УДК 621.3

Шаповал І.В.,
Лебедев Д.Ю., к.т.н

АЛГОРИТМ РОБОТИ ПРИСТРОЮ AES ШИФРАТОРА

Національний технічний університет України
«Київський політехнічний інститут

_shapoval_ivan@ukr.net
www@netto.com.ua

Розглянуто апаратну реалізацію алгоритму роботи пристрою AES шифратора на базі ПЛІС Altera Cyclone II FPGA

Ключові слова: ПЛІС, алгоритм, шифрування даних, *AES*, *Rijndael*, *FPGA*, ключ шифру

Вступ

В сучасному світі збереження конфіденційної інформації є актуальною проблемою. Серед різноманіття алгоритмів шифрування, один з них посідає особливе місце, *AES* (*Advanced Encryption Standard*) або *Rijndael* – схвалений *FIPS* (*Federal Information Processing Standard*) симетричний алгоритм блочного шифрування, прийнятий в якості стандарту по результатам конкурсу *AES*.

В 1998 *National Institute of Standards and Technology* представив 15 алгоритмів серед яких потрібно було обрати один, який стане послідовником застарілому алгоритму *DES*. Для нового алгоритму було висунуто такі вимоги:

– стійкість не нижча ніж у *DES*,

– швидкість шифрування не менше ніж 3*DES*,

– прозору структуру,

– ефективну програмну і апаратну реалізацію.

В жовтні 2000 року за результатами конкурсу було обрано переможцем алгоритм *Rijndael* [1] (пізніше *AES*).

Станом на 2009 рік *AES* є найбільш популярним алгоритмом блочного симетричного шифрування, а також має підтримку фірми *Intel* та використовується в процесорах x86 починаючи з *Intel Core* i7-*980X Extreme Edition*, а потім і в процесорах *Sandy Bridge*.

Головним завданням даної роботи є розгляд реалізації алгоритму роботи шифратору для підвищення швидкодії, що досягається, за допомогою апаратної реалізації шифрування *AES*.

У якості бази взято ПЛІС FPGA, яка з високою паралельністю і гнучкістю, найкраще підходить до вирішення даної задачі. Для апаратної реалізації буде використовуватись система на кристалі (SoC system) з Nios II вбудованим на Altera Cyclone II FPGA. Для дешифрування та шифрування пристрій має зчитувати дані з SD карти. Крім того, очікується, що час, за який буде відпрацьовуватись алгоритм буде значно менший, ніж аналогічна програмна реалізація.

Опис особливості алгоритму

AES зашифровує та дешифровує 128-бітні блоки даних, і дозволяє використовувати ключі трьох різних типів 128, 192 і 256 біт. В залежності від довжини ключа відрізняють три типи алгоритму (*AES*-128, *AES*-192 або *AES*-256). Процес шифрування та дешифрування складається з певної кількості ідентичних, за алгоритмом, крім останнього, раундів шифрування, число яких залежить від довжини ключа:

– довжина ключа 128 біт – 10 раундів,

– довжина ключа 192 біт – 12 раундів,

– довжина ключа 256 біт – 14 раундів.

Алгоритм *Rijndael* володіє не тільки високою криптостійкістю, а і високою

швидкістю шифрування. Час програмної реалізації на машині з частотою 2 ГГц дозволяє шифрувати данні зі швидкістю приблизно 700Мбіт/с. Такої швидкості достатньо, щоб шифрувати відео в форматі *MPEG*-2 в режимі реального часу. Але слід відзначити, що апаратні реалізації шифратора працюють значно швидше.

Безпека AES алгоритму

Для трьох варіантів ключів кількість операцій для повного перебору вимагає 2^{127}, 2^{191} і 2^{255} операцій, навіть для найменшого з цих значень метод перебору не має практичного значення. Відповідно до досліджень *AES* алгоритм шифрування стійкий до таких видів атак:

- диференціальний криптоаналіз,
- лінійний крипто аналіз,
- криптоаналіз на основі зв'язних ключів.

З 2003 АНБ США проінформував загал про те, що *AES* являється достатньо надійним для передачі державних таємниць. Для рівня *SECRET* було дозволено використовувати ключі 128-бітної довжини, а для *TOP SECRET* ключі 192- і 256 - бітної довжини.

На відміну від інших симетричних алгоритмів шифрування *AES* має простий математичний опис. Взявши за основу це твердження в 2002 другому році Николя Куртуа і Йозеф Пепшик описали теоретичну атаку названу *XLS*-атакою, яка могла б зламати *AES*, однак результати роботи не були сприйняті позитивно всіма дослідниками.

Атаки по стороннім каналам не використовують математичні особливості алгоритму, але використовують особливості реалізації системи. В квітні 2005, *Daniel J. Bernstein* опублікував роботу з описом атаки, в якій використовувалось опис часу виконання кожної операції шифрування, дана атака вимагала більше ніж 200 мільйонів вибраних шифротекстів для надходження ключа. В жовтні 2005 була опублікована робота яка отримувала ключ після 800 операцій, але вимагалась можливість для запуску атаки на тій самій машині, де і виконувалось шифрування.

В грудні 2009 року було опубліковано роботу, в якій використовувався диференційний аналіз помилок, штучно згенерованих в матриці станів на 8-му раунді шифрування, це дозволило згенерувати ключ на 2^{32} операції.

Загальна структура AES

Вхідними даними для шифрування є масив з 16 байт $in_0, in_1, \ldots in_{15}$, перед початком шифрування байти масиву поміщаються послідовно в стовбці матриці, за розмірами 4x4, *InputBlock* (зверху в низ). Також для обрахунку шифру використовується матриця станів (*State*) представлена аналогічною за розмірами, з матрицею вхідних даних. Кінцеве значення матриці станів називають матрицею *OutputBlock* яка перетворюється в послідовність байт вихідних даних. Послідовність ключа також поміщається в матрицю 4x4 *InputKey* [2]. Схематичне відношення між матрицями зображено на рисунку 1.

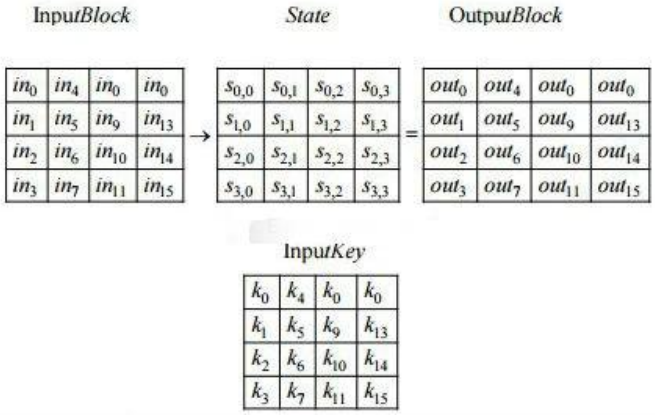

Рис.1. Відношення між матрицями

Як відмічалось ключ *AES*-128 складається з 128 бітів поділених на 16 байт $k_0, k_1...k_{15}$ і записується в стовбці матриці *InputKey*, отже фактично ключ це чотири слова $w_0w_1w_2w_3$ де $w_0 = k_0k_1k_2k_3$. З цих слів за допомогою спеціального алгоритму створюється послідовність з 44 слів($w_0 \ldots w_{43}$) і на кожний раунд шифрування подається по чотири слова описаної послідовності. На рисунку 2 можна побачити блок-схему алгоритму шифрування *AES*-128.

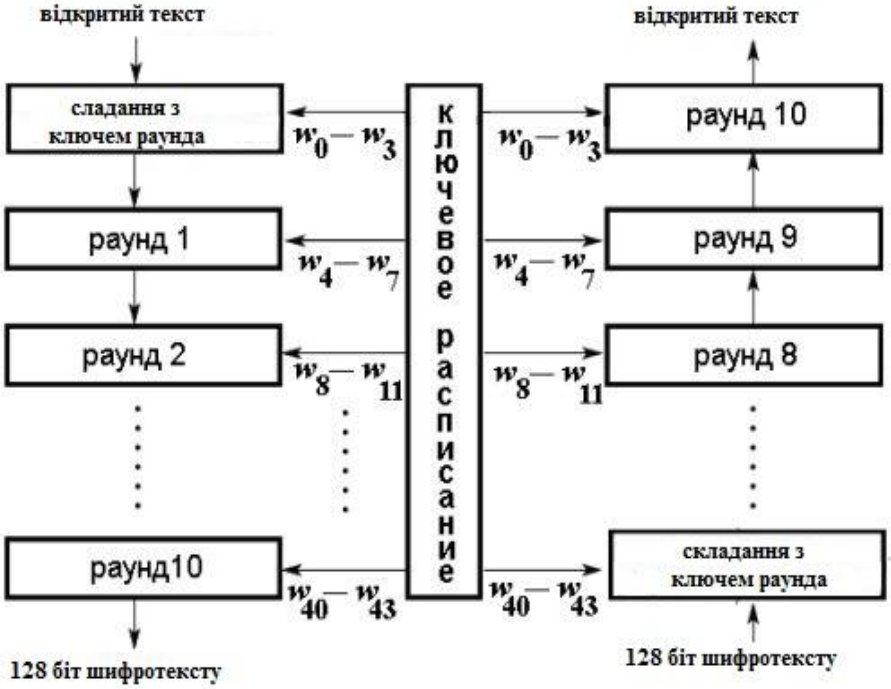

Рис. 2. Блок-схема алгоритму

Кожний раунд складається з чотирьох різних перетворень

– *SubBytes* – по байтова підстановка в матрицю станів з фіксованої таблиці замін,

– *ShiftRows* – побайтовий зсув рядків у матриці станів на різну кількість байт,

– *MixColumms* – переміщення байт в стовбцях,

– *AddRounKey* - складання з раундовим ключем (*XOR*).

Останній раунд відрізняється від інших тим що не задіює функцію *MixColumms*. Послідовність виконання раундових перетворень зображена на рисунку 3 [3].

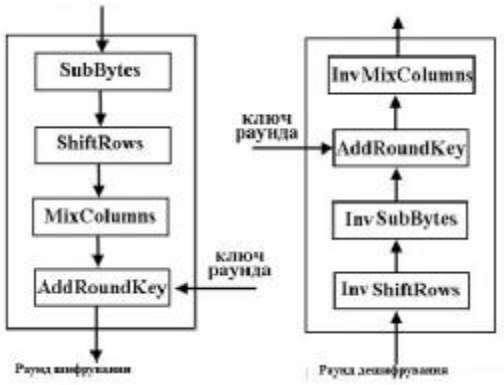

Рис. 3. Послідовність виконання раундових операцій

Вирішення поставленого завдання

Як відомо, що шифрування складається з визначеної кількості раундів однаковою між собою (крім останнього). Отже, раунд шифрування буде реалізовуватись блоком зображеним на рисунку 4 [4].

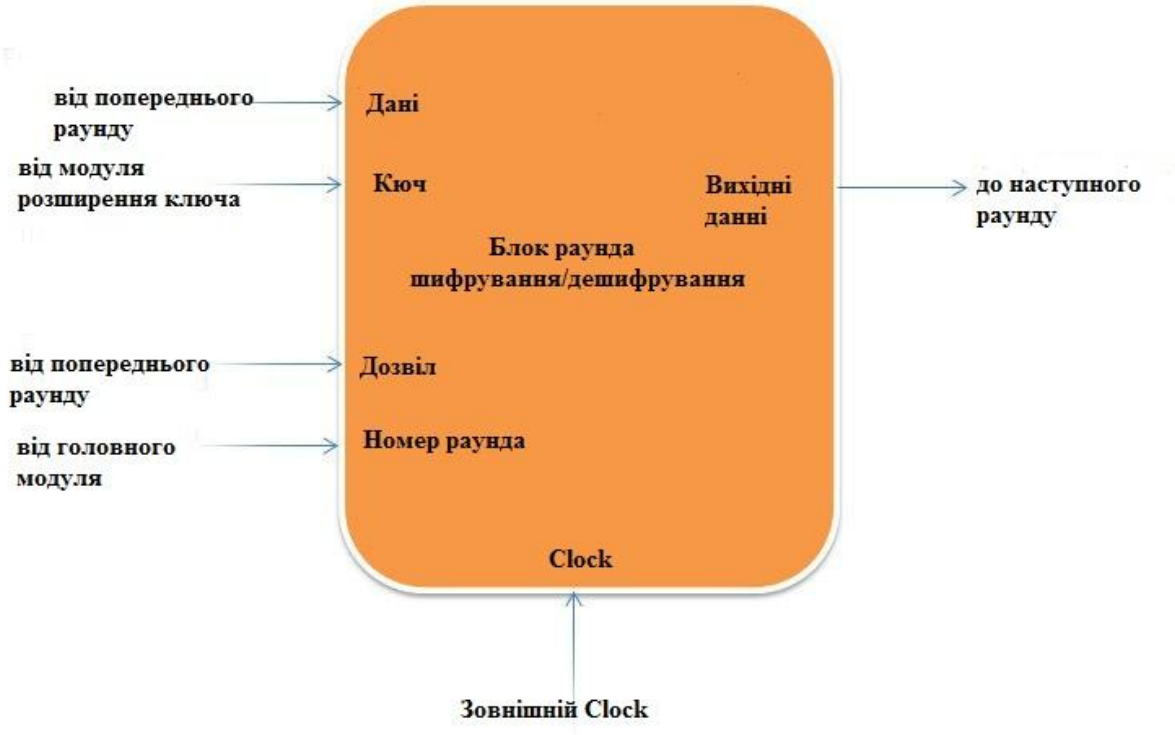

Рис.4. Одиночний блок для одного раунду шифрування

Кожний блок буде приймати дані з виходу попереднього блоку, а також номер раунду для того, щоб відрізнити останній раунд. Щоб уникнути часових невідповідностей потрібно додати часових затримок для кожного модуля. Для виведення зображення з *Nios* II на монітор, необхідно створили екземпляр контролера *VGA*. Контролер *VGA* повинен передавати дані через роз'єм *VGA* на платі *DE2*-115.

Для зв'язку *SD*-карти з ядром *Nios* будуть використані порти вводу/виводу, також для введення ключа для шифровки/дешифровки користувачем, буде підключений *JTAG UART* інтерфейс до *Nios*. Отже отримаємо таке схематичне зображення архітектури даного пристрою, зображеного на рисунку 5 [5].

Висновки

Розглянутий алгоритми роботи і архітектура пристрою буде покладено в основу створення макету пристрою шифрування даних на базі ПЛІС *Altera Cyclone II FPGA.*

Апаратна реалізація алгоритму *AES* (*Rijndael*) дозволить користуватись такими перевагами передачі даних, як швидкість та надійність, у різних системах, включаючи системи шифрування та дешифрування конфіденційної інформації, системах з електронним підписом та інших цифрових приладах.

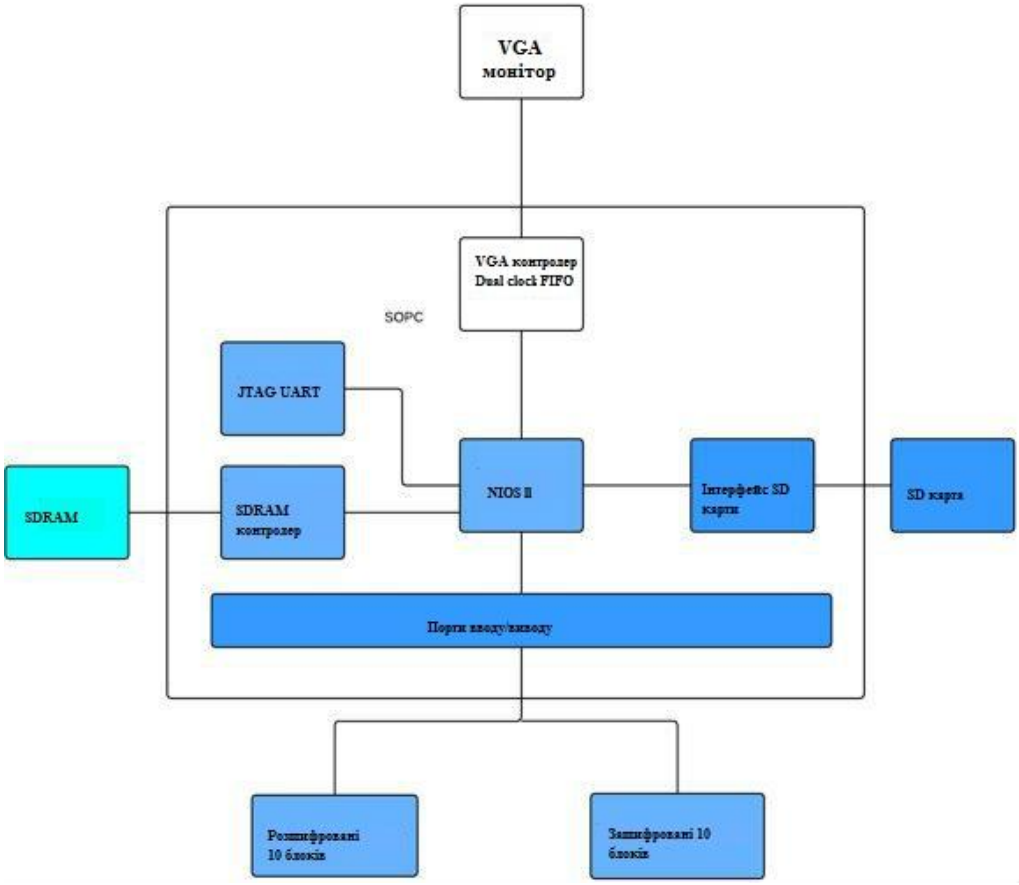

Рис.5. Структурна схема AES шифратора

Список літератури

1. J. Daemen. The Design of Rijndael: AES - The Advanced Encryption Standard / J. Daemen, V. Rijmen. - Springer Science & Business Media, 14.02.2002p. – 238p.

2. Avi Kak. Lecture 8: AES: The Advanced Encryption Standard Lecture Notes on "Computer and Network Security" – Purdue University, 01.05.2015 – 79p.

3. AES Encrypter/Decrypter [Електронний ресурс]: ECE 5760: Final Project / A. Laxminarayana, A. Ravani, M. Venkatraman — Режим доступу до статті: http://people.ece.cornell.edu/land/courses/ece5760/FinalProjects/s2015/ar856/ECE5760webpage/ECE5760%20webpage/webpage_files.html

4. In the blink of an eye: There goes your AES key [Електронний ресурс]: Cryptology ePrint Archive / Sergei Skorobogatov, Christopher Woods – 28.05.2012 – 7p. — Режим доступу до статті: https://eprint.iacr.org/2012/296.pdf

5. Kris Gaj. FPGA and ASIC Implementations of AES / Kris Gaj, Pawel Chodowiec - Springer US - 2009 – pp. 235-294.

Статтю подано до редакції 15.03.2016