УДК 004.056.55
DOI https://doi.org/10.24144/2616-7700.2021.39(2).145-151

А. О. Гедеон[1], О.М. Гапак[2]

[1] ДВНЗ «Ужгородський національний університет»,
асистент кафедри комп'ютерних систем та мереж,
hanna.hedeon@uzhnu.edu.ua
ORCID: https://orcid.org/0000-0002-5684-6932

[2] ДВНЗ «Ужгородський національний університет»,
доцент кафедри комп'ютерних систем та мереж,
кандидат педагогічних наук
oksana.hapak@uzhnu.edu.ua
ORCID: https://orcid.org/0000-0003-3448-6670

АПАРАТНА РЕАЛІЗАЦІЯ МОДУЛІВ ХЕШУВАННЯ НА БАЗІ АЛГОРИТМІВ CRC-32 І ADLER-32

У статті представлені результати дослідження хеш-функцій. Для досягнення оптимальної швидкодії та надійності захисту інформації обрана апаратна реалізація алгоритмів хешування. Саме вона гарантує цілісність розробки та виключає можливість перехоплення інформації.

Розроблено апаратний модуль хешування на основі алгоритмів *CRC*-32 і *Adler*-32, який відрізняється від існуючих розробок відсутністю мікропрограм та запрограмованих блоків. Роботою модуля керують спеціальні блоки керування, що базуються на автоматах Мура. Спроектований модуль представляє собою цілісну розробку, яка включає сукупність блоків, що відповідають за конкретні етапи обчислень. Перебачена можливість вдосконалення та додавання нових алгоритмів хешування.

Запропоновані алгоритми хешування забезпечують швидкодію обчислення контрольної суми, що в сотні разів перевищує можливості програмних додатків. Імовірність злому апаратного блоку вважається мінімальною, адже передбачає процес повного розбору пристрою на складові та прорахунок всіх можливих значень, що поступають від складових модуля.

Встановлено, що апаратна реалізація алгоритму *Adler*-32 виконує обчислення контрольної суми для вхідного повідомлення однакової довжини приблизно в 1,481 разів швидше, ніж апаратний модуль *CRC*-32.

Практична цінність отриманих у роботі результатів полягає в тому, що запропонований спосіб реалізації алгоритмів дозволяє оцінити можливості та переваги апаратних розробок, забезпечити цілісність та захищеність пристрою хешування, дослідити різницю між програмними та апаратними розробками, в тому числі й у відношенні часових затрат на проектування, та забезпечити максимальну швидкодію в обчисленні хеш-сум.

Ключові слова: контрольна сума, хешування, хеш-сума, хеш, блок керування, алгоритм, модуль, апаратний модуль, CRC, Adler.

1. Вступ. Процес пошуку даних у великих обсягах інформації пов'язаний з часовими витратами, які обумовлені необхідністю постійно переглядати та порівнювати з ключем пошуку значне число елементів. Скорочення пошуку відбувається завдяки використанню різних алгоритмів і способів, які упорядковують інформацію відповідно до тих, чи інших вимог. Поширеним методом забезпечення швидкого доступу до інформації, що зберігається в зовнішній пам'яті, є хешування.

Хеш-функції використовують для контролю цілісності файлів операційної системи, конфіденційних документів і програм; для побудови асоціативних масивів і унікальних ідентифікаторів; пошуку дублікатів у серіях наборів даних і

контрольного підсумовування з метою виявлення випадкових чи навмисних помилок при зберіганні або передачі інформації; у сфері захисту інформації в інформаційних, телекомунікаційних та інформаційно-телекомунікаційних системах [1].

Для досягнення оптимальної швидкодії і надійності захисту інформації раціонально використовувати апаратні розробки для шифрування та хешування, що гарантують цілісність розробки, забезпечують максимально можливу швидкість обробки даних і виключають можливість перехоплення інформації.

Більшість криптографічних модулів захисту даних реалізовані у вигляді спеціальних фізичних пристроїв, що під'єднують до лінії зв'язку. Апаратна реалізація дозволяє зробити процес хешування непомітним для користувача, підвищити стійкість системи до зовнішнього впливу і створити внутрішню логічну структуру, внесення мінімальних коректив в яку призводить до повного виходу з ладу системи та неможливості отримати доступ до конфіденційних даних [2].

Основоположниками теорії хеш-функцій вважаються дослідники Картер Дж. Л., Вегман М. Н., Біербрауер Ю. У перших версіях відповідні алгоритми задіяні в якості інструментарію для формування унікальних образів послідовностей символів довільної довжини з подальшою метою їх ідентифікації і перевірки на предмет достовірності [3]. До відомих алгоритмів хешування відносяться: *Adler*-32, *BTIH*, *Fletcher*, *CRC-32*, ГОСТ Р 34.11-94, сімейства *MD*, *IPEMD SHA*, *TTH*.

Стандарт *MD5 (*сімейство *MD;* 128-бітний алгоритм хешування) опублікований Рональдом Ривестом з Массачусетського технологічного інституту. В рамках проекту *MD5CRK* [4] китайські криптографи Сяоюнь Ван (*Xiaoyun Wang*), Денгуо Фен (*Dengguo Feng*), Сюецзя Лай (*Xuejia Lai*) і Хонбо Ю (*Hongbo Yu*) довели, що при знаходженні контрольних сум *MD5* зустрічаються колізії (на їх визначення витрачається від 15 секунд до двох годин). Швидкість хешування *MD5* на *IBM PS*/2 (16 *MHz* 80386) складає 1849 Кбіт/с, тоді як *SHA* – 710 Кбіт/с.

У дослідженнях авторів *RFC* 3385 [5] проведено аналіз ефективності виявлення помилок контрольних сум *Adler*-32, *IEEE*-802, *Fletcher*, *CRC-32.* Результати представлені в табл. 1, де: *d* - мінімальна відстань на блоці довжини *Block*, *Block* - довжина блоку в бітах, *i/byte* - кількість програмних інструкцій на байт, *Pudb* - ймовірність невиявлених групових помилок, *Puds* - ймовірність невиявлених одиничних помилок.

Таблиця 1.

Результати дослідження *RFC* 3385.

Алгоритм	*d*	*Block*	*i/byte*	*Pudb*	*Puds*
Adler-32	3	2^19	3	10^-36	10^-35
Fletcher-32	3	2^19	2	10^-37	10^-36
IEEE-802	3	2^16	2.75	10^-41	10^-40
*CRC*32*C*	3	2^31-1	2.75	10^-41	10^-40

Ймовірності невиявлених помилок, приведені в таблиці, обчислені при рівномірному розподілі даних та дозволяють відслідкувати суттєву різницю у відношенні алгоритмів хешування *Adler*-32 і *CRC-32.*

Метою роботи є дослідження стандартів *Adler*-32 і *CRC-32,* визначення швидкодії апаратних модулів хешування при симуляції роботи проектованих схем у пакеті *NI Multisim*, а також порівняння швидкодії формування хеш-сум при програмній і апаратній реалізації цих алгоритмів.

2. Постановка задачі. Об'єктом дослідження є апаратний модуль хешування на базі алгоритмів загального призначення *CRC*-32 і *Adler*-32.

Предметом дослідження є методи побудови апаратних модулів хешування, що забезпечують підвищення ефективності захисту інформації.

Проектовані модулі хешування на базі алгоритмів загального призначення повинні відповідати вимогам:

1) модульна архітектура;
2) цілісність розробки (відсутня можливість розділити пристрій на окремі незалежні складові для подальшого дослідження, зміни параметрів і підбору вхідних значень);
3) оптимальна швидкодія;
   (а) хешування та зберігання контрольної суми відбувається в самій мікросхемі, а не в оперативній пам'яті комп'ютера;
   (б) незалежний блок керування на базі скінченного автомата для кожного з алгоритмів;
   (в) оптимальне споживання ресурсів;
   (г) завантаження ключів і формування проміжних значень результату відбувається без використання оперативної пам'яті і системної шини комп'ютера (виключає можливість перехоплення значень).

3. Опис модулів *CRC*-32 і *Adler*-32. Апаратний модуль хешування реалізовано в пакеті *NI Multisim*, програмна реалізація алгоритмів *CRC*-32 і *Adler*-32 виконана у вигляді функцій на мові програмування *C#* в середовищі розробки *Microsoft Visual Studio*.

Алгоритм *CRC*-32[6]:

1) На регістр зберігання даних поступає вхідне слово, яке перетворюється в послідовність одиниць та нулів.
2) Блок хешування аналізує кожен біт, що поступає в відповідний регістр, та виконує обчислення з заданим поліномом.
3) Лічильник контролює функціонування блоку хешування та в разі необхідності подає сигнал дозволу на запис регістру хешу.
4) Після подання сигналу обчислення блоком хешування припиняються та регістр хешу демонструє результати – контрольну суму.

Алгоритм *Adler*-32[7]:

1) На регістр зберігання даних поступає вхідне слово, яке перетворюється в послідовність одиниць та нулів.
2) Блок хешування зчитує посимвольно вхідне слово та розділяє кожен символ (послідовність 1 та 0) на частини A та B. Кожна частина надсилається на відповідний блок для проведення обрахунків.
3) Лічильник контролює функціонування блоку хешування та в разі необхідності подає сигнал дозволу на перевірку розміру хешу.
4) Блок хешування формує хеш з отриманих частин A та B та перевіряє, чи не перевищений ліміт.

5) Блок керування передає сигнал дозволу на демонстрацію отриманої контрольної суми.

Спроектований блок хешування на базі модулів *CRC*-32 *і Adler*-32 включає наступні компоненти:

– блок керування (2 шт.);
– блок хешування (2 шт.);
– блок індикації (спеціальні дисплеї для відображення символів в 16-тковій системі числення та відображення англійського алфавіту і цифр);
– перетворювач сигналів (спеціальний пристрій, що дозволяє взаємодіяти дисплеям різних типів з вихідними сигналами регістрів);
– блок формування хешу;
– блок ідентифікації вхідного слова.

Схематична реалізація блоку приведена на рис. 1. Спроектований пристрій представляє собою цілісну розробку, яка включає сукупність блоків, що відповідають за конкретні етапи обчислень.

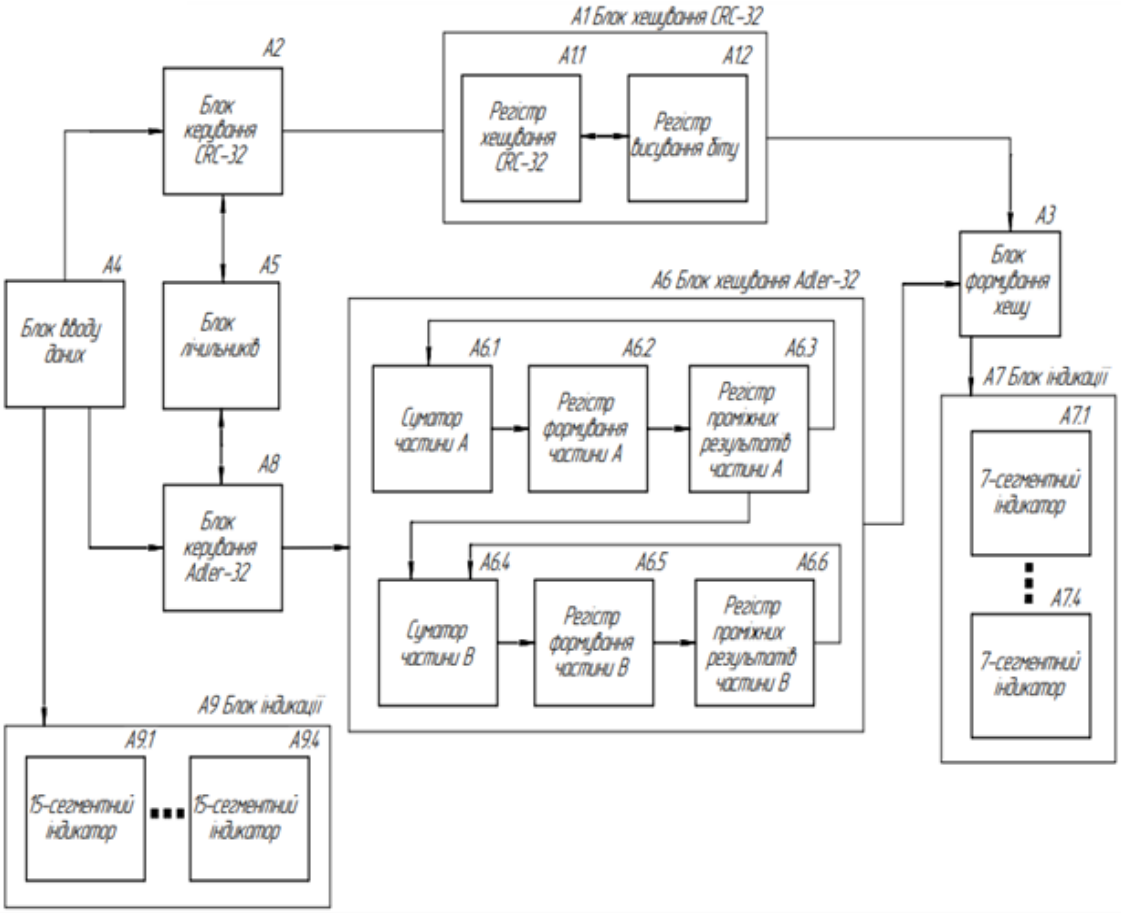

Рис. 1. Структурна схема блоку хешування

Починаючи з перших, блоки керування (далі в тексті БК, A2 та A7) повинні взаємодіяти з блоками хешування та у заданий час подавати керуючі сигнали, спроможні контролювати всю роботу апаратного модуля. Кожен БК включає

перелік дій та сигналів, що взаємодіють між собою для виконання вказаного алгоритму (*CRC*-32 або *Adler*-32).

Блоки хешування (*A*3 та A8) чітко розподіляються на окремі модулі, не пов'язані між собою (передбачається можливість розширення схеми та додавання нових блоків хешування). В залежності від того, на який з БК буде поданий сигнал запуску, буде активований відповідний модуль хешування (складається з цілого набору компонентів).

Блоки індикації (A6 та A9) для демонстрації результатів з'єднані з блоками формування хешу та вхідними регістрами, що записують початкове слово, введено користувачем. В залежності від того, де розташований дисплей і для чого він використовується, будуть встановлені різні варіанти: стандартні дисплеї *Multisim* (*DCD_HEX_DIG*) для демонстрації 16-тковій системи числення та 15-сегментні *ALPHA_NUMERIC_COM*, що призначені для демонстрації англійського алфавіту і цифр від 0 до 9. Останні підключаються до спеціальних перетворювачів сигналів, які в свою чергу – до виходів 32-бітних регістрів.

Важливим компонентом є блок ідентифікації даних (A1), котрий дозволяє ввести відповідні дані (в тому числі початкове слово та ключ, що відповідає за обрання алгоритму), блок формування хешу (A4), що зберігає кінцевий результат хешування та передає його на блоки індикації, та блок лічильників (A5), котрий відповідає за контроль обчислень та при необхідності передає сигнали блоку керування, якщо необхідно завершити розрахунок.

Апаратний модуль, що включає декілька алгоритмів хешування, можна використовувати в якості окремої мікросхеми, що під'єднується до каналу зв'язку, або як окремий пристрій для миттєвого відображення контрольної суми для введеного вхідного повідомлення. Основні завдання проектованого модуля: обчислення контрольної суми для заданого повідомлення, надання вибору двох варіантів обчислень, швидкий обрахунок та відсутність збоїв.

Проведено дослідження швидкодії обрахунків для кожного з алгоритмів, додатково визначено співвідношення швидкодії апаратних і програмних модулів та відстежино зміни швидкості при зміни тактової частоти процесора. Серія тестування для кожного алгоритму включила по 30 дослідів для вхідних слів розміром 40, 48, 56 і 64 біт згенерованих випадковим чином (загалом 480 симуляцій роботи апаратних і програмних модулів). Тестування програмних модулів проводилось на базі процесора 3-го покоління *Intel i*3 з тактовою частотою 2,4 *GHz*.

За висновками досліджень середня швидкість роботи апаратного модуля хеш-функції *CRC*-32 із заданою тактовою частотою 24 *MHz* в 1,913 рази перевищила можливості програмної реалізації. Під час тестування із тактовою частотою 240 *MHz* виявлено, що швидкість обчислень апаратного модуля перевищує можливості програмної функції в 189,5 разів.

Симуляція роботи апаратних блоків *Adler*-32 проводилась із заданою тактовою частотою в 2,4 *MHz* і 24 *MHz*. Швидкість обчислень апаратного модуля у середньому в 14,35 і 143,63 разів відповідно перевищила можливості програмної функції.

Що стосується апаратної реалізації алгоритмів *Adler*-32 і *CRC*-32, апаратний модуль *Adler*-32 виконує обчислення контрольної суми для вхідного повідомлення аналогічної довжини приблизно в 1,481 разів швидше, ніж апаратний

модуль *CRC*-32.

Результати дослідження показують, що запропоновані алгоритми хешування забезпечують швидкодію обчислення контрольної суми, яка в сотні разів перевищує можливості програмних додатків. Можливість злому апаратного блоку вважається мінімальною, адже передбачає процес повного розбору пристрою на складові та розрахунок всіх можливих значень, що надходять від складових модуля.

4. Висновки. Апаратна реалізація алгоритмів загального призначення гарантує цілісність розробки: після змінення параметрів пристрою блок повністю виходить з ладу, адже будь-які корективи впливають на результати обчислень та не дозволяють підібрати вхідне слово. Відсутність підпрограм компенсується блоками управління на базі скінченних автоматів (автоматів Мура), що також підвищують швидкодію і надійність апаратного модулю в порівнянні з програмними аналогами.

Під час розробки апаратного блоку передбачена можливість вдосконалення загальної схеми, шляхом додавання нових алгоритмів хешування, для яких зарезервовані бітові комбінації, що надходять на блок вводу даних. Це дозволяє доповнити загальну схему додатковими алгоритмами хешування, що будуть працювати незалежно від наявних модулів.

Практична цінність отриманих в роботі результатів полягає в тому, що запропонований спосіб реалізації алгоритмів дозволяє оцінити можливості та переваги апаратних розробок, забезпечити цілісність та захищеність пристрою хешування, дослідити різницю між програмними та апаратними розробками, в тому числі й у відношенні часових затрат на проектування, та забезпечити максимальну швидкодію в обчисленні хеш-сум.

Список використаної літератури

1. Вишня В. Б., Гавриш О. С., Рижков Е. В. Основи інформаційної безпеки: навч. посіб. Дніпро: ДДУВС, 2020. 128 с.
2. Варлатая С. К., Шаханова М. В. Криптографические методы и средства обеспечения информационной безопасности: учебно-метод. комп. Владивосток: ДВФУ, 2015. 143 с.
3. Основы криптографии: учебное пособие. 2-е Изд. 2-е, испр. и доп. / Алферов А. П., Зубов А. Ю., Кузьмин А. С., Черемушкин А. В. Москва: Гелиос АРВ, 2002. 420 с.
4. Collisions for Hash Functions MD4, MD5, HAVAL-128 and RIPEMD / Xiaoyun W., Dengguo F., Xuejia L., Hongbo Y. International Association for Cryptologic Research, 2004. P. 4.
5. Internet Protocol Small Computer System Interface (iSCSI) Cyclic Redundancy Check (CRC). Checksum Considerations / Sheinwald D., Satran J., Thaler P., Cavanna V. 2002. P. 23.
6. Castagnoli G., Braeuer S., Herrman M. Optimization of Cyclic Redundancy-Check Codes with 24 and 32 Parity Bit. EEE Transactions on Communications. T. 41, № 6, 1993. P. 883-892.
7. Maxino T. Revisiting Fletcher and Adler Checksums. Carnegie Mellon University. Pittsburgh: DSN 2006 Student Forum, 2006. P. 3.

Hedeon A. O., Hapak O. M. Hardware implementation of hashing modules based on algorithms CRC-32 and Adler-32.

The article presents the results of the study of hash functions. To achieve optimal speed and reliability of information protection, the hardware implementation of hashing algorithms is chosen. It guarantees the integrity of the development and excludes the possibility of interception of information.

A hardware hashing module based on CRC-32 and Adler-32 algorithms has been developed, which differs from existing developments by the absence of micro program and

programmed blocks. The operation of the module is controlled by special control units based on Moore machine. The designed module is a holistic development, which includes a set of blocks responsible for specific stages of calculations. The possibility of improving and adding new hashing algorithms is provided.

The proposed hashing algorithms provide the speed of calculating the checksum, which exceeds a hundred times the capabilities of software applications. The probability of hacking the hardware unit is considered minimal, because it involves the process of complete disassembly of the device into components and the calculation of all possible values coming from the components of the module.

It has been found that the hardware implementation of the Adler-32 algorithm performs a checksum calculation for an incoming message of the same length approximately 1,481 times faster than the CRC-32 hardware module.

The practical value of the obtained results in the work is that the proposed method of algorithms implementation allows to assess the capabilities and benefits of hardware development, ensure the integrity and security of the hashing device, investigate the difference between software and hardware development, including the time spent on design, and provide maximum speed in calculating of hash sums.

Keywords: checksum, hashing, hash sum, hash, control unit, algorithm, module, hardware module, CRC, Adler.

References

1. Vyshnia, V. B., Gavrish, O. S., & Rizhkov, E. V. (2020). Basics of information security: textbook. Dnipro: DDUVS.
2. Varlataya, S. K., & Shakhanova, M. V. (2015). Cryptographic methods and means of ensuring of information security: textbook. Vladivostok: FEFU.
3. Alferov, A. P., Zubov, A. Yu., Kuzmin, A. S., & Cheremushkin, A. V. (2002). Fundamentals of cryptography: textbook, 2nd ed., Rev. and add. Moscow: Helios ARV.
4. Xiaoyun, W., Dengguo, F., Xuejia, L., & Hongbo, Y. (2004). Collisions for Hash Functions MD4, MD5, HAVAL-128 and RIPEMD. *International Association for Cryptologic Research.*
5. Sheinwald, D., & Satran, J. (2002). Internet Protocol Small Computer System Interface (iSCSI) Cyclic Redundancy Check (CRC). *Checksum Considerations.*
6. Castagnoli, G., Braeuer, S., & Herrman, M. (1993). Optimization of Cyclic Redundancy-Check Codes with 24 and 32 Parity Bits. *EEE Transactions on Communications, 41*(6), 883-892.
7. Maxino, T. (2006). Revisiting Fletcher and Adler Checksums. *Carnegie Mellon University. Student Forum.* Pittsburgh: DSN.

Одержано 27.09.2021