УДК 681.3

д.т.н., професор Дичка І.А., магістрант Онай М.В.

**Національний технічний університету України
«Київський політехнічний інститут»**

СПОСОБИ ЗНАХОДЖЕННЯ МУЛЬТИПЛІКАТИВНОГО ОБЕРНЕНОГО ЕЛЕМЕНТА В СКІНЧЕННИХ ПОЛЯХ

Abstract

Ivan A. Dychka, prof., DSc; Mykola V. Onai, student
Methods of inverse multiplicative element calculation in finite fields

Two methods of inverse multiplicative element calculation and their hardware realization are represented. Comparison of their hardware costs and speed capability is also performed.

Вступ

Використання апаратних засобів для розв'язування спеціалізованих задач дозволяє підвищити продуктивність комп'ютерної системи.

Операції в скінченних полях використовуються при реалізації різноманітних криптографічних протоколів та алгоритмів, а також знаходять широке застосування у завадостійкому кодуванні [1].

Постановка задачі

Однією з найбільш складних операцій у скінченних полях є знаходження (обчислення) мультиплікативного оберненого елемента.

Тому актуальною є задача апаратної реалізації операції знаходження мультиплікативного оберненого елемента в скінченному полі.

Теоретичні відомості

Для будь-якого цілого числа $a > 0$ взаємно простого з модулем n існує обернене за модулем n число, що позначається a^{-1}, таке що $a \times a^{-1} \equiv 1 \bmod n$. Число a^{-1} називається мультиплікативно оберненим за модулем n [2, 3].

Якщо модулем є просте число (позначимо його p), то мультиплікативний обернений елемент за модулем p існує для будь-якого числа $0 < a < p$ [2, 3].

Скінченне поле, кількість елементів якого є простим числом, позначають $GF(p)$, де p – просте число. Елементами такого поля є числа $\{0, 1, 2, ..., p-1\}$, а операції в ньому виконують за модулем простого числа p.

Стандартним способом знаходження оберненого елемента в полі $GF(p)$ є множення відповідного елемента (для якого шукається обернений) почергово на всі елементи поля $GF(p)$ поки не буде отримано одиничний елемент.

Розглянемо множину $L = \{1, 2, ..., p-1\}$ ненульових елементів поля. Візьмемо будь-який елемент $a \in L$. Помножимо кожен з елементів множини L на a за модулем p і отримаємо множину $M = \{a \bmod p, (2a) \bmod p, ..., ((p-1)a) \bmod p\}$. Очевидним є те, що $L = M$. Множини M та L є ізоморфними з точністю до перестановки елементів.

Наприклад, якщо $p = 5$, то поле $GF(5)$ складається з таких елементів $\{0, 1, 2, 3, 4\}$, а $L = \{1, 2, 3, 4\}$.

Нехай $a = 3$. Тоді $M = \{1 \cdot 3 \bmod 5,\ 2 \cdot 3 \bmod 5,\ 3 \cdot 3 \bmod 5,\ 4 \cdot 3 \bmod 5\} = \{3, 1, 4, 2\}$, отже $L = M$.

Множину M можна представити у такому вигляді:

$$M = \left\{ a \bmod p,\ (a+a) \bmod p,\ (a+a+a) \bmod p, ...,\ \left(\underbrace{a+a+...+a}_{p-1} \right) \bmod p \right\} \quad (1)$$

Нехай $a, b \in M$, де a – перший елемент множини M. Елемент b є оберненим до елемента a (тобто $a^{-1} = b$), якщо $ab = 1$:

$$ab = 1 \Leftrightarrow -1 + ab = 0 \Leftrightarrow \underbrace{p-1}_{-1} + ab = 0 \quad (2)$$

У виразі (2) операції виконуються за модулем p. Для пошуку значення елемента b необхідно почергово замість b підставляти елементи множини L і обчислювати вираз (2) та перевіряти чи буде він дорівнювати нулю.

Вираз (2) складається з двох компонент (доданків). Другою компонентою цього виразу є добуток, а операція множення, як відомо, є складною.

Реалізуємо множення через додавання. Нехай елемент з індексом s (s-й елемент) множини M є одиничним (такий елемент у

множині M завжди існує). Тоді $\left(p-1+\underbrace{a+a+...+a}_{s}\right) \bmod p = 0$ або $(p-1+as) \bmod p = 0$. Це означає, що s є оберненим елементом до a. А оскільки для $\forall a \in M$ в M існує лише один обернений елемент, то $s = b$ (*див* (2)).

Отже, для знаходження елемента оберненого до a необхідно обчислити s-й елемент множини M за формулою (1).

Реалізація операції знаходження оберненого елемента

Реалізуємо цей спосіб апаратно (*рис*. 1).

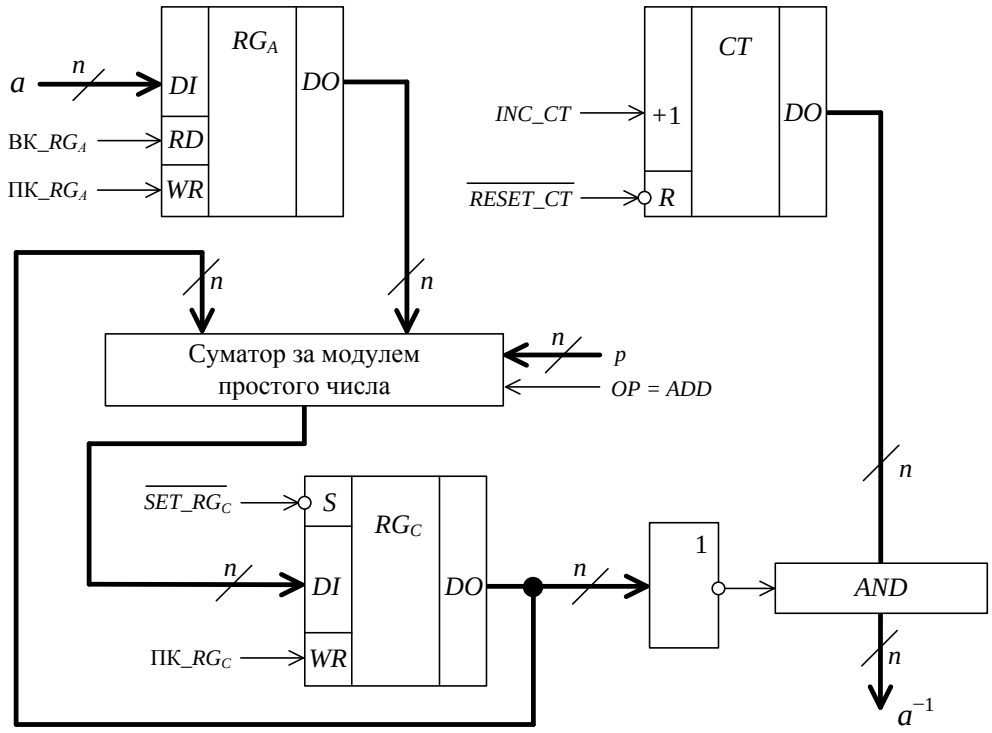

Рис. 1. Структурна схема блока обчислення оберненого елемента

Алгоритм роботи блока обчислення оберненого елемента:

1. $RG_C := p - 1$; $CT := 0$;
2. $RG_C := (RG_C + a) \bmod p$; *INC_CT*;
3. Якщо $(RG_C) = 0$, то $a^{-1} = (CT)$, інакше перейти до п.2.

Регістр A (RG_A) призначений для зберігання значення елемента a, для якого шукаємо обернений. В регістрі C (RG_C) відбувається накопичення результату. Початковим станом регістра C є $p - 1$. Після кожної операції

додавання відбувається інкремент лічильника, таким чином в лічильнику буде записано кількість виконаних операцій додавання.

Після отримання на виході регістра C нульового результату в лічильнику CT буде записане число, яке є оберненим до a, тобто $CT = a^{-1}$.

Максимальна кількість операцій для знаходження оберненого елемента дорівнює $3p - 3$, а середня – $t_1 = 0{,}875p - 0{,}875$.

Знаходження оберненого елемента за допомогою розширеного алгоритму Евкліда

Відома теорема про *НСД* [3], з якої випливає, що для будь-якого невід'ємного цілого числа a та будь-якого додатного цілого b справедливе відношення $НСД(a, b) = НСД(b, a \bmod b)$. На основі цієї теореми побудований алгоритм Евкліда.

Існують два різновиди алгоритму Евкліда.

Розглянемо модифікацію розширеного алгоритму Евкліда, за допомогою якого можна знаходити обернений елемент у полі $GF(p)$.

Для зберігання даних при апаратній реалізації необхідно шість регістрів. Назвемо їх u_1, u_2, v_1, v_2, h_1 та h_2.

Алгоритм складається з наступних кроків:

1. Ініціалізація. Присвоїти $(u_1, u_2) \leftarrow (0, p)$, $(v_1, v_2) \leftarrow (1, a)$.
2. Обчислити $q \leftarrow [u_2 / v_2]$ та $r \leftarrow \bmod (u_2 / v_2)$.
3. Якщо $r = 0$, то видати код на вихід регістра v_1 (це і буде результатом) і завершити виконання алгоритму, інакше перейти до п.4.
4. Виконати наступні операції:
   $h_1 \leftarrow u_1 - v_1 q$,
   $h_2 \leftarrow r$,
   $(u_1, u_2) \leftarrow (v_1, v_2)$,
   $(v_1, v_2) \leftarrow (h_1, h_2)$.
   Перейти до п.2.

Розглянемо приклад.

Нехай $a = 4$, $p = 23$.

З наведеної цифрової діаграми (*табл.* 1) бачимо, що найбільшим спільним дільником числа a, яке дорівнює 4 і модуля p, який дорівнює 23, є 1, а елементом оберненим до 4 є 6.

Табл. 1. Цифрова діаграма роботи розширеного алгоритму Евкліда

q	r	u_1	u_2	v_1	v_2
–	–	0	23	1	4
5	3	1	4	-5	3
1	1	-5	3	6	1
3	0	–	–	–	–

Максимальна кількість ітерацій даного алгоритму дорівнює $2\lfloor \log_2 p \rfloor$, а максимальна кількість операцій дорівнює $p\lfloor \log_2 p \rfloor + 5\lfloor \log_2 p \rfloor + 4\lfloor \log_2 p \rfloor^2$. Середня кількість ітерацій даного алгоритму дорівнює $\lfloor \log_2 p \rfloor$, а середня кількість операцій $t_2 = 0{,}25p\lfloor \log_2 p \rfloor + 2{,}75\lfloor \log_2 p \rfloor + 1{,}5\lfloor \log_2 p \rfloor^2$.

Висновки

Оцінюючи апаратні та часові витрати ($t_1 < t_2$) для кожного з двох запропонованих способів знаходження оберненого мультиплікативного елемента у скінченному полі, бачимо, що перевагу має перший спосіб (*рис*. 1) – знаходження оберненого елемента шляхом самопідсумовування (додавання до самого себе) заданого елемента.

Подальші дослідження слід зосередити на розробці швидких алгоритмів апаратної реалізації операцій піднесення до степеня та знаходження дискретного логарифма, оскільки за допомогою цих двох операцій також можна реалізувати операцію знаходження оберненого елемента.

Література

1. *Блейхут Р.* Теория и практика кодов, контролирующих ошибки. – М.: Мир, 1986. – 576 с.
2. *Кормен, Томас Х., Лейзерсон, Чарльз И., Ривест, Рональд Л., Штайн, Клифорд*. Алгоритмы: построение и анализ, 2-е издание.: Пер. с англ. – М.: Издательский дом «Вильямс», 2009. – 1296 с.
3. *Кнут Дональд Эрвин* Искусство программирования, том 2. Получисленные методы, 3-е изд.: Пер. с англ. – М.: ООО “И.Д. Вильямс”, 2007. – 832 с.