіційно оголошений новим стандартом шифрування SHA в 2001 році, він є безпечнішим, швидкодіючим і гнучким. Алгоритм ґрунтований на декількох замінах, підстановках і лінійних перетвореннях, кожне з яких виконується блоками по 16 байт, тому він називається блоковим шифром. Операції повторюються кілька разів, кожен з яких називається «раунд». На основі ключа шифрування виконується обчислення унікального ключа для кожного раунду і вбудовується в обчислення. Завдяки подібній блоковій структурі SHA, досить зміни одного біта або в ключі, або в текстовому блоці, щоб повністю змінився увесь шифр. Різниця між SHA - 128, SHA - 192 і SHA - 256 алгоритмами полягає тільки в довжині ключа. Ключ, завдовжки 128, 192 або 256 біт, криптостійкий до атак порівняно з 56-бітовим ключем DES. На сьогодні усі спроби зламати шифр SHA увінчалися крахом. Завдяки описаним перевагам, SHA залишається переважним алгоритмом для урядових організацій, банків і інших систем, що вимагають високий рівень безпеки, по всьому світу [14]

Облаштування Virtex 4, 5, 6 і Spartan 6 зберігають ключ шифрування усередині, у виділеній ОЗП, живленою маленькою, підключеною зовні батареєю або в одного разу програмованих осередках (OTP fuses – Xilinx eFUSE). Ключ шифрування може бути прошитий в пристрої тільки через порт JTAG. В процесі конфігурації пристрій здійснює зворотну операцію, розшифрувавши бітовий потік, що входить, використовуючи вбудований в FPGA апаратний блок розшифровки SHA. Цей блок не може бути використаний для інших цілей, окрім як розшифровка бітового потоку, тому його логіка не доступна для призначеного для користувача проекту і не може бути використана для розшифровки будь-яких даних окрім конфігураційного бітового потоку.

Оскільки доданий новий шар безпеки, зашифрований бітовий потік не може бути завантажений в FPGA ПЛІС, в який раніше був завантажений не зашифрований бітовий потік без початкової ініціалізації повного циклу прошивки, який, передусім, очищає вміст конфігураційної пам'яті. Аналогічно, завантаження незашифрованого бітового потоку в FPGA ПЛІС, раніше конфігурованої зашифрованим бітовим потоком, вимагає повної реконфігурації. Ця вимога(очищати конфігураційну пам'ять перед прошивкою) заважає атакам реверс-інжиніринг [13].

2.3.1 Зберігання ключів

Найбільш важливим для тих, що підтримують шифрування бітового потоку чіпів являється безпечний спосіб зберігання ключа. Virtex4, 5,6 і Spartan 6 підтримують зберігання 256-бітового ключа в ОЗП, що живиться від батареї. Облаштування Virtex 6 і Spartan6 забезпечують додаткову опцію енергонезалежного зберігання ключа в OTP- осередках [13].

2.3.2 ОЗП з живленням від батареї

256-бітовий ключ зберігається в енергозалежних елементах пам'яті, що знаходяться на мікросхемі. Це спеціальна пам'ять повинна отримувати безперервне живлення від окремого джерела енергії для підтримки її вмісту. Впродовж роботи ПЛІС, ці елементи пам'яті живляться від вхідної напруги, щоб продовжити довговічність батареї. Застосування може скористатися перевагою цієї необхідності в зовнішньому живленні, позбавивши живлення і ПЛІС, і сховище ключа при спробі втручання або неавторизованого доступу.

Крім того, в облаштуваннях Xilinx є додаткова функція безпеки : будь-яка спроба читання або запису батарейно-живленої пам'яті викликає очищення її вмісту і конфігурації FPGA до діставання доступу [13].

2.3.3 eFUSE- регістри

Virtex 6 і Spartan 6 мають наступні eFUSE- регістри:

– FUSE_KEY для зберігання 256-бітового ключа SHA;

– FUSE_USER для зберігання 32-бітового призначеного для користувача коду(тільки Virtex 6);

– FUSE_ID для зберігання 57-бітового Devise DNA ID (усі облаштування Virtex 6 і Spartan 6);

– FUSE_CNTL для зберігання 32-х контрольних бітів;

Ці осередки дозволяють користувачеві безпечно зберігати ключ для дешифрування бітового потоку без необхідності використання батарейно-живленої пам'яті і самої батареї.

OTP eFUSE- осередку прошиваються перманентно через зовнішнє джерело напруги і можуть бути використані в застосуваннях, де використання батарейно-живленої ОЗП не бажано. Крім того, усі eFUSE- осередки продубльовані для збільшення надійності зберігання ключів [14].

2.3.4 Атака на вбудовані методи захисту

Кожен захист може бути вразливий до атак. Шифрування даних за допомогою алгоритму SHA було визнане найнадійнішим серед запропонованих, і SHA був прийнятий як стандарт шифрування даних. Він стійкий до теоретичного криптоаналізу, проте атака по сторонніх каналах використовує інформацію про фізичні процеси, що відбуваються в пристрої, які не розглядаються в теоретичному опис криптографічного алгоритму. У криптографії існує цілий клас атак, які називають атаками по другорядних каналах, що використовують фізичні параметри обчислювального пристрою для визначення ключів шифрів.

Криптографічний алгоритм записується у вигляді математичних операторів, тому при дослідженні алгоритму вивчається математична стійкість або криптостійкість. Якщо подивитися на виконання алгоритму з точки зору фізики — дані відбивають фізичний стан логічних елементів, тоді як обчислення — це фізичні процеси, які переводять стан логічних елементів з одного в інше. Отже, виконання програми є перетворення фізичних сигналів, і з такої точки зору результат роботи алгоритму визначається законами фізики. Таким чином, реалізація криптографічного алгоритму може розглядатися в математичній, програмній і фізичній середовищах.

Будь-який алгоритм виконується за допомогою апаратних засобів, під якими розуміються будь-які обчислювальні механізми, здатні виконувати логічні операції І, АБО і операцію логічного заперечення. Усі обчислювальні засоби мають дві загальні властивості: для виконання обчислення необхідно витратити енергію (електричну енергію), для коректного виконання операцій усі обчислювальні механізми вимагають нормальних зовнішніх умов(необхідність в постійній напрузі і струмі). На цих двох властивостях ґрунтовані апаратні атаки (hardware attacks).

Виконання алгоритму змінює властивості обчислювального пристрою. Алгоритм SHA симетричний, він має різне число базових операцій : 11

складань з ключем, по 10 операцій таблиці заміни (S - box), і лише 9 операцій над колонками Mix Column (детальний опис алгоритму пункт 2.8). Атака по сторонніх канала х (Side Channel Attacks) ґрунтована на вимірі вхідної напруги за допомогою додаткових пристроїв. Виміряна напруга дозволяє визначити:

– Почало і закінчення роботи шифру для обчислення часу роботи усього шифру;

– Почало і закінчення роботи кожного раунду, що дозволяють вичислити час роботи раунду.

– Операції кожного раунду : складання з ключем, таблиця заміни Sbox, перемішування колонок, побайтовий зсув рядків.

Кожна окрема група інструкцій (окрема інструкція) споживає індивідуальну кількість енергії. Знаючи те, яка кількість енергії споживається при виконанні інструкції, яка залежить від значення ключа і відомих нам параметрів, то можна визначити правильне значення ключа [15].

2.4 Метод захисту за допомогою вбудовування дешифратора

Розглянута нижче концепція розроблена на основі вбудованого методу захисту шифрування бітового потоку. Вона є деяким аналогом апаратного блоку шифрування SHA, але реалізується програмним способом.

Основна ідея полягає в наступному: середовище розробки компілює основну програму. Цей файл шифрується сторонніми засобами (наприклад, за допомогою крипто-бібліотеки). На вхід модуля шифрування поступає основна програма, а на виході формується масив зашифрованих байтів. Ці байти поміщаються в модуль дешифратора. У середовищі розробки створюється і компілюється модуль дешифратора; потім модуль дешифратора і зашифрована програма усередині нього прошивається на ПЗП, а на ПЛІС поміщається ключ шифрування, в енергозалежну пам'ять, що підживлюється зовні батареєю. Під час старту ПЛІС завантажується модуль дешифратора і

запускається, далі розшифровує програму за допомогою вбудованого в ПЛІС секретного ключа, потім реконфігурує вентильну матрицю. Після цього програма запускається в робочому режимі [16].

У разі атаки на цю криптосистему все, що отримає зловмисник - це зашифрований потік байтів, який не має для нього практичної цінності, якщо він не володіє секретним ключем (ключем шифрування). Таким чином, у нього відсутня можливість декомпіляції, модифікації або нелегального копіювання.

У зв'язку з тим, що тільки новітні серії ПЛІС допускають можливість динамічної реконфігурації, цей спосіб захисту буде доступний тільки на них. А вони, як правило, вже мають у своєму складі вбудовані методи шифрування бітових потоків [15].

2.4.1 Атака на метод захисту, що використовує вбудований дешифратор

Метод захисту, описаний вище в пункті 2.4, також не захищений від атак по сторонніх каналах, що використовує вимір вхідної напруги і параметри алгоритму для знаходження секретного ключа. Ця атака описана в п. 2.3.4.

2.5 Використання мікросхем спеціальної пам'яті

Пристрої, засновані на ПЛІС з архітектурою FPGA, випускаються по технології статичного ОЗП, конфігурація яких зберігається в спеціалізованій зовнішній пам'яті (постійний запам'ятовуючий пристрій). При включенні живлення конфігураційний файл у вигляді бітового потоку даних поступає в оперативний пристрій (вентильна матриця), що запам'ятовує. При несанкціонованих діях зловмисником цей конфігураційний потік даних може бути перехоплений і використаний для копіювання проекту на інших пристроях - це означає, що проекти, реалізовані на FPGA, уразливі для копіювання. Сучасніші серії ПЛІС Xilinx використовують кодований

конфігураційний потік, вони, як правило, дорожче і містять додатковий вбудований дешифратор на самому чіпі і секретний ключ, прошитий в спеціальний енергозалежний регістр, що підживлюється додатковою батареєю. Цей метод захисту дозволяє запобігти нелегальному клонуванню, як описано вище. Оскільки він підтримується тільки новітніми серіями, встає питання про те, як же захистити старіші версії ПЛІС з архітектурою FPGA. Для таких сімейств ефективним рішенням проблеми стало використання мікросхем спеціальної пам'яті.

Використання мікросхем спеціальної пам'яті, по-іншому «визначення - друг або ворог» (Identification Friendor Foe, IFF) для забезпечення захисту ПЛІС FPGA від копіювання. Головна ідея такого методу захисту полягає в тому, що доки не співпаде геш-послідовність, вичислена спеціальним блоком усередині FPGA і геш-послідовність, вичислена в зовнішній мікросхемі спеціальної пам'яті проекту ПЛІС FPGA не запуститься і, відповідно, не почне функціонувати. Навіть при перехопленні конфігураційного файлу знайти блок розрахунку геш-послідовності майже нереально, а в мікросхемі спеціальної пам'яті немає доступу до розрахунку блоку геш-послідовності, тому проект захищений при такому способі атаки. Звідси витікає, що мікросхема спеціальної пам'яті використовується як допоміжний захист для ПЛІС FPGA.

2.5.1 Реалізація методу

Концепція Identification Friendor Foe вимагає допоміжної мікросхеми пам'яті, що реалізовує геш-алгоритм. На рис. 2.1 представлена реалізація цієї концепції на основі спеціальної мікросхеми DS28E01.

Ця мікросхема, що випускається фірмою Maxim Integrate Product, підходить для реалізації додаткового захисту ПЛІС. Вона містить пам'ять EEPROM місткістю 1024 біта, має вбудований модуль для апаратного обчислення геш-послідовності з використанням алгоритму SHA – 1 (Secure

Hash Algorithm). Геш-послідовність являється 160-бітовим MAC (Message Authentication Code), обчислювана, за допомогою алгоритму SHA - 1 в мікросхемі DS28E01 для згенерованого масиву випадкових чисел(формується в спеціальному модулі в FPGA). Масив згенерованих чисел передається в мікросхему, записується і зберігається в пам'яті EEPROM. Цей захист ґрунтований на симетричному шифруванні, секретний ключ завдовжки 64 біта розташовується в секретній області EEPROM, і до нього немає доступу, а на ПЛІС він знаходиться в конфігураційному файлі, що робить його практично невидимим для хакера.

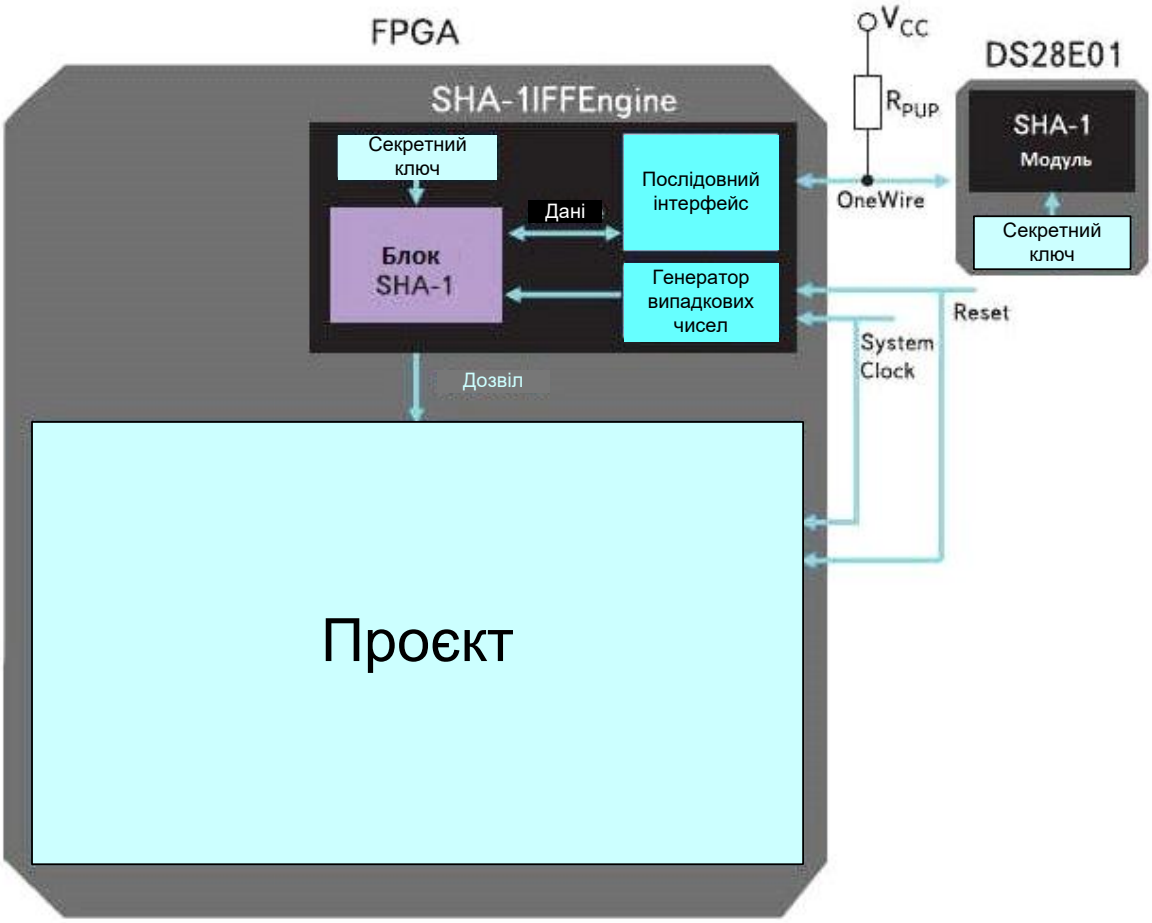


Рисунок 2.1 – Реалізація концепції IFF

Як видно на рисунку (див. рис. 2.1), підключення FPGA до мікросхеми DS28E01 реалізоване завдяки однопровідному інтерфейсу 1 - Wire введення/виведення, що є двонаправленою лінією. Для забезпечення рівня логічної одиниці лінія підтягується зовнішнім резистором до напруги

живлення відповідного банку введення/виведення.

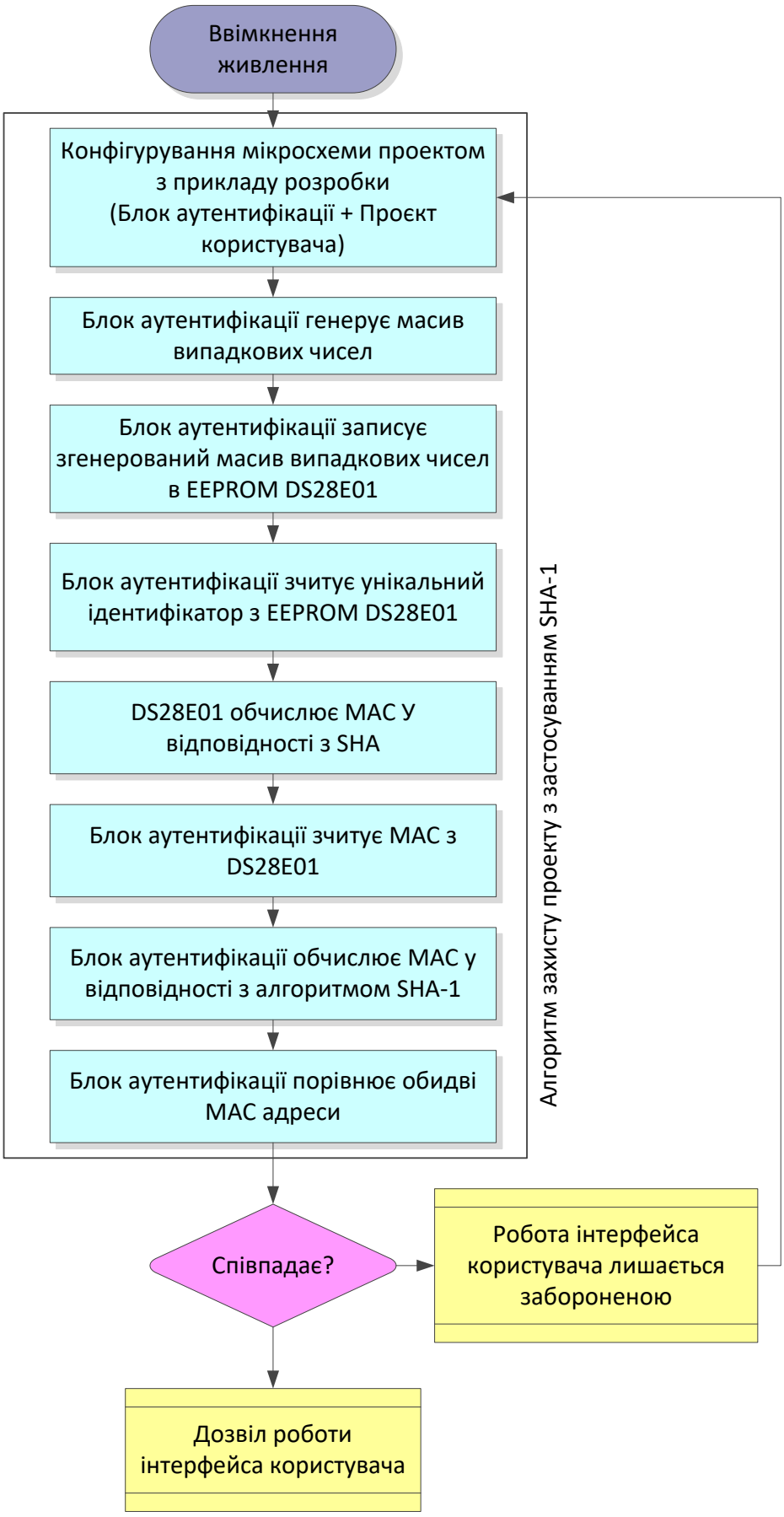


Рисунок 2.2 – Алгоритм блоку перевірки при реалізації концепції IFF

Концепція IFF реалізована компанією Altera для захисту проектів в FPGA сімейства Cyclone 3 від нелегального клонування.

На рис. 2.2 показано алгоритм блоку аутентифікації концепції IFF.

Використання такого методу захисту проекту рекомендується при невеликому виробництві, якщо потрібно масове виробництво, можна замовити у компанії-виробника мікросхеми із запрограмованим на фабриці секретним ключем [18].

2.5.2 Атака на метод захисту, що використовує мікросхему спеціальної пам'яті

На пристроях ПЛІС, побудовані по методу захисту з використанням мікросхем спеціальної пам'яті, можна виділити два напрями для атаки:

– Атака безпосередньо на мікросхему спеціальної пам'яті

– Читання бітового потоку при завантаженні конфігураційного файлу з ПЗП в ПЛІС

Атака на мікросхему спеціальної пам'яті полягає в тому, що зловмисник спробує витягнути ключ, використовуваний при обчисленні MAC, який зберігається в енергонезалежній пам'яті мікросхеми. Така атака приречена на провал, тому що секретний ключ, знаходиться в захищеній пам'яті, яка не підтримує операції читання з неї [19].

Оскільки спроби атаки на мікросхему спеціальної пам'яті не дають позитивних результатів, єдиним способом, що залишився, обійти захист в такій системі - читання бітового потоку при завантаженні конфігураційного файлу з ПЗП в ПЛІС, його декомпіляція, виявлення місця, де знаходиться блок аутентифікації з подальшим з'ясуванням ключа або видаленням цілого блоку аутентифікації з програми. Проте в цьому випадку, порушник стикається з проблемою декомпіляції, описаної нижче, в пункті 2.6.

2.6 Проблема декомпіляції конфігураційного файлу

У криптографії проблема декомпіляції схожа з проблемою знаходження прообразу односторонньої функції. Одностороння функція — це ефективно обчислювана функція, але для завдання її інвертування не існує ефективних алгоритмів. Під інвертуванням розуміється масове завдання знаходження по заданому значенню функції одного (будь-кого) значення з прообразу(в деяких випадках знаходження прообразу неможливе зважаючи на відсутність зворотної функції). Декомпіляція конфігураційного файлу є зворотною функцією компіляції конфігураційного файлу. Абсолютний захист конфігурації досягається тоді і тільки того, коли для функції компіляції конфігураційного файлу за заданим початковим уявленням (наприклад, на мові VHDL) не існує зворотної функції. Можливість декомпіляції програм оцінюється по аналогії з побудовою зворотної функції.

Експерименти по декомпіляції представлення конфігураційного файлу у форматі XDL в представлення на VHDL показують, що декомпіляція для важливих окремих випадків — коли початкове представлення на VHDL задає комбінаційну схему — можна виконати на практиці [20].

2.7 Концепція розробленого модуля криптозахисту

Для вбудованого програмного забезпечення ПЛІС необхідно розробити ефективну концепцію захисту від копіювання. Метод, реалізований в цій концепції, ґрунтований на аутентифікації серійного номера, тобто на тому, чи може чіп з цим номером використати основну програму.

2.8 Опис вибраного криптоалгоритма шифрування

Поліпшений стандарт шифрування SHA був уперше представлений Національним інститутом стандартів і технологій (NIST) в листопаді 2001 р.

Він прийшов на зміну стандарту DES, довжина ключа якого складає всього 56 біт, що робить цей алгоритм недостатньо надійним.

Алгоритм Rijndael, названий так по іменах своїх бельгійських творців, Йоана Дамена (Joan Daemen) і Вінсента Раймена (Vincent Rijmen), за результатами конкурсу NIST оголошений кращим серед усіх пропозицій різних алгоритмів і прийнятий як стандарт шифрування даних, завдяки своїй надѐжности, простоті реалізації і невисоким вимогам до об'єму пам'яті.

Існують три різні версії SHA : SHA - 128, SHA - 192, SHA - 256. Усі вони мають довжину блоку 16 байт, тоді як допустима довжина ключа складає 128, 192 або 256 біт відповідно.

У моїй роботі розглядається шифрування/розшифровка бітового потоку за допомогою алгоритму SHA - 128. Алгоритм є блоковим і симетричним.

2.9 Концепція алгоритму

Об'єкт шифрування - набір байтів. Реалізація розшифровки за допомогою 128-бітового ключа.

Алгоритм SHA складається з раундів шифрування, що часто повторюються, кількість раундів залежить від довжини ключа(у нашому випадку 10 раундів для довжини ключа 128 біт), кожен раунд включає 4 операції:

- заміщення байтів, відповідна операція «Subbytes»;
- зрушення рядків, відповідна операція «Shiftrows»;
- перемішування стовпців, відповідна операція «Mixcolumns»;
- додавання ключа раунду, відповідна операція «Addroundkey».

Для кожного раунду формується свій ключ раунду, розміром 128 біт. Для отримання ключа раунду визначена операція «Keyexpansion». Щоб отримати ключ раунду з індексом (n+1) з ключа раунду з індексом n :

1. Обчислюють новий перший стовпець наступного ключа раунду, відповідно до рис. 2.3.

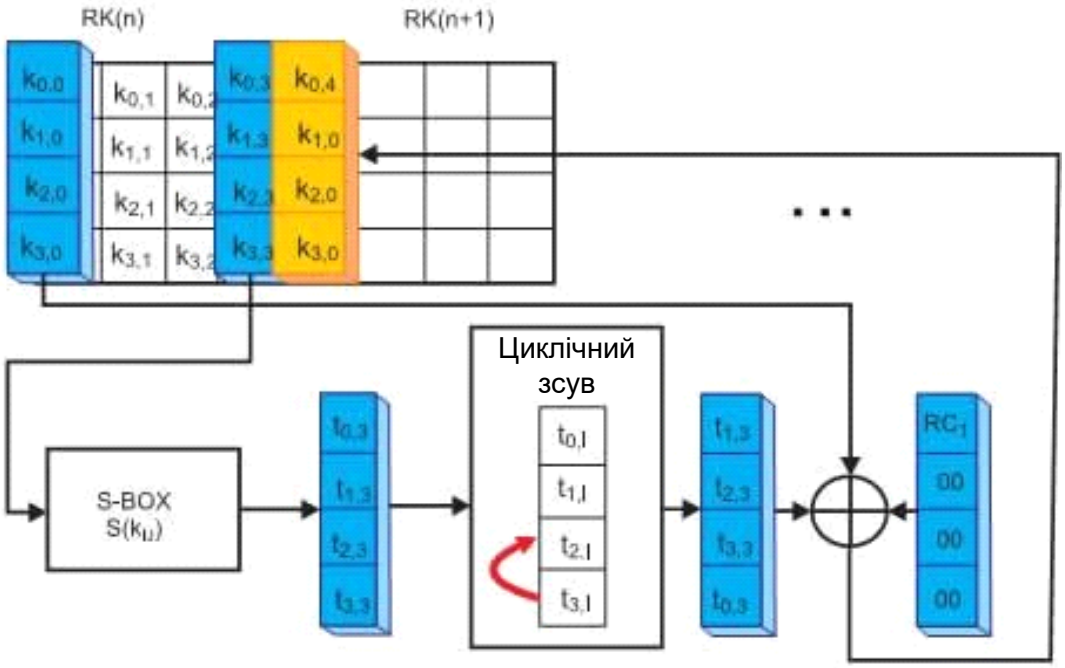


Рисунок 2.3 – Отримання першого стовпця наступного ключа раунду

Знаходження нового(поточного) ключа за допомогою операцій над попереднім ключем раунду. Спочатку усі 4 байти четвертого стовпця попереднього ключа треба замістити за допомогою операції «Subbytes» (операція описана нижче). Ці чотири байти зрушуються вертикально на один байт, а потім виконується логічна операція XOR(що виключає АБО) цих байтів з першим стовпцем попереднього ключа. В результаті цих дій виходить новий перший стовпець для поточного ключа.

2. Стовпці з 2 по 4 нові ключі раунду розраховуються так:

[новий другий /третій/четвертий стовпчик] = [новий перший/ другий/третій/ стовпчик] XOR [старий другий /третій/четвертий стовпчик]

Кожен раунд включає перетворення з використанням відповідного криптографічного ключа, для того, щоб забезпечити секретність шифрування, як показано на рис. 2.4.

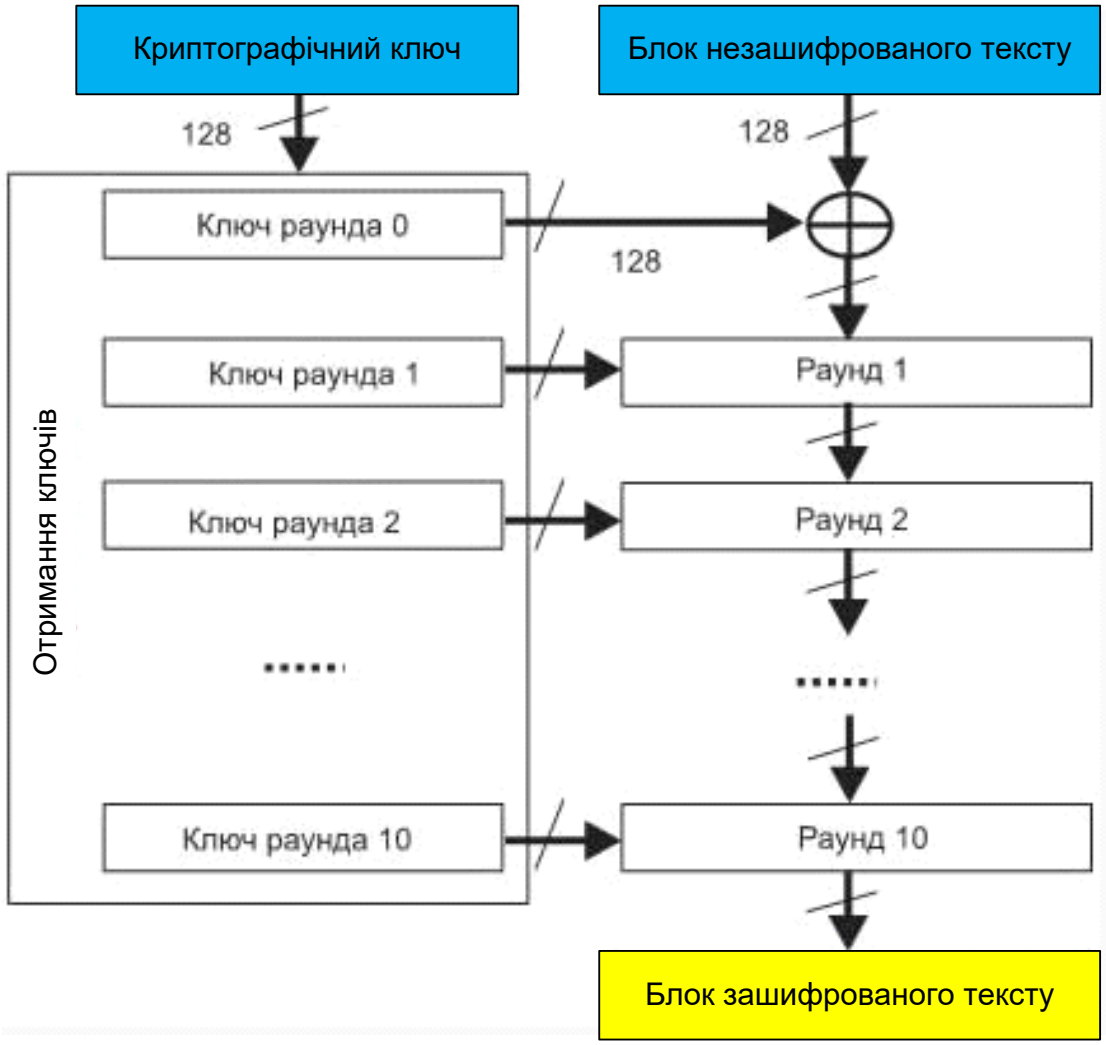


Рисунок 2.4 – Структура алгоритму SHA - 128

Після початкового раунду, під час якого здійснюється логічна операція XOR першого ключа раунду і незашифрованого тексту (операція «Addroundkey»), слідують дев'ять однаково структурованих раундів. Десятий раунд подібний до раундів з першого по дев'ятий, але без операції перемішування стовпців. Нижче пояснюються ці чотири операції [21].

2.9.1 Структура ключа і початкових даних

І ключ, і початкові дані (що називаються також «станом») розміром 16 байт структуровані у вигляді матриці байтів розміром 4x4. На рис. 2.5 показано, як 128-бітовий ключ і блок даних розподіляються по осередках матриць.

Стан

a_0	a_4	a_8	a_{12}
a_1	a_5	a_9	a_{13}
a_2	a_6	a_{10}	a_{14}
a_3	a_7	a_{11}	a_{15}

Ключ

k_0	k_4	k_8	k_{12}
k_1	k_5	k_9	k_{13}
k_2	k_6	k_{10}	k_{14}
k_3	k_7	k_{11}	k_{15}

Рисунок 2.5 – Структура ключа і стану

2.9.2 Заміщення байтів (операція « Subbytes»)

Операція «Subbytes» - це нелінійне заміщення. Це головна причина надійності шифрування алгоритму SHA. Етап заміщення байтів - пошук по таблиці S-Box.

Ця таблиця є 256-байтним масивом і використовується для заміни одного байта іншим. Значення елементів S-Box представлені в шістнадцятковій системі числення. Сама ж таблиця рис.2.6 отримана за допомогою перетворень вже відомого нам поля GF(28). Складено по: [22].

		y														
		0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
x	0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
	1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
	2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
	3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
	4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
	5	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
	6	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
	7	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
	8	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
	9	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
	a	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
	b	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
	c	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
	d	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
	e	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
	f	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

Рисунок 2.6 – Таблиця S - Box

Складено по: [22].

Кожен байт стану можна записати як (xy) в шістнадцятиричній системі числення. Тоді цей байт (xy) замінюється на іншій, що стоїть на перетині рядка x і стовпця у таблиці S - Box. Наприклад, (54) заміниться на (20).

За допомогою цієї таблиці пошуку елементи матриці стану(початкових даних) розміром 4x4 заміщаються відповідними значеннями, знайденими в таблиці S, - Box, як показано на рис.2.7.

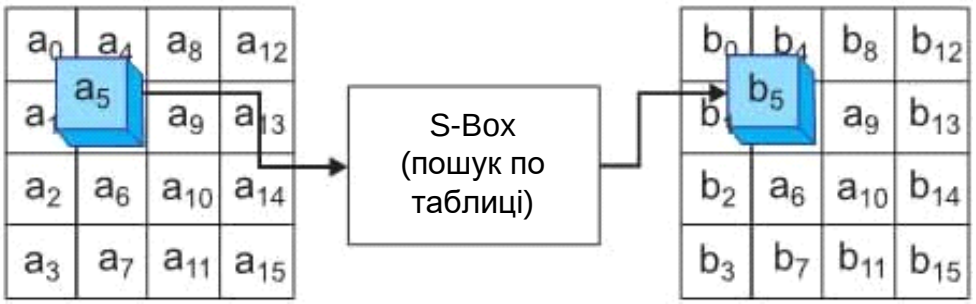


Рисунок 2.7 – Операція заміщення байтів

2.9.3 Зсув рядків (операція «Shiftrows»)

Операція зрушення рядків обробляє стан відрядковий - зрушення рядка на різну величину зміщення залежно від номера рядка. Загальну формулу зміщення можна записати так: n-ий рядок зрушується на (n - 1) байт вліво, починаючи з до (n =4). Відповідно, виходячи з цієї формули, перший рядок стану у вигляді байтової матриці 4x4 залишається незмінною, другий рядок - зміщується в матриці на один байт вліво, третій рядок зміщується на два байти вліво, а четвертий рядок зміщується на три байти вліво. На рис.2.8 показана робота операції зсуву рядків.

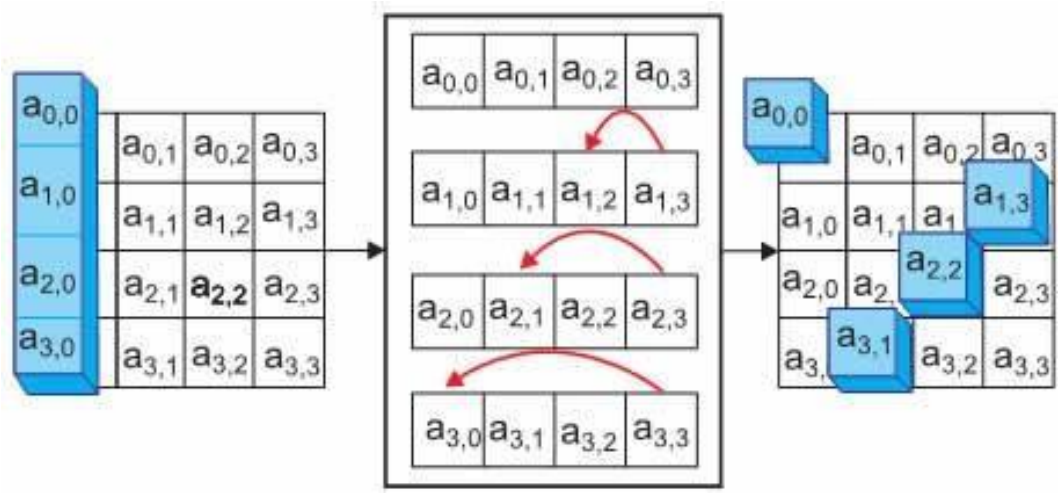


Рисунок 2.8 – Операція ззсуву рядків

2.9.4 Перемішування стовпців (операція «Mixcolumns»)

Операція перемішування стовпців, ймовірно, є найскладнішою з точки зору її програмної реалізації. На рис. 2.9 показано, як працює операція перемішування стовпців.

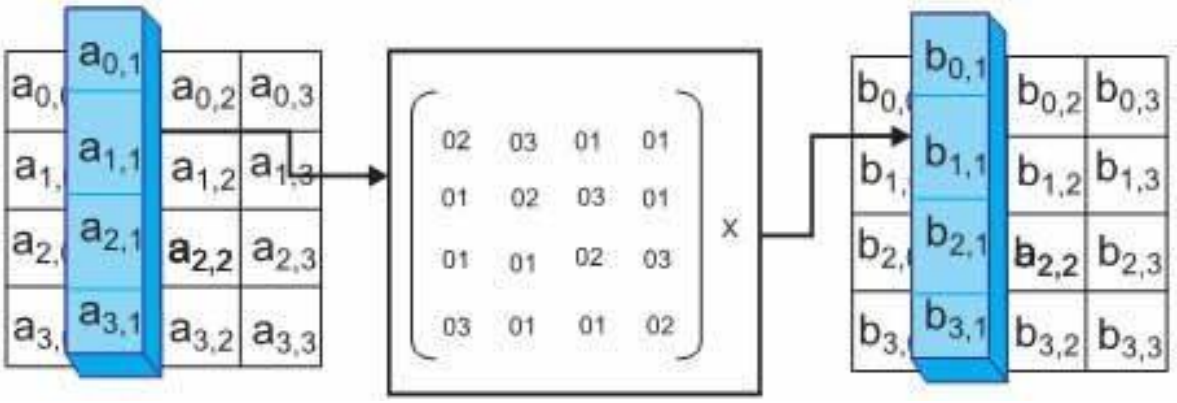


Рисунок 2.9 – Операція перемішування стовпців

Операція перемішування стовпців обробляє стовпці в матриці стану 4x4, помноживши стовпці стану на матрицю.

Ця операція зворотна, тому що в SHA використовуються операції в полі Галуа, де складання відповідає логічній операції того, що виключає АБО, а множення має складніший еквівалент.

Той факт, що в матриці множення операції перемішування стовпців багато членів виду « 01», полегшує реалізацію цієї операції на комп'ютері.

2.9.5 Додавання ключа раунду (операція «Addroundkey»)

Додавання ключа раунду полягає в операції XOR з відповідними байтами стану і отриманим ключем, як показано на рис. 2.10.

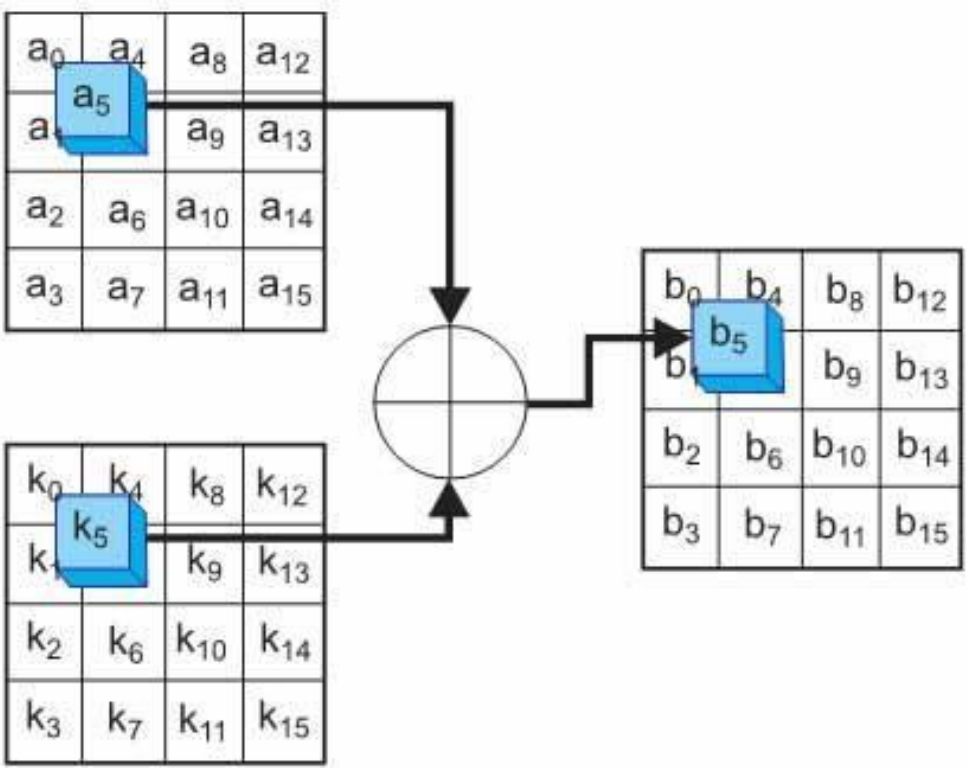


Рисунок 2.10 – Операція додавання ключа раунду

2.9.6 Завершення шифрування

Шифрування блокове, SHA шифрує рівно один блок вхідних байтів (для нашого варіанту з ключем в 128 біт це 16 байт), а оскільки розмір початкових даних не обов'язково кратний 16, врешті-решт, може залишитися блок, розміром від 1 до 15 байт, який не формує цілісний блок. Тоді використовується padding (доповнення до блоку). Цей останній блок можна,

не шифруючи, віддати на запис в кінцевий файл, але іноді, у кінці файлу міститься важлива інформація, в цьому випадку такий варіант не підходить.

Доповнення нульовими бітами не захищене від атак, до того ж одержувач не зможе знайти кінець корисних даних. Тому на практиці застосовне рішення, стандартизоване як «Метод доповнення 2» в ISO/IEC 9797-1, що додає одиничний біт в кінець повідомлення і заповнююче місце, що залишилося, нулями. В цьому випадку була доведена стійкість до подібних атак [22].

2.10 Висновки по розділу

У цьому розділі були розглянуті основні способи захисту програмного забезпечення ПЛІС, представлені в таблиці 2.1 від різного роду атак, а також їх слабкі місця і запропонована розроблена концепція захисту.

Таблиця 2.1 Порівняння методів захисту

Метод захисту	Опис	Зберігання і довжина ключа, біт	Використовуваний алгоритм шифрування	Атаки по аналізу живлення	Декомпіляція
Вбудований метод захисту	Апаратне дешифрування	Енергозалежна пам'ять, 256 біт	SHA - 256	Схильний	Побічно схильний
Вбудований дешифратор	Динамічна реконфігурація вентильної матриці	Енергозалежна пам'ять, 128 - 256 біт	SHA - 128 SHA - 192 SHA - 256	Є можливість протистояння	Побічно схильний

Метод захисту	Опис	Зберігання і довжина ключа, біт	Використовуваний алгоритм шифрування	Атаки по аналізу живлення	Декомпіляція
Використання мікросхем спеціальної пам'яті	Аутентифікація HMAC	На мікросхемі в секретній області EEPROM, на ПЛІС в конфігураційному файлі, 64 біта	SHA - 1 MAC	Є можливість протистояння	Схильний
Аутентифікація серійного номера	Контроль доступу до основної програми	У конфігураційному файлі, 128 біт	SHA - 128	Не схильний	Схильний

Етапи проектування пристрою в середовищі розробки WebPack ISE наведена в додатку В 08-36.МКР.012.00.000 ПЛ2.

Реалізація концепції IFF наведена в додатку Г 08-36.МКР.012.00.000 ПЛ3.

3 РОЗРОБКА МОДУЛЯ КРИПТОЗАХИСТУ ПЗ ПЛІС

3.1 Опис роботи модуля аутентифікації

Мною був розроблений метод захисту вбудованого ПЗ ПЛІС від клонування його на інші пристрої. Суть його полягає в перевірці серійного номера. На пристрої, окрім ПЛІС і ПЗП з конфігураційним файлом, також є присутнім додатковий модуль пам'яті, призначений для зберігання валідних серійних номерів в зашифрованому виді. Вони записуються в окрему мікросхему пам'яті для можливості модифікації списку в мікросхемі без зміни початкового коду і прошивки для ПЛІС. Це дозволяє без збитку конфігураційному файлу видаляти, доповнювати або змінювати список дозволених серійних номерів чіпів.

Основна ідея розробленої концепції полягає в аутентифікації серійного номера чіпа, на якому працює програма [23]. В результаті успішного проходження блоку аутентифікації, основна програма отримує дозвіл на виконання; якщо серійний номер не знайде збігу серед допустимих - вона не запуститься. Блок-схема повного алгоритму аутентифікації серійного номера представлена на рисунку 3.1.

Перш ніж отримати доступ до основної програми, необхідно отримати зашифровані дані про валідних номери, відправити ці дані і ключ шифрування на модуль aes_dec, що відповідає за дешифрування даних, і отримати назад розшифровані байти. Цей процес обміну даними можна розділити на декілька етапів:

Фаза 0: первинно, необхідно дозволити виконання модуля aes_dec (дешифрування), шляхом подання логічної одиниці на вхід CE (Clock Enabled дозвіл роботи лічильника). Далі слідує перехід до наступної фази - фазе1. Прапор S = «notready» - відповідає за результат програми; в даному випадку, це не відмова і не дозвіл роботи основної програми. Ділянка блок-схеми, що відповідає фазі 0, представлена на рис. 3.1.

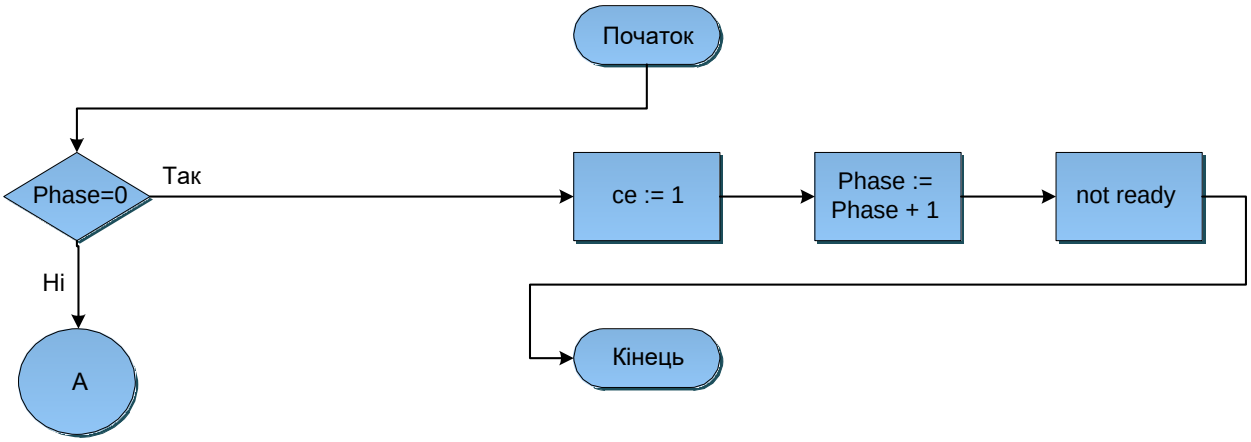


Рисунок 3.1 – Ділянка блок-схеми, що відповідає Фазі 0

Фаза 1: наступним кроком необхідно подати логічну одиницю на вхід reset, що приведе до очищення внутрішнього стану модуля aes_dec. Збільшуємо фазу і переходимо на неї. Ділянка блок-схеми, що відповідає фазі 1, представлена на рис.3.2

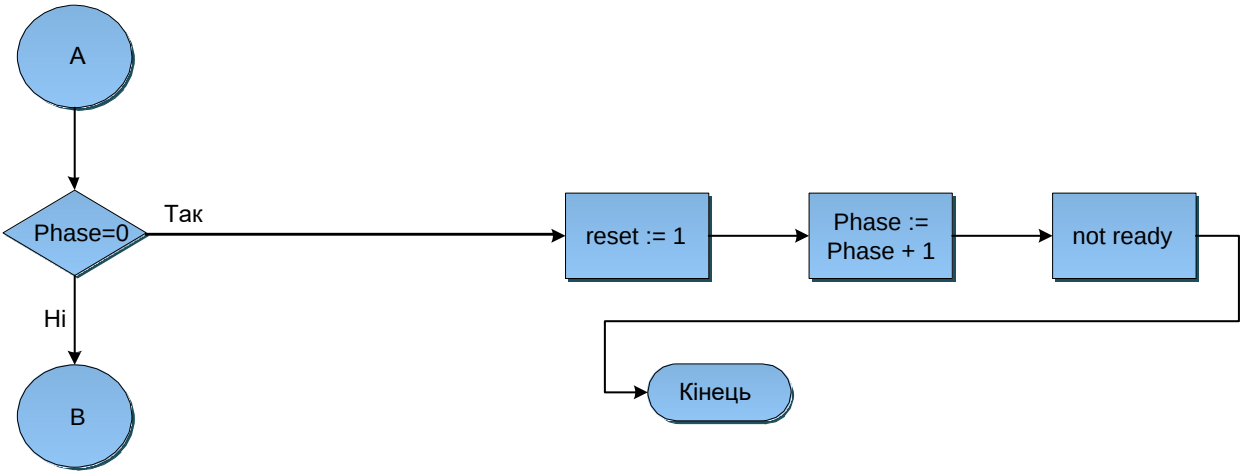


Рисунок 3.2 – Ділянка блок-схеми, що відповідає фазі 1

Фаза 2: повертаємо нуль на вхід reset модуля aes_dec, тепер модуль готовий до роботи - переходимо до наступної фази(збільшуємо на одиницю). Ділянка блок-схеми, що відповідає фазі 2, представлена на рис. 3.3.

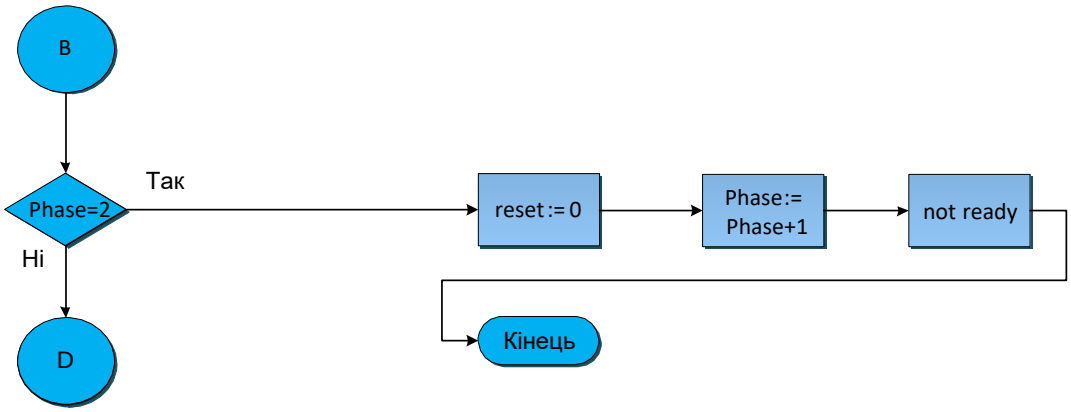


Рисунок 3.3 – Ділянка блок-схеми, що відповідає фазі 2

Фаза 3: фаза завантаження секретного ключа в модуль aes_dec.

Встановлюємо прапор valid_key в одиницю, що означає, що ключ шифрування по байту завантажується в модуль дешифрування, поки індекс поточного байта не перевищить значення, рівне максимальній довжині ключа. Довжина ключа в алгоритмі SHA - 128 равна128/8 = 16 байт, отже, на виконання завантаження ключа в модуль потрібно 16 тактів. Ділянка блок-схеми, що відповідає фазі 3, представлена на рис. 3.4.

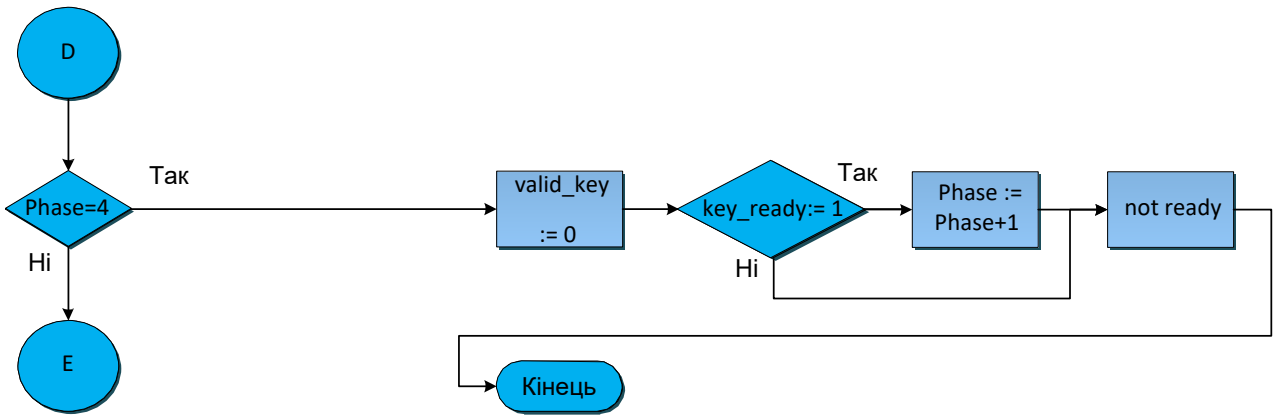


Рисунок 3.4 – Ділянка блок-схеми, що відповідає фазі 3

Фаза 4: завантаживши на попередньому етапі ключ шифрування в модуль aes_dec, чекаємо відповіді. Приймаюча сторона обробляє дані (ключ), формує 10 ключів раундів за допомогою операції «Key Expansion». Якщо прапор key_ready (готовність ключа) встановився в одиницю, то збільшивши фазу, переходимо далі. На роботу цієї ділянки потрібно 133 такти. Ділянка блок-схеми, що відповідає фазі 4, представлена на рис. 3.5.

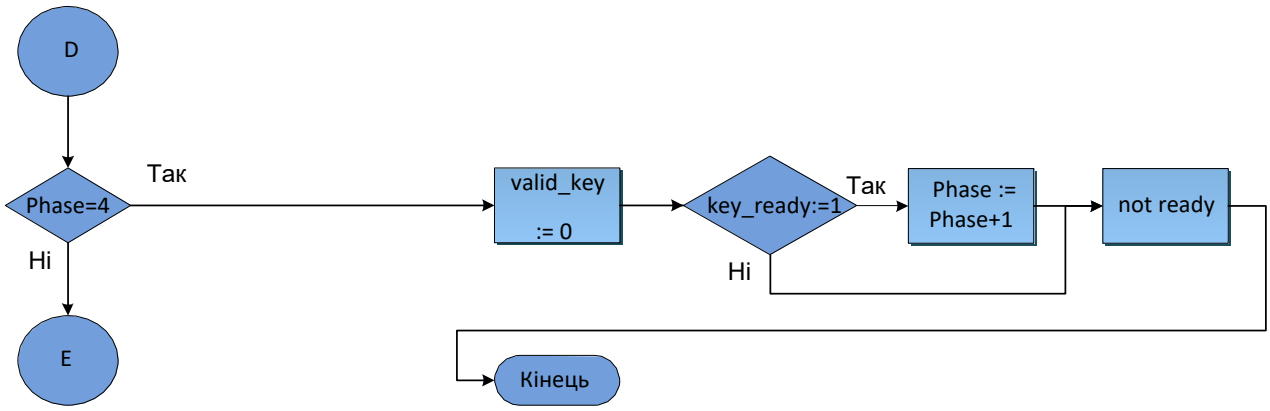


Рисунок 3.5 – Ділянка блок-схеми, що відповідає фазі 4

Фаза 5: Отримавши дані про те, що ключ успішно сформований, переходить до завантаження зашифрованого блоку даних в модуль aes_dec. Для дозволу завантаження необхідно встановити прапор valid_data в одиницю, поки індекс поточного байта в завантажуваному блоці не більший за максимально дозволене значення. Оскільки SHA- алгоритм шифрує дані блоками по 16 байт, те максимальне значення індексу і кількість тактів на проведення цієї операції відповідають значенню 16 . Далі перевіряється: чи останній блок даних завантажується. Якщо не останній - переходимо до наступної фази, фази очікування розшифрованого блоку даних, якщо останній - переходимо до фази 8, фазі очікування останнього розшифрованого блоку даних. Ділянка блок-схеми, що відповідає фазі 5, представлена на рис. 3.6.

Фаза 6, 7, 8,9 : ця одна фаза розділена логічно на 4:

Фаза 6: очікування дешифрування модулем aes_dec не останнього блоку даних. Коли прапор valid_out перейде в стан логічної одиниці, можна переходити до наступної фази. Дешифрування 16-байтного блоку даних в модулі aes_dec залежить від довжини ключа і в даному випадку складає 99 тактів.

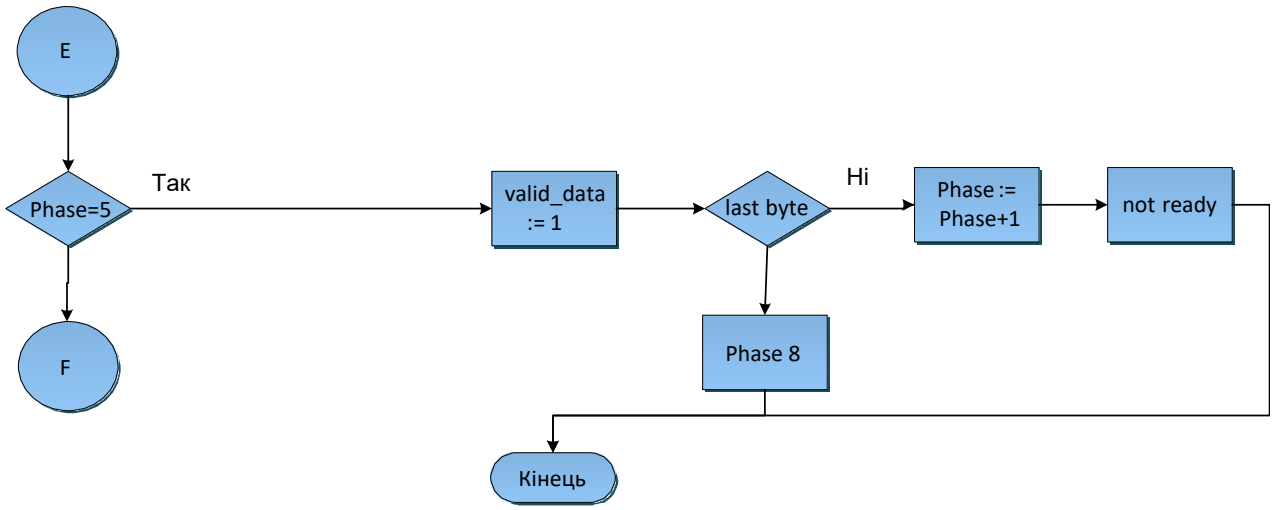


Рисунок 3.6 – Ділянка блок-схеми, що відповідає фазі 5

Фаза 7: прапор valid_out встановлений в одиницю - можна вивантажити з модуля aes_dec наступний розшифрований блок даних. На цій операції дані вивантажуються побайтно. Розмір блоку - 16 байт, відповідно, 16 тактів потрібні для виконання цієї фази. Байти заносяться в спеціальний масив для перевірки у фазі 10. Після читання останнього, 16-го байта з модуля aes_dec, прапор valid_out встановиться в нуль. Потім, програма знову перейде до фази 5 для завантаження наступного зашифрованого блоку даних.

Фаза 8: аналогічна фазі 6 з тією єдиною відмінністю, що в ній очікується дешифрування останнього блоку даних. Коли прапор valid_out перейде в стан логічної одиниці, станеться перехід до фази 9.

Фаза 9: аналогічна фазі 7 з тією єдиною відмінністю, що в ній здійснюється дешифрування останнього блоку даних. На цій фазі розшифровка завершується, і програма переходить до фази 10.

Ділянка блок-схеми, що відповідає сукупності фаз 6, 7, 8, 9, представлений на рис. 3.7.

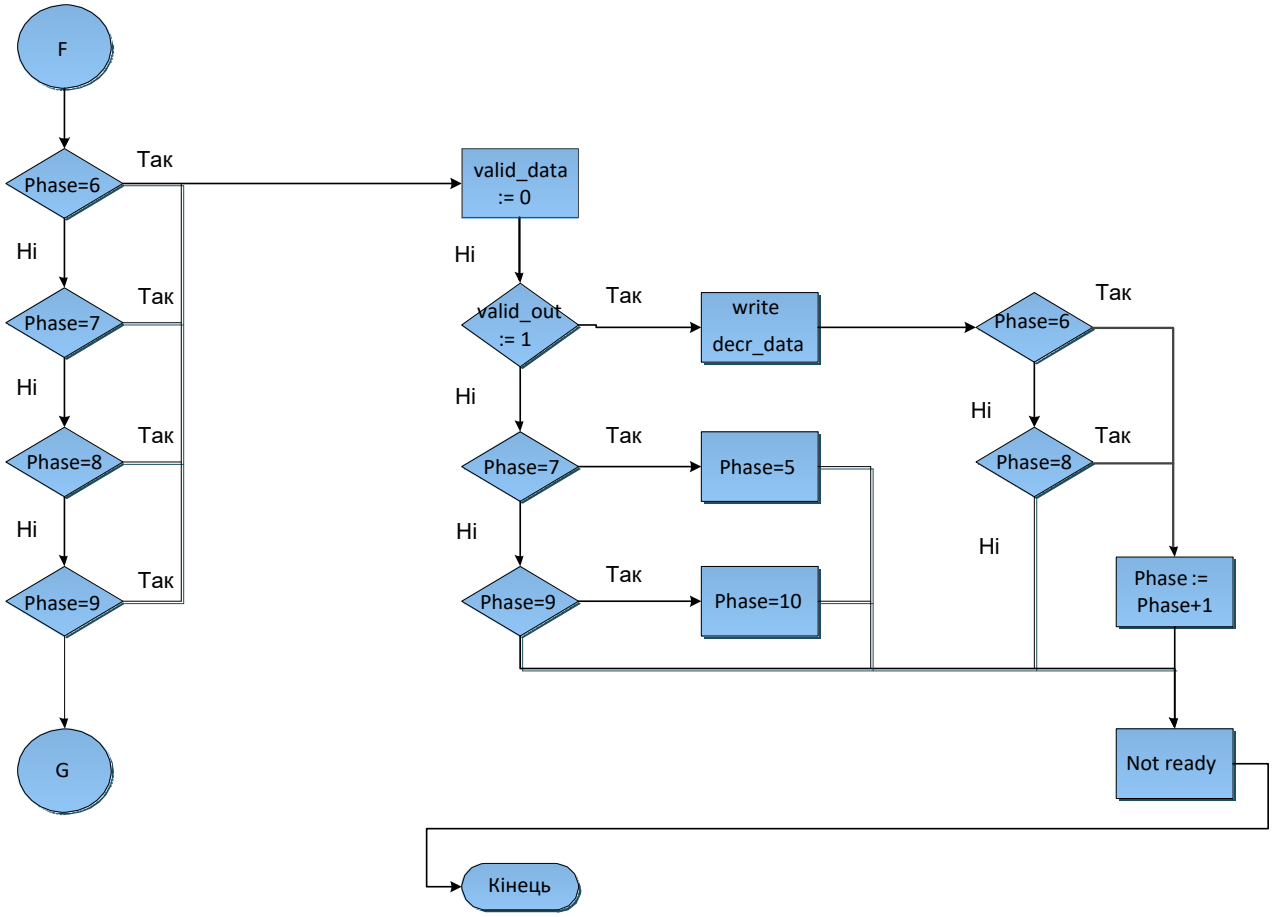


Рисунок 3.7 – Ділянка блок-схеми, що відповідає сукупності фаз 6, 7, 8, 9

Фаза 10: завершальна фаза блоку аутентифікації серійного номера ПЛІС. Модуль aes_dec більше не потрібно, тому вхід CE (Clock Enabled) встановлюється в нуль. У цій фазі здійснюється побайтне порівняння серійного номера і елементів масиву з розшифрованими даними. Для перевірки одного елементу масиву (8 байт) потрібно один такт. Якщо перевірка пройшла успішно, здійснюється перехід до наступної фази, якщо ні, індекс масиву збільшується, і в наступному такті серійний номер порівнюється з наступним елементом. Тривалість цієї фази залежить від індексу знайденого серійного номера в масиві, і, у гіршому разі, кількість тактів дорівнюватиме кількості елементів в масиву (1000 у рамках цієї роботи). Ділянка блок-схеми, що відповідає фазі 10, представлена на рис. 3.8.

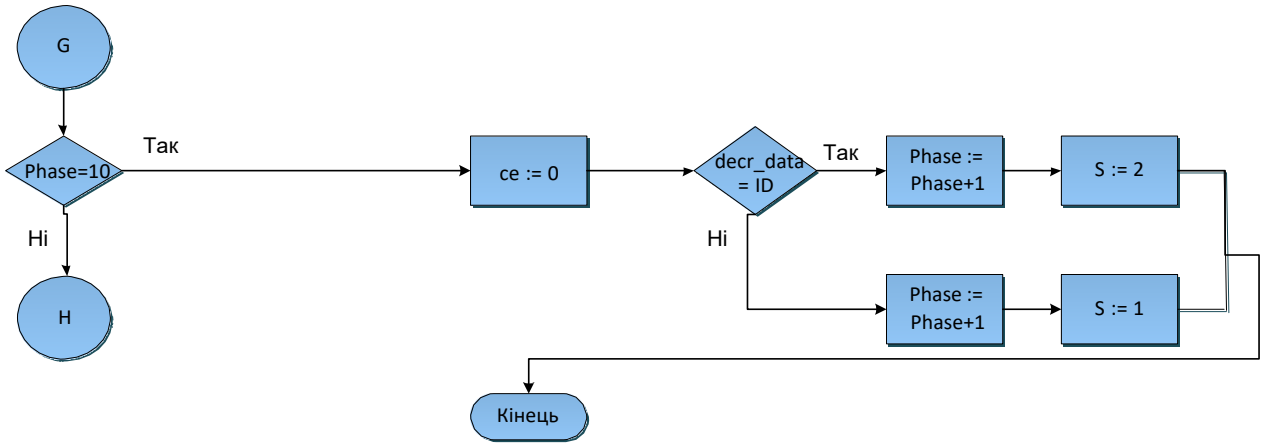


Рисунок 3.8 – Ділянка блок-схеми, що відповідає фазі 10

Фаза 11: блок аутентифікації завершив роботу по дешифруванню і пошуку серійного номера в масиві. Якщо перевірка пройшла успішно, основна програма починає роботу, інакше, виводиться повідомлення про помилку і основна програма не отримує дозвіл на виконання. Ділянка блок-схеми, що відповідає фазі 11, представлена на рис. 3.9 [24].

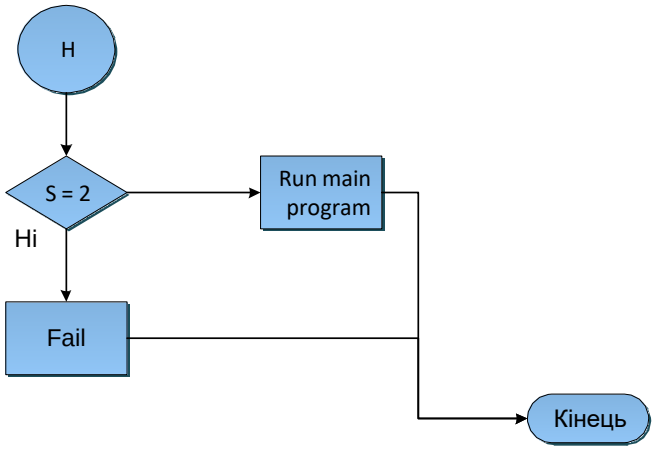


Рисунок 3.9 – Ділянка блок-схеми, що відповідає фазі 11

3.2 Атаки і рівень криптостійкости системи

Оскільки здійснюється контроль доступу до програмного забезпечення за допомогою порівняння використовуваного серійного номера пристрою з тими, на яких доступне використання цього ПЗ, у зловмисника є два напрями для атаки:

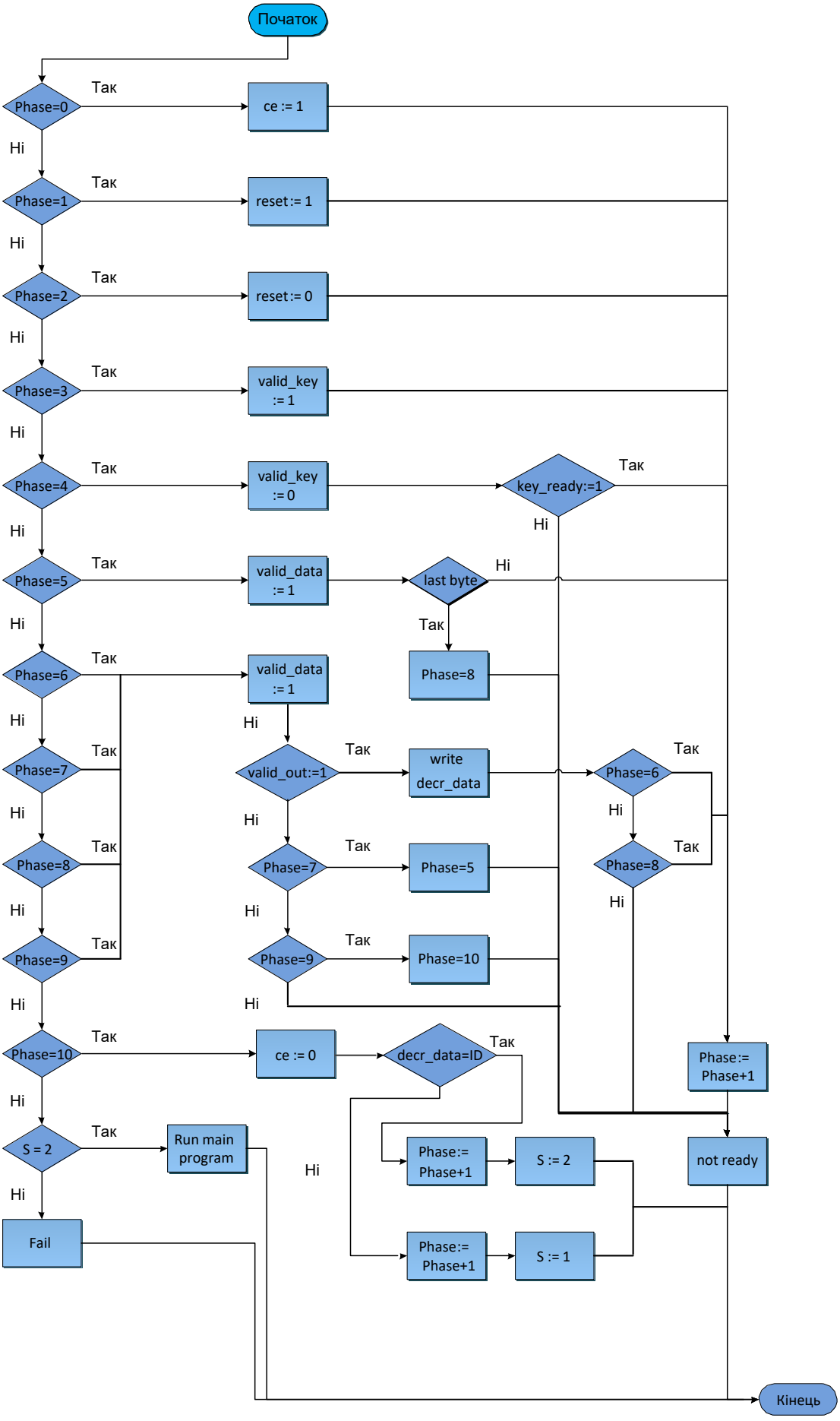


Рисунок 3.10 – Алгоритм аутентифікації серійного номера

– Модифікувати список вірних серійних номерів, що зберігаються в зовнішній пам'яті ПЛІС, чого він не може зробити, не знаючи секретного ключа;

– Поміняти ключ, перезаписавши вміст енергозалежної пам'яті;

Якщо ключ зберігатиметься в енергозалежній пам'яті, то, попри те, що читання не доступне в цьому регістрі пам'яті, зловмисник може перезаписати цей регістр своїм ключем, зашифрувати потрібний йому список серійних номерів і прошити їх на мікросхему пам'яті. Таким чином, йому вдасться пройти аутентифікацію. Тому в розробленій концепції ключ зберігається безпосередньо в початковому коді програми, що сильно ускладнює завдання підміни секретного ключа.

Для захисту від атак по сторонніх каналах був розроблений паралельний процес генерації псевдовипадкових чисел, на кожному такті, це утрудняє знаходження секретного ключа по викидах електроживлення завдяки внесенню шуму у виміри напруги.

Ця концепція не захищена від декомпіляції конфігураційного файлу : видалення місця перевірки серійного номера, зчитування ключа дешифрування.

3.3 Висновки по розділу

У цьому розділі розглянутий реалізований метод захисту програмного забезпечення з описом його роботи, приведені і проаналізовані можливі атаки на цей метод.

Структура алгоритму шифрування наведена в додатку Д 08-36.МКР.012.00.000 ПЛ4.

Опис алгоритму модуля аутентифікації наведена в додатку Е 08-36.МКР.012.00.000 ПЛ5.

4 ОПИС ПЕРЕВІРКИ ПРАЦЕЗДАТНОСТІ І РЕЗУЛЬТАТИ СИМУЛЯЦІЇ

Після завершення розробки модуля криптозахисту програмного забезпечення його необхідно протестувати. За допомогою середовища розробки Xilinx було змодельоване поведінка модуля в різних ситуаціях. У разі успішної аутентифікації на індикатор повинне виводитися число 2, що є дозволом виконання основної програми, інакше 1 - серійний номер цього пристрою не знайдений в числі допустимих серійних номерів і забороняється перехід до роботи основної програми. Таким чином, здійснюється контроль використання цього ПЗ тільки на облаштуваннях розробника.

Результат симуляції є вікном з часовими діаграмами вибраних сигналів. Розглянемо основні сигнали, що пояснюють роботу модуля :

– Clock (тактовий сигнал, clock signal) сигнал синхронізації;

– Y два семисегментні індикатори, які виводять або 01(«011111100001102»), або 02(«011111110110112»);

– Phase поточний етап виконання модуля;

– Valid_data режим завантаження блоку даних в модуль дешифрування (0 - завантаження завершене, 1 - процес завантаження);

– Valid_out режим читання розшифрованих даних з модуля дешифрування(0 - читання завершене, 1 - процес читання);

– Valid_key режим завантаження секретного ключа в модуль дешифрування(0 - завантаження завершене, 1 – процес завантаження);

– Key_ready очікування готовності формування ключів раундів на модулі дешифрування(0 - ключі сформовані, 1 - процес формування ключів);

– CE сигнал дозволу роботи модуля дешифрування;

– Reset сигнал очищення модуля дешифрування;

4.1 Перевірка фаз роботи пристрою

В якості тесту роботи програми на вхід було записано валідне значення серійного номера, отже, на індикаторі чекаємо число «02». На рис.4.1 спостерігається наростання сигналу CE по передньому фронту сигналу на вході clock, phase набув значення 12. Наступні два такти змінюють стан reset для очищення входів aes_dec, їм відповідають phase 102 і phase 112. Далі на вхід сигналу valid_key подається логічна одиниця для можливості завантажити ключ шифрування в модуль aes_dec (завантаження побайтне, тому на один такт доводиться завантаження одного байта ключа).

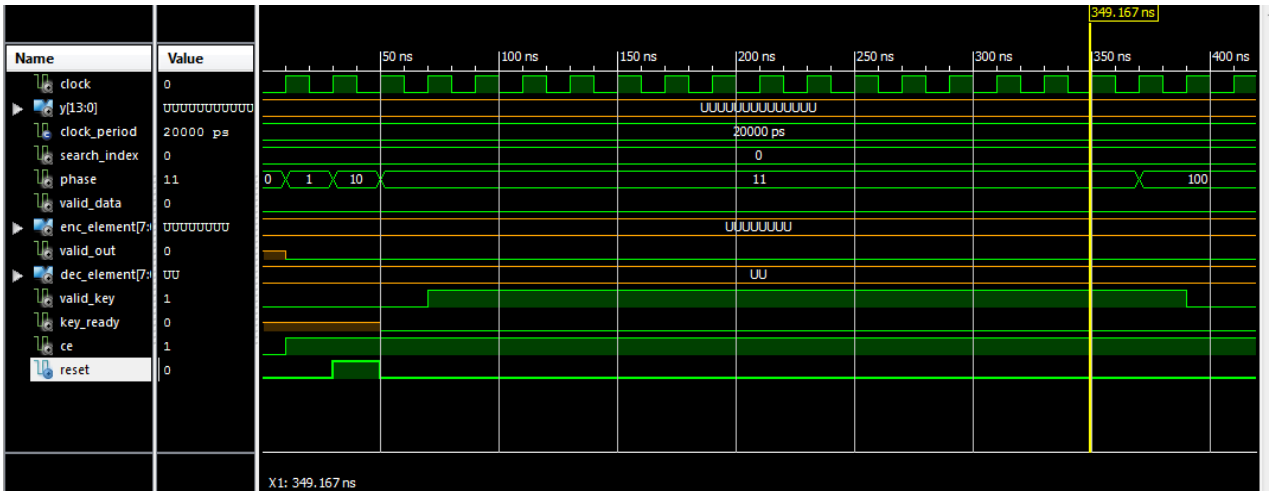


Рисунок 4.1 – Перші три фази роботи модуля

Після завантаження ключа фаза збільшується на одиницю, як показано на рис. 4.2. на цьому етапі очікується формування ключів на стороні модуля aes_dec; з перемиканням сигналу key_ready в одиницю цей етап завершується і значення фази збільшується на одиницю одночасно з поданням на вхід сигналу valid_data одиниці для завантаження блоку даних. Як тільки дані завантажилися, здійснюється перехід до фази 1102 - очікування дешифрування блоку, з подальшим переходом до фази 1112, де розшифровані дані записуються в масив. Ці три етапи повторюються до тих пір, поки у вхідному масиві залишаються зашифровані байти.

4.2 Перевірка етапів циклічного запису даних з модуля дешифрування

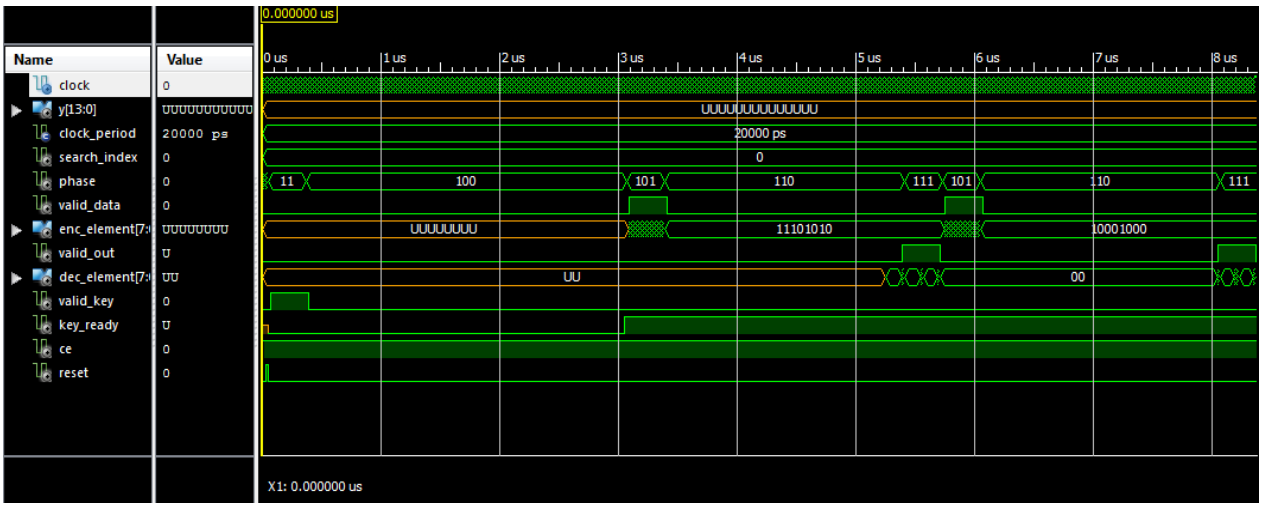


Рисунок 4.2 – Етапи циклічного запису даних з модуля дешифрування

На рис.4.3 відбитий час дешифрування даних 1,333мс. На цьому етапі дешифрування завершене, CE встановлюється в нуль, оскільки модуль дешифрування більше не потрібний, і починається пошук серійного номера серед розшифрованих даних.

4.3 Перевірка вірності функціонування процесу пошуку серійного номера

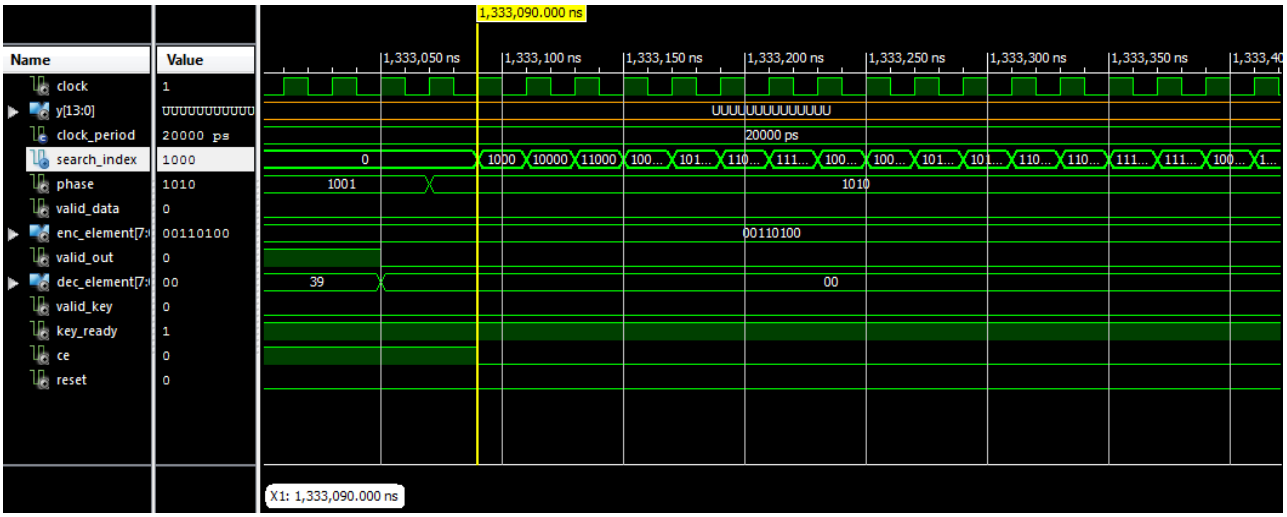


Рисунок 4.3 – Пошук серійного номера

Вихід індикатора видає - 011111110110112, що відповідає «02» успішних аутентифікації на 1,351мс, як показано на рис. 4.4. Отже, перевірка займає 0,018мс.

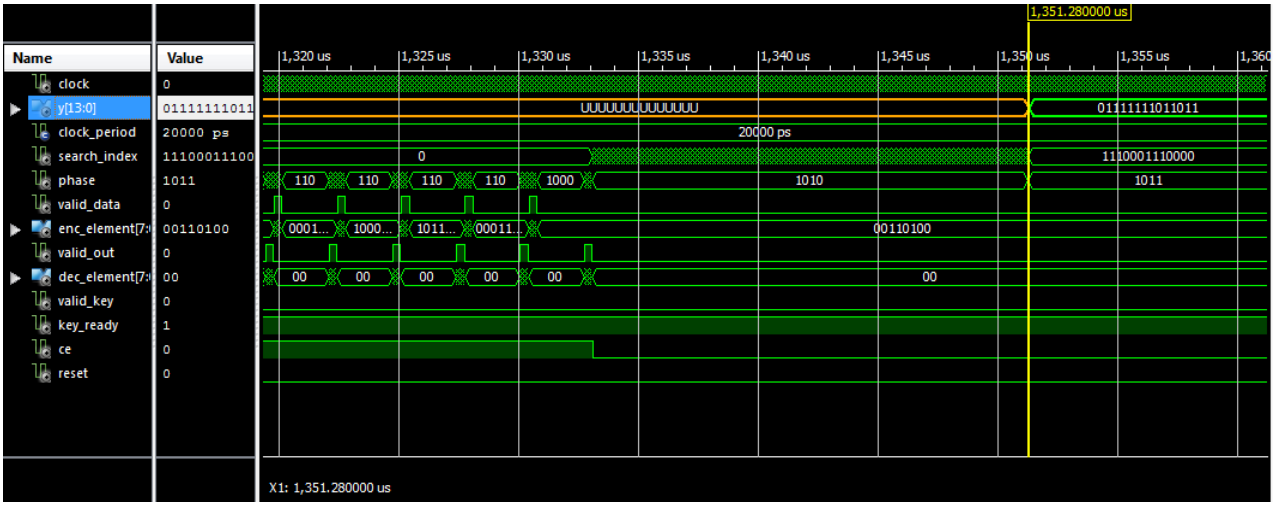


Рисунок 4.4 – Результат роботи модуля

4.4 Генерація випадкових чисел

Генерація випадкових чисел показана на тимчасовій діаграмі з міткою randd на рис. 4.5.

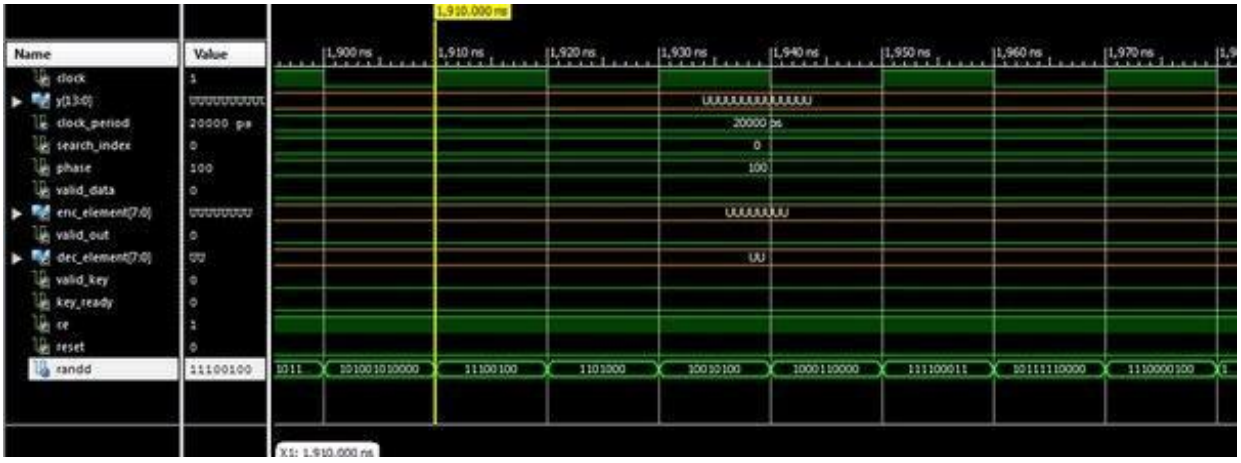


Рисунок 4.5 – Генерація випадкових чисел

Опис перевірки працездатності і результати симуляції наведена в додатку Ж 08-36.МКР.012.00.000 ПЛ6.

5 ЕКОНОМІЧНА ЧАСТИНА

В техніко-економічному обґрунтуванні представленому в першому розділі даної магістерської кваліфікаційної роботи було приблизно обґрунтовано доцільність проведення НДДКР. Тому в даному розділі будуть проведені більш детальні розрахунки витрат на проведення НДДКР стосовно дослідження з розробки апаратної платформи для реалізації SHA-алгоритмів на FPGA.

Для економічного розрахунку проведення НДДКР потрібно скласти кошторис витрат, який передбачає розрахунок визначених основних статей витрат.

Основна заробітна плата дослідників та розробників, яка розраховується за формулою [**10**]

$$З_о = \frac{M}{T_р} \cdot t \quad , \tag{5.1}$$

де M – місячний посадовий оклад конкретного розробника (дослідника), грн.;

$T_р$ – число робочих днів в місяці, 22 дн;

t – число днів роботи розробника (дослідника).

Проведені розрахунки зводимо до таблиці.

Таблиця 5.1 – Основна заробітна плата дослідників та розробників

Найменування посади	Місячний посадовий оклад, грн.	Оплата за робочий день, грн.	Число днів роботи	Витрати на заробітну плату, грн.
1. Керівник проекту	12400,00	563,64	52	29309,09
2. Консультатнт-розробник архітектури ЕОМ	10100,00	459,09	15	6886,36
3. Інженер-схемотехнік	9200,00	418,18	35	14636,36
4. Інженер-конструктор радіоелектронних систем	9000,00	409,09	40	16363,64
5. Лаборант	5200,00	236,36	45	10636,36
Разом				77831,82

Витрати на основну заробітну плату працівників (Зр), що здійснюють підготовку робочих місць необхідних для дослідження з розробки апаратної платформи для реалізації SHA-алгоритмів на FPGA, підготовку та формування баз даних, підготовку та монтаж обладнання, макетів, виготовлення дослідних зразків тощо, розраховуються на основі норм часу, які необхідні для виконання даної роботи, за формулою **[10]**

$$З_р = \sum_{1}^{n} t_i \cdot С_i \cdot К_с \quad , \qquad (5.2)$$

де t_i - норма часу (трудомісткість) на виконання конкретної роботи, годин;

n - число робіт по видах та розрядах;

$К_с$ - коефіцієнт співвідношень, який установлений в даний час Генеральною тарифною угодою між Урядом України і профспілками, $К_с$ = 1,75;

Ci - погодинна тарифна ставка робітника відповідного розряду, який виконує відповідну роботу, грн./год.

C_i визначається за формулою **[10]**

$$С_i = \frac{М_м \cdot К_i}{Т_р \cdot Т_{зм}}, \qquad (5.3)$$

де, $М_н$ – прожитковий мінімум працездатної особи, грн., $М_н$ = 2197,00 грн.;

$К_i$ - тарифний коефіцієнт робітника відповідного розряду;

$Т_р$ - число робочих днів в місяці, $Т_р$ = 22 дн;

$Т_{зм}$ - тривалість зміни, $Т_{зм}$ = 8 годин.

Проведені розрахунки винесемо до таблиці 5.2.

Таблиця 5.2 – Витрати на основну заробітну плату працівників

Найменування робіт	Трудомісткість, нормо-годин	Розряд роботи	Тарифний коефіцієнт	Погодинна тарифна ставка, грн.	Величина оплати, грн.
1. Встановлення допоміжного обладнання	11,0	2	1,1	24,03	264,33
2. Інсталяція програмного забезпечення	9,5	4	1,35	29,49	280,16
3. Встановлення модулів апаратних платформ FPGA: Stratix III і Stratix IV	20,0	3	1,2	26,21	524,28
4. Монтаж імітаторів апаратної платформи для реалізації SHA-алгоритмів на FPGA	18,0	5	1,7	37,14	668,46
5. Формування бази даних результатів вимірювань	124,0	3	1,2	26,21	3250,56
6. Підготовка лабораторії	16,0	2	1,1	24,03	384,48
Разом					5372,27

Додаткова заробітна плата розробників, дослідників та працівників, які приймали участь в дослідженнях та розробці НДДКР розраховується як 11% від основної заробітної плати розробників та працівників

$$З_д = З_о \text{ *· } 11 / 100\% , \quad (5.4)$$

$$З_д = (77831,82 + 5372,27) \text{ *· } 11 / 100\ \% = 9152,45 \text{ (грн.)}.$$

Нарахування на заробітну плату дослідників та працівників.

Згідно діючого законодавства нарахування на заробітну плату складають 22% від суми основної та додаткової заробітної плати

$$Н_з = (З_о + З_д) \cdot\text{* } 22\% / 100\%, \quad (5.5)$$

$H_з$ = (77831,82+ 5372,27+ 9152,45) * 22% / 100% = 20318,44 (грн.).

Витрати на матеріали на даному етапі проведення НДДКР пов'язані з використанням моделей елементів та моделювання роботи і досліджень за допомогою комп'ютерної техніки та створення експериментальних блоків і компонентів, тому дані витрати формуються на основі як офісних витратних матеріалів так і обмеженого переліку матеріалів.

Витрати на матеріали, що були використані при проведенні досліджень, розраховуються по кожному виду матеріалів за формулою **[10]**

$$M = \sum_{1}^{n} H_i \cdot Ц_i \cdot K_i \quad , \qquad (5.6)$$

де, - H_i - витрати матеріалу *і*-го найменування, кг;

$Ц_i$ - вартість матеріалу *і*-го найменування, грн./кг.;

K_i - коефіцієнт транспортних витрат, $К_і$ = 1,1;

n - кількість видів матеріалів,

Проведені розрахунки зводимо до таблиці 5.3

Таблиця 5.3 – Витрати на основні матеріали

Найменування матеріалу, марка, тип, сорт	Одиниця виміру	Ціна за одиницю, грн.	Витрачено	Вартість витраченого матеріалу, грн.
Папір офісний	уп.	81,00	6	486,00
Диск оптичний	шт.	10,50	6	63,00
Канцтовари	компл.	200,00	6	1200,00
Тека пластикова	шт	110,00	6	660,00
Картридж Canon 23-12X	шт	345,00	2	690,00
Блок памяті 32 Gb	шт	299,00	2	598,00
Бензосуміш	кг	40,00	0,25	10,00
Лак	кг	135,00	0,1	13,50
Склотекстоліт СФ-2Н-35	кг	152,00	0,2	30,40
Смола поліамідна 68С	кг	68,70	0,15	10,31
Пресматеріал К- 124-38	кг	74,30	0,1	7,43
Стержні текстолітові	кг	110,00	0,2	22,00
Стрічка поліхлорвінілова ізоляційна	кг	105,00	0,08	8,40
Всього				3799,04

Таблиця 5.4 – Витрати на комплектуючі для формування компонентів для НДДКР

Найменування комплектуючих	Кількість, шт.	Ціна за штуку, грн.	Сума, грн.
1. Варикапи			
КВ130А	2	2,00	4,00
2. Мікросхеми			
LM26	1	60,00	60,00
НІН-4000	1	160,00	160,00
MPX4115	1	148,00	148,00
AD1B60BJ	1	220,00	220,00
ATtiny25	1	26,00	26,00
3. Конденсатори			
К10-36-50В±20%	2	1,20	2,40
К10-19-32В±20%	9	2,30	20,70
К10-23-16В±20%	4	2,30	9,20
КПЕ-30В±20%	1	8,40	8,40
К10-40-50В±20%	1	2,40	2,40
4. Кварц			
ГК-163	1	30,00	30,00
РВ-59	1	15,00	15,00
5. Резистори			
С2-10± 5%	9	1,00	9,00
С2-23± 5%	8	2,10	16,80
С2-11± 5%	7	1,30	9,10
6. Роз’єми			
ОНП-КГ-4-5/16-Р	2	2,20	4,40
7. Транзистори			
MRF962	1	6,00	6,00
2N3600	4	5,40	21,60
8. Індуктивність	9	7,30	65,70
Всього			838,70

З врахуванням транспортних витрат вартість матеріалів складе

$$М = 3799,04 * 1,1 = 4178,94 \text{ грн.}$$

Витрати на комплектуючі (основне обладнання, емулятори, моделі, комплектуючі макетів), що були використані при дослідженні з розробки апаратної платформи для реалізації SHA-алгоритмів на FPGA, розраховуються за формулою

$$Н = \sum_{1}^{n} Н_i \cdot Ц_i \cdot К_i \,, \qquad (5.7)$$

де: *Нi* - кількість комплектуючих *і*-го виду, шт.;

Цi - покупна ціна комплектуючих *і*-го виду, грн.;

$К_i$ - коефіцієнт транспортних витрат, $К_i = 1{,}10$;

n - кількість видів матеріалів.

Проведені розрахунки зводимо до таблиці.

Витрати на комплектуючі з урахуванням транспортних витрат складають

$$H = 838{,}70 \text{ *} \cdot 1{,}10 = 922{,}57 \text{ (грн.)}.$$

Амортизація обладнання для проведення досліджень

В спрощеному вигляді амортизаційні відрахування по кожному виду обладнання, приміщень та програмному забезпеченню можуть бути розраховані з використанням прямолінійного методу амортизації за формулою

$$A_{обл} = \frac{Ц_б}{T_в} \cdot \frac{t_{вик}}{12}, \qquad (5.8)$$

де $Ц_б$ – балансова вартість обладнання, приміщень тощо, які використовувались для розробки нового технічного рішення, грн.;

$t_{вик}$ – термін використання обладнання, приміщень під час розробки, місяців;

$T_в$ – строк корисного використання обладнання, приміщень тощо, років.

Проведені розрахунки необхідно звести до таблиці 5.5

Таблиця 5.5 - Величина амортизаційних відрахувань

Найменування обладнання	Балансова вартість, грн	Строк корисного використання, років	Термін використання обладнання, міс.	Величина амортиза-ційних відрахувань, грн
Аналітичний комплекс обробки даних	32000,00	5	2	1066,67
Система метрологічного аналізу	27000,00	5	2	900,00
Апаратна платформа для реалізації SHA-алгоритмів на FPGA	21000,00	4	2	875,00
Повноциклова автоматизована система проектування	32800,00	5	2	1093,33
Оргтехніка	15000,00	4	2	625,00
Приміщення лабораторії	330000,00	25	2	2200,00
Всього				6760,00

Витрати на силову електроенергію на проведення досліджень розраховують за формулою **[10]**

$$В_е = В \cdot П \cdot Ф \cdot К_п, \quad (5.9)$$

де, *В* – вартість 1 кВт-години електроенергії, *В* = 2,91 грн./кВт –година;

П – встановлена потужність обладнання, кВт.;

Ф – фактична кількість годин роботи обладнання, годин. ;

$К_п$ – коефіцієнт використання потужності.

Всі проведені розрахунки зведемо до таблиці.

Таблиця 5.6 – Витрати на електроенергію при проведенні досліджень

Найменування обладнання	Кількість годин роботи обладнання, год.	Встановлена потужність, кВт	Коефіцієнт використання потужності	Величина оплати
Аналітичний комплекс обробки даних	400	0,65	0,93	703,64
Система метрологічного аналізу	300	0,35	0,93	284,16
Апаратна платформа для реалізації SHA-алгоритмів на FPGA	300	0,32	0,93	259,80
Повноциклова автоматизована система проектування	350	0,65	0,93	615,68
Оргтехніка	200	0,86	0,93	465,48
Всього				2328,77

Інші витрати охоплюють: загальновиробничі витрати, адміністративні витрати, витрати на відрядження, матеріали, окремі непередбачені витрати, зв’язок, витрати на інтернет-послуги тощо.

Інші витрати доцільно приймати як 200...300% від суми основної заробітної плати дослідників та робітників. Величина інших витрат складе

$$I = (77831,82 + 5372,27) * 250\% / 100\% = 208010,23 \text{ (грн.)}.$$

Загальні витрати на проведення науково-дослідної роботи.

Сума всіх попередніх статей витрат дає загальні витрати на проведення науково-дослідної роботи

В =77831,82+5372,27+9152,45+20318,44+3799,04+922,57+6760,00+2328,77+
+208010,23=334495,59 (грн.).

5.1 Визначення коефіцієнта наукової значимості отриманих результатів науково-дослідної роботи

Коефіцієнт наукової значимості результатів проведеної НДР $K_{ЗН}$ можна підрахувати за формулою

$$K_{ЗН} = \frac{\sum_{1}^{3} b_i \cdot d_i}{\sum_{1}^{3} b_{\max} \cdot d_i}, \qquad (5.10)$$

де b_i - значимість отриманих результатів: b_1 - ступінь наукової новизни, b_2 - рівень теоретичної обґрунтованості, b_3 - ступінь експериментальної перевірки результатів.

Бальна оцінка отриманих результатів наведена в таблиці .

Максимальне значення отриманих результатів можна прийняти в межах 7…10 балів;

d_i - питома вага кожної характеристики, значення якої наведено в таблиці;

З – кількість характеристик, за якими була зроблена оцінка результатів науково-дослідної роботи.

Таблиця 5.7 – Показники для оцінювання наукової значимості результатів виконання НДР

Характеристики	Питома вага характеристик	Бальна оцінка характеристик		
		Ступінь новизни b_1	Рівень теоретичної обґрунтованості b_2	Ступінь експериментальної перевірки результатів b_3
		1	3…5	7…10
b_1	0,500	Часткове удосконалення виробів, технологій, матеріалів, програмного продукту, тощо	Суттєве удосконалення виробів, технологій, матеріалів, програмного продукту, тощо	Нові напрямки в розробці виробів, технологій, матеріалів, програмного продукту, тощо. Створення принципово нової техніки
b_2	0,333	Позитивне рішення на основі зроблених узагальнень	Установлення залежностей, які використовувались в інших випадках	Відкриття нових шляхів рішення задачі
b_3	0,167	Експериментальна перевірка не робилась	Результати перевірялись на невеликій кількості даних	Результати перевірені на великій кількості даних

Підставляючи числові дані $d_1 = 0{,}5$, $d_2 = 0{,}333$, $d_3 = 0{,}167$, b_1= 7, b_2= 7, b_3= 7, $b_{\max} = 10$ у вираз ( ) оцінимо наукову значимість отриманих результатів

$$K_{ЗН} = \frac{b_1 \cdot 0{,}5 + b_2 \cdot 0{,}333 + b_3 \cdot 0{,}167}{10 \cdot 0{,}5 + 10 \cdot 0{,}333 + 10 \cdot 0{,}167} = 0{,}70.$$

5.2 Внесок дослідника в досягнення отриманих результатів НДР

Внесок дослідника в досягнення отриманих результатів НДР можна розрахувати за формулою

$$V = \frac{k_{TBI} \cdot З_i}{\sum_{1}^{n} k_{TBI} \cdot З_i}, \qquad (5.11)$$

де k_{TBI} - коефіцієнт творчої участі кожного виконавця НДР, який оцінюється наступним чином: проведення досліджень – 3 бали, робоче проектування – 1,5 бали, освоєння – 1,0 балів.

Якщо виконавець приймав участь в декількох видах робіт, то береться сума відповідних балів;

$З_i$ - заробітна плата кожного виконавця НДР;

n - кількість всіх виконавців НДР

Таблиця 5.8 – Показники для оцінювання наукової значимості результатів виконання НДР

Найменування посади	Місячний посадовий оклад, грн.	Коефіцієнт творчої участі
1. Керівник проекту	12400,00	3
2. Консультант-розробник архітектури ЕОМ	10100,00	3
3. Інженер-схемотехнік	9200,00	1,5
4. Інженер-конструктор радіоелектронних систем ***(проектувальник)***	9000,00	3
5. Лаборант	5200,00	1

Розраховуємо внесок дослідника

*V= (3*9000,00) / (3*12400,00+3*10100,00+1,5*9200,00+1*5200,00= 0,31.*

5.3 Висновки до економічного розділу

Загалом запланована науково-дослідна робота з проведення дослідження з розробки апаратної платформи для реалізації SHA-алгоритмів на FPGA вимагає грошового вкладення для виконання в межах 334495,59 грн.

Отримані результати досліджень мають високий рівень наукової значимості (в межах 0,70), що свідчить про доцільність проведення розробок та значимість науково-дослідної роботи з дослідження з розробки апаратної платформи для реалізації SHA-алгоритмів на FPGA.

6 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

В даній частині розділу необхідно провести дослідження безпеки роботи апаратної платформи для реалізації SHA-алгоритмів в умовах дії іонізуючих випромінювань та електромагнітного імпульсу.

Вплив радіації призводить як до оборотних, так і необоротних змін електричних властивостей твердотільних приладів і інтегральних схем. Оскільки такі зміни можуть приводити до відмов електронних підсистем, значні зусилля останнім часом направляються на розробку методів, що дозволяють уникнути погіршення параметрів мікроелектронного обладнання при опроміненні. У багатьох випадках вирішувати цю проблему доцільно шляхом створення радіаційно-стійких приладів і інтегральних схем [27].

До часток, які при влученні в прилади й схеми можуть викликати небажані наслідки, відносяться електрони, протони, фотони, альфа-частинки, важкі іони. Основні наслідки впливу опромінення на електронні пристрої – іонізація і зсув атомів. Вони викликають різні зміни в напівпровідникових приладах.

При бомбардуванні кремнію фотони й заряджені частинки можуть викликати зсув атомів з положення рівноваги. У випадку фотонів цей процес реалізується за рахунок появи комптоновських електронів з досить великою енергією, які потім взаємодіють із ядрами мішені. Поки ядрам передається мінімальна енергія (для кремнію 21 еВ), буде відбуватися зсув атомів. Відсутність атома у своєму нормальному положенні – це перший вид радіаційного дефекту.

В опроміненому кремнії можуть зустрічатися різні типи дефектів. Елементарні дефекти іноді називають точковими або ізольованими. Можливе виникнення областей з більшим числом досить близько розташованих дефектів. Така область називається кластером дефектів або каскадом зсувів. Згідно з фізикою напівпровідників, атоми домішки в решітках кремнію мають дискретні рівні енергії, які лежать у забороненій зоні між мінімумом зони провідності й максимумом валентної зони. Порушення періодичності атомної решітки може привести до виникнення деяких рівнів енергії в забороненій зоні. До їхньої появи приводять, зокрема, радіаційні дефекти, і саме ці дефектні стани або центри впливають на електричні властивості напівпровідникових приладів. Тому є необхідним провести дослідження стійкості роботи апаратної платформи саме при впливах іонізуючих випромінюваннях та електромагнітному імпульсі.

6.1 Технічні рішення з гігієни праці та виробничої санітарії

6.1.1 Дослідження безпеки роботи апаратної платформи для реалізації SHA-алгоритмів в умовах дії іонізуючих випромінювань

Дослідження безпеки роботи апаратної платформи для реалізації SHA-алгоритмів в умовах дії іонізуючих випромінювань

За критерій безпеки роботи апаратної платформи в цих умовах приймається таке максимальне значення дози опромінення елементної бази ($Д_{грі}, P$), при якому в елементній базі можуть виникнути зміни, але РЕА ще буде працювати з необхідною якістю.

В якості критерію по стійкості роботи можна використати граничне значення дози радіації :

$$Д_{гр} = k \cdot Д_{зв} \cdot k_{посл}, \qquad (6.1)$$

де k – коефіцієнт надійності (приймається k = 0,92);

$Д_{зв}$ – доза, яка відповідає початку зворотних змін в елементах;

$k_{посл}$ - коефіцієнт послаблення радіації (приймається $k_{посл} = 5$).

Доза, яка відповідає початку зворотних змін в елементах, що дорівнює мінімальному значенню $Д_{грі}$. Оскільки дана апаратна платформа міститиме в своїй будові мікросхеми, тому допустима експозиційна доза цих елементів РЕС становитиме $Д_{грі}$ =10^3 (Р). Отже подальші розрахунки проводяться відповідно до цього значення.

Шляхом підстановки числових значень в (7.1) розраховується $Д_{гр}$:

$$Д_{гр} = 0,92 \cdot 10^3 \cdot 5 = 4600(P).$$

Отже, працездатність апаратної платформи в умовах дії іонізуючих випромінювань можлива при $Д_{гр}$<4600Р.

6.1.2 Дослідження безпеки роботи апаратної платформи для реалізації SHA-алгоритмів в умовах дії електромагнітного імпульсу

Початкові дані: напруга живлення: $U_{ж} = 12B$; максимальна довжина горизонтальної струмопровідної частини електричної принципової схеми: $l_B = 5$м.

В якості показника стійкості елементів системи до дії електромагнітного імпульсу використовують коефіцієнт безпеки [20]

$$K_б = 20\lg\frac{U_д}{U_{B(Г)}} \geq 40[дБ], \tag{6.2}$$

де $U_д$ - допустиме коливання напруги живлення;

$U_{B(Г)}$- напруга наведена за рахунок електромагнітного імпульсу у вертикальних (горизонтальних) струмопровідних системах.

Спочатку визначається допустиме коливання напруги живлення

$$U_д = U_{ж} + \frac{U_{ж}}{100} \cdot N, \tag{6.3}$$

де N - допустимі коливання (приймається $N = 5\%$)

Шляхом підстановки числових даних в (7.3) отримується

$$U_д = 12 + \frac{12}{100} \cdot 5 = 12,6(B).$$

Визначається максимально очікувана напруга в горизонтальних лініях [20]

$$U_B = \frac{U_д}{10^{\frac{K}{20}}} \quad . \tag{6.4}$$

Після підстановки числових даних

$$U_B = \frac{12,6}{10^{\frac{40}{20}}} = 0,126(B).$$

З формули визначається горизонтальна складова напруженості електричного поля[20]

$$U_B = E_Г \cdot l_B . \tag{6.5}$$

Отже, $E_Г$ визначається

$$E_Г = \frac{U_В}{l_В}. \quad (6.6)$$

Після підстановки числових даних в формулу (6.6)

$$E_Г = \frac{0{,}126}{5} = 0{,}0252(В/м).$$

Вертикальна складова напруженості електричного поля визначається з формули[20]

$$E_Г = 10^{-3} \cdot E_В. \quad (6.7)$$

Тоді $E_В$ буде

$$E_В = 0{,}0252 \cdot 1000 = 25{,}2(В/м).$$

Це значення вертикальної складової напруженості електромагнітного поля можна вважати граничним, до якого гарантується безпечна робота апаратної платформи .

6.2 Висновки до розділу

Отже, в даному розділі було проведено дослідження безпеки роботи апаратної платформи для реалізації SHA-алгоритмів при дії іонізуючих випромінювань та електромагнітного імпульсу. Як видно з отриманих результатів, апаратної платформи є стійким до дії іонізуючих випромінювань.

Згідно з виконаними розрахунками безпека роботи апаратної платформи для реалізації SHA-алгоритмів в умовах дії електромагнітного імпульсу можлива при напруженості вертикальної складової електричного поля $E_В$ <25,2В/м.

ВИСНОВКИ

Розроблений криптографічний модуль для захисту ПЗ ПЛІС може бути використаний для захисту від нелегального використання цільової програми на пристроях, ґрунтованих на архітектурі FPGA ПЛІС. Метод полягає в аутентифікації серійного номера, що дозволяє протидіяти різним видам атак.

Застосування програмованих логічних інтегральних схем набагато спрощує реалізацію основних вузлів (таких, як лічильник, вузол порівняння та ін.) та їх взаємодію. Звідси виникають переваги у порівняно малій ціні пристрою.

В ході роботи було перевірено можливість роботи по частинах над проектом, про що свідчить створення апаратної платформи для реалізації SHA-алгоритмів на FPGA. Після виконання даної роботи достатньо легко виконати об’єднання цих проектів в одне ціле для створення цифрової частини на програмованій логічній інтегральній схемі.

Також, що є суттєвим для цифрової техніки, зменшується в декілька разів собівартість пристрою. Ще одною із переваг такої реалізації вимірювача фази на FPGA є низька потужність споживання та застосування сучасних програмованих логічних інтегральних схем.

До недоліків слід віднести шуми квантування та похибка, яка виникає через неспівпадання квантуючого імпульсу із початком (чи закінченням пускового сигналу тригера). За таким методом потрібно виконати декілька етапів роботи, а потім просумувавши результат поділити його на кількість цих вимірювань. Таким чином буде отримано середнє значення, яке наближатиметься до ідеального з наближенням кількості вимірювань до нескінченності.

Для виробництва такі апаратні платформи для реалізації SHA-алгоритмів на FPGA цілком придатні і мають майбутню перспективу

використання в лабораторіях, ремонтних дільницях побутової радіо- та телеапаратури та ін.

Роботою були визначені заходи щодо охорони праці, а саме, були визначені технічні рішення щодо безпечної експлуатації об’єкта, технічні рішення з гігієни праці та виробничої санітарії.

Додаткові розрахунки на економічність приладу показали, що його впровадження у виробництво є економічно ефективним. Аналізуючи ринок можна розраховувати на значний попит на наш виріб. Підтвердженням цьому є технічні параметри даного пристрою, які кращі за параметри аналога.

ПЕРЕЛІК ПОСИЛАНЬ

1. Вичужанин В. Стан ринку і розширення сфери застосування ПЛІС[Електронний ресурс] URL: http://www.kit-e.ru/articles/plis/2004 5 60.php (дата звернення : 18.03.2015).

2. Хоровиц П., Хилл .У. Мистецтво схемотехніки [Текст]: У 3-х томах: Т. 2. Пер. з англ.-4-е видавництвом, перераб. і доп.-М.: Світ, 1993. - 153 с.

3. Туригин И.Г. Метод вибору програмованих логічних інтегральних схем на основі цільового функціонала при проектуванні облаштувань цифрової обробки інформації [Текст]:Дисертація канд. техн. наук - Пенза, 2014. - 134 с.

4. Елементна база електроніки[Електронний ресурс] URL: http://www.electronics.ru/files/article pdf/0/article 874 75.pdf#00.

5. Programmable Devices [Электроннийресурс] // All Programmable FPGAs Xilinx Inc.: [сайт]. [2020]. URL: http://www.xilinx.com/#00.

6. Програмовані логічні інтегральні схеми [Електронний ресурс] URL: http://mehatronics.ru/2011/01/ программируемие-логические-интеграл/ (дата звернення : 18.03.2020).

7. Сергиенко А.М. VHDL для проектування обчислювальних пристроїв [Текст]:- К.: Вид-во ТОВ «ТИД » ДС«, 203. - 203с.

8. Налагоджувальна плата NI Digital Electronics FPGA Board [Електронний ресурс] URL: ftp://ftp.ni.com/pub/branches/russia/ni elvis/prototyping board with fpga design. pdf#00.

9. Прищепа С.Л. Проектування цифрових схем з допомогою САПР WebPACKISE. [Текст] - Мінськ.:БГУИР, 2006.- 56с.

10. Методичні вказівки до виконання студентами-магістрантами наукового напрямку економічної частини магістерських кваліфікаційних робіт / Уклад. В.О. Козловський – Вінниця: ВНТУ, 2012. – 22 с.

11. Козловський В.О. Техніко-економічні обґрунтування та

економічні розрахунки в дипломних проектах та роботах. Навчальний посібник. – Вінниця : ВДТУ, 2003. – 75с.

12. Максфилд К. Проектування на ПЛІС. Курс молодого бійця[Текст]. - М.: Видавничий дім «Додэка - XXI», 2007. Стр.62 (Серія «Програмовані системи»).

13. McNeilS. Solving Today's Design Security Concerns [Електронний ресурс] URL: http://www.xilinx.com/support/documentation/white papers/wp365 Solving Security Concerns.pdf (дата звернення: 16.11.2020).

14. Шифрування SHA - 256 [Електронний ресурс] URL: https://www.boxcryptor.com/ru/шифрование #00.

15. Коркикян Р. Криптографія під прицілом II : диференціальний аналіз живлення [Електронний ресурс] URL: https://xakep.ru/2015/01/27/kriptogry-191/#00

16. Moradi A., Kasper M., PaarC. Black - Box Side - Channel Attacks Highlight the Importance of Countermeasures - An Analysis of the Xilinx Virtex - 4 and Virtex - 5 Bitstream Encryption Mechanism [Електронний ресурс] URL: http://www.rsaconference.com/writable/presentations/file upload/cryp- 107.pdf #00.

17. Стецко И.П., Кулик В.Д. Комп'ютерне проектування цифрових систем [Електронний ресурс] URL: http://www.rfe.by/media/kafedry/kaf4/publication/stetsko/program-cifr-elektonika/lab.pdf (дата звернення: 17.11.2020).

18. Комолов Д., Золотухо Р. Использование микросхем специальной памяти для обеспечения защити ПЛІС FPGAот копирования//Компоненти и технологии. [Електронний ресурс] URL: http://www.kit-e.ru/articles/plis/2008 12 24.php#00.

19. DS28E01 - 100 1Kb Protected 1 - Wire EEPROM with SHA - 1 Engine [Електронний ресурс] URL: www.maxim-ic.com/fullds/DS28E01- 100 (дата звернення: 14.11.2020).

20. Черемесинов Д. жур. Прикладная дискретная математика Випуск

№ 2 / 2014 О защите интеллектуальной собственности в процессе проектирования устройств на основе FPGA XILINX [Електронний ресурс] URL: http://cyberleninka.ru/article/n/o-zaschite-intellektualnoy-sobstvennosti-v-protsesse-proektirovaniya-ustroystv-na-osnove-fpga-xilinx#ixzz3b2nhxgPp#00.

21. Advanced Encryption Standard (SHA) (FIPS PUB 197). Announcing the ADVANCED ENCRYPTION STANDARD(SHA) [Електронний ресурс] URL: http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf#03.

22. Announcing the Advanced Encryption Standard (FIPS PUB 197)

23. Бибило П.Н. Основи язика VHDL [Текст]: - М.: Изд-во СОЛОН-Р, 2002. - 217 с.

24. ГОСТ 19.701-90(ИСО 5807-85) Единая система программной документации.

25. Сакевич В.Ф. Основи розробки питань цивільної оборони в дипломних проектах. Навч. посібник: - Вінниця: ВДТУ, 2001- 109с.

26. Сакевич В.Ф., Буров В. М. Организационно-метокические указания по разработке вопросов гражданской обороны в дипломных проектах. Для студентов всех специальностей.- Винница: ВПИ 1987.- 113 с.

27. Атоманюк В.Г. Ширшнев А.Г. Акимов Н.И. Гражданская оборона. Учебник для вузов. Под ред. Михайлика Д.И.- М: Высш. Шк., 1986-207с.

28. Методичні вказівки щодо опрацювання розділу “Безпека життєдіяльності” в дипломних проектах і роботах студентів радіотехнічних спеціальностей /Уклад. О.П. Терещенко, О.В. Кобилянський. – В .: ВНТУ, 2004.- 45 с.

29. СНиП 2.09.04-87 – «Адміністративні і побутові будинки і приміщення виробничих підприємств»

30. Долин Петро Олександрович. Основи техніки безпеки у електроустановках. – М.: Энергоатомиздат. 1984. - 448 с.

31. ГОСТ 12.1.005-88. ССБТ. Общие санитарно-гигиенические требования к воздуху рабочей зоны.

32. ГОСТ 12.04.05-86 Отопленіє вентиляция и кондиционированіє.

33. СНиП II-4-79 – «Естесвенное и икуственное освещение».

34. Справочная книга для проектирования электрического освещения/ Под ред. Г.М.Кнорринга. - Л.: Энергия, 1976. - 346 с.

35. Боротьба с шумом на производстве: Справочник / Под ред. Е.Я.Юдина. - М.: Машиностроение, 1985. - 400 с.

36. ДНАОП 0.03-33.14-85. Санітарні норми допустимих рівнів шуму на робочих місцях.

37. ГОСТ 12.1.003-83. Шум. Общие требования безопасности.

38. ГОСТ 12.1.012-90. Вибрационная безопасность.

39. СниП 2.01.02-85 – «Противопожарные нормы».

40. ОНТП 24-86. Определение категорий помещений и зданий на взрывоопасность и пожарной безопасности. МВД СССР.- М.,1986.

Додаток А
(обов’язковий)

ВНТУ

ПОГОДЖЕНО

“__” _____ 2020 р.

ЗАТВЕРДЖУЮ
Зав. кафедри РТ ВНТУ,
докт. техн. наук, професор
О.В. Осадчук
“__” _____ 2020 р.

ТЕХНІЧНЕ ЗАВДАННЯ
на виконання магістерської кваліфікаційної роботи

Розробка апаратної платформи для реалізації SHA-алгоритмів на FPGA

08-36.МКР.012.00.000 ТЗ

Керівник роботи: к.т.н. доцент
_____ Воловик А.Ю..
(підпис)
“____”_____2020 р.
Виконавець: студент гр. РТ-19м
_____ Прокопчук С.С.
(підпис)
“____”_____2020 р.

Вінниця 2020

1 ПІДСТАВА ДЛЯ ВИКОНАННЯ РОБОТИ

Робота проводиться на підставі наказу ректора по Вінницькому національному технічному університету №_214_ від 25.09.2020 р. та індивідуального завдання на магістерську кваліфікаційну роботу.

Дата початку роботи: 03.09.2020 р.

Дата закінчення: 10.12.2020 р.

2 МЕТА І ПРИЗНАЧЕННЯ МКР

Метою роботи є створення та дослідження апаратної платформи для реалізації SHA-алгоритмів на FPGA.

Об'єктом дослідження є розробка теоретичних засад, методів та засобів для шифрування програмного забезпечення для ПЛІС.

Предметом дослідження – є платформи для реалізації SHA-алгоритмів на FPGA.

В магістерській кваліфікаційній роботі для досягнення поставленої мети **розв'язуються такі завдання**:

1. Проведення аналізу сучасного стану розробки апаратної платформи для реалізації SHA-алгоритмів на FPGA та виявлено базові переваги та недоліки приладів, що вже існують.

2. Попередній розрахунок структурної схеми апаратної платформи для реалізації SHA-алгоритмів на FPGA з функцією передачі даних, що включає в себе розробку таких блоків та вузлів: блок циклічного запису даних з модуля дешифрування , модуль пошуку серійного номера, генератор випадкових чисел.

3. Розрахунок каскадів апаратної платформи для реалізації SHA-алгоритмів на FPGA, а саме: модель порушника, вбудовані методи захисту шифруванням бітового потоку, зберігання ключів, ОЗП з живленням від батареї, eFUSE- регістри, атака на вбудовані методи захисту, метод захисту за допомогою вбудовування дешифратора, атака на метод захисту, що використовує вбудовуваний дешифратор, використання мікросхем спеціальної пам'яті, атака на метод захисту, що використовує мікросхему спеціальної пам'яті, проблема декомпіляції конфігураційного файлу, концепція розробленого модуля криптозахисту, опис вибраного криптоалгоритма шифрування.

4. Проведення експериментальних досліджень блоку циклічного запису даних з модуля дешифрування, модуля пошуку серійного номера, генератора випадкових чисел за допомогою системи автоматизованого проектування показало, що результати моделювання відповідають умовам технічного завдання та принципу роботи пристрою в цілому.

5. Розробка блоків апаратної платформи для реалізації SHA-алгоритмів на FPGA з застосуванням нової елементної бази, дозволить підвищити точність та діапазон роботи пристрою, при покращенні масо-габаритних властивостей.

3 ДЖЕРЕЛА РОЗРОБКИ

1. Вичужанин В. Стан ринку і розширення сфери застосування ПЛІС[Електронний ресурс] URL: http://www.kit-e.ru/articles/plis/2004 5 60.php (дата звернення : 18.03.2015).

2. Хоровиц П., Хилл .У. Мистецтво схемотехніки [Текст]: У 3-х томах: Т. 2. Пер. з англ.-4-е видавництвом, перераб. і доп.-М.: Світ, 1993. - 153 с.

3. Туригин И.Г. Метод вибору програмованих логічних інтегральних схем на основі цільового функціонала при проектуванні облаштувань цифрової обробки інформації [Текст]:Дисертація канд. техн. наук - Пенза, 2014. - 134 с.

4. Елементна база електроніки[Електронний ресурс] URL: http://www.electronics.ru/files/article pdf/0/article 874 75.pdf#00.

5. Programmable Devices [Электроннийресурс] // All Programmable FPGAs Xilinx Inc.: [сайт]. [2020]. URL: http://www.xilinx.com/#00.

6. Програмовані логічні інтегральні схеми [Електронний ресурс] URL: http://mehatronics.ru/2011/01/ программируемие-логические-интеграл/ (дата звернення : 18.03.2020).

7. Сергиенко А.М. VHDL для проектування обчислювальних пристроїв [Текст]:- К.: Вид-во ТОВ «ТИД » ДС«, 203. - 203с.

8. Налагоджувальна плата NI Digital Electronics FPGA Board [Електронний ресурс] URL: ftp://ftp.ni.com/pub/branches/russia/ni elvis/prototyping board with fpga design. pdf#00.

9. Прищепа С.Л. Проектування цифрових схем з допомогою САПР WebPACKISE. [Текст] - Мінськ.:БГУИР, 2006.- 56с.

10. Методичні вказівки до виконання студентами-магістрантами наукового напрямку економічної частини магістерських кваліфікаційних робіт / Уклад. В.О. Козловський – Вінниця: ВНТУ, 2012. – 22 с.

11. Козловський В.О. Техніко-економічні обґрунтування та економічні розрахунки в дипломних проектах та роботах. Навчальний посібник. – Вінниця : ВДТУ, 2003. – 75с.

4 ВИКОНАВЕЦЬ

Вінницький національний технічний університет, кафедра радіотехніки, студент групи РТ-19м Прокопчук С.С.

5 ВИМОГИ ДО ВИКОНАННЯ МКР

- Кількість системних вентилів: до 50 тисяч;
- Кількість логічних комірок: до 1047;
- Сімейство логіки: КМОН;
- Об'єм вбудованого EEPROM (ЕСППЗП): до 4 Мбіт;
- Вбудований інтерфейс для відладки/завантаження ПЛІС на основі шини USB;
- Лінії введення/виведення загального призначення: 32 цифрових лінії введення/виведення,
- максимальна напруга 3.3 В,
- максимальний струм 8 мА;
- Джерело живлення постійного струму : 15 В
- постійного струму, 650 мА;
- Загальна споживана потужність: 6 Вт макс;

6 ЕТАПИ МКР І ТЕРМІНИ ЇХ ВИКОНАННЯ

№ з/п	Назва етапів магістерської кваліфікаційної роботи	Термін виконання		Очікувані результати	Звітна документація
1.	Огляд літературних джерел. Вибір та узгодження теми МКР	03.09.2020	14.09.2020	Проведено огляд літературних джерел. Вибрана тема	Узгодження теми МКР на кафедрі
2.	Аналіз літературних джерел. Попередня розробка основних розділів	15.09.2020	21.09.2020	Проведений аналіз літературних джерел по даній тематиці. Підготовлений матеріал основних розділів	Вступ
3.	Затвердження теми. Розробка технічного завдання	21.09.2020	25.09.2020	Розроблене ТЗ	Наказ ВНТУ про затвердження теми. Додаток А
4.	Аналіз вирішення поставленої задачі. Розробка структурної схеми	26.09. 2020	09.10. 2020	Проведений аналіз. Розроблені схеми пристрою	Звіт по переддипломній практиці Вступ Розділ 1-3
5.	Електричні розрахунки. Експериментальне дослідження	10.10.2020	25.10. 2020	Проведені розрахунки та дослідження	Розділ 4-6
6.	Розділ моделювання	26.10. 2020	04.11. 2020	Проведено моделювання	Результати моделювання
7.	Розробка графічної частини МКР	05.11. 2020	15.11. 2020	Плакати. Структурні та електричні схеми	Графічна частина
8.	Аналіз економічної ефективності розробки	16.11. 2020	19.11. 2020	Економічна частина	Розділ 7
9.	Охорона праці (ОП)	19.11. 2020	22.11. 2020	Частина БЖДПБ	Розділ 8
10.	Оформлення пояснювальної записки та графічної частини	23.11. 2020	29.11. 2020	Оформлена документація	ПЗ та графічна частина
11.	Нормоконтроль	30.11. 2020	01.12. 2020	Підпис нормоконтроля	Оформлена ПЗ та графічна частина
12.	Попередній захист МКР, доопрацювання, рецензування МКР	02.12. 2020	04.12. 2020	Позитивні відзиви	Відзив. Рецензія
13.	Захист МКР ЕК	11.12. 2020	14.12. 2020	Позитивний захист	Протокол ЕК

7 ОЧІКУВАНІ РЕЗУЛЬТАТИ ТА ПОРЯДОК РЕАЛІЗАЦІЇ МКР

У результаті виконання роботи будуть розроблені:

- математичне моделювання основних характеристик апаратної платформи для реалізації SHA-алгоритмів на FPGA;
- нові межі використання апаратної платформи для реалізації SHA-алгоритмів на FPGA;
- розділ безпеки життєдіяльності і ЦЗ;
- економічна частина МКР.

Результати, отримані в процесі виконання даної роботи, можуть бути впроваджені в різних галузях науки і техніки.

8 МАТЕРІАЛИ, ЯКІ ПОДАЮТЬ ПІСЛЯ ЗАКІНЧЕННЯ РОБОТИ ТА ПІД ЧАС ЕТАПІВ

За результатами виконання МКР до ЕК подаються пояснювальна записка, графічна частина МКР, відгук керівника і рецензія.

9 ПОРЯДОК ПРИЙМАННЯ МКР ТА ЇЇ ЕТАПІВ

Поетапно результати виконання МКР розглядаються керівником роботи та обговорюються на засіданні кафедри.

Захист магістерської кваліфікаційної роботи відбувається на відкритому засіданні ЕК.

10 ВИМОГИ ДО РОЗРОБЛЮВАНОЇ ДОКУМЕНТАЦІЇ

Документація, що розробляється в процесі виконання досліджень повинна містити:

- дослідження поставленого питання;

- проектування розроблюваних апаратної платформи для реалізації SHA-алгоритмів на FPGA;
- методи дослідження апаратної платформи для реалізації SHA-алгоритмів на FPGA;
- економічну частину та розділ БЖДПБ.

11 ВИМОГИ ЩОДО ТЕХНІЧНОГО ЗАХИСТУ ІНФОРМАЦІЇ З ОБМЕЖЕНИМ ДОСТУПОМ

У зв’язку з тим, що інформація не є конфіденційною, заходи з її технічного захисту не передбачаються.

Додаток Б
(обов’язковий)

Розробка апаратної платформи для реалізації SHA-алгоритмів на FPGA

Налагоджувальна плата NI Digital Electronics FPGA Board

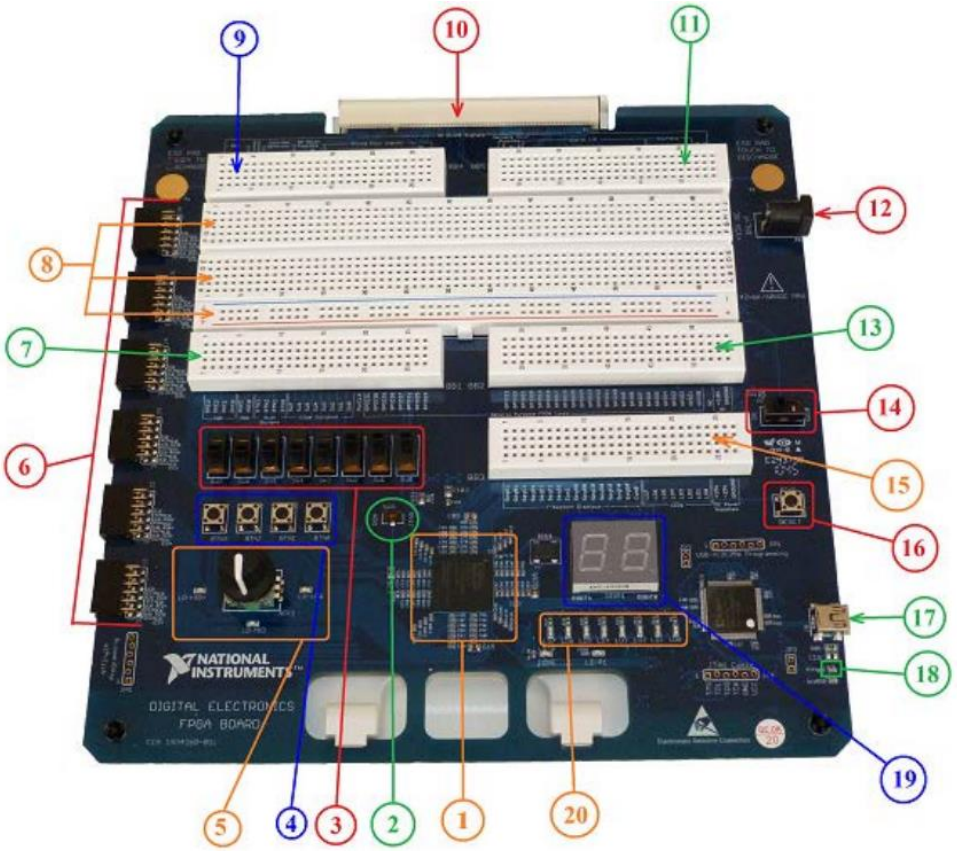


Рисунок Б1 – Налагоджувальна плата NI Digital Electronics FPGA Board

1. ПЛІС XC3S500E Xilinx Spartan - 3E
2. Перемикач SW9
3. Рухомі перемикачі (SW0 - SW7)
4. Кнопки (BTN0 - BTN3)
5. Натискний перемикач
6. Роз'єми Digilent Pmod
7. Зона макетування: сигнальний роз'єм BB1
8. Роз'єми загального призначення
9. Зона макетування: сигнальний роз'єм BB4
10. Роз'єм для підключення до N1 ELVIS II+.
11. Зона макетування: сигнальний роз'єм BB5
12. Роз'єм підключення джерела живлення
13. Зона макетування: сигнальний роз'єм BB2
14. Вимикач живлення
15. Зона макетування: сигнальний роз'єм BB3
16. Кнопка скидання (Reset)
17. Роз'єм USB
18. Світлодіод LD - G
19. Семисегментні індикатори
20. Світлодіоди (LD0 - LD7)

Додаток В
(обов’язковий)

Розробка апаратної платформи для реалізації SHA-алгоритмів на FPGA

Етапи проектування пристрою в середовищі розробки WebPack ISE

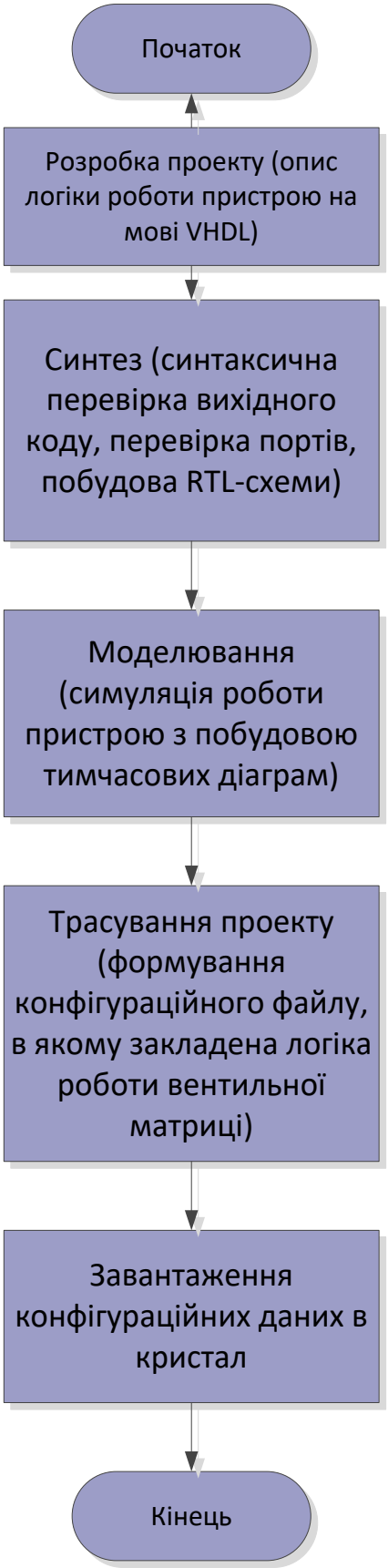
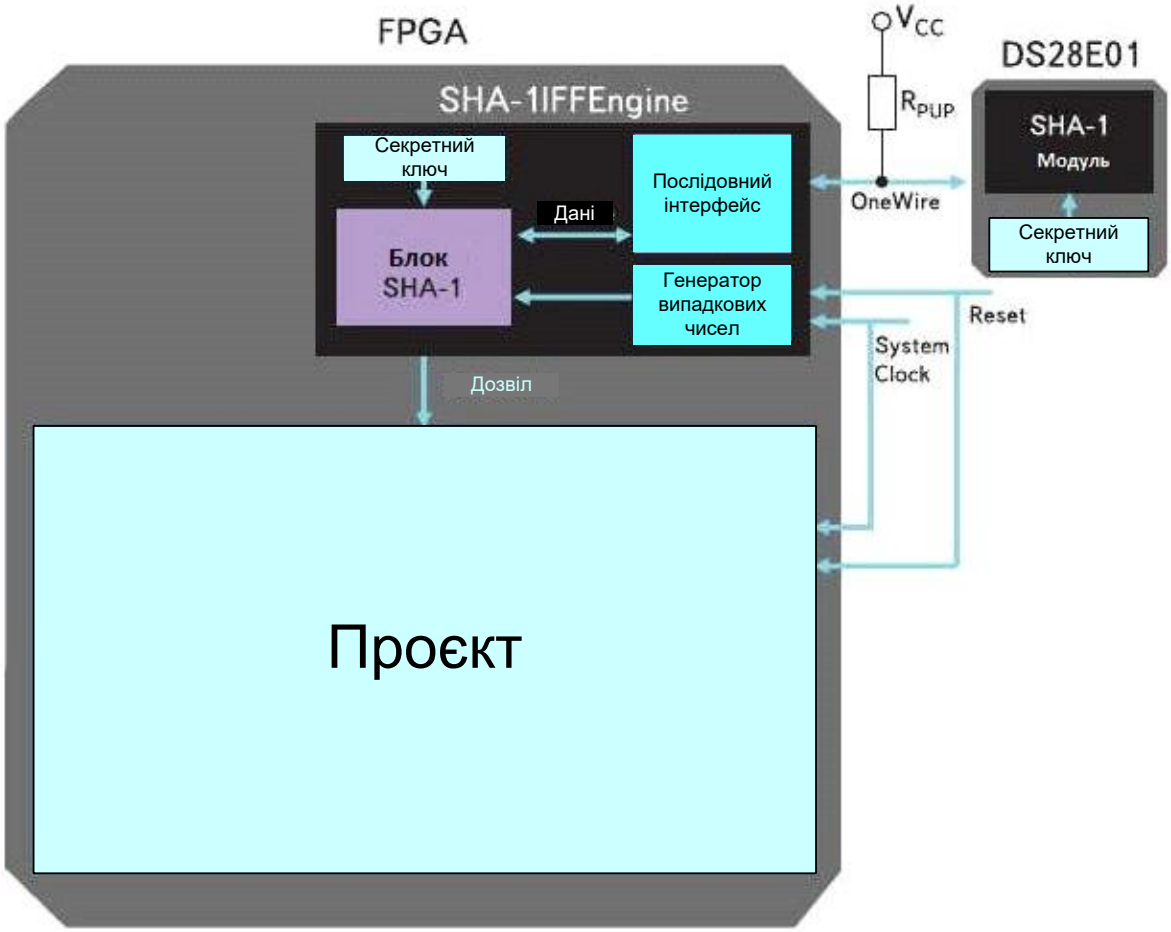


Рисунок В – Етапи проектування пристрою в середовищі розробки WebPack ISE

Додаток Г
(обов’язковий)

Розробка апаратної платформи для реалізації
SHA-алгоритмів на FPGA

Реалізація концепції IFF



Рисинок Г1 – Реалізація концепції IFF

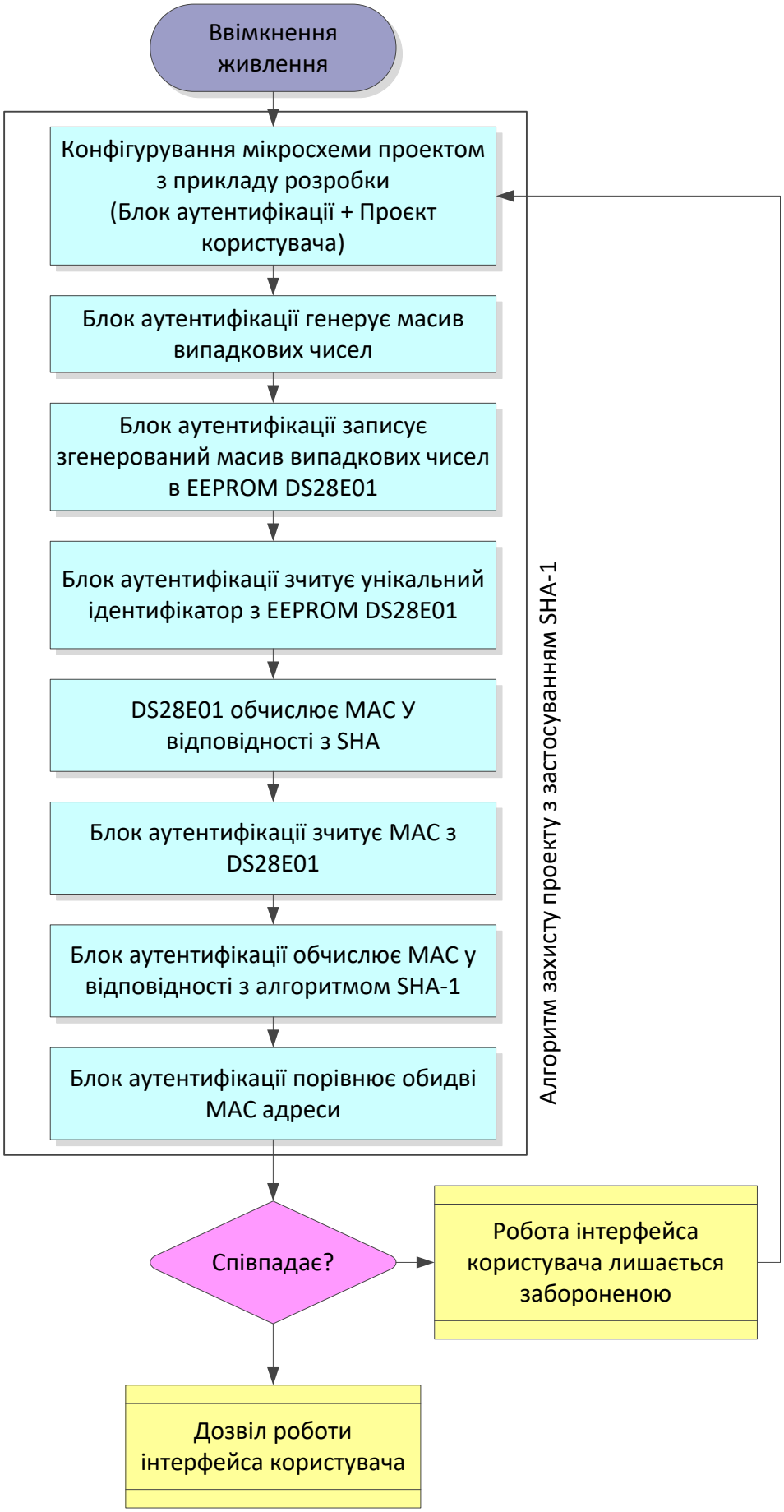


Рисунок Г2 – Алгоритм блоку перевірки при реалізації концепції IFF

Додаток Д
(обов’язковий)

Розробка апаратної платформи для реалізації
SHA-алгоритмів на FPGA

Структура алгоритму шифрування

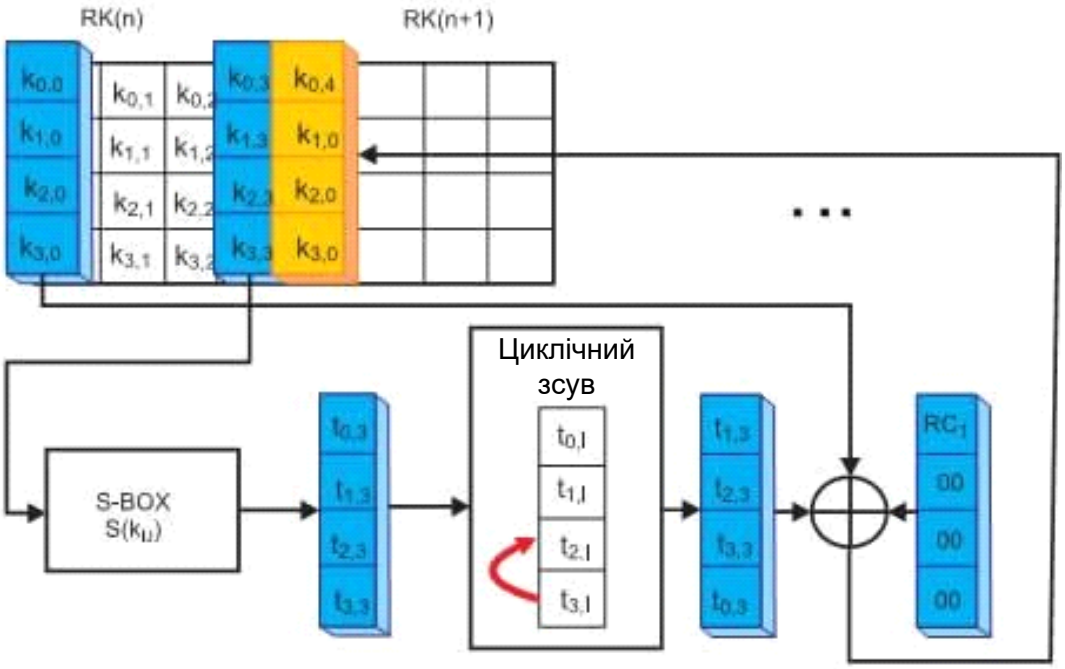


Рисунок Д.1 – Отримання першого стовпця наступного ключа раунду

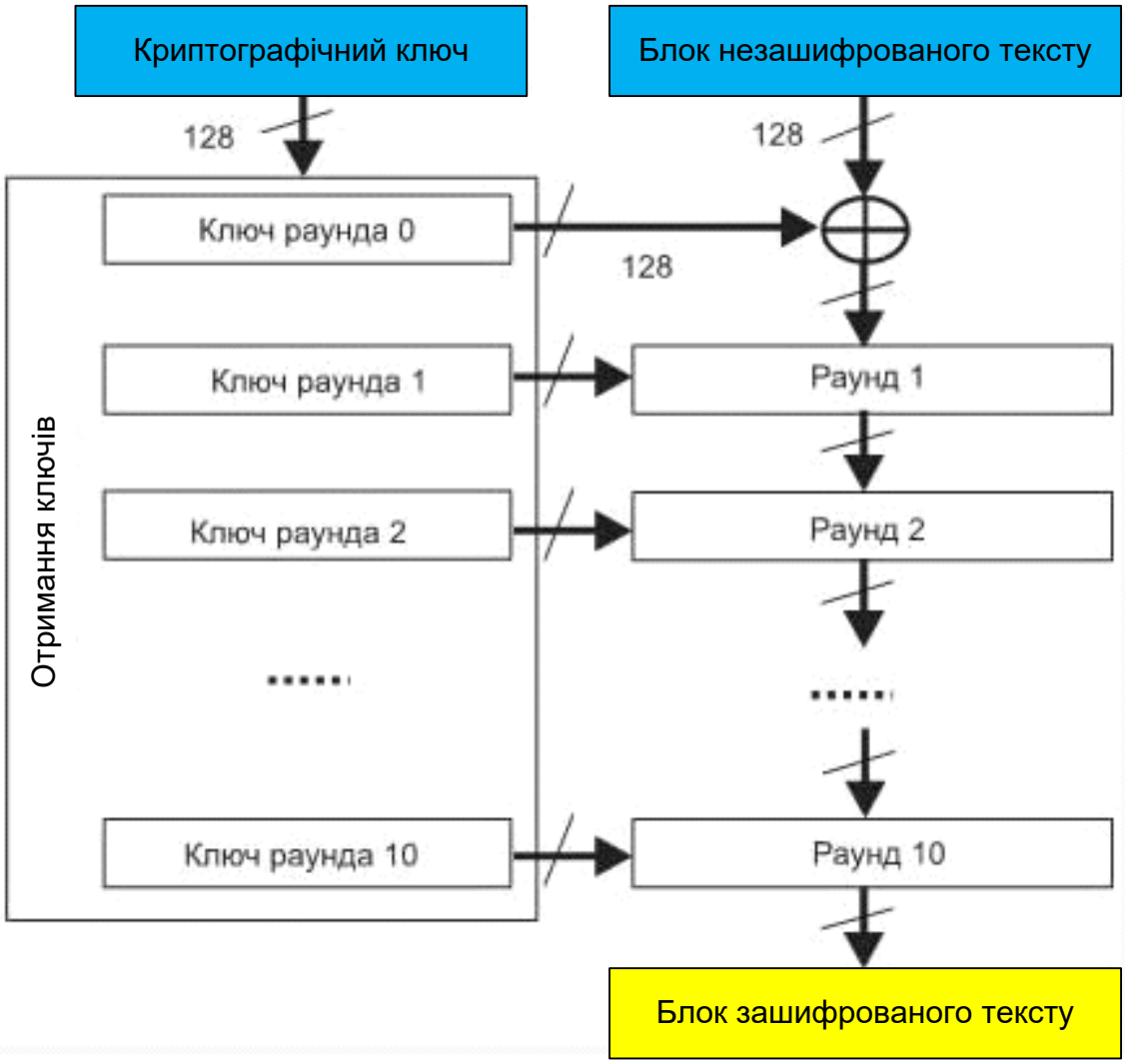


Рисунок Д.2 – Структура алгоритму SHA - 128

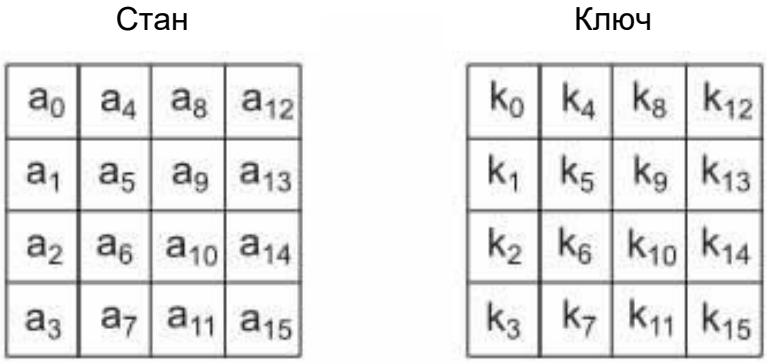


Рисунок Д.3 – Структура ключа і стану

		y														
		0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
x	0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
	1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
	2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
	3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
	4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
	5	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
	6	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
	7	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
	8	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
	9	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
	a	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
	b	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
	c	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
	d	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
	e	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
	f	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

Рисунок Д.4 – Таблиця S–Box

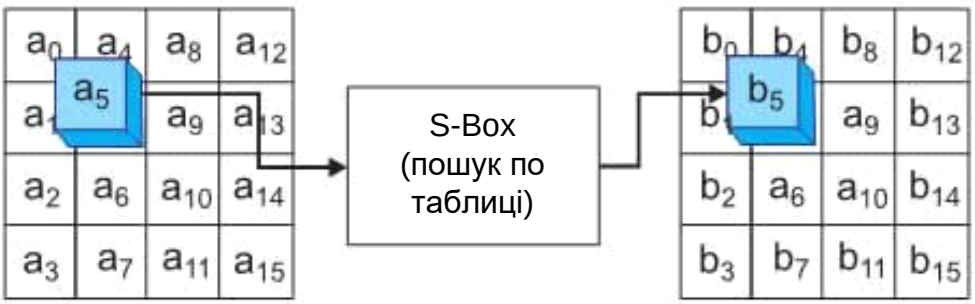


Рисунок Д.5 – Операція заміщення байтів

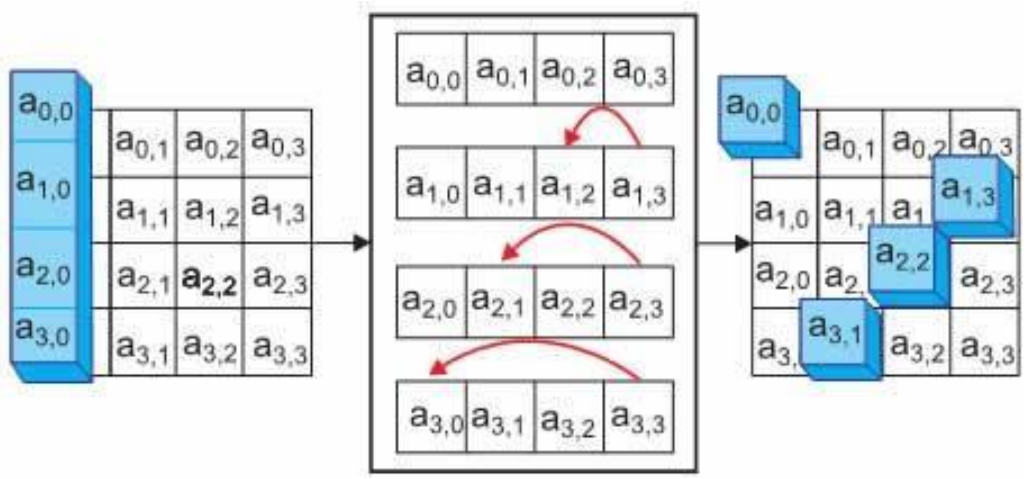


Рисунок Д.6 – Операція зсуву рядків

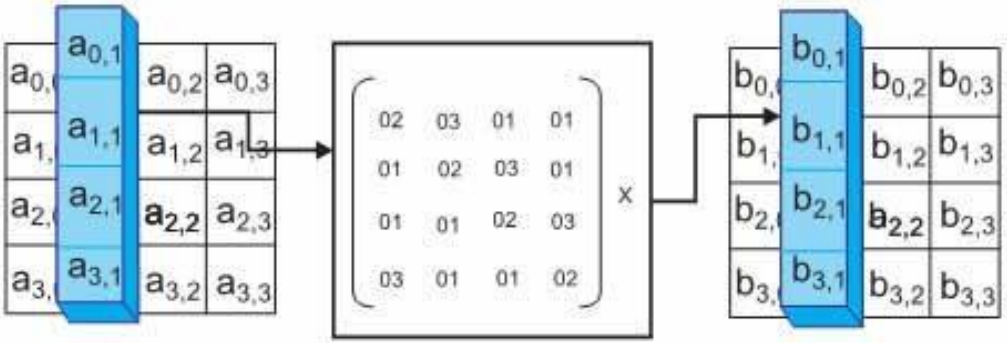


Рисунок Д.7 – Операція перемішування стовпців

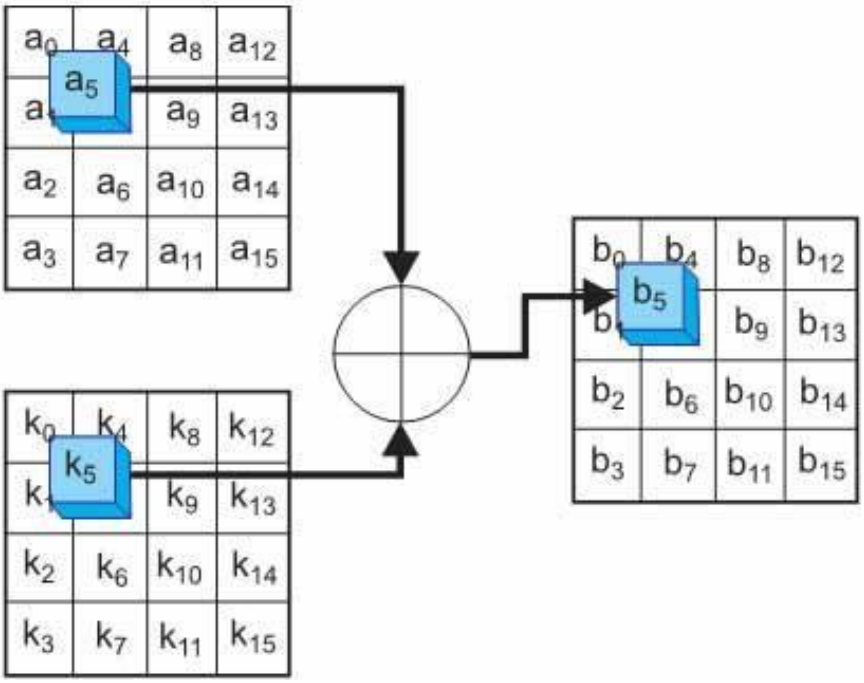


Рисунок Д.8 – Операція додавання ключа раунду

Додаток Е
(обов’язковий)

Розробка апаратної платформи для реалізації
SHA-алгоритмів на FPGA

Опис алгоритму модуля аутентифікації

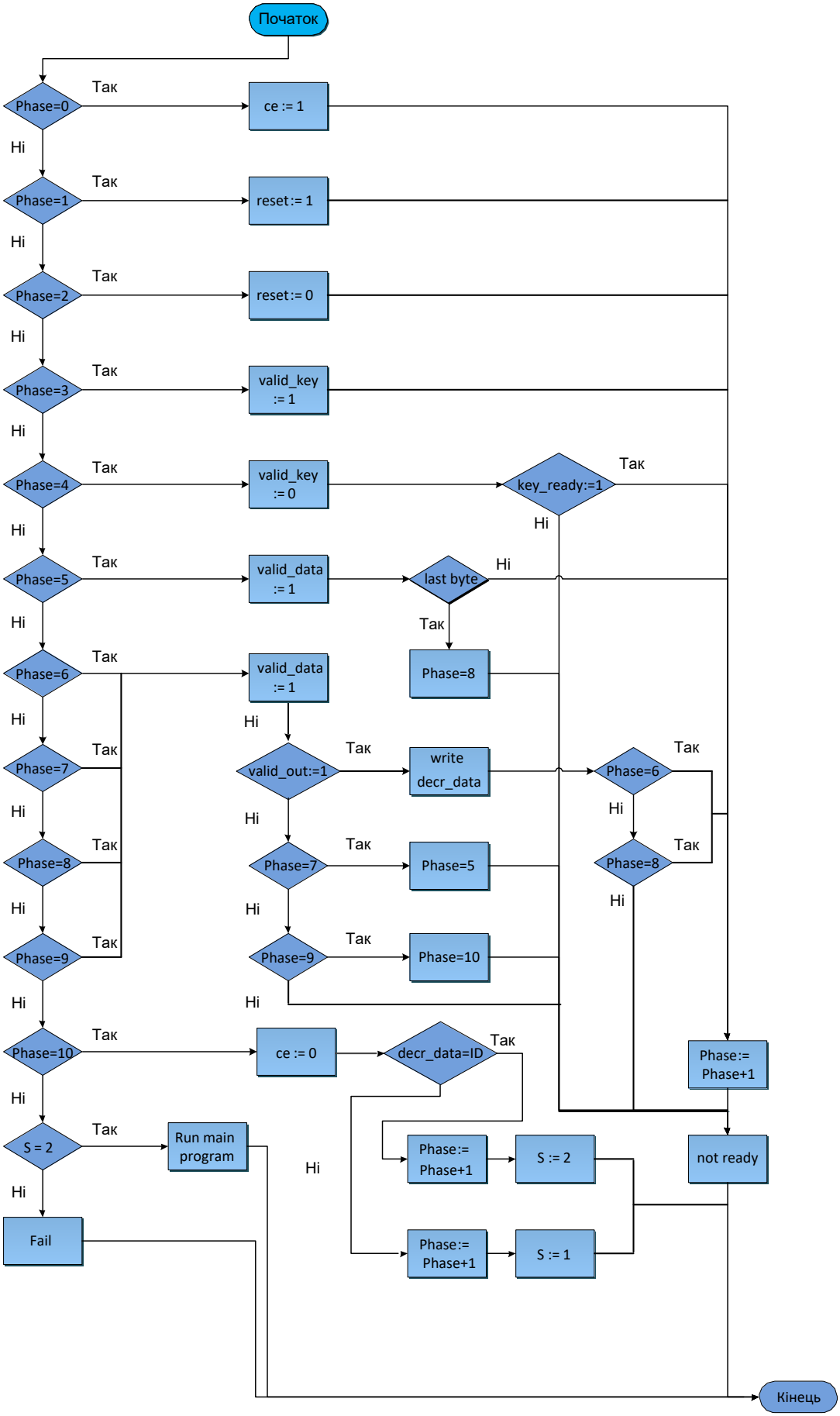


Рисунок Е.1 – Алгоритм аутентифікації серійного номера

Додаток Ж
(обов’язковий)

Розробка апаратної платформи для реалізації SHA-алгоритмів на FPGA

Опис перевірки працездатності і результати симуляції

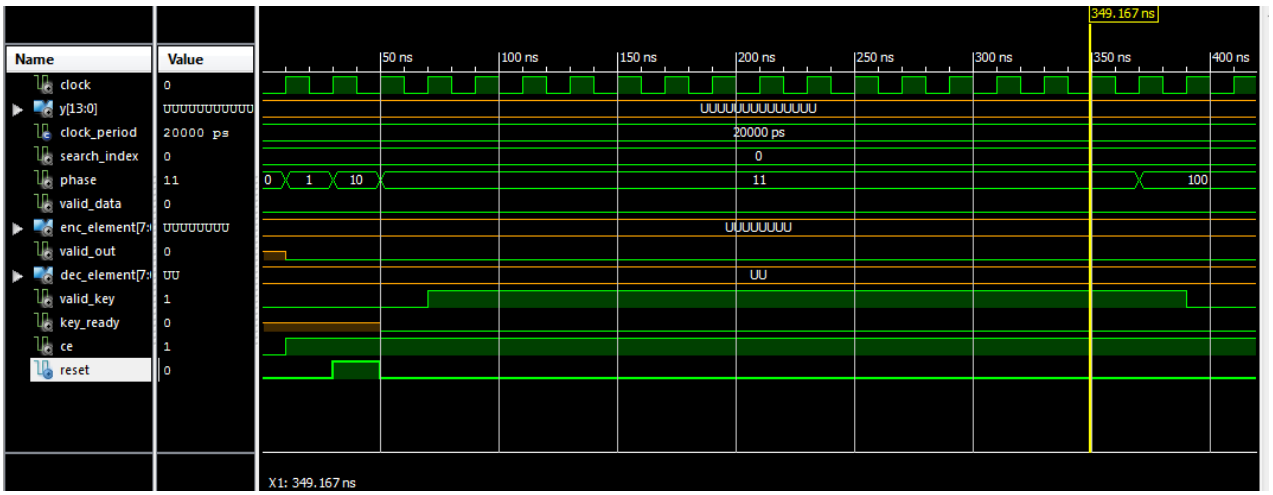


Рисунок Ж.1 – Перші три фази роботи модуля

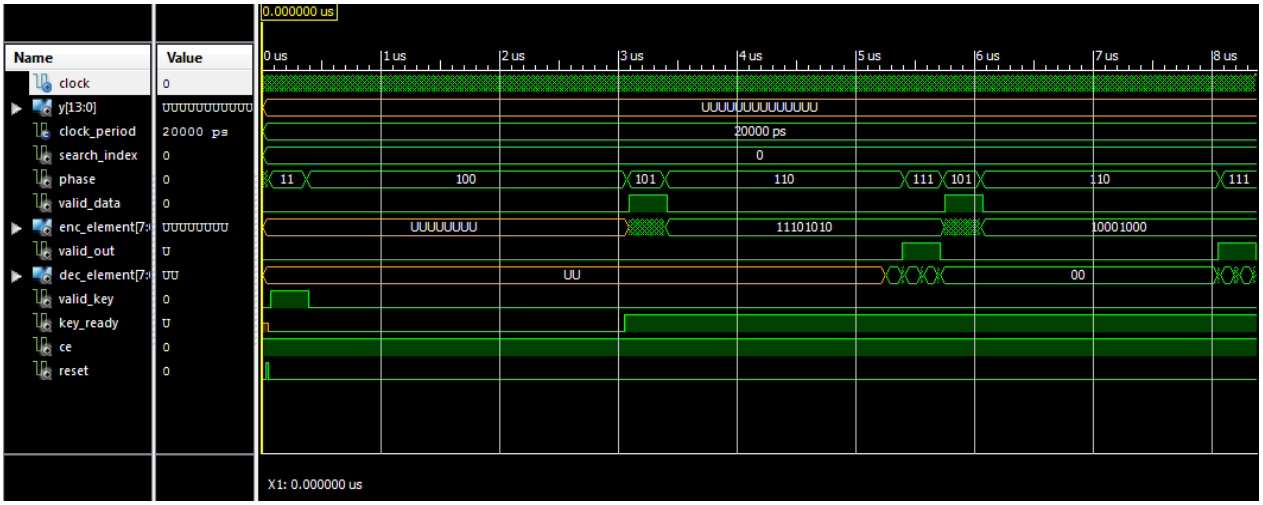


Рисунок Ж.2 – Етапи циклічного запису даних з модуля дешифрування

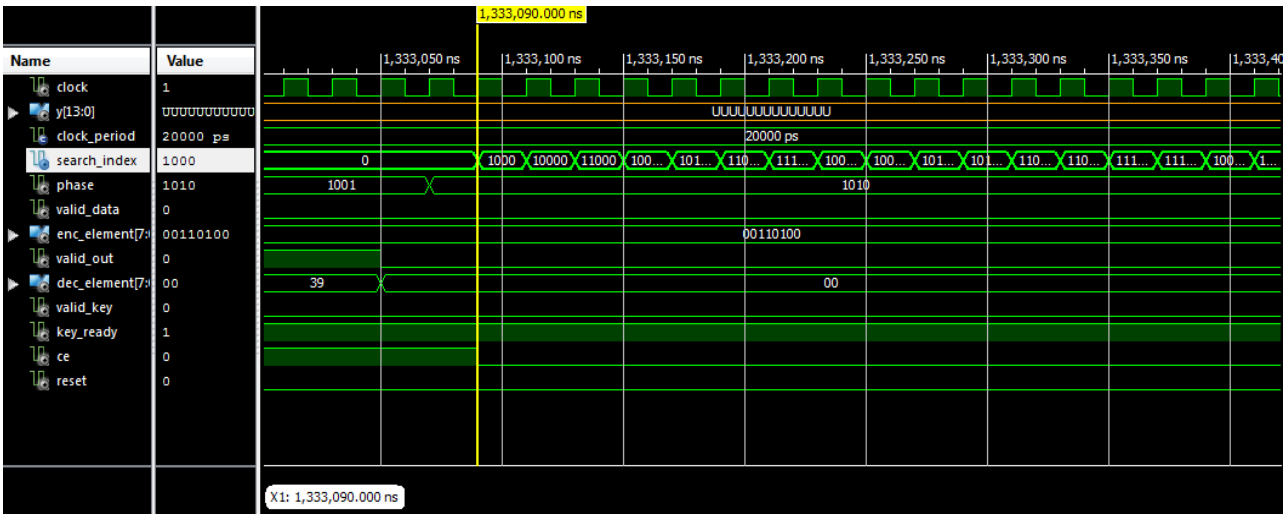


Рисунок Ж.3 – Пошук серійного номера

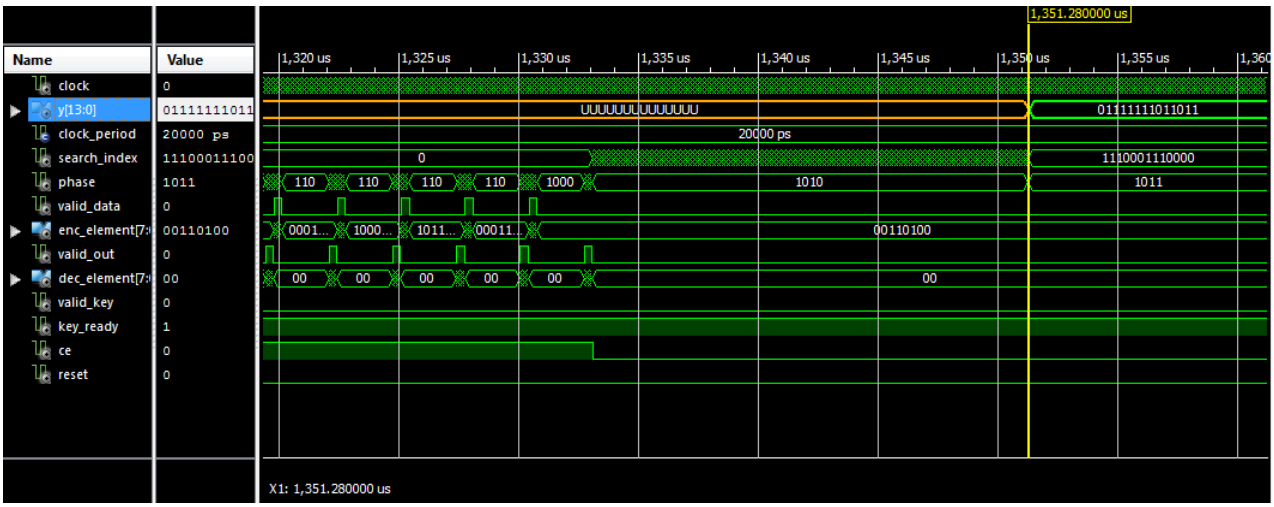


Рисунок Ж.4 – Результат роботи модуля

Рисунок Ж.5 – Генерація випадкових чисел

Додаток К

(довідниковий)

Розробка апаратної платформи для реалізації SHA-алгоритмів на FPGA

Лістинги програми

```
-- ModuleName: segm7
----------------------------------------------------------------------------------
library IEEE;
use IEEE.STD LOGIC 1164.ALL;
useieee.std_logic_unsigned.all; useieee.math_real.all;
-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
--use IEEE.NUMERIC_STD.ALL;
-- Uncomment the following library declaration if instantiating
-- any Xilinx primitives in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

entity segm7 is
Port ( clock : in      STD_LOGIC;
y : out     STD_LOGIC_VECTOR(13 downto 0) );
end segm7;
architecture Behavioral of segm7 is
COMPONENT aes_dec GENERIC(
KEY_SIZE                    :     in        integer range 0 to 2 :=0
-- 0-128; 1-192; 2-256
); PORT(
DATA_I : IN std_logic_vector(7 downto 0);
VALID_DATA_I : IN std_logic;
KEY_I : IN std_logic_vector(7 downto 0); VALID_KEY_I : IN std_logic;
RESET_I : IN std_logic; CLK_I : IN std_logic; CE_I : IN std_logic;
KEY_READY_O : OUT std_logic;
VALID_O : OUT std_logic;
DATA_O : OUT std_logic_vector(7 downto 0)
);
END COMPONENT;
```

```
type array256 is array(0 to 15) of std_logic_vector(7 downto 0);
type array32 is array(0 to 15) of std_logic_vector(7 downto 0);
type array8 is array(0 to 7) of std_logic_vector(7 downto 0);
signal delay : std_logic_vector(27 downto 0) := " 0000000000000000000000000000";
signalx_in : std_logic_vector(7 downto 0) := " 00000000";
constant serial_number : array8 := (x"31", x"30", x"30", x"30", x"30", x"30", x"30", x"39");
constantenc : array256 := (
x"f0", x"87", x"1d", x"92", x"a4", x"b1", x"89", x"60", x"7b"
x"b7", x"52", x"fb", x"cb", x"24", x"2a", x"ea"
);
signalenc_element : std_logic_vector(7 downto 0);
signalvalid_data : std_logic := '0'; constant key : array32 := (
x"4a", x"19"
x"d0", x"25", x"60", x"38", x"7a", x"89", x"2a", x"d5", x"5a", x"34", x"44", x"fb", x"44", x"f0"
);
signalkey_element : std_logic_vector(7 downto 0); -- для передачі 1 байта ключа aes_dec 'y
signalvalid_key : std_logic := '0'; -- флаг передачі ключа, на '1 '- передача всех байтов
signalce : std_logic := '0'; -- флаг дозворлу работи модуля aes_dec
signalreset : std_logic := '0'; -- флаг скидання стану модуля aes_dec
signalkey_ready : std_logic; --флаг готовності ключа на модулі aes_dec
signalvalid_out : std_logic; --флаг готовності наступного розшифрованого блока даних, 16 байт=блок
signaldec_element : std_logic_vector(7 downto 0);--для передачі одного байта розшифрованих даних из aes_dec
signaldecrypted_data : array256;--масив розшифрованих даних
```

signal index : INTEGER RANGE 0 TO 31 := 0;--індекс для ключа signalenc_index : INTEGER RANGE 0 TO 8000 := 0;-- індекс для зашифрованих данних

signaldecr_index : INTEGER RANGE 0 TO 8000 := 0;-- індекс для розшифрованих даних

signalsearch_index : INTEGERRANGE 0 TO 8000 := 0;--індекс для пошука по розшифрованому массиву

signalblock_index : INTEGER RANGE 0 TO 15 := 0;--індекс блока

signal phase : INTEGER RANGE 0 TO 31 := 0;-- даний етап розшифровки масива

signal success : integer range 0 to 2 := 0; -- 0 - not ready, 1 - fail, 2 - success

constantmax_key_index : INTEGER := 15;--максимальний індекс ключа

constantmax_enc_index : INTEGER := 15;--максимальний індекс блока

constantmax_data_index : INTEGER := 7999;--максимальний індекс масивів даних

signalrandd : integer;

begin

Inst_aes_dec: aes_dec GENERIC MAP(

KEY_SIZE => 0 --для aes 128

) PORT MAP(

DATA_I =>enc_element, VALID_DATA_I =>valid_data, KEY_I =>key_element

VALID_KEY_I =>valid_key, RESET_I => reset

CLK_I => clock

CE_I =>ce

KEY_READY_O =>key_ready, VALID_O =>valid_out, DATA_O =>dec_element

);

auth_module: process(clock) begin

```
ifrising_edge(clock) then -- тільки на передньому фронті для синхронізації

case phase is
when 0 => -- установка CE
ce<= '1'; --дозвіл работи модуля aes_dec
phase<= phase + 1;
when 1 => -- установка бита reset reset<= '1';
phase<= phase + 1; when 2 => -- скидання біта reset
reset<= '0';
phase<= phase + 1; when 3 => -- завантаження ключа
valid_key<= '1';
key_element<= key(index); --передача одного байта, всього 16 тактів
if(index <max_key_index) then index<= index + 1;
else
index<= 0;  phase<= phase + 1;
endif;
when 4 => -- очікування готовності ключа valid_key<= '0';
if(key_ready = '1') then phase<= phase + 1;
endif;
when 5 => -- завантаження зашифрованого блока даних в модуль aes_dec
valid_data<= '1';
enc_element<= enc(enc_index);
if(enc_index mod 16 = max_enc_index) then phase<= phase + 1;
end if;
if(enc_index = max_data_index) then
phase<= 8;
end if;
enc_index<= enc_index + 1;
--фаза 6 очікування розшифрованого блока даних
--фаза 7 завантаження розшифрованого блока даних
--фаза 8 очікування останнього розшифрованого блока даних
```

```
--фаза 9 завантаження останнього розшифрованого блока даних
when 6 to 9 =>
valid_data<= '0';
if(valid_out = '1') then decrypted_data(decr_index) <= dec_element; decr_index<= decr_index + 1;
if(phase mod 2 = 0) then phase<= phase + 1;
end if;
elsif(phase = 7) then
phase<= 5; elsif(phase = 9) then
phase<= 10; endif;
when 10 =>                    --пошук серійного номера в розшифрованому масиві
ce<= '0';
if(decrypted_data(search_index)  = serial_number(0)) then
then
if(decrypted_data(search_index + 1)  = serial_number(1))
then
if(decrypted_data(search_index + 2)  = serial_number(2))
then
if(decrypted_data(search_index + 3)  = serial_number(3))
then
if(decrypted_data(search_index + 4)  = serial_number(4))
then
if(decrypted_data(search_index + 5)  = serial_number(5))
then
if(decrypted_data(search_index + 6)  = serial_number(6))
then
if(decrypted_data(search_index + 7)  = serial_number(7))
success<= 2; phase<= phase + 1;
end if; end if; end if; end if;
end if; end if; end if; end if;
```

```
if(success /= 2) then
if(search_index + 8 <max_data_index) then search_index<= search_index + 8;
else
success<= 1; phase<= phase + 1;
end if; endif;
whenothers => --фаза 11 : на цьому модуль завершує свою работу
end case;
end if;
end process;
led:
process(clock) begin
if(success = 1) then                    -- FAIL
Y <= "01111110000110";
elsif(success = 2) then -- SUCCESS Y <= "01111111011011";
end if;
end process; end Behavioral;
```

Підключення семисегментного індикатора

```
---------------------------------------------------------------------------------
Net "y[7]" LOC="H6" | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 8;
Net "y[8]" LOC="K2" | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 8;
Net "y[9]" LOC="H3" | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 8;
Net "y[10]" LOC="K1" | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 8;
Net "y[11]" LOC="G4" | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 8;
```

```
Net "y[12]" LOC="J2" | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 8;
Net "y[13]" LOC="G3" | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 8;
NET "y[6]" IOSTANDARD = LVCMOS33;
NET "y[5]" IOSTANDARD = LVCMOS33;
NET "y[4]" IOSTANDARD = LVCMOS33;
NET "y[3]" IOSTANDARD = LVCMOS33;
NET "y[2]" IOSTANDARD = LVCMOS33;
NET "y[1]" IOSTANDARD = LVCMOS33;
NET "y[0]" IOSTANDARD = LVCMOS33;
NET "y[6]" LOC = C2;
NET "y[5]" LOC = C1;
NET "y[4]" LOC = E4;
NET "y[3]" LOC = D1;
NET "y[2]" LOC = G5;
NET "y[1]" LOC = E1;
NET "y[0]" LOC = E3;
NET "y[3]" DRIVE = 8;
NET "y[2]" DRIVE = 8;
NET "y[1]" DRIVE = 8;
NET "y[6]" DRIVE = 8;
NET "y[4]" DRIVE = 8;
NET "y[0]" DRIVE = 8;
NET "y[5]" DRIVE = 8;
```

Генерація випадкових чисел

---

rnd:

process(clock) variable rand: real; variable rand2: real;

variableint_rand: integer;

variable int_rand2: integer; variable seed1: positive := 53; variable seed2: positive := 31; begin

if(success = 0) then UNIFORM(seed1, seed2, rand); UNIFORM(seed1, seed2, rand2);

int_rand := INTEGER(TRUNC(rand*64.0)); int_rand2 := INTEGER(TRUNC(rand2*64.0));

randd<= int_rand * int_rand2 ; end if;

end process;